UNIVERSITÉ CATHOLIQUE DE LOUVAIN

Distances, centralities and model estimation methods based on randomized shortest paths for network data analysis

Author:
Ilkka KIVIMÄKI

Advisor:
Prof. Marco SAERENS

Thesis submitted in fulfillment
of the requirements for the degree of

Doctor of Engineering Science

of the Université catholique de Louvain

PhD jury:

President: Prof. Charles Pecheur
Prof. Michel Verleysen
Prof. Gianluca Bontempi
Prof. François Fouss
Dr. Bram Van Moorter

"The most that can be expected from any model is that it can supply a useful approximation to reality: All models are wrong; some models are useful."

G.E.P. Box, J.S. Hunter, W.G. Hunter [31, page 440]

Abstract

The emergence of networks and network data in different forms in the near past has given rise to development of new data analysis methods with a shift in focus from vector spaces to graph topologies. While some of these methods have been designed specifically for analyzing networks, others have been transformed from more traditional methods, for instance from the machine learning and data mining literature, to function also with networks or graph-based data. The main focus in the thesis is on the recently developed randomized shortest paths (RSP) framework, from which multiple interesting graph-based measures can be derived. These measures can be used either for analyzing networks directly or as tools augmenting existing machine learning or data mining methods. Essentially, the RSP framework defines a probability distribution on the set of paths connecting two nodes on a graph. The distribution focuses mostly on shortest paths but also assigns non-zero probability to longer connections between the two nodes. The extent of this spread of the probability mass is controlled by a temperature parameter, inspired by a thermodynamical analogy. From this distribution several meaningful quantities can be derived and computed conveniently, which have interesting connections with more traditional network measures. As one highlight, the thesis presents the free energy distance, which is a novel metric that is appealing theoretically in many ways, and has been shown to provide desirable and competitive results in different network analysis tasks compared with other measures. In addition, different measures for quantifying the centrality of a node in a network or the overall functionality of a network are proposed based on the RSP framework. The theory of RSPs is also extended with model estimation methods for fitting the RSP model to a dataset of trajectories on a network. In order to demonstrate and evaluate the derived methods the thesis deals with multiple application domains, ranging from the evaluation of ecological landscapes and quantifying centrality in an urban street network to clustering of text document collections. On the other hand, the methods are developed in a general theoretical setting and also discussed from different viewpoints in order to enable their use in other possible domains.

Acknowledgements

The process of completing this thesis has been a long one, and so is the catalogue of people who deserve credit for it being complete and accepted.

First of all, I thank my advisor, Prof. Marco Saerens, who provided the backbone of my research with the ideas and the theory that he developed. I could also always rely on his vast knowledge of the literature in almost any topic relevant for our research. While we were still in the same country, he was always present for discussions and help. Even later, when I was living across the globe, he would still take the time to write to me lengthy emails providing guidance.

I also thank the members of my thesis jury: Prof. Charles Pecheur for leading the jury, Prof. Michel Verleysen and Prof. Gianluca Bontempi for great remarks and improvements for my work, already in the early stages of my studies, Prof. François Fouss who, in addition to some great ideas, also had a sharp eye for detecting smaller issues, and Dr. Bram Van Moorter who provided vital feedback on the ecological applications of the thesis.

Bram also deserves separate credit for inspiring me during our many discussions with his excitement about the usefulness of my research in ecology. Also, thanks to Dr. Manuela Panzacchi, who, along with Bram, originally contacted us for collaboration resulting in many of the ideas and findings presented in the thesis. I look forward to continuing the collaboration and to someday ultimately saving the reindeer!

I am also extremely grateful to Prof. Jari Saramäki, who invited me to join the Complex Systems research group at Aalto University after my relocation to Finland. His help made it possible for me to continue the PhD. In addition, at Aalto I got a great experience meeting and collaborating with a big group of wonderful people.

I also want to thank all the other colleagues and co-authors that I worked with for the articles that were written during my PhD studies, especially Prof. Masashi Shimbo, Dr. Bertrand Lebichot, Dr. Alexander Panchenko, Dr. Dries Verdegem, Dr. Pascal Francq, Prof. Cédrick Fairon, Prof. Hugues Bersini, Dr. Robin Devooght, Dr. Amin Mantrach, Prof. Fabrice Rossi, Kevin Françoisse, and Dr. Silvia García Díez.

Thanks also to all my work friends at the Louvain School of Management, at the IRIDIA Laboratory at Université Libre de Bruxelles, and at the Complex Systems research group at Aalto University, for great working environments and a lot of good times.

Part of the work for this thesis was carried out during the project Elis-IT funded by *Région Wallonne* and the project RenewableReindeer funded by the *Norwegian Research Council*. In addition, funding from the *Academy of Finland*, projects n:o 297195 and 295499 is acknowledged. The thesis work was also supported by grants from the *Emil Aaltonen Foundation* and the *Finnish Foundation for Technology Promotion*. I thank all the funding institutions for the opportunity to carry out fundamental and applied research.

I also thank my parents for never doubting or questioning my plans concerning the PhD, or actually ever doubting or questioning any of my plans whatsoever. I think that partly made me also trust that I can have this done. Thanks also to all other family and friends for support and for engaging me in often fruitful and inspiring conversations about my research.

Hanne, my darling wife, thank you for the infinitude of support and encouragement. If it wasn't for you this thesis would not exist, for so many reasons.

And finally, I want to thank — and should perhaps apologize to — our daughter Matilda, who only at the last phases of the process realized that I was working on a doctoral thesis ("väitöskirja", in Finnish) instead of a colouring book ("värityskirja"). I still cannot decide which is more heartbraking, the fact that you were maybe expecting to get a colouring book, or that you thought I had to decline all those chances of playing with you, in order to instead work on a colouring book. Well, here is the book. You can of course colour it, if you want. And when you are a bit older, perhaps you will appreciate it in other ways.

Contents

List of Figures

List of Tables

List of Symbols

G	a graph (usually a directed, strongly connected graph)
V	node set of graph G
E	edge set of graph G
s, t, i, j, etc.	node indices
$Succ(i)$	set of successors of node i
(i, j), etc.	edge (i, j) of graph G
$\mathbf{x}$, etc.	(column) vector
$\mathbf{X}$, etc.	matrix
$(\mathbf{X})_{ij}$	element (i, j) of matrix $\mathbf{X}$
$\mathbf{x}_i^{\mathrm{c}}$	column i of matrix $\mathbf{X}$ as a column vector
$\mathbf{x}_i^{\mathrm{r}}$	row i of matrix $\mathbf{X}$ as a column vector
$\mathbf{Diag}(\mathbf{X})$	diagonal matrix consisting of the diagonal of matrix $\mathbf{X}$
$\mathbf{diag}(\mathbf{X})$	vector of diagonal elements of matrix $\mathbf{X}$
$\exp(\mathbf{X})$	elementwise exponential of matrix $\mathbf{X}$
$\log(\mathbf{X})$	elementwise logarithm of matrix $\mathbf{X}$
$\mathbf{log}(\mathbf{X})$	matrix logarithm of $\mathbf{X}$
$\mathbf{X} \circ \mathbf{Y}$	elementwise product of matrices $\mathbf{X}$ and $\mathbf{Y}$
$\mathbf{X} \div \mathbf{Y}$	elementwise division of matrices $\mathbf{X}$ and $\mathbf{Y}$
$\mathbf{I}_{ij}$	matrix with 1 at element (i, j) and 0 elsewhere
a_{ij}	affinity of edge (i, j)
$\mathbf{A}$	affinity matrix
p_{ij}^{rw}	unbiased random walk transition probability of edge (i, j)
$\mathbf{P}^{\mathrm{rw}}$	unbiased random walk transition probability matrix
c_{ij}	cost of edge (i, j)
$\mathbf{C}$	cost matrix
$\mathbf{L}$	Laplacian matrix
$\mathbf{L}^+$	Moore-Penrose pseudoinverse of Laplacian matrix
$\mathrm{vol}(G)$	volume of graph G
$\wp$	a path on graph G
$\wp_{st}$	a path from s to t
$\wp_{st}^*$	a least cost path from s to t
$\mathcal{P}$	set of all hitting paths on G
$\mathcal{P}_{st}$	set of hitting paths from node s to node t

$\widetilde{\mathcal{P}}$	set of all (regular) paths on G
$\widetilde{\mathcal{P}}_{st}$	set of regular paths from node s to node t
$\mathcal{P}^{(k)}$, $\mathcal{P}^{(k)}_{st}$	sets of paths of fixed length, k
$L(\wp)$	length of (i.e. number of edges along) path $\wp$
$\wp(i)$	the i-th node of path $\wp$ (where $i = 0, 1, \ldots, L(\wp)$)
$\wp(i-1, i)$	the i-th edge of path $\wp$
$\wp(i:j)$	the subpath of $\wp$ from the i-th to the j-th node
$\tilde{c}(\wp)$	cost of path $\wp$
$\tilde{c}^*_{st}$	least cost from s to t
$\eta_{ij}(\wp)$	number of visits over edge (i, j) along path $\wp$
$\eta_i(\wp)$	number of visits to node i along path $\wp$
$\mathrm{P}^{\mathrm{rw}}_{st}$	natural random walk probability distribution over $\mathcal{P}_{st}$
P_{st}, $\mathrm{P}^{\mathrm{RSP}}_{st}$	RSP probability distribution over $\mathcal{P}_{st}$
P^*_{st}	shortest path likelihood distribution over $\mathcal{P}_{st}$
$\langle L \rangle^{\mathrm{rw}}_{st}$	expected length of hitting paths from s to t over $\mathrm{P}^{\mathrm{rw}}_{st}$
$\langle L \rangle_{st}$	expected length of hitting paths from s to t over P_{st}
$\langle \tilde{c} \rangle^{\mathrm{rw}}_{st}$	expected cost of hitting paths from s to t over $\mathrm{P}^{\mathrm{rw}}_{st}$
$\langle \tilde{c} \rangle_{st}$	expected cost of hitting paths from s to t over P_{st}
$\mathbb{J}(\mathrm{P}_{st} \Vert \mathrm{P}^{\mathrm{rw}}_{st})$	relative entropy of P_{st} with respect to $\mathrm{P}^{\mathrm{rw}}_{st}$
$\mathcal{L}$	Lagrangian function (also likelihood function)
λ, μ	Lagrangian parameters
T	temperature
$\beta = 1/T$	inverse temperature
w_{ij}	likelihood of edge (i, j)
$\widetilde{w}(\wp)$	likelihood of path $\wp$
$\mathbf{W}$	likelihood matrix
$\widehat{\mathbf{W}}$	absorbing likelihood matrix with a row t set to zero
$\rho(\mathbf{X})$	spectral radius of matrix $\mathbf{X}$
$\mathcal{Z}_{st}$	partition function of hitting paths from s to t
$\mathcal{Z}$	partition function of hitting paths on G
$\widehat{\mathbf{Z}}$	fundamental matrix of hitting paths to a node t
$\mathbf{Z}$	fundamental matrix of regular paths
$\mathbf{Z}_{\mathrm{h}}$	fundamental matrix of hitting paths
L^*_{st}	shortest path length from s to t
$\tilde{c}^*_{st}$	least cost from s to t
$\langle L \rangle^{\mathrm{rw}}_{st}$	expected hitting time from s to t over $\mathrm{P}^{\mathrm{rw}}_{st}$
$\langle \tilde{c} \rangle^{\mathrm{rw}}_{st}$	expected hitting cost from s to t over $\mathrm{P}^{\mathrm{rw}}_{st}$
$\langle L \rangle_{st}$	expected hitting time from s to t over P_{st}
$\langle \tilde{c} \rangle_{st}$	expected hitting cost from s to t over P_{st}
ϕ_{st}	free energy from s to t over distribution P_{st}
$\bar{\eta}^{\mathrm{rw}}_{ij}(s, t)$	expected number of visits over edge (i, j) over $\mathrm{P}^{\mathrm{rw}}_{st}$

$\bar{\eta}_{ij}(s,t)$	expected number of visits over edge (i,j) over P_{st}
$\mathbf{N}^{ij}$	matrix with $\bar{\eta}_{ij}(s,t)$ at element (s,t)
$\bar{\eta}_i^{\mathrm{rw}}(s,t)$	expected number of visits to node i over $\mathrm{P}_{st}^{\mathrm{rw}}$
$\bar{\eta}_i(s,t)$	expected number of visits to node i over P_{st}
$\Delta(s,t), \Delta_{st}$	distance between s and t
$\Delta_{st}^{\mathrm{LC}}, \Delta_{st}^{\mathrm{SP}}$	least cost (=shortest path) distance between s and t
$\Delta_{st}^{\mathrm{uSP}}$	unweighted shortest path distance from s to t
$\Delta_{st}^{\mathrm{Res}}$	resistance distance between s and t
$\Delta_{st}^{\mathrm{CT}}$	commute time distance between s and t
$\Delta_{st}^{\mathrm{CC}}$	commute cost distance between s and t
$\Delta_{st}^{\mathrm{Euc}}$	Euclidean distance between s and t
$\Delta_{st}^{\mathrm{RSP}}$	RSP dissimilarity between s and t
$\Delta_{st}^{\mathrm{FE}}$	free energy distance between s and t
$\Delta_{st}^{\mathrm{SP-CT}}$	SP-CT combination distance between s and t
$\Delta_{st}^{\mathrm{logFor}}$	logarithmic forest distance between s and t
$\Delta_{st}^{p\mathrm{Res}}$	p-resistance distance between s and t
$\mathcal{W}(G)$	Wiener index of G
$\mathcal{K}(G)$	Kirchhoff index of G
$\overline{\mathbf{C}}$	matrix of expected costs
$\mathbf{\Phi}$	matrix of free energies
Φ_t	vector of free energies from all nodes to node t
Φ_Ω	matrix of free energies from all nodes to set of nodes Ω
$\mathbf{\Delta}^{\mathrm{RSP}}$	matrix of RSP dissimilarities
$\mathbf{\Delta}^{\mathrm{FE}}$	matrix of free energy distances
$\mathbf{K}$	kernel, or similarity matrix
λ	parameter of SP-CT combination distance
α	parameter of logarithmic forest distance
p	parameter of p-resistance distance
$\bar{\Delta}_{\mathcal{C}_k}$	mean distance within cluster $\mathcal{C}_k$
$\bar{\Delta}_{\mathcal{C}_k,(\Omega\setminus\mathcal{C}_k)}$	mean distance from points inside cluster $\mathcal{C}_k$ to points outside the cluster
$w(\mathcal{C}_k,\mathcal{C}_l)$	total weight of links between communities $\mathcal{C}_k$ and $\mathcal{C}_l$
Q	modularity of a partition of graph nodes
$\mathbf{Q}_{\mathrm{BoP}}$	bag-of-paths (BoP) modularity
P^{rw}	natural random walk probability distribution over $\mathcal{P}$
$\mathrm{P}^{\mathrm{BoP}}$	BoP path probability distribution over $\mathcal{P}$
Π_{ij}	BoP probability
$\mathbf{\Pi}$	matrix of BoP probabilities
δ_{ij}	Kronecker delta
$\mathrm{bet}_i^{\mathrm{RSP}}$	simple RSP betweenness of node i
$\mathrm{bet}_i^{\mathrm{net}}$	RSP net betweenness of node i

$\boldsymbol{\pi}$	stationary distribution on G
C_i^{Close}	closeness centrality of node i
C_i^{Bet}	shortest path betweenness centrality of node i
C_i^{LikeBet}	shortest path likelihood betweenness of node i
$p_{ij}^{(t)}$	transition probability of edge (i,j) of a biased random walk to node t
Ω_{st}	set of trajectories from s to t
$\mathcal{L}$	likelihood function (also Lagrangian function)
ρ	random variable corresponding to drawing a trajectory
ε	r.v. corresponding to observation of an edge on a trajectory
ν	r.v. corresponding to observation of a node on a trajectory
μ	r.v. corresponding to number of observations (also Lagrangian parameter)
$\eta_{e_1 \rightsquigarrow e_2}(\wp)$	number of times edges e_1 and e_2 appear on $\wp$ in that order
$\eta_{\tilde{e}}(\wp)$	number of times edges in sequence $\tilde{e}$ appear on $\wp$ in that order
$\hat{\beta}_{\mathrm{MLE}}$	maximum likelihood estimate of β
$\mathbb{E}_{st}[\bar{c}]$	expected average edge cost of paths in $\mathcal{P}_{st}$ w.r.t. P_{st}
HF_s	habitat functionality of node s
LF_G	landscape functionality of graph G
K_{st}	proximity of nodes s and t
Q_i	quality of node i
C_i^{Harm}	harmonic centrality of node i
$C_i^{\mathrm{h-RSP}}$	harmonic RSP centrality of node i

Chapter 1

Introduction

1.1 General introduction

Machine learning, pattern recognition and data mining methods have for a large part been based on vector space models, which embed the data elements as vectors, or points, in a Euclidean space [202, 21, 7, 61, 145]. This embedding can often be produced by a simple association between the coordinate axes of the space and the features of the data. Manipulating and analyzing the data in this space can then be achieved by using the vast toolboxes of linear algebra, differential geometry, probability theory, multivariate statistics, topology and other traditional and well-studied mathematical and statistical disciplines.

The somewhat sudden appearance and abundance of data about networks in the last two to three decades have encouraged scientists to develop data analysis methods tailored especially for network data. This has given rise to *network science*, which combines graph theory, statistics, statistical physics, differential geometry and various other disciplines in order to study and analyze real-world networks — for instance social, informational, biological, or transportation networks, to name a few — and phenomena occurring on them [211, 201, 35, 158, 48, 123, 136, 154, 67, 79, 191, 13].

This thesis focuses on developing methods for network data analysis, often from a traditional data analysis perspective. The methods considered in the thesis relate to a few key topics of network analysis. One of them, that has been studied well in traditional data analysis in vector spaces, is the quantification of *relatedness* of two nodes of a network. This problem is approached through definition of *distance measures*, which can be used, for instance, for finding highly interlinked groups of nodes or for quantifying the overall quality of a network. The definition and study of distances on networks is a non-trivial and currently very active area of research.

Another network analysis topic that occurs in the thesis often, is the quantification of the *centrality*, in some sense, of a node in a network. This is one of the most studied network analysis problems that, likewise, does not entail a unique simple solution. A third topic that appears in the thesis is the modeling of mobility and spreading processes on networks, which are vital, for instance, for designing intelligent infrastructures, or for understanding how information spreads over social networks.

All in all, the thesis deals with the following topics:

- distances on networks that take into account global connectivity
- community detection on networks and clustering its nodes
- semisupervised classification of network nodes
- ranking nodes according to their centrality, importance or functionality
- quantifying the overall functionality of a network
- estimation of parameters for modeling movement on networks

Mostly these topics will be considered in the context of the *randomized shortest paths* (RSP) framework [215, 182, 87, 115, 117]. Many standard measures in network analysis are based on considering either shortest paths between the nodes of a network, or, alternatively, random walks on the network. The RSP framework considers these two traditional paradigms as extremes and defines an intermediate solution by considering probability distributions over paths of the network that focus on paths “close" to the shortest ones, but that also spread over longer paths.

This thesis studies the theory of RSPs developed earlier, in addition to developing and investigating new measures derived from it. The RSP framework is based on considering a probability distribution on the set of paths between two nodes of a network. From this distribution, several interesting quantities with intuitive interpretations can be computed in algebraically convenient ways. The thesis also considers the development of efficient algorithms for computing the measures on networks, the validation of the algorithms and often comparison with other methods. One highlight of these developments is the discovery of a new distance measure, called the *free energy distance*, which has several nice theoretical properties, in addition to which it has been shown to function well also experimentally.

The thesis considers the developed methods in a broad selection of application areas and with various types of network data sets. The distance measures developed in the thesis are tested in machine learning tasks, more precisely in clustering and semi-supervised classification of nodes in networks constructed

from text corpora, as well as in networks extracted from the World Wide Web. Network centralities are evaluated with experiments using street network data and a network extracted from the Wikipedia hyperlink network. A large part of the thesis also focuses on studying the utility of the RSP framework for ecology, based on the suitability of the assumptions of the framework for modeling animal movement. For this purpose, new measures are developed based on the RSP framework for quantifying the functionality of the nodes of a network as well as the complete network, in terms of its overall connectivity.

1.2 List of contributions

During my doctoral studies I have authored and co-authored several scientific articles. The list of articles where I have made a substantial contribution (either as the first author, or as a co-author), and the specific contributions in each article, are as follows:

(1) **I. Kivimäki et al. "A graph-based approach to skill extraction from text". In: *Graph-Based Methods for Natural Language Processing* (2013), pp. 79–87.**

This paper presents a web application for text analysis focusing on the task of *skill extraction*, meaning associating professional skills to arbitrary text documents. Such a method could potentially be used for profiling job applicants, or for targeting questions to experts on online question forums. The application is based on a method employing an associative network of concepts extracted from the online encyclopedia, Wikipedia and a list of skills retrieved from the LinkedIn website which hosts a professional social networking service. The paper demonstrates how the web application works and also evaluates its functionality experimentally. This work was done within a *Région Wallonne project* and is a collaboration of several authors. My responsibility was mostly in writing and compiling the paper and developing and implementing a spreading activation algorithm on the Wikipedia network as well as the evaluation experiments for the paper. However, this work is omitted from the thesis, as the topic of the article is largely unrelated to the topics dealt within the thesis.

(2) **I. Kivimäki, M. Shimbo, and M. Saerens. "Developments in the theory of randomized shortest paths with a comparison of graph node distances". In: *Physica A: Statistical Mechanics and its Applications* 393 (2014), pp. 600–616.**

This article deals with the development of two distance measures based on the RSP framework. The first measure is the RSP dissimilarity, which has been presented earlier in the literature. In this paper a novel batch algorithm is derived for computing the RSP dissimilarities between all pairs of nodes at once in closed form.

The paper also presents and studies the *free energy distance*, which was introduced also, in parallel, in article number (4) in this list, as the *potential distance*. In addition, a comparison of the RSP dissimilarity and the free energy distance with other similar graph node distance measures is performed, showing that the measures usually provide satisfying results in clustering and semisupervised classification tasks. The free energy distance has nice features from a theoretical point of view, in addition to which it was also shown to perform well in semisupervised classification in the parallel article number (4), and in clustering in [192] and [193].

I was responsible for the contents of the article otherwise, but the theory behind the work was developed in collaboration with Prof. Saerens.

(3) **I. Kivimäki et al. "Two betweenness centrality measures based on Randomized Shortest Paths". In: *Scientific Reports* 6.19668 (2016).**

This article defines two graph node betweenness centrality measures based on the RSP framework that can be considered as parametrized generalizations of certain traditional network centrality measures. The methods and their functionality are then evaluated with some example networks. Also in this work, part of the theory behind the methods was developed jointly with Prof. Saerens. In addition, some of the implementations of the methods were provided by Dr. Bertrand Lebichot, and some ideas of the experiments came from Prof. Jari Saramäki. Otherwise I was mostly responsible for the contents of the paper.

(4) **K. Françoisse et al. "A bag-of-paths framework for network data analysis". In: *Neural Networks* 90 (2017), pp. 90–111.**

This article presents the *bag-of-paths* (BoP) framework, which is a reinterpretation of the RSP framework, where a probability distribution is defined over the set of paths between all node-pairs on a graph, instead of only a fixed pair of nodes, as in the RSP framework. From this framework, two graph node distance measures are defined, namely the *potential* and the *surprisal* distances. The definition of the potential distance actually coincides with the definition of the free energy distance, presented from a different approach in [115] (see point number (2) in this list). The functionality of the potential (i.e. the free energy) and surprisal distances are evaluated in a semisupervised classification experiment, where both

distances provide good results, when compared to other state-of-the-art methods.

My contribution in this paper was in the development and derivation of the distance measures with Prof. Saerens. I also contributed a section to the paper about a new modularity measure and a community detection method based on it. This section was in the end withdrawn from the paper to make the paper more concise. Although the section was finally left unpublished elsewhere, it is presented in the thesis.

1.2.1 Other contributions

Aside from the above list of works, I have also contributed to other articles during the making of this thesis. In these articles my contribution has mostly been in brainstorming as well as reviewing and reading the manuscripts during the writing process and giving remarks and suggestions. The list of these co-authored works is as follows:

(5) B. Lebichot et al. "Semisupervised Classification Through the Bag-of-Paths Group Betweenness". In: *Neural Networks and Learning Systems, IEEE Transactions on* 25.6 (2014), pp. 1173–1186

(6) R. Devooght et al. "Random walks based modularity: application to semi-supervised learning". In: *Proceedings of the 23rd international conference on World wide web*. 2014, pp. 213–224

(7) M. Panzacchi et al. "Predicting the continuum between corridors and barriers to animal movements using Step Selection Functions and Randomized Shortest Paths". In: *Journal of Animal Ecology* 85.1 (2016), pp. 32–42

(8) G. Guex, I. Kivimäki, and M. Saerens. "Randomized Optimal Transport on a Graph: Framework and New Distance Measures". In: *Network Science* (2018). Accepted for publication

(9) B. Lebichot et al. "A Constrained Randomized Shortest-Paths Framework for Optimal Exploration". In: *arXiv preprint arXiv:1807.04551* (2018) (manuscript submitted for publication)

1.3 Structure of the thesis

The thesis is structured in the following way. Chapter 2 presents the definitions, terminology and notation, as well as the preliminary theoretical concepts and results and literature review for the rest of the thesis.

Chapters 3-8 are each based on a separate piece of work, and they have been sorted according to the relatedness of their topics in order to form a narrative as continuous as possible throughout the thesis. Some of the material from the articles have been moved to Chapter 2 in order to avoid repetition of definitions and results and to facilitate reading. Below is a description of each chapter based on the list of articles presented above in Section 1.2, and the numbering therein.

- **Chapter 3** is based on Article (2).
- **Chapter 4** was originally a section in Article (4). It presents a community detection method based on a reinterpretation of the modularity criterion in terms of the bag-of-paths (BoP) framework. As explained in Section 1.2, the section was in the end withdrawn from the article. The reason for this was that the community detection topic was somewhat separate from the main topic of the article, in addition to which the article was already too long. Although the material was left unpublished elsewhere, the theory was complete, and the results of the community detection experiment with the BoP modularity were promising, which is why the section is included as a chapter in this thesis.
- **Chapter 5** is based on Article (3).
- **Chapter 6** is based on an unfinished manuscript that presents a survey of the free energy measure in terms of the RSP framework for network analysis. The chapter contains a collection of results that have not been published before, as well as a list of interpretations of free energy and the probability distribution over paths on a graph, in order to convince a broad audience of network scientists to consider the method for different application purposes.
- **Chapter 7** is also based on an unfinished manuscript that considers the problem of estimating the parameters of the RSP model when modeling data consisting of trajectories on networks. The chapter develops methods for computing the maximum likelihood estimate (MLE) of the inverse temperature parameter β, which controls the balance between optimality and randomness of the trajectories, in scenarios where the trajectories are recorded either completely or only partly. The methods are validated with artificial examples with success. In addition, the method

for computing the MLE for incomplete trajectories is demonstrated with real data consisting of trajectories of wild reindeer on a geographic landscape, modeled as a network. The work was initially motivated by the research conducted within Article (7).

- **Chapter 8** is based on yet another unfinished manuscript, which is a larger collaborative effort for developing a method for quantifying the *functionality* of a habitat for an animal species in a geographic landscape. This project is partly a sequel to the study conducted in Chapter 7, as well as to the work in Article (7). The problem is tackled, again, with graph-theoretic formalism by modeling the landscape as a network and then quantifying the functionality of the nodes of the network based on the features of the node, as well as its connectivity with other nodes, and finally quantifying the overall functionality of the whole network. The chapter discusses the properties of the proposed Habitat Functionality metric from a network analysis viewpoint, as well as demonstrates its behavior in some artificial and real-life landscapes.

Finally, Chapter 9 sums up the thesis and its contributions, and discusses issues related to the presented methodologies. In addition, possible solutions to these issues along with other possible future developments are discussed.

Chapter 2

Theoretical foundations and related work

This thesis lies in the middle ground between network science, machine learning and data mining. This chapter first introduces the main topics of the thesis and their interplay, providing background motivation for the work. After that, the general setting of the thesis, from definitions and terminology to the broader concepts and methodology are introduced. We especially introduce the *randomized shortest paths (RSP) framework*, and present some relevant measures derived from it after first discussing background related to distance and centrality measures on graphs. The chapter ends with a literature review related to the main themes of the thesis. Some parts of this chapter were extracted from the articles listed in Introduction, especially parts that concern topics that recur in several chapters in the thesis.

2.1 General background and setting

Network science has its roots in many different scientific disciplines, two of which deserve specific merit. The first one is graph theory, from Euler's study of the bridges of Königsberg [71, 153] to the papers of Erdős and Rényi [63] on random graphs and further. The second important field has been the study of social networks, which has given rise to many of the standard measures used for studying networks in general [211]. One of the most well-known social network studies is the article on the *small-world phenomenon* by Travers and Milgram [206] which gave rise to the famous idea of *six degrees of separation*, i.e. that people (at least in the USA) are acquainted with each other through five other people.

A great boost for the study of networks towards being a science of its own was given by three paramount articles published at the end of the second millennium. The first one, by Watts and Strogatz [212], in 1998, showed how a simple and small shift from a regular graph towards a more random one can bring out interesting natural features in a network, including the small-world phenomenon, as well as clustering of the graph. The second noteworthy article, by Page et al. [164], published in 1999, introduced the Pagerank algorithm, which has given a strikingly concrete example of commercial potential of network applications, as it provided the technological backbone for the company founded by the authors — named Google. The third paper, by Barabási and Albert [14], also published 1999, coined the term *scale-free networks*, by demonstrating that the degree distributions of many real-world networks are, quite surprisingly, power-law distributions. Such networks that exhibit interesting and unintuitive statistical features are called *complex networks*. These models can be seen as initiators for a burst of studies in network-related research dealing with topics such as the evolution of networks [58], network robustness [5], epidemic spreading [169] as well as other stochastic processes occurring on networks [15], to name a few.

In addition to being used for studying the structure and foundations of networks, network data has also been analyzed for reasons that are often the basis of machine learning and data mining applications. As with any data, visualization is a relevant and important task also when dealing with network data [197]. One approach for visualization is through defining distances between graph nodes and then applying a distance-based visualization method. This is done in Section 3.5.1 of the thesis, where network visualization is performed based on different graph node distance measures.

Visualization is a subproblem of the more general problem of *network embedding*, meaning the problem of representing a network in another topological space, such as a Euclidean, or a hyperbolic space [24]. Embedding in a Euclidean space can be achieved, for instance, by defining distances between nodes according to a Euclidean distance measure and finding a set of points in a Euclidean space whose distances equal the graph distances. This property holds for random walk distances, as discussed in Section 2.4.2, and, in some cases, the RSP dissimilarity and free energy distance, studied in this thesis in general. This method was, for instance, used in the semisupervised classification experiments in [82] (Article (4) in the list of Chapter 1), where nodes were classified by linear support vector machines in a Euclidean space.

Another example of a network application that often draws inspiration from vector space based methods is community detection [74], which corresponds, broadly, to clustering in traditional data analysis. In the former, the task is to find groups of nodes in a network that are well-connected with each other,

but less connected with nodes in other groups. In the latter, the task is to find groups of points that are close in space to each other but distant from points in other groups. An example of community detection is presented in Chapter 4, where a new community detection method is applied on similarity graphs constructed from a text corpus. Furthermore, the kernel k-means method [216, 217], applied for clustering in Chapter 3, as well as the spectral clustering method [209], are examples of methods that operate in a vector space but are based on a graph representation of data. As another example, many non-linear dimensionality reduction techniques are based on constructing a similarity graph from data in a Euclidean space, and then recomputing a representation of the data based on distances on the graph instead of the Euclidean distances in the data space. One of the first proposals for such a strategy was the Isomap algorithm for dimensionality reduction [200, 131].

In other words, there exists a lot of interplay between network science and machine learning. However, combining traditional data analysis methods with the constantly growing understanding of real life networks still involves many open problems. The contributions presented in this thesis were always developed keeping in mind the above ideas.

2.2 Terminology and notation

Here, the terminology, notation and conventions that are used throughout this thesis are defined. The definitions are divided into paragraphs, with a separate discussion, in its own subsection, on the consideration of two types of weights on graphs.

Graphs. In general, $G = (V, E)$ denotes a finite, strongly connected directed weighted graph consisting of node set $V = \{1, 2, \ldots, n\}$ and edge set $E = \{(i, j)\}$. Nodes i and j such that $(i, j) \in E$ are called *adjacent* or *connected*. The set of *successor nodes* of a node i is defined as $Succ(i) = \{j \in V \mid (i, j) \in E\}$. The graph G is *undirected* if the edge set is symmetric, meaning that $(i, j) \in E$ if and only if $(j, i) \in E$. Note that this differs from the (perhaps more standard) convention where the edges in an undirected graph are considered as unordered pairs. As a result, in the above formulation the number of edges in an undirected graph is twice the number of edges in the more standard formulation.

Although a *network* is usually considered as a model of a real world object consisting of elements and their relations, and graphs are in principle used

as *abstractions* of networks, nevertheless, in this work the terms graph and network are mostly used interchangeably.

Weights. Through most of the thesis we consider graphs that are associated with two kinds of weights — affinities, a_{ij}, and costs, c_{ij}. The motivation for such setting is dedicated its own discussion below, in Section 2.2.1. In short, the affinities are used for defining a *random walk* on G according to transition probabilities on edges $(i, j) \in E$ as

$$p_{ij}^{\mathrm{rw}} = \frac{a_{ij}}{\sum\limits_{j \in Succ(i)} a_{ij}}. \tag{2.1}$$

In other words, a random walker will more likely follow an edge with high affinity, when moving between nodes. The random walk with the above transition probabilities corresponds to a first-order Markov chain with state set V. We sometimes refer to the random walk generated by the transition probabilities p_{ij}^{rw} as the *natural*, *unbiased*, or *reference* random walk.

The edge costs, c_{ij}, on the other hand define what is considered as optimal movement, as costs are something that should be avoided. They can be considered, for instance, in terms of the energy consumption, geographical distance, time duration, monetary value, or other form of expense related to the step over an edge. The edge affinities and costs can be defined based on each other, but are considered independent, for generality.

Paths. A *path* (or *walk*, interchangeably) on G, of length L, is a sequence of nodes $\wp = (i_0, \ldots, i_L)$, where $1 \leq L < \infty$ and $(i_{l-1}, i_l) \in E$ for all $l = 1, \ldots, L$. The lth node of path $\wp$ is denoted by $\wp(l)$ and the lth edge by $\wp(l-1, l)$. The *subpath* or *subsequence* of path $\wp$, from the l_1-th the l_2-th node will be denoted by $\wp(l_1 : l_2)$.

The set of all paths on G is denoted by $\widetilde{\mathcal{P}}$, whereas the set of all paths from a fixed source, or starting node s to a fixed target, or destination node t is denoted by $\widetilde{\mathcal{P}}_{st}$. We speak of a pair of starting node s and target node t concisely as an *s-t-pair* and a path going from s to t an *s-t-path*.

A *hitting* path (or *absorbing* path) is a path which contains the last node only once. Formally, a path $\wp = (i_0, \ldots i_L)$ is a hitting path if $i_l \neq i_L$ for all $l = 0, \ldots, L-1$. The set of all hitting paths on G and the set of hitting paths of an s-t-pair are denoted, respectively, by $\mathcal{P}$ and $\mathcal{P}_{st}$. The thesis often deals specifically with hitting paths, which explains the use of the form with the tilde for the set of all paths and the form without the tilde for hitting paths.

We sometimes use the term *regular paths* to refer to all paths (hitting and non-hitting) to make the distinction between the sets $\widetilde{\mathcal{P}}$ and $\mathcal{P}$ and their elements. For any of the four path set notations above, we denote the corresponding subset of paths of a fixed length $k \geq 1$ with the superscript (k); e.g., the set of hitting paths from s to t of a fixed length k is denoted by $\mathcal{P}_{st}^{(k)}$. Moreover, a *self-avoiding path* is a path that does not visit any node more than once, i.e. a path $\wp$ for which $\wp(i) \neq \wp(j)$ for all $0 \leq i < j \leq L(\wp)$.

The cost of a path $\wp \in \widetilde{\mathcal{P}}$ is defined as the sum of edge costs along the path[1]

$$\tilde{c}(\wp) = \sum_{i=1}^{L(\wp)} c_{\wp(i-1),\wp(i)}. \tag{2.2}$$

A *least cost path*, or *shortest path* between two nodes, s, $t \in V$, $s \neq t$, is a path $\wp^*$ such that $\tilde{c}(\wp^*) = \min\{\tilde{c}(\wp) \mid \wp \in \widetilde{\mathcal{P}}_{st}\}$. First, note that shortest paths in this thesis always refer to the path costs $\tilde{c}(\wp)$, and not the length, $L(\wp)$, i.e. the number of edges along a path. Also, we use the terms "least cost" and "shortest" for paths interchangeably throughout the thesis. The term shortest path is more standard in graph theory and network science, but in some contexts, especially with spatial networks such as the geographic networks considered in Chapters 7 and 8, the term least cost is more suitable to emphasize the consideration of edge costs and not only the spatial length of paths.

Many of the methods and results developed in this thesis are based on hitting paths. One reason for convenience about hitting paths is that for a fixed s-t-pair on G, with $s \neq t$, the *probability distribution over hitting paths* $\wp \in \mathcal{P}_{st}$ is given naturally by the product of transition probabilities along the path

$$\mathrm{P}_{st}^{\mathrm{rw}}(\wp) = \prod_{i=1}^{L(\wp)} p_{\wp(i-1),\wp(i)}^{\mathrm{rw}}. \tag{2.3}$$

Especially, with this definition, $\sum_{\wp \in \mathcal{P}_{st}} \mathrm{P}_{st}^{\mathrm{rw}}(\wp) = 1$, which is explicitly proved in [82], Appendix A. Note that the results derived in this work for hitting paths can be easily generalized to regular paths as well. We focus only on hitting paths for two reasons — simply for the sake of brevity, and also because the network measures derived from the hitting paths RSP framework are more directly related to many traditional network measures.

[1] The tilde ($\sim$) is used in the thesis sometimes to emphasize the difference between quantities related to paths (like path costs, $\tilde{c}(\wp)$ from quantities related to edges (like edge costs, c_{ij}).

Matrix notation. Vectors are generally denoted by small boldface characters and matrices by capital boldface characters, $\mathbf{e}_i$ denoting the i-th basis vector, i.e. 1 at element i and 0 elsewhere, and $\mathbf{e}$ a column vector (of length determined by context) whose each element is 1. The length of vectors and size of matrices is determined depending on the context, if not stated explicitly. For a square matrix $\mathbf{A}$, let $\mathbf{Diag}(\mathbf{A})$ denote the diagonal matrix, of same size as $\mathbf{A}$, whose diagonal elements are the diagonal elements of $\mathbf{A}$ and by $\mathbf{diag}(\mathbf{A})$ the column vector of the diagonal elements of $\mathbf{A}$. Likewise, for vector $\mathbf{v}$, $\mathbf{Diag}(\mathbf{v})$ denotes the diagonal square matrix containing the elements of vector $\mathbf{v}$ on its diagonal. We use $\exp(\mathbf{A})$ and $\log(\mathbf{A})$ to denote the elementwise exponential and logarithm, respectively; and $\mathbf{log}(\mathbf{A})$ to denote the matrix logarithm of $\mathbf{A}$. Furthermore, we use $\mathbf{A} \circ \mathbf{B}$ and $\mathbf{A} \div \mathbf{B}$ for elementwise product and division, respectively, of matrices $\mathbf{A}$ and $\mathbf{B}$. In addition, $\mathbf{I}_{ij}$ denotes the matrix whose element (i, j) is 1 and other elements are 0.

2.2.1 Affinities and costs

As mentioned already, the thesis deals mostly with weighted graphs where the edges of the graph are assigned two kinds of weights, affinities and costs. The reason for considering two types of weights that the thesis often considers a balance between two types of fundamental concepts on graphs, namely *random walks* and *shortest paths*. These concepts are very different in nature and deserve definitions based on independent elements.

Conventionally in graph theory or its applications, when speaking of weighted graphs, the weights reflect the strength of connection between two nodes in some sense. This corresponds to the concept of affinities, a_{ij}, in the thesis. Such weights can also be considered as (or defined according to) *similarities*, or *conductances* related to the edge, or the nodes connected by the edge. When considering random walks on a weighted graph, it makes sense to define the transition probabilities of the walk according to the weights such that a random walker, when moving away from a node, is more likely to follow an edge with a large weight instead of an edge with a small weight. In other words, the walker is allowed to utilize the local information contained in the edges leaving from the node that it is residing in.

However, considering shortest paths between nodes on weighted graphs may cause some conceptual confusion. Sometimes the weights are not utilized but shortest paths are simply defined in terms of the (unweighted) path length, i.e. the number of edges over a path. However, it often makes sense to take advantage of the edge weights for redefining a "weighted length" of paths. Such quantity should intuitively be additive over the edges of the path and

should be minimized for optimality. But in such a case the weights should be transformed into a counterpart quantity so that a short-length path would correspond to a path following edges with large weights.

These counterpart quantities correspond to the edge costs, c_{ij} in this work. This means that the edge costs define the interpretation of shortest paths, i.e. the low temperature behavior of the system, whereas the edge weights determine the interpretation of a random walk, i.e. the high temperature behavior. The interplay between weights and costs is thus similar to a tradeoff between *exploration*, based on local possible movements, and *exploitation*, based on long-term preferred movements. The question of how to then define the edge costs from the edge affinities is not at all a trivial one, but should be always considered with each application, by keeping in mind the correspondence between the affinities and the random walk, and between the costs and optimality.

Partly because of the above, the general assumption in the thesis about affinities and costs is that they are defined independent of each other. This assumption is used, for instance, when deriving quantities within the RSP framework. One example of how they can be considered independent is the case where the affinities are set to unity even though the costs have arbitrary values. In this setting the random walk chooses transitions uniformly from the set of outgoing edges but is blind to edge costs.

The assumption of independence between affinities and costs does not mean that they cannot be defined based on each other — just that they do not need to be. In some applications it can indeed be practical to define affinities and costs based on each other. One common convention is to define the costs as reciprocals of the affinities $c_{ij} = 1/a_{ij}$. Especially, when dealing with an undirected graph, this choice is analogous to an *electric network*, where the affinities and costs correspond to conductances and resistances, respectively. The connection with electric networks is faced repeatedly in the thesis, as the electric analogue has been used often for defining measures in network science.

Other possible transformations between affinities and costs include the exponential transformation $c_{ij} = \exp(-a_{ij})$ or the logarithmic transformation $c_{ij} = -\log(a_{ij})$. The latter choice is especially convenient, when the affinities correspond to probabilities, i.e. $0 < a_{ij} \leq 1$ for all i, j. With this choice, the path costs (the sums of costs) reflect path probabilities or path likelihoods (the products of the affinities). In particular, in this setting, least cost paths correspond to *most probable paths*.

2.3 Randomized shortest paths framework

This section contains the definition of the *randomized shortest paths (RSP) framework* [215, 182, 87, 115, 117] which is the main object of study in most of this thesis. As discussed earlier, the two most standard paradigms for studying movement on networks consider it as occurring over shortest paths or over random walks. These paradigms are, however, often too simplified and inadequate for building realistic models. In real life, movement or flow rarely happens only along shortest paths, nor is it completely random either. In addition, the standard network metrics derived from these paradigms often have limitations and may in some cases fail in their objectives. A discussion of such issues with distance measures derived from these paradigms is presented in Section 2.4.5.

The above reasons have motivated the development of alternatives for the traditional paradigms. One of them is given by the *randomized shortest paths (RSP) framework* [215, 182, 115, 117], which defines a Gibbs-Boltzmann distribution over paths from a *source node* s to a *target node* t involving an *inverse temperature parameter*, $\beta = 1/T > 0$, which controls the degree of randomization from optimal, least cost paths. At the limits of the parameter range, the distribution focuses either on solely the least cost paths ($\beta \longrightarrow \infty$) or spreads over all hitting paths from s to t according to the natural random walk probabilities ($\beta \longrightarrow 0^+$).

In addition to providing an intermediate between optimal paths and random walks, the RSP distribution over paths has several other nice properties. Most importanly, its algebraic convenience enables the computation of many interpretable metrics on a network, such as distance measures [215, 115], presented in Section 2.4.6 as well as centrality measures [117], discussed in Section 2.5. In addition to defining network metrics, the RSP framework can be used for modeling or predicting path selection on networks. It was recently used for modeling movement and locating movement corridors of wild reindeer in a geographic network [168], and has been familiarized with the ecological community thanks to the implementation in the `gdistance` R package [70]. This line of application of RSPs is studied in Chapters 7 and 8.

2.3.1 Gibbs-Boltzmann distribution over paths

The choice of the Gibbs-Boltzmann distribution for considering path probabilities can be justified in many ways, some of which are discussed later in this thesis, in Section 6.3. Here we recall the most common, and perhaps the most intuitive definition. Namely, we seek for a probability distribution, P_{st}, that

minimizes the expected cost of hitting paths from s to t subject to a relative entropy constraint:

$$\underset{\mathrm{P}_{st}}{\text{Minimize}} \sum_{\wp \in \mathcal{P}_{st}} \mathrm{P}_{st}(\wp)\tilde{c}(\wp) \quad \text{subject to} \quad \begin{cases} \mathbb{J}(\mathrm{P}_{st} \| \mathrm{P}_{st}^{\mathrm{rw}}) = J_0 \\ \sum\limits_{\wp \in \mathcal{P}_{st}} \mathrm{P}_{st}(\wp) = 1, \end{cases} \tag{2.4}$$

where $J_0 \geq 0$ and

$$\mathbb{J}(\mathrm{P}_{st} \| \mathrm{P}_{st}^{\mathrm{rw}}) = \sum_{\wp \in \mathcal{P}_{st}} \mathrm{P}_{st}(\wp) \log\left(\frac{\mathrm{P}_{st}(\wp)}{\mathrm{P}_{st}^{\mathrm{rw}}(\wp)}\right) \tag{2.5}$$

is the *Kullback-Leibler divergence*, or *relative entropy* of P_{st} w.r.t. the unbiased random walk path probability distribution $\mathrm{P}_{st}^{\mathrm{rw}}$, from Equation (2.3). In other words, the value J_0 fixes the level of randomness of the sought distribution so that with small values of J_0 the distribution is similar to the random walk distribution, $\mathrm{P}_{st}^{\mathrm{rw}}$, but with increasing values, it starts to deviate from P_{st} and focuses more on low-cost paths. Strictly speaking, the minimization would also require stating non-negativity constraints, but they are not needed here, as is usual in maximum entropy problems.

Although the solution of the minimization problem (2.4) is a standard result in statistical physics and maximum entropy modeling, and although it has been proven elsewhere (e.g. in [182]), we nevertheless include a proof here for completeness. The Lagrangian of the problem writes out, with Lagrangian parameters λ and μ for the constraints, as

$$\begin{aligned} \mathcal{L}(\mathrm{P}_{st}, \lambda, \mu) = & \sum_{\wp \in \mathcal{P}_{st}} \mathrm{P}_{st}(\wp)\tilde{c}(\wp) \\ & + \lambda \left(\sum_{\wp \in \mathcal{P}_{st}} \mathrm{P}_{st}(\wp) \log\left(\frac{\mathrm{P}_{st}(\wp)}{\mathrm{P}_{st}^{\mathrm{rw}}(\wp)}\right) - J_0 \right) \\ & + \mu \left(\sum_{\wp \in \mathcal{P}_{st}} \mathrm{P}_{st}(\wp) - 1 \right). \end{aligned} \tag{2.6}$$

Taking the partial derivative w.r.t. $\mathrm{P}_{st}(\wp)$, for an arbitrary path $\wp \in \mathcal{P}_{st}$ and setting it to zero gives

$$\mathrm{P}_{st}(\wp) = \mathrm{P}_{st}^{\mathrm{rw}}(\wp) \exp(-\beta\tilde{c}(\wp) - \beta\mu - 1), \tag{2.7}$$

where we have defined $\beta = 1/\lambda$. From this we already see that the non-negativity constraint is not required in the minimization, as the above expression is always non-negative. Enforcing the constraint on the probabilities to sum to one then gives

$$\exp(-\beta\mu - 1) = \frac{1}{\mathcal{Z}_{st}}, \tag{2.8}$$

where

$$\mathcal{Z}_{st} = \sum_{\wp \in \mathcal{P}_{st}} \mathrm{P}^{\mathrm{rw}}_{st}(\wp) \exp(-\beta \tilde{c}(\wp)) \tag{2.9}$$

is the *partition function* of hitting paths from s to t, which plays an important role in many definitions and derivations related to RSPs. Finally, the solution to the minimization is given after inserting (2.8) to (2.7) by the *Gibbs-Boltzmann distribution*

$$\mathrm{P}^{\mathrm{RSP}}_{st}(\wp) = \frac{1}{\mathcal{Z}_{st}} \mathrm{P}^{\mathrm{rw}}_{st}(\wp) \exp(-\beta \tilde{c}(\wp)), \tag{2.10}$$

which we also often call in this thesis the *RSP probability distribution*. The parameter β should be non-negative in order for the solution to provide a minimum expected cost. This is because with a negative value of β the probability of a path becomes inversely related to its cost. In other words, if $\beta < 0$, then out of two paths which have the same random walk probability $\mathrm{P}^{\mathrm{rw}}_{st}$, the one with a higher cost would have a higher probability according to (2.10), meaning that the distribution with $\beta < 0$, although it may exist, cannot correspond to a minimum expected cost.

The relative entropy, J_0, of the distribution is now determined by parameter β so that with $\beta = 0$, also $J_0 = 0$, in which case the distribution is equal to the random walk distribution, $\mathrm{P}^{\mathrm{rw}}_{st}$. When β increases from 0, then J_0 increases as well, and the distribution $\mathrm{P}^{\mathrm{RSP}}_{st}$ focuses more and more on paths from s to t with a low cost, but a high natural random walk probability. Ultimately, as $\beta \longrightarrow \infty$, the distribution converges to a distribution focused only on the least cost paths. If there are K shortest paths from s to t, say $\wp^*_1, \ldots, \wp^*_K$, then at the limit $\beta \longrightarrow \infty$ the probability of each of those paths is determined by the natural random walk probabilities as

$$\mathrm{P}^{\mathrm{RSP}}_{st}(\wp^*_j) \longrightarrow \frac{\mathrm{P}^{\mathrm{rw}}_{st}(\wp^*_j)}{\sum_{k=1}^{K} \mathrm{P}^{\mathrm{rw}}_{st}(\wp^*_k)} \triangleq \mathrm{P}^{*}_{st}(\wp^*_j). \tag{2.11}$$

This can be justified by noting that as $\beta \to \infty$, for all paths $\wp \in \mathcal{P}_{st}$, whose cost $\tilde{c}(\wp) > \tilde{c}^*_{st}$, $\exp(-\beta\tilde{c}(\wp)) \ll \exp(-\beta\tilde{c}^*_{st})$, because of which the partition function $\mathcal{Z}_{st}$ of Equation (2.9) becomes dominated by the terms determined by the shortest paths. We call distribution P^{*}_{st} the *shortest path likelihood distribution* from s to t.

Because of the above, and the interpretation of the Gibbs-Boltzmann distribution in thermodynamics, we may interpret parameter β as the *inverse temperature* $\beta = 1/T$. When applying the RSP framework, the user is assumed to give β as input, instead of J_0. The parameter β thus has a similar role as J_0 in the formulation of the minimization problem, i.e. that it controls the degree of randomness of the distribution $\mathrm{P}_{st}^{\mathrm{RSP}}$, although the relative entropy, J_0, is, in principle, a more appropriate quantifier of the degree of randomness. If indeed a particular level of relative entropy is desired, then the appropriate value of β can be determined by line search. However, the methods developed in this thesis from the RSP framework are often computed in batch mode, for the whole graph at once, meaning that a fixed value of β is used for all s-t-pairs, even though then the relative entropies $\mathbb{J}(\mathrm{P}_{st} \| \mathrm{P}_{st}^{\mathrm{rw}})$ are different for different s-t-pairs.

The question of how to choose the value of β when applying the RSP model is eventually an open one, and depends on the application. This topic is considered in Chapter 7, which develops maximum likelihood estimation methods for fitting the RSP distribution to a data set of trajectories observed on a network. The methods are tested with artificial examples, but also with real data consisting of trajectories of wild reindeer recorded from GPS devices. However, the decision on the value of β in other context can be based on very different grounds.

2.3.2 Computation of the partition function

As stated already, the partition function of hitting paths from s to t, $\mathcal{Z}_{st}$, can be used for computing different relevant quantities related to the RSP model. To show how the partition function $\mathcal{Z}_{st}$ can be computed, we consider the random walk transition probability matrix $\mathbf{P}^{\mathrm{rw}}$ and the cost matrix $\mathbf{C}$ consisting of the elements p_{ij}^{rw} and c_{ij}, respectively, at indices such that $(i, j) \in E$ and zeros elsewhere. Using these, we define the *likelihood matrix* $\mathbf{W}$ as

$$\mathbf{W} = \mathbf{P}^{\mathrm{rw}} \circ \exp(-\beta \mathbf{C}), \tag{2.12}$$

where $\circ$ is the element-wise, or Hadamard product and the exponential is taken element-wise as well. The likelihood matrix elements, $w_{ij} = p_{ij}^{\mathrm{rw}} \exp(-\beta c_{ij})$, are called the *edge likelihoods*. Furthermore, for a path $\wp$ the product of edge likelihoods along it define the *path likelihood* of $\wp$:

$$\widetilde{w}(\wp) = \prod_{i=1}^{L(\wp)} w_{\wp(i-1),\wp(i)} = \mathrm{P}_{st}^{\mathrm{rw}}(\wp) \exp(-\beta \tilde{c}(\wp)), \tag{2.13}$$

Note that for hitting paths this quantity corresponds to the numerator in the expression for the RSP distribution, $\mathrm{P}_{st}^{\mathrm{RSP}}$ in Equation (2.10), and that we can thus simplify the expressions for $\mathrm{P}_{st}^{\mathrm{RSP}}$ and the partition function, $\mathcal{Z}_{st}$, as

$$\mathrm{P}_{st}^{\mathrm{RSP}}(\wp) = \frac{\widetilde{w}(\wp)}{\mathcal{Z}_{st}} \quad \text{and} \quad \mathcal{Z}_{st} = \sum_{\wp \in \mathcal{P}_{st}} \widetilde{w}(\wp). \tag{2.14}$$

This matrix is substochastic, i.e. it is non-negative and has row sums less than 1. Thus it can be interpreted as a new transition matrix defining a *killed*, or *evaporating* random walk on G [78]. This means that at each step of the walk the random walker has a non-zero probability of stopping its walk, i.e. getting killed. Another way of interpreting this is by imagining an additional, absorbing "cemetery" node [17], where the walker can end up from each node i according to the residual probability $1 - \sum_j w_{ij}$ of the node.

Remember now that we only consider hitting paths, meaning that we want to make the target node t absorbing. For this, we define a new matrix by setting the row t of $\mathbf{W}$ to zero (similarly to the computation of absorbing Markov chains; see, e.g., [94])

$$\widehat{\mathbf{W}} = \mathbf{W} - \mathbf{I}_{tt}\mathbf{W} \tag{2.15}$$

which is equivalent to considering a deletion of all edges leaving node t. In other words, matrix $\widehat{\mathbf{W}}$ defines a killed and *absorbing* random walk to node t on G.

It is then not difficult to see that the overall likelihood of the set $\mathcal{P}_{st}^{(k)}$ of hitting s-t-paths of a given length $k \in \{1, 2, \ldots\}$ is given by elements (s, t) of powers of $\widehat{\mathbf{W}}$:

$$\mathbf{e}_s^{\mathsf{T}} \widehat{\mathbf{W}}^k \mathbf{e}_t = \sum_{\wp \in \mathcal{P}_{st}^{(k)}} \widetilde{w}(\wp). \tag{2.16}$$

As a result, the partition function, $\mathcal{Z}_{st}$, for any s-t-pair is given by element (s, t) of the *fundamental matrix of killed absorbing random walks to* t:

$$\widehat{\mathbf{Z}} = \mathbf{I} + \widehat{\mathbf{W}} + \widehat{\mathbf{W}}^2 + \widehat{\mathbf{W}}^3 + \cdots = \left(\mathbf{I} - \widehat{\mathbf{W}}\right)^{-1}. \tag{2.17}$$

Here we include in the series the identity matrix, i.e. the 0^{th} power, $\widehat{\mathbf{W}}^0 = \mathbf{I}$, although a more appropriate definition of the fundamental matrix would possibly discard that. However, this does not affect the derivations in the following, but instead makes the notation simpler. The same convention is adopted later as well.

Note that the matrices $\widehat{\mathbf{W}}$ and $\widehat{\mathbf{Z}}$ depend on the fixed target node t. Thus,

computing quantities with the above approach can be costly, when considering different target nodes, t, as the matrix inverse appearing in Equation (2.17) has to be computed separately for each t. The authors in [215] developed a way for computing the partition functions, $\mathcal{Z}_{st}$, for all s-t-pairs on a graph avoiding the computation of the above inverse (in Equation (2.17)) separately for each t. This method was then used for computing the RSP dissimilarity (presented in Section 2.4.6) between all node-pairs of a graph. However, the algorithm presented in [215] only computes the RSP dissimilarities to one target node at a time and performs a loop over all nodes as targets, which can still be quite slow.

One of the contributions of this thesis, which was presented both in [115] and in [82], and is proved in Section 3.2 is a result that simplifies and reduces the complexity of the computation of partition functions of all s-t-pairs even further from the method presented in [215]. The result states that, in fact, the partition function $\mathcal{Z}_{st}$, for any s-t-pair, such that $s \neq t$, can be simply expressed as

$$\mathcal{Z}_{st} = z_{st}/z_{tt}, \tag{2.18}$$

where z_{st} is element (s, t) of the *fundamental matrix of regular paths*:

$$\mathbf{Z} = \mathbf{I} + \mathbf{W} + \mathbf{W}^2 + \mathbf{W}^3, \cdots = (\mathbf{I} - \mathbf{W})^{-1}. \tag{2.19}$$

We leave the derivation of this result to Section 3.2. It is then used in Section 3.2 for deriving an algorithm for computing the all-pairs RSP dissimilarities more efficiently than the original algorithm in [215].

Note that the likelihood matrix $\mathbf{W}$ is non-negative, substochastic and irreducible (as G is assumed strongly connected), which implies that its spectral radius is less than one, $\rho(\mathbf{W}) < 1$, ensuring that the above matrix series converges and can be computed as the above inverse [148]. The same justification applies with matrix $\widehat{\mathbf{W}}$ in Equation (2.17) (as the spectral radius of a matrix cannot increase by setting some of its elements to zero).

In other words, the computation of the hitting path partition functions for all pairs requires only the computation of the matrix inverse in Equation (2.19), or, equivalently, solving the system of linear equations

$$(\mathbf{I} - \mathbf{W})\mathbf{Z} = \mathbf{I}, \tag{2.20}$$

The *fundamental matrix of hitting paths*, whose element (s, t) (for all $s \neq t$) is the hitting path partition function $\mathcal{Z}_{st}$, can be computed as

$$\mathbf{Z}_{\mathrm{h}} = \mathbf{Z}\mathbf{D}_{\mathbf{Z}}^{-1}, \tag{2.21}$$

where $\mathbf{D_Z} = \mathbf{Diag}(\mathbf{Z})$ is the matrix consisting of the diagonal of matrix $\mathbf{Z}$. Note that as we keep the identity matrix $\mathbf{I}$ in the series defining $\mathbf{Z}$ in Equation (2.19), as a result the diagonal elements of $\mathbf{Z}_{\mathrm{h}}$ are 1. However, according to our definitions, the hitting path partition functions from a node to itself, $\mathcal{Z}_{tt}$ should be zero. Nevertheless, the definition of $\mathbf{Z}_{\mathrm{h}}$ as above is convenient for computations later in the thesis.

2.4 Distances on graphs

One of the main themes of the thesis is the development and consideration of distance measures between nodes of a graph. Perhaps the most intuitive conception, for most people, of distance is the Euclidean distance between two points in a (normally 2 or 3-dimensional) Euclidean space, which can also be called the "straight-line" distance. But when measuring geographic distance, the Euclidean distance fails, as distance needs to be measured along the surface of the Earth. One may then consider the *great-circle* distance, which considers the Earth as a sphere and measures the distance between two points along the sphere. However, the Earth is not exactly a sphere, but rather an ellipsis. Furhermore, the surface of the Earth is not smooth, but the difference in elevation also affects the actual distance along the surface between two points.

On the other hand, sometimes the conception of distance might be more meaningful in terms of travel time. This can be relevant, for instance, when considering movement inside a city using the different possible modes of transportation. Such a distance can be then used for constructing visualizations of the city that reflect the level of connectivity of the city. However, when considering travel time as a distance, the distance from one place to another may not be the same as the distance in the other direction. Going uphill is often slower than going downhill.

This thought process should illustrate that the concept of a distance can be ambiguous. Moreover, the definition of a distance is not a trivial task, in any topological space. This is also true when the topological space considered is a graph.

The standard mathematical conception of a distance is the concept of *metric* in topology. For a set X, a function $\Delta : X \times X \to [0, \infty)$ is a *metric*, if it satisfies, for all $x, y, z \in X$ the following axioms (see, e.g. [55]):

- $\Delta(x, y) \geq 0$ (*non-negativity*)
- $\Delta(x, y) = 0$, iff $x = y$ (*identity of indiscernibles*)

- $\Delta(x, y) = \Delta(y, x)$ (*symmetry*)
- $\Delta(x, z) \leq \Delta(x, y) + \Delta(y, z)$ (*triangle inequality*)

Sometimes the term distance is defined as synonymous to a metric. Moreover, sometimes methods are specifically designed such that the metric axioms are assumed. For instance, the triangle inequality can be taken advantage of in order to make some distance-based algorithms more efficient.

However, sometimes some of the axioms of metric can be relaxed while maintaining the idea of distance. For instance, as is evident in the above example with travel times, the constraint of symmetry may not always be relevant. This situation occurs also often when defining distances on a graph, especially when considering directed graphs (which can be also used as a model of a city and travel times in it). Accordingly, the consideration of distances in this thesis covers both the consideration of symmetric, as well as asymmetric, or *directed* distances. In order to avoid confusion, the term distance refers normally only to a symmetric quantity, or else the term "directed distance" is used.

Next, we give definitions and list interesting properties of two types of traditional graph node distance measures. The first type is the *shortest path distance*, and its directed and weighted versions. The second type is what we call the *random walk distances*, which contains three different distance measures, namely the *resistance distance*, the *commute time distance* and the *commute cost distance*. These three can be considered equivalent, when dealing with a single fixed graph, as will be discussed later in the section. The main motivation for this presentation is the fact that two of the distance measures derived from the RSP framework that are studied most in this thesis, namely the *RSP dissimilarity* and the *free energy distance*, both interpolate, depending on a parameter, between the commute cost distance and the shortest path distance.

2.4.1 Shortest path distance

The most standard distance measure on a graph is the *shortest path distance*, sometimes called the *geodesic distance*, especially when estimating distances along a curved surface. Its definition on undirected unweighted graphs is simply the minimal number of edges along a path that connects two nodes. Generalizing the shortest path distance to weighted graphs was discussed already in Section 2.2.1. Accordingly, in the thesis, the "shortness", or optimality, of a path is defined based on the edge costs, c_{ij}, considered assigned to all edges $(i, j) \in E$. Accordingly, the term shortest path distance is sometimes replaced in the thesis with the term *least cost distance* to emphasize that the distance is defined based on the edge costs, and not only the number of edges

along paths.[2] In other words, the terms shortest path distance and least cost distance always refer to the same concept in the thesis.

Generalizing the shortest path distance to directed graphs requires making some situation-dependent decisions. Especially, one must decide whether to consider distance as a directed, asymmetric measure of separation, or as a somehow symmetrized version of the asymmetric measure.

The *least cost*, or the *directed shortest path distance*, from node s to node t on G, is defined as

$$\tilde{c}^*_{st} = \min\{\tilde{c}(\wp) \mid \wp \in \widetilde{\mathcal{P}}_{st}\}. \tag{2.22}$$

The *shortest path distance*, or *least cost distance* between s and t is then defined as the symmetrized least cost

$$\Delta^{\mathrm{SP}}_{st} = \Delta^{\mathrm{LC}}_{st} = \tfrac{1}{2}(\tilde{c}^*_{st} + \tilde{c}^*_{ts}). \tag{2.23}$$

Both notations, Δ^{SP} and Δ^{LC}, appear exchangeably in the thesis. It follows fairly trivially from the definition that $\Delta^{\mathrm{SP}}_{st}$ satisfies the triangle inequality and thus it is a metric. Note that on undirected graphs the least cost is symmetric, and thus $\Delta^{\mathrm{SP}}_{st} = \tilde{c}^*_{st} = \tilde{c}^*_{ts}$, which, however, does not necessarily hold on directed graphs. Obviously, when $c_{ij} = 1$ for all $(i, j) \in E$ (corresponding to an unweighted graph) and when the graph is undirected, the least cost and shortest path distance are synonymous. We additionally define, though they are used less often, the *directed shortest path length* as

$$L^*_{st} = \min\{L(\wp) \mid \wp \in \widetilde{\mathcal{P}}_{st}\} \tag{2.24}$$

and the *unweighted shortest path distance as*

$$\Delta^{\mathrm{uSP}}_{st} = \tfrac{1}{2}(L^*_{st} + L^*_{ts}). \tag{2.25}$$

2.4.2 Random walk distances

Commute distances. The distances based on random walks can be defined by considering hitting paths. Remember that a path $\wp$ on G is a hitting (or absorbing) path, $\wp \in \mathcal{P}_{st}$, if the last node of $\wp$ does not appear as an intermediate node in $\wp$. Remember also the definition of the transition probabilities, p^{rw}_{ij}, from Section 2.2, which define the unbiased random walk on the graph.

[2] Some authors, e.g. Chebotarev in [45], instead call the shortest path distance based on edge weights the *weighted* shortest path distance and use the term shortest path distance only for the distance based on the number of edges on paths.

Considering hitting paths, we can define the *expected random walk hitting time* (also sometimes called the expected/mean/average first passage time) and the *expected random walk hitting cost* from s to t as

$$\langle L \rangle_{st}^{\mathrm{rw}} = \sum_{\wp \in \mathcal{P}_{st}} \mathrm{P}_{st}^{\mathrm{rw}}(\wp) L(\wp) \quad \text{and} \quad \langle \tilde{c} \rangle_{st}^{\mathrm{rw}} = \sum_{\wp \in \mathcal{P}_{st}} \mathrm{P}_{st}^{\mathrm{rw}}(\wp) \tilde{c}(\wp), \tag{2.26}$$

respectively. In words, $\langle L \rangle_{st}$ and $\langle \tilde{c} \rangle_{st}$ are the expected length, i.e. number of steps, i.e. discrete time duration $L(\wp)$, and the expected cumulated cost $\tilde{c}(\wp)$, respectively, of a path taken by a random walker to reach t from s for the first time.

By symmetrization, we obtain the definition of the *commute time distance* and the *commute cost distance* between s and t, as

$$\Delta_{st}^{\mathrm{CT}} = \langle L \rangle_{st}^{\mathrm{rw}} + \langle L \rangle_{ts}^{\mathrm{rw}} \quad \text{and} \quad \Delta_{st}^{\mathrm{CC}} = \langle \tilde{c} \rangle_{st}^{\mathrm{rw}} + \langle \tilde{c} \rangle_{ts}^{\mathrm{rw}}, \tag{2.27}$$

respectively.

It is not difficult to show that the expected hitting time and cost can likewise be expressed, respectively, as,

$$\langle L \rangle_{st}^{\mathrm{rw}} = \sum_{(i,j) \in E} \bar{\eta}_{ij}^{\mathrm{rw}}(s,t) \quad \text{and} \quad \langle \tilde{c} \rangle_{st}^{\mathrm{rw}} = \sum_{(i,j) \in E} \bar{\eta}_{ij}^{\mathrm{rw}}(s,t) c_{ij}, \tag{2.28}$$

where $\bar{\eta}_{ij}^{\mathrm{rw}}(s,t)$ denotes the *expected number of passages over edge* (i,j) with respect to the unbiased probability distribution over hitting paths from s to t, i.e.

$$\bar{\eta}_{ij}^{\mathrm{rw}}(s,t) = \sum_{\wp \in \mathcal{P}_{st}} \mathrm{P}_{st}^{\mathrm{rw}}(\wp) \eta_{ij}(\wp), \tag{2.29}$$

where $\eta_{ij}(\wp)$ denotes the number of times that path $\wp$ traverses edge (i,j) (in that direction). Accordingly, the commute time and commute cost distances can be written as

$$\Delta_{st}^{\mathrm{CT}} = \sum_{(i,j) \in E} \left(\bar{\eta}_{ij}^{\mathrm{rw}}(s,t) + \bar{\eta}_{ij}^{\mathrm{rw}}(t,s) \right) \quad \text{and} \tag{2.30}$$

$$\Delta_{st}^{\mathrm{CC}} = \sum_{(i,j) \in E} \left(\bar{\eta}_{ij}^{\mathrm{rw}}(s,t) + \bar{\eta}_{ij}^{\mathrm{rw}}(t,s) \right) c_{ij}. \tag{2.31}$$

Resistance distance. Another related and traditional distance measure on graphs is the *resistance distance* [118], Δ^{Res}, which originally gained interest in

chemical graph theory, but has then been used widely in graph theory applications, including graph-based machine learning. Resistance distance can be defined in multiple ways. The standard definitions are based on considering the graph as an electric network and studying electric current flows on that network. Accordingly, resistance distance is normally only defined for undirected graphs, as electric networks are likewise undirected.

Let G thus be undirected, and let the edges (i, j) correspond to resistors in an electric circuit with conductances corresponding to the edge affinities a_{ij}, and, accordingly, resistances corresponding to the costs $r_{ij} = 1/a_{ij}$. Based on this setting, we first list four equivalent, all somewhat informal definitions of resistance distance, $\Delta_{st}^{\mathrm{Res}}$, between an s-t-pair on G. The definitions are all based on [59] and [27]. Namely, $\Delta_{st}^{\mathrm{Res}}$ is

- the *effective resistance* between s and t, when the terminals of a 1-volt battery are attached to s and t.
- the inverse of *effective conductance* between s and t, which, in turn, equals the amount of electric current running in the network, when a 1-volt battery is attached to s and t.
- the amount of voltage required from a battery attached to s and t for generating an electric current of 1 ampere on the network.
- the overall resistance of the network, when reducing the whole network into a network with only the terminal nodes and one single resistor between them (if such reduction is possible).

To understand better the connection between random walks and the resistance distance, we present a formal definition of $\Delta_{st}^{\mathrm{Res}}$ using the approach and formalism for hitting paths presented above, augmented with the concepts concerning flows. A function $I : E \to \mathbb{R}^m$ on the set of edges E is a *unit flow* from s to t, if

$$\sum_{j \in Succ(i)} I_{ij} - \sum_{j \in Succ(i)} I_{ji} = \delta_{is} - \delta_{it}, \quad i \in V. \tag{2.32}$$

In other words, the sum of currents flowing into a node and out of it cancel out to zero, except for s, where a unit of current departs, and t, where the flow stops (or exits the network). Furthermore, if a unit flow I satisfies *Ohm's law*, i.e. that there exist *voltages*, U_i, on the nodes $i \in V$ such that for every edge (i, j), with $j \neq t$,

$$I_{ij} = \frac{U_i - U_j}{r_{ij}}, \tag{2.33}$$

then the flow defines an electric current flow of 1 ampere between s and t.

Let I then be a unit electric current flow. Equivalently to the descriptive definitions presented earlier, resistance distance can now be defined as the *power* consumed by the network due to I:

$$\Delta_{st}^{\mathrm{Res}} = \frac{1}{2} \sum_{(i,j)\in E} I_{ij}^2 r_{ij}. \tag{2.34}$$

It is shown in [59] that the current I_{ij} is equal to the *net expected number of traversals over* (i, j) over hitting paths from s to t with respect to the unbiased random walk probabilities, i.e. $I_{ij} = |\bar{\eta}_{ij}(s,t) - \bar{\eta}_{ji}(s,t)|$. Based on this, the resistance distance can be rewritten as

$$\Delta_{st}^{\mathrm{Res}} = \frac{1}{2} \sum_{(i,j)\in E} (\bar{\eta}_{ij}(s,t) - \bar{\eta}_{ji}(s,t))^2 r_{ij}. \tag{2.35}$$

The sum of all resistance distances on an undirected graph, G, is called the *Kirchhoff index*

$$\mathcal{K}(G) = \sum_{s,t=1}^{n} \Delta_{st}^{\mathrm{Res}}. \tag{2.36}$$

This quantity has been used to quantify the overall connectedness, or compactness of a graph, so that when comparing graphs of same size (in terms of number of nodes), the one with a smaller Kirchhoff index can be considered better connected. Also, $\mathcal{K}(G)$ has been shown to measure the robustness of a graph against removal or rewiring of edges. This was demonstrated, for example, in [175] by considering the change in average connectivity of nodes after a split of the graph into components caused by edge removals.

2.4.3 Relation between commute and resistance distances

As is evident from comparing the form in Equation (2.35) for Δ^{Res} with the similar forms for Δ^{CT} and Δ^{CC} in Equation (2.28), the three distances are very closely related. In fact, it is well known (although it does not follow directly from the above comparison) that on undirected graphs resistance distances are proportional to commute time distances, as well as the commute cost distances [42, 115, 92].

A proof for this result concerning commute cost distance was proven in slightly different form in [115] by referring to results presented earlier in [77]. The proof is extracted from [115] and presented here. The result was proven in a more elementary method already in [42]. Later, the result has been presented

in [92], which also shows that the result holds between the commute time and commute cost distances also on directed graphs (remember that the resistance distance is not usually defined on directed graphs).

Theorem 2.4.1. On an undirected graph, the commute cost distance is proportional to the commute time and resistance distance

Proof. For deriving this result, we present the well-known method for computing the resistance distance from the *Laplacian matrix* of graph G, $\mathbf{L} = \mathbf{D} - \mathbf{A}$, where $\mathbf{D} = \mathbf{diag}(\mathbf{Ae})$ is the diagonal matrix of row sums of $\mathbf{A}$ (see, e.g. [118]). Namely, the $\Delta_{st}^{\mathrm{Res}}$ can be computed using the *Moore-Penrose pseudoinverse*, $\mathbf{L}^+$ of the Laplacian, as

$$\Delta_{st}^{\mathrm{Res}} = l_{ss}^+ + l_{tt}^+ - l_{st}^+ - l_{ts}^+. \tag{2.37}$$

In fact, this result is often stated as the definition of resistance distance (as it does not require explanations about electric network theory) [121, 12]. It was shown in [77], first, that the commute-time distance can be computed, based on the above form, as

$$\Delta_{st}^{\mathrm{CT}} = \Delta_{st}^{\mathrm{Res}} \sum_{(i,j)\in E} a_{ij}, \tag{2.38}$$

in addition to which the authors in [77] derive a formula for computing the expected hitting cost, $\langle \tilde{c} \rangle_{st}^{\mathrm{rw}}$, (see [77], Appendix B, Equation (18)), as

$$\langle \tilde{c} \rangle_{st}^{\mathrm{rw}} = \sum_{i=1}^{n} (l_{si}^+ - l_{st}^+ - l_{ti}^+ + l_{tt}^+) \sum_{j\in Succ(i)} a_{ij} c_{ij}. \tag{2.39}$$

From this we can obtain the commute cost distance $\Delta_{st}^{\mathrm{CC}}$ by symmetrization:

$$\begin{aligned} \Delta_{st}^{\mathrm{CC}} &= \langle \tilde{c} \rangle_{st}^{\mathrm{rw}} + \langle \tilde{c} \rangle_{ts}^{\mathrm{rw}} \\ &= \sum_{i=1}^{n} (l_{si}^+ - l_{st}^+ - l_{ti}^+ + l_{tt}^+ + l_{ti}^+ - l_{ts}^+ - l_{si}^+ + l_{ss}^+) \sum_{j\in Succ(i)} a_{ij} c_{ij} \\ &= (l_{ss}^+ + l_{tt}^+ - 2l_{st}^+) \sum_{(i,j)\in E} a_{ij} c_{ij}, \end{aligned} \tag{2.40}$$

which holds as the graph is undirected. Comparing this result with Equation (2.38) we see that the distances only differ from each other by a multiplying factor. Moreover, we see that this factor is

$$\frac{\Delta_{st}^{\mathrm{CC}}}{\Delta_{st}^{\mathrm{CT}}} = \frac{\sum_{i,j=1}^{n} a_{ij} c_{ij}}{\sum_{i,j=1}^{n} a_{ij}}. \tag{2.41}$$

□

The above relations between the resistance distance, commute time distance and commute cost distance can be gathered together as

$$\Delta_{st}^{\mathrm{Res}} = \frac{\Delta_{st}^{\mathrm{CT}}}{\sum\limits_{(i,j)\in E} a_{ij}} = \frac{\Delta_{st}^{\mathrm{CC}}}{\sum\limits_{(i,j)\in E} a_{ij}c_{ij}}. \tag{2.42}$$

This result means that when comparing distances between node pairs within a graph, any of the three measures will give the same result. However, it is good to note that the same is not true when comparing distances over different graphs, or when making changes in the graph. For instance, the resistance distance obeys *Rayleigh's monotonicity law* [59] stating that when the resistance of an edge is decreased, the resistance distances on the graph cannot increase. This principle, however, does not hold for the commute distances, which is discussed, for instance, in [140]. Consider the simplest possible example of two instances of a chain graph of three nodes, i, j, and k and two edges (i, j) and (j, k), first with unit edge resistances and affinities, $a_{ij} = a_{jk} = r_{ij} = r_{jk} = 1$. Consider then a second instance of the same graph, where the resistance of edge (i, j) is lowered to one tenth, i.e. $r_{ij} = 1/10$ and $a_{ij} = 10$. The distances in the first instance are $\Delta_{ij}^{\mathrm{Res}} = \Delta_{jk}^{\mathrm{Res}} = 1$ and $\Delta_{ij}^{\mathrm{CT}} = \Delta_{jk}^{\mathrm{CT}} = 2$, but in the second instance they become $\Delta_{ij}^{\mathrm{Res}} = 1/10$, $\Delta_{jk}^{\mathrm{Res}} = 1$, $\Delta^{\mathrm{CT}} = 11/10$, and $\Delta_{jk}^{\mathrm{CT}} = 11$. Thus, the monotonicity principle fails with $\Delta_{jk}^{\mathrm{CT}}$, as a random walker at node j is more likely to move away from node k after the decrease in r_{ij}.

In Equation (2.42), note also that if one considers the reciprocal relation between affinities and costs, $c_{ij} = 1/a_{ij}$, then the denominator on the right-hand side becomes the number of edges $\sum_{(i,j)\in E} a_{ij}c_{ij} = m$. Moreover, although the commute time and commute cost are well-defined on directed graphs, the resistance distance is only defined on undirected networks. Because of the strong relation between the resistance, commute time and commute cost distances (which can be considered as equivalent when dealing with only one, fixed graph), in the remainder of this chapter, we will use the term *random walk distances* to refer to the group of these three measures.

The random walk distances satisfy the triangle inequality and thus they are metrics (see, e.g. [118]). In addition, they are *squared Euclidean distances* [181, 77]. This means that the graph nodes can be embedded in $\mathbb{R}^k$, for some $k \leq n$, in such a way that the squared Euclidean distances between the points in that space equal the random walk distance between the corresponding nodes on the graph. The square root of the commute time distance was coined in [181, 77] the *Euclidean commute time distance*. This also partly motivates the use of

the random walk distances in machine learning applications, as the relations between nodes can be more easily studied and perceived in a Euclidean space. Furthermore, kernel methods, for instance, contain the assumption that the kernel used is positive-semidefinite, which is true for any kernel corresponding to an inner product matrix of vectors in a Euclidean space [186, 189]. It was shown in [181, 77] that the pseudoinverse of the Laplacian matrix of a graph, $\mathbf{L}^+$ is the kernel associated with the Euclidean commute-time distances, i.e. that the elements l^+_{st} correspond to inner products of (centered) vectors in $\mathbb{R}^n$ that are separated exactly by the Euclidean commute time distances on the graph. This *commute time kernel* has been studied for clustering [216, 217] and for collaborative recommendation [77, 78].

2.4.4 Computation of standard distance measures

The literature on computation of least costs is vast, with many algorithms, such as the Dijkstra algorithm, Bellman-Ford algorithm or the Floyd-Warshall algorithm [188], as well as heuristic algorithms which incorporate knowledge about the graph structure, such as the A* algorithm for spatial networks [178].

Random walk distances can also be computed in different ways, some of which, however, are only suitable for undirected graphs. The recent article [92] provides a thorough review, in addition to developing new methods, for computing expected hitting times and costs as well as commute time and commute cost distances on directed graphs.

2.4.5 Caveats of shortest path and random walk distances

The success of the least cost distance and the random walk distances has been faced recently with criticism and the discovery of surprising caveats. Least cost distance, for instance, only relies on the existence of a path of a certain distance, but does not otherwise take into account connectivity, i.e. the number of short connections between nodes. Moreover, on unweighted graphs, the least cost (i.e. in that case the shortest path) distance produces many ties when comparing distances between different node pairs. Also, unlike the random walk distances, the least cost distance is in most cases not Euclidean nor squared Euclidean, which makes it unappealing to use in graph-based machine learning.

One more potential hazard with the least cost distance can occur when constructing unweighted geometric graphs, i.e. graphs whose nodes correspond to points in a vector space, which are connected by edges to other points within

some limited neighborhood. Namely, if the edges are, indeed, defined as unweighted, then the least cost, i.e. shortest paths can often run through the periphery of the graph, instead of the more dense center of the graph, which is undesired and counter-intuitive [4].

Unlike the least cost distance, the random walk distances do take into account the connectivity between two nodes by actually considering *all* paths between them. Because of this the random walk distances have gained popularity especially in graph-based clustering and semi-supervised classification tasks, as they often manage to consider points within a well-connected group as closer to each other than to nodes in other groups. A good list of literature of applications of random walk distances is included in [142].

However, the random walk distances reflect connectivity in a too strong way — in large networks they become only dependent on the local connectivity of the nodes, in many cases the node degrees [210, 142]. This phenomenon was called the *lost-in-space phenomenon* in [210], and the *global information loss problem* in [161, 163].

One more feature that may discourage the use of random walk distances is that on a given graph they give equal distance to node pairs that are connected by only one self-avoiding path of equal length or cost (see e.g. Theorem D in [118]). This can be undesired, for instance, in applications where a node pair should be considered more related, when connected with a path through peripheral nodes than with a path through hub nodes (or vice versa).

2.4.6 Distances based on randomized shortest paths

Here we define two distance measures that have been derived from the RSP framework, namely the *RSP dissimilarity* and the *free energy distance*. One nice feature of both of the measures is that they interpolate, depending on the value of β, between the least cost distance, when $\beta \longrightarrow \infty$, and the commute cost distances, when $\beta \longrightarrow 0^+$. Thus, they can be considered as an effort to overcome the limitations of the least cost and random walk distances listed in Section 2.4.5.

RSP dissimilarity. Let us fix an s-t-pair, such that $s \neq t$, and consider the RSP distribution $\mathrm{P}_{st}^{\mathrm{RSP}}$ with a fixed value of the inverse temperature parameter, β. Now, similar to the definition of expected hitting cost in Equation (2.26), we consider the *expected cost* over the RSP distribution (we sometimes also use the

term *expected RSP cost*) from s to t

$$\langle \tilde{c} \rangle_{st} = \sum_{\wp \in \mathcal{P}_{st}} \mathrm{P}_{st}^{\mathrm{RSP}}(\wp)\tilde{c}(\wp). \tag{2.43}$$

The RSP dissimilarity is then defined simply as the symmetrized expected cost

$$\Delta_{st}^{\mathrm{RSP}} = \tfrac{1}{2}(\langle \tilde{c} \rangle_{st} + \langle \tilde{c} \rangle_{ts}). \tag{2.44}$$

It is not difficult to see that when $\beta \longrightarrow \infty$, $\Delta_{st}^{\mathrm{RSP}}$ converges to $\Delta_{st}^{\mathrm{SP}}$, and when $\beta \longrightarrow 0^{+}$, $\Delta_{st}^{\mathrm{RSP}}$ converges to $\frac{1}{2}\Delta_{st}^{\mathrm{CC}}$.

To see how the partition function is essential for deriving different quantities related to the RSP framework, note that the expected RSP cost from node s to node t can be expressed as

$$\begin{aligned} \langle \tilde{c} \rangle_{st} &= \sum_{\wp \in \mathcal{P}_{st}} \mathrm{P}_{st}^{\mathrm{RSP}}(\wp)\tilde{c}(\wp) = \frac{1}{\mathcal{Z}_{st}} \sum_{\wp \in \mathcal{P}_{st}} \mathrm{P}_{st}^{\mathrm{rw}}(\wp)\exp(-\beta\tilde{c}(\wp))\tilde{c}(\wp) \\ &= -\frac{1}{\mathcal{Z}_{st}}\frac{\partial \mathcal{Z}_{st}}{\partial \beta} = -\frac{\partial \log \mathcal{Z}_{st}}{\partial \beta}. \end{aligned} \tag{2.45}$$

The RSP dissimilarity was originally introduced in [215] with application to graph node clustering. Unfortunately, it was already noted there that Δ^{RSP} does not always (i.e. with some parameter values) satisfy the triangle inequality, and it is thus not a metric. The full method computation of expected costs and RSP dissimilarities is presented later, in Section 3.2.

Free energy distance. Considering the same setting as above, we define the *free energy* of an arbitrary probability distribution P_{st} on the set $\mathcal{P}_{st}$, as

$$\phi(\mathrm{P}_{st}) = \langle \tilde{c} \rangle_{st} + \tfrac{1}{\beta}\mathbb{J}(\mathrm{P}_{st} \| \mathrm{P}_{st}^{\mathrm{rw}}). \tag{2.46}$$

The reason for the name free energy for the above functional follows from the fact that the form is very similar to the Helmholtz free energy in thermodynamics. This functional was first studied in [115], presented in Chapter 3, and the same idea was explored independently in [17]. It is interesting for several reasons. The first interesting aspect is that the Gibbs-Boltzmann distribution of Equation 2.10 minimizes the free energy over all distributions (see [115] and Chapter 3). In other words, the minimization problem in Equation (2.4) could be stated equivalently as the problem of minimization of $\phi(\mathrm{P}_{st})$.

A second interesting point is that inserting the Gibbs-Boltzmann distribution $\mathrm{P}_{st}^{\mathrm{RSP}}$ of Equation (2.10) to Equation (2.46) gives the surprising form

$$\phi_{st} = \phi(\mathrm{P}_{st}^{\mathrm{RSP}}) = -\tfrac{1}{\beta} \log \mathcal{Z}_{st}, \tag{2.47}$$

which can be computed easily based on the result presented in Equation (2.18). This form of the free energy at the optimum is well-known in statistical physics. Other interesting phenomena related to the free energy are listed in Section 6.3 of the thesis.

The *free energy distance* between s and t is defined, again, by symmetrization

$$\Delta_{st}^{FE} = \tfrac{1}{2}(\phi_{st} + \phi_{ts}). \tag{2.48}$$

The form of free energy in Equation (2.47) was also used in [82], where its symmetrized version was called the *potential distance*. However, the definition of potential distance, though it was derived from the *bag-of-paths* framework instead of the RSP, is exactly equal to the definition of the free energy distance.

It was shown in [82] that Δ^{FE}

- interpolates between $\Delta_{st}^{\mathrm{SP}}$ (when $\beta \to \infty$), and $\frac{1}{2}\Delta_{st}^{\mathrm{CC}}$ (when $\beta \to 0^{+}$), similar to Δ^{RSP},
- satisfies the triangle inequality, and thus that it defines a metric between nodes on a graph, unlike the RSP dissimilarity, and
- is *cut-point additive*, or *graph-geodesic*, meaning that $\Delta_{st}^{\mathrm{FE}} = \Delta_{si}^{\mathrm{FE}} + \Delta_{it}^{\mathrm{FE}}$ only if all paths between s and t go through node i.

In addition, Δ^{FE} (as well as Δ^{RSP}) was shown in [96], at least with some example graphs, to be a squared Euclidean distance for some values of β, which is in a way expected, as their limit function, $\Delta_{st}^{\mathrm{CC}}$, is also squared Euclidean [181].

In addition to the above desirable properties for a graph node distance, Δ^{FE} was also evaluated in [82] in semisupervised classification with good results. Moreover, the graph node clustering results presented in Section 3.5.3 from [115], as well as further clustering results from [192] and [193] have shown Δ^{FE} to function well as a distance measure reflecting the cluster structure of networks.

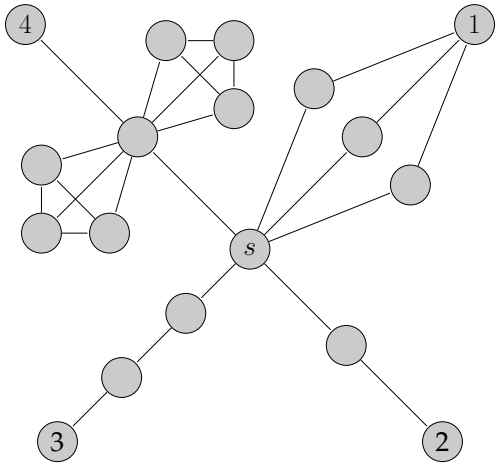


FIGURE 2.1: Example graph illustrating the caveats of shortest path and random walk distances on graphs

2.4.7 Illustration of traditional and RSP distances

Now that the main distance measures studied in the thesis have been defined, we may study them through some elementary examples. The first graph, depicted in Figure 2.1, illustrates some of the undesired features of the least cost and random walk distances listed in Section 2.4.5. The edge costs and affinities in the graph are set to unity and we study the distances between the central node s and the four numbered target nodes.

Below is a list of the least cost, commute time, and free energy distances between the central node s and the numbered peripheral nodes. In this example the behaviors of the RSP dissimilarity and free energy distance do not have any qualitative difference, so we present the results only for the free energy distance, Δ^{FE}. We show the Δ^{FE} for two different values of parameter β, for which Δ^{FE} ranks the target nodes differently.

From the table we see that all distance measures provide different rankings of the target nodes. First, Δ^{LC} is the most conservative one, as it gives (obviously) equal distance to targets 1, 2 and 4. Especially, Δ^{LC} is blind to the fact that there exist three paths of length 2 between node s and node 1, meaning that s and node 1 can be considered more strongly connected together than node s with node 2. This is noted by Δ^{CC}, as well as Δ^{FE}, for which the distance between s and node 1 is much smaller than between s and node 2.

t	$\Delta_{st}^{\mathrm{SP}}$	$\Delta_{st}^{\mathrm{FE}}, \beta = 0.1$	$\Delta_{st}^{\mathrm{FE}}, \beta = 0.01$	$\Delta_{st}^{\mathrm{CT}}$
1	2	6.60	13.27	16.67
2	2	11.11	32.41	50
3	3	15.43	45.00	75
4	2	18.71	40.59	50

TABLE 2.1: Distances according to different distance measures between the central node s and the four target nodes 1-4 in the graph of Figure 2.1.

However, Δ^{CC} considers node s at equal distance from node 2 and node 4. This is because there exists only one self-avoiding path between s and node 4, which implies $\Delta_{s4}^{\mathrm{SP}} = \Delta_{s4}^{\mathrm{Res}} = 2\Delta_{s4}^{\mathrm{CC}}/m$, according to Equation (2.42) with $m = 50$ the number of edges. Instead, for the free energy distance we have $\Delta_{s2}^{\mathrm{FE}} < \Delta_{s4}^{\mathrm{FE}}$, i.e. the free energy distance considers node s closer to node 2 than node 4. This can be a desirable result for some applications. For instance, in link prediction in social networks, node s may be more likely to become acquainted with node 2, as they are connected by a node with no other neighbors as s and 2, whereas s is connected to node 4 only through a hub node, and thus possibly less likely to become linked to it. Furthermore, with $\beta = 0.1$ we even have $\Delta_{s3}^{\mathrm{FE}} < \Delta_{s4}^{\mathrm{FE}}$. In other words, node 3, at three steps away from s, is still closer to s according to Δ^{FE} than node 4, which is only two steps away. In this case, the connection between s and 4 through the hub node is penalized even more than the simple connection between s and 3 of three steps. This result is a bit more questionable, but may nevertheless make sense in some situations. Similar experiments are also presented later, in Section 3.5.2.

The reason behind the above result is that node 4 can be considered more difficult to reach from node s than node 2. This aspect is engrained in the definition of free energy in Equation (2.46), as the relative entropy term, $\mathbb{J}(\mathrm{P}_{st} \| \mathrm{P}_{st}^{\mathrm{rw}})$, can be considered as an *information cost* of going from s to t, which is discussed in more depth in Section 6.3.3. Remembering, from Section 2.4.6, that for a given value of β the RSP distribution minimizes the free energy functional. But obtaining a distribution that focuses mostly on the direct, least cost path requires a larger deviation from the random walk distribution $\mathrm{P}_{st}^{\mathrm{rw}}$ when considering paths between s and 4 than when considering paths between s and 2, essentially because of the hub node residing in between nodes s and 4. Thus, in order to minimize the free energy, the RSP distribution focuses less on the least cost paths, thus obtaining a "reward" for deviating less from the random walk distribution.

When comparing different proposals for distance measures, one easily faces

the question of what defines a "good" distance measure [55]. The above example shows some of the problems related to this question when considering distances on graphs, as in the end it is difficult to argue that one of the columns in the above table is "better" than the others. The "goodness" of a distance measure can, to some extent, be determined according to its theoretical properties such as whether it satisfies the axioms of metric or some other general properties. The result in the above example with the graph of Figure 2.1, that the RSP distances penalize connections through hub nodes, can also be seen as a general feature of the RSP distance measures. Another type of feature of a distance measure can be observed with the *walk distances*, studied in [46], which sometimes have the property that distances between peripheral nodes are relatively longer than between central nodes.

A distance measure is often considered as a tool in tasks with a specific goal, such as the link prediction problem mentioned above, or the clustering and classification tasks considered in Chapter 3. In such cases goodness can often be quantified by empirical evaluation in achieving the desired goal. Furthermore, a distance measure can be considered good if it achieves desired goals in various different tasks of different nature. This requires a measure that allows some flexibility in its use in order to be suitable for different situations. The above example with the graph in Figure 2.1 shows that the RSP distances have that kind of flexibility in terms of parameter β, as with different values of β the RSP distances produce different kinds of results and interpretations. Finally, the goodness of a measure is partly determined by its practicality, for instance in terms of computational complexity, as well as conceptual complexity or intuitiveness. As ultimately a tool is often more useful if it is simple to learn to use and to understand.

As another illustrative example, we consider distances on a regular 2-D grid, consisting of 21 rows and 51 columns of nodes. We consider two standard constructions of such graphs, namely the simple "square grid", where each node is connected to its vertical and horizontal neighbors, with all affinities and costs set to unity, and also a "diagonal grid", where nodes are also connected to their diagonal neighbors, with the diagonal edges having cost $c_{ij} = \sqrt{2}$ and affinities $a_{ij} = 1/\sqrt{2}$.

This example is motivated mostly by the fact that it illustrates in a simple way the lost-in-space (LIS) phenomenon of the random walk distance measures discussed in Section 2.4.5, but it also shows an interesting feature of the RSP distance measures. Again, we only present the result for the free energy distance, as the plot obtained with the RSP dissimilarity is fairly similar to the free energy plot.

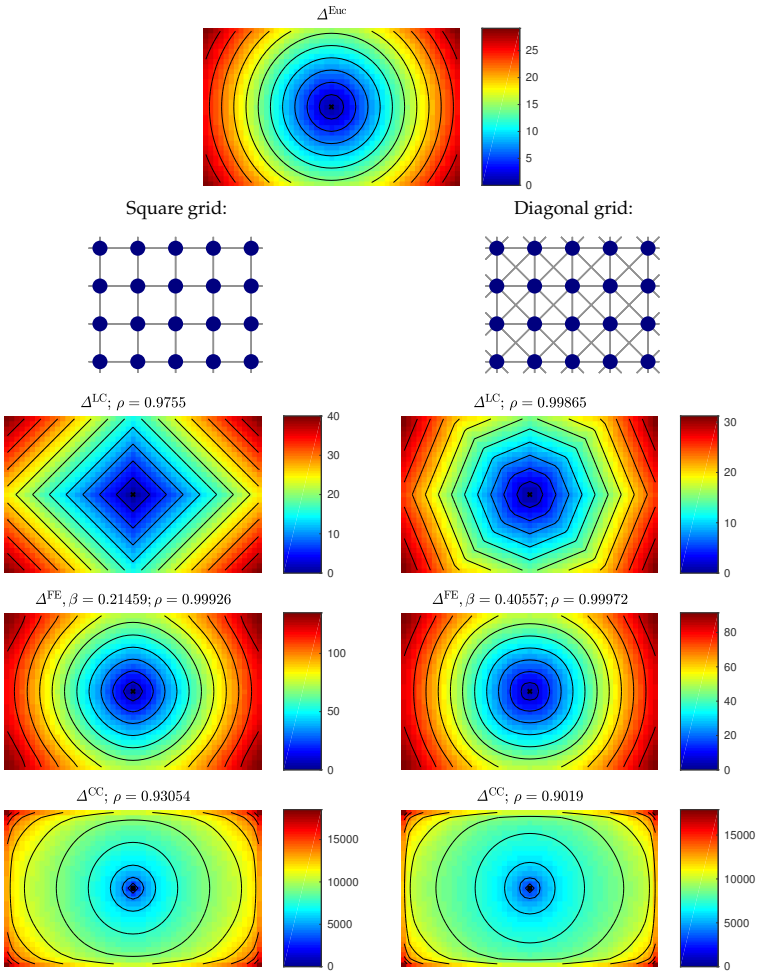


FIGURE 2.2: Comparison of graph node distances and the Euclidean distance on grids. The plots contain heatmaps and contour lines for distances measured from the center of the grid. The top center plot shows the Euclidean distances from the center. The left column shows heat maps for the "square grid" and the right column for the "diagonal grid" with the least cost distance, Δ^{LC}, free energy distance, Δ^{FE} and commute cost distance, Δ^{CC}. For each plot, the Pearson correlation ρ between all-pairs distances with the distance measure in question and with Δ^{Euc} is reported.

The plots in Figure 2.2 contain heatmaps and contour lines of distances between the center node of the grid and all other nodes. The most evident result from these plots is that the plot with the commute cost distances, Δ^{CC}, has a strange structure. Especially, it shows that the distances from the center to the edges, and especially the corners of the grid become unproportionally large, whereas distances to nodes within the interior of the grid become quite flat. This illustrates the LIS phenomenon of the random walk distances. As the grid size increases, the more the corresponding plot flattens, except for the edges of the grid, where nodes have a lower degree than in the interior.

A second observation about Figure 2.2 is about the shapes of the contour lines in the different plots. The least cost distance on the square grid corresponds to the L^1 distance on the plane. Accordingly, the contour lines of Δ^{LC} form a diamond-shape, underestimating the distances in diagonal directions. On the diagonal grid, Δ^{LC} approximates the Euclidean distance much better. The contour lines produced by Δ^{CC} are round close to the center, but become more oval-like for longer distances.

The most similar plots to the Euclidean one are provided by Δ^{FE}. The value of β for the plots has been selected so that the Pearson correlation between Δ^{FE} and Δ^{Euc} over all node-pairs is maximized. The optimal value of β is reported above the plots, in addition to the correlation coefficient with respect to the Euclidean distances, ρ, which is also reported for Δ^{LC} and Δ^{CC}. Comparison of the correlation coefficients shows that with the optimal value of β the free energy imitates the Euclidean distance much more closely than the traditional distance measures.

2.5 Graph node centrality

The concept of graph node centrality has many interpretations, and for most of the interpretations there exists a lengthy catalog of different proposed measures, which are often derived for different application purposes. In addition there have been efforts of stating axioms that a centrality measure should have [180, 35, 25]. Also, as discussed by Kolaczyk [123], there have been attempts to define a typology of centrality measures, for instance by Borgatti [30].

One main type of centrality measures are the different forms of *betweenness centrality*. A more thorough review of betweenness measures is presented in Chapter 5, which is based on [117], and where two betweenness centrality measures are derived from the RSP framework. These measures are the *simple RSP betweenness centrality* and the *RSP net betweenness centrality*, which are denoted, for node $i \in V$ by $\mathrm{bet}_i^{\mathrm{RSP}}$ and $\mathrm{bet}_i^{\mathrm{net}}$, respectively.

The derivation of the methods and their computation are developed in Chapter 5, but we state here the definitions in a concise form. Namely, for any $i \in V$ and for a fixed value of β we consider the RSP distributions (Equation (2.10)) over all s-t-pairs on the graph and define

$$\begin{aligned} \mathrm{bet}_i^{\mathrm{RSP}} &= \sum_{s=1}^{n}\sum_{t=1}^{n}\sum_{j\in Succ(i)} \bar{\eta}_{ij}(s,t) \quad \text{and} \\ \mathrm{bet}_i^{\mathrm{net}} &= \sum_{s=1}^{n}\sum_{t=1}^{n}\sum_{j\in Succ(i)} |\bar{\eta}_{ij}(s,t) - \bar{\eta}_{ji}(s,t)|. \end{aligned} \tag{2.49}$$

Again, varying the value of β affects the shape of the RSP distributions and thus also the dispersal of betweenness scores on the nodes of the graph. When $\beta \longrightarrow \infty$, the distributions converge to the shortest path likelihood distribution, defined in Equation 2.11, and both $\mathrm{bet}_i^{\mathrm{RSP}}$ and $\mathrm{bet}_i^{\mathrm{net}}$ converge to the *shortest path likelihood betweenness centrality*, which was introduced in [117]. The shortest path likelihood betweenness is very closely related to the original shortest path betweenness centrality, which is one of the most common conceptions of graph node centrality, and which is presented and reviewed in Section 5.2.

At the random walk limit, when $\beta \longrightarrow 0^+$, the simple RSP betweenness converges to the *stationary distribution* (multiplied by a constant) of the natural random walk on G, i.e. the unique vector $\boldsymbol{\pi}$ for which $\boldsymbol{\pi}^{\mathsf{T}}\mathbf{P}^{\mathrm{rw}} = \boldsymbol{\pi}$. This is a surprising result, as it entails the result that the sum of expected numbers of visits to a node over hitting paths over the distributions $\mathrm{P}_{st}^{\mathrm{rw}}$ is proportional to the stationary probability of a node. The result is proved explicitly in Section 5.2.2 and it seems that the exact same result has not appeared in literature. The proof also has an interesting connection with a very recent paper [26] that develops methods for computation of random walk related quantities. The RSP net betweenness, as $\beta \longrightarrow 0^+$, converges to the *current flow betweenness centrality*, which has an electric network analogue, similar to the resistance distance in the context of distance measures. The current flow betweenness is also discussed and studied more in Chapter 5.

The above convergence properties make the RSP betweenness measures appealing, in addition to which they can be shown to detect some aspects in node roles in networks that the more traditional limiting measures do not notice, as demonstrated in Chapter 5. Furthermore, the simple RSP betweenness can also be interpreted as a sensitivity index, as it quantifies the change in the sum of free energy distances of a graph caused by a perturbation of the edge costs of edges leaving from a node. This property is studied in more depth in Section 6.4.

Besides betweenness, centrality has also been characterized with other additional terms such as *closeness*, *feedback* and *criticality*, to name a few [124]. Such descriptions or groupings of centralities however always overlap to some extent. This is also evident from the simple RSP betweenness, which, thanks to the interpretation in terms of sensitivity analysis, can be considered also as a vitality measure, in addition to being a betweenness measure. The concept of centrality can also be considered with respect to edges, instead of nodes. Although in this work we focus on centrality according to the betweenness interpretation, we nonetheless mention some of the other interpretations in this section, as the RSP framework could be combined with many of them too.

Closeness centrality measures are based on some interpretation of overall proximity of a node to the other nodes of a network. The *shortest path closeness centrality* [19, 180] of node s is classically defined based on shortest path distances as

$$C_s^{\text{Close}} = \frac{1}{\sum_{t=1}^{n} \Delta_{st}^{\text{SP}}}. \tag{2.50}$$

Considering communication on a network, betweenness centrality can be interpreted as the amount of *control* of a node, whereas closeness centrality measures the *efficiency* of the communication of the node [84, 207].

Instead of using the shortest path distance to define closeness centrality, other distances can be used too. For instance, White and Smyth [213] define *Markov centrality* by replacing the shortest path distance in the closeness centrality definition with the expected hitting time, $\langle L \rangle_{st}^{\text{rw}}$. Related to that, Brandes and Fleischer [36] considered a current flow analogy of closeness centrality by replacing the distance between two nodes with their potential difference. They managed to show that the current flow closeness centrality is equivalent to the information centrality defined by Stephenson and Zelen [194]. The equivalence has been confirmed with another proof by Bozzo and Franseschet [33].

Feedback centrality measures, also often termed as *spectral* centrality measures, are in many ways related to random walk based measures. The *eigenvector centrality* [28] is the archetype feedback measure and is based on the idea that the centrality of a node should be the sum of the centralities of its neighbors. The solution of the formulation is the eigenvector of the adjacency matrix corresponding to the largest eigenvalue. The previously mentioned PageRank [164] can also be considered in this sense and formulated as an eigenvector. The Katz centrality [114] is based on the same idea of feedback, but considers also longer dependencies than only the neighbors of a node. The effect of longer dependencies decays according to the distance between nodes. The *subgraph centrality* [69] and *communicability betweenness* [68] are based on similar ideas using the matrix exponential of the adjacency matrix. The relationship between

all the feedback centrality measures listed above has been studied by Benzi and Klymko [20].

2.6 Related work

There exists already a lot of recent literature concentrating on measures based on similar, or even equivalent considerations graph-based measures as the ones derived within the RSP framework. With some of these studies the connection with RSPs is fairly clear, but in some cases the relation can be more difficult to see. In this section we review the relevant literature and try to elaborate the connections and differences between the different methods.

As already stated, the RSP framework was originally presented in [182] inspired by the model developed in [1] in transportation science. The RSP dissimilarity was introduced in [217] where it was shown to function well in graph node clustering. In [87], a the consideration of RSPs was extended to continuous space, showing that the continuous-state version corresponds to a diffusion process with a potential-like drift towards the destination. The RSP framework was generalized as the *bag-of-paths* (BoP) framework in [81, 82] by defining a Gibbs-Boltzmann distribution similarly to Equation 2.10, but over the set of *all* paths of a graph, instead of only the set of paths between a fixed s-t-pair. In the BoP framework, the inverse temperature parameter β has a slightly different effect from the RSP framework in that it controls the *outreach* of the distribution — with large values of β, the distribution focuses only on paths between close-by nodes, whereas with values of β close to zero, the distribution spreads also over paths between more distant node-pairs.

Two distance measures were defined within the BoP framework in [81, 82]. The first one, called the *surprisal distance*, quantifies the amount of surprisal, or information associated with the drawing of a path between a given s-t-pair from the bag of paths. The second, potential distance, is equivalent to the free energy distance, Δ^{FE} derived earlier from the RSP framework in Section 2.4.6.

The BoP framework was extended further to define a *path-based modularity* measure for community detection, presented in Chapter 4, which was used for semi-supervised classification with good results in [54]. In addition, the *bag-of-paths betweenness* measure was developed for semi-supervised classification [129] and the *bag-of-paths criticality* measure for ranking nodes in a graph based on the effect of their removal from the graph to the overall connectivity of the graph [128]. The idea of the BoP framework was already engrained in the earlier article [146] which defined a *sum-over-paths covariance kernel* which considers two nodes similar to each other when they appear often on the same

paths, based on the same Gibbs-Boltzmann distribution as used in the BoP framework.

Graph measures similar to the ones presented in this thesis have been defined and studied in parallel in a series of papers [17, 96, 95] focusing on free energy minimization in a similar fashion to the free energy minimization interpretation of the RSP distribution, as discussed in Section 2.4.6. The free energy in the listed papers is expressed according to *local* relative entropies, between the *biased* and unbiased transition probabilities on edges, in contrast to the RSP framework, where the relative entropy is considered according to the distribution over paths. As a continuation of this series, a collaborative paper [97] extended the BoP framework for situations where the distributions on the source nodes and target nodes are fixed, which can be interpreted as a randomized version of the problem of *optimal transportation on a graph*.

Results related to the lost-in-space phenomenon of random walk distances, discussed in Section 2.4.5, have been gathered in the article [142]. Extensions of those results were proposed in a recent master's thesis [163]. Remedies for the LIS phenomenon were proposed in [210], with the *amplified commute time distance*, in [3] with *p-resistance distance*, which is studied more in Chapter 3, and in [161] with R_p *and* R_{12} distances. Moreover, in [103], the authors consider a scaled *logarithmic Laplace transformed hitting time (log-LHTH)* as a distance measure, both in continuous space and on geometric graphs. The definition is seemingly equivalent to the free energy distance, Δ^{FE}, which is also noted by the authors, although no other connection is made to free energy. It is also shown for the p-resistance, R_p, R_{12}, and log-LHTH distances in the cited works that these distances indeed avoid the LIS phenomenon. Specifically, as the result holds for the log-LHTH distance, it should also hold for the free energies, ϕ, although verifying this is challenging because of the different formalisms.

Another line of work, by Chebotarev et al., has resulted in the proposal of several different distance measures on graphs, many of which are considered in [46]. The logarithmic forest distance proposed in [45] is studied more extensively in 3. Also, in [110], several graph node distances were evaluated in clustering and many of the distance measures that performed well were the ones involving a logarithmic transformation, including the free energy distance, $\Delta^{\mathrm{FE}}_{st}$.

The works cited in this section so far are all mostly related to machine learing applications. However, ideas similar to the RSP framework, and especially the free energy, ϕ_{st}, have also appeared in other contexts. In [196] an algorithm is developed for computing a distributed consensus on a graph of agents based

on the *log-sum-exp* function with a parameter. A special version of the algorithm generalizes, at the ends of the parameter range, the Bellman-Ford algorithm for computing the shortest path distances to a node, and the iteration for computing expected hitting times to a node. The log-sum-exp interpretation of the free energy, ϕ, is discussed in Section 6.3.2.

Another parametrized method for the computation of shortest path and commute time distances is developed in [91, 90] by considering *evaporating random walks* (called killed random walks in this thesis) and computing the path taken by a walker conditioned on its survival to a target node. A similar interpretation applies to the RSP framework, as discussed in Section 6.3.1. In [138, 137] a routing scheme is devised based on minimization of a parameterized mixture of L_1 and L_2 norms of the vector of flows on the edges such that the flow interpolates between the shortest path flow and the potential (electric) flow.

The authors in [108] propose the *random walk effective distance* for estimating disease arrival times in epidemic spreading which is essentially equivalent to the free energy, ϕ, but where the inverse temperature parameter, β, is replaced with a parameter incorporating the infection, recovery, and mobility rates associated to the epidemic. It is noted there that the measure is in fact the cumulant-generating function of the expected hitting time, although the log-Laplace transform interpretation of [103], mentioned above, is perhaps more accurate.

Finally, randomization from optimality has been studied a lot in reinforcement learning and Markov decision processes. In [9] a *mellowmax* operator is introduced in the SARSA algorithm for computing a soft policy for Markov decision processes. This has similarities with our parallel work [130], which proposes a randomization of the policy based on the RSP framework and free energy minimization. The connection between soft optimum functions and the free energy are also discussed in Section 6.3.2. Ideas similar to free energy minimization also appear in stochastic optimal control [205, 177, 80, 112, 204, 10, 11]. Specifically, in [203], many of the proposed frameworks for solving optimal control problems are expressed in a unified way in terms of free energy minimization.

Chapter 3

Developments in the theory of randomized shortest paths with a comparison of graph node distances

3.1 Introduction

Defining distances and similarities between nodes of a graph based on its structure has become an essential task in the analysis of network data [211, 201, 48, 139, 123, 136, 154, 141, 64]. In the simplest case, a binary network can be presented as an adjacency matrix or adjacency list which can be difficult to interpret. Acquiring meaningful information from such data requires sophisticated methods which often need to be chosen based on the context. Being able to measure the distance between the nodes of a network in a meaningful way of course provides a fundamental way of interpreting the network. With the information of distances between the nodes, one can apply traditional multivariate statistical or machine learning methods for analyzing the data.

As discussed in Section 2.4, the most common ways of defining a distance on a graph are to consider either the lengths or costs of the shortest paths between nodes, leading to the definition of the *shortest path distance* or *least cost distance*, or the expected lengths or costs of random walks on the graph, which can be used to derive the *commute time distance* [88] and the *commute cost distance* [42, 77]. Remember also from Section 2.4.2, that the commute cost, commute time, and resistance distances are all equal up to a constant factor. The main topic of this chapter is the examination of generalized distances

on graphs that interpolate, depending on a parameter, between the least cost distance and the commute time or resistance distance.

The chapter contains several separate contributions: First, in Section 3.2, we develop the theory of RSP dissimilarities, presented in Section 2.4.6 [215, 182]. We derive a new algorithm for computing it for all pairs of nodes of a graph in closed form, and thus much more efficiently than before. In Section 3.3 we then derive another generalized distance from the RSP framework based on the Helmholtz free energy between two states of a thermodynamic system. We show that this *free energy distance* actually coincides with the *potential distance*, proposed in the parallel work [82]. However, the new derivation presented here gives a nice theoretical background for the distance. In Section 3.4 we review different generalized graph node distances and in Section 3.5 make a comparison of the behavior and performance of those distances. The comparisons are conducted by observing the relative differences of distances between nodes in small example graphs and by examining the performance of the different distance measures in clustering and classification tasks. Finally, Section 3.6 sums up the content of the chapter.

3.2 Algorithm for faster computation of RSP dissimilarities

In this section we derive the result expressed in Section 2.3 for the computation of the partition functions of hitting paths, $\mathcal{Z}_{st}$, for all s-t-pairs on a graph with simple matrix operations (see Equations (2.18), (2.20) and (2.21)).

Based on this, we develop an algorithm that allows computing the set of all pairwise RSP dissimilarities between the nodes of a graph in a batch mode. The algorithm in the original reference [215] performs a loop over all the nodes of the graph and computes the needed quantities considering one node as the destination node at a time. Our new algorithm is based solely on matrix manipulations and can thus provide faster execution than a naïve looping.

We will start by proving the result stated in Equation (2.18), namely that the partition function of hitting paths, $\mathcal{Z}_{st}$, for any s-t-pair such that $s \neq t$, can be computed based on the fundamental matrix of regular paths, $\mathbf{Z}$, from Equation 2.19, as $\mathcal{Z}_{st} = z_{st}/z_{tt}$. Recall, that it was shown in Section 2.3 that $\mathcal{Z}_{st}$ is given by element (s,t) of the fundamental matrix of killed absorbing walks to t, $\widehat{\mathbf{Z}} = (\mathbf{I} - \widehat{\mathbf{W}})^{-1}$, where $\widehat{\mathbf{W}}$ is matrix $\mathbf{W}$ where row t is set to zero, and is thus dependent on the target node.

In the original reference [215], the authors then derived a way of expressing the matrix $\widehat{\mathbf{Z}}$ based on matrix $\mathbf{Z} = (\mathbf{I} - \mathbf{W})^{-1}$, from Equation (2.19), i.e. the fundamental matrix of the killed, non-absorbing Markov chain, which is not dependent on t. This is done in [215] by applying the Sherman-Morrison rule for a rank-one update of a matrix (Equation (20) in [215]):

$$\widehat{\mathbf{Z}} = \mathbf{Z} - \frac{\mathbf{Z}\mathbf{e}_t(\mathbf{w}_t^{\mathrm{r}})^{\mathsf{T}}\mathbf{Z}}{1 + (\mathbf{w}_t^{\mathrm{r}})^{\mathsf{T}}\mathbf{Z}\mathbf{e}_t}, \tag{3.1}$$

where $\mathbf{w}_t^{\mathrm{r}} = (\mathbf{e}_t^{\mathsf{T}}\mathbf{W})^{\mathsf{T}}$ is row t (hence the superscript "r") of matrix $\mathbf{W}$ as a column vector.

They then use this form of $\widehat{\mathbf{Z}}$ for computing a vector of dissimilarities from all nodes of the graph to one fixed absorbing destination node t at a time. In order to compute all dissimilarities in the graph, the algorithm performs a loop over t across all the nodes of the graph considering them as absorbing one at a time.

However, what was left unnoticed from the authors is that as we are only interested in computing the element (s, t) of the matrix $\widehat{\mathbf{Z}}$, that can be done by using (3.1) as:

$$\begin{aligned} \mathcal{Z}_{st} = \mathbf{e}_s^{\mathsf{T}}\widehat{\mathbf{Z}}\mathbf{e}_t = z_{st} - \frac{z_{st}(\mathbf{w}_t^{\mathrm{r}})^{\mathsf{T}}\mathbf{Z}\mathbf{e}_t}{1 + (\mathbf{w}_t^{\mathrm{r}})^{\mathsf{T}}\mathbf{Z}\mathbf{e}_t} = z_{st}\left(1 - \frac{1 + (\mathbf{w}_t^{\mathrm{r}})^{\mathsf{T}}\mathbf{Z}\mathbf{e}_t - 1}{1 + (\mathbf{w}_t^{\mathrm{r}})^{\mathsf{T}}\mathbf{Z}\mathbf{e}_t}\right) \\ = \frac{z_{st}}{\mathbf{e}_t^{\mathsf{T}}\mathbf{e}_t + \mathbf{e}_t^{\mathsf{T}}\mathbf{W}\mathbf{Z}\mathbf{e}_t} = \frac{z_{st}}{\mathbf{e}_t^{\mathsf{T}}(\mathbf{I} + \mathbf{W}\mathbf{Z})\mathbf{e}_t} = \frac{z_{st}}{\mathbf{e}_t^{\mathsf{T}}\mathbf{Z}\mathbf{e}_t} = \frac{z_{st}}{z_{tt}}, \end{aligned} \tag{3.2}$$

where $\mathbf{Z} = \mathbf{I} + \mathbf{W}\mathbf{Z}$ follows from $(\mathbf{I} - \mathbf{W})\mathbf{Z} = \mathbf{I}$.

Thus, we can compute the matrix $\mathbf{Z}_{\mathrm{h}}$, whose element (s, t) is the partition function $\mathcal{Z}_{st}$ of hitting paths from node s to node t without having to compute $\widehat{\mathbf{Z}}$ for each t separately. This can be done simply using the matrix $\mathbf{Z}$ as

$$\mathbf{Z}_{\mathrm{h}} = \mathbf{Z}\mathbf{D}_{\mathbf{Z}}^{-1}, \tag{3.3}$$

where $\mathbf{D}_{\mathbf{Z}} = \mathbf{Diag}(\mathbf{Z})$ is the diagonal matrix with elements z_{ii} on its diagonal. The advantage compared to the original algorithm presented in [215] is that $\mathbf{Z}_{\mathrm{h}}$ can be used for computing the RSP dissimilarities between all pairs of nodes instead of having to apply Equation (3.1) and then computing the RSP dissimilarities separately for each target t. This reduces the computational complexity of the computation of $\mathcal{Z}_{st}$ and makes implementation of computation of other relevant quantities much more straightforward to than before.

Note that a similar derivation for computing $\mathcal{Z}_{st}$ as the above one is also presented in Appendix A of [82]. It is also showed there that, in fact, the partition

function in this context gives the probability that a random walker starting from s using the transition matrix $\mathbf{W}_t^{\mathrm{h}}$ will reach t before getting killed, i.e. before ending in the imaginary cemetery node. This interpretation of the RSP probabilities is discussed more in Section 6.3.1.

Now we can finally derive the new matrix formula for the RSP dissimilarities between all pairs of nodes. Remember, from Equation (2.45) that the expected RSP cost, $\langle \tilde{c} \rangle_{st}$, from s to t can be computed as the derivative of the logarithm of the partition function, which can now be written further, thanks to Equation (3.2), as

$$\langle \tilde{c} \rangle_{st} = -\frac{\partial \log \mathcal{Z}_{st}}{\partial \beta} = -\frac{\partial \log(z_{st}/z_{tt})}{\partial \beta} = -\frac{\partial \log z_{st}}{\partial \beta} + \frac{\partial \log z_{tt}}{\partial \beta}. \tag{3.4}$$

The first term can be computed by

$$\begin{aligned}
\frac{\partial \log z_{st}}{\partial \beta} &= \frac{1}{z_{st}} \frac{\partial z_{st}}{\partial \beta} = \frac{1}{z_{st}} \frac{\partial \mathbf{e}_s^\mathsf{T} \mathbf{Z} \mathbf{e}_t}{\partial \beta} = \frac{1}{z_{st}} \mathbf{e}_s^\mathsf{T} \frac{\partial (\mathbf{I} - \mathbf{W})^{-1}}{\partial \beta} \mathbf{e}_t \\
&= -\frac{1}{z_{st}} \mathbf{e}_s^\mathsf{T} (\mathbf{I} - \mathbf{W})^{-1} \frac{\partial (\mathbf{I} - \mathbf{W})}{\partial \beta} (\mathbf{I} - \mathbf{W})^{-1} \mathbf{e}_t \\
&= \frac{1}{z_{st}} \mathbf{e}_s^\mathsf{T} \mathbf{Z} \frac{\partial \mathbf{W}}{\partial \beta} \mathbf{Z} \mathbf{e}_t \\
&= -\frac{1}{z_{st}} \mathbf{e}_s^\mathsf{T} \mathbf{Z} (\mathbf{C} \circ \mathbf{W}) \mathbf{Z} \mathbf{e}_t,
\end{aligned}$$

where we used $\dfrac{\partial \mathbf{W}}{\partial \beta} = \dfrac{\partial}{\partial \beta} (\mathrm{P}_{st}^{\mathrm{rw}} \circ \exp(-\beta \mathbf{C})) = -(\mathbf{C} \circ \mathbf{W})$.

Thus, we can write Equation (3.4) as

$$\langle \tilde{c} \rangle_{st} = \frac{\mathbf{e}_s^\mathsf{T} \mathbf{Z} (\mathbf{C} \circ \mathbf{W}) \mathbf{Z} \mathbf{e}_t}{z_{st}} - \frac{\mathbf{e}_t^\mathsf{T} \mathbf{Z} (\mathbf{C} \circ \mathbf{W}) \mathbf{Z} \mathbf{e}_t}{z_{tt}} \tag{3.5}$$

Let us then denote $\mathbf{S} = (\mathbf{Z}(\mathbf{C} \circ \mathbf{W})\mathbf{Z}) \div \mathbf{Z}$, where $\div$ marks elementwise division. In fact, $\mathbf{S}$ is the matrix form of the first term on the right side of Equation (3.5), and contains the expected costs of regular paths. We can thus write out the matrix form of computing all the expected RSP costs over hitting paths as

$$\overline{\mathbf{C}} = \mathbf{S} - \mathbf{e}\mathbf{d}_{\mathbf{S}}^\mathsf{T},$$

where $\mathbf{d}_{\mathbf{S}} = \mathbf{diag}(\mathbf{S})$ is the vector of diagonal elements of $\mathbf{S}$. Finally, the matrix of RSP dissimilarities $\boldsymbol{\Delta}^{\mathrm{RSP}}$ is defined by symmetrizing $\overline{\mathbf{C}}$: $\boldsymbol{\Delta}^{\mathrm{RSP}} = \frac{1}{2}(\overline{\mathbf{C}} + \overline{\mathbf{C}}^\mathsf{T})$. The whole procedure of computing $\boldsymbol{\Delta}^{\mathrm{RSP}}$ is presented in Algorithm 1, along with the computation of free energy distances, developed in the next section.

The advantage of the new algorithm compared to the one presented in [215] is that the matrix of dissimilarities can be computed in closed form by matrix multiplication instead of a loop.

The RSP dissimilarity converges to the least cost distance as $\beta \to \infty$ and to the commute cost distance as $\beta \to 0^+$. The reason why the RSP dissimilarity is called a dissimilarity, rather than a distance, is that for intermediate values of the parameter β, it does not satisfy the triangle inequality. In the experimental part of the chapter, we focus on the effect of the choice of a distance measure on clustering and classification algorithms. When studying such algorithms, it is often assumed that they are used in conjunction with a metric, i.e., a distance measure that satisfies the triangle inequality. Also, the triangle inequality can be exploited in order to improve the efficiency of some distance-based algorithms, cf. [62]. However, in our study, we use a kernel k-means algorithm for clustering and a simple 1-nearest-neighbor algorithm for semi-supervised classification, which both can be used even with a dissimilarity that does not satisfy the triangle inequality. Furthermore, it has been already shown that using the RSP dissimilarity with its intermediate parameter values provides good results in graph node clustering [215].

3.3 A new generalized distance based on Helmholtz free energy

As already mentioned earlier, one of the drawbacks of the RSP dissimilarity is that it is not a metric as it does not necessarily satisfy the triangle inequality for intermediate values of β. To overcome this problem we derive a new distance measure called the *free energy (FE) distance*, which is based on the same idea behind the RSP dissimilarity. We conclude that the proposed FE distance is actually the same as the *potential distance* defined, in parallel with this work, in Equation (33) of [82] based on a *bag-of-paths framework*. However, in that reference, the derivation of the potential distance is left rather unmotivated. The derivation provided here gives a more sound theoretical background to the distance measure and thus we suggest to call the distance the free energy distance instead of the potential distance.

Free energy has already been used in various contexts in network theory. In [53], the authors define a ranking method called the free-energy rank (in the spirit of the well-known PageRank [164]) by computing the transition probabilities minimizing the free energy rate encountered by a random walker. Then, the stationary distribution of the defined Markov chain is the free-energy rank score. In [17], the authors compute edge flows minimizing the free energy

between two nodes. The resulting flows define some new edge and node betweenness measures, balancing exploration and exploitation through an adjustable temperature parameter. Their model is quite close to the RSP framework and was developed parallel to our article. However, the authors do not define a distance measure based on the free energy.

We now derive the FE distance and then show that it coincides with the potential distance. Now, instead of the expected cost, let us consider a random walker choosing a path from node s to node t according to the distribution P_{st} over the set $\mathcal{P}_{st}$ that minimizes the free energy, as defined in Equation (2.46), recalled here for convenience:

$$\phi(\mathrm{P}_{st}) = \langle \tilde{c} \rangle_{st} + \tfrac{1}{\beta}\mathbb{J}(\mathrm{P}_{st} \| \mathrm{P}^{\mathrm{rw}}_{st}). \tag{3.6}$$

Recall now, from Section 2.3, that originally the RSP distribution was defined in Equation (2.4) according to minimization of the expected cost $\langle \tilde{c} \rangle_{st}$ subject to a relative entropy constraint. We can instead consider the minimization of free energy, and solve the problem

$$\begin{aligned} &\underset{\mathrm{P}_{st}}{\text{Minimize}} \sum_{\wp \in \mathcal{P}_{st}} \mathrm{P}_{st}(\wp)\tilde{c}(\wp) + \frac{1}{\beta} \sum_{\wp \in \mathcal{P}_{st}} \mathrm{P}_{st}(\wp) \log(\mathrm{P}_{st}(\wp)/\mathrm{P}_{st}(\wp)), \\ &\text{subject to} \sum_{\wp \in \mathcal{P}_{st}} \mathrm{P}_{st}(\wp) = 1. \end{aligned} \tag{3.7}$$

It is not difficult to see that this problem becomes equivalent to the minimization problem (2.4) and thus the optimal solution is again the Gibbs-Boltzmann distribution $\mathrm{P}^{\mathrm{RSP}}_{st}$ from Equation (2.10). Denoting $\phi_{st} = \phi(\mathrm{P}^{\mathrm{RSP}}_{st})$, we define the free energy distance between nodes s and t as the symmetrized minimum free energy between these two nodes, in other words

$$\Delta^{\mathrm{FE}}_{st} = \tfrac{1}{2}(\phi_{st} + \phi_{ts}). \tag{3.8}$$

In order to show that the FE distance coincides with the potential distance defined in Equation (33) of [82], we insert the RSP probability (Equation (2.10)) into Equation (3.6). Using the fact that $\sum_{\wp \in \mathcal{P}_{st}} \mathrm{P}^{\mathrm{RSP}}_{st}(\wp) = 1$, we can write out

the expression for the relative entropy:

$$\begin{aligned}\mathbb{J}(\mathrm{P}_{st}\|\mathrm{P}^{\mathrm{rw}}_{st}) &= \sum_{\wp\in\mathcal{P}_{st}} \mathrm{P}^{\mathrm{RSP}}_{st}(\wp)\log\big(\mathrm{P}^{\mathrm{RSP}}_{st}(\wp)/\mathrm{P}^{\mathrm{rw}}_{st}(\wp)\big)\\ &= \sum_{\wp\in\mathcal{P}_{st}} \mathrm{P}^{\mathrm{RSP}}_{st}(\wp)\log\left(\frac{\exp(-\beta\tilde{c}(\wp))}{\mathcal{Z}_{st}}\right)\\ &= -\beta\langle\tilde{c}\rangle_{st} - \log(\mathcal{Z}_{st})\end{aligned}$$

When combining this result with Equation (3.6), the FE becomes

$$\phi_{st} = -\tfrac{1}{\beta}\log(\mathcal{Z}_{st}) \tag{3.9}$$

which after the symmetrization (3.8) equals the potential distance defined in Equation (33) of [82]. Thanks to the derivation of the new algorithm for the RSP dissimilarities in Section 3.2, the matrix of free energies between all pairs of nodes

$$\boldsymbol{\Phi} = -\tfrac{1}{\beta}\log\mathbf{Z}_{\mathrm{h}} \tag{3.10}$$

can also be computed straightforwardly by performing Equations (2.12), (2.19) and (3.3). Again, the matrix of all FE distances is given by symmetrization as $\boldsymbol{\Delta}^{\mathrm{FE}} = \frac{1}{2}(\boldsymbol{\Phi} + \boldsymbol{\Phi}^{\mathsf{T}})$. The computation of $\boldsymbol{\Delta}^{\mathrm{FE}}$ is presented, along with the computation of the matrix $\boldsymbol{\Delta}^{\mathrm{RSP}}$ of all RSP dissimilarities, in pseudocode in Algorithm 1.

Thus, we have shown that the potential distance derived within the bag-of-paths framework in [82], in fact can be derived from the RSP framework by considering the minimum Helmholtz free energy as the distance, instead of the minimum expected cost. The minimization of the FE can be interpreted as a regularized version of the RSP model where the random walker finds a compromise between the expected cost and predictability of its path from s to t. The compromise is controlled by the inverse temperature β.

Of course, the FE distance also satisfies all the properties that were proved for the potential distance in [82]. Most importantly, it was shown that the distance obeys the triangle inequality (in Inequality (37) of [82]) and is thus a metric, as opposed to the RSP dissimilarity. The distance also converges to the least cost distance when $\beta \to \infty$ and to the commute cost distance[1] when $\beta \to 0^+$ (see Appendices D and E in [82]). In addition, in Appendix C of [82], it is shown to be *graph geodetic* [45, 120], meaning that $\Delta^{\mathrm{FE}}_{st} = \Delta^{\mathrm{FE}}_{sk} + \Delta^{\mathrm{FE}}_{kt}$ if and only if all paths from node s to node t go through node k. This shows that the minimum free energy between two nodes defines a meaningful

[1] To the commute cost distance divided by 2, to be precise.

Algorithm 1 Computation of the matrices of all pairwise RSP dissimilarities and free energy distances of G.

Input:
- A graph G containing n nodes
- The $n \times n$ affinity matrix $\mathbf{A}$
- The $n \times n$ cost matrix $\mathbf{C}$
- The inverse temperature parameter β

Output:
- The $n \times n$ matrices containing the RSP dissimilarities $\Delta_{st}^{\mathrm{RSP}}$ and free energy distances $\Delta_{st}^{\mathrm{FE}}$ for all $s, t \in V$

1. $\mathbf{D} \leftarrow \mathbf{Diag}(\mathbf{Ae})$
2. $\mathbf{P}^{\mathrm{rw}} \leftarrow \mathbf{D}^{-1}\mathbf{A}$ ▷ reference transition probability matrix
3. $\mathbf{W} \leftarrow \mathbf{P}^{\mathrm{rw}} \circ \exp[-\beta \mathbf{C}]$ ▷ likelihood matrix
4. $\mathbf{Z} \leftarrow (\mathbf{I} - \mathbf{W})^{-1}$ ▷ fundamental matrix
5. $\mathbf{S} \leftarrow \mathbf{Z}(\mathbf{C} \circ \mathbf{W})\mathbf{Z} \div \mathbf{Z}$
6. $\mathbf{d_S} \leftarrow \mathbf{diag}(\mathbf{S})$ ▷ vector of diagonal values of $\mathbf{S}$
7. $\overline{\mathbf{C}} \leftarrow \mathbf{S} - \mathbf{e}\mathbf{d}_\mathbf{S}^\mathsf{T}$ ▷ matrix of expected costs
8. $\mathbf{D}_\mathbf{Z}^{-1} \leftarrow \mathbf{Diag}(\mathbf{Z})$
9. $\mathbf{Z}_{\mathrm{h}} \leftarrow \mathbf{Z}\mathbf{D}_\mathbf{Z}^{-1}$ ▷ fundamental matrix of hitting paths
10. $\mathbf{\Phi} \leftarrow -\frac{1}{\beta} \log \mathbf{Z}_{\mathrm{h}}$ ▷ matrix of (directed) free energies
11. $\mathbf{\Delta}^{\mathrm{RSP}} \leftarrow (\overline{\mathbf{C}} + \overline{\mathbf{C}}^\mathsf{T})/2$ ▷ symmetrization
12. $\mathbf{\Delta}^{\mathrm{FE}} \leftarrow (\mathbf{\Phi} + \mathbf{\Phi}^\mathsf{T})/2$ ▷ symmetrization
13. **return** $\mathbf{\Delta}^{\mathrm{RSP}}, \mathbf{\Delta}^{\mathrm{FE}}$

distance measure between graph nodes with nice properties. Interestingly, as can be seen from Equation (3.6), the FE distance is actually a metric resulting from the sum of two dissimilarities, namely the expected cost (i.e. the RSP dissimilarity, after symmetrization) and the relative entropy, neither of which satisfy the triangle inequality by themselves. We also note that the quantity $\log \mathcal{Z}_{st}$ already appeared in [87] as a potential inducing a drift for a random walker in a continuous-state extension of the RSP framework.

3.4 Related work on generalized graph distances

There have been a few other suggestions for graph distances that generalize the resistance or CT and the SP distances, which we will discuss in this section. Of course, the simplest interpolation between the two distances is their weighted average, which we will experiment with as a null model. In addition, Chebotarev has defined several parametrized graph distance measures [45, 43, 44, 46]. In this chapter, we focus on the logarithmic forest distances [45]. Alamgir and von Luxburg defined a generalized distance called the *p-resistance distance*

in order to tackle the problem of the resistance distance becoming meaningless with large graphs [3]. Indeed they show that with certain values of the parameter p, the p-resistance distance avoids this pitfall.

3.4.1 The SP-CT combination distance

The graph distance families presented and reviewed in this chapter all involve a sophisticated theoretical derivation. In the experiments we want to compare these distances also to a baseline model that generalizes the SP and CT distances in the most simple way, namely the weighted average of the two distances:

$$\Delta_{st}^{\mathrm{SP-CT}} = \lambda \Delta_{st}^{\mathrm{SP}} + (1-\lambda)\Delta_{st}^{\mathrm{CT}}, \tag{3.11}$$

where $\lambda \in [0,1]$.[2] We call it straightforwardly the SP-CT combination distance; it satisfies the triangle inequality because a convex combination of metrics is also a metric. Although the SP-CT combination does not contain as interesting details as the other distances, we can see that in some cases even using this simple choice can be competitive with the other parametrized distances.

3.4.2 Logarithmic forest distances

The logarithmic forest distance [45] has its foundation in the matrix-forest theorem and another family of distances developed earlier by Chebotarev called simply the forest distance [43, 44]. The definition of the logarithmic forest distance goes as follows. First, we define the Laplacian (or Kirchhoff) matrix of a graph G as $\mathbf{L} = \mathbf{D} - \mathbf{A}$, where $\mathbf{D} = \mathbf{Diag}(\mathbf{Ae})$. Then we consider the matrix

$$\mathbf{Q} = (\mathbf{I} + \alpha\mathbf{L})^{-1},$$

where $\alpha > 0$. The elements of this matrix measure the *relative forest accessibilities* [43] which can be considered as similarities between nodes of the graph after all its edge weights have been multiplied by the constant α. In fact, in [45] Chebotarev handles a more general case by considering arbitrary transformations of the edge weights and multigraphs instead of graphs. The definition proceeds by taking the elementwise logarithmic transformation

$$\mathbf{M} = \gamma(\alpha - 1)\log_\alpha \mathbf{Q},$$

[2] In the experiments, we actually use a linear combination of the SP distance and resistance distance. This is because the CT distance values tend to be very large compared to the SP distances, whereas resistance distance values are in the same magnitude as SP distances.

where $\gamma > 0$ is another parameter and the logarithm is taken elementwise in basis α. This expression provides another similarity measure. From it, the matrix of logarithmic forest distances is derived as

$$\boldsymbol{\Delta}^{\text{logFor}} = \frac{1}{2}(\mathbf{m}\mathbf{e}^{\mathsf{T}} + \mathbf{e}\mathbf{m}^{\mathsf{T}}) - \mathbf{M}, \tag{3.12}$$

where $\mathbf{m} = \mathbf{diag}(\mathbf{M})$. The last transition is a classical way of defining a matrix of distances from a matrix of similarities [29].

The definition provides a metric which also satisfies the graph-geodetic property (see Section 3.3). It involves two parameters γ and α. For any positive value of the parameter γ, the logarithmic forest distance becomes proportional to the CT and the SP distances as $\alpha \to 0^+$ and $\alpha \to \infty$, respectively[3]. In the special case of $\gamma = \log(e + \alpha^{2/n})$, Chebotarev shows that the logarithmic forest distance approaches exactly these two other distances. However, this form is not very practical, because even with moderate size graphs the exponent $2/n$ cancels out the effect of setting a large value to α. Thus, we decided simply to assign $\gamma = 1$ in our experiments.

3.4.3 *p*-resistance distance

Alamgir and von Luxburg [3] defined a generalization of the resistance distance, called the *p-resistance distance*. Like the resistance distance, the p-resistance distance considers the graph as an electrical resistance network, where the edges $(k, l) \in E$ of the network have resistances r_{kl} and a unit volt battery is attached to the target nodes whose distance is being measured, forming a unit flow I from s to t (see Section 2.4.2 for the definition of unit flows). Then for a constant $p > 0$, the p-resistance distance is defined as the minimized p-resistance (w.r.t. the unit flow) between s and t, formally as

$$\Delta_{st}^{p\text{Res}} = \min\left\{ \sum_{(i,j)\in E} r_{ij}|I_{ij}|^p \;\middle|\; I \text{ is a unit flow from } s \text{ to } t \right\}. \tag{3.13}$$

When the parameter $p = 2$, the above definition becomes the definition of effective resistance, i.e. the resistance distance and when $p = 1$ the distance coincides with the SP distance. Alamgir and Von Luxburg [3] show that there exists a value $1 < p < 2$ for which the p-resistance distance avoids the lost-in-space phenomenon with large graphs, introduced in Section 2.4.5.

[3] More accurately, the logarithmic forest distance converges to the unweighted shortest path distance (see Equation 2.25) but we nevertheless include it in our comparison.

Another, almost identical form of the p-resistance distance has been proposed by Herbster in [104]. Also, in a closely related work [137], the authors study network flow optimization in the same spirit as with the p-resistance. Their viewpoint is based on network routing problems and provides a spectrum of routing options that make a compromise between latency and energy dissipation in selecting routes in a network. They, however, do not explicitly define a graph node distance.

The p-resistance distance is theoretically sound, but it lacks a closed form expression for computing all the pairwise distances of a graph. Thus, the result can only be obtained by solving a minimization problem for each pair of nodes separately. This currently limits the method to be applicable only for small graphs, which is why we were able to include it only in the experiments with small artificial graphs of Sections 3.5.1-3.5.2, but not the others.

3.4.4 Other graph node distances

In the comparisons in Section 3.5, we focus on the above presented distance families that specifically find a compromise between the SP and CT distances. However, we would like to mention other interesting distances that have been defined for graphs. In [82], where the free energy distance was already presented as the potential distance, the authors also define another distance family, called the *surprisal distance*. Its definition shares the same background with the free energy, but it does not generalize the SP and CT distances. However, it is shown in [82] to provide good results in clustering and semisupervised learning.

In [210], the authors define the *amplified commute time distance* in order to correct the inconvenience of the commute time distance in large graphs. Of all the references above in Section 3.4 to the work of Chebotarev, we want to highlight the *walk distance* [46], which has some interesting, quite unintuitive properties, especially when considering the peripheral nodes of a network. Finally, we mention the *communicability distance* [66, 65], which measures how well a disturbance is carried over the network between two nodes when considering the network as a quantum harmonic oscillator network in a thermal bath. This model shares similarities with the RSP framework, including an inverse temperature parameter assigned to the network. Bavaud [18] investigated the spectrum of different Euclidean distances that can be defined on weighted graphs. He also discusses the notion of *focused* distances, i.e. distances which are zero for nodes that are adjacent to the same set of neighboring nodes. He also studies the distances by applying them in *thermodynamic clustering*, which, interestingly enough, is based on minimizing a free energy functional. There

also exists a vast literature on kernels on graphs, which can always be converted to matrices of squared Euclidean distances. For a thorough review on graph kernels, we advice the reader to see [78].

3.5 Experiments

In this section, we compare the different distance families presented in the previous sections, namely the RSP dissimilarity, the FE distance, the p-resistance distance, the logarithmic forest distance and the SP-CT combination distance. First, we consider small artificial graphs and study the behavior of the different distances with different parameter values. This is done by seeing how the relation between distances of different pairs of nodes changes as the parameter value is altered. As also notified by Chebotarev [46], the interest in comparing different distance measures does not lie in the pairwise distances themselves, but in the proportions between the pairwise distances. We then study the applicability of the distance families in a clustering experiment using networks constructed from the Newsgroups corpus of text documents. Finally, we compare the performances of the different distance families in a graph-based semisupervised learning experiment.

3.5.1 Visualization

In order to show that the different distance families are appealing to use, we present a graph visualization example using an artificially generated graph. The graph is generated with the benchmark algorithm of Lancichinetti, Fortunato and Radicchi (LFR) [125]. The algorithm generates graphs that contain a community structure and obey a power law distribution both for node degrees and community sizes. The user can also set a mixing parameter which essentially means the likelihood of inter-community edges. Moreover, it is possible to include overlapping nodes that belong to several communities. The particular graph used in the visualization was generated to consist of four communities of 80 nodes, with the mixing parameter set to 0.2, the power law exponent of the degree distribution set to -2, the average degree set to 8, the maximum degree to 100.

The graph is visualized using the set of distances between all pairs of nodes provided by the different distance families, namely the SP distance, the CT distance, the RSP distances and the other parametrized distances introduced in Sections 3.4.1-3.4.3. We present two visualizations for each distance family, one obtained with classical multidimensional scaling (CMDS) [29], and another

with t-distributed Stochastic Neighbor Embedding (t-SNE) [143]. CMDS is one of the most used and most studied linear dimensionality reduction methods and it minimizes the squared error between the inner products of the visualization vectors and the inner products in the embedding space defined by the distances, if those distances are squared Euclidean. t-SNE, on the other hand, is a state-of-the-art non-linear dimensionality reduction method that is based on preserving the neighborhood-relation of the original data space in the low-dimensional embedding. CMDS requires no parameters (except for the parameters related to the distance and the number of embedding dimensions), but t-SNE involves a *perplexity* parameter which is related to the neighborhood size considered by the method.

We utilize the known cluster labels of the nodes in order to evaluate the ability of the different distance measures in capturing the cluster structure of the graph in the visualizations. We first compute the visualization for different values of the parameter related to each parametrized distance, as well as for different values of the perplexity parameter for the t-SNE method. We then measure the quality of the visualizations according to a specially designed criterion, namely the sum of the ratios of the mean distance within a cluster (where the cluster is determined by the known cluster labels) and the mean distance to nodes outside the cluster. The smaller this ratio is, the better the embedding represents the cluster structure. Formally, for a clustering (i.e. partition) $\{\mathcal{C}_1, \ldots, \mathcal{C}_K\}$ of the node set V, the quality of the visualization is defined as

$$\sum_{k=1}^{K} \frac{\bar{\Delta}_{\mathcal{C}_k}}{\bar{\Delta}_{\mathcal{C}_k,(\Omega\setminus\mathcal{C}_k)}}, \tag{3.14}$$

where $\bar{\Delta}_{\mathcal{C}_k}$ is the mean distance within cluster $\mathcal{C}_k$ and $\bar{\Delta}_{\mathcal{C}_k,(\Omega\setminus\mathcal{C}_k)}$ is the mean distance from points inside cluster $\mathcal{C}_k$ to points outside the cluster, i.e., for distance measure Δ,

$$\bar{\Delta}_{\mathcal{C}_k} = \frac{1}{|\mathcal{C}_k|^2} \sum_{i,j\in\mathcal{C}_k} \Delta_{ij} \quad \text{and} \quad \bar{\Delta}_{\mathcal{C}_k,(\Omega\setminus\mathcal{C}_k)} = \frac{1}{|\mathcal{C}_k|(n-|\mathcal{C}_k|)} \sum_{i\in\mathcal{C}_k} \sum_{j\notin\mathcal{C}_k} \Delta_{ij}. \tag{3.15}$$

The intuition behind the criterion in Equation (3.14) is that a low value of the criterion requires tight clusters, that are at the same time dispersed from each other. Accordingly, we selected, for each distance measure, the parameter value that produced the embedding with lowest criterion value according to Equation (3.14). Moreover, for t-SNE we chose the combination of the distance parameter value and the perplexity value that minimized the criterion. Thus, for each distance measure, we present the visualization out of all computed visualizations that can be considered to show the cluster structure the best.

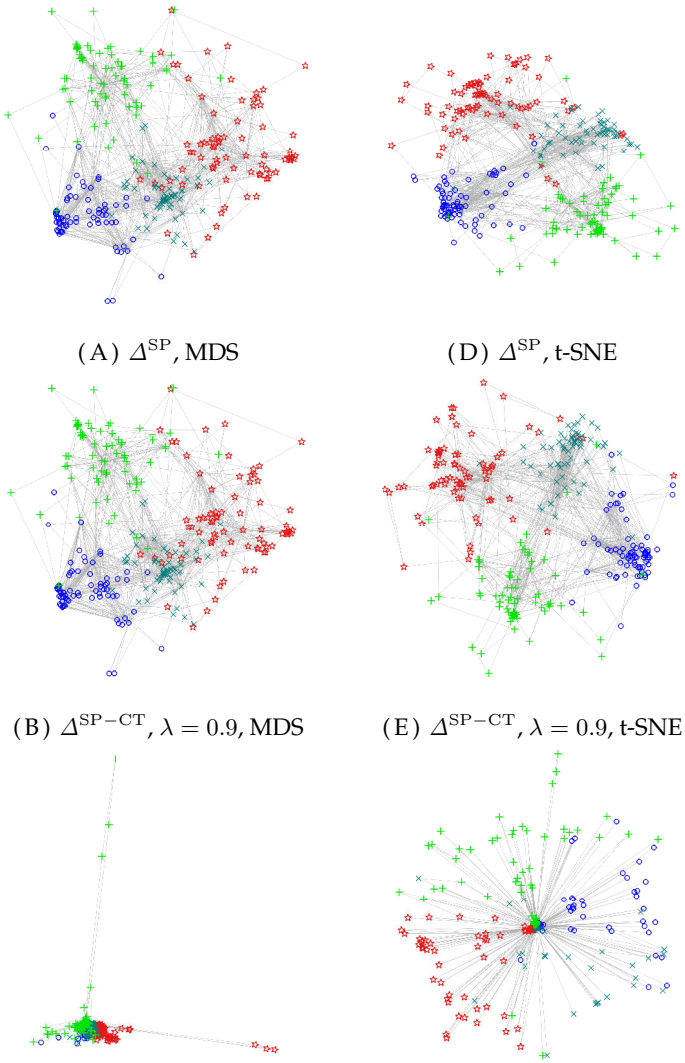
The visualizations, both with MDS and with t-SNE, are depicted in Figure 3.1 for Δ^{SP}, Δ^{CT} and $\Delta^{\mathrm{SP-CT}}$, in Figure 3.2 for Δ^{RSP}, Δ^{FE} and in Figure 3.3 for Δ^{logF} and $\Delta^{p\mathrm{Res}}$. There are two main notions that can be drawn from the plots. The first is that the parametrized distances, excluding the SP-CT combination distance, clearly manage to present the community structure better than the SP and CT distances. The plots obtained with the SP and SP-CT combination distances contain much more overlap between the communities than the other parametrized distances. The plot obtained with the CT distance also contains a lot of overlap between communities, but in addition to that, the plot is distorted by the distances of two nodes which get drawn far apart from the rest of the nodes. The degree of the two nodes is 3, which is the minimum degree of the graph. Thus, the distortion of the plot is in correspondence with the undesired dependence of the CT distance on the degrees of nodes, discussed in Section 2.4.5. In fact, these nodes have, among all the nodes, the two largest sums of CT distances to all other nodes of the graph.

The second main notion is that the plots obtained with the other parametrized distances, apart from the SP-CT distance, are quite similar to each other. There is a bit more overlap between the two upper communities in the plot obtained with the p-resistance distance compared to the others, but this difference is very small.

3.5.2 Comparisons with small graphs

In the first example, we use the simple graph depicted in Figure 3.4 consisting of a triangle, i.e. a 3-clique connected to an isolated node. We call it the extended triangle graph. We observe the proportions of distances between nodes 1 and 2 and nodes 2 and 3, i.e. the quantities Δ_{12}/Δ_{23} for all the different distance families. These results are also plotted in Figure 3.4 using 20 different parameter values for each family of distances. The parameter values are scaled in such a way that the relevant parameter range of each distance family becomes visible. In addition, the abscissa is logarithmic for all other parameters but linear for the λ of the SP-CT combination distance.

First of all we can observe that all curves converge to unity on the right hand end of the plot. This happens as all the distances converge to the shortest path distance and thus $\Delta_{12} = \Delta_{23} = 1$ for all distances. On the left end of the plot, all curves approach the value 1.5 which is the ratio of the CT distances between the nodes. In other words, for the CT distance $\Delta^{\mathrm{CT}}_{12} > \Delta^{\mathrm{CT}}_{23}$ holds which is caused by the fact that nodes 2 and 3 are, in a sense, better connected together (namely through node 4) than nodes 1 and 2.



(A) Δ^{SP}, MDS

(D) Δ^{SP}, t-SNE

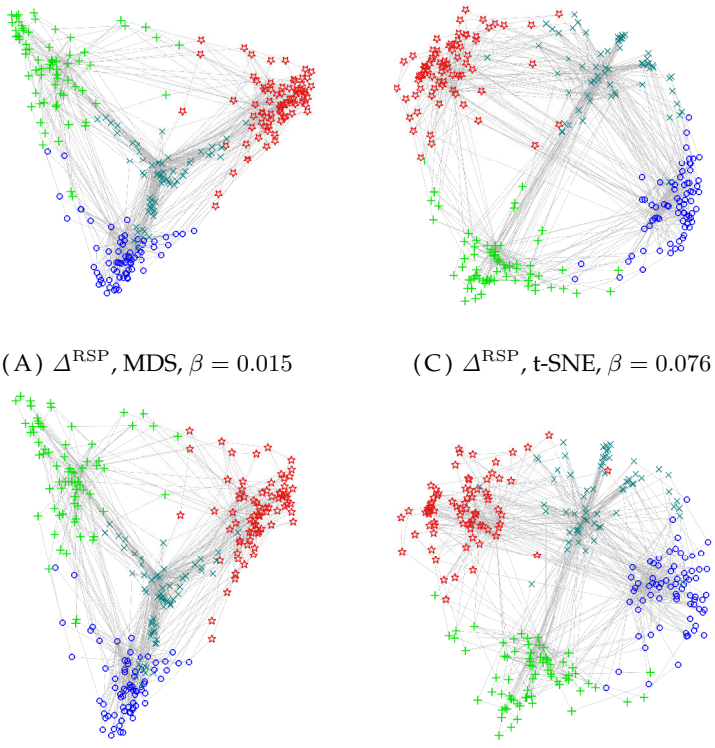
(B) $\Delta^{\mathrm{SP-CT}}$, $\lambda = 0.9$, MDS

(E) $\Delta^{\mathrm{SP-CT}}$, $\lambda = 0.9$, t-SNE

(C) Δ^{CT}, MDS

(F) Δ^{CT}, t-SNE

FIGURE 3.1: Visualizations of a graph, generated with the LFR algorithm consisting of four communities, with CMDS and t-SNE based on the shortest path distances (A and D), the SP-CT distances (B and E) and the commute time distances (C and F).



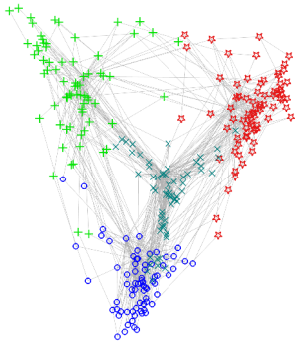
(A) Δ^{RSP}, MDS, $\beta = 0.015$

(C) Δ^{RSP}, t-SNE, $\beta = 0.076$

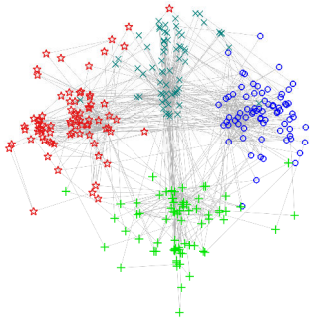
(B) Δ^{FE}, MDS, $\beta = 0.076$

(D) Δ^{FE}, t-SNE, $\beta = 0.039$

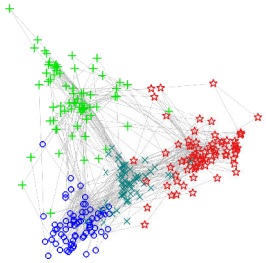
FIGURE 3.2: Visualizations of a graph, generated with the LFR algorithm consisting of four communities, with CMDS and t-SNE based on the RSP dissimilarities (A and C) and the free energy distances (B and D).



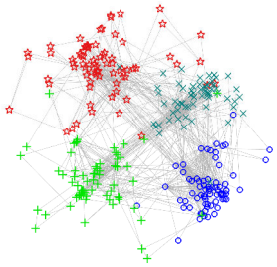
(A) Δ^{logFor}, MDS, $\alpha = 1.31$



(C) Δ^{logFor}, t-SNE, $\alpha = 0.088$



(B) $\Delta_{st}^{p\mathrm{Res}}$, MDS, $p = 1.30$



(D) $\Delta_{st}^{p\mathrm{Res}}$, t-SNE, $p = 1.07$

FIGURE 3.3: Visualizations of a graph, generated with the LFR algorithm consisting of four communities, with CMDS and t-SNE based on the logarithmic forest distances (A and C) and the p-resistance distances (B and D).

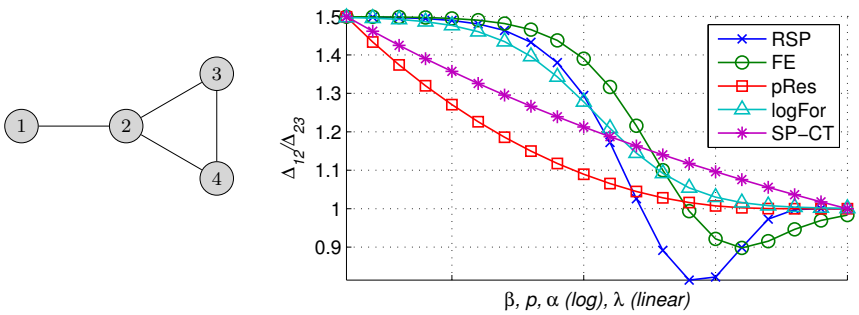


FIGURE 3.4: The extended triangle graph (left) and the ratio of distances Δ_{12}/Δ_{23} (right) with the RSP dissimilarity (RSP) and the FE distance (FE), both with $\beta \in [10^{-4}, 20]$, the p-resistance distance (pRes) with $p \in [1, 2]$ (reversed), the logarithmic forest distance (logFor) with $\alpha \in [10^{-2}, 500]$ (reversed) and the SP-CT combination distance (SP-CT) with $\lambda \in [0, 1]$.

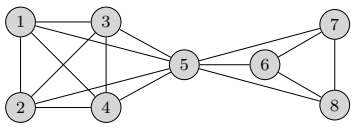


FIGURE 3.5: A graph with a 4-clique and a 3-clique and a hub node between them.

The real interest in Figure 3.4 lies in the transformation that takes place in the intermediate parameter values of the distance families. We can observe that the ratio Δ_{12}/Δ_{23} changes monotonously with respect to the parameter value change in three cases, with the p-resistance, the logarithmic forest distance and obviously the SP-CT combination. In other words, these three metrics always consider the distance between nodes 2 and 3 smaller than the distance between node 2 and the isolated node 1.

However, with the FE distance and the RSP dissimilarity, the ratio behaves non-monotonously. In other words, for a range of intermediate parameter values, these functions consider the distance between the isolated node 1 and the central node 2 to be smaller than the distance between nodes 2 and 3 (and between 2 and 4). Allowing this possibility could prove useful for a distance measure in applications. For example, in a social network, a relationship with an isolated person can in some situations and contexts be considered stronger than the relationship with a member of a group.

The phase transition that occurs with the FE distance and the RSP dissimilarity in our small example case can have implications in more practical situations as well. Obviously, it can affect nearest neighbor related methods but also clustering applications. Consider, for example, a larger scale situation, as the graph depicted in Figure 3.5. This graph consists of two cliques of sizes 4 and 3 which are connected through a hub node (node 5) that shares edges with all the other nodes of the graph. Consider then a clustering of the graph nodes into two clusters. The nodes in the two cliques obviously should belong to their own clusters. But which cluster should the hub node 5 be assigned to? This is generally a question of context and taste. In some cases there might be a preference for classifying the hub node to the smaller cluster, whereas in others it should be considered part of the larger cluster. One option would also be to put the hub node into its own cluster. However, here we are interested in cases where the number of clusters is fixed and a decision on the cluster assignment of the hub node has to be made.

In this specific case, the p-resistance distance, the logarithmic forest distance and the SP-CT combination distance always consider node 5 closer to the larger clique than the small one. Thus, for example, when performing a k-means based clustering with $k = 2$, using the mentioned distances will always result in assigning the hub node 5 into the larger cluster. However, the other three distances are more flexible. Namely, thanks to the phase transition seen in Figure 3.4, performing k-means with these distances can result in two different partitions depending on the parameter value. Worth pointing out is that since the shortest path distance between node 5 and all other nodes is 1, a k-means clustering can result in either of the two interesting partitions, because with both partitions the global minimum within-cluster inertia is achieved.

These examples illustrate that there are subtle differences between the generalized distance families. These differences may be useful for deciding which distance measure should be used in which case and they might give insight about how to select the parameter value of a generalized distance depending on the nature and properties of the data. In Sections 3.5.3 and 3.5.4, we test the different distance measures in clustering and semisupervised learning in order to observe the capabilities of the different distance families in detecting desired clusters in data. In the future we will extend this investigation to other problems such as link prediction [139, 141].

3.5.3 Graph node clustering

In order to see whether the distance families in fact achieve to extract a meaningful representation of a graph, we use them in a graph node clustering task

and evaluate the quality of the partitions found. For the clustering, we employ the kernel k-means algorithm introduced in [217]. It is based on an iteration procedure similar to k-means that searches for prototype vectors, but in a *sample space* and by using a kernel containing similarities between data points, instead of using explicit data vectors in a feature space. This is convenient in our case, because we only possess information of distances between data points instead of a vector representation. The dimension of the sample space is n, i.e. the number of data points, and each data point is represented in the sample space as the corresponding column vector of the kernel. Inner products between the data and prototype vectors in the sample space correspond to distances in an *embedding space* by application of the *kernel trick* [189]. The goal of the algorithm is to find prototype vectors in the sample space that minimize the within-cluster inertia in the embedding space, i.e. the sum of the squared distances of each data point to its corresponding prototype.

In order to use the kernel k-means algorithm we need to convert each matrix of distances into a matrix of similarities. We transform the matrices of distances $\mathbf{\Delta}$ into similarity matrices $\mathbf{K}$ in a classical way [29, 185, 93] as

$$\mathbf{K} = -\frac{1}{2}\mathbf{H}\mathbf{\Delta}\mathbf{H}, \tag{3.16}$$

where $\mathbf{H} = \mathbf{I} - \mathbf{e}\mathbf{e}^{\mathsf{T}}/n$ is the centering matrix. When the matrix $\mathbf{\Delta}$ contains squared Euclidean distances, matrix $\mathbf{K}$ will contain inner products of centered vectors in the same Euclidean space. We use the distances without raising them to the square, because the commute time distance already is the square of a Euclidean distance, called simply the Euclidean commute time distance [181]. However, for intermediate parameter values, the generalized distances are not necessarily Euclidean (or squared Euclidean) distances and thus the corresponding similarity matrices are not necessarily kernels in the traditional sense as they might not be positive definite. The positive definiteness could be ensured by forcing the negative eigenvalues of the similarity matrix to zeros. However, we have noticed in experiments that this does not affect much the results nor the convergence of the kernel k-means.

In addition to the similarity matrices derived from the generalized graph distance matrices through (3.16), we also use the *sigmoid commute time kernel* proposed by Yen et al. [217]. They construct the kernel by taking a sigmoid transformation of the elements of the commute time kernel which can be computed as the Moore-Penrose pseudoinverse $\mathbf{L}^+$ of the graph Laplacian. Thus, the similarities given by this method are $(\mathbf{K}_{\mathrm{SigCT}})_{st} = 1/(1 + \exp(-al^+_{st}/\sigma))$. The parameter a controls the smoothing of the similarity values caused by the sigmoid transformation and σ is the standard deviation of the elements l^+_{ij}. The

sigmoid commute time kernel has been shown to perform well in many machine learning tasks, especially in the kernel k-means method used in this chapter. We consider it as a baseline for the clustering performance comparisons.

We use a collection of text document networks extracted from the 20 Newsgroups data set[4]. A more detailed description of the collection that we use can be found in [217]. In short, our collection consists of ten different weighted undirected networks, where the nodes represent text documents and edges and their affinities are formed according to the co-occurrence of words within the documents. The affinities a_{ij} have been converted to costs c_{ij} as $c_{ij} = 1/a_{ij}$. Each network has been constructed by combining subsets of 200 documents from one topic. The networks consist of either two, three or five of such subsets of documents resulting in networks of 400, 600 and 1000 nodes. The goal of the clustering is then to detect the division of each network according to the topics. Unfortunately, we could not obtain results in this experiment with the p-resistance because of its high computational cost discussed already in Section 3.4.3 and the sizes of the networks in the experiment.

We compare the partitions found by the clusterings with the classification according to the topics by computing the *normalized mutual information* (NMI) [195] between a clustering partition X and the topic classification Y as

$$NMI(X,Y) = \frac{I(X,Y)}{\sqrt{H(X)H(Y)}},$$

where $I(X,Y)$ is the mutual information of partitions X and Y and H is entropy. In order to avoid the expensive running of the experiments every time for a wide range of parameter values, we perform a simple parameter tuning. We assume that the tuning results generalize from one data set of a particular kind (here, a text document collection) to another. We separate one of the ten networks in our collection as a tuning data network. We perform the kernel k-means clustering for this network with 20 different random initializations of the prototype vectors. Out of these 20 runs, we observe the NMI score of the clustering that achieves the smallest within-cluster inertia. This process is repeated another 20 times for a set of parameter values distributed either logarithmically (or linearly, in the case of the SP-CT-distance) on a given range of values. The parameter value producing the largest mean NMI score is chosen as the tuned value. The same procedure is done for all the distance measures as well as the sigmoid CT kernel.

[4] http://qwone.com/~jason/20Newsgroups/

Distance	Similarity matrix	Parameter range	Optimal value
RSP dissimilarity	$\mathbf{K}_{\mathrm{RSP}}$	$[10^{-4}, 20]$	$\beta = 0.02$
FE distance	$\mathbf{K}_{\mathrm{FE}}$	$[10^{-4}, 100]$	$\beta = 0.07$
Logarithmic forest distance	$\mathbf{K}_{\mathrm{logFor}}$	$[10^{-2}, 500]$	$\alpha = 0.95$
SP-CT combination	$\mathbf{K}_{\mathrm{SP-CT}}$	$[0, 1]$	$\lambda = 1$
Sigmoid commute time	$\mathbf{K}_{\mathrm{SigCT}}$	$[10^{-2}, 10^{3}]$	$a = 26$

TABLE 3.1: The notation of the similarity matrices and the optimal parameter values obtained on the tuning data set. 20 values were tested within the parameter range for each parameter, distributed linearly for $\mathbf{K}_{\mathrm{SP-CT}}$ and logarithmically for the other kernels.

Note that the above method for parameter tuning is not entirely realistic, nor practical for traditional clustering where class labels of the data are normally unknown. However, as the main goal of the experiment is to evaluate the ability of different distance measures in reflecting the cluster structure (as determined by the class labels), it can be considered fair for the comparison to choose the parameter value for each distance measure in such an optimal way. In [193], a comparison of graph distance measures (including some of the distances considered here) in clustering is performed similarly to the comparison presented here, but there the parameter tuning is determined by the clustering with largest modularity. Also in that setting, the RSP distances provide very promising results.

The ranges of parameter values and the optimal values for each method are reported in Table 3.1. Note that with the SP-CT combination distance, the optimal parameter value is $\lambda = 1$. This means that already for the 600 node tuning network the best clustering results with the SP-CT distance are obtained when focusing only on the SP distance and neglecting the CT distance.

To give an idea of the effect of considering the generalized distances instead of the standard shortest path, or random walk distances, Table 3.2 shows the Pearson correlation between each generalized distance, and both the shortest path distance, and the commute cost distance, when computed on the 9 Newsgroups data sets used for clustering evaluation. Each generalized distance matrix was computed using the tuned parameter value, reported in Table 3.1. From the table we see that the RSP distances have fairly high correlation with Δ^{SP}, the correlation with Δ^{CC} is actually surprisingly low, especially for the RSP dissimilarity, with all correlation coefficients less than 0.15. The result with the logarithmic forest distance is different, as it correlates moderately with both Δ^{SP} and Δ^{CC}. In other words, Δ^{logFor} seems more like a midway solution between Δ^{SP} and Δ^{CC}, whereas the RSP distances, especially Δ^{RSP}, are very different from the random walk distance Δ^{CC}. As the tuned parameter value for $\Delta^{\mathrm{SP-CT}}$ is $\lambda = 1$ the columns corresponding to it only serve to

	Correlation with Δ^{SP}				Correlation with Δ^{CC}			
Data set	Δ^{RSP}	Δ^{FE}	Δ^{logFor}	$\Delta^{\mathrm{SP-CT}}$	Δ^{RSP}	Δ^{FE}	Δ^{logFor}	$\Delta^{\mathrm{SP-CT}}$
G-2cl-1	0.7185	0.7888	0.6160	1.0000	0.1104	0.3062	0.6757	0.3097
G-2cl-2	0.7238	0.7900	0.6139	1.0000	0.1228	0.3719	0.7260	0.3176
G-2cl-3	0.7649	0.8157	0.6194	1.0000	0.1476	0.3534	0.7223	0.3492
G-3cl-1	0.7366	0.7944	0.5644	1.0000	0.0472	0.2665	0.7006	0.2419
G-3cl-2	0.6845	0.7571	0.5548	1.0000	0.0770	0.3188	0.7426	0.2736
G-3cl-3	0.6981	0.7903	0.6097	1.0000	0.0722	0.3109	0.7252	0.3350
G-5cl-1	0.6893	0.7615	0.5248	1.0000	0.0743	0.2963	0.7515	0.2579
G-5cl-2	0.6903	0.7631	0.5303	1.0000	0.0526	0.2921	0.7404	0.2711
G-5cl-3	0.6737	0.7436	0.5389	1.0000	0.0687	0.2786	0.6808	0.2569

TABLE 3.2: The Pearson correlation between the generalized distances (i.e. the RSP dissimilarities, the free energy distances, the logarithmic forest distances, and the SP-CT combination distances) and their limit distances (i.e. the shortest path distance and the commute cost distances). Each matrix of generalized distances was computed with the tuned parameter value, reported in Table 3.1.

show the correlation between the shortest path distances and random walk distances, which is quite low on these graphs.

We then use the tuned parameter values to perform the clustering on the nine remaining networks. As in the tuning phase, we again perform the clustering with 20 different initializations and choose the clustering that has the smallest within-cluster inertia. This is again done another 20 times and the mean and standard deviations of the NMI scores of these 20 best clusterings are collected. The results for each of the nine data sets are reported in Table 3.3. For each data set, we performed one-sided t-tests with significance level 0.05 to determine whether a mean NMI score with one method is significantly higher than with another, which is indicated in boldface.

From the results we see that the highest NMI scores are generally obtained with the similarity matrices based on the FE distance, the RSP dissimilarity and the sigmoid commute time kernel. We can see that the RSP dissimilarity and the FE distance provide scores very close to each other. The clusterings obtained based on the logarithmic forest distances do not correspond that well to the topic labeling except with the network G-2cl-B, for which all the distances, excluding the SP distance, give quite similar results. In general, all the distance families provide NMI scores that are quite close to each other, with the exception that with some networks, the score provided by the SP distance is clearly smaller than the others. Thus, the experiment indicates that the parametrized distances, when used with intermediate parameter values, do manage to capture more meaningful global information of the graph structure than the SP and CT distance do.

NMI Datasets	$\mathbf{K}_{\mathrm{RSP}}$	$\mathbf{K}_{\mathrm{FE}}$	$\mathbf{K}_{\mathrm{logFor}}$	$\mathbf{K}_{\mathrm{SP-CT}}$	$\mathbf{K}_{\mathrm{SigCT}}$
G-2cl-1	**84.5** $\pm$ 0.00	80.7 $\pm$ 1.09	83.1 $\pm$ 1.47	65.2 $\pm$ 0.59	81.6 $\pm$ 0.00
G-2cl-2	**58.7** $\pm$ 0.38	**58.7** $\pm$ 1.74	**58.8** $\pm$ 1.94	51.2 $\pm$ 0.46	56.8 $\pm$ 2.18
G-2cl-3	81.0 $\pm$ 0.00	81.1 $\pm$ 0.00	75.0 $\pm$ 1.13	**85.9** $\pm$ 0.00	79.6 $\pm$ 0.00
G-3cl-1	76.6 $\pm$ 0.00	76.2 $\pm$ 0.00	75.4 $\pm$ 0.72	74.2 $\pm$ 0.28	**77.3** $\pm$ 0.00
G-3cl-2	77.0 $\pm$ 0.00	**78.3** $\pm$ 0.83	75.5 $\pm$ 1.42	62.6 $\pm$ 0.51	73.0 $\pm$ 0.00
G-3cl-3	76.5 $\pm$ 0.28	**77.0** $\pm$ 0.50	74.4 $\pm$ 1.57	71.5 $\pm$ 0.50	75.9 $\pm$ 0.43
G-5cl-1	**69.6** $\pm$ 0.15	69.0 $\pm$ 0.66	60.4 $\pm$ 3.43	68.1 $\pm$ 0.43	66.8 $\pm$ 0.16
G-5cl-2	64.0 $\pm$ 0.42	**64.6** $\pm$ 0.34	58.7 $\pm$ 3.49	59.6 $\pm$ 0.59	60.4 $\pm$ 1.36
G-5cl-3	**61.2** $\pm$ 0.71	**61.6** $\pm$ 0.87	57.3 $\pm$ 2.77	47.8 $\pm$ 0.92	57.3 $\pm$ 0.46

TABLE 3.3: Clustering performances (Normalized Mutual Information, multiplied by 100) for each similarity matrix on the nine Newsgroup subsets

We also took a more detailed look at the individual cluster assignments of nodes to compare the partitions obtained with the different similarity matrices. We can only describe the results because a thorough presentation would require too much space. First of all, there are nodes in almost all of the networks whose cluster assignment never corresponds to the topical classification with any of the similarity matrices. This only indicates that the data is noisy and that the topical classification cannot be perfectly inferred from the network structure. Also, all the similarity matrices produce a clustering of the network G-5cl-2 where the first two classes are clustered mostly together and one of the topical classes is divided into two smaller clusters. This indicates a hierarchical structure of the classes and it seems to be interpreted by the different similarity matrices in different ways.

The similarity of the NMI scores between the RSP dissimilarity and the FE distance is explained by looking at the respective cluster assignments, which indeed are almost exactly similar with all the networks. However, the difference between the partitions based on these two distances and the others is evident from the individual assignments. Interestingly, especially with the larger networks, the clusterings based on the logarithmic forest distance and the sigmoid commute time kernel share some similarities. These similarities are most evident in the clusterings of the network G-5cl-2, mentioned above. Also, with this network, the clustering produced by the SP distance is in fact quite close to the ones produced by the RSP dissimilarity and the FE distance. All in all, the individual cluster assignments indicate many small differences in the clusterings with different similarity matrices. However, in order to get a stronger intuition of the reasons causing the differences, we would need an even more thorough investigation which we leave for future work.

3.5.4 Semisupervised 1-nearest-neighbor classification

The kernel k-means algorithm used in the clustering experiment in the previous section considers the distances of the graph globally as it aims to minimize the within-cluster inertia. We also wanted to employ the different distance families in a semisupervised learning algorithm based on nearest neighbors, i.e. only on local distances. In order to observe differences between the distance families, we use the propagating 1-nearest-neighbor algorithm [219], where, iteratively, the unlabeled node that is the closest to any labeled node gets assigned the label of the labeled node. This is obviously not a state-of-the-art method, but it provides a comparison of the distance families when only the most local distances are being applied.

We conduct the semisupervised learning experiments for the same Newsgroups datasets as in the clustering experiments in Section 3.5.3 but also for another collection of networks, the WebKB data set [144]. It is a collection of four co-citation networks collected from the websites of four American universities. The nodes in the networks are webpages and the edges of the network represent co-citations, meaning that two pages are connected by an edge, if they contain a link to the same target page. We perform the propagating 1-NN algorithm using the RSP dissimilarity, the FE distance, the logarithmic forest distance and the SP-CT combination distance. Again, as was the case with the the clustering experiments in Section 3.5.3, we cannot use the p-resistance in the comparison, because of its slow computation.

We perform the learning task with five different labeling rates, 10%, 30%, 50%, 70% and 90%. For each labeling rate, we average the score over 20 different randomized label deletions and for each label deletion the score is given by the mean classification rate according to a 10-fold cross validation. Within each fold of the cross-validation, we run an inner 10-fold cross validation to define the optimal parameter values for each distance family. Thus, the parameter tuning is performed in a very different manner compared to the clustering experiments, which makes more sense in the semisupervised setting.

Figure 3.6 shows the results of the propagating 1-NN algorithm with the nine Newsgroups data sets and four WebKB data sets. With some of the graphs, the results obtained with different distance families seem very similar to each other. However, where there is a noticeable difference, in fact the results obtained using the logarithmic forest distance seem quite good. On the other hand, the results obtained with the RSP dissimilarity are in many cases worse than the others. In other words, the results seem a bit contrary to the results obtained in the clustering experiments.

Labeling rate	10%		30%		50 %		70 %		90 %		Overall	
Sim. matrix	Rank	Score	Rank	Score	Rank	Score	Rank	Score	Rank	Score	Rank	Score
$\mathbf{K}_{\mathrm{RSP}}$	4	-14	4	-16	4	-16	4	-17	3	-14	4	-77
$\mathbf{K}_{\mathrm{FE}}$	3	-5	2	2	2	9	2	12	2	11	2	29
$\mathbf{K}_{\mathrm{logFor}}$	1	11	1	17	1	18	1	20	1	21	1	87
$\mathbf{K}_{\mathrm{SP-CT}}$	2	8	3	-3	3	-11	3	-15	4	-18	3	-39

TABLE 3.4: The ranking of the distance families according to Copeland's method based on the results in the propagating 1-NN learning task. The results are presented for each labeling rate separately across all data sets ($10\%, \ldots, 90\%$), and for all labeling rates together (Overall).

We perform a ranking of the distance families using Copeland's voting method [179], by computing the times that one distance family provides a significantly better result than another one with a 0.05 significance level. Copeland's method simply gives a score of $+1$ to a distance family that achieves a significantly superior score than another one on a given data set, and correspondingly a score of -1 to the other one. If there is no significant difference between two methods, they both are assigned a score 0. The ranking of the methods is then computed by summing the scores over all pairwise comparisons of methods and over all data sets.

We compute the ranking separately for each labeling rate by summing the scores over all data sets. Thus, for one labeling rate, the maximum score that a distance family can get is 39 (13 data sets and comparisons with 3 other distances). The ranks and scores obtained with Copeland's method are shown in Table 3.4. They confirm the observation that the logaritmic forest distance provides good results whereas using the RSP dissimilarity results in more misclassifications. The FE distance seems to perform quite well also, being ranked second, and the SP-CT combination distance is ranked third.

The reason why the RSP dissimilarity performs so poorly in this task is a bit surprising, at least considering its suitability for clustering which was showed earlier. This result may have something to do with the fact that in some cases the RSP dissimilarity does not satisfy the triangle inequality. The triangle inequality ensures an intuitive relationship between local and global distances, literally that a sum of local distances should be larger than the corresponding global distance. Another way to interpret this is that when the inequality is not satisfied, the local distances can be smaller than a metric global topology would allow. The 1-NN algorithm is based on purely local properties of the space, whereas the kernel k-means algorithm considers the distances more globally, by depending on the within-cluster inertia. However, this is only a suggestion for explaining the results and we have not been able to verify it theoretically, nor by demonstration.

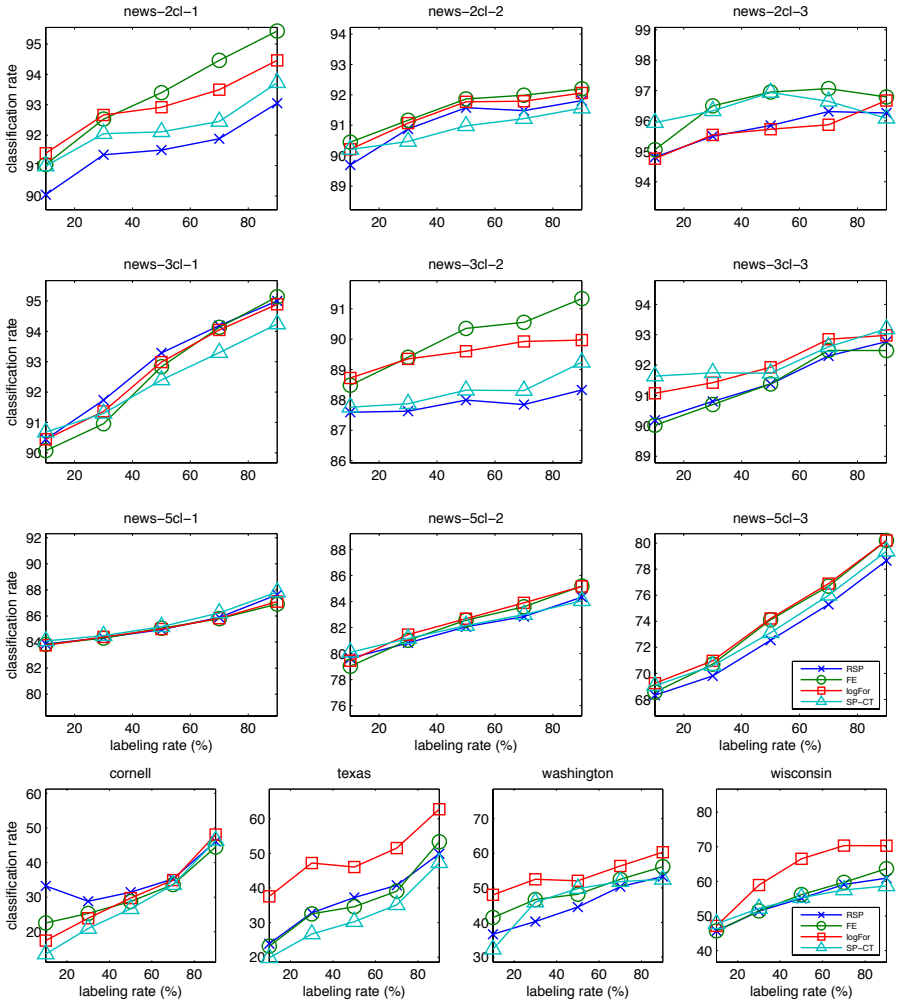


FIGURE 3.6: The results of the propagating 1-NN algorithm with the nine Newsgroups graphs (top rows) and the four WebKB co-citation graphs (bottom row)

3.6 Discussion

In this chapter, we concentrated on graph node distances that generalize the SP and CT distances. We first presented the setting of the chapter where we concentrate on graphs that have a structure based on edge weights and another

based on edge costs. Within this setting, we also proved that the commute cost distance is equal to the commute time distance up to a constant factor. This small result is interesting because it means that changing a cost of one edge of the graph has the same effect for all the pairwise commute cost distances in the graph, remembering that the transition probabilities defining the corresponding Markov chain are independent of the costs.

We then developed the theory behind one parametrized distance family, the RSP dissimilarity, by providing a new closed form algorithm for computing all pairwise dissimilarities of a graph. In addition, we derived the FE distance based on the Helmholtz free energy. Although we show that the FE distance coincides with the potential distance, proposed earlier, our new derivation provides a solid theoretical background to the distance. The derivation also reveals a closer resemblance of the FE distance with the RSP dissimilarity. The significant difference between the two is that only the FE distance is a metric. The FE distance has other nice features as well, including the graph-geodetic property and a closed form solution for computing all pairwise distances.

The other focus of this chapter was to compare different generalized graph node distances. We gave simple examples of subtle differences between some of the distance families using small artificial graphs. We then employed the distances on different network data analysis tasks in order to compare them and to evaluate their applicability. Graph visualizations obtained with the generalized distances indicated that they manage to respect the community structure of the graph much better than the traditional distances. Clusterings based on the parametrized distances with the tuned parameter values corresponded more to the inherent topic-induced classification of the graph nodes than the clustering found based on the SP and CT distances. We also performed semi-supervised graph node classification based on a simple nearest neighbor learning algorithm. In this experiment, the logarithmic forest distance performed best, while the RSP dissimilarity in many cases gave surprisingly poor results. Thus, it seems that the RSP dissimilarity can capture the global cluster structure of a graph quite well, but when focusing on individual, local distances, it is not so reliable. The logarithmic forest distance works well in the nearest neighbor learning task, but fails in many cases in the clustering task. The FE distance, however, performs well in both tasks. The FE distance is also defined in this chapter on a solid foundation based on statistical thermodynamics, and it satisfies many desirable properties of a distance. One future plan is to use the different distance families in other network analysis tasks in order to characterize their differences more and give more insight on which distance is appropriate in which context.

Chapter 4

Community detection using bag-of-paths probabilities

4.1 Introduction

In this chapter, we will use path probabilities for performing community detection. Community detection means the task of partitioning the nodes of a graph into dense groups, i.e. groups containing many edges within them but that only share a few edges with each other [74]. It is analogous to clustering in a vector space, but with the difference that in clustering, the number of clusters is often set beforehand, whereas in community detection, finding a natural number of communities is part of the task. Recent surveys of graph node clustering and community detection can be found in [74, 126, 184, 217, 76, 41]. In this chapter, we will only focus on community detection.

The most widely used criterion for designing community detection methods is the *modularity criterion* defined by Girvan and Newman [160]. It compares the amount of edges that appear within a community with the amount of them that would appear in a corresponding random graph. Several algorithms have been suggested for finding a partition with a maximal modularity – a task that is known to be computationally intractable [37]. One of the most popular methods for this is called the *Louvain method* [23]. In this chapter we will present a reinterpretation of the modularity criterion by focusing on *paths* instead of *edges*. Particularly, we replace edge probabilities with path probabilities according to the *bag-of-paths* framework [82, 129], and derive the *bag-of-paths modularity*. Moreover, we rephrase the Louvain method to maximize this new path modularity and show with a small set of networks that it outperforms the standard modularity as a community criterion on these data.

The chapter was originally a part of the article [81, 82], but was removed from it because of length constraints and to make the article more focused. The work presented here was since left unpublished, although the results obtained in the reported experiment were very promising. The idea of the bag-of-paths modularity developed for the mentioned article was also considered in the context of semisupervised classification in [54], where a bag-of-paths modularity kernel outperformed state-of-the-art methods.

4.1.1 Background

In order to estimate the quality of a partition, Newman and Girvan [160] introduced a new criterion called *modularity*. They used a method for detecting communities that selected edges for deletion according to the value of edge centrality, with respect to different edge betweenness measures. It then used the modularity criterion to select the best partition out of the hierarchy of edge deletions.

The modularity criterion has two advantages. It has a clear interpretation and it is able to detect a natural number of communities. It is based on the difference between the observed proportions of links within communities and the same quantity assuming independence of the community structure. The original definition of modularity [160] only deals with unweighted, undirected, graphs but it was soon generalized also for weighted graphs [156]. This measure became very popular and has been studied in an important number of publications [159, 57, 60, 75, 98, 199]. Nevertheless, other criteria allowing to find the natural number of communities or clusters exist in the literature [149]. Several optimization algorithms have since been applied to maximize modularity such as simulated annealing [60] and deterministic annealing [176]. Other references can be found in [74, 154, 76].

Different methods for greedy optimization of modularity were reviewed and compared in [162]. The paper defines four different features in which the approaches differ. The first greedy algorithm for maximizing modularity was suggested by Newman [159]. It is an agglomerative hierarchical clustering method, where groups of vertices are successively joined to form larger communities such that modularity increases after merging. A different greedy approach has been introduced by Blondel et al. [23]. Their method, often cited as the "Louvain method", will be investigated later in Section 4.2.2.

Arenas et al. [8] generalized Newman-Girvan modularity by replacing edges with motifs. Motif extraction on graphs has attracted a lot of attention in the past years in community detection but also in graph comparison [106]. A motif

is defined as a predefined connected subgraph, or a weakly connected structure. Newman and Girvan derived new versions of modularity based on short paths, cycles or any motifs of a network to define and identify communities.

The community detection method presented in this chapter relies on paths as motifs for measuring the connectedness of nodes. In the experiments of Section 4.3, we combine the motif-based approach with the Louvain method mentioned above, and obtain promising results in community detection.

4.1.2 Louvain method

The aim of this chapter is to extend the modularity criterion to consider longer, path-based dependencies in a graph, in order to improve and refresh the current community detection methodology. In order to evaluate the functioning of our approach, we use the Louvain method [23] for community detection. As already mentioned, it is a heuristic optimization method designed for maximizing modularity in the traditional sense but it can be easily extended for our purposes. The Louvain method proceeds by iterating two phases. First vertices are moved from one community to another, whenever it increases the modularity. After the modularity cannot be increased by moving vertices, the communities are merged as vertices of a new graph. The weight of a link in the new graph is determined as the sum of the weights of all the links between the vertices belonging to the corresponding two communities. The iteration of these phases is continued until no further increase in modularity can be obtained by merging communities.

According to the terminology of [162] the Louvain method is a combination of a simple fast greedy refinement algorithm and a multi-level coarsening algorithm. The experiments in [162] support the result of the developers of the Louvain method [23] that the algorithm is very fast and performs reasonably well in maximizing modularity. This is why we found it suitable for testing the effectiveness of the path-based modularity in community detection.

4.1.3 Modularity interpreted through a *bag-of-links* model

One of the major issues in network science has been the search for a rigorous and thorough definition of community in graphs. One natural approach is to consider communities as sets of nodes that are more connected together than the nodes of a random graph with a similar degree distribution. The *modularity criterion* [160, 154] precisely measures this departure from independence. It can be formulated by considering the probabilities that a link of the graph has

ending nodes that belong to the same class, i.e. the probabilities $\mathrm{P}(s \in \mathcal{C}_k, e \in \mathcal{C}_k)$, where s and e are the random variables corresponding to the starting and ending node of a link. This can be considered as a *bag-of-links model*, analogous to the bag-of-words model used in natural language processing. Then, the modularity of a partition $\{\mathcal{C}_1, \ldots, \mathcal{C}_m\}$ of the nodes of a graph G is defined as

$$Q(\mathcal{C}_1, \ldots, \mathcal{C}_m) = \sum_{k=1}^{m} (\mathrm{P}(s \in \mathcal{C}_k, e \in \mathcal{C}_k) - \mathrm{P}(s \in \mathcal{C}_k)\mathrm{P}(e \in \mathcal{C}_k)) \tag{4.1}$$

where $s \in \mathcal{C}_k$ ($e \in \mathcal{C}_k$) means that the starting (ending) node belongs to the kth community when sampling a link from the bag of links. A large value for modularity means a strong community structure.

Let us see how to estimate the modularity defined in Equation (4.1) on a particular graph G. Let $w(\mathcal{C}_k, \mathcal{C}_l) = \sum_{i \in \mathcal{C}_k} \sum_{j \in \mathcal{C}_l} a_{ij}$ denote the total weight of links in G starting from a node belonging to community k and ending in a node belonging to community l. Similarly, $w(\mathcal{C}_k, \mathcal{V}) = \sum_{i \in \mathcal{C}_k} \sum_{j=1}^{n} a_{ij} = \sum_{i \in \mathcal{C}_k} a_{i\bullet}$, where $\mathcal{V}$ is the set of nodes of G, denotes the total weight of links in G starting from a node belonging to community k, and $w(\mathcal{V}, \mathcal{C}_l) = \sum_{i=1}^{n} \sum_{j \in \mathcal{C}_l} a_{ij} = \sum_{j \in \mathcal{C}_l} a_{\bullet j}$ denotes the total weight of links in G ending in a node belonging to community l. The probability of picking a link connecting community k and community l can therefore be estimated by

$$\mathrm{P}(s \in \mathcal{C}_k, e \in \mathcal{C}_l) = \frac{w(\mathcal{C}_k, \mathcal{C}_l)}{\sum_{k',l'=1}^{m} w(\mathcal{C}_{k'}, \mathcal{C}_{l'})} = \frac{w(\mathcal{C}_k, \mathcal{C}_l)}{w(\mathcal{V}, \mathcal{V})} = \frac{\sum_{i \in \mathcal{C}_k} \sum_{j \in \mathcal{C}_l} a_{ij}}{a_{\bullet\bullet}} \tag{4.2}$$

where $w(\mathcal{V}, \mathcal{V}) = \sum_{i,j=1}^{n} a_{ij} = a_{\bullet\bullet}$. Recalling that $a_{i\bullet} = \sum_{j=1}^{n} a_{ij}$, the estimates of the corresponding marginal probabilities are

$$\begin{cases} \mathrm{P}(s \in \mathcal{C}_k) = \sum_{l=1}^{m} \mathrm{P}(s \in \mathcal{C}_k, e \in \mathcal{C}_l) = \dfrac{w(\mathcal{C}_k, \mathcal{V})}{w(\mathcal{V}, \mathcal{V})} = \left(\sum_{i \in \mathcal{C}_k} a_{i\bullet} \right) / a_{\bullet\bullet} \\ \mathrm{P}(e \in \mathcal{C}_l) = \sum_{k=1}^{m} \mathrm{P}(s \in \mathcal{C}_k, e \in \mathcal{C}_l) = \dfrac{w(\mathcal{V}, \mathcal{C}_l)}{w(\mathcal{V}, \mathcal{V})} = \left(\sum_{j \in \mathcal{C}_l} a_{\bullet j} \right) / a_{\bullet\bullet} \end{cases} \tag{4.3}$$

Therefore, the modularity of a partition of graph G can be computed through

$$Q = Q(\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_m) = \sum_{k=1}^{m} \left[\frac{w(\mathcal{C}_k, \mathcal{C}_k)}{w(\mathcal{V}, \mathcal{V})} - \frac{w(\mathcal{C}_k, \mathcal{V})\, w(\mathcal{V}, \mathcal{C}_k)}{(w(\mathcal{V}, \mathcal{V}))^2} \right] \tag{4.4}$$

or, elementwise,

$$Q = \sum_{k=1}^{m} \left[\frac{\sum_{i \in \mathcal{C}_k} \sum_{j \in \mathcal{C}_k} a_{ij}}{a_{\bullet\bullet}} - \frac{\left(\sum_{i \in \mathcal{C}_k} a_{i\bullet} \right) \left(\sum_{j \in \mathcal{C}_k} a_{\bullet j} \right)}{(a_{\bullet\bullet})^2} \right] \tag{4.5}$$

There is a balance between the two terms in Equation (4.5). To maximize the first term, the communities should contain many edges whereas the minimization of the second term is achieved by splitting G into many communities with small degrees each [38].

In order to ease computation and notation we introduce a $n \times 1$ binary membership vector $\mathbf{u}_k$ for each community k ($[\mathbf{u}_k]_i = 1$ if node i belongs to community k; otherwise $[\mathbf{u}_k]_i = 0$) and observe that $a_{\bullet\bullet}$ is the volume of G, $\mathrm{vol}(G) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij}$. We also define the *modularity matrix* as

$$\mathbf{Q} = \mathbf{A} - \frac{(\mathbf{A}\mathbf{e})(\mathbf{e}^{\mathrm{T}}\mathbf{A})}{\mathrm{vol}(G)}. \tag{4.6}$$

Then, Equation (4.5) can be rewritten in matrix form as

$$Q(\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m) = \frac{1}{\mathrm{vol}(G)} \sum_{k=1}^{m} \mathbf{u}_k^{\mathrm{T}} \mathbf{Q} \mathbf{u}_k \tag{4.7}$$

The goal in community detection is then to find a partition that maximizes the modularity Q with respect to the number of communities m as well as the binary membership vectors $\mathbf{u}_k$.

4.2 The bag-of-paths model

The modularity of a partition was defined in Equation (4.1) according to a bag-of-links framework. Our idea is to modify the definition by taking into account longer dependencies of nodes of the graph than just adjacencies. In order to achieve this, we replace the probabilities of links with probabilities of *paths* by using the *bag-of-paths (BoP)* framework [82]. It is a model inspired by the RSP framework, where a Gibbs-Boltzmann distribution is defined over the set of *all* paths on the graph. The distribution can be considered as prescribing probabilities of picking up a given path from a bag containing all the paths of

a graph so that short paths have a high probability and long paths have a low probability.

4.2.1 The bag-of-paths probabilities

In more detail, the BoP framework is defined by solving the problem of minimization of expected cost of paths drawn from the bag of paths (as described above), with a fixed relative entropy with respect to a reference probability distribution, as in the definition of the RSP framework. Formally, we seek for the distribution that solves

$$\underset{\mathrm{P}}{\text{Minimize}} \sum_{\wp\in\mathcal{P}} \mathrm{P}(\wp)\tilde{c}(\wp) \quad \text{subject to} \quad \begin{cases} \mathbb{J}(\mathrm{P}\|\mathrm{P}^{\mathrm{rw}}) = J_0 \\ \sum_{\wp\in\mathcal{P}} \mathrm{P}(\wp) = 1, \end{cases} \tag{4.8}$$

For the set $\mathcal{P}$ of all hitting paths on G, the reference path probability of a path $\wp \in \mathcal{P}$, whose starting and ending nodes are s and t is defined by normalizing the natural random walk probabilities by the number of possible s-t-pairs, as $\mathrm{P}^{\mathrm{rw}}(\wp) = \frac{1}{n(n-1)}\mathrm{P}^{\mathrm{rw}}_{st}(\wp)$. The solution of the BoP minimization is, again, the Gibbs-Boltzmann distribution

$$\mathrm{P}^{\mathrm{BoP}}(\wp) = \frac{\widetilde{w}(\wp)}{\mathcal{Z}}, \tag{4.9}$$

where $\mathcal{Z} = \sum_{\wp\in\mathcal{P}} \widetilde{w}(\wp)$ is the *bag-of-paths partition function*, and $\widetilde{w}(\wp)$ is the likelihood of path $\wp$, defined, as before, according to Equation (2.13), but by replacing the random walk probability $\mathrm{P}^{\mathrm{rw}}_{st}(\wp)$ related to the s-t-pair in question with the global random walk probability $\mathrm{P}^{\mathrm{rw}}(\wp)$.

Based on this, for a pair of nodes i and j, the *BoP probability*, Π_{ij}, is defined as the probability that a path picked from the bag of paths has i and j as its starting and ending node, respectively. It is given by

$$\begin{aligned} \Pi_{ij} = \mathrm{P}(\wp \in \mathcal{P}_{ij}) &= \frac{\sum_{\wp\in\mathcal{P}_{ij}} \mathrm{P}^{\mathrm{ref}}(\wp)\exp[-\beta\widetilde{c}(\wp)]}{\sum_{\wp'\in\mathcal{P}} \mathrm{P}^{\mathrm{ref}}(\wp')\exp[-\beta\widetilde{c}(\wp')]} \\ &= \frac{\sum_{\wp\in\mathcal{P}_{ij}} \mathrm{P}^{\mathrm{rw}}_{st}(\wp)\exp[-\beta\widetilde{c}(\wp)]}{\sum_{i',j'\in V}\sum_{\wp'\in\mathcal{P}_{i'j'}} \mathrm{P}^{\mathrm{rw}}_{st}(\wp')\exp[-\beta\widetilde{c}(\wp')]} = \frac{\mathcal{Z}_{ij}}{\sum_{i',j'\in V}\mathcal{Z}_{i'j'}}. \end{aligned} \tag{4.10}$$

Based on similar derivations as in chapter 3, the matrix of BoP probabilities between all pairs of nodes can be computed as (for more details, see [81, 82])

$$\mathbf{\Pi} = \frac{\mathbf{Z}\mathbf{D}_{\mathrm{h}}^{-1} - \mathbf{I}}{\mathbf{e}^{\mathrm{T}}(\mathbf{Z}\mathbf{D}_{\mathrm{h}}^{-1} - \mathbf{I})\mathbf{e}}, \text{ with } \mathbf{Z} = (\mathbf{I} - \mathbf{W})^{-1} \text{ and } \mathbf{D}_{\mathrm{h}} = \mathbf{Diag}(\mathbf{Z}).$$

4.2.2 Modularity with paths and its maximization

Let us now redefine the modularity criterion by replacing the link probabilities in the bag-of-links interpretation of Equation (4.1) with the BoP probabilities. For this, let us consider, again, a partition of the nodes of a graph $\{\mathcal{C}_1, \ldots, \mathcal{C}_m\}$. Let us then denote with $\mathrm{P}(s \in \mathcal{C}_k, e \in \mathcal{C}_l)$ the probability that the starting and ending node of a path drawn from the bag of paths belong to groups $\mathcal{C}_k$ and $\mathcal{C}_l$, respectively. Then, the *bag-of-paths modularity* (BoP modularity) of the partition is

$$Q_{\mathrm{BoP}}(\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_m) = \sum_{k=1}^{m} \mathrm{P}(s \in \mathcal{C}_k, e \in \mathcal{C}_k) - \mathrm{P}(s \in \mathcal{C}_k)\mathrm{P}(e \in \mathcal{C}_k) \qquad (4.11)$$

which extends modularity by taking all path lengths into account. In matrix form, if we first define the BoP modularity matrix as

$$\mathbf{Q}_{\mathrm{BoP}} = \mathbf{\Pi} - (\mathbf{\Pi}\mathbf{e})\left(\mathbf{e}^{\mathrm{T}}\mathbf{\Pi}\right) \qquad (4.12)$$

then, using the membership vectors $\mathbf{u}_i$, the BoP modularity can be computed as

$$Q_{\mathrm{BoP}}(\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m) = \sum_{k=1}^{m} \mathbf{u}_k^{\mathrm{T}} \mathbf{Q}_{\mathrm{BoP}} \mathbf{u}_k \qquad (4.13)$$

4.3 Experiments

In these experiments, we wanted to examine how the bag-of-paths framework could be applied in community detection. For this we use the Louvain method for modularity maximization but by interpreting the modularity using the BoP probabilities. The process is the same as in the standard Louvain method, briefly explained above in Section 4.2.2. Indeed, one only needs to give the matrix of path probabilities as input to the algorithm instead of the adjacency matrix of the graph, as is done in the standard case.

It is worth noting that the Louvain method often only finds a local maximum of the modularity criterion. Moreover, the end result depends on the order in which the nodes are browsed through, when moving them to other communities. Because of this, when evaluating the method, we must repeat the method several times with different random initial conditions. Another important point to remember is that the number of communities is not fixed for the Louvain method, but is determined by the method itself. For evaluating the quality of the communities with respect to the original classes, we use the adjusted Rand index [174, 218] and the normalized mutual information [195].

The experimental setting is similar to the clustering experiment with kernel k-means in chapter 3, section 3.5.3: the same Newsgroups data sets are used, and the same separate data set is left for tuning the β parameter for the BoP model. Again, for the tuning, the algorithm is run 30 times and the partition with the largest BoP modularity is picked. The adjusted Rand index and normalized mutual information score are then calculated for this partition. This process is repeated 20 times for different β's, this time ranging between $\beta \in [10^{-3}, 1]$. We observed that the best mean values for the evaluation measures were obtained with $\beta = 0.007$.

The Louvain method (LM; which tries to maximize modularity), and our novel bag-of-paths extension of it (LM-BoP; which tries to maximize the BoP modularity), are then applied on the nine different Newsgroups datasets described in chapter 3. We run the algorithm 30 times (trials) and choose the partition with the largest standard modularity or BoP modularity, depending on the algorithm, and calculate the adjusted Rand index and normalized mutual information for that partition. The mean and standard deviation of the results of 20 such runs are computed.

The results are reported in Table 4.1. They show quite consistently that maximizing the BoP modularity instead of the standard one improves the community detection performance according to the evaluation measures used on these datasets. With only two out of the nine datasets, "G-3cl-A" and "G-5cl-C", the path-based modularity does not provide better results, although even in those cases the results are competitive with the traditional modularity.

4.4 Discussion

This chapter introduced briefly a community detection method derived from the bag-of-paths framework. The method reinterprets modularity, which normally measures the quality of a partition of a graph into communities based on

ARI Datasets	LM	LM-BoP
G-2cl-A	49.9 ± 2.72	**69.9** ± 0.87
G-2cl-B	27.2 ± 2.28	**50.5** ± 3.22
G-2cl-C	47.2 ± 0.90	**67.4** ± 0.79
G-3cl-A	**70.4** ± 1.13	69.6 ± 0.53
G-3cl-B	63.2 ± 1.92	**80.2** ± 1.81
G-3cl-C	65.6 ± 2.02	**77.9** ± 1.88
G-5cl-A	64.5 ± 0.58	**74.1** ± 1.24
G-5cl-B	55.7 ± 0.83	**60.2** ± 1.45
G-5cl-C	**57.1** ± 2.91	55.1 ± 1.22

NMI Datasets	LM	LM-BoP
G-2cl-A	55.3 ± 2.77	**68.0** ± 1.35
G-2cl-B	38.4 ± 2.78	**45.7** ± 1.89
G-2cl-C	55.7 ± 1.14	**67.0** ± 1.75
G-3cl-A	**67.9** ± 1.31	67.5 ± 0.61
G-3cl-B	63.5 ± 2.07	**73.7** ± 2.37
G-3cl-C	64.3 ± 2.03	**72.0** ± 1.76
G-5cl-A	66.1 ± 0.66	**70.4** ± 1.25
G-5cl-B	62.8 ± 1.21	**65.3** ± 1.53
G-5cl-C	56.9 ± 2.13	**58.2** ± 1.69

TABLE 4.1: Performance comparison with Adjusted Rand Index (ARI) and Normalized Mutual Information (NMI) between the Louvain method (LM) and the extended Louvain method using the BoP probabilities (LM-BoP).

probabilities of observing edges within a community. The BoP modularity extends this idea by considering probabilities of observing paths between nodes within a community, instead of just edges. As explained earlier, the theory and the small experiment presented here were originally written as a section in [82]. The section was withdrawn from the article, but it was never extended to a full paper, which it might actually still deserve, as the idea behind the BoP-modularity is intuitive and attractive, and because the initial experimental results presented here indicate a potential of this approach for community detection.

Chapter 5

Two betweenness centrality measures based on randomized shortest paths

5.1 Introduction

One of the most fundamental and popular topics in network science is determining the *centrality* of a node in a network according to the structure of the network. The concept of centrality can be interpreted in many ways and a vast number of measures have been proposed based on different interpretations. One commonly used interpretation is *betweenness centrality*, which reflects the extent to which a node lies in between pairs or groups of other nodes of the graph. This can be also stated as the extent to which a node is an intermediate in communication over the network. Different models have been proposed to measure the participation of a node in this communication ranging from the *shortest path betweenness centrality* of Freeman [83, 84], which considers communication flowing only along the shortest paths, to the *current flow betweenness centrality* [157, 36], which interprets communication flowing as electric current or as random walks in the network.

In this chapter we propose two families of betweenness centrality measures based on the *Randomized Shortest Paths* (RSP) framework [215, 182, 115]. The framework is based on Boltzmann probability distributions over paths between the nodes of a network which focus on short, optimal paths, but give some probability mass also to longer paths. The extent of focus on optimal paths is controlled by an inverse temperature parameter β. The RSP framework has previously been shown to function well when defining distance measures on networks for clustering and classification of network nodes [115]. In this

work we extend the study of RSPs by showing their potential also in defining centrality measures. The two RSP betweenness centralities presented in this chapter measure the involvement of each node in RSPs between the nodes of the graph. The first measure, which we call the simple RSP betweenness, measures the expected number of visits to a node during RSPs, while the second one, called the RSP net betweenness is the sum of expected net flows over the edges connected to a node.

The proposed RSP betweenness measures are attractive both theoretically as well as in practice. Theoretical interest is ensured by the fact that both measures can be seen as generalizations of classical betweenness measures. With large values of the parameter β, both RSP betweenness measures converge to a measure that we introduce as the shortest path likelihood betweenness, which is very closely related to the original betweenness centrality defined by Freeman [83, 84] and its other similar variant, the load centrality [89]. The reason for defining two different betweenness measures with RSPs is in their behaviour as β is decreased. Namely, the simple RSP betweenness then converges to the stationary distribution of a random walk on the network (multiplied by a constant), whereas the RSP net betweenness converges to the current flow betweenness [157, 36]. The definition, as well as the computation, of the simple RSP betweenness are more straightforward than for its counterpart, the RSP net betweenness, which can also be stated of their corresponding limit functions. In addition, the experiments in Section 5.4 indicate that the simple RSP betweenness can in practice be more useful than the RSP net betweenness or the current flow betweenness. However, the choice of which definition to rely on in the end depends on the application domain.

Considering betweenness based on RSPs is motivated by the fact that measures based only on shortest paths or on random walks alone often involve undesirable features. Shortest paths in a complex network tend to pass through only a small fraction of the nodes of the network, which can cause highly skewed betweenness score distributions and fail in differentiating between the other nodes of the network [89, 16]. Also, when considering communication or navigation in networks, it is not always even realistic to consider that they would occur along only the shortest, nor completely random paths. Instead, movement may follow a partly random route, with a drift towards a destination, for example when the navigating agent does not know the optimal way or wants to add secrecy and unpredictability to its route. As a result, the trajectories that are actually used in the network can be more spread over different nodes in areas of the network that contain many connections. Because of this, RSPs can help, for instance, in detecting bottlenecks of the network, where there exist no alternatives for the shortest path.

Measures based solely on random walks do take into account the abundance

of connections between nodes. However, they may in many situations depend heavily on local features of a graph, especially for large graphs [139, 151, 210] instead of capturing its interesting global properties. Thus, some regularisation over the degree of randomness is needed, which in the RSP framework is controlled by the inverse temperature parameter β.

Models that find a compromise between the optimal shortest path and a random walk have recently received a lot of attention. Compared to these, the attractive aspect in using the RSP framework is that the compromise between the shortest and random paths is optimal by definition, as the Gibbs-Boltzmann distribution minimizes the expected cost of paths subject to a fixed relative entropy [215, 182]. The minimization can also be expressed with respect to free energy [115]. In addition to the optimality aspect, the computation of quantities related to RSPs is fairly straightforward to implement and can be run in reasonable time for moderate-sized graphs. However, a drawback of the algorithms presented in the chapter is that they rely on matrix inversion, or, alternatively, solving a system of linear equations, which can be prohibitive for computations with very large networks. However, in the future we plan to develop more specialized methods that will enable the RSP quantities to be computed with large networks as well, e.g. taking advantage of the possible sparseness of the matrices and linear systems involved in the computations.

Ideas similar to the RSP framework of interpolating between the two extremes can be found in the work of Alamgir and von Luxburg [3], involving p-resistances and graph node distances based on them, of Chebotarev involving distances based on the matrix forest theorem [45, 43], of Zhang and Boley [137] with focus on routing schemes, of Estrada, who has defined the subgraph centrality [69] and the communicability betweenness [68]. Even more related to this chapter are the recent works of Bavaud and Guex [17] and Lebichot et al. [129]. In fact, the two betweenness centrality measures presented in this chapter can be shown equal to the betweenness measures proposed by Bavaud and Guex [17], although the relation between the two works is not entirely obvious. We will discuss this relation and the relation between the RSP betweenness measures and other previously proposed betweenness measures in more detail in Section 5.2.

To sum up, the contributions of this chapter are:

- we define two betweenness centrality measures which form a spectrum between measures based on shortest paths and pure random walks, therefore integrating information about both the optimality and the abundance of paths between nodes of the network,
- we derive algorithms for computing these measures in a convenient and efficient way, and

- we demonstrate with example networks that the proposed RSP betweenness measures may provide a more interesting ranking of the network nodes than the classical measures that they generalize.

The structure of the chapter is as follows: Section 5.2 reviews the literature of betweenness centrality measures of different kind. Section 5.3 reviews the RSP framework and then introduces the RSP betweenness centrality measures and the methodology for computing them. Section 5.4 then presents example cases that illustrate the benefits of using RSP betweenness measures.

5.2 Betweenness centralities

In this section, we extend the more general review of Section 2.5 on network centrality measures. However, here we focus more specifically on betweenness centrality measures.

Note that when considering betweennes measures there are different possibilities in whether or not the source and target nodes, s and t should be considered also as intermediate nodes of a path for centrality considerations. In this chapter we use the convention where the first node of a path does affect the betweenness scores, but the last one does not. For betweenness measures based on shortest paths this choice only changes the overall betweenness scores by an additive constant in a strongly connected graph, and thus only affects the ranking of nodes when the network has several components. In contrast, when measuring betweenness based on random walks, further visits to the starting node s after the first step may increase the betweenness score of node s and thus may also affect the rankings. Betweenness measures are also often normalized, e.g., according to the number of possible node pairs, when considering shortest paths between nodes. However, in this chapter we leave the normalization out of consideration in all the definitions, because it never affects the rankings of nodes within a strongly connected network.

5.2.1 Betweenness based on shortest paths

Possibly the best-known centrality measure of all is the original *betweenness centrality* of Freeman [83, 84], which counts the fraction of shortest paths between a pair of nodes that an intermediate node lies on and sums these fractions over all node pairs. We will also refer to it as *shortest path betweenness*, for specificity.

Let us denote the set of shortest paths from node s to node t by $\mathcal{P}^*_{st}$, the total number of such paths by $|\mathcal{P}^*_{st}|$ and the cost of the shortest path from s to t by

$\tilde{c}^*_{st}$. The directed graph that consists only of the nodes and directed edges that belong to the shortest paths from s to t is called the *directed shortest path graph from s to t*.

Formally, the shortest path betweenness of node i can be expressed as

$$C_i^{\mathrm{Bet}} = \sum_{s,t=1}^{n} \frac{n(i \in \mathcal{P}^*_{st})}{|\mathcal{P}^*_{st}|}, \tag{5.1}$$

where $n(i \in \mathcal{P}^*_{st})$ means the number of shortest paths that contain node i. Notice that if there are more than one shortest path connecting s to t, each of these paths will contribute a score of $1/|\mathcal{P}^*_{st}|$ to the betweenness of the nodes on them.

There exist several variations of the shortest path betweenness. A thorough review of these variants and their efficient computation was provided by Brandes [34]. One variant, called the *load centrality* [89, 155, 34], replaces the fractional term inside the sum in (5.1) with the *branching probability* of a path, i.e. the probability that a random walker moving in the directed shortest path graph from s to t follows a path that contains node i, when at each branching point in this graph it selects the edge to follow with uniform probability over all outgoing edges. For illustration of the difference between the shortest path betweenness and load betweenness, see [34]. With weighted graphs the two measures are usually equal, as there often exists a unique shortest path between all nodes, especially if the weights are real-valued.

In what follows, we will need to consider another variant of the shortest path betweenness, which we have not encountered in the literature previously. We call this measure the *shortest path likelihood betweenness*. We define it for a node $i \in V$ similarly to the load centrality, by replacing the term inside the sum in (5.1) with the sum of *shortest path likelihoods*, $\mathrm{P}^*_{st}(\wp^*_k)$ from Equation (2.11) of the shortest paths that contain node i. In other words, we define the shortest path likelihood betweenness of node i as

$$C_i^{\mathrm{LikeBet}} = \sum_{s,t}^{n} \sum_{\wp^*_{st} \in \mathcal{P}^*_{st}} \mathrm{P}^*_{st}(\wp^*_{st})[i \in \wp^*_{st}], \tag{5.2}$$

where $[i \in \wp^*_{st}]$, based on the *Iverson brackets* [122], is 1 if node i appears on path $\wp^*_{st}$, and 0 otherwise.

Note, that the shortest path likelihood of a path is different from its branching probability which is based on transition probabilities in the directed shortest path graph from s to t, instead of the transition probabilities defined originally

for the whole graph. The shortest path likelihood betweenness is introduced here for the sake of completeness because it is the limiting function of both RSP betweenness measures presented in this chapter. However, the differences between the three variants of shortest path betweenness presented above are very small, and in practice they provide very similar rankings of the nodes of a network. Especially, when the shortest paths between node-pairs are unique (which is often the case for real-valued edge costs), the standard shortest path betweenness, the load betweenness and the shortest path likelihood betweenness all coincide.

5.2.2 Betweenness based on random walks

Betweenness has also been considered with respect to other than just shortest paths, namely by considering random walks or flows on a network. The first such measure was proposed by Freeman [85], who defined the *flow betweenness centrality* as the amount of flow through a node over maximum flows between all node pairs. The idea of considering flows was developed further by Newman [157], who defined the *current flow betweenness centrality*, which measures the centrality of a node as the total sum of electrical current that flows through it, when considering all node pairs as source-sink pairs of a unit current flow. The current flow betweenness was also coined the *random walk betweenness centrality* because of the well-established connection between electric current flows and random walks [59]. Indeed, it can also be interpreted as the sum of expected net flows of a random walk over the edges connected to a node, meaning that the number of times that the walk enters or leaves the node along an edge cancel each other out. Brandes and Fleischer [36] developed an algorithm which improves the efficiency of computing the current flow betweenness for all nodes of a network. The properties and computation of the current flow betweenness have also been studied by Bozzo and Franceschet [33].

Instead of considering net flows, a more straightforward definition of betweenness based on random walks is simply the overall expected number of visits to a node during a random walk. For arbitrary, non-hitting paths, this quantity is not necessarily well-defined, as the probability of non-hitting paths cannot necessarily be well-defined. However, it is possible to compute the proportion of steps of such a walk that the walker spends in a node. This is also, loosely speaking, equal to the probability of finding the walker at the node after a long walk, which is known as the *stationary probability* of the node in a random walk on G. The stationary probabilities define the *stationary distribution*, which is the unique vector $\boldsymbol{\pi}$ with positive elements that satisfies $\boldsymbol{\pi} = (\mathbf{P}^{\mathrm{rw}})^{\mathsf{T}}\boldsymbol{\pi}$

and $\sum_k \pi_k = 1$. For directed graphs that are not strongly connected the stationary distribution is not necessarily unique or may not even exist. To overcome this issue, for instance, the PageRank algorithm [164] uses a teleportation probability which transforms any kind of a graph into a strongly connected approximation and computes the stationary distribution on the transformed graph. However, for strongly connected graphs, the stationary distribution always exists and is unique [94]. Moreover, it is well known that the stationary probabilities are proportional to the recurrence times and, if the graph is undirected, to the degree centralities (or strength, for weighted graphs) of nodes [94, Theorem 11.10].

Even though the expected number of visits to a node is not well-defined for arbitrary paths, it is well-defined for hitting paths. A bit surprisingly, we have discovered that when computed over all hitting random walks on a strongly connected graph, this measure is equal to the stationary distribution, up to a multiplying constant. The multiplying constant is the sum, over all s-t-pairs of the graph, of expected hitting times, $\langle L \rangle_{st}^{\mathrm{rw}}$. Especially, when considering undirected graphs, as the commute time distances are propo

This result is not completely obvious and did not seem to appear explicitly in the literature at the time of writing the article [117] upon which this chapter is based on. In the original article, we left the result without proof, as it would have been a bit of a sidestep from the focus of the article. However, here we give a proof, which was devised with help from Jean-Charles Delvenne, whom we thank. Interestingly, the result is also stated in a recent paper [92], which has appeared after the publication of [117].

Let $\bar{\eta}_i^{\mathrm{rw}}(s,t)$ denote the expected number of visits to node i over all hitting paths going from node s to node t, w.r.t. the natural random walk probabilities $\mathrm{P}_{st}^{\mathrm{rw}}$. In the definition, the first node of a path is counted as a visit to node s, but the last node of the path is not counted as a visit to t. Moreover, let $\bar{\eta}_i^{\mathrm{rw}} = \sum_{s,t} \bar{\eta}_i^{\mathrm{rw}}(s,t)$.

Theorem 5.2.1.

$$\bar{\eta}_i^{\mathrm{rw}} = \pi_i \sum_{s,t} \langle L \rangle_{st}^{\mathrm{rw}}. \tag{5.3}$$

For proving Theorem 5.2.1, we first derive an expression for $\bar{\eta}_i(s,t)$ using the elements of the *fundamental matrix* of the graph

$$\mathbf{Z} = (\mathbf{I} - \mathbf{P}^{\mathrm{rw}} + \mathbf{e}\boldsymbol{\pi}^{\mathsf{T}})^{-1}. \tag{5.4}$$

Note that although the matrix $\mathbf{I} - \mathbf{P}^{\mathrm{rw}}$ is singular, adding matrix $\mathbf{e}\boldsymbol{\pi}^{\mathsf{T}}$ to it makes it invertible, as $\boldsymbol{\pi}$ spans the left null space of $\mathbf{I} - \mathbf{P}^{\mathrm{rw}}$. It is well known (Theorem 11.16 in [94]) that the expected hitting time can be expressed using

the elements of $\mathbf{Z}$, as

$$\langle L \rangle_{st}^{\mathrm{rw}} = \frac{z_{tt} - z_{st}}{\pi_t} \tag{5.5}$$

Lemma 5.2.2 provides a similar expression for the number of visits in a similar fashion as the proof for (5.5) in [94]. As a sidenote, a result similar to Lemma 5.2.2 could be derived by combining Lemmas 2.9 and 2.11 in [6].

Lemma 5.2.2.

$$\bar{\eta}_i^{\mathrm{rw}}(s,t) = z_{si} - z_{ti} + \pi_i \langle L \rangle_{st}^{\mathrm{rw}} \tag{5.6}$$

In [92] the above quantity is interpreted as element (s,i,t) of the 3-dimensional *fundamental tensor* of random walks, which can be used for computing many different random walk based measures on a graph. The authors there define a way for computing the quantity for any triplet (s,i,t) requiring the computation of the pseudoinverse of the Laplacian matrix only once. This result provides an alternative for that solution, relying on the fundamental matrix $\mathbf{Z}$ instead of the pseudoinverse of the Laplacian.

Proof of Lemma 2. The function $\bar{\eta}_i^{\mathrm{rw}}(s,t)$ is zero for the cases $s = t$ and $t = i$. For other cases it can be expressed recursively as

$$\bar{\eta}_i^{\mathrm{rw}}(s,t) = \delta_{si} + \sum_{k=1}^{n} p_{sk}^{\mathrm{rw}} \bar{\eta}_i^{\mathrm{rw}}(k,t),$$

where δ_{si} is the Kronecker delta. Thus, for arbitrary s, t and i:

$$\begin{aligned}
\bar{\eta}_i^{\mathrm{rw}}(s,t) &= (1-\delta_{st})(1-\delta_{ti})(\delta_{si} + \sum_{k=1}^{n} p_{sk}^{\mathrm{rw}} \bar{\eta}_i^{\mathrm{rw}}(k,t)) \\
&= \delta_{si} - \delta_{sti} + \sum_{k=1}^{n} p_{sk}^{\mathrm{rw}} \bar{\eta}_i^{\mathrm{rw}}(k,t) - \delta_{st} \sum_{k=1}^{n} p_{sk}^{\mathrm{rw}} \bar{\eta}_i^{\mathrm{rw}}(k,t) \\
&\quad \underbrace{- \delta_{ti} \sum_{k=1}^{n} p_{sk}^{\mathrm{rw}} \bar{\eta}_i^{\mathrm{rw}}(k,t)}_{=0} + \underbrace{\delta_{sti} \sum_{k=1}^{n} p_{sk}^{\mathrm{rw}} \bar{\eta}_i^{\mathrm{rw}}(k,t)}_{=0}
\end{aligned}$$

The last two terms of the above expression are zero, as, by definition, $\bar{\eta}_i^{\mathrm{rw}}(s,i) = 0$. The expression can be written in matrix form as

$$\mathbf{N} = \mathbf{e}_i \mathbf{e}^{\mathsf{T}} - \mathbf{e}_i \mathbf{e}_i^{\mathsf{T}} + \mathbf{P}^{\mathrm{rw}} \mathbf{N} - \mathbf{D}, \tag{5.7}$$

where $\mathbf{D} = \mathbf{diag}(\mathbf{P}^{\mathrm{rw}}\mathbf{N})$. Next, we show that the diagonal elements of $\mathbf{D}$ have a simple form. Namely, by rewriting (5.7) as

$$(\mathbf{I} - \mathbf{P}^{\mathrm{rw}})\mathbf{N} = \mathbf{e}_i\mathbf{e}^{\mathsf{T}} - \mathbf{e}_i\mathbf{e}_i^{\mathsf{T}} - \mathbf{D}, \tag{5.8}$$

and multiplying this from the left with $\boldsymbol{\pi}^{\mathsf{T}}$, because $\boldsymbol{\pi}^{\mathsf{T}}(\mathbf{I} - \mathbf{P}^{\mathrm{rw}}) = \mathbf{0}$, we get

$$\mathbf{d}^{\mathsf{T}} = \pi_i\mathbf{e}^{\mathsf{T}} - \pi_i\mathbf{e}_i^{\mathsf{T}} \tag{5.9}$$

where $d_j = \pi_j(\mathbf{P}^{\mathrm{rw}}\mathbf{N})_{jj}$. From this we know that

$$(\mathbf{D})_{jj} = (\mathbf{P}^{\mathrm{rw}}\mathbf{N})_{jj} = \begin{cases} \pi_i/\pi_j, & \text{for } j \neq i \\ 0, & \text{for } j = i. \end{cases} \quad \text{i.e.} \quad (\mathbf{D})_{jj} = (1 - \delta_{ji})\frac{\pi_i}{\pi_j}. \tag{5.10}$$

Adding $\mathbf{e}\boldsymbol{\pi}^{\mathsf{T}}\mathbf{N}$ on both sides of (5.8) gives

$$(\mathbf{I} - \mathbf{P}^{\mathrm{rw}} + \mathbf{e}\boldsymbol{\pi}^{\mathsf{T}})\mathbf{N} = \mathbf{e}_i\mathbf{e}^{\mathsf{T}} - \mathbf{e}_i\mathbf{e}_i^{\mathsf{T}} - \mathbf{D} + \mathbf{e}\boldsymbol{\pi}^{\mathsf{T}}\mathbf{N} \tag{5.11}$$

and multiplying from the left with $\mathbf{Z}$, we get

$$\mathbf{N} = \mathbf{Z}\mathbf{e}_i\mathbf{e}^{\mathsf{T}} - \mathbf{Z}\mathbf{e}_i\mathbf{e}_i^{\mathsf{T}} - \mathbf{Z}\mathbf{D} + \mathbf{Z}\mathbf{e}\pi^{\mathsf{T}}\mathbf{N}. \tag{5.12}$$

The above equation holds for any s, i and t. Let us now assume $i \neq t$. Then knowing that $\mathbf{Z}\mathbf{e} = \mathbf{e}$, (as also $(\mathbf{I} - \mathbf{P}^{\mathrm{rw}} + \mathbf{e}\boldsymbol{\pi}^{\mathsf{T}})\mathbf{e} = \mathbf{e}$), we can write $\bar{\eta}_i^{\mathrm{rw}}(s,t)$, using Equation (5.10) as

$$\bar{\eta}_i^{\mathrm{rw}}(s,t) = \mathbf{e}_s^{\mathsf{T}}\mathbf{N}\mathbf{e}_t = z_{si} - z_{si}\underbrace{\mathbf{e}_i^{\mathsf{T}}\mathbf{e}_t}_{=0} - z_{st}(\mathbf{D})_{tt} + \boldsymbol{\pi}^{\mathsf{T}}\mathbf{N}\mathbf{e}_t \tag{5.13}$$

$$= z_{si} - \frac{\pi_i}{\pi_t}z_{st} + \sum_{k=1}^{n}\pi_k\bar{\eta}_i^{\mathrm{rw}}(k,t). \tag{5.14}$$

We know that $\bar{\eta}_i^{\mathrm{rw}}(t,t) = 0$, from which we get

$$\sum_{k=1}^{n}\pi_k\bar{\eta}_i^{\mathrm{rw}}(k,t) = \frac{\pi_i}{\pi_t}z_{tt} - z_{ti}. \tag{5.15}$$

Inserting this to (5.14) and remembering Equation (5.5) then gives

$$\bar{\eta}_i^{\mathrm{rw}}(s,t) = z_{si} - z_{ti} + \frac{\pi_i}{\pi_t}(z_{tt} - z_{st}) = z_{si} - z_{ti} + \pi_i\langle L\rangle_{st}^{\mathrm{rw}}. \tag{5.16}$$

The proof is finished by noting that the above holds also for the case $i = t$, as the right side of the equation becomes 0, as expected. □

Proof of Theorem 1. The result follows from Lemma 2 by summing (5.6) over all s and t, as the terms $z_{si} - z_{ti}$ cancel each other out. □

Note that the result of Theorem 5.2.1 can also be stated, based on the Kirchhoff index, $\mathcal{K}(G)$ (see Equation (2.36)), as $\sum_{s,t}\langle L\rangle_{st}^{\mathrm{rw}} = \mathcal{K}(G)\sum_{(i,j)\in E} a_{ij}$, remembering Equation (2.42).

The RSP betweenness centrality measures proposed in this chapter are also based on random walks, more particularly on the RSP framework, presented in Section 2.3. The first betweenness measure is based on counting the expected number of passages through a node of a random walker moving according to the RSP probabilities, while the other computes the expected net flow of walkers going through the node. Accordingly, the limit of the first betweenness measure, for node i, is $\bar{\eta}_i^{\mathrm{rw}}$, as will be discussed later.

In fact, the RSP betweenness measures coincide with betweenness measures proposed recently, independently of our work, by Bavaud and Guex [17]. They propose a framework which interpolates between shortest paths and random walks by the minimization of free energy in a similar fashion as the free energy derivation of the RSP framework of Kivimäki et al. [115]. One main difference in the work of Bavaud and Guex [17], compared to the derivation of RSPs, is that a more general form of energy functionals, besides the expected length (or cost) of a path, is considered. In addition to that the relative entropy is considered with respect to transition probabilities instead of path probabilities, as is done in the definition of RSPs. The path distribution derived by Bavaud and Guex [17] can, however, be shown to equal the path distribution defined in the RSP framework, although this requires some lengthy and uninteresting derivations and is left out of this chapter. The relation between the two approaches has also been studied in the more recent work of Guex and Bavaud [96]. Compared to the work of Bavaud and Guex [17], our work focuses more on the computational and practical aspects of the methodology. Thanks to the recent developments in the RSP framework [115], we are also able to present more efficient and straightforward algorithms for computing the quantities in question and illustrate the use of the methodology with practical examples.

Also closely related to the idea behind the RSP betweenness centralities is the *bag-of-paths (BoP) betweenness centrality* [129]. It is based on the BoP framework, which also defines a Gibbs-Boltzmann distribution on the paths of a graph, but considering the whole set of paths on the graph instead of between a fixed s-t-pair. The BoP betweenness is then defined as the a posteriori probability that a path selected according to the Gibbs-Boltzmann distribution visits an intermediate node i, when it walks from node s to node t according to the BoP probabilities. The BoP betweenness is also defined for groups and used for semi-supervised classification of graph nodes. A similar group betweenness

measure can be defined from the RSP betweenness measures proposed here, but this is left out of the scope of this chapter. The node classification task was also tackled by Devooght et al. [54] by using a modularity measure derived from the BoP framework.

5.3 Randomized shortest paths betweenness centralities

In this section we introduce two betweenness measures based on the randomized shortest paths (RSP) framework [215, 182, 115]. These measures both generalize the shortest path likelihood betweenness measure defined in Section 5.2.1. In addition to that, one of the two measures also generalizes the stationary distribution of the graph while the other generalizes the current flow betweenness centrality.

5.3.1 The simple RSP betweenness

We first define the simple RSP betweenness centrality of a node i *with respect to paths from* s *to* t, as the expected number of visits through i over all hitting s-t-paths, denoted by $\bar{\eta}_i(s,t)$, with respect to the RSP probabilities of Equation (2.10). The more exact definition is stated as the expected number of passages through edges leaving from node i over all s-t-paths, denoted by $\bar{\eta}_{ij}(s,t)$, as:

$$\mathrm{bet}_i^{\mathrm{RSP}}(s,t) = \bar{\eta}_i(s,t) = \sum_{j:(i,j)\in E} \bar{\eta}_{ij}(s,t) \tag{5.17}$$

The function $\mathrm{bet}_i^{\mathrm{RSP}}$ can be useful for visualizing path distributions and for path planning tasks between two fixed nodes of the graph [87, 168]. Moreover, it can be used to investigate how central an intermediate actor is with respect to, for instance, the communication between two other actors in a social network.

We then define the overall *simple RSP betweenness centrality* of node i as the sum of contributions over all s-t-pairs on the graph:

$$\mathrm{bet}_i^{\mathrm{RSP}} = \sum_{s=1}^{n}\sum_{t=1}^{n} \mathrm{bet}_i^{\mathrm{RSP}}(s,t). \tag{5.18}$$

Next, we derive the method for computing this quantity in closed form. Let us denote by $\eta_{ij}(\wp)$ the number of times that the edge (i,j) is traversed along a

path $\wp$. Then, we can express $\bar{\eta}_{ij}(s,t)$ in terms of the RSP distribution (Equation (2.10)) based on the partition function of hitting paths from s to t, $\mathcal{Z}_{st}$:

$$\begin{aligned}\bar{\eta}_{ij}(s,t) &= \sum_{\wp\in\mathcal{P}_{st}} \mathrm{P}_{st}(\wp)\eta_{ij}(\wp) \\ &= \frac{1}{\mathcal{Z}_{st}} \sum_{\wp\in\mathcal{P}_{st}} \mathrm{P}^{\mathrm{rw}}_{st}(\wp)\exp(-\beta\tilde{c}(\wp))\eta_{ij}(\wp) \\ &= \frac{1}{\mathcal{Z}_{st}} \sum_{\wp\in\mathcal{P}_{st}} \mathrm{P}_{st}(\wp)\exp(-\beta\tilde{c}(\wp))\,\frac{\partial\tilde{c}(\wp)}{\partial c_{ij}} \\ &= -\frac{1}{\beta}\frac{\partial\log\mathcal{Z}_{st}}{\partial c_{ij}}.\end{aligned} \tag{5.19}$$

Thus, the expected number of transitions $i \to j$ over hitting s-t-paths can be computed by differentiating the logarithm of the partition function, which is common knowledge in statistical physics (see e.g. [170]). Note that (5.19) holds only if there exists a path from s to t. Otherwise, naturally, $\bar{\eta}_{ij}(s,t) = 0$.

Remembering then, from Equation (3.2), that the partition function of hitting paths can be computed from matrix $\mathbf{Z}$, of Equation (2.19), as $\mathcal{Z}_{st} = z_{st}/z_{tt}$, we can write Equation (5.19) further as:

$$\begin{aligned}\bar{\eta}_{ij}(s,t) &= -\frac{1}{\beta}\frac{\partial\log\mathcal{Z}_{st}}{\partial c_{ij}} = -\frac{1}{\beta}\frac{\partial\log(z_{st}/z_{tt})}{\partial c_{ij}} \\ &= -\frac{1}{\beta}\left(\frac{1}{z_{st}}\frac{\partial z_{st}}{\partial c_{ij}} - \frac{1}{z_{tt}}\frac{\partial z_{tt}}{\partial c_{ij}}\right).\end{aligned} \tag{5.20}$$

Therefore, we need to compute $\partial z_{st}/\partial c_{ij}$, which can be achieved using matrix formalism. If we denote by $\mathbf{e}_i$ the $(n\times 1)$-vector whose element i is 1 and others are 0, then

$$\begin{aligned}&\frac{\partial z_{st}}{\partial c_{ij}} = \frac{\partial(\mathbf{e}_s^\mathsf{T}\mathbf{Z}\mathbf{e}_t)}{\partial c_{ij}} = \frac{\partial(\mathbf{e}_s^\mathsf{T}(\mathbf{I}-\mathbf{W})^{-1}\mathbf{e}_t)}{\partial c_{ij}} \\ &= \mathbf{e}_s^\mathsf{T}\frac{\partial(\mathbf{I}-\mathbf{W})^{-1}}{\partial c_{ij}}\mathbf{e}_t = \mathbf{e}_s^\mathsf{T}\left(-\mathbf{Z}\frac{\partial(\mathbf{I}-\mathbf{W})}{\partial c_{ij}}\mathbf{Z}\right)\mathbf{e}_t \\ &= \mathbf{e}_s^\mathsf{T}\left(\mathbf{Z}\frac{\partial\mathbf{W}}{\partial c_{ij}}\mathbf{Z}\right)\mathbf{e}_t = -\beta\, p^{\mathrm{ref}}_{ij}\exp[-\beta c_{ij}]\mathbf{e}_s^\mathsf{T}\mathbf{Z}\mathbf{e}_i\mathbf{e}_j^\mathsf{T}\mathbf{Z}\mathbf{e}_t \\ &= -\beta w_{ij} z_{si} z_{jt}\end{aligned} \tag{5.21}$$

where we used $\frac{\partial \mathbf{X}^{-1}}{\partial x} = -\mathbf{X}^{-1}\frac{\partial \mathbf{X}}{\partial x}\mathbf{X}^{-1}$ (see, e.g., [102, 187]). Equation (5.20) can therefore be rewritten as

$$\begin{aligned}\bar{\eta}_{ij}(s,t) &= \frac{1}{z_{st}} z_{si} w_{ij} z_{jt} - \frac{1}{z_{tt}} z_{ti} w_{ij} z_{jt} \\ &= \left(\frac{z_{si}}{z_{st}} - \frac{z_{ti}}{z_{tt}}\right) w_{ij} z_{jt} \end{aligned} \tag{5.22}$$

Furthermore, the total number of visits to node i is

$$\bar{\eta}_i(s,t) = \sum_{j=1}^{n} \bar{\eta}_{ij}(s,t) = \left(\frac{z_{si}}{z_{st}} - \frac{z_{ti}}{z_{tt}}\right) \sum_{j=1}^{n} w_{ij} z_{jt} \tag{5.23}$$

The expression on the right-hand side of Equation (5.23) can be further simplified in the following way. We know that $\mathbf{Z}(\mathbf{I} - \mathbf{W}) = \mathbf{I}$ and therefore that $\mathbf{Z} = \mathbf{ZW} + \mathbf{I}$, or element-wise, $z_{it} = \sum_{j=1}^{n} w_{ij} z_{jt} + \delta_{it}$, where δ_{it} is the Kronecker delta. However, the term δ_{it} can be discarded when considering Equation (5.23), because when $i = t$, we anyway have $(z_{si}/z_{st} - z_{ti}/z_{tt}) = 0$. Thus, Equation (5.23) simplifies to

$$\bar{\eta}_i(s,t) = \left(\frac{z_{si}}{z_{st}} - \frac{z_{ti}}{z_{tt}}\right) z_{it} \tag{5.24}$$

Finally, the simple RSP betweenness centrality (Equation (5.18)) can be computed in matrix form as

$$\begin{aligned}\mathrm{bet}_i^{\mathrm{RSP}} &= \sum_{s,t=1}^{n} \bar{\eta}_i(s,t) = \sum_{s,t=1}^{n} \left(\frac{z_{si}}{z_{st}} - \frac{z_{ti}}{z_{tt}}\right) z_{it} \\ &= \sum_{s,t=1}^{n} z_{it} \frac{1}{z^{\mathrm{t}}_{ts}} z_{si} - \sum_{s,t=1}^{n} z_{it} \frac{1}{z_{tt}} z_{ti} \\ &= \sum_{s,t=1}^{n} z_{it} \frac{1}{z^{\mathrm{t}}_{ts}} z_{si} - n \sum_{t=1}^{n} z_{it} \frac{1}{z_{tt}} z_{ti} \\ &= \left[\mathbf{diag}\Big(\mathbf{Z}\big(\mathbf{Z}^{\div}\big)^{\mathsf{T}}\mathbf{Z}\Big) - n\,\mathbf{diag}\big(\mathbf{Z}\,\mathbf{Diag}\big(\mathbf{Z}^{\div}\big)\mathbf{Z}\big)\right]_i \\ &= \left[\mathbf{diag}\Big(\mathbf{Z}\big(\mathbf{Z}^{\div} - n\,\mathbf{Diag}\big(\mathbf{Z}^{\div}\big)\big)^{\mathsf{T}}\mathbf{Z}\Big)\right]_i \end{aligned} \tag{5.25}$$

which provides a way for computing the vector of betweenness values for all nodes at once. Here $\mathbf{diag}(\mathbf{X})$ and $\mathbf{Diag}(\mathbf{X})$ are, respectively, a column vector

Algorithm 2 Computing the simple RSP betweenness vector of a graph G.

Input:
– A directed strongly connected graph G with n nodes.
– The $n \times n$ reference transition probability matrix $\mathbf{P}^{\mathrm{rw}}$ (defined from the adjacency matrix as $\mathbf{P}^{\mathrm{rw}} = \mathbf{D}^{-1}\mathbf{A}$)
– The $n \times n$ non-negative cost matrix $\mathbf{C}$
– The inverse temperature parameter β.
Output:
– The $n \times 1$ simple RSP betweenness vector $\mathbf{bet}^{\mathrm{RSP}}$.
1. $\mathbf{W} \leftarrow \mathbf{P}^{\mathrm{rw}} \circ \exp[-\beta \mathbf{C}]$
2. $\mathbf{Z} \leftarrow (\mathbf{I} - \mathbf{W})^{-1}$
3. $\mathbf{Z}^{\div} \leftarrow \mathbf{e}\mathbf{e}^{\mathsf{T}} \div \mathbf{Z}$
4. $\mathbf{bet}^{\mathrm{RSP}} \leftarrow \mathbf{diag}\big(\mathbf{Z}(\mathbf{Z}^{\div} - n\,\mathbf{Diag}(\mathbf{Z}^{\div}))^{\mathsf{T}}\mathbf{Z}\big)$

and a diagonal matrix containing the diagonal of $\mathbf{X}$, $\mathbf{X}^{\div}$ denotes the element-wise reciprocal matrix, i.e., $x_{ij}^{\div} = 1/x_{ij}$ and the superscript t denotes elements from the transposed matrix, i.e., $z_{ij}^{\mathrm{t}} = z_{ji}$.

The pseudocode for computing the simple RSP betweenness for all nodes is presented in Algorithm 2.[1] In conclusion, the simple RSP betweenness scores of all nodes can be computed by performing the matrix inversion $\mathbf{Z} = (\mathbf{I} - \mathbf{W})^{-1}$ and then simple matrix operations according to Equation (5.25). Thus, the computational bottleneck of the algorithm is the matrix inversion, which, in general, has time complexity $\mathcal{O}(n^3)$ and space complexity $\mathcal{O}(n^2)$. To give a rough idea of the computation time required by the algorithm, computing the simple RSP betweenness scores for one value of β on the Manhattan street network, studied later in Section 5.4.2, which contains 2250 nodes and 3863 (undirected) edges, takes around 2.6 seconds on a laptop computer with a 2,3 GHz Intel Core i5 processor and 8GB of RAM.

5.3.2 RSP net betweenness

Instead of only considering the overall outgoing flow of paths, as in the definition of Equation (5.18), it may in some cases make more sense to compute the *net flow* so that the outgoing and ingoing flows through one edge neutralize each other. This corresponds to the random walk interpretation of the current flow betweenness [157, 36, 33]. According to this approach, we define the *RSP*

[1] The Matlab code for the algorithms presented in the chapter, as well as the real world networks used in the experiments, are available online at https://github.com/ikivimak/RSP-betweenness

net betweenness centrality of node i as

$$\mathrm{bet}_i^{\mathrm{net}} = \sum_{s,t=1}^{n} \sum_{j \in Succ(i)} |\bar{\eta}_{ij}(s,t) - \bar{\eta}_{ji}(s,t)| \tag{5.26}$$

The computation of this quantity for all nodes at once is a bit more involved than in the case of the simple RSP betweenness, because of the absolute value in the expression. A naïve algorithm would loop over all s-t-pairs and compute the contribution of the corresponding paths to each intermediate node. However, a more efficient solution is to perform a loop over each edge of the network, compute separately the net flow through that edge over all s-t-paths, and to add this net flow to the betweenness score of the starting node of the edge. This change in looping strategy improves the complexity by a factor from $O(n^2)$ to $O(m)$, which makes a big difference when dealing with a sparse network. The faster computation can be achieved by writing out matrix $\mathbf{N}^{ij}$, whose element (s,t) is $\bar{\eta}_{ij}(s,t)$ from Equation (5.22):

$$\begin{aligned} \left[\mathbf{N}^{ij}\right]_{st} &= \bar{\eta}_{ij}(s,t) = \left(\frac{z_{si}z_{jt}}{z_{st}} - \frac{z_{ti}z_{jt}}{z_{tt}}\right) w_{ij} \\ &= \left[\mathbf{z}_i^{\mathrm{c}}(\mathbf{z}_j^{\mathrm{r}})^{\mathsf{T}} \div \mathbf{Z} - \mathbf{e}\big(\mathbf{z}_i^{\mathrm{c}} \circ \mathbf{z}_j^{\mathrm{r}} \div \mathbf{diag}(\mathbf{Z})\big)^{\mathsf{T}}\right]_{st} w_{ij}, \end{aligned} \tag{5.27}$$

where $\circ$ and $\div$ denote elementwise matrix multiplication and division, respectively, $\mathbf{z}_i^{\mathrm{c}}$ and $\mathbf{z}_j^{\mathrm{r}}$ denote the $(n \times 1)$-vectors corresponding, respectively, to the i-th column and j-th row of matrix $\mathbf{Z}$ and $\mathbf{e}$ is the $(n \times 1)$-vector whose all elements are 1.

The overall net flow through edge (i,j) can then be computed as

$$\bar{\eta}_{ij}^{\mathrm{net}} = \sum_{s,t=1}^{n} \left[\, |\mathbf{N}^{ij} - \mathbf{N}^{ji}| \,\right]_{st} = \mathbf{e}^{\mathsf{T}} |\mathbf{N}^{ij} - \mathbf{N}^{ji}| \mathbf{e} \tag{5.28}$$

after which the betweenness score of each node can be simply computed by summing up the contributions of each edge connected to the node:

$$\mathrm{bet}_i^{\mathrm{net}} = \sum_{j:(i,j)\in E} \bar{\eta}_{ij}^{\mathrm{net}}.$$

Algorithm 3 contains the pseudocode for computing the RSP net betweenness. In principle, the algorithm can be used with directed graphs and the result can be interpreted according to the net flow of random walks, even though the electric current interpretation only makes sense with undirected graphs. With undirected graphs, Algorithm 3 should be altered so that it considers each

Algorithm 3 The RSP net betweenness vector of a graph G.

Input:
Same as Algorithm 2.
Output:
– The $n \times 1$ RSP net betweenness vector $\mathbf{bet}^{\mathrm{RSPnet}}$.

1. $\mathbf{W} \leftarrow \mathbf{P}^{\mathrm{rw}} \circ \exp[-\beta \mathbf{C}]$
2. $\mathbf{Z} \leftarrow (\mathbf{I} - \mathbf{W})^{-1}$
3. $\mathbf{bet}^{\mathrm{RSPnet}} \leftarrow \mathbf{0}$
4. **for all** $(i, j) \in E$ **do**
5. $\quad \mathbf{z}_i^{\mathrm{c}} \leftarrow \mathbf{Z}\mathbf{e}_i$, $\mathbf{z}_i^{\mathrm{r}} \leftarrow \mathbf{Z}^{\mathsf{T}}\mathbf{e}_i$
6. $\quad \mathbf{z}_j^{\mathrm{c}} \leftarrow \mathbf{Z}\mathbf{e}_j$, $\mathbf{z}_j^{\mathrm{r}} \leftarrow \mathbf{Z}^{\mathsf{T}}\mathbf{e}_j$
7. $\quad \mathbf{N}^{ij} \leftarrow w_{ij}\Big[\big(\mathbf{z}_i^{\mathrm{c}}(\mathbf{z}_j^{\mathrm{r}})^{\mathsf{T}} \div \mathbf{Z}\big) - \mathbf{e}\big((\mathbf{z}_i^{\mathrm{c}} \circ \mathbf{z}_j^{\mathrm{r}}) \div \mathbf{diag}(\mathbf{Z})\big)^{\mathsf{T}}\Big]$
8. $\quad \mathbf{N}^{ji} \leftarrow w_{ji}\Big[\big(\mathbf{z}_j^{\mathrm{c}}(\mathbf{z}_i^{\mathrm{r}})^{\mathsf{T}} \div \mathbf{Z}\big) - \mathbf{e}\big((\mathbf{z}_j^{\mathrm{c}} \circ \mathbf{z}_i^{\mathrm{r}}) \div \mathbf{diag}(\mathbf{Z})\big)^{\mathsf{T}}\Big]$
9. $\quad \mathrm{bet}_i^{\mathrm{net}} \leftarrow \mathrm{bet}_i^{\mathrm{net}} + \mathbf{e}^{\mathsf{T}}\big|\mathbf{N}^{ij} - \mathbf{N}^{ji}\big|\mathbf{e}$
10. $\quad \mathrm{bet}_j^{\mathrm{net}} \leftarrow \mathrm{bet}_j^{\mathrm{net}} + \mathbf{e}^{\mathsf{T}}\big|\mathbf{N}^{ij} - \mathbf{N}^{ji}\big|\mathbf{e}$
11. **end for**

undirected edge only once and increments also the betweenness of node j in addition to node i at step 10. This reduces the computation time on undirected networks by a half. Algorithm 3 also contains the same matrix inversion as Algorithm 2 of time complexity $\mathcal{O}(n^3)$. In addition to this, the other consuming task is the loop over all m edges of the graph in steps 4.-11., inside which elementary matrix operations of time complexity $\mathcal{O}(n^2)$ have to be performed. Thus, the total time complexity of the algorithm is $\mathcal{O}(n^3+mn^2)$. Computing the RSP net betweenness scores for one value of β on the Manhattan street network, studied in Section 5.4.2, takes roughly 20 minutes on the same computer setting as mentioned in the end of Section 5.3.1, thus almost 500 times as long as the computation of the simple RSP betweenness scores on this network.

Although the RSP net betweenness can be computed for directed graphs, we have not found a good use case for this purpose. The definition on directed graphs is not as intuitive as the simple betweenness and, moreover, current flows and the current flow betweenness are normally defined only for undirected graphs. Nevertheless, it is possible to use Algorithm 3 with directed graphs.

RSP betweenness centralities at the limit $\beta \longrightarrow \infty$

The simple RSP betweenness centrality counts the expected number of visits to each node during RSPs between all s-t-pairs of the graph. In the low temperature limit, i.e. when $\beta \to \infty$, the RSP probability distribution of Equation (2.10) converges to the shortest path likelihood distribution of Equation (2.11) focusing solely on the shortest paths of the graph. Thus, the simple RSP betweenness converges to the shortest path likelihood betweenness defined in Section 5.2.1.

The same result holds for the RSP net betweenness. Intuitively, as the path distribution focuses more and more on the shortest paths, one of the two terms in the net flow in Equation (5.26) becomes zero, as the walker will only move in one direction along each edge for a given s-t-pair. Thus, the RSP net betweenness also approaches the shortest path likelihood betweenness, as $\beta \longrightarrow \infty$.

RSP betweenness centralities at the limit $\beta \to 0^+$

In the high temperature limit, as $\beta \longrightarrow 0^+$, $\exp(-\beta\tilde{c}(\wp)) \longrightarrow 1$ for all $\wp$, and the RSP probabilities of Equation (2.10) converge to the unbiased random walk probabilities, determined by the reference transition probabilities, i.e. $\widetilde{\mathrm{P}}_{st} \longrightarrow \widetilde{\mathrm{P}}^{\mathrm{ref}}_{st}$. This means that the simple RSP betweenness converges to the expected number of visits to a node over all hitting paths with respect to the unbiased random walk probabilities. As presented in Section 5.2.2, this measure is proportional to the stationary distribution, if the network is strongly connected, where the multiplicative factor is the sum of expected hitting times between all s-t-pairs. Thus, as mentioned in Section 5.2.2, for undirected networks, the measure becomes proportional to the degree (or strength) centrality. In the same limit, on undirected graphs, as $\beta \longrightarrow 0^+$, the RSP net flow converges to the current flow betweenness centrality [157, 36], as the net edge flows $\bar{\eta}_{ij}(s,t)$ converge to the absolute electric current flows.

5.4 Experiments

The interpolation between common centrality measures already makes the simple RSP and RSP net betweenness centralities interesting. Furthermore, there are also cases in which the RSP betweenness centralities can be more relevant than their limit functions, the shortest path likelihood betweenness, the current flow betweenness and the stationary distribution. In this section these benefits will be illustrated, first with artificially generated networks, and later with two real networks of very different nature. The artificial examples

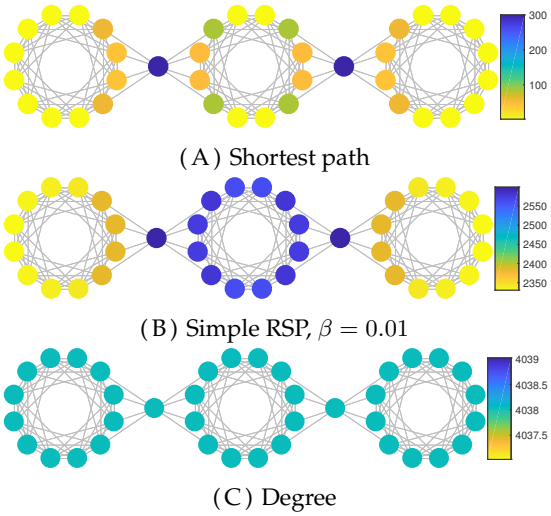
show the behavior of the RSP betweenness measures in a network consisting of communities. The first real network is the street network of Lower and Midtown Manhattan, which serves as an example of an undirected network. The second is a subset of the directed Wikipedia hyperlink network. All of the examples indicate benefits of using the RSP betweenness measures over the shortest path and random walk based betweenness measures.

5.4.1 Overall betweenness of an in-between community

One possible use for the RSP betweenness measures is the detection of groups of nodes that are central in a network. Consider a network consisting of three disjoint communities, A, B and C which are highly intraconnected but loosely interconnected. In addition, community B is connected to communities A and C, which, however, do not share any edges with each other. In other words, community B is in between communities A and C and all paths between nodes of communities A and C have to go through community B. Such an organization is possible, for instance, in a hierarchical social network, where community B could represent a directoral board of a company. If, moreover, the graph is in general sufficiently sparse, then the shortest paths between communities A and C will run through only a few of the nodes of community B. Thus, betweenness measures based on shortest paths will only highlight those nodes of community B, whereas the other nodes of community B will get no contribution from the connections between communities A and C. For some applications, however, the nodes of community B should, in general, be considered more in-between than the nodes of communities A and C. Defining betweenness based on random walks, and especially RSPs can help in this matter, as will be demonstrated next.

We first consider a simple example of a regular graph with communities organized in the order described above. The example is depicted in Figure 5.1 which shows an 8-regular graph containing three communities of 12 nodes, two of which are connected to the third one through separate bridge nodes. The figure contains the heat plots of the betweenness values of the nodes with the simple RSP betweenness (Figure 5.1B) and its limit functions, i.e. the shortest path likelihood betweenness (Figure 5.1A; which in this example equals the standard shortest path betweenness) and the stationary distribution (which, as the network is undirected, corresponds to the degree centrality; Figure 5.1C).

The heat plots show that the simple RSP betweenness highlights the nodes of the central community more than its limit functions. The low temperature limit function, i.e. the shortest path likelihood betweenness (which in this case also equals the standard shortest path betweenness) highlights the bridge



(A) Shortest path

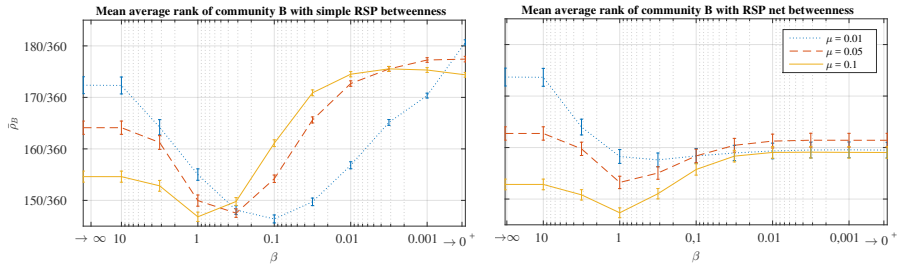
(B) Simple RSP, $\beta = 0.01$

(C) Degree

FIGURE 5.1: An 8-regular graph with three communities, and the heat plots of its betweenness values with the shortest path likelihood betweenness (A), the simple RSP betweenness (B) and the degree centrality (C). Blue indicates low, and red high betweenneess values.

nodes, but the betweenness scores of the nodes in the central community are of the same magnitude as the scores of the nodes in the peripheral communities. In the high temperature limit the simple RSP betweenness converges to the degree centrality, which is constant for all nodes, as the graph is regular. Although the heat plots show the actual betweenness scores, and not the rankings of the nodes according to them, the rankings also comply with the above findings. Using the RSP net betweenness (for which the results are not illustrated here), however, brings no benefit in this example, when compared to its limit functions. Namely, the current flow betweenness ranks the nodes of the central community a bit higher than the shortest path likelihood betweenness, but the RSP net betweenness does not increase those ranks with any intermediate values of β. On the other hand, the current flow betweenness values of the central community nodes are relatively much lower than the simple RSP betweenness values of Figure 5.1B. Moreover, the RSP net betweenness, can be beneficial in a similar setting, but with a bit of more complexity, which will be shown next.

For a more complex example, we generate random networks with the LFR algorithm of Lancichinetti et al. [127] designed to construct scale-free networks



(A) (B)

FIGURE 5.2: The mean average rank of the nodes of a community lying in between two other communities based on their RSP betweenness (A) and RSP net betweenness values (B) over 200 networks of 360 nodes generated using the LFR algorithm, as described in the body text (with low rank meaning a high betweenness score). The results are plotted for varying values of β and for three values of the mixing parameter μ with error bars indicating the standard error of the mean over the 200 runs. In both plots, the values at the left end of the curves (as $\beta \to \infty$) show the results with the shortest path likelihood betweenness and at the right end (as $\beta \to 0^+$) the results with the degree centrality in (A), and the current flow betweenness in (B).

with a community structure. This experiment confirms further the usefulness of the simple RSP, as well as the RSP net betweenness measure. We generated graphs consisting of three communities, A, B and C, and then simply removed the edges between two of the communities A and C. The size of the communities was set to 120 nodes, resulting in networks with 360 nodes, the average degree of the network was set to 10, the maximum degree to 120 and the power-law exponent of the degree distribution to -2. We tested three different values of the mixing coefficient $\mu = \{0.01, 0.05, 0.1\}$, which essentially determines the probability of having an edge between two communities after the degrees of nodes have been fixed. For each generated network, we computed the shortest path likelihood betweenness, the degree centrality, the current flow betweenness as well as the simple RSP and RSP net betweenness scores with several different values of the parameter β. We then rank the nodes according to each list of betweenness scores (with rank 1 naturally meaning the node with the highest score) and compute the average rank of the nodes in the central community B. We repeat the graph generation and the computation of average ranks 200 times and report the mean average rank of the nodes in the in-between community over these 200 runs.

The results are plotted in Figure 5.2, which shows that in most cases both the simple RSP and the RSP net betweenness, with some intermediate values of β, rank the nodes of the in-between community more central than their limiting functions. The plots show the mean average rank, as well as the standard error of the mean, of the nodes of the central community B with different values of the inverse temperature β. The values at the extremes of the plots correspond to the results obtained with the limiting functions, with the left end corresponding to the low-temperature (high β) case, i.e. the shortest path likelihood betweenness in both plots, and the right end to the high-temperature (low β) case, i.e. the degree centrality in Figure 5.2A, and the current flow betweenness Figure 5.2B. It is evident from the figures that with some intermediate value of β the RSP betweenness measures in this setting often manage to rank the nodes of the central group B higher than the limit functions by taking into account other connections besides only the shortest ones. Note that this does not mean that the RSP betweenness rankings are lower than the rankings with the limit functions on each individual network, but that the result holds on average over the 200 generated networks. The fluctuations of the rankings are indicated by the error bars showing the standard error of the mean over the 200 networks.

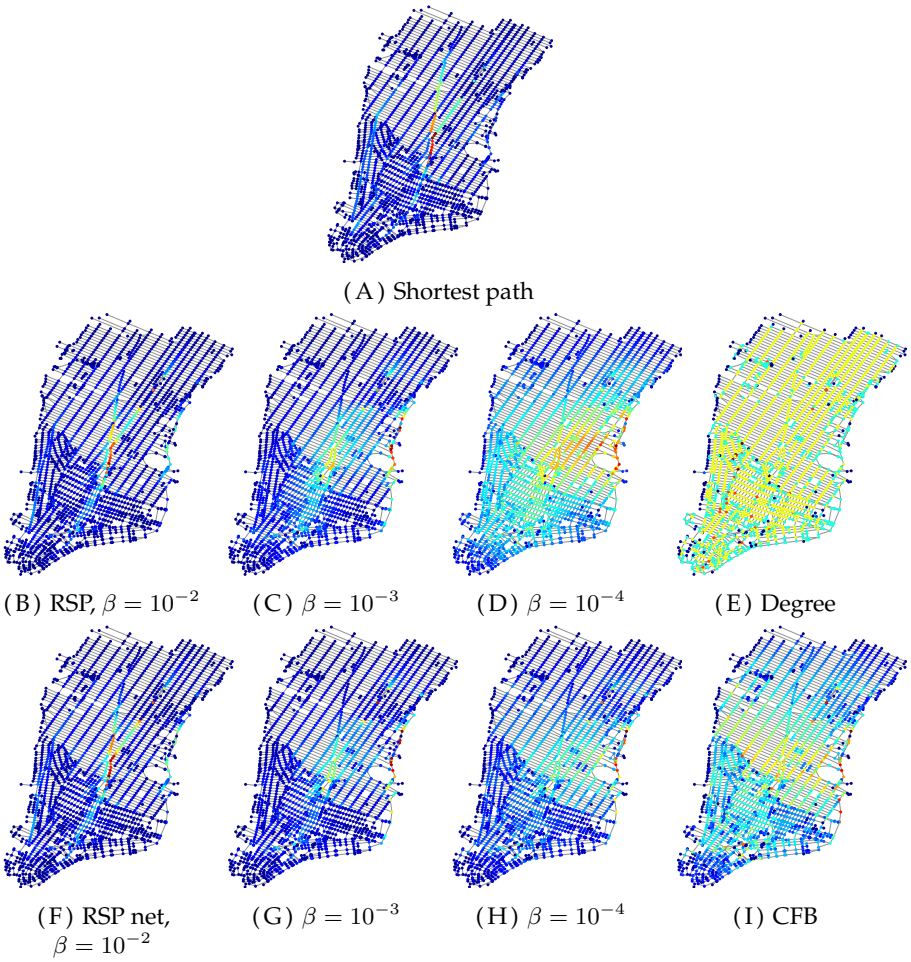
5.4.2 Manhattan street network

One promising application area for RSPs are routing or path planning problems. RSPs allow the modeling of routing in situations that include an element of randomness, such as navigation of people or animals in an environment. On the other hand, by definition, RSPs can be used for planning paths in an optimal way while keeping the predictability of the path at a desired level. RSPs could also be used for avoiding congestion problems in transportation and traffic networks.

We illustrate the use of RSPs for routing in a network by analyzing the street network of Midtown and Lower Manhattan. We have extracted this network from the Open Street Maps[2] website. The nodes in the network correspond to intersections and the edges are the street segments between the intersections. We treat the network as undirected, and analyze the network using both the simple RSP and the RSP net betweenness.

The length of each street segment is assigned as the cost of the corresponding edge. Accordingly, the overall cost of a path is its overall length. However, we define here the reference transition probabilities of the random walk according to the degree of each node, $p_{ij}^{\mathrm{ref}} = 1/d_i$, i.e. only according to the number of

[2]http://www.openstreetmap.org/



(A) Shortest path

(B) RSP, $\beta = 10^{-2}$ (C) $\beta = 10^{-3}$ (D) $\beta = 10^{-4}$ (E) Degree

(F) RSP net, $\beta = 10^{-2}$ (G) $\beta = 10^{-3}$ (H) $\beta = 10^{-4}$ (I) CFB

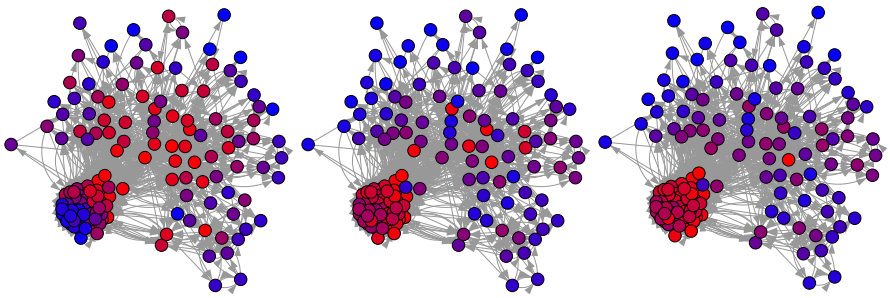
FIGURE 5.3: The interpolation between the shortest path and random walk betweenness measures with the simple RSP and the RSP net betwenness measures on the Midtown and Lower Manhattan street network.

edges connected to the node and independent of the edge costs. This seems like a reasonable choice for moving in a street network, if we consider that the decision of direction of the random walker in an intersection is not affected by the lengths of the street segments. Remember that this means that the shortest path likelihood betweenness on the graph is based on the edge costs, whereas the random walk based betweenness measures do not depend on the costs, but only the degrees of nodes.

The heat plots of the simple RSP and the RSP net betweenness measures and their limit functions on the Manhattan street network are depicted in Figure 5.3. At low temperatures (large β), both RSP based betweenness measures converge to the shortest path likelihood betweenness, shown in Figure 5.3A. Figures 5.3B, 5.3C and 5.3D show the betweenness values obtained with the simple RSP betweenness, and Figures 5.3F, 5.3G and 5.3H the values obtained with the RSP net betweenness with different values of β. As the network is undirected and connected, and because we use reference probabilities based only on degrees, instead of costs, the limit function of the simple RSP betweenness, when $\beta \longrightarrow 0^+$, is equal to the degree centrality multiplied by a constant, shown in Figure 5.3E. Finally, the limit function of the RSP net betweenness is the current flow betweenness, which is presented in Figure 5.3I.

This example shows one strength of both of the RSP betweenness measures. It is evident from the plot that the shortest path betweenness is focused on Broadway, which functions as a diagonal shortcut in many routes on the grid-like Midtown. However, when β is decreased, both RSP based measures rank highest the intersections along the FDR Drive on the eastern shore. This is mainly caused by the sparsity of streets on the east shore close to the residential areas of Stuyvesant Town and Peter Cooper Village. As a result, the FDR Drive is a vital connection between the upper and lower eastern parts of the map. This aspect is not clear from the shortest path likelihood betweenness, but becomes apparent by computing the RSP-based betweenness values. The current flow betweenness also ranks high the intersections of FDR Drive, but as a drawback the importance of Broadway is not as apparent as it perhaps deserves.

Thus, it seems that the RSP based betweenness measures, by assuming sub-optimal navigation between points in the network, can highlight bottlenecks such as the FDR Drive on the Manhattan network better than the deterministic shortest path betweenness or the unbiased current flow betweenness. Comparing between the two RSP betweenness, it is hard to find any major differences. However, in a street network, when considering the movement of people or vehicles, the simple RSP betweenness has a more sensible physical interpretation than the net betweenness. Moreover, again, the computation of the simple RSP betweenness is much more efficient and its interpretation clearer than the



(A) Shortest path

1. Network science
2. Network theory
3. Social network
4. Graph theory
5. Social network analysis
6. PageRank
7. Small-world network
8. Sociology
9. Small world experiment
10. Orkut

(B) RSP, $\beta = 10^{-1}$

1. Graph theory
2. Social network
3. Social network analysis
4. Mathematics
5. Sociology
6. Network theory
7. Computer science
8. Social networking service
9. PageRank
10. Network science

(C) Stationary distribution

1. Graph theory
2. Social network
3. Social networking service
4. Myspace
5. Social network analysis
6. Pinterest
7. Orkut
8. Small world experiment
9. Tumblr
10. Soc. network analysis software

FIGURE 5.4: The interpolation between the shortest path likelihood betweenness the stationary distribution with the simple RSP betwenness on the subnetwork of Wikipedia. The color of a node in the plots indicates its rank w.r.t. the betwenness measure; red and blue indicate high and low rank (i.e. high and low betweenness values), respectively. The dense cluster of nodes in the lower left corner consists mostly of nodes related to social networks. Below the plots, the 10 highest-ranked nodes are listed.

net betweenness, which make it a more preferable candidate. On the other hand, there may also appear applications, for which the net flow interpretation is more relevant, in which case the net betweenness should be used.

5.4.3 A subnetwork of Wikipedia

Here we illustrate the behavior of the simple RSP betweenness on a directed real network. The network is a subnetwork of the hyperlink network of Wikipedia. It consists of the Wikipedia page on Network Science and the pages that contain a hyperlink to it, or are linked to from it. We only consider the largest strongly connected component of this network which contains 151 nodes. We only report the results obtained with the simple RSP betweenness, and not the RSP net betweenness, because the net flow interpretation is not particularly

suitable when studying the World Wide Web, and in this example it does not provide any interesting results.

Figure 5.4 shows the change in the rankings of the nodes of the Wikipedia subnetwork with the simple RSP betweenness centrality as well as with its limiting functions, the shortest path likelihood betweenness and the stationary distribution. In addition to the heat plots, there are lists of the top ten nodes according to each betweenness centrality. The plots directly illustrate the general structure of the network, with a division to a tightly intra-connected cluster, appearing on the lower left corner of the plots, and a more sparsely connected peripheral part. The dense cluster comprises of nodes related mostly to social networks, whereas the other nodes correspond to more general concepts.

The shortest path likelihood betweenness ranks quite high many nodes from both of the two groups of nodes, i.e. general nodes such as the seed node 'Network science', whereas the stationary distribution focuses mostly on the dense cluster of nodes related to social networks, highlighting especially particular social networking websites, such as 'Myspace' and 'Pinterest'. Interestingly, even the seed node 'Network science' is not on the top ten nodes according to the stationary distribution. Here, again, the simple RSP betweenness makes an interesting compromise between these two extremes by respecting the high connetivity of the social networks cluster while also highlighting important, general nodes, such as 'Mathematics' and 'Computer Science' from the peripheral group.

This example indicates the potential of the simple RSP betweenness in analyzing and exploiting semantic and associative networks, as in [116]. Moreover, the example can help in applications of web design and marketing, for instance, considering situations where a user tries to find a certain page on a web site by following hyperlinks of that site. This can happen, for example by browsing videos on Youtube[3], or when playing the Wiki Game[4], in which the purpose is to find a target Wikipedia page from a starting page only by clicking the hyperlinks on the pages.

5.5 Discussion

We have presented two new graph node betweenness centrality measures based on Randomized Shortest Paths. The first measure, the simple RSP betweenness centrality, counts the expected number of visits to a node, while the second, the RSP net betweenness, is based on the overall net flow over edges

[3] http://www.youtube.com/
[4] http://www.thewikigame.com/

connected to a node. Both of these measures are parametrized generalizations of more traditional betweenness centrality measures. The RSP net betweenness and its high-temperature limit function, the current flow betweenness seem theoretically more elaborate than the simple RSP betweenness and its limit function, the stationary distribution. However, based on our experiments, the simple RSP betweenness seems to provide more satisfying and practical results than the net betweenness, in addition to which (and perhaps partly because) it is easier to interpret. In general, our experiments have shown that the RSP betweenness centralities can provide interesting insight into the role and importance of the nodes in a network in ways that the more traditional betweenness measures based on either shortest paths or unbiased random walks can not achieve.

The RSP betweenness measures could be further compared with other centrality measures, which could be a subject for future work. However, the main purpose of this chapter is only to focus on betweenness centrality measures, to introduce the RSP based methods and their computation and to provide some examples where one could benefit from using them. Also, the chapter only considers betweenness as a global measure on nodes, but the methods can easily be extended to other uses such as edge betweenness, betweenness w.r.t. a group of nodes or betweenness between groups of source and target nodes, all of which have relevant applications. One drawback of the RSP framework is that it often requires a full matrix inverse, because of which it is not currently practical for very large networks. One topic for future research is to develop methods that would allow estimating the RSP-based quantities for large networks either by using more specialized computational methods or by approximation.

Although the computational complexity of the RSP-based methods can be too high for very large problems, the implementation and the interpretation of the computations is quite straightforward. Moreover, the framework lies on solid theoretical grounds, and considering the generalization of shortest paths by randomization makes sense for many application scenarios. In our examples we have shown them to give promising results in highlighting nodes that belong to a central group of nodes, in detecting possible bottlenecks in street networks for navigation modeling and also in evaluating the visit rate of pages on the World Wide Web.

Chapter 6

Additional observations and results related to free energy on networks

6.1 Introduction

The concept of free energy, as it was introduced in Sections 2.4.6 and 3.3, has proven to be an extremely interesting and functional quantity for different graph and network analysis tasks. First, the free energy distance, Δ^{FE}, is appealing for several, both theoretical as well as practical reasons, including that

- it satisfies the triangle inequality and thus defines a metric, unlike the RSP dissimilarity, Δ^{RSP} (see Section 2.4.6 and [82]).
- it is cutpoint additive (see Section 2.4.6 and [82]).
- it has been shown in [103] (although with a different formalism) to avoid the lost-in-space phenomenon (see Section 2.4.5, and [142]).
- it has been shown to perform well in several applications, namely in clustering and semisupervised classification (in Chapter 3 and in [192, 110, 82]) as well as in link prediction [103].

On the other hand, part of the appeal of free energy comes from the fact that the RSP probability distribution of Equation (2.10) can as well be defined according to free energy minimization, as explained in Section 3.3. Moreover, based on the related work listed in Section 2.6, a lot of recent literature has concentrated on considering or defining measures that are similar, or even equivalent to the

free energy. The wide interest in similar studies already by iself suggests the idea of free energy to be a fundamental concept in network analysis.

However, the general definition of the free energy in Equation (2.46), or the expression in Equation (2.47) in the case of the RSP probability distribution of Equation (2.10), are not necessarily easy to interpret directly. Part of this chapter is dedicated to giving a more intuitive conception of the free energy ϕ, which is done by collecting a list of the different interpretations of free energy and the different contexts that it appears in. This will hopefully help in understanding the foundations of the methodology and partly explain the obtained good results in practical applications. In addition to providing intuition behind the free energy, this chapter also lists some more quantities, methods and discovered theoretical results that have not been stated earlier in the thesis, nor in the literature. This may open up new possibilities for applications for free energy, and clarify some issues that may arise when considering free energy in applications.

6.2 Computation of free energy

In this section, we present some more methods and measures that can be derived from the RSP framework, with special focus on quantities related to free energy, ϕ. In addition, we explain explicitly how free energies can be computed for only a limited set of s-t-pairs, and also deal with issues that sometimes emerge with free energy computations. The standard way of computation of free energies (as well as expected RSP costs) in the RSP framework is by solving systems of linear equations, as the quantities always depend on values of the fundamental matrix of non-hitting paths $\mathbf{Z}$ (see Equation (2.19)). However, sometimes these methods may fail due to numerical issues. In such cases the quantities can be, however, computed by an iterative scheme, which is presented in Section 6.2.2.

Free energies to a single target. Section 3.3 presented the method for computing free energies, ϕ, and free energy distances, Δ^{FE}, for all node-pairs on a graph. However, often distances are only needed between a limited set of nodes. For a single s-t-pair, the free energy of the RSP distribution over hitting s-t-paths is given by combining Equations (2.18) and (2.47) as

$$\phi_{st} = -\tfrac{1}{\beta} \log \mathcal{Z}_{st} = \tfrac{1}{\beta}(\log z_{tt} - \log z_{st},) \tag{6.1}$$

meaning that two elements of the matrix $\mathbf{Z}$ are required. However, instead of computing the whole matrix $\mathbf{Z}_{\mathrm{h}}$, as in Section 3.3, it is in this case simpler to compute the full column t of the matrix by solving the linear system

$$(\mathbf{I} - \mathbf{W})\mathbf{z}_t = \mathbf{e}_t \tag{6.2}$$

which provides the free energies from all source nodes $s \in V$ to t at once as a vector[1]

$$\Phi_t = -\tfrac{1}{\beta}(\log z_{tt} - \log \mathbf{z}_t). \tag{6.3}$$

Note, however, that computation of free energy *distances*, Δ^{FE}, for one target is more cumbersome, as it requires also the free energies in the other direction, from the target to all others. But computing free energies from a single source s to all targets $t \in V$ is more difficult, as that requires knowing the diagonal values z_{tt} for all t. Methods for computing the diagonal of a matrix inverse have been studied in [198].

Free energies to multiple targets. Sometimes an application might require computing the free energies from all source nodes to a given set of target nodes $\Omega = \{t_1, \ldots, t_M\} \subset V$. Then the matrix Φ_Ω of free energies from all source nodes to the target nodes in Ω can be computed by first solving, for $(n \times M)$-matrix $\mathbf{Z}_\Omega$,

$$(\mathbf{I} - \mathbf{W})\mathbf{Z}_\Omega = \mathbf{I}_\Omega, \tag{6.4}$$

where $\mathbf{I}_\Omega$ is the $(n \times M)$-matrix, whose i-th column, for all $i = 1, \ldots, M$, is $\mathbf{e}_{t_i}$, i.e. contains 1 on the t_i-th element and zeros elsewhere.

Define then the matrix $\mathbf{D}_\Omega$ so that its i-th diagonal element is element (i, t_i) of matrix $\mathbf{Z}_\Omega$. The *partial* fundamental matrix $\mathbf{Z}_{\Omega,\mathrm{h}}$ is then computed, similarly to the computation of the full fundamental matrix of hitting paths, $\mathbf{Z}_{\mathrm{h}}$, in Equation (2.21), as

$$\mathbf{Z}_{\Omega,\mathrm{h}} = \mathbf{Z}_\Omega \mathbf{D}_\Omega^{-1} \tag{6.5}$$

and, finally, the $(n \times M)$-matrix of free energies to the set of targets Ω is given by

$$\Phi_\Omega = -\frac{1}{\beta} \log \mathbf{Z}_{\Omega,\mathrm{h}}. \tag{6.6}$$

Again, the computation of free energy distances, Δ^{FE}, is more difficult, similarly to the reason explained in the single target case above. Note that when

[1] Note that vector Φ_t can be computed equivalently with the standard method for absorbing Markov chains (as in Section 2.3.2) by first solving $(\mathbf{I} - \widehat{\mathbf{W}})\hat{\mathbf{z}}_t = \mathbf{e}_t$, where $\widehat{\mathbf{W}}$ is obtained by setting row t of matrix $\mathbf{W}$ to zero, as then $\Phi_t = -\frac{1}{\beta} \log \hat{\mathbf{z}}_t$.

the set of target nodes is the whole node set, $\Omega = V$, then the computation coincides with the method presented in Section 3.3 for all-pairs free energy computation.

6.2.1 Computation of other quantities

We then list other meaningful quantities that can be computed from the RSP distribution that are relevant for this work. Many of them can be expressed based on derivatives of the partition function, as is common in statistical physics.

Biased random walk. Recall, from Section 5.3.1, that the expected number of traversals through edge (i, j) and the expected number of visits to node i over hitting s-t-paths w.r.t. the RSP distribution (2.10), can be computed, respectively, as (see Section 5.3.1 for details)

$$\bar{\eta}_{ij}(s,t) = \left(\frac{z_{si}}{z_{st}} - \frac{z_{ti}}{z_{tt}}\right) w_{ij} z_{jt} \quad \text{and} \quad \bar{\eta}_i(s,t) = \left(\frac{z_{si}}{z_{st}} - \frac{z_{ti}}{z_{tt}}\right) z_{it}. \tag{6.7}$$

Remember also that summing the latter quantity over all node pairs $(s,t) \in V \times V$ defines the simple RSP betweenness centrality of node i (again, see Section 5.3.1).

$$\mathrm{bet}_i^{\mathrm{RSP}} = \sum_{s,t=1}^{n} \bar{\eta}_i(s,t). \tag{6.8}$$

The expected visits over edges and nodes over paths from s to t can be used to determine a *biased random walk*, containing a drift towards the target node t. The *biased transition probabilities* of this new random walk, or Markov chain, are defined as

$$p_{ij}^{(\mathrm{t})} = \frac{\bar{\eta}_{ij}(s,t)}{\sum_{k \in V} \bar{\eta}_{ik}(s,t)} = \begin{cases} \dfrac{w_{ij} z_{jt}}{z_{it}}, & \text{if } i \neq t, \\ 0, & \text{if } i = t. \end{cases} \tag{6.9}$$

Note that this holds for any starting node s, with the restriction $s \neq t$. In other words, the biased transition probabilities are only dependent on the target node t and not on the starting node s. This explains the notation $p_{ij}^{(\mathrm{t})}$ containing only t, although (an arbitrary) s is used for the definition. The biased transition probabilities can be seen as a randomized policy driving the random walker towards the target, t.

It turns out, as explained in [182], that this biased Markov chain actually induces the RSP path probability distribution (2.10), meaning that for any

$\wp \in \mathcal{P}_{st}$,

$$\mathrm{P}_{st}^{\mathrm{RSP}}(\wp) = \prod_{(i,j)\in\wp} p_{ij}^{(t)}, \tag{6.10}$$

which can be confirmed easily by writing out the product and inserting Equation (6.9). Note however, that the fact that the RSP distribution can be reduced to the biased random walk probabilities is only a feature of that distribution, but does not hold generally; i.e. an arbitrary path distribution from s to t cannot be necessarily reduced to a first-order Markov chain.

Relative entropy. Recall from Equation (2.46) the definition of free energy as $\phi_{st} = \langle \tilde{c} \rangle_{st} + \frac{1}{\beta}\mathbb{J}(\mathrm{P}_{st}\|\mathrm{P}_{st}^{\mathrm{rw}})$ and from Section 3.2 that the expected RSP cost from s to t, $\langle \tilde{c} \rangle_{st}$, can be computed as

$$\langle \tilde{c} \rangle_{st} = -\frac{\partial}{\partial\beta} \log \mathcal{Z}_{st} = \sum_{(i,j)\in E} \left(\frac{z_{si}}{z_{st}} - \frac{z_{ti}}{z_{tt}} \right) w_{ij} c_{ij} z_{jt}. \tag{6.11}$$

Then, from Equation (2.46), the relative entropy of P_{st} can be computed after computing the free energy and the expected RSP cost as

$$\mathbb{J}(\mathrm{P}_{st}\|\mathrm{P}_{st}^{\mathrm{rw}}) = \beta(\phi_{st} - \langle \tilde{c} \rangle_{st}). \tag{6.12}$$

Using the derivative form in Equation (6.11), we obtain another interesting relation between the relative entropy and the free energy:

$$\mathbb{J}(\mathrm{P}_{st}\|\mathrm{P}_{st}^{\mathrm{rw}}) = -\beta^2 \frac{\partial \phi_{st}}{\partial \beta}. \tag{6.13}$$

6.2.2 Iterative computation avoiding numerical underflow

As stated in the beginning of Section 6.2, the computation of the free energy and other quantities may fail in some cases due to numerical issues. More precisely, this happens, when considering very high values of β, i.e. when the RSP distribution is focused mostly on the least cost paths. Such a situation is common, for instance, when trying to approximate or regularize least costs between nodes with free energies in very large graphs.

The problem is that the free energies converge to least costs very slowly, when $\beta \to \infty$. Meanwhile, the edge likelihoods become very small, as $w_{ij} = p_{ij}^{\mathrm{rw}} \exp(-\beta c_{ij}) \to 0$. As a result, the computation of matrix $\mathbf{Z}$ (see Equations (2.19) and (2.20)) can produce underflow causing zero elements which ultimately lead to infinite free energies or expected costs. This situation occurs, for

instance, on large geographic networks, such as the landscape networks considered in Chapters 7 and 8, or generally any networks that are not small-world networks.

However, for the free energies we are essentially interested in the logarithm of the elements of $\mathbf{Z}$. Fortunately, we can avoid the explicit computation of matrix $\mathbf{Z}$ and compute the logarithms inspired by dynamical programming, similar to the computation of least cost, or shortest paths with the Bellman-Ford algorithm. We first present this iterative algorithm for computing free energies, and then show how it can be used for computation of expected costs $\langle \tilde{c} \rangle_{st}$ in cases where the standard approach causes underflow issues.

These derivations appeared in [82], Appendix E, for proving that the free energy distance converges to the shortest path distance on undirected graphs, when $\beta \to \infty$. Here, we harness the idea further for solving the problem of computing free energies in the limit $\beta \to \infty$ on large graphs. The solution is based on the *log-sum-exp trick* used e.g. in machine learning (see e.g. [150]) and Hidden Markov modeling [107] for avoiding underflow.

Free energy. First, we refer to [82], Section 4.3, where it is shown that for partition functions of hitting paths to t, for any $s \neq t$, the following holds:

$$\mathcal{Z}_{st} = \sum_{j=1}^{n} w_{sj} \mathcal{Z}_{jt}. \tag{6.14}$$

In addition, from Equation (2.47) we can write for any $j \in V$, $\mathcal{Z}_{jt} = \exp(-\beta \phi_{jt})$. Combining this with Equation (6.14) and the definition of the edge likelihood w_{ij}, we can further write

$$\phi_{st} = -\tfrac{1}{\beta} \log \sum_{j=1}^{n} p^{\mathrm{rw}}_{sj} \exp(-\beta(c_{sj} + \phi_{jt})), \tag{6.15}$$

which provides an iterative process for computing free energies, similar to the iteration of the Bellman-Ford algorithm for computing shortest paths on a graph [188]. More specifically, the difference is that the minimum operator in the standard Bellman-Ford algorithm is replaced by the *soft minimum* operator, which is discussed more in Section 6.3.2.

The above iteration can be used for computing the free energies from all source nodes s to one target node t, and a similar development can be made for computing all free energies from one source s to all targets t. However, the numerical underflow issue faced with the standard algorithm of solving linear

systems still resides. Namely, for large β, the exponentials in the sum appearing in Equation (6.15) may cause underflow issues, causing the exponential to be evaluated as zero which in turn can result in an infinite value for the free energy. This is where the log-sum-exp trick comes to use. First, we denote for any $s, j \in V$,

$$x_{sjt} = c_{sj} + \phi_{jt}. \tag{6.16}$$

Moreover, we denote the minimum of the above quantity, over all successors of s, i.e. over all nodes j such that $(s, j) \in E$, as

$$x^*_{st} = \min_{j \in Succ(s)} \{c_{sj} + \phi_{jt}\}. \tag{6.17}$$

We can then extract the term $\exp(-\beta x^*_{st})$ out of the sum in (6.15) and rewrite the iteration as

$$\phi_{st} = -\frac{1}{\beta} \log \left(\exp\big(-\beta x^*_{sj}\big) \sum_{j=1}^{n} p^{\mathrm{rw}}_{sj} \exp(-\beta(x_{sjt} - x^*_{st})) \right) \tag{6.18}$$

$$= x^*_{st} - \frac{1}{\beta} \log \sum_{j=1}^{n} p^{\mathrm{rw}}_{sj} \exp(-\beta(x_{sjt} - x^*_{st})). \tag{6.19}$$

The above iteration can be used to compute the vector Φ_t of free energies from all nodes to one target node t at once. The iteration can be initialized e.g. by setting the elements ϕ_{st} to some large value for all $s \neq t$ and $\phi_{tt} = 0$. Unfortunately, this approach cannot be applied for computing all-pairs free energies, but distances to each node have to be computed separately.

Expected cost. Although the convergence rate of the expected cost towards the least cost, with $\beta \longrightarrow \infty$, is much faster than with free energy, the iterative algorithm can still come in necessary, when dealing with large graphs. The problem is the same as when computing free energies – the computation of matrix $\mathbf{Z}$ or its vectors may result in underflow. Unfortunately, the expected costs rely on the actual values of matrix $\mathbf{Z}$, instead of the logarithms. However, we now show that there is another way to compute the expected costs directly without computing the values of $\mathbf{Z}$ explicitly.

Namely, from the vector $\mathbf{\Phi}_t$ of free energies to t, we can obtain the biased transition probabilities to t, from Equation (6.9), that induce the RSP path probabilities (see Section 6.2.1) from any source s to t. Remembering that

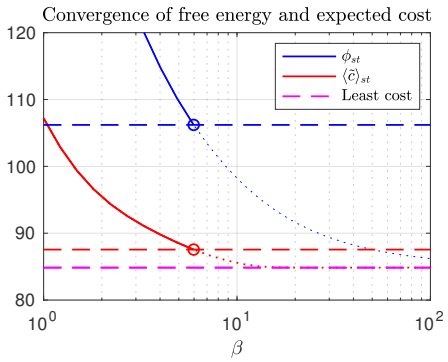


FIGURE 6.1: The plot showing the convergence of ϕ_{st} and $\langle\tilde{c}\rangle_{st}$, to the least cost in function of β. The dashed blue and red lines show the points where the standard procedure for computing ϕ_{st} and $\langle\tilde{c}\rangle_{st}$, respectively, fails because of numerical underflow.

$\mathcal{Z}_{st} = z_{st}/z_{tt}$, from Equation (6.9) we get

$$\begin{aligned} p_{ij}^{(\mathrm{t})} &= \frac{z_{jt}}{z_{it}} w_{ij} = \frac{\mathcal{Z}_{jt}}{\mathcal{Z}_{it}} w_{ij} = \exp(-\beta(\phi_{jt} - \phi_{it}))w_{ij} \\ &= p_{ij}^{\mathrm{rw}} \exp(-\beta(c_{ij} + \phi_{jt} - \phi_{it})). \end{aligned} \tag{6.20}$$

These can be computed avoiding the underflow issue for any t. The expected cost from any $s \in V$ to t can then be computed by the standard iterative form for computing expected costs (see, e.g. [77]):

$$\langle\tilde{c}\rangle_{st} = \begin{cases} \sum_{j=1}^{n} p_{sj}^{(t)}(c_{sj} + \langle\tilde{c}\rangle_{jt}), & \text{for } s \neq t \\ 0, & \text{for } s = t. \end{cases} \tag{6.21}$$

Therefore, computing expected costs to t with large β on a large graph without underflow problems requires two iterations: first the computation of free energies to t, which provides biased transition probabilities, and then the iteration for computing expected costs with those biased probabilities.

We demonstrate the behavior of the results obtained with the two algorithms in Figure 6.1. We consider a 61×61 square grid graph where each node is connected to its vertical, horizontal, and diagonal neighbors. The edge costs are set according to the distance between the nodes, i.e. $c_{ij} = 1$ for vertical and horizontal edges, and $c_{ij} = \sqrt{2}$ for diagonal edges. We then compute the free energy ϕ_{st} and the RSP expected cost $\langle\tilde{c}\rangle_{st}$ from one corner of the

grid (node s) to the opposite corner (node t) for a range of values of β (on the logarithmic x-axis). The least cost from s to t, along the diagonal of the grid, is $60 \times \sqrt{2} \approx 84.853$, shown in Figure 6.1 by the dashed magenta line. The blue and red round points and dashed lines show the point where the standard algorithms (based on solving linear systems) for computing ϕ_{st} and $\langle \tilde{c} \rangle_{st}$, respectively, fails because of numerical underflow (with double precision floating numbers). The dotted blue and red lines then show values obtained with the iterative algorithms, presented in this section, based on the log-sum-exp trick. The convergence of the dotted lines shows that these procedures can be used to obtain free energy and expected cost values arbitrarily close to the least cost.

6.3 Different interpretations of free energy

So far, the RSP framework has been defined based on two approaches. In Section 2.3, the definition was given based on minimization of expected cost with a relative entropy constraint, while in Section 3.3 a definition was stated based on free energy minimization. In this section we present a list of other interpretations and different ways of arriving at the definition of the RSP distribution and free energy, as these different approaches can prove to be more relevant for particular applications.

6.3.1 According to killed random walks

As explained in Section 2.3, the RSP model can be also interpreted in terms of *killed random walks*. The killed random walk can be defined from the substochastic matrix $\mathbf{W}$, where the residue probability at each node i, $1 - \sum_j w_{ij}$, can be thought of as the probability of transition from i to an imaginary, absorbing "cemetery node". Based on this, the partition function, $\mathcal{Z}_{st}$, can be interpreted as the *survival probability*, i.e. the probability of reaching t from s instead of being killed during the killed random walk, i.e. before ending up in the cemetery node..

Taking minus the logarithm of a probability is known as a measure of *surprisal* of the corresponding event. To obtain the free energy, the surprisal is furthermore divided by the inverse temperature.

This interpretation can be valuable, for instance, for epidemic spreading applications, as the survival probability of walks relates strongly to the probability of contagion of a disease. For instance, in [108] the authors propose a measure

for estimating the contagion times in epidemic spreading. Essentially, their measure is similar to the scaled free energy $\beta\phi = -\log \mathcal{Z}_{st}$, where the spreading rate of the disease has a similar effect to the temperature $T = 1/\beta$ in the RSP model. In other words, a disease with a high spreading rate corresponds to the RSP model with high temperature (low β), which in turn corresponds to short contagion times (i.e. $\beta\phi \to 0$, as $\beta \to 0^+$), whereas a low spreading rate corresponds to long contagion times (i.e. $\beta\phi \to \infty$, as $\beta \to \infty$).

6.3.2 Weighted soft minimum or log-sum-exp

Another interpretation of ϕ is obtained in relation to the *soft minimum* function (more often considered as the soft maximum), sometimes also called the *log-sum-exp* function in the literature. The soft minimum is traditionally defined for a dataset (i.e. a multiset) of numbers $X = \{x_1, \dots, x_n\}$ in the form

$$\operatorname{softmin}(X|\beta) = -\frac{1}{\beta} \log \left(\sum_{i=1}^{n} \exp(-\beta x_i) \right). \tag{6.22}$$

It is often used in optimisation as a proxy for evaluating a minimum, as when the parameter $\beta \to \infty$, the function converges to the true, hard minimum [49, 32]. Its convenience lies in the fact that it is differentiable, and thus easier to minimize than the hard minimum.

However, the traditional form of the soft minimum in fact underestimates the minimum, i.e. we have, for all β, $\operatorname{softmin}(X \mid \beta) \leq \min(X)$, which can sometimes be considered undesirable. On the other hand, the logarithmic form of the free energy in Equation (2.47) is slightly different from the standard soft minimum expression in (6.22). For comparison, by writing out the expression for the partition function $\mathcal{Z}_{st}$, we rewrite the free energy as

$$\phi = -\frac{1}{\beta} \log \left(\sum_{\wp \in \mathcal{P}_{st}} \mathrm{P}_{st}^{\mathrm{rw}}(\wp) \exp(-\beta \tilde{c}(\wp)) \right). \tag{6.23}$$

From this we see that the difference between the free energy and the soft minimum is that the free energy contains also the reference probabilities as weights of the exponentials. Such weighting actually turns out very convenient, as this weighted soft minimum is always larger than the hard minimum.

In fact, if we consider, in place of the set of numbers X, any discrete random variable with non-negative outcomes, and use its probabilities as weights, then the weighted soft minimum interpolates between the minimum and the

expected value of that random variable. This result holds generally for any discrete random variable for which the expected value exists. It was proved for the free energy distance (i.e. that it interpolates between the least cost and the commute cost) in [82], by considering the cost of a path, $\tilde{c}(\wp)$ in place of the random variable x_i, and the reference hitting path probabilities as the weights (remember that the hitting path probabilities sum to one [82]). In addition, the result was shown, in a similar context to ours, in [196] to hold for a finite set of values with uniform, constant, weights. The nice thing about this interpretation of free energy is that it can be stated straightforwardly, without need for discussion of e.g. minimization problems, path distributions or temperature.

6.3.3 Free energy as a combined energy and information cost

The role of the relative entropy term, $\mathbb{J}(\mathrm{P}_{st}\|\mathrm{P}_{st}^{\mathrm{rw}})$, in the minimization problems (2.4) and (3.7) can be interpreted in many ways. An information theoretic viewpoint is that $\mathbb{J}(\mathrm{P}_{st}\|\mathrm{P}_{st}^{\mathrm{rw}})$ is the *reduction in average code length* required to describe a sequence of paths generated by the distribution P_{st}, when using optimal coding instead of using a coding based on the distribution $\mathrm{P}_{st}^{\mathrm{rw}}$ [150]. In other words, the relative entropy of P_{st} w.r.t. $\mathrm{P}_{st}^{\mathrm{rw}}$ can be interpreted as an *information cost* of the distribution P_{st} (as considered similarly in [171]).

Namely, for an agent to move according to an optimal path distribution requires it to possess global information of the graph structure in order to make "smart", decisions avoiding costs. The amount of this information is quantified by the relative entropy $\mathbb{J}(\mathrm{P}_{st}\|\mathrm{P}_{st}^{\mathrm{rw}})$. In contrast, movement similar to a random walk requires no global information, but is only based on the local transition probabilities. The relative entropy is then close to zero, meaning that the entropy of P_{st} is very close to the entropy of $\mathrm{P}_{st}^{\mathrm{rw}}$. In this case the average code length is fairly long, as more paths need to be encoded more likely. The coding length is minimizal for the distribution focusing only on shortest paths; especially, if there exists only one shortest path, that can be encoded by a single bit. Note that an even longer average code length, i.e. larger entropy, than the one given by the unbiased random walk distribution $\mathrm{P}_{st}^{\mathrm{rw}}$, would be given by the *maximum entropy random walk* [39, 53].

The above interpretation has similarities with research done in stochastic optimal control [205, 204, 112]. Especially [203] shows that many models proposed for stochastic optimal control problems can be expressed according to free energy minimization, where the relative entropy is also considered as an information, or control, cost.

We now approach the definition of the Gibbs-Boltzmann distribution from a slightly different angle. Namely, the above consideration of the relative

entropy as an information cost of the distribution suggests defining an energy functional that combines the standard expected cost and the information cost of a path distribution. A naive way for such combination is the direct sum

$$E_{st} = \langle \tilde{c} \rangle_{st} + \mathbb{J}(\mathrm{P}_{st} \| \mathrm{P}^{\mathrm{rw}}_{st}), \tag{6.24}$$

which we call the *effort* of going from s to t. We can then consider the distribution that minimizes the effort. Not surprisingly, the solution of the minimization is the RSP distribution of Equation (2.10) with $\beta = 1$.

However, one problem with the above definition of effort is that the units of measurement, and hence the scales, of the two costs are not the same. The expected cost, $\langle \tilde{c} \rangle_{st}$, can take any value, depending on the (unbounded) scale of the edge costs. The information cost, $\mathbb{J}(\mathrm{P}_{st} \| \mathrm{P}^{\mathrm{rw}}_{st})$, on the other hand, is always bounded based on the size and structure of the network. One solution for alleviating this is to multiply the edge costs with a scaling term β, i.e. to replace c_{ij} with βc_{ij}. This scaling of course also scales the path costs, i.e. $\tilde{c}(\wp)$ is replaced by $\beta\tilde{c}(\wp)$ and likewise $\langle \tilde{c} \rangle_{st}$ is replaced by $\langle \beta\tilde{c} \rangle_{st} = \beta \langle \tilde{c} \rangle_{st}$.

If we then rewrite the effort in Equation (6.24) considering the scaled edge costs, we obtain the scaled free energy:

$$\hat{E}_{st} = \beta \langle \tilde{c} \rangle_{st} + \mathbb{J}(\mathrm{P}_{st} \| \mathrm{P}^{\mathrm{rw}}_{st}) = \beta \phi_{st}. \tag{6.25}$$

Of course, the minimization of this quantity, again is equivalent to the minimization of free energy, and is solved by the RSP distribution (2.10). This gives yet another interpretation of the RSP distribution: it minimizes the sum of an information cost and the normal expected cost, after scaling the edge costs with β.

Determining the value of β. The interpretation presented above may be useful for determining a suitable value for β in some applications. As already mentioned, the relative entropy on a given graph between s and t is bounded. When the RSP distribution is considered, this bound can be easily estimated, as the maximum relative entropy (obtained when $\beta \to \infty$) equals the relative entropy obtained with the distribution that focuses only on the least cost paths from s to t.

Let us assume that there is only one least cost path from s to t, denoted as $\wp^*$, with cost $\tilde{c}(\wp^*)$. The limit distribution of the RSP distribution, when $\beta \to \infty$ is the distribution P^*_{st}, for which $\mathrm{P}^*_{st}(\wp^*) = 1$ and $\mathrm{P}^*_{st}(\wp) = 0$ for all $\wp \neq \wp^*$. And

the relative entropy of this distribution is given by

$$J^*_{st} = \sum_{\wp \in \mathcal{P}_{st}} \mathrm{P}^*_{st}(\wp) \log(\mathrm{P}^*_{st}/\mathrm{P}^{\mathrm{rw}}_{st}(\wp)) = \log(1/\mathrm{P}^{\mathrm{rw}}_{st}(\wp^*)) = -\sum_{(i,j)\in\wp^*} \log p^{\mathrm{rw}}_{ij}. \quad (6.26)$$

One could then choose the value of β by comparing the values $\tilde{c}(\wp^*)$ and J^*_{st}. For instance, if $\tilde{c}(\wp^*) \gg J^*_{st}$, then β should be set to a relatively low value in order to avoid the RSP distribution to focus solely on the least cost paths (at least of that node pair). Conversely, if $J^*_{st} \gg \tilde{c}(\wp^*)$, a large value of β would avoid the distribution to spread too much over random walks. One might also consider the ratio $\tilde{c}(\wp^*)/J^*_{st}$, or the mean of that ratio over all node-pairs on the graph as a factor when deciding on the value of β for a given application.

6.3.4 Expected augmented cost

Let us now consider P_{st} as any probability distribution over $\mathcal{P}_{st}$ that can be reduced to a first-order absorbing Markov chain on G with absorbing state t. In other words, let us consider an arbitrary set of Markov chain transition probabilities $p^{(t)}_{ij}$ on G such that $p^{(t)}_{tj} = 0$ for all $j \in V$, which generate path probabilities $\mathrm{P}_{st}(\wp) = \prod_{i=1}^{L(\wp)} p^{(t)}_{\wp(i-1),\wp(i)}$. Remember that in the case of the RSP distribution, the reduction to a Markov chain is given by the biased transition probabilities in Equation (6.31).

For any such distribution P_{st}, the free energy from s to t can be rewritten as

$$\begin{aligned}
\phi_{st} &= \sum_{\wp\in\mathcal{P}_{st}} \mathrm{P}_{st}(\wp)c(\wp) + T\sum_{\wp\in\mathcal{P}_{st}} \mathrm{P}_{st}(\wp)\log(\mathrm{P}_{st}(\wp)/\mathrm{P}^{\mathrm{rw}}_{st}(\wp)) \\
&= \sum_{\wp\in\mathcal{P}_{st}} \mathrm{P}_{st}(\wp)\left(\sum_{(i,j)\in\wp} c_{ij} + T\log\prod_{(i,j)\in\wp}(p^{(t)}_{ij}/p^{\mathrm{rw}}_{ij})\right) \\
&= \sum_{\wp\in\mathcal{P}_{st}} \mathrm{P}_{st}(\wp)\sum_{(i,j)\in\wp}\left(c_{ij} + T\log(p^{(t)}_{ij}/p^{\mathrm{rw}}_{ij})\right) \\
&= \sum_{\wp\in\mathcal{P}_{st}} \mathrm{P}_{st}(\wp)\hat{c}(\wp), \qquad (6.27)
\end{aligned}$$

where $\hat{c}(\wp) = \sum_{(i,j)\in\wp}\hat{c}_{ij}$ is a new path cost defined as the sum of *augmented* edge costs

$$\hat{c}_{ij} = c_{ij} + T\log(p^{(t)}_{ij}/p^{\mathrm{rw}}_{ij}) = c_{ij} + Ts_{ij} \quad (6.28)$$

along the path, where we have defined, for any $(i,j) \in E$ such that $i \neq t$, a new cost term

$$s_{ij} = \log(p_{ij}^{(t)}/p_{ij}^{\mathrm{rw}}) = -\log p_{ij}^{\mathrm{rw}} + \log p_{ij}^{(t)} = I_{ij}^{\mathrm{rw}} - I_{ij}^{(t)}, \tag{6.29}$$

where I_{ij}^{rw} and $I_{ij}^{(t)}$ denote the amount of information, or surprisal, related to observing a path traversing edge (i,j), when assuming the transition probabilities p_{ij}^{rw}, or $p_{ij}^{(t)}$, respectively. Thus, s_{ij} is the corresponding difference in surprisal, which is positive when $p_{ij}^{(t)} > p_{ij}^{\mathrm{rw}}$. In addition to the above, we separately also define $s_{tj} = 0$ for all $j \in V$. This shows that the free energy can be written as an expected cost, but considering the augmented edge costs where the standard edge costs c_{ij} are combined with the new cost terms s_{ij}, scaled by the temperature, T.

Moreover, note, that the augmented edge costs $\hat{c}_{ij}$ can be negative, namely

$$\hat{c}_{ij} < 0 \quad \text{iff} \quad p_{ij}^{(t)} < p_{ij}^{\mathrm{rw}} \exp(-\beta c_{ij}) = w_{ij}. \tag{6.30}$$

This shows that the standard problem of free energy minimization can also be formulated as a minimization problem of the expected augmented path cost (6.27) over all path distributions generated by an absorbing first-order Markov chain. The solution is again, of course, the standard RSP distribution.

The interpretation of augmented costs. Let us then consider the augmented cost in terms of the RSP distribution. Now, we know, from Equation (6.9) that when the path probabilities are determined by the RSP distribution, then the biased transition probabilities, for any $i \neq t$, can be computed as

$$p_{ij}^{(t)} = \frac{w_{ij} z_{jt}}{z_{it}} = \frac{z_{jt}}{z_{it}} p_{ij}^{\mathrm{rw}} \exp(-\beta c_{ij}). \tag{6.31}$$

Inserting this to the definition of the augmented cost (6.28), and remembering that $T = 1/\beta$ and that $\mathcal{Z}_{jt} = z_{jt}/z_{tt}$ gives

$$\begin{aligned} \hat{c}_{ij} &= T \log(z_{jt}/z_{it}) = T \log \frac{(z_{jt}/z_{tt})}{(z_{it}/z_{tt})} = T(\log \mathcal{Z}_{jt} - \log \mathcal{Z}_{it}) \\ &= \phi_{it} - \phi_{jt} \end{aligned} \tag{6.32}$$

In addition, we should define separately $\hat{c}_{tj} = 0$ for any $(t,j) \in E$. So the augmented edge cost is actually the same as the difference of free energies to t at the ends of the edge. Especially, for any $(j,t) \in E$, we have $\hat{c}_{jt} = \phi_{jt}$.

This has an interesting consequence. Namely, for the augmented path cost of an arbitrary hitting path from s to t, $\wp = (s, i_1, i_2, \dots, i_L, t)$, we have

$$\begin{aligned}\hat{c}(\wp) &= \hat{c}_{si_1} + \hat{c}_{i_1 i_2} + \cdots + \hat{c}_{i_L t} \\ &= \phi_{st} - \phi_{i_1 t} + \phi_{i_1 t} - \phi_{i_2 t} + \cdots + \phi_{i_{L-1} t} - \phi_{i_L t} + \phi_{i_L t} - \phi_{tt} \\ &= \phi_{st}\end{aligned} \tag{6.33}$$

Moreover, the augmented path cost of a loop, i.e. a path from s to itself, is zero. This notion enforces the interpretation of free energy as a potential, and implies a possible interpretation of free energy according to electrical currents in the same spirit as the electrical analogue for random walks. This connection is still under study and is left for future research

6.3.5 Relation to other free energies

We then discuss the reason behind referring to the quantity defined in Equation (2.46) as free energy. Colloquially speaking, free energy often is used in thermodynamics to describe the amount of energy that is theoretically possible to extract from a thermodynamic system, taking into account the fact that some of the internal energy of the system is "lost" during extraction due to change in the entropy of the system. For instance, the *Helmholtz free energy*, defined as

$$\phi_{st}^{\text{Helmholtz}} = U - TH, \tag{6.34}$$

where U is the internal energy and $H = \sum_{\wp \in \mathcal{P}_{st}} \mathrm{P}_{st}(\wp) \log \mathrm{P}_{st}(\wp)$ the entropy of a thermodynamic system, is very similar to the free energy in Equation (2.46).

The apparent difference, however, is that the Helmholtz FE is defined in terms of *entropy*, whereas the FE in (2.46) is in terms of *relative entropy*. Nonetheless, we can rewrite the free energy of (2.46) slightly differently as

$$\phi_{st} = \langle \tilde{c} \rangle_{st} + T\mathbb{J}(\mathrm{P}_{st} \| \mathrm{P}_{st}^{\text{rw}}) = \sum_{\wp \in \mathcal{P}_{st}} \mathrm{P}_{st}(\wp) \tilde{c}(\wp) + T \sum_{\wp \in \mathcal{P}_{st}} \mathrm{P}_{st}(\wp) \log \left(\frac{\mathrm{P}_{st}(\wp)}{\mathrm{P}_{st}^{\text{rw}}(\wp)} \right) \tag{6.35}$$

$$= \sum_{\wp \in \mathcal{P}_{st}} \mathrm{P}_{st}(\wp)(\tilde{c}(\wp) - T \log \mathrm{P}_{st}^{\text{rw}}(\wp)) + T \sum_{\wp \in \mathcal{P}_{st}} \mathrm{P}_{st}(\wp) \log \mathrm{P}_{st}(\wp) \tag{6.36}$$

$$= \langle \tilde{c}^{\mathrm{S}} \rangle_{st} - TH(\mathrm{P}_{st}), \tag{6.37}$$

where $H(\mathrm{P}_{st})$ is the Shannon entropy of the path probability distribution

$$H(\mathrm{P}_{st}) = -\sum_{\wp \in \mathcal{P}_{st}} \mathrm{P}_{st}(\wp) \log \mathrm{P}_{st}(\wp), \tag{6.38}$$

and where we define, similarly to the augmented paths costs introduced in Section 6.3.4, for path $\wp$ a *combined path cost* as

$$\tilde{c}^{\mathrm{S}}(\wp) = \tilde{c}(\wp) - T \log \mathrm{P}^{\mathrm{rw}}_{st}(\wp) \tag{6.39}$$

and $\langle \tilde{c}^{\mathrm{S}} \rangle_{st}$ is its expected value. The superscript S refers to the fact that the normal edge costs are combined with temperature T times the *surprisal*, i.e. the negative logarithm, of observing path $\wp$ considering the natural, unbiased random walk transition probabilities, p^{rw}_{ij}.

Moreover, as $\log \mathrm{P}^{\mathrm{rw}}_{st}(\wp) = \sum_{(i,j)\in\wp} \log p^{\mathrm{rw}}_{ij}$, we may simply define *combined edge costs* as

$$c^{\mathrm{S}}_{ij} = c_{ij} - T \log p^{\mathrm{rw}}_{ij} \tag{6.40}$$

as then for any path $\wp \in \mathcal{P}_{st}$, $\tilde{c}^{\mathrm{S}}(\wp) = \sum_{(i,j)\in\wp} c^{\mathrm{S}}_{ij}$.

In other words, by defining the combined edge costs, c^{S}_{ij}, based on the original edge costs, c_{ij}, and the unbiased transition probabilities, p^{rw}_{ij}, the free energy in Equation (2.46) can be rewritten as in Equation (6.37) based on the Shannon entropy, instead of relative entropy, which is a more similar form to the standard free energy formulation, such as the Helmholtz free energy in Equation (6.34).

6.3.6 Relation to maximum entropy

The RSP framework can also be understood as a maximum entropy model. Namely, the definition, stated in Section 2.3, in terms of minimization of expected cost with constrained relative entropy, may as well be stated conversely based on minimization of relative entropy with constrained expected cost. This is more reminiscent of a more standard formalization of a maximum entropy model [113, 111], as minimization of relative entropy with respect to the natural random walk probability distribution, $\mathrm{P}^{\mathrm{rw}}_{st}$, tends to maximize the Shannon entropy of a path probability distribution. However, as stated in the interpretation of $\mathbb{J}(\mathrm{P}_{st} \| \mathrm{P}^{\mathrm{rw}}_{st})$ as an information cost in Section 6.3.3, an even higher entropy is possible to achieve with a maximum entropy random walk [39, 53].

6.4 Sensitivity of free energy

One nice feature of free energy is obtained in terms of sensitivity analysis. Namely, we can rephrase the expression from Equation (5.19) for the expected number of visits through edge $(i.j)$ over the RSP distribution from s to t, i.e. its simple RSP edge betweenness with respect to s and t, as

$$\bar{\eta}_{ij}(s,t) = -\frac{1}{\beta}\frac{\partial}{\partial c_{ij}}\log \mathcal{Z}_{st} = \frac{\partial}{\partial c_{ij}}\phi_{st}. \tag{6.41}$$

This shows that the simple RSP edge betweenness index is also an index of the sensitivity of free energy between s and t to small perturbations of the edge costs.

Instead of perturbing edges, we may also consider perturbations on a node, i, which in practice means the sum of perturbations of all edges leaving (or, altenatively, entering) i. This then corresponds to the expected number of visits to i, i.e. its (simple) RSP betweenness with respect to s and t:

$$\bar{\eta}_{i}(s,t) = \sum_{j\in V}\bar{\eta}_{ij}(s,t) = \sum_{j\in V}\frac{\partial}{\partial c_{ij}}\phi_{st}, \tag{6.42}$$

The definition of $\bar{\eta}_{ij}$ could also be stated based on incoming edges, which would be more natural, as the cost associated to a node normally means the cost of entering the node. However, we use the above form as it is consistent with the results presented in Chapter 5. We omit the term "simple" in this chapter, as it was used in Chapter 5 only for differentiating from the *RSP net betweenness*.

Summing the above betweenness index over all s-t-pairs on the graph then gives the overall *RSP betweenness* of node i,

$$\mathrm{bet}_i^{\mathrm{RSP}} = \sum_{s\in V}\sum_{t\in V}\bar{\eta}_i(s,t) = \sum_{j\in V}\frac{\partial}{\partial c_{ij}}\sum_{s,t\in V}\phi_{st} = \sum_{j\in V}\frac{\partial}{\partial c_{ij}}\Phi(G), \tag{6.43}$$

where we have defined the sum of free energies between all s-t-pairs, which is also the sum of free energy distances over all s-t-pairs:

$$\Phi(G) = \sum_{s,t\in V}\phi_{st} = \sum_{s,t\in V}\Delta_{st}^{\mathrm{FE}}. \tag{6.44}$$

Remember then, from Section 5.3.1 that the RSP betweenness is known to interpolate between the *shortest path likelihood betweenness* (defined in Equation 5.2;

when $\beta \to \infty$) and the *stationary distribution* of the graph, multiplied by a constant (when $\beta \to 0^+$).

This means that the RSP betweenness of a node i actually quantifies the effect of a perturbation in the (outgoing) edge costs of i on the sum of all free energies and free energy distances on the graph. The sum of all distances has been traditionally used for various purposes in graph and network analyses, including quantifying the overall connectivity of a graph. On undirected graphs the sum of all shortest path distances is known as the *Wiener index* of the graph, which we denote $\mathcal{W}(G) = \sum_{s,t \in V} \Delta_{st}^{\mathrm{LC}}$, and in the case of the resistance distance as the *Kirchhoff index*, $\mathcal{K}(G) = \sum_{s,t \in V} \Delta_{st}^{\mathrm{Res}}$, as already introduced in Section 2.4.2, in Equation (2.36).

Sensitivity at the limits of β. Remember then that free energy generalizes both the least cost and the commute cost distances, and that the commute cost distance equals the resistance distance multiplied by the constant $C(G) = \sum_{(i,j) \in E} a_{ij} c_{ij}$ (see Equation (2.42)). Thus, we have

$$\Phi(G) \underset{\beta \to \infty}{\longrightarrow} \mathcal{W}(G) \qquad \text{and} \qquad \Phi(G) \underset{\beta \to 0^+}{\longrightarrow} C(G)\mathcal{K}(G) \tag{6.45}$$

On a quick glance, this might seem to imply that the limit functions of the simple RSP betweenness of a node, i.e. the shortest path likelihood betweenness (as $\beta \to \infty$) and the stationary distribution (as $\beta \to 0^+$), could be used to estimate the sensitivity of $\mathcal{W}(G)$ and $\mathcal{K}(G)$, respectively, in terms of perturbation of the nodes' edges. However, this is not exactly correct. For the Wiener index, $\mathcal{W}(G)$, the statement above is true only when there exist unique shortest paths between all s-t-pairs. But when there exists multiple shortest paths between an s-t-pair, the shortest path distance does not necessarily change, even if the edge cost along one of the shortest paths changes. Nevertheless, the simple RSP betweenness at the limit

For the Kirchhoff index, $\mathcal{K}(G)$ the error in the above statement is easy to detect by remembering that the edge costs, c_{ij}, are considered independent of the affinities, a_{ij}, and thus also of the resistances, $r_{ij} = 1/a_{ij}$. Accordingly, the partial derivative with respect to c_{ij} does not influence the affinities nor resistances. Moreover, the resistance distance, and thus also the Kirchhoff index, are independent of the edge costs, and thus a change in edge costs does not cause a change in resistance distances.

The sensitivity of $\mathcal{W}(G)$ and $\mathcal{K}(G)$ to perturbations of edge weights was considered by Klein in [119] as an edge centrality measure (more exactly, the sensitivity multiplied by the edge affinity, a_{ij}). The sensitivity of $\mathcal{W}(G)$ was

there briefly compared with the shortest path edge betweenness centrality of Freeman. Moreover, it was shown by Klein that the sensitivity of $\mathcal{K}(G)$ to a perturbation of an edge resistance r_{ij} is equivalent to what he calls the *power centrality* of edge (i, j), i.e. the power dissipated by edge (i, j) over electric flows between all node-pairs of the graph (see also [79], Chapter 4.9).

6.5 Discussion

This chapter presented some results and features related to free energy and the RSP framework that have not appeared in existing literature and were not presented in the earlier chapters of the thesis. First, the chapter presented methods for the computation of free energies and free energy distances in different practical circumstances. Another large part of the chapter was devoted to stating different interpretations of the free energy, as well as the RSP framework in general. These interpretations were derived from different viewpoints, for instance from an information theoretical stand, or in a statistical physics context. Lastly, the chapter investigated the interpretation of the simple RSP betweenness centrality of a node or an edge in terms of sensitivity of free energies towards perturbations of edge costs. All the observations and results made in the chapter increase the intuition and motivation behind the concept of free energy and can be valuable for discovering new application domains for free energy. However, the work presented here is still incomplete and could be extended even more for a better understanding of the concept of free energy.

Chapter 7

Maximum likelihood estimation for randomized shortest paths with trajectory data

7.1 Introduction

So far in this thesis, the RSP framework has been applied for defining metrics on graphs for various tasks. Another natural application of the framework is in modelling movement or flow phenomena in networks, which is one of the fundamental focus points in network science. This chapter focuses on the application of the RSP framework as a model of movement on a network, and especially on the problem of fitting the model to data of trajectories on networks. The motivation for the work comes from ecology and modelling of animal movement on geographical landscapes, which is also considered in Chapter 8, but the presentation is made generic so that the methods can be used for any movement or flow phenomenon on a network that are suitable for the RSP model.

7.1.1 Background and motivation

One problem that has not yet been tackled in depth in research on RSPs is the estimation of the parameters of the RSP model. In this work we derive methods for computing maximum likelihood estimates (MLEs) of the parameters of the RSP model in cases where the data consists of trajectories on a network. We

cover separately the cases where the trajectories have been recorded fully or are incomplete. Such data may arise whenever recording a movement or flow process on a network, including, for instance

- geolocation data of animals moving on a geographical network,
- trip data gathered from public transportation networks, or
- clicking behavior data of web surfers.

The focus in this work is especially on estimation of the inverse temperature parameter β, when fitting the RSP model to trajectory data and when considering that the network structure is known. A relatively high value of β (i.e. low temperature) describes data based on optimal or near-optimal paths from a source node to a destined target node, whereas a low value of β (high temperature) can describe trajectories resembling a random walk with a drift towards the target. By setting β to an appropriate value, the RSP model can take into account the assumption that movement or flow on a network often does not occur completely optimally, nor completely randomly. When considering movement on a network, such an assumption can be valid for various reasons. For example, the agent (e.g. an animal) moving on the network might not have global information of the network or be intelligent enough in order to follow optimal paths. Or the agent can simply have a simultaneous preference for both randomness (i.e. exploration) and optimality (exploitation), which can help in obtaining more knowledge of the environment or in distracting a prey or an opponent trying to predict and intercept the agent.

In addition to β, the RSP framework involves two other kinds of parameters, namely edge affinities a_{ij} (or, alternatively, transition probabilities), which define the random walk, and edge costs c_{ij}, which define the optimality of movement on the graph. Although we do also briefly discuss in this work the estimation of edge costs from trajectory data, in Section 7.2.3, the focus is mostly on the estimation of β, as commonly in RSP applications the graph structure, i.e. the edge costs and affinities, are considered fixed and known. When applying the model, the value of β is then often selected by some form of tuning. However, these tuning schemes are not based on solid statistical grounds and can be very specific for the application in question. For instance, remember from the clustering experiment in Chapter 3 (as well as from [115]) that the value of β was fixed by maximizing the clustering performance using a held-out part of the data.

The main contribution of this work is the derivation of methods for computing the likelihood of observing data consisting of complete or incomplete trajectories that are assumed being generated according to the RSP probability distribution with a fixed value of β. We confirm the validity of these methods with

artificial examples, focusing on simulated geographic networks.

In addition we test the functionality of the developed MLE methods on real data of trajectories in a geographic landscape, based on GPS recordings of movements of wild reindeer in a region of Norway. This example is devised mostly to show that the MLE computations can be run in practice and that the methods provide reasonable results. Recently the RSP framework was applied to similar data in [168], in addition to which it has been familiarized with the ecological community thanks to the implementation in the `gdistance` R package [70]. As the purpose of this work is only in developing the RSP theory for fitting the RSP model to trajectory data, we do not consider here the evaluation of the model in more specific ecological applications. Likewise, we do not make a comparison here of the RSP model with other possible tools for such applications, which have been developed in the ecological, as well as spatiotemporal statistics literature [see, e.g. 40, 56, 51]. Such empirical comparisons will be carried out in future work.

Although the initial motivation for this work came from animal movement modelling, the likelihood computation methods derived here can in general be used for modelling any kind of movement or spreading phenomena on any networks that satisfy the assumptions behind the RSP model. In addition, in future research we will investigate the use of the methods developed here for parameter estimation in the context of clustering and classification of graph nodes.

The chapter structure is as follows: In the rest of this section, we list some of the related work concerning, for instance, parameter estimation in Markov chains. In Section 7.2 we derive and validate MLEs for data consisting of full trajectories. Section 7.3 deals with MLEs in situations where the data consists of incomplete trajectories, i.e. where only a subset of the edge or node sequence of a path is observed. In Section 7.4 we use the methods derived in Section 7.3 to fit the RSP model to data consisting of GPS trajectories of wild reindeer by estimating the inverse temperature parameter associated to the trajectories.

7.1.2 Related work

The problem of estimating the parameters in the RSP model has similarities with the more general problem of estimation of Markov chain transition probabilities from trajectory data (see e.g. [50, 147, 190]). Indeed, when assuming the RSP model, a given value of β defines the biased transition probabilities towards the target node, according to Equation (6.9), and thus estimating β implies estimating the transition probabilities. However, the general estimation

of Markov chains, studied in the works cited above, does not involve assumptions on the distribution of paths on the graph, and also does not consider affinities, costs or other weights related to the transitions. Moreover, Markov chain estimation has been mostly studied for continuous time Markov jump processes, for the estimation of the *generator matrix* of the process. Such techniques have been developed e.g. for healthcare research, for studying disease progression and treatment effects, where the temporal aspect is of course vital. Instead, the RSP framework is only analogous to a discrete time Markov chain, although extending the idea of RSPs to consider continuous time is a planned topic for future research.

Another problem related to the current work is the inference of *aggregate Markov chains*, which has been applied for ion channel modeling [172]. There the state space is divided into aggregates and only the aggregate that the system is in, instead of the actual state, is observed. Aggregated Markov chains have also been only considered in the context of continuous time chains.

Besides the temporal aspect there are also other differences between the general estimation of Markov chains and the estimation of RSP parameters. Namely, the methods for Markov chain estimation normally require data containing observations at each state of the chain, as the transition probabilities for unobserved states cannot be estimated [22]. Estimation of the inverse temperature parameter in the RSP model, however, does not require this, and it can be estimated even for a sparse set of incomplete trajectories, as derived, and confirmed with experiments, in Section 7.3.

7.2 Fully observed trajectories

The rest of this chapter is dedicated to deriving and verifying methods for computing the likelihood of the RSP model parameters when modelling trajectory data, and computing the maximum likelihood estimates (MLEs). The focus is mostly on the likelihood of the inverse temperature parameter β. For estimating β, the *graph structure*, i.e. the edge affinities (or the reference transition probabilities) and edge costs, are always assumed known completely and exactly. We denote the MLE of β in general as $\hat{\beta}_{\mathrm{MLE}}$.

This section deals with data sets of *full trajectories* on a network, i.e. where each node (and thus each edge) of the observed path has been recorded. We derive the results by first considering the simple case where all trajectories are observed to go from one starting node s to one destination node t, and then extend to considering trajectories between several s-t-pairs. The methods are then validated with artificial data. We also present briefly in this section a

method for estimating the edge costs c_{ij} from full trajectories, when considering that only the affinities, or reference transition probabilities are known, but otherwise the focus of the chapter is on estimating β. Later, in Section 7.3, we consider likelihood MLE computation in the case where the data consists of incomplete trajectories, where only a part of the edges or nodes visited along each trajectory is observed.

7.2.1 Single source and target

Consider a data set Ω_{st} containing K full observed trajectories (i.e. hitting paths) going from s to t. Assuming independence between the trajectories, the likelihood of β is then

$$\begin{aligned}\mathcal{L}(\beta|\Omega_{st}) = \mathrm{P}(\Omega_{st};\beta) = \prod_{\wp\in\Omega_{st}} \mathrm{P}_{st}(\wp) = \prod_{\wp\in\Omega_{st}} \frac{\widetilde{w}(\wp)}{\mathcal{Z}_{st}} \\ = \prod_{\wp\in\Omega_{st}} \frac{\mathrm{P}^{\mathrm{rw}}_{st}(\wp)\exp(-\beta\tilde{c}(\wp))}{\mathcal{Z}_{st}}.\end{aligned} \tag{7.1}$$

Furthermore, the log-likelihood is

$$\log\mathcal{L} = \sum_{\wp\in\Omega_{st}} (\log \mathrm{P}^{\mathrm{rw}}_{st}(\wp) - \beta\tilde{c}(\wp)) - K\log\mathcal{Z}_{st} \tag{7.2}$$

Taking the derivative of the log-likelihood with respect to β, and setting it to zero gives us the criterion for the MLE:

$$\frac{\partial\log\mathcal{L}}{\partial\beta} = -\sum_{\wp\in\Omega}\tilde{c}(\wp) - K\frac{\partial\log\mathcal{Z}_{st}}{\partial\beta} = 0 \iff \langle\tilde{c}\rangle_{st} = \frac{1}{K}\sum_{\wp\in\Omega}\tilde{c}(\wp) \tag{7.3}$$

where we used Equation (2.45). Thus, the MLE of the inverse temperature, $\hat{\beta}_{\mathrm{MLE}}$, is the value for which the expected cost over the RSP distribution equals the average cost of the observed trajectories, as commonly observed in a large class maximum entropy estimation procedures [113].

Note that the average cost of the observed trajectories may be higher than the expected cost of the reference random walk. This can happen e.g. when generating a set of paths according to the RSP distribution with a value of β close to zero as then individual paths can become longer (more costly) than the average random walk cost. In such a case, the MLE of β does not exist, as for any $\beta > 0$, the expected cost over RSPs will be smaller than the expected cost of the reference random walk. Instead, one can then consider the RSP

distribution with $\beta < 0$, which defines a biased Markov chain where the bias is *away* from the destination, instead of being *towards* the destination, as with the standard RSP distribution with $\beta > 0$. On the other hand, also when trajectories in the data follow exactly the least cost paths, the MLE of β often does not exist. This is because the expected cost over the RSP distribution from s to t with any $\beta < \infty$ is strictly larger than the least cost, excluding some special cases, such as when there exists only one path from s to t (remember that in our terminology, paths can contain cycles).

7.2.2 Multiple sources and targets

Let us then extend to the case, where full trajectories are observed between a set of different s-t-pairs. Let then Ω be the set of all trajectories and let X denote the set of s-t-pairs for which at least one trajectory has been observed, $\Omega_{st} \subset \Omega$ the set of trajectories in the data that go from a particular s to a particular t, and $K_{st} = |\Omega_{st}|$ the number of trajectories from s to t. The likelihood is then given simply by the product of the likelihoods for each s-t-pair:

$$\begin{aligned} \mathcal{L}(\beta \mid \Omega) &= \prod_{\wp \in \Omega} \mathrm{P}(\wp) = \prod_{(s,t) \in X} \prod_{\wp_{st} \in \Omega_{st}} \mathrm{P}_{st}(\wp_{st}) \\ &= \prod_{(s,t) \in X} \prod_{\wp_{st} \in \Omega_{st}} \frac{\mathrm{P}^{\mathrm{rw}}_{st}(\wp_{st}) \exp(-\beta \tilde{c}(\wp_{st}))}{\mathcal{Z}_{st}}. \end{aligned} \tag{7.4}$$

The log-likelihood is

$$\log \mathcal{L} = \sum_{(s,t) \in X} \left(-K_{st} \log \mathcal{Z}_{st} + \sum_{\wp_{st} \in \Omega_{st}} (\log \mathrm{P}^{\mathrm{rw}}_{st}(\wp_{st}) - \beta \tilde{c}(\wp_{st})) \right). \tag{7.5}$$

Again, setting the derivative to zero, we see that the MLE of β should satisfy

$$\frac{\partial \log \mathcal{L}}{\partial \beta} = - \sum_{(s,t) \in X} \left(K_{st} \frac{\partial \log z_{st}}{\partial \beta} + \sum_{\wp_{st} \in \Omega_{st}} \tilde{c}(\wp_{st}) \right) = 0 \tag{7.6}$$

$$\iff \sum_{(s,t) \in X} K_{st} \langle \tilde{c} \rangle_{st} = \sum_{(s,t) \in X} \sum_{\wp_{st} \in \Omega_{st}} \tilde{c}(\wp_{st}). \tag{7.7}$$

Thus, as before, the most likely value of β corresponds to the one for which the empirical average agrees with the expected value given by the model.

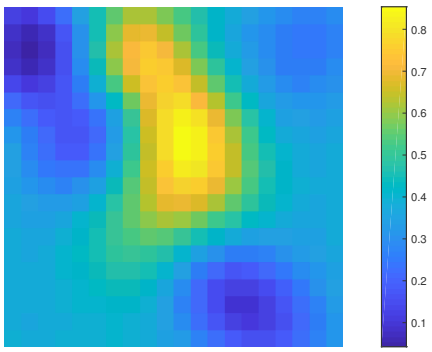


FIGURE 7.1: The generated 20×20 Gaussian landscape used in the experiments. Each pixel j is colored based on the cost of moving to the node, i.e. the cost c_{ij} for any i s.t. $(i, j) \in E$.

Validation of full-path maximum likelihood estimation

The above results and their accuracy were evaluated by generating trajectories on two kinds of artificial geographic landscape graphs. The first graph is a simple 20×20 grid with uniform edge costs. The second is a 20×20 Gaussian landscape, where edge costs are determined on the cost associated to the ending node of the edge, and the nodes costs have been assigned based on a mixture of Gaussian distributions on a plane overlayed on the grid.

The Gaussian landscape used in the experiments is depicted in Figure 7.1. The landscape consists of five low-cost and five high-cost Gaussian patches, which cause decrease and increase of the costs from a base value of 0.5. In both the uniform grid and the Gaussian landscape, each node is connected to its adjacent and diagonal neighbors and the cost of the diagonal connections is multiplied by $\sqrt{2}$. Affinities are fixed as reciprocals of costs, i.e. $a_{ij} = 1/c_{ij}$, and reference transition probabilities (as explained earlier) as the normalized affinities, $p^{\mathrm{rw}}_{ij} = a_{ij} / \sum_k a_{ik}$.

Using a set of values for β, 200 paths were generated on both landscapes based on each studied value of β. Each path was generated by first drawing an s-t-pair uniformly randomly on the grid, however only accepting node pairs that were at least 3 steps apart on the grid. A path between each s-t-pair was then generated by using the biased transition probabilities from Equation 6.9. Then, for each such set of paths Ω, the value $\hat{\beta}_{\mathrm{MLE}}(\Omega)$ was inferred by finding, by a simple line search, the value of β that satisfied Equation (7.7). The above procedure was repeated 10 times for each studied value of β and the mean and

β	$\hat{\beta}_{\mathrm{MLE}}$, uniform grid	$\hat{\beta}_{\mathrm{MLE}}$, simulated landscape
0.001	0.00096 ± 0.00020	0.00111 ± 0.00024
0.005	0.00486 ± 0.00053	0.00526 ± 0.00064
0.01	0.00970 ± 0.00085	0.01029 ± 0.00115
0.05	0.04874 ± 0.00306	0.04894 ± 0.00511
0.1	0.09785 ± 0.00497	0.09956 ± 0.00392
0.5	0.49601 ± 0.01908	0.50897 ± 0.02236
1	1.01719 ± 0.03833	0.99922 ± 0.02422
5	5.07901 ± 0.23531	4.99453 ± 0.17301
10	10.08117 ± 1.04427	10.05533 ± 0.35628

TABLE 7.1: The MLEs in the experiment with complete observed trajectories. The first column on the left shows the value of β used for generating the paths and the two other columns show the mean $\pm$ the standard deviation of the MLEs over 10 repetitions.

standard deviation of the MLEs was recorded. The results in Table 7.1 show that the MLEs, on average, indeed closely match the true values of β used for generating the paths.

7.2.3 Estimation of edge costs

So far we have only focused on the problem of estimating the inverse temperature parameter β in situations where the graph structure, i.e. the edge affinities (or the reference transition probabilities) and the edge costs are known a priori. In this section, however, we consider briefly the problem of estimation of edge costs from trajectories.

Problem description

Assume that only the edge affinities (or the reference transition probabilities) are known a priori. Moreover, consider that the value of β is fixed to some constant value. We then try to estimate the edge costs based on the observed trajectories.

It turns out that this estimation problem actually contains the problem of estimating β, meaning that solving the edge cost estimation problem also solves the estimation problem of β. This is due to the fact that concerning the RSP distribution over paths (Equation (2.10)), parameter β functions simply as a scaling factor of the path costs. Namely, consider that a data set of trajectories is generated according to the RSP distribution with a fixed inverse temperature, say β^*, which is unknown to the user. Thus, the probability of a path $\wp$ is

as in Equation (2.10):

$$\mathrm{P}_{st}(\wp) = \frac{\mathrm{P}^{\mathrm{rw}}_{st}(\wp)\exp(-\beta^*\tilde{c}(\wp))}{\sum_{\wp\in\mathcal{P}_{st}} \mathrm{P}^{\mathrm{rw}}_{st}(\wp)\exp(-\beta^*\tilde{c}(\wp))}. \tag{7.8}$$

But this is the same distribution as one would obtain by considering the RSP distribution with $\beta = 1$, but on a modified graph with edge costs $\hat{c}_{ij} = \beta^* c_{ij}$, as then, for each $\wp$,

$$\beta^*\tilde{c}(\wp) = \beta^* \sum_{(i,j)\in\wp} c_{ij} = \sum_{(i,j)\in\wp} \hat{c}_{ij} = \hat{c}(\wp). \tag{7.9}$$

Expressed conversely, if we try to estimate the costs from the trajectories, we may simply assume that $\beta = 1$, as a result of which we should end up with cost estimates $\hat{c}_{ij}$. But these estimates contain the information of both the "original" edge costs and the value β^* used to generate the data on the original graph. This shows that the estimation of β is in practice only a subproblem of the edge cost estimation problem.

Additional remarks. There are a few points worth emphasizing related to the edge cost estimation problem. The first is that the unit of measurement of the "original" costs, c_{ij}, is different from the estimated edge costs $\hat{c}_{ij} = \hat{\beta} c_{ij}$. This can be neglected when the interest is only on the path distribution itself, but it should be kept in mind when making judgements related to the actual costs. For instance, if the original edge costs correspond to time units and the trajectories are generated by a randomized process with a fixed value of β, the estimated edge costs $\hat{c}_{ij}$ will not necessarily correspond to time units (which determine e.g. the duration of fastest paths on the graph). In such a case, one would need additional knowledge of actual durations of the trajectories, with which the value of β could be determined separately from the edge costs.

The second point worth noting is that in this approach the affinities, or reference transition probabilities must be considered known a priori. However, one can argue that the estimation of transition probabilities is an easier task than the estimation of edge costs, as the transition probabilities are based on simply local features of the graph.

A final point is that an edge cost can only be estimated reasonably if at least some of the trajectories pass through the edge. And obviously, the more trajectories the data contains, the more reliable the estimates normally are. Partly as a consequence of this, we consider the edge cost estimation problem in this work only in the setting where the data contains complete trajectories.

The possible extension for the case of incomplete trajectories is left for future research.

Derivation of MLE

We now derive the MLE of edge costs in the case of complete trajectories. For this, we assume the value $\beta = 1$, and search for edge costs that maximize the likelihood of the trajectories. Recall the log-likelihood of observing a set Ω_{st} of K trajectories from s to t, from Equation (7.2) (with $\beta = 1$):

$$\log \mathcal{L} = \left(\sum_{\wp \in \Omega_{st}} (\log \mathrm{P}^{\mathrm{rw}}_{st}(\wp) - \tilde{c}(\wp)) \right) - K \log \mathcal{Z}_{st}. \tag{7.10}$$

Consider then the partial derivative w.r.t. an edge cost c_{ij}:

$$\frac{\partial}{\partial c_{ij}} \log \mathcal{L} = K \bar{\eta}_{ij}(s,t) - \sum_{\wp \in \Omega_{st}} \eta_{ij}(\wp). \tag{7.11}$$

where $\eta_{ij}(\wp)$ is the number of times the edge (i,j) is traversed on a observed trajectory $\wp$ while $\bar{\eta}_{ij}(s,t)$ is the expected number of times edge (i,j) is traversed on paths from s to t. As a result, the likelihood is maximized when, for all $(i,j) \in E$,

$$\bar{\eta}_{ij}(s,t) = \frac{1}{K} \sum_{\wp \in \Omega_{st}} \eta_{ij}(\wp). \tag{7.12}$$

A direct corollary of the above result is that when the data set consists of paths between multiple s-t-pairs, $\Omega = \cup_{s,t} \Omega_{st}$, with $K_{st} = |\Omega_{st}|$, the likelihood is maximized, when

$$\sum_{(s,t) \in X} \bar{\eta}_{ij}(s,t) = \sum_{(s,t) \in X} \frac{1}{K_{st}} \sum_{\wp_{st} \in \Omega_{st}} \eta_{ij}(\wp_{st}). \tag{7.13}$$

Unfortunately, as was the case with the MLE of β, the above cannot be solved analytically for c_{ij}. Moreover, the dimension of the problem is now the number of edges, so the solution cannot be found only by a simple line search.

The MLE of the edge costs can, instead, be sought e.g. by performing a gradient ascent towards the direction given by the edge derivatives in Equation (7.11), above. We demonstrate this idea by using a similar experimental setting and a similar, though smaller (for practical purposes) Gaussian landscape as in

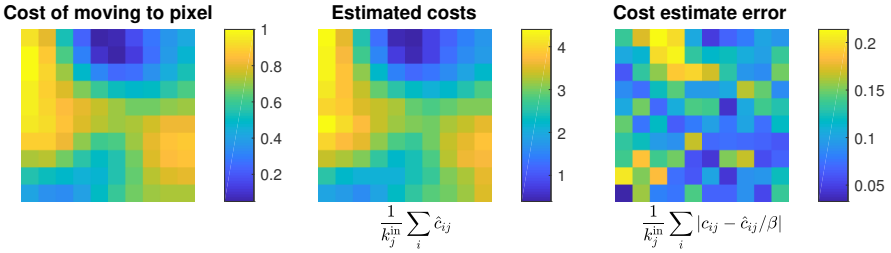


FIGURE 7.2: Heatplots showing, for each pixel j, the cost of moving to j (i.e. c_{ij}, for any i s.t. $(i,j) \in E$) (a), the average of the estimated costs $\hat{c}_{ij}$ for any i s.t. $(i,j) \in E$, obtained with gradient ascent (b), and the average absolute error of the estimated edge costs (c). The Pearson correlation between the original and the estimated costs was 0.92. Note that the estimated costs actually estimate the scaled original costs, $\hat{c}_{ij} = \beta c_{ij}$, as explained in the text.

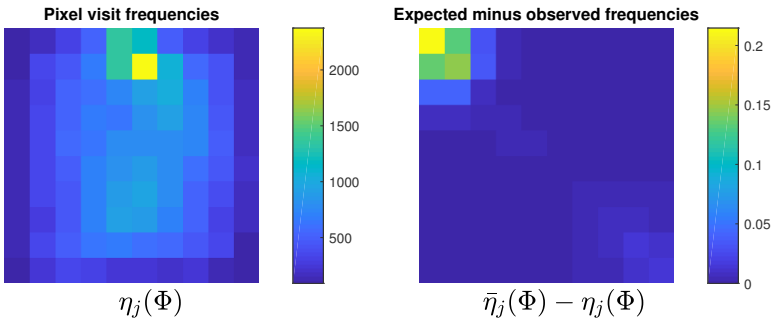


FIGURE 7.3: Heatmaps of number of visits at each pixel during the 10000 generated trajectories (a) and the difference between the expected number of visits according to RSP and the observed frequencies (b). Note the difference in scale of the two plots, indicated by the colorbar, which shows that the difference between the expected and observed values is in practice negligible.

Section 7.2.2. An illustration of the landscape is in Figure 7.2A, which shows a heatmap of the cost of moving into each pixel. We generated a set of paths on the landscape by randomly selecting 10000 pairs of starting and ending nodes, and by drawing paths between the node pairs according to the RSP probabilities with a fixed value of $\beta = 2$.

We then assumed that the edge affinities are known, but initialized the cost matrix to costs of unity on each edge. We ran gradient ascent by updating the edge costs based on Equation (7.11) as

$$c_{ij} \longleftarrow c_{ij} + \gamma \sum_{s,t=1}^{n} \left(K_{st} \bar{\eta}_{ij}(s,t) - \sum_{\wp_{st} \in \Omega_{st}} \eta_{ij}(\wp_{st}) \right), \tag{7.14}$$

with a suitable learning factor γ, until convergence. We ran the process until the sum, over all edges, of absolute differences between the observed and expected numbers of traversals was less than one. Remember that in this example the edge costs are determined by the cost on the node, i.e. that for any pixel j, the edge costs c_{ij} are constant for all $i \in V$. However, this is not used as an assumption, when estimating the costs.

Figure 7.3 shows that the process indeed finds edge costs for which the expected number of visits over the RSP distribution is close to the observed number of visits. However, regardless of the similarity of the numbers of visits, the ML edge cost estimates are actually somewhat different from the original costs. The estimated edge costs are visualized in Figure 7.2B, which shows the average cost of moving to a pixel (again, remember that the estimated edge costs can be different for edges going into one node, which is why we can only visualize the cost related to a pixel as the average of the edge costs). The Pearson correlation between the original and estimated edge costs is 0.92. The difference between the original and estimated edge costs can be seen in Figure 7.2C, which shows, for each pixel j, the average absolute difference between the actual cost of moving to a node and the estimated edge costs of moving to that node, where the estimated costs are scaled by division with the value of β used for generating the trajectories (as the process is only able to estimate the scaled costs $\hat{c}_{ij} = \beta c_{ij}$).

Of course, the quality of the estimates is strongly dependent on the number of trajectories in the data. For this experiment, the graph size is kept fairly small and the number of trajectories is quite large, but still the cost estimates differ from the real ones to some extent. This show that the exact estimation of the edge costs is a fairly complex and difficult task, if no other prior information can be used for the estimation besides the edge affinities of the graph and the trajectories.

7.3 Incomplete trajectories

This section deals with the case where data consists of *incomplete trajectories* from s to t. For a path $\wp \in \mathcal{P}_{st}$, an incomplete trajectory means a subsequence of either the edge sequence or the node sequence of $\wp$. The task in this section is to derive a method for computing the likelihood of the inverse temperature parameter β given an observed data set Ω of incomplete trajectories. This provides a useful tool for many applications, as often in empirical routing or tracking studies trajectories can only be recorded partially.

We first consider the simple case where the data set Ω consists of incomplete trajectories that contain only one edge, observed from one path. We derive the *one-edge likelihood*, and generalize it then to the *one-node likelihood*, i.e. the case where each incomplete trajectory contains only one node instead of an edge. We then generalize to incomplete trajectories containing two observed edges, and from there to data with incomplete trajectories containing any number of edges or nodes. We also demonstrate experimentally the behavior of the likelihood and validity of the methods. Details on the implementation of computation of the quantities presented in this section is discussed in more depth in Appendix 7.C.

The methods developed here for likelihood computation for incomplete trajectories are generic in the sense that they make the most naive possible assumption on the method of observation, i.e. the sampling of edges or nodes along the trajectories. In more detail, they assume that the observed edges have been drawn uniformly randomly from the actual trajectory, and, on the other hand, makes no special assumptions on the number of observations, or the time duration or number of intermediate steps occuring between observations. Although somewhat simplistic, this makes the method applicable to many different use case scenarios. The sampling assumption is discussed more in Section 7.3.3.

7.3.1 One observed edge

Consider first a situation, where a hitting path $\wp \in \mathcal{P}_{st}$ has been observed to traverse an edge (i, j). Let us denote by ρ the random variable corresponding to the path along which the edge is observed and by ε the random variable corresponding to the observed edge. Based on the uniformity assumption discussed above in Section 7.3, the conditional probability of drawing edge (i, j), given a particular path $\wp$, is given by the number of traversals over (i, j),

η_{ij}, divided by the total number of steps along the path $\wp$, i.e. its length, $L(\wp)$:

$$\mathrm{P}(\varepsilon = (i,j) | \rho = \wp) = \frac{\eta_{ij}(\wp)}{L(\wp)}. \tag{7.15}$$

Thus, the joint probability of observing edge (i,j) along path $\wp$ is

$$\mathrm{P}(\varepsilon = (i,j), \rho = \wp) = \mathrm{P}(\rho = \wp)\mathrm{P}(\varepsilon = (i,j) | \rho = \wp) = \mathrm{P}_{st}(\wp)\frac{\eta_{ij}(\wp)}{L(\wp)}. \tag{7.16}$$

Recall from Equation (5.19) that the number of traversals over (i,j) can be expressed with the derivative $\frac{\partial}{\partial c_{ij}}\tilde{c}(\wp) = \eta_{ij}(\wp)$. Moreover, for the path likelihood $\widetilde{w}(\wp)$ (Equation (2.13)),

$$\frac{\partial}{\partial c_{ij}}\widetilde{w}(\wp) = \partial_{c_{ij}}(\mathrm{P}^{\mathrm{rw}}_{st}(\wp)\exp(-\beta\tilde{c}(\wp))) = -\beta\widetilde{w}(\wp)\eta_{ij}(\wp). \tag{7.17}$$

The probability of observing edge (i,j), i.e. the likelihood of β, is then given by marginalizing out ρ:

$$\mathcal{L}(\beta \mid (i,j)) = \mathrm{P}(\varepsilon = (i,j); \beta) = \sum_{\wp \in \mathcal{P}_{st}} \mathrm{P}_{st}(\wp)\frac{\eta_{ij}(\wp)}{L(\wp)} \tag{7.18}$$

$$= \frac{1}{\mathcal{Z}_{st}} \sum_{\wp \in \mathcal{P}_{st}} \frac{\widetilde{w}(\wp)\eta_{ij}(\wp)}{L(\wp)} \tag{7.19}$$

$$= -\frac{1}{\beta\mathcal{Z}_{st}}\frac{\partial}{\partial c_{ij}} \sum_{\wp \in \mathcal{P}_{st}} \frac{\widetilde{w}(\wp)}{L(\wp)} \tag{7.20}$$

$$= -\frac{1}{\beta\mathcal{Z}_{st}}\frac{\partial}{\partial c_{ij}}\widetilde{\mathcal{Z}}_{st}, \tag{7.21}$$

where we define a new partition function

$$\widetilde{\mathcal{Z}}_{st} = \sum_{\wp \in \mathcal{P}_{st}} \frac{\widetilde{w}(\wp)}{L(\wp)} = \sum_{k=1}^{\infty} \sum_{\wp_k \in \mathcal{P}^{(k)}_{st}} \frac{\widetilde{w}(\wp_k)}{k}, \tag{7.22}$$

with $\mathcal{P}^{(k)}_{st}$ being the set of hitting paths from s to t of length k. Recalling Equation (2.16) and that $\widehat{\mathbf{W}}$ is the matrix obtained from matrix $\mathbf{W}$ by setting row t to zero, we see that the new partition function can be expressed in matrix form

using the *matrix logarithm* [105]:

$$\widetilde{\mathcal{Z}}_{st} = \mathbf{e}_s^\mathsf{T}\left(\widehat{\mathbf{W}} + \frac{\widehat{\mathbf{W}}^2}{2} + \frac{\widehat{\mathbf{W}}^3}{3} + \cdots\right)\mathbf{e}_t = \mathbf{e}_s^\mathsf{T}\Big(-\mathbf{log}(\mathbf{I} - \widehat{\mathbf{W}})\Big)\mathbf{e}_t, \tag{7.23}$$

which exists, as $\rho(\widehat{\mathbf{W}}) < 1$.

Next we show how the derivative of the above quantity, needed for the computation of the log-likelihood (7.21), can be computed.[1] First of all, for any $(i,j) \in E$ such that $i \neq t$,

$$\partial_{c_{ij}}\widehat{\mathbf{W}} = \big[\partial_{c_{ij}} w_{ij}\big]\mathbf{I}_{ij} = \big[\partial_{c_{ij}} p_{ij}^{\mathrm{rw}} \exp(-\beta c_{ij})\big]\mathbf{I}_{ij} = -\beta \mathbf{W}_{ij}, \tag{7.24}$$

where $\mathbf{I}_{ij} = \mathbf{e}_i\mathbf{e}_j^\mathsf{T}$ is the matrix with 1 at element (i,j) and zero elsewhere, and where we have defined $\mathbf{W}_{ij} = w_{ij}\mathbf{I}_{ij}$. We can thus write the derivative of $\widehat{\mathbf{W}}^k$ as

$$\begin{aligned}
\frac{\partial}{\partial c_{ij}}\widehat{\mathbf{W}}^k &= (\partial_{c_{ij}}\widehat{\mathbf{W}})\widehat{\mathbf{W}}^{k-1} + \widehat{\mathbf{W}}(\partial_{c_{ij}}\widehat{\mathbf{W}}^{k-1}) \\
&= -\beta\Big(\mathbf{W}_{ij}\widehat{\mathbf{W}}^{k-1} + \widehat{\mathbf{W}}\Big(\mathbf{W}_{ij}\widehat{\mathbf{W}}^{k-2} + \widehat{\mathbf{W}}\Big(\mathbf{W}_{ij}\widehat{\mathbf{W}}^{k-3} + \cdots\Big)\Big)\Big) \\
&= -\beta\sum_{l=1}^{k}\widehat{\mathbf{W}}^{l-1}\mathbf{W}_{ij}\widehat{\mathbf{W}}^{k-l} \triangleq -\beta\mathbf{S}_{ij}^{(k)},
\end{aligned} \tag{7.25}$$

where we have defined, for convenience,

$$\mathbf{S}_{ij}^{(k)} = \sum_{l=1}^{k}\widehat{\mathbf{W}}^{l-1}\mathbf{W}_{ij}\widehat{\mathbf{W}}^{k-l}. \tag{7.26}$$

This matrix can be computed with the help of an auxiliary $(2n \times 2n)$ block matrix

$$\mathbf{Q}_{ij} = \begin{bmatrix} \widehat{\mathbf{W}} & \mathbf{W}_{ij} \\ \mathbf{0} & \widehat{\mathbf{W}} \end{bmatrix}, \tag{7.27}$$

as the k-th power of this matrix can be shown to be:

$$\mathbf{Q}_{ij}^k = \begin{bmatrix} \widehat{\mathbf{W}}^k & \mathbf{S}_{ij}^{(k)} \\ \mathbf{0} & \widehat{\mathbf{W}}^k \end{bmatrix}, \tag{7.28}$$

[1] Although there exist tools for computing the matrix logarithm and its Fréchet derivative [2], we found these unreliable and inaccurate in some cases (see Appendix 7.C), because of which we present here an elementary derivation of the computation.

Accordingly, the derivative of the new partition function can be computed as

$$\partial_{c_{ij}} \widetilde{\mathcal{Z}}_{st} = \mathbf{e}_s^\mathsf{T} \sum_{k=1}^{\infty} \left(\partial_{c_{ij}} \frac{\widehat{\mathbf{W}}^k}{k} \right) \mathbf{e}_t \tag{7.29}$$

$$= -\beta \mathbf{e}_s^\mathsf{T} \sum_{k=1}^{\infty} \frac{\mathbf{S}_{ij}^{(k)}}{k} \mathbf{e}_t \tag{7.30}$$

$$= -\beta \mathbf{e}_s^\mathsf{T} \sum_{k=1}^{\infty} \frac{\mathbf{Q}_{ij}^k}{k} \mathbf{e}_{n+t} \tag{7.31}$$

$$= -\beta \mathbf{e}_s^\mathsf{T} \mathbf{L}_{ij} \mathbf{e}_{n+t} \tag{7.32}$$

$$= -\beta [\mathbf{L}_{ij}]_{s,n+t}, \tag{7.33}$$

where $\mathbf{L}_{ij} = \sum_{k=1}^{\infty} \frac{\mathbf{Q}_{ij}^k}{k} = -\mathbf{log}(\mathbf{I} - \mathbf{Q}_{ij})$.

Finally, combining (7.21) and (7.33), the likelihood can be expressed as

$$\mathcal{L}(\beta \mid (i,j)) = -\frac{1}{\beta \mathcal{Z}_{st}} (-\beta [\mathbf{L}_{ij}]_{s,n+t}) = \frac{[\mathbf{L}_{ij}]_{s,n+t}}{\mathcal{Z}_{st}}. \tag{7.34}$$

The computation of the single-edge likelihood can also be understood from another, more informal, perspective. Namely, matrix $\mathbf{Q}_{ij}$ from Equation (7.27) can be interpreted as a likelihood matrix defining a new graph consisting of the original graph G, with edge likelihoods w_{ij}, augmented with its copy G' and with an edge from node i of subgraph G to node j of subgraph G' (i.e. node $n+j$ of the new graph). Element $(s, n+t)$ of the k-th power of matrix $\mathbf{Q}_{ij}$ thus enumerates all s-t-paths of length k that traverse edge (i,j) and cumulates the corresponding path likelihoods. This idea is illustrated in Figure 7.4.

Example of one observed edge

We demonstrate the functionality of the method with a simple example illustrated in Figure 7.5A. We consider a 5×5 grid with links to diagonal neighbors included. The edge costs are 1 for horizontal and vertical edges, and $\sqrt{2}$ for diagonal edges and affinities as inverse costs, $a_{ij} = 1/c_{ij}$. We consider paths from node 7, in the lower left corner, to node 25 in the upper and right-most corner. We then compute the likelihood of observing each of the edges leaving from node 7 as an edge of such a path. The likelihood is computed for different values of β.

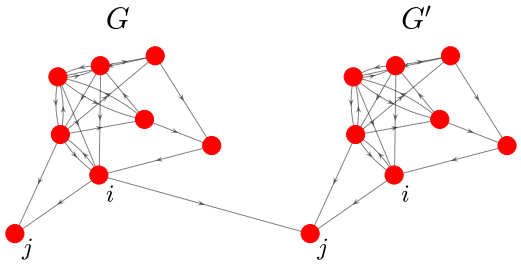


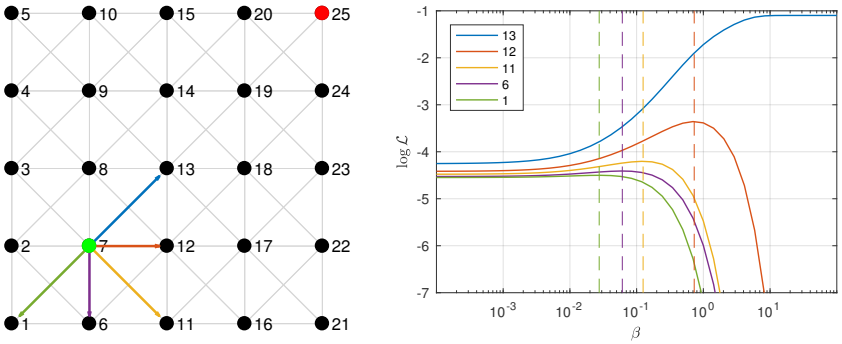
FIGURE 7.4: The computation of the one-edge likelihood can be interpreted as making a copy G' of the original graph G and adding an edge from node i of G to node j of G' and considering likelihoods of paths from node s of G to node t of G'.

The results are plotted in Figure 7.5B, which shows the likelihoods for each edge as a function of β as separate curves. Moreover, the dashed vertical lines indicate the values of β showing the maximum likelihood of observing the corresponding edge. As is expected, the β that obtains maximum likelihood is highest for the edge $(7, 13)$, which lies on the least cost path from node 7 to node 25 (in fact, in this case, the likelihood increases indefinitely, indicating that $\hat{\beta}_{\mathrm{MLE}} = \infty$). In addition, the β obtaining the maximum likelihood decreases as we consider edges that move more and more away from the shortest path.

7.3.2 One observed node

Maximum likelihood estimation

The likelihood of β can also be computed in the case where a trajectory is observed to visit a particular node instead of an edge. For this, we denote with ν the random variable corresponding to the intermediate node drawn from a path (and the r.v. corresponding to the path with ρ, as before). We derive the *one-node likelihood* using the fact that the number of visits to a node i along a path $\wp$, $n_i(\wp)$, can be expressed as the sum of travelsals out of that node: $n_i(\wp) = \sum_{j \in V} \eta_{ij}(\wp)$. Note that this definition of $n_i(\wp)$ directly means that the observed node cannot be the absorbing terminal node t, i.e. that the observation is made before the end of the path. However, the possibility that the intermediate node, drawn from a path $\wp \in \mathcal{P}_{st}$, is actually node s, must be considered more carefully. In this work, we assume that the outcome of ν can



(A) The grid (B) The likelihood functions w.r.t. β

FIGURE 7.5: The 5×5 grid, where the green node 7 is the starting node, s, the red node 25 is the absorbing target node, t (A); the log-likelihoods of observing each of the coloured edges for different values of β (B). Note the logarithmic scale on the horizontal axis.

indeed be node s, but that the observation is done *after the first step*, i.e. that the path revisits the starting node s and one of these revisits was observed.

In general, we assume that ν is drawn from the subsequence $\wp(1, \ldots, L(\wp)-1)$, excluding the first node $\wp(0) = s$ and the last node $\wp(L(\wp)) = t$. Moreover, we assume that the node is sampled uniformly from this subsequence. The exclusion of the first node $\wp(0)$ from consideration complicates the derivation a bit compared to the derivation of the single-edge likelihood. Because of this, the full derivation of the result is presented separately in Appendix 7.A.

The one-node likelihood can be computed similarly to the one-edge likelihood in Equation (7.34), as:

$$\mathcal{L}(\beta \mid i) = \frac{1}{\mathcal{Z}_{st}} \sum_{h \in V} w_{su} [\mathbf{L}_i]_{h,n+t}, \tag{7.35}$$

which contains, again, a matrix logarithm, $\mathbf{L}_i = -\mathbf{log}(\mathbf{I} - \mathbf{Q}_i)$, where

$$\mathbf{Q}_i = \begin{bmatrix} \widehat{\mathbf{W}} & \mathbf{W}_{\mathrm{r}(i)} \\ \mathbf{0} & \widehat{\mathbf{W}} \end{bmatrix}, \tag{7.36}$$

and $\mathbf{W}_{\mathrm{r}(i)} = \mathbf{I}_{ii}\mathbf{W}$ is the matrix containing the i-th row of matrix $\mathbf{W}$ on its i-th row, but zeros elsewhere.

7.3.3 Multiple observed edges or nodes

In many data sets of incomplete trajectories, the trajectories contain several observations, meaning that the trajectories have been observed to pass through more than only one edge or node of the network. The next step therefore generalizes the above derivations to such cases. We proceed by first extending from the one-edge likelihood to the two-edge likelihood. This is only done as a clarifying intermediate step for the derivation of the *multiple-edge* and *multiple-node likelihoods*.

Two-edge likelihood

First we will consider a generalization from the one-edge likelihood to the case where a path $\wp \in \mathcal{P}_{st}$ is observed to traverse *two edges*. Note that also the order of the edges is recorded, i.e. we know which of the two edges appears before the other one. We then make the assumption, as elsewhere in the chapter, that the observations of the two edges are drawn uniformly randomly from the edge sequence of $\wp$. We will discuss the relevance and consequences of this uniformity assumption for the method more later in this section, when we deal with the problem of multiple-edge likelihood.

With the uniform distribution assumption, the probability of observing a particular pair of intermediate edges $e_1 = (i_1, j_1)$ and $e_2 = (i_2, j_2)$ (in that order), over a particular path $\wp$ is given by the number of times that e_1 and e_2 can be drawn from $\wp$ in that order, divided by the number of all possible combinations of two edges from $\wp$. Formally, if we now denote by ε_1 and ε_2 the random variables corresponding to the first and second drawn edges, respectively, and, again, by ρ the random variable corresponding to the trajectory drawn according to the RSP probability, then we have, for any $\wp \in \mathcal{P}_{st}$,

$$\mathrm{P}_{st}(\varepsilon_1 = e_1, \varepsilon_2 = e_2 | \rho = \wp) = \frac{\eta_{e_1 \rightsquigarrow e_2}(\wp)}{\binom{L(\wp)}{2}} = \frac{2\eta_{e_1 \rightsquigarrow e_2}(\wp)}{L(\wp)(L(\wp)-1)}, \tag{7.37}$$

where $\eta_{e_1 \rightsquigarrow e_2}(\wp)$ denotes the number of possibilities of drawing edges e_1 and e_2 from $\wp$ in that order.

Again, the full probability of observing the edge pair is obtained by marginalizing out ρ:

$$\mathcal{L}(\beta \mid (e_1, e_2)) = \sum_{\wp \in \mathcal{P}_{st}} \mathrm{P}_{st}(\varepsilon_1 = e_1, \varepsilon_2 = e_2, \rho = \wp) \tag{7.38}$$

$$= \sum_{\wp \in \mathcal{P}_{st}} \mathrm{P}_{st}(\wp)\mathrm{P}_{st}(\varepsilon_1 = e_1, \varepsilon_2 = e_2 | \rho = \wp) \tag{7.39}$$

$$= \sum_{\wp \in \mathcal{P}_{st}} \mathrm{P}_{st}(\wp)\frac{\eta_{e_1 \rightsquigarrow e_2}(\wp)}{\binom{L(\wp)}{2}} = \frac{1}{\mathcal{Z}_{st}} \sum_{\wp \in \mathcal{P}_{st}} \frac{\widetilde{w}(\wp)\eta_{e_1 \rightsquigarrow e_2}(\wp)}{\binom{L(\wp)}{2}} \tag{7.40}$$

The exact derivation of the computation of the above quantity is a bit cumbersome and is presented in Appendix 7.B. Here we only give an intuitive justification of the method based on the idea of copying the graph, explained for the one-edge case earlier, in Section 7.3.1.

Extending to the case of two observed edges $e_1 = (i_1, j_1)$ and $e_2 = (i_2, j_2)$ follows then by considering a third copy of the original graph, which is attached to the second one with a directed edge from node i_2 of the second copy to node j_2 of the third copy (i.e. node $2n + j_2$ of the extended graph), as illustrated in Figure 7.6. The weighted likelihood matrix of such a graph is given by the $3n \times 3n$ block matrix

$$\mathbf{Q}_{e_1 \rightsquigarrow e_2} = \begin{bmatrix} \widehat{\mathbf{W}} & \mathbf{W}_{i_1 j_1} & \mathbf{0} \\ \mathbf{0} & \widehat{\mathbf{W}} & \mathbf{W}_{i_2 j_2} \\ \mathbf{0} & \mathbf{0} & \widehat{\mathbf{W}} \end{bmatrix}. \tag{7.41}$$

This time, the enumeration and likelihood of the paths that at some point traverse edges e_1 and e_2 is given by element $(s, 2n + t)$ of the powers of matrix $\mathbf{Q}_{e_1 \rightsquigarrow e_2}$.

Formally, the 2-edge likelihood can be written using matrix $\mathbf{Q}_{e_1 \rightsquigarrow e_2}$, continuing from (7.40):

$$\sum_{\wp \in \mathcal{P}_{st}} \frac{\widetilde{w}(\wp)\eta_{e_1 \rightsquigarrow e_2}(\wp)}{\binom{L(\wp)}{2}} = \sum_{k=2}^{\infty} \sum_{\wp_k \in \mathcal{P}_{st}^k} \frac{\widetilde{w}(\wp)\eta_{e_1 \rightsquigarrow e_2}(\wp)}{\binom{k}{2}} \tag{7.42}$$

$$= \sum_{k=2}^{\infty} \mathbf{e}_s^{\mathsf{T}} \frac{1}{\binom{k}{2}} \mathbf{Q}_{e_1 \rightsquigarrow e_2}^k \mathbf{e}_{2n+t} \tag{7.43}$$

$$= \mathbf{e}_s^{\mathsf{T}} \mathbf{L}_{e_1 \rightsquigarrow e_2} \mathbf{e}_{2n+t}, \tag{7.44}$$

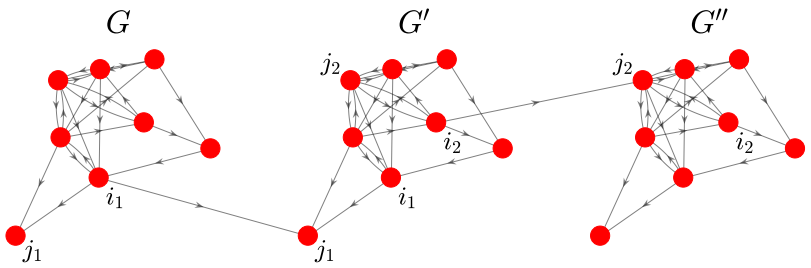


FIGURE 7.6: The computation of the two-edge likelihood can be interpreted as making two copies, G' and G'', of the original graph G and adding the required edges between the copies.

where the sum runs starting from $k = 2$, as we need to only consider paths of length 2 or longer, and where we have defined

$$\mathbf{L}_{e_1 \rightsquigarrow e_2} = \sum_{k=2}^{\infty} \frac{1}{\binom{k}{2}} \mathbf{Q}^k_{e_1 \rightsquigarrow e_2}. \tag{7.45}$$

Note that unlike in the one-edge or one-node likelihood cases, this quantity has no direct connection with the matrix logarithm.

The likelihood is finally given, from Equations (7.40) and (7.44) as

$$\mathcal{L}(\beta \mid (e_1, e_2)) = \frac{[\mathbf{L}_{e_1 \rightsquigarrow e_2}]_{s,2n+t}}{\mathcal{Z}_{st}}. \tag{7.46}$$

Multiple-edge likelihood

The main result of this section is presented here, namely, the computation of likelihood of β given a sequence of multiple observed edges. As stated earlier, the generalization of this result to a sequence of multiple observed nodes follows similarly as the generalization from the one-edge likelihood in Section 7.3.1 to the one-node likelihood in Section 7.3.2. The case of an arbitrary number of observed edges follows naturally from the case of two edges. However, now, we do not assume a fixed number of observations from one path, as was done in the one-edge, one-node, and two-edge likelihood derivations. Instead, the number of observations will be considered as its own random variable.

Let us thus denote by μ the r.v. corresponding to the number of edges that are observed, given a path $\wp$. Similar to the uniformity assumption with respect to the edge and node observation processes, we also assume a uniform distribution for μ, i.e. $\mathrm{P}(\mu = M \mid \rho = \wp) = 1/L(\wp)$ for any $M = 1, \ldots, L(\wp)$.

Summing up the assumptions behind the sampling process, we assume when an incomplete edge-trajectory is observed that the number of edges observed is a uniform random number $\mu \sim \mathrm{unif}\{1, L(\wp)\}$, and then the set of observations is selected uniformly from the edge sequence. The uniform assumptions correspond in this case to maximum entropy modeling, which implies that these assumptions are the most naive and generic ones possible.

One alternative for the uniformity assumption would be to consider a binomial distribution, instead of a uniform one, over the number of observations, $\mu \sim \mathrm{Bin}(L(\wp), p_\mu)$ with the condition that $L(\wp) > 0$. This then would involve the additional parameter, p_μ, i.e. the probability of making an observation at each step, which could be estimated as the relative frequency of observations over the number of steps, whenever such an estimate can be reasonably made.

Further stricter assumptions on μ may make the model more accurate in cases where the actual sampling is known to behave in a certain way, but the naive uniformity assumption can be useful when the sampling process is not known or is very irregular. Such is the case, for instance, with the real data example in Section 7.4, where we estimate the inverse temperature β when fitting the RSP model to a data set of incomplete trajectories of wild reindeer. Although for that data set, the animals' locations are mostly measured at constant time intervals, there are also missing observations causing long gaps between measurements. Also, as the RSP model describes the process in discrete time and discrete space, a constant time interval between observations does not mean a constant number of steps between observations on the graph, as the speed of the animals varies. This can be observed also from the plots of some of the individual trajectories that are presented in Appendix 7.E.

We then proceed with the derivation of the likelihood using the assumption of uniform probability for the number of observations, μ. Given a path $\wp \in \mathcal{P}_{st}$, an observed edge sequence, of length M will be denoted by $\tilde{e} = (e_1, \ldots, e_M)$, where M is drawn according to $\mu \sim \mathrm{unif}\{1, L(\wp)\}$, i.e. $\mathrm{P}(\mu = M \mid \rho = \wp) = 1/L(\wp)$, and where $e_m = (i_m, j_m) \in E$ and $i_m \neq t$ for all $m = 1, \ldots, M$, and $j_m \neq t$ for all $m = 1, \ldots, M-1$.

If we then denote by $\tilde{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_M)$ the random vector corresponding to the observed sequence, and the observed individual edges, then we can write the likelihood, in similar fashion to Equations (7.18) and (7.40), however taking μ

into account separately, as

$$\mathcal{L}(\beta \mid \Omega) = \mathrm{P}(\tilde{\varepsilon} = \tilde{e}, \mu = M) \tag{7.47}$$

$$= \sum_{\wp \in \mathcal{P}_{st}} \mathrm{P}_{st}(\wp) \mathrm{P}(\tilde{\varepsilon} = \tilde{e} \mid \mu = M, \rho = \wp) \mathrm{P}(\mu = M \mid \rho = \wp) \tag{7.48}$$

$$= \sum_{\wp \in \mathcal{P}_{st}} \mathrm{P}_{st}(\wp) \frac{\eta_{\tilde{e}}(\wp)}{\binom{L(\wp)}{M}} \cdot \frac{1}{L(\wp)} \tag{7.49}$$

$$= \frac{1}{\mathcal{Z}_{st}} \sum_{k=M}^{\infty} \sum_{\wp_k \in \mathcal{P}_{st}^{(k)}} \frac{\widetilde{w}(\wp_k) \eta_{\tilde{e}}(\wp_k)}{\binom{k}{M} k}, \tag{7.50}$$

where $\eta_{\tilde{e}}(\wp)$ denotes the number of times that the observed edges $e_1, \ldots, e_M$ appear on path $\wp$ in that order, and where the sum is only over paths of length M or longer.

The above can be computed similar to the computation of the other likelihoods dealt with earlier. The computation can be justified using the same graph extension argument as explained in the two-edge likelihood case in Section 7.3.3 and illustrated in Figure 7.6. Namely, we can, again, consider producing M duplicates of the original graph G and connect the duplicates via the edges of the sequence $\tilde{e}$. This extended graph can be described by the extended likelihood matrix

$$\mathbf{Q}_{\tilde{e}} = \begin{bmatrix} \widehat{\mathbf{W}} & \mathbf{W}_{i_1 j_1} & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \widehat{\mathbf{W}} & \mathbf{W}_{i_2 j_2} & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \widehat{\mathbf{W}} & \ddots & & \vdots \\ \vdots & \vdots & \ddots & \ddots & & \mathbf{0} \\ & & & \mathbf{0} & \widehat{\mathbf{W}} & \mathbf{W}_{i_M j_M} \\ \mathbf{0} & \ldots & & \ldots & \mathbf{0} & \widehat{\mathbf{W}} \end{bmatrix}. \tag{7.51}$$

If we also now define

$$\mathbf{L}_{\tilde{e}} = \sum_{k=M}^{\infty} \frac{\mathbf{Q}_{\tilde{e}}^k}{\binom{k}{M} k}, \tag{7.52}$$

then the multiple-edge likelihood is given by

$$\mathcal{L}(\beta \mid \tilde{e}) = \frac{[\mathbf{L}_{\tilde{e}}]_{s, Mn+t}}{\mathcal{Z}_{st}}. \tag{7.53}$$

A more thorough and rigorous proof for this result is omitted for brevity, as it is a direct generalization of the proof for the two-edge likelihood given in Appendix 7.B.

Multiple-node likelihood

The likelihood of β given an incomplete node-trajectory can be computed similarly to the likelihood of the incomplete edge-trajectory. The derivation is similar to the extension from the one-edge likelihood to the one-node likelihood in Section 7.3.2. Namely, if the data set consists of one sequence of M nodes $\Omega = \{\tilde{v}\} = \{(i_1, i_2, \ldots, i_M)\}$, then we define matrix $\mathbf{Q}_{\tilde{v}}$ as in Equation (7.51), however, replacing the matrices $\mathbf{W}_{i_m j_m}$ with the node-based matrices $\mathbf{W}_{\mathrm{r}(i_m)} = \mathbf{I}_{i_m i_m}\mathbf{W}$, which contain row i_m of $\mathbf{W}$ on row i_m and zeros elsewhere (see Equation (7.36)). Likewise, we define matrix $\mathbf{L}_{\tilde{v}}$ equivalently to Equation (7.52), by replacing matrix $\mathbf{Q}_{\tilde{e}}$ by matrix $\mathbf{Q}_{\tilde{v}}$.

Again, as in the one-node likelihood case, in Section 7.3.2, the possibility of the first node being s must be considered separately. As was done there, we here also make the assumption that the first observed node can be s, but it cannot be the first node, $\wp(0)$ of the observed path $\wp$, i.e. it can only be a revisit to node s after the first step of the path.

With these assumptions, the multiple-node likelihood is given, similar to Equations (7.35) and (7.53) by

$$\mathcal{L}(\beta \mid \tilde{v}) = \frac{1}{\mathcal{Z}_{st}} \sum_{h \in Succ(s)} w_{su} [\mathbf{L}_{\tilde{v}}]_{h, Mn+t}. \tag{7.54}$$

Validation of MLE with multiple observations

We evaluated the applicability and precision of the MLE method in the case of incomplete trajectories from s to t. We only considered incomplete node trajectories and the corresponding method for computing the MLE. The setting was similar to the experiment for validating the full trajectory MLEs in Section 7.2. Namely, we again generated 200 paths for different values of β between uniformly distributed s-t-pairs, where s and t were at least 3 steps apart from each other. We, again, used the two grid graphs that were used in Section 7.2, i.e. one with uniform costs, and another based on a simulated Gaussian landscape.

From the 200 generated paths, we extract a set Ω of 200 incomplete node trajectories by sampling M nodes, where $M = \min(300, \hat{M})$, and $\hat{M}$ is drawn uniformly from $1, \ldots, L(\wp)$. We limit the maximum length of an incomplete trajectory to 300 for computational efficiency. We then compute $\hat{\beta}_{\mathrm{MLE}}$ by performing a line search on β for finding the maximum value of $\log \mathcal{L}(\beta \mid \Omega)$. This

β	$\hat{\beta}_{\mathrm{MLE}}$, uniform grid	$\hat{\beta}_{\mathrm{MLE}}$, simulated landscape
0.001	0.00101 ± 0.00016	0.00106 ± 0.00013
0.005	0.00497 ± 0.00043	0.00510 ± 0.00069
0.01	0.00980 ± 0.00070	0.00992 ± 0.00088
0.05	0.05014 ± 0.00275	0.05091 ± 0.00250
0.1	0.10117 ± 0.00704	0.09433 ± 0.00807
0.5	0.50810 ± 0.03167	0.49411 ± 0.02237
1	1.01074 ± 0.07147	0.98349 ± 0.03991
5	4.92557 ± 0.27878	4.93601 ± 0.24915
10	12.73153 ± 5.07237	10.08324 ± 0.28333

TABLE 7.2: The MLEs in the experiment with incomplete node trajectories. The first column on the left shows the value of β used for generating the paths and the two other columns show the mean $\pm$ the standard deviation of the MLEs over 10 repetitions.

is again repeated 10 times and we report the mean and standard deviation of the 10 obtained MLE values.

The results are gathered in Table 7.2. They show that the MLEs are accurate throughout the tested range of values of β and thus confirm that the methodology derived in Section 7.3 can be used in practice. Moreover, quite surprisingly, the results are almost as accurate as the results in Table 7.1 in the experiment with full trajectories of Section 7.2.2.

7.4 Application to animal movement modelling

In order to test the applicability of the method for computing the MLE of β in a real data setting, we ran the MLE method for a set of GPS trajectory data collected from individual wild mountain reindeer in the Austhei area of Norway. The reason for studying this data is based on the apparent suitability of RSPs for modelling animal movement. This holds especially for the study of wild reindeer, which are a migratory species, that often have a good sense an knowledge of their environment. Accordingly, they can be considered to follow fairly optimal routes when moving in a landscape. It would be, however, unrealistic to assume that the animals always follow only the optimal path on a static landscape. Instead their movement decisions may be considered to involve some randomness. This is exactly the kind of scenario that the RSP framework is designed for. However, the aim of the experiments reported here was simply to test whether the MLEs are practical to compute for real data with the methods derived above and to see whether the estimates obtained seem sensible. We leave a more careful analysis of the obtained results, and their

applicability in more focused ecological problems, such as actual prediction of movement, for future work.

The data considered here consists of incomplete trajectories recorded during springtime migration, when the reindeer traverse the landscape from north to south crossing over a road passage cutting through the landscape. The same area and partly the same data was studied previously in the context of RSPs by Panzacchi et al. [168], although there the graph was constructed differently compared to the experiment presented here. Panzacchi et al. [168] also devised a method for estimating the inverse temperature parameter, however based on a very different, more situation-dependent approach, compared to the more generic method introduced in this work.

The landscape is modeled as a rectangular grid graph, where each node represents a 500 meter wide square pixel of a map image of the landscape. The size of the landscape is 80.5 km $\times$ 46 km, resulting in a graph with 14812 nodes. As before, each pixel was again connected to its 8 surrounding pixels. First, the edge affinities a_{ij} were defined as Step Selection Probabilities (SSP) [133], inferred by detecting features on each pixel with remote sensing methods and by measuring the preference of those features for movement based on the reindeer GPS data. A more detailed explanation of how the affinities were defined is provided in Appendix 7.D. The edge costs were defined simply as the reciprocals of affinities, i.e. $c_{ij} = 1/a_{ij}$. The landscape is illustrated in Figure 7.7A as a heatmap where the pixels are colored according to the average incoming edge affinity of each pixel. In addition, Figure 7.7B visualizes the trajectory data by marking the number of times any of the 32 trajectories were observed at each pixel. Plots of the individual trajectories are presented separately in Appendix 7.E.

The data contains 32 incomplete trajectories recorded during the years 2007-2013 from 14 different individuals. The GPS measurements were made for the most part every 3 hours, 50 % of trajectories contained all locations. However, many trajectories contain gaps of 6 hours to one day, and one trajectory is missing locations for nearly nearly two weeks. Trajectories started in the winter range between March 1st and 20th, and ended in the summer range between June 8th and 30th. We cut the original trajectories after they have crossed a certain line which can be interpreted as a border of the summer range. The number of observations in the resulting trajectories was between 57 and 535. For each observation, we detect the pixel that the animal is in, and use the method derived in Section 7.3 for computing the multiple-node likelihood. We consider the trajectories as extracted from hitting paths, where the hitting node is selected to be the pixel where the animal is observed after crossing to the summer range.

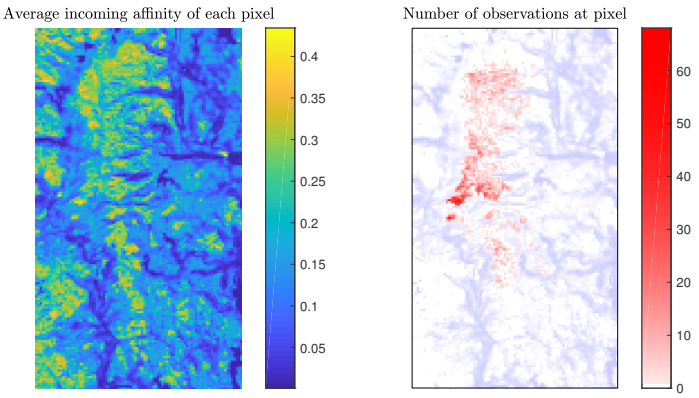


FIGURE 7.7: Visualization of the landscape used for studying wild reindeer movement (a). The heatmap shows the average incoming edge affinity of each pixel. The heatmap of number of times an animal was observed at each pixel based on the GPS data (b). The blue hue in (b) marks the low-affinity (i.e. high cost) areas (in order to give an idea of the shape of the landscape) whereas the red hue marks the number of times an animal was observed at each pixel.

We first computed the MLE for each trajectory separately, resulting in 32 MLE values. In addition, we considered the whole set of trajectories as being generated by the same value of β, and thus computed the population-wide MLE value using the whole set of trajectories. For this, we considered the likelihood of the set as the product of the likelihoods of each trajectory, considering the trajectories as independent.

The individual trajectory MLEs are presented in Appendix 7.E. The MLEs remained for the most part in a fairly consistent range of values, between $[0.001, 0.01]$, with one exception, trajectory number 19, obtaining a MLE below this range, namely $\hat{\beta}_{\mathrm{MLE}}(\wp_{19}) = 0.0004715$. The mean of the individual MLEs was $\langle \hat{\beta}_{\mathrm{MLE}}(\wp) \rangle_{\wp \in \Omega} = 0.003311$. The MLE given by the whole set of trajectories was $\hat{\beta}_{\mathrm{MLE}}(\Omega) = 0.002466$.

Figure 7.8 contains plots for the expected numbers of visits according to the RSP model with different values of β. These were computed by considering the starting and ending nodes, s and t, of each trajectory in the data separately, computing the expected number of visits to each pixel for that s-t-pair, and by summing the contributions of each s-t-pair. In each plot, the blue hue

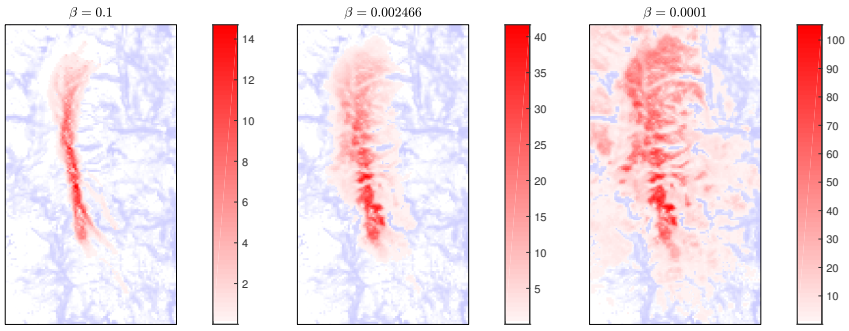


FIGURE 7.8: The expected number of visits to each pixel over the RSP probability distributions between each observed s-t-pair, with three values of $\beta = 0.1$ (a), $\hat{\beta}_{\mathrm{MLE}}(\Omega) = 0.002466$ (b; the MLE based on the set of trajectories Ω), and $\beta = 0.0001$ (c). The red hue at pixel i marks the value $\sum_{(s,t)\in X} \bar{n}_i(s,t)$, given the value of β depicted above the plot. Here, X denotes the set of s-t-pairs of the trajectories in the data set Ω.

represents the average incoming edge cost of each pixel, and the transparent red hue represents the observed or expected number of visits by an animal to a pixel.

Figure 7.8A corresponds to the model computed with $\beta = 0.1$, and Figure 7.8C with $\beta = 0.0001$ for each s-t-pair. In between, Figure 7.8B shows the result with the value $\beta = 0.002466$, which is the MLE estimated by using the whole set of trajectories. We also constructed another plot by using for each s-t-pair the MLE of β of the corresponding trajectory, but it is not presented here, because it resembles almost exactly the plot obtained with the global MLE in Figure 7.8B.

A visual inspection of the different plots obtained with the RSP distribution using different values of β indicates that the model using the MLE values of β has most resemblance with the plot of the actual observations. Note, however, that the plots in Figure 7.8 are not exactly expected to resemble the plot in Figure 7.7B. Namely, as explained before, the RSP model does not take into account the temporal aspect of the movement. The temporal aspect, however, is grained in the trajectories, as the speed of movement and the time interval between observations vary. For instance, after crossing the road over to the south side, the animals tend to move faster away from the road [165]. This results generally in a fewer number of observations in the pixels south of the

road. In other words, Figure 7.7B rather depicts the *time spent* at each pixel by the animals. Instead, the plots in Figure 7.8, given by the RSP model, reflect the importance of each pixel as an intermediate point of movement, which can be more crucial, for instance, for detecting corridors and barriers in a landscape.

7.5 Further developments

The computational tools, and especially, the new partition function, $\widetilde{\mathcal{Z}}_{st}$, developed in Section 7.3 for the MLEs, can be used for other purposes too. One interesting idea is to consider the average edge cost over each path from s to t and to compute the expected value of the average edge costs over the set $\mathcal{P}_{st}$ with respect to the RSP distribution.

7.5.1 Average edge cost over paths

The average edge cost of a path $\wp$ is $\bar{c}(\wp) = \tilde{c}(\wp)/L(\wp)$, and thus the expected average edge cost is

$$\mathbb{E}_{st}[\bar{c}] = \sum_{\wp \in \mathcal{P}_{st}} \mathrm{P}_{st}(\wp) \frac{c(\wp)}{L(\wp)}. \tag{7.55}$$

This can be interpreted as a measure of the overall difficulty of moving from s to t.

The computation of $\mathbb{E}[\bar{c}]$ resembles the computation of the single-edge likelihood in Equations (7.18)-(7.21), involving however the derivative w.r.t. β, instead of c_{ij}. Namely, we can write

$$\mathbb{E}_{st}[\bar{c}] = \frac{1}{\mathcal{Z}_{st}} \sum_{\wp \in \mathcal{P}_{st}} \widetilde{w}(\wp) \frac{c(\wp)}{L(\wp)} \tag{7.56}$$

$$= -\frac{1}{\mathcal{Z}_{st}} \frac{\partial}{\partial \beta} \widetilde{\mathcal{Z}}_{st} \tag{7.57}$$

where, again, $\widetilde{\mathcal{Z}}_{st}$ is the altered partition function given by element (s,t) of the matrix logarithm $-\mathbf{log}(\mathbf{I} - \widehat{\mathbf{W}})$, as in Equations (7.22) and (7.23), where, again, $\widehat{\mathbf{W}}$ is obtained from matrix $\mathbf{W}$ by setting row t to zero.

The derivative of $\widetilde{\mathcal{Z}}_{st}$ w.r.t. β can be computed in a similar fashion to the computation of the derivative w.r.t. c_{ij} in Section 7.3. This time, we have

$$\frac{\partial}{\partial \beta}\widehat{\mathbf{W}} = -\mathbf{C} \circ \widehat{\mathbf{W}} \tag{7.58}$$

Thus, if we define the $2n \times 2n$ block matrix

$$\mathbf{Q}_\beta = \begin{bmatrix} \widehat{\mathbf{W}} & -\mathbf{C} \circ \widehat{\mathbf{W}} \\ \mathbf{0} & \widehat{\mathbf{W}} \end{bmatrix}, \tag{7.59}$$

then we can obtain element (s, t) of the derivative of $-\mathbf{log}(\mathbf{I} - \widehat{\mathbf{W}})$ this time as element $(s, n + t)$ of the matrix

$$\mathbf{L}_\beta = \sum_{k=1}^{\infty} \frac{\mathbf{Q}_\beta^k}{k} = -\mathbf{log}(\mathbf{I} - \mathbf{Q}_\beta), \quad \text{i.e.} \quad \frac{\partial}{\partial_\beta}\widetilde{\mathcal{Z}}_{st} = [\mathbf{L}_\beta]_{s,n+t}. \tag{7.60}$$

Accordingly, the expected average edge cost over RSPs from s to t can be computed as

$$\mathbb{E}_{st}[\bar{c}] = -\frac{[\mathbf{L}_\beta]_{s,n+t}}{\mathcal{Z}_{st}}. \tag{7.61}$$

Note, however, that as we consider hitting paths with absorbing target node t, the matrices $\mathbf{Q}_\beta$ and $\mathbf{L}_\beta$ also depend on t. Thus, the above quantity can only be computed for one target node t at a time.

Instead, if we consider the expected average edge cost over *all* paths, not only hitting ones, then the matrix of the values $\mathbb{E}_{st}(\bar{c})$ between all s-t-pairs can be computed in batch form. This can be done simply by replacing matrix $\widehat{\mathbf{W}}$, which depends on t, with matrix $\mathbf{W}$ in Equations (7.59-7.60). Moreover, the partition function, $\mathcal{Z}_{st}$, in Equation (7.61) should be replaced by the partition function of all paths, $\mathcal{Z}_{st}^{\mathrm{A}} = \sum_{\wp \in \mathcal{P}_{st}^{\mathrm{N}}} \widetilde{w}(\wp)$, where $\mathcal{P}_{st}^{\mathrm{N}}$ is the set of all paths, not only hitting ones. The partition functions of all paths are the elements of $\mathbf{Z}$, i.e. the fundamental matrix of all paths, from Equation (2.19).

This means that we can compute the matrix containing the expected average costs over all paths, between all s-t-pairs as elements (s, t) of

$$\mathbf{E} = [\mathbf{L}_\beta]_{\mathbf{1},\mathbf{2}} \div \mathbf{Z}, \tag{7.62}$$

where $\mathbf{L}_\beta$ is now defined based on $\mathbf{W}$, instead of $\widehat{\mathbf{W}}$, and where $[\mathbf{L}_\beta]_{\mathbf{1},\mathbf{2}}$ denotes the block $(1, 2)$ of size $(n \times n)$ of $\mathbf{L}_\beta$, i.e. the submatrix on rows $[1, \ldots, n]$ and

columns $[n+1, \ldots, 2n]$.

7.6 Discussion

This chapter focused on the estimation of parameters when fitting the RSP model to data containing trajectories on a network, with most focus on the estimation of the inverse temperature parameter β. Methods were derived for computing and maximizing the likelihood of values of β given a data set of either fully or only partly observed trajectories. The methods were shown to provide accurate estimates of β with artificial examples where trajectories were generated on simulated geographic grid graphs. In addition to estimation of β, also maximum likelihood estimates of the edge costs, $c_{ij}, (i,j) \in E$, of a graph were derived and evaluated.

The MLE method derived for incomplete trajectories was also tested on real data of trajectories of wild mountain reindeer on a landscape area in Norway. The purpose of the experiment was only to verify further that the MLEs of β can be computed in practice and that the estimates seem reasonable and sensible. This indicates that the MLE method provides a well-founded and functional way for model fitting, when using the RSP model for more specific applications related to animal movement.

As an additional sidestep of the chapter, in Section 7.5 the techniques developed for the computation of the MLE of β with incomplete trajectories were used for also computing the average edge cost of paths between two nodes, which provides a measure of the ease of connection between node pairs on the graph. This idea was only briefly presented and its use in applications, such as Markov Decision processes, is left for future research.

Appendix: Additional material and proofs of main results of the chapter

7.A Derivation of the one-node likelihood

This section contains the derivation of the one-node likelihood, presented in Equation (7.35). Again, let ρ be the random variable corresponding to the path drawn from the set $\mathcal{P}_{st}$ according to the RSP distribution (Equation (2.10)). Given a path $\wp = (\wp(0) = s, \wp(1), \ldots, \wp(L(\wp)) = t)$, let us consider two random variables,

- ν_0, corresponding to an intermediate node drawn uniformly from the subsequence of $\wp$ excluding the last node, but *including the first node*, i.e. $(\wp(0), \ldots, \wp(L(\wp) - 1))$, and
- ν_1 corresponding to an intermediate node drawn uniformly from the subsequence of $\wp$ *excluding both the first and last node*, i.e. $(\wp(1), \ldots, \wp(L(\wp) - 1))$.

Recalling the assumptions stated in Section 7.3.2, the likelihood is given by the probability distribution of ν_1. However, we first derive the distribution of ν_0 as an intermediate result and use it to derive the distribution of ν_1.

Using $\eta_i(\wp) = \sum_{j \in Succ(i)} \eta_{ij}(\wp)$, as earlier, the conditional probability of node i as the outcome of ν_0, given a path $\wp \in \mathcal{P}_{st}$, is

$$\mathrm{P}(\nu_0 = i \mid \rho = \wp) = \frac{\bar{\eta}_i(\wp)}{L(\wp)} = \frac{\sum_{j \in Succ(i)} \bar{\eta}_{ij}(\wp)}{L(\wp)}. \tag{7.63}$$

Then, marginalizing out ρ, we can write the distribution of ν_0, in similar fashion to Equations (7.18)-(7.21) for the one-edge likelihood, as

$$\mathrm{P}(\nu_0 = i; \beta) = \sum_{\wp \in \mathcal{P}_{st}} \mathrm{P}(\rho = \wp)\mathrm{P}(\nu_0 = i | \rho = \wp) \tag{7.64}$$

$$= \sum_{\wp \in \mathcal{P}_{st}} \mathrm{P}_{st}(\wp) \frac{n_i(\wp)}{L(\wp)} \tag{7.65}$$

$$= \frac{1}{\mathcal{Z}_{st}} \sum_{j \in Succ(i)} \sum_{\wp \in \mathcal{P}_{st}} \widetilde{w}(\wp) \frac{\eta_{ij}(\wp)}{L(\wp)} \tag{7.66}$$

$$= -\frac{1}{\beta \mathcal{Z}_{st}} \sum_{j \in Succ(i)} \frac{\partial}{\partial c_{ij}} \widetilde{\mathcal{Z}}_{st}. \tag{7.67}$$

where $Succ(i)$ is the set of successor nodes of i. This equation is the equivalent of Equation (7.21) for edges. Now, similar to Equation (7.25), we have, for any $k \geq 1$

$$\sum_{j \in Succ(i)} \frac{\partial}{\partial c_{ij}} \widehat{\mathbf{W}}^k = -\beta \sum_{j \in Succ(i)} \sum_{l=1}^{k} \widehat{\mathbf{W}}^{l-1} \mathbf{W}_{ij} \mathbf{W}^{k-l} \tag{7.68}$$

$$= -\beta \sum_{l=1}^{k} \widehat{\mathbf{W}}^{l-1} \left(\sum_{j \in Succ(i)} \mathbf{W}_{ij} \right) \mathbf{W}^{k-l} \tag{7.69}$$

$$= -\beta \sum_{l=1}^{k} \widehat{\mathbf{W}}^{l-1} \mathbf{W}_{\mathrm{r}(i)} \widehat{\mathbf{W}}^{k-l} \tag{7.70}$$

$$\triangleq -\beta \mathbf{S}_i^{(k)}, \tag{7.71}$$

where $\mathbf{W}_{\mathrm{r}(i)} = \mathbf{I}_{ii}\mathbf{W}$ is the matrix containing the i-th row of matrix $\mathbf{W}$ on its i-th row, but zeros elsewhere, and $\mathbf{S}_i^{(k)}$ defined analogously to $\mathbf{S}_{ij}^{(k)}$ in Equation (7.26).

The rest of the derivation of the distribution of ν_0 proceeds as in the single-edge likelihood case, but with matrix $\mathbf{W}_{\mathrm{r}(i)}$ in place of matrix $\mathbf{W}_{ij}$. Namely, the derivative expression appearing in Equation (7.67) can be computed (as in Equations (7.29)-(7.33)), as

$$\sum_{j \in Succ(i)} \frac{\partial}{\partial c_{ij}} \widetilde{\mathcal{Z}}_{st} = \mathbf{e}_s^{\mathsf{T}} \sum_{k=1}^{\infty} \frac{\mathbf{S}_i^{(k)}}{k} \mathbf{e}_t, \tag{7.72}$$

which, again, can be computed as element $(s, n+t)$ of the matrix logarithm $\mathbf{L}_i = -\mathbf{log}(\mathbf{I} - \mathbf{Q}_i)$, with the $(2n \times 2n)$ block matrix (see Equation (7.31))

$$\mathbf{Q}_i = \begin{bmatrix} \widehat{\mathbf{W}} & \mathbf{W}_{\mathrm{r}(i)} \\ \mathbf{0} & \widehat{\mathbf{W}} \end{bmatrix}. \tag{7.73}$$

In conclusion, we can write the distribution of ν_0 as

$$\mathrm{P}(\nu_0 = i) = -\frac{1}{\beta \mathcal{Z}_{st}}(-\beta[\mathbf{L}_i]_{s,n+t}) = \frac{[\mathbf{L}_i]_{s,n+t}}{\mathcal{Z}_{st}}. \tag{7.74}$$

Using the above, we can then derive the one-node likelihood according to the distribution of ν_1. We do this by considering a split of each path $\wp \in \mathcal{P}_{st}$ into a path consisting of the first step, from s to one of its successor nodes $u \in Succ(s)$, and the rest of the path, from u to t. We denote, generally, by $\wp_u \in \mathcal{P}_{st}$ a path from s to t whose second node is successor $u \in Succ(s)$, and by $\wp'_u \in \mathcal{P}_{ut}$ the remainder of path $\wp_u$ after the first step. The likelihood is then given by

$$\begin{aligned}
\mathcal{L}(\beta \mid i) = \mathrm{P}(\nu_1 = i; \beta) &= \sum_{\wp \in \mathcal{P}_{st}} \mathrm{P}_{st}(\wp)\mathrm{P}(\nu_1 = i \mid \rho = \wp) && (7.75)\\
&= \frac{1}{\mathcal{Z}_{st}} \sum_{\wp \in \mathcal{P}_{st}} \widetilde{w}(\wp)\mathrm{P}(\nu_1 = i \mid \rho = \wp) && (7.76)\\
&= \frac{1}{\mathcal{Z}_{st}} \sum_{u \in Succ(s)} \sum_{\substack{\wp_u \in \mathcal{P}_{st} \\ \wp(1)=u}} w(\wp_u)\mathrm{P}(\nu_1 = i \mid \rho = \wp_u) && (7.77)\\
&= \frac{1}{\mathcal{Z}_{st}} \sum_{u \in Succ(s)} w_{su} \sum_{\wp'_u \in \mathcal{P}_{ut}} w(\wp'_u)\mathrm{P}(\nu_0 = i \mid \rho = \wp'_u) && (7.78)\\
&= \frac{1}{\mathcal{Z}_{st}} \sum_{u \in Succ(s)} w_{su}\mathrm{P}(\nu_0 = i)\mathcal{Z}_{ut} && (7.79)\\
&= \frac{1}{\mathcal{Z}_{st}} \sum_{u \in Succ(s)} w_{su}[\mathbf{L}_i]_{u,n+t}, && (7.80)
\end{aligned}$$

where we used the fact that $\mathrm{P}(\nu_0 = i) = \dfrac{1}{\mathcal{Z}_{ut}} \sum_{\wp'_u \in \mathcal{P}_{ut}} w(\wp'_u)\mathrm{P}(\nu_0 = i \mid \rho = \wp'_u)$ (as in Equation (7.66)), where $\mathrm{P}(\nu_0 = i)$ is based on considering paths from u, instead of s, to t. This is the desired result, expressed in Equation (7.35).

7.B Derivation of the two-edge likelihood

Here we write a more rigorous derivation, compared to the heuristic justification of Section 7.3.3, for the computation of the two-edge likelihood, i.e. the likelihood of observing two edges $e_1 = (i_1, j_1), e_2 = (i_2, j_2)$ along path $\wp$. We begin by recalling the form presented in Equation (7.40) for the likelihood:

$$\begin{aligned}\mathcal{L}(\beta \mid (e_1, e_2)) &= \frac{1}{\mathcal{Z}_{st}} \sum_{\wp \in \mathcal{P}_{st}} \frac{\widetilde{w}(\wp)\eta_{e_1 \rightsquigarrow e_2}(\wp)}{\binom{L(\wp)}{2}} \\ &= \frac{1}{\mathcal{Z}_{st}} \sum_{k=2}^{\infty} \frac{1}{\binom{k}{2}} \sum_{\wp_k \in \mathcal{P}_{st}^{(k)}} \widetilde{w}(\wp_k)\eta_{e_1 \rightsquigarrow e_2}(\wp_k).\end{aligned} \tag{7.81}$$

Note that the proof for the case of an arbitrary number of observed edges could be derived in a similar way. However, here we deal only with the two-edge case for conciseness and clarity.

Now, $\eta_{e_1 \rightsquigarrow e_2}$ can be calculated by iterating over all edges of the path, checking if the edge corresponds to e_1, and then computing the number of times e_2 appears on the path after that. Summing the occurences of e_2 after each occurence of e_1 then gives $\eta_{e_1 \rightsquigarrow e_2}$. Formally, for any $\wp_k \in \mathcal{P}_{st}^{(k)}$ with $k \geq 2$, where $\wp_k = (\wp(0), \wp(1), \ldots, \wp(k))$, we have

$$\eta_{e_1 \rightsquigarrow e_2}(\wp_k) = \sum_{l=0}^{k-2} [\wp_k(l, l+1) = e_1] \, \eta_{e_2}(\wp_k(l+1 : k)), \tag{7.82}$$

where $\wp_k(l_1 : l_2) = (\wp(l_1), \ldots, \wp(l_2))$ denotes the subpath of $\wp_k$ from the l_1-th node to the l_2-th node, with $0 \leq l_1 < l_2 \leq k$, and the brackets $[\,]$ are the Iverson brackets [122], i.e. 1 if the statement within is true and 0 otherwise.

Using the above, the sum appearing in Equation (7.81) can be expressed, for path length $k \geq 2$, in matrix form:

$$\begin{aligned}
&\sum_{\wp_k \in \mathcal{P}_{st}^{(k)}} \widetilde{w}(\wp_k)\eta_{e_1 \rightsquigarrow e_2}(\wp_k) \\
=&\sum_{\wp_k \in \mathcal{P}_{st}^{(k)}} \sum_{l=0}^{k-2} w(\wp_k(0:l))\,[\wp_k(l,l+1) = e_1] \\
&\qquad \times w_{i_1 j_1} w(\wp_k(l+1:k))\eta_{e_2}(\wp_k(l+1:k)) \\
=&\sum_{l=0}^{k-2} \sum_{\wp_k \in \mathcal{P}_{st}^{(k)}} w(\wp_k(0:l))\,[\wp_k(l) = i_1)] \\
&\qquad \times w_{i_1 j_1}\,[\wp_k(l+1) = j_1] \left(-\frac{1}{\beta}\frac{\partial}{\partial c_{i_2 j_2}} w(\wp_k(l+1:k))\right) \\
=&-\frac{1}{\beta}\sum_{l=0}^{k-2} \mathbf{e}_s^{\mathsf{T}} \widehat{\mathbf{W}}^l \mathbf{e}_{i_1} w_{i_1 j_1} \mathbf{e}_{j_1}^{\mathsf{T}} \left(\frac{\partial}{\partial c_{i_2 j_2}} \widehat{\mathbf{W}}^{k-l-1}\right)\mathbf{e}_t \\
=&\frac{1}{\beta^2}\sum_{l=0}^{k-2} \mathbf{e}_s^{\mathsf{T}} \widehat{\mathbf{W}}^l \left(\frac{\partial}{\partial c_{i_1 j_1}} \widehat{\mathbf{W}}\right)\left(\frac{\partial}{\partial c_{i_2 j_2}} \widehat{\mathbf{W}}^{k-l-1}\right)\mathbf{e}_t.
\end{aligned} \tag{7.83}$$

But in fact, the matrix appearing above

$$\mathbf{S}^{(k)}_{e_1 \rightsquigarrow e_2} = \frac{1}{\beta^2}\sum_{l=0}^{k-2} \widehat{\mathbf{W}}^l \left[\frac{\partial}{\partial c_{i_1 j_1}} \widehat{\mathbf{W}}\right]\left[\frac{\partial}{\partial c_{i_2 j_2}} \widehat{\mathbf{W}}^{k-l-1}\right] \tag{7.84}$$

can be computed as the $(\mathbf{1}, \mathbf{3})$-block of the k-th power of matrix $\mathbf{Q}_{e_1 \rightsquigarrow e_2}$, from Equation (7.41), for any $k \geq 2$. For this, note that we can write

$$\mathbf{Q}_{e_1 \rightsquigarrow e_2} = \begin{bmatrix} \widehat{\mathbf{W}} & \widehat{\mathbf{W}}_{i_1 j_1} & \mathbf{0} \\ \mathbf{0} & \widehat{\mathbf{W}} & \widehat{\mathbf{W}}_{i_2 j_2} \\ \mathbf{0} & \mathbf{0} & \widehat{\mathbf{W}} \end{bmatrix} = \begin{bmatrix} \widehat{\mathbf{W}} & -\frac{1}{\beta}\partial_{c_{i_1 j_1}} \widehat{\mathbf{W}} & \mathbf{0} \\ \mathbf{0} & \widehat{\mathbf{W}} & -\frac{1}{\beta}\partial_{c_{i_2 j_2}} \widehat{\mathbf{W}} \\ \mathbf{0} & \mathbf{0} & \widehat{\mathbf{W}} \end{bmatrix}. \tag{7.85}$$

It can then be shown by induction, that, for all $k \geq 2$,

$$\mathbf{Q}^k_{e_1 \rightsquigarrow e_2} = \begin{bmatrix} \widehat{\mathbf{W}}^k & -\frac{1}{\beta}\partial_{c_{i_1 j_1}} \widehat{\mathbf{W}}^k & \frac{1}{\beta^2}\mathbf{S}^{(k)}_{e_1 \rightsquigarrow e_2} \\ \mathbf{0} & \widehat{\mathbf{W}}^k & -\frac{1}{\beta}\partial_{c_{i_2 j_2}} \widehat{\mathbf{W}}^k \\ \mathbf{0} & \mathbf{0} & \widehat{\mathbf{W}}^k \end{bmatrix}. \tag{7.86}$$

We omit the proof for brevity.

Using the above, we can finally write the two-edge likelihood, continuing from Equation (7.81) as

$$\mathcal{L}(\beta \mid (e_1, e_2)) = \frac{1}{\mathcal{Z}_{st}} \mathbf{e}_s^\mathsf{T} \sum_{k=2}^{\infty} \frac{1}{\binom{k}{2}} \mathbf{Q}^k_{e_1 \rightsquigarrow e_2} \mathbf{e}_{2n+t} = \frac{[\mathbf{L}_{e_1 \rightsquigarrow e_2}]_{s,2n+t}}{\mathcal{Z}_{st}}, \tag{7.87}$$

where $\mathbf{L}_{e_1 \rightsquigarrow e_2}$ is as defined in Equation (7.45).

7.C Practical computation of likelihood for incomplete trajectories

In order to make computation of the likelihoods for incomplete trajectories, presented in Section 7.3, feasible with large graphs, a few tricks are required. Here we deal only with the computation in the case of incomplete edge-trajectories. The derivations for the incomplete node-trajectories follow fairly directly, although they are slightly complicated because of the more complex expression of the multiple-node likelihood in Equation (7.54) compared to the expression for multiple-edge likelihood in Equation (7.53).

Computation of $[\mathbf{L}_{ij}]_{s,n+t}$. According to Equation (7.34) the likelihood of detecting a particular edge involves computing an element of the matrix logarithm $\mathbf{L}_{ij} = \mathbf{log}(\mathbf{I} - \mathbf{Q}_{ij})$. The current standard for matrix logarithm computations is the algorithm developed by Al-Mohy et al. [2], which is implemented, for instance, in Matlab and SciPy as the `logm` function. However, we are only interested in computing one element of the matrix logarithm, and for this purpose the algorithm of Al-Mohy et al. [2] is not reliable enough numerically. More exactly, the algorithm produces inaccurate values of the element $(s, n+t)$ of matrix $\mathbf{L}_{ij}$ (see Equation (7.33)), when β is relatively large, causing the values $[\mathbf{L}_{ij}]_{s,n+t}$ to be very small. To get more reliable values, we compute the matrix logarithm element iteratively based on the power series expression of

the logarithm, by computing a finite sum according to the power series in Equation (7.23).

The same procedure can be used for computing elements of matrix $[\mathbf{L}_{\tilde{e}}]_{s,Mn+t}$ in the general case of an incomplete edge trajectory $\tilde{e}$ with multiple observed edges, which does not correspond to a matrix logarithm. The procedure is detailed in the following.

Computation of $[\mathbf{L}_{\tilde{e}}]_{s,Mn+t}$. Remember, from Equation (7.53) that we are eventually only interested in one element of matrix $\mathbf{L}_{\tilde{e}}$, i.e. the sum of the corresponding elements of matrices $\mathbf{Q}_{\tilde{e}}^k/k$. We compute the elements of $\mathbf{Q}_{\tilde{e}}^k$ by a backward iteration, updating iteratively column $Mn+t$ of the matrix powers according to the power method

$$\mathbf{q}_{Mn+t}^{(0)} \longleftarrow \mathbf{e}_{Mn+t}, \tag{7.88}$$

$$\mathbf{q}_{Mn+t}^{(k+1)} \longleftarrow \mathbf{Q}_{\tilde{e}}\mathbf{q}_{Mn+t}^{(k)}, \quad k = 1, 2, \ldots, \tag{7.89}$$

where $\mathbf{e}_{Mn+t}$ is the column vector of length $(M+1)n$ with 1 at element $Mn+t$ and zero elsewhere. This can be interpreted as tracing all (s,t)-paths backwards from t to s through the observed edges in $\tilde{e}$ and computing the cumulated likelihoods.

However, construction of the matrix $\mathbf{Q}_{\tilde{e}}$ and direct computations with it would be quite memory-restrictive when the graph size becomes large. This can be alleviated by taking advantage of the sparse block-diagonal form of the matrix and breaking the computation into parts. Namely, if we denote with $\hat{\mathbf{q}}_l^{(k)}$ the l-th segment of length n of vector $\mathbf{q}_{Mn+t}^{(k)}$, i.e. the section of the vector between elements $(l-1)n$ and ln, then we can write the update of Equation (7.89) as

$$\hat{\mathbf{q}}_l^{(k+1)} \longleftarrow \begin{cases} \mathbf{W}\hat{\mathbf{q}}_l^{(k)} + \mathbf{W}_{i_l j_l}\hat{\mathbf{q}}_{l+1}^{(k)}, & l = 1, \ldots, M \\ \mathbf{W}\hat{\mathbf{q}}_l^{(k)} & l = M+1. \end{cases} \tag{7.90}$$

The update can be simplified even further by stacking the vector $\mathbf{q}_{Mn+t}^{(k)}$ as an $(n \times (M+1))$-matrix $\hat{\mathbf{Q}}_{\tilde{e}}^{(k)}$, whose l-th column is segment $\hat{\mathbf{q}}_l^{(k)}$ and remembering that the matrices $\mathbf{W}_{i_l j_l}$ contain only one non-zero element, w_{ij}, at position $(i.j)$. Then the update rule for $l = 1, \ldots, M$ in Equation (7.90) can be expressed by

breaking the update into two parts:

$$\hat{\mathbf{Q}}_{\tilde{e}}^{(k+1)} \longleftarrow \mathbf{W}\hat{\mathbf{Q}}_{\tilde{e}}^{(k)}, \tag{7.91}$$

$$\left[\hat{\mathbf{Q}}_{\tilde{e}}^{(k+1)}\right]_{l,i_l} \longleftarrow \left[\hat{\mathbf{Q}}_{\tilde{e}}^{(k+1)}\right]_{l,i_l} + w_{i_l j_l}\left[\hat{\mathbf{Q}}_{\tilde{e}}^{(k)}\right]_{(l+1),j_l}, \quad l = 1, \ldots, M, \tag{7.92}$$

where the two rows correspond to separate computation the two terms of the sum in the update formula in Equation (7.90).

Note the difference between the two matrices, $\mathbf{Q}_{\tilde{e}}^{k}$ and $\hat{\mathbf{Q}}_{\tilde{e}}^{(k)}$ despite almost identical notation: $\mathbf{Q}_{\tilde{e}}^{k}$ is the k-th power of an $((M+1)n \times (M+1)n)$ block matrix, whereas $\hat{\mathbf{Q}}_{\tilde{e}}^{(k)}$ is the $(n \times (M+1))$ matrix representing vector $\mathbf{q}_{Mn+t}^{(k)}$, i.e. column $Mn+t$ of matrix $\mathbf{Q}_{\tilde{e}}^{k}$.

Avoiding underflow. One computational challenge is caused by the fact that when dealing with long incomplete trajectories, the binomial coefficient appearing in the definition of matrix (from Equation (7.52)),

$$\mathbf{L}_{\tilde{e}} = \sum_{k=M}^{\infty} \frac{\mathbf{Q}_{\tilde{e}}^{k}}{\binom{k}{M}k} \tag{7.93}$$

may exceed maximum representable numerical (e.g. double precision floating point) value. One might then consider computing the summand quotients using exponentials and logarithms as

$$\frac{\left[\mathbf{Q}_{\tilde{e}}^{k}\right]_{s,Mn+t}}{\binom{k}{M}k} = \exp\left(\log\left[\mathbf{Q}_{\tilde{e}}^{k}\right]_{s,Mn+t} - \log\binom{k}{M} - \log k\right). \tag{7.94}$$

However, this does not solve the problem completely, because the term appearing inside the exponential may still obtain a very large negative value causing the exponential to be evaluated as zero.

This underflow problem can be solved by applying the *log-sum-exp trick* [150]. For this, let us denote, for a given incomplete edge trajectory $\tilde{e}$ and for $k \geq M$ the terms appearing inside the exponential as

$$x_k = \log\left[\mathbf{Q}_{\tilde{e}}^{k}\right]_{s,Mn+t} - \log\binom{k}{M} - \log k. \tag{7.95}$$

We can thus write the multiple-edge log-likelihood based on Equation (7.53) as

$$\log \mathcal{L}(\beta \mid \tilde{e}) = \log \sum_{k=M}^{\infty} \exp x_k - \log \mathcal{Z}_{st}. \tag{7.96}$$

The problem then is that for large incomplete trajectories (i.e. for large M), we may have $x_k \ll 0$, for some k, which causes the exponential terms in the above expression to be evaluated as zeros. The idea behind the log-sum-exp trick is that if we know the maximum value $x_{\max} = \max_k(x_k)$, then we may compute the result as

$$\log \mathcal{L}(\beta \mid \tilde{e}) = x_{\max} + \log \sum_{k=M}^{\infty} \exp(x_k - x_{\max}), \tag{7.97}$$

which avoids the underflow issue.

The maximum $x_{\max}$ can be computed, as when k increases, the term x_k becomes more and more dominated by the binomial coefficient term, meaning that after a certain value of power k, the values x_k will only decrease. The values x_k can then be precomputed up to a power k', keeping track of the maximum $x_{\max} = \max_{1 \le k \le k'}$, until the first such k' that $x_{\max} - x_{k'} > C$ for some threshold value C. For instance, choosing $C = 20$ means that for any $k'' > k'$, $\exp(x_{k''} - x_{\max}) < \exp(-20) \approx 2.06 \times 10^{-9}$.

7.D Estimation of Edge Affinities from a Step Selection Probability Function

We estimated the edge affinities a_{ij} for the graph using a Step Selection Probability Function [72, 133] using the data and general approach described by Panzacchi et al. [168] from GPS data for more than 200 wild mountain reindeer from 7 of the largest wild reindeer management areas (including Austhei). However, in the Step Selection Function estimated by Panzacchi et al. [168] no intercept was estimated, and therefore the model yielded values proportional, but not equal, to the probability of selecting a step. In order to identify baseline probabilities for the affinities a_{ij} in this paper, we used the same models, but we refitted them using the method developed by Lele and Keim [132] to estimate actual probabilities of selection, using Step Selection Probability Functions SSPF. Using the `ResourceSelection` package [134] for R [173],

we maximized the following likelihood (see [133] for details):

$$\mathcal{L}(\beta; \mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_m}) = \sum_{k=1}^{n} [\log \pi(\mathbf{x_k}; \beta) - \log P_k(\beta)] \tag{7.98}$$

where $\mathbf{x_k}$ is the vector of covariates associated to an observed step k between two consecutive locations i and j, n is the number of observed steps in the data set, π the probability of selection. Furthermore,

$$P_k(\beta) = \int \pi(\mathbf{x}; \beta) f_k^A(\mathbf{x}; s_k) d\mathbf{x}, \tag{7.99}$$

where $f_k^A(\mathbf{x}; s_k)$ denotes the distribution of resources available for step s_k. We defined the area within 2 km from the start location i of each step s_k as available. We used Akaike's Information Criterion for model selection (see [168] for details). Table 7.3 shows the parameter estimates that maximizes the likelihood in Equation (7.98).

	Estimate	Std. Error	z-value	Pr($>\|z\|$)
Intercept	-3.79	0.79	-4.77	< 0.001
Step Length	-1.24×10^{-3}	2.51×10^{-5}	-49.53	< 0.001
(max. Slope)^2	-1.33×10^{-3}	4.79×10^{-5}	-27.81	< 0.001
max. Solar Radiation	0.30	0.02	17.77	< 0.001
max. Trail dens.	-0.19	0.03	-6.34	< 0.001
max. Road dens.	0.46	0.27	1.70	0.09
crossing Road	-1.19	0.37	-3.20	< 0.001
prop. Bog	-0.31	0.41	-0.76	0.45
prop. non-Forage	0.20	0.16	1.21	0.22
prop. Forage	0.88	0.14	6.37	< 0.001
prop. Lakes	-1.40	0.33	-4.21	< 0.001
prop. Reservoirs	-3.97	0.84	-4.70	< 0.001

TABLE 7.3: Fitted coefficients for the summer step selection model of reindeer. "Max." denotes the maximum value along the step, "prop." the proportion of the land cover class along the step, and "dens." is the road density, which is the length of road within a 5 km radius. The methods are detailed in [168], our only extension was to fit the models using a Step Selection Probability Function (see main text for further details).

We predicted the affinities a_{ij} as the probability of a step between adjacent pixels i and j using the coefficients from the SSPF:

$$a_{ij} = \frac{\exp(\beta_{SSPF}\mathbf{x})}{1 + \exp(\beta_{SSPF}\mathbf{x})} \tag{7.100}$$

where β is a row vector with m elements corresponding to the coefficients from the SSPF (see Table 7.3), and $\mathbf{x}$ is a column vector with m elements describing the environmental characteristics of the transition (first element is the intercept, and equals 1). Thus, a_{ij} is the probability of selection of step i-to-j based on the vector of covariates (e.g. geographic distance between i and j, road crossing, proportion of each land cover) characterizing this transition.

7.E MLEs of reindeer trajectories

Figures 7.9-7.11 contain plots of all the 32 trajectories used in the experiment in Section 7.4. The starting pixel is marked by a green square, the ending pixel by a magenta asterisk, and pixels where the animal was observed along the trajectory by red dots. Above each plot is shown the MLE of β corresponding to the trajectory. Figure 7.12 shows a histogram of the 32 MLE values computed for each trajectory in the reindeer trajectory data set.

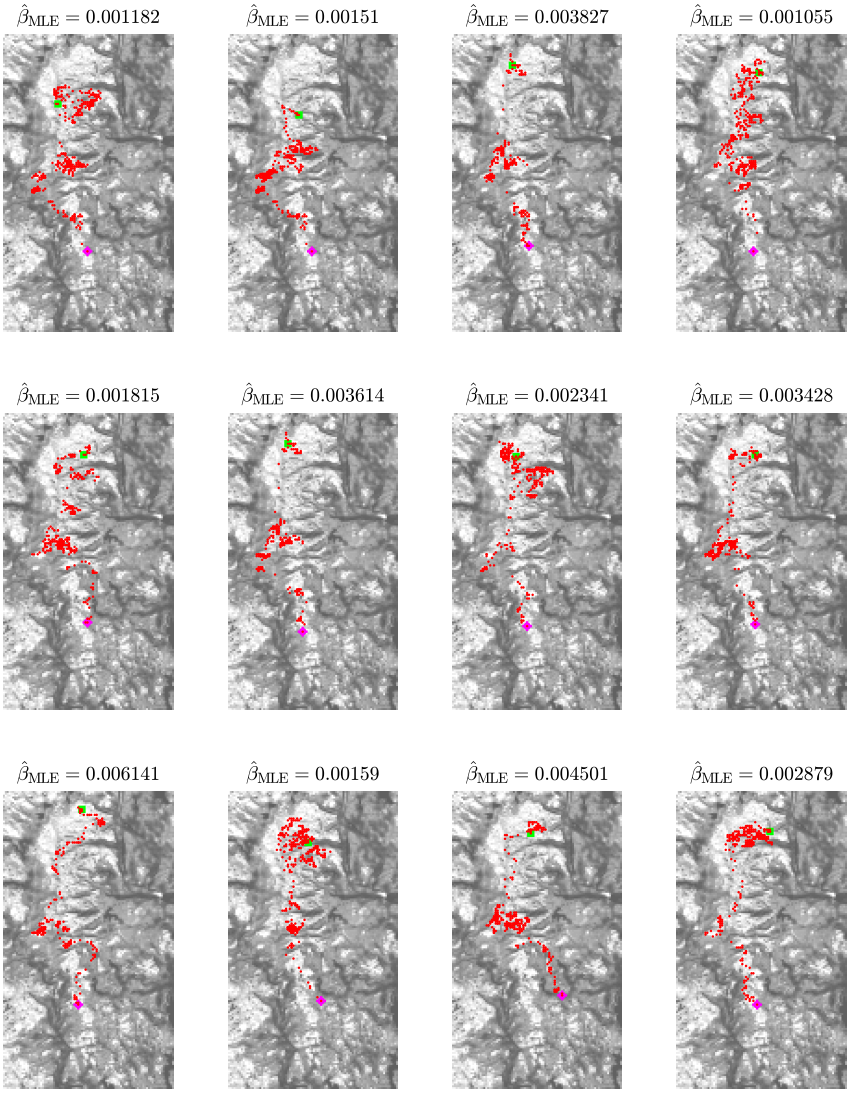


FIGURE 7.9: Trajectories 1-12

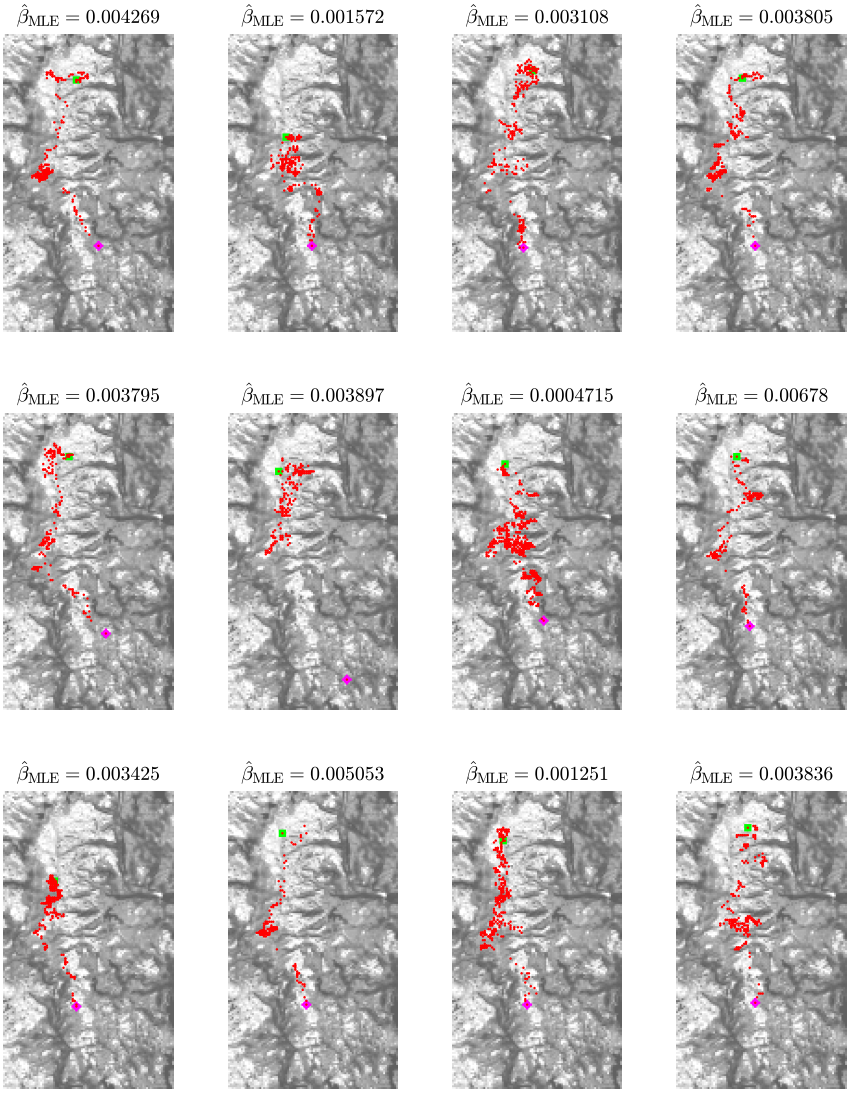


FIGURE 7.10: Trajectories 13-24

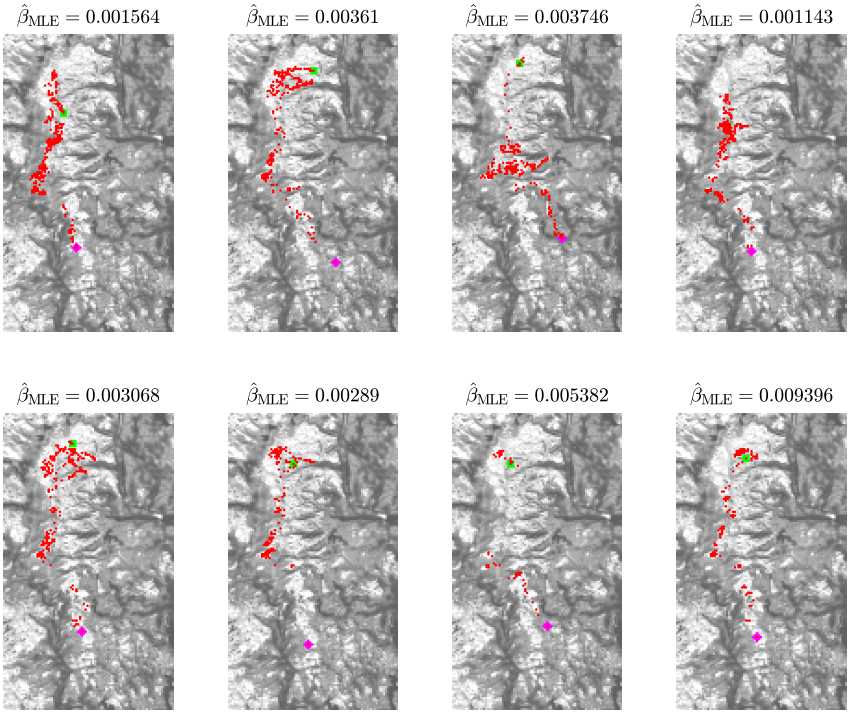


FIGURE 7.11: Trajectories 25-32

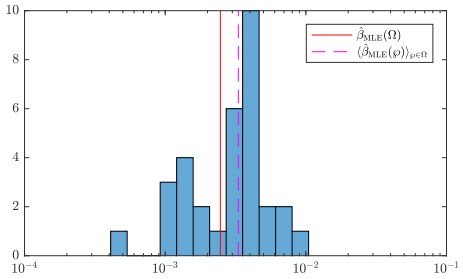


FIGURE 7.12: Histogram of the MLE values for the 32 trajectories of reindeer. The solid red line marks the MLE obtained with the whole set of trajectories, $\hat{\beta}_{\mathrm{MLE}}(\Omega) = 0.002466$, whereas the magenta dashed line shows the mean of the individual trajectory MLEs, $\langle\hat{\beta}_{\mathrm{MLE}}(\wp)\rangle_{\wp\in\Omega} = 0.003311$.

Chapter 8

Habitat Functionality metric for evaluating ecological landscapes in terms of connectivity and quality

8.1 Introduction

This chapter is based on yet unpublished and preliminary work aiming to develop methods in ecology for environmental impact assessment using the RSP framework. It follows work conducted in [168] where the potential of RSPs for modeling animal movement was initially studied. More precisely, the focus here is on quantifying the *functionality*, or *suitability* of a habitat or a broader geographic landscape for a particular animal species in terms of connectivity and quality. This is done by combining recent advances in landscape ecology, animal movement ecology and network analysis to synthesize the quality, abundance, and connectivity of each landscape unit into a single *Habitat Functionality (HF) metric*.

The metric is defined through graph theoretic formalism, which has generally become a popular paradigm in ecology [52]. Especially ecological movement models are often constructed by considering a geographic landscape as a network of habitat patches linked to each other [183, 86, 73]. The problems of quantifying the suitability of a patch or the full landscape for an animal can thus be considered analogous to the problems of quantifying the centrality (in some sense) of a node in a network, and the overall quality (in some sense) of the network. This work especially focuses on the case where the network

is based on a map image, where each pixel represents a habitat patch and is connected with edges to the neighboring pixels.

One main goal, when designing such a measure, would be that it responds to the impact of different infrastructure development scenarios on the habitat in a sensible way. The HF metric proposed here is also assessed in this context, both on a theoretical level as well as experimentally. The definition of the HF metric is stated in a heuristic manner, and it can be interpreted as a modification of the *harmonic centrality* of a node in a network. One modification is the consideration of the distances derived from the RSP framework in place of the shortest path distance, which normally defines harmonic centrality. The motivations for considering the RSP framework in this ecological context were discussed earlier in Section 7.4. Although the definition of the HF metric is intuitive, we also show some unexpected, and even undesired results in experiments, and how they can be dealt with.

8.2 Habitat Functionality metric

The HF metric is defined on a landscape, which can be a rectangular or other shape of geographic area $\Omega \subset \mathbb{R}^2$. The metric combines the consideration of the *quality* of a location, $\mathbf{s} \in \Omega$, denoted by $Q(\mathbf{s})$, and its connectivity with other locations on the landscape. The quality $Q(\mathbf{s})$ is, in general considered, following [100], as its ability to provide conditions appropriate for individual and population persistence. The exact definition of how it is computed for the case study of this chapter is discussed later.

Although $Q(\mathbf{s})$ is often, by itself, used as a proxy of habitat suitability, it ignores the spatial configuration and connectivity of resources (i.e. the encounter probability), and therefore risks to misrepresent the real contribution of a location to population persistence. Therefore, we define Habitat Functionality of a location $\mathbf{s}$, $HF(\mathbf{s})$, not only based on its habitat quality, but also on the quality of all other target locations $\mathbf{t}$ in the landscape Ω that are accessible from s:

$$HF(\mathbf{s}) = Q(\mathbf{s}) \times \int_{\mathbf{t}\in\Omega} Q(\mathbf{t}) \times K(\mathbf{s},\mathbf{t})\,\mathrm{d}\mathbf{t} \tag{8.1}$$

with $K(\mathbf{s},\mathbf{t})$ the *proximity* between location $\mathbf{s}$ and $\mathbf{t}$. For the proximity, we only state a loose description that it should be a measure of the ease of movement from location s and t. Especially, it can be defined as symmetric or not, depending on the application. Hence, the functionality of a location increases with its own habitat quality and with its connectivity to other high quality locations.

Note that a poor-quality location has low functionality, but may still contribute to the functionality of other pixels by allowing for their connectivity.

Integrating $HF(\mathbf{s})$ over $\mathbf{s}$ measures the overall functionality of the landscape Ω, which we call the *landscape functionality (LF) of* Ω:

$$LF(\Omega) = \iint_{\mathbf{s},\mathbf{t}\in\Omega} Q(\mathbf{s}) \times Q(\mathbf{t}) \times K(\mathbf{s},\mathbf{t})\,\mathrm{d}\mathbf{t}\,\mathrm{d}\mathbf{s} \tag{8.2}$$

We estimate HF and LF on a landscape by discretizing the landscape into a rectangular grid graph G, whose nodes correspond to pixels of a map image of the landscape. Each pixel is connected with edges to the nearest longitudal, latitudal and diagonal neighbors, resulting in at maximum 8 incoming and outgoing edges for each pixel. The choice of this size of neighborhood, as well as the choice of whether to consider directed or undirected edges are in general application-dependent.

Furthermore, we consider the habitat qualities and proximities as discretizations (denoted now by Q_s and K_{st}, respectively, for any $s,t \in V$) on the grid (the qualities Q_s can be weighted by pixel size). The discrete version of the local HF and global LF measures can then be written as:

$$HF_s = Q_s \times \sum_{t=1}^{n} Q_t \times K_{st} \quad \text{and} \quad LF_G = \sum_{s=1}^{n}\sum_{t=1}^{n} Q_s \times Q_t \times K_{st} \tag{8.3}$$

This definition incorporates straightforwardly the ideas of the local quality and global connectivity of locations for assessing the functionality of the whole landscape. The formulation is very general, and the actual definition of qualities and proximities should be decided for each application separately. The definition of LF is similar, and partly inspired by the *probability of connectivity (PC)* index proposed in [183]. The LF metric can actually be considered as a generalization of the PC index, as the latter can be obtained as a special case of the former, by considering the quality of a location as the land area of that location, and the proximity according to the likelihood of the most likely path between two locations.

This work explores the idea of deriving the proximity measure K_{st} based on the RSP framework, and the distances derived from it, which is also the main novel contribution of this work. The motivation for using RSP distances lies in the suitability of the assumptions behind the RSP model for animal movement modeling, as already discussed in Section 7.4. Especially, the expected cost over the RSP distribution, $\langle \tilde{c} \rangle_{st}$, provides an intuitive measure of connectivity between different locations on the landscape. However, the exact definition

and usage of the RSP model and its distances for this scenario require careful considerations over several decisions depending on the application features.

8.3 Graph-theoretical view of Habitat Functionality

The definition of the HF and LF metrics in Equations (8.1) and (8.2) and their discretized versions in Equation (8.3) seem inuitive from an ecological perspective. However, the measures deserve first a careful study from a more general graph theoretic perspective. Thus, we first discuss the relation of the proposed formulation of HF in the general context of graph centrality measures and LF in the context of network indices.

Let us consider, in this Section, the general situation where a graph G is given, as elsewhere in this thesis, as a strongly connected directed graph, with edge affinities a_{ij} and edge costs c_{ij}. Moreover, we consider the case where the node qualities are set to unity, $Q_i = 1$, for all $i \in V$, and the proximity is the inverse of the least cost distance, $K_{st} = 1/\Delta_{st}^{\mathrm{LC}}$. Then the HF of a location s corresponds to the *harmonic centrality* [25] of node s

$$C_s^{\mathrm{Harm}} = \sum_{t \neq s} \frac{1}{\Delta_{st}^{\mathrm{SP}}}, \tag{8.4}$$

The harmonic centrality was originally proposed as an improvement of *closeness centrality* (see Section 2.5 for details) to deal with disconnected graphs, for which the distance between unconnected nodes is infinite, leading the centrality scores of all nodes to be zero. Even considering closeness centrality separately within each connected component is undesirable, as it will lead to nodes in small components having larger centrality scores than nodes in large components. The harmonic centrality solves both of these two problems, in addition to which it can provide a ranking of nodes very different from the ranking obtained with closeness centrality. As mentioned in Section 2.5, closeness centrality, and similarly the harmonic centrality the HF metric, can be interpreted as a measure of the *efficiency* of a node, when considering dissemination of communication on a graph [84, 207], which also motivates the use of Equations (8.1) and (8.3).

A nice review of the history and properties of harmonic centrality (as well as network centralities in general) is given in [25]. The cited work contains an effort for stating axioms that a network centrality measure should (at least)

satisfy. A bit surprisingly, out of a comparison of 11 popular centrality measures, the harmonic centrality is noted to be the only one that satisfies the three presented axioms. Harmonic centrality has been also studied independently in chemical graph theory (see, e.g. [109]).

The inverse is, of course, only one of various ways of defining the proximity K_{st} from a distance. Instead, the HF measure can be (and likewise, the harmonic centrality could be) considered according to any other sensible transformation from distance to proximity, or even by considering directly a proximity measure without consideration of a distance measure. Moreover, the harmonic centrality is only one among the vast catalogue of graph node centrality measures, some of which was already reviewed in Section 2.5 as well as in Section 5.2 focusing on betweenness centralities. In the future, we plan to modify RSP betweenness centrality, which was studied in Chapter 5, and to consider it on ecological landscapes in the similar setting as considered in this chapter, especially by considering weights on the starting and ending nodes.

Functionality could also be considered in terms of *feedback*, or *spectral* centralities (again, see Section 2.5), such as the eigenvector centrality, or PageRank, to name a few. This could be achieved, for instance, by altering the definitions in Equations (8.1) and (8.3) so that the HF of location s would become recursively dependent on the HFs of other locations. Such an approach would lead to a measure very similar to the *metapopulation capacity* developed in [101], which is defined as the leading eigenvalue of a matrix similar to the matrix of proximities weighted by respective qualities. However, in this chapter we focus on the study of HF according to the harmonic centrality formalization of Equations (8.1) and (8.3).

The HF generalizes the harmonic centrality in two aspects — in terms of introduction of weights (according to the qualities Q_i) and by allowing consideration of an arbitrary graph node distance or proximity measure. When considering either the RSP dissimilarity or the free energy as the distance measure, the generalization from the least cost based harmonic centrality is continuous in the sense that the harmonic centrality can be approximated arbitrarily closely by choosing a large enough value of β. In a recent study [214], harmonic centrality was generalized to consider a Jaccard distance and a cosine distance, instead of the shortest path distance. However, it appears that harmonic centrality has not been considered before in terms of random walk distances, nor RSP distances.

We examine first the nature of the HF and LF metrics based on the harmonic centrality using the RSP dissimilarity as the distance measure with a fixed value of β for all s-t-pairs, and by considering unit qualities, $Q_s = 1$ for all nodes $s \in V$. Let us thus investigate this idea by defining the *harmonic RSP*

centrality of node s as:

$$C_s^{\mathrm{h-RSP}} = \sum_{t \neq s} \frac{1}{\Delta_{st}^{\mathrm{RSP}}}. \tag{8.5}$$

Note that although the proximity measure K_{st} is not required to be symmetric, the examples presented below in Section 8.3.1 give an indication as to why at least considering only the outgoing directed distances in the above definition may lead to counterintuitive and perhaps undesired results. Because of this the definition is stated with the symmetric RSP dissimilarities.

As explained in Section 2.4.2, $\Delta_{st}^{\mathrm{RSP}}$ interpolates between the least cost distance, $\Delta_{st}^{\mathrm{SP}}$, (when $\beta \longrightarrow \infty$) and the commute cost distance $\Delta_{st}^{\mathrm{CC}}$ (which is, once again, proportional to the commute time and resistance distances; when $\beta \longrightarrow 0^+$). Thus, $C_s^{\mathrm{h-RSP}}$ generalizes the harmonic centrality, and its corresponding form defined with the commute cost distance.

8.3.1 Simple examples

The HF metric is designed for ecological applications on geographic networks. Such networks can be constructed so that nodes correspond to larger patches in a landscape and the edges correspond to corridors between the patches. In this work, however, the focus is more on the case where the network is discretized into pixels based on a map image of the landscape, which, in the simplest case, is a rectangle-shaped grid of square pixels.

Thus, we first examine the behavior of the RSP distances and the harmonic RSP centrality on a simple rectangle-shaped grid. We use a 41×31 sized grid, where pixels are connected to their nearest vertical, horizontal and diagonal neighbors. The heatmaps in Figure 8.1 show the sum of incoming expected RSP costs on the left-hand column, the sum of outgoing expected RSP costs in the middle column and the sum of RSP dissimilarities in the right-hand column, for each pixel and for different values of β. The value of β used is reported above each plot, and decreases from top to bottom. The top-most middle plot corresponds to the sum of least cost distances (the limit $\beta \longrightarrow \infty$). As the graph is undirected, the sums of outgoing and the sums incoming least costs produce identical plots to the sums of least cost distances. The bottom row, from left to right shows the sums of incoming and sums of outgoing expected hitting costs of the random walk and the sums of commute time distances for each pixel (the limit $\beta \longrightarrow 0^+$).

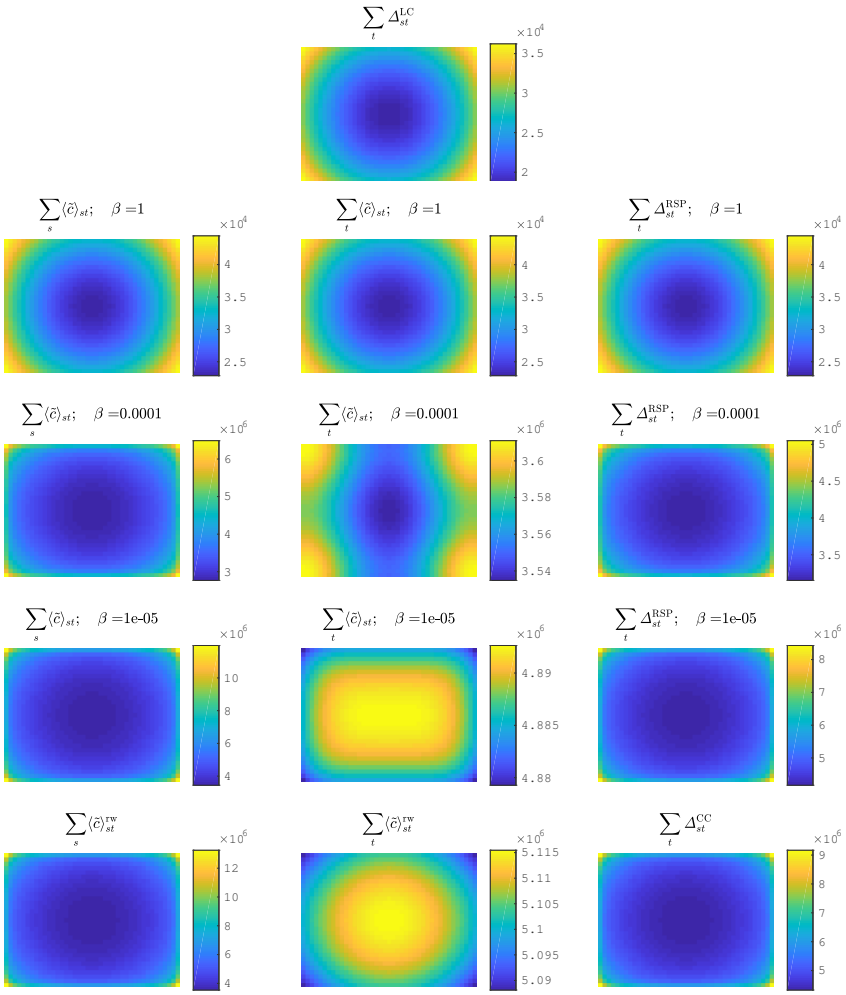


FIGURE 8.1: Sums of incoming and outgoing expected RSP costs and RSP dissimilarities for each pixel, with different values of β. On top, the sums of least cost distances from each pixel to all others is shown. The bottom row shows the sums of the random walk expected hitting costs and commute cost distances.

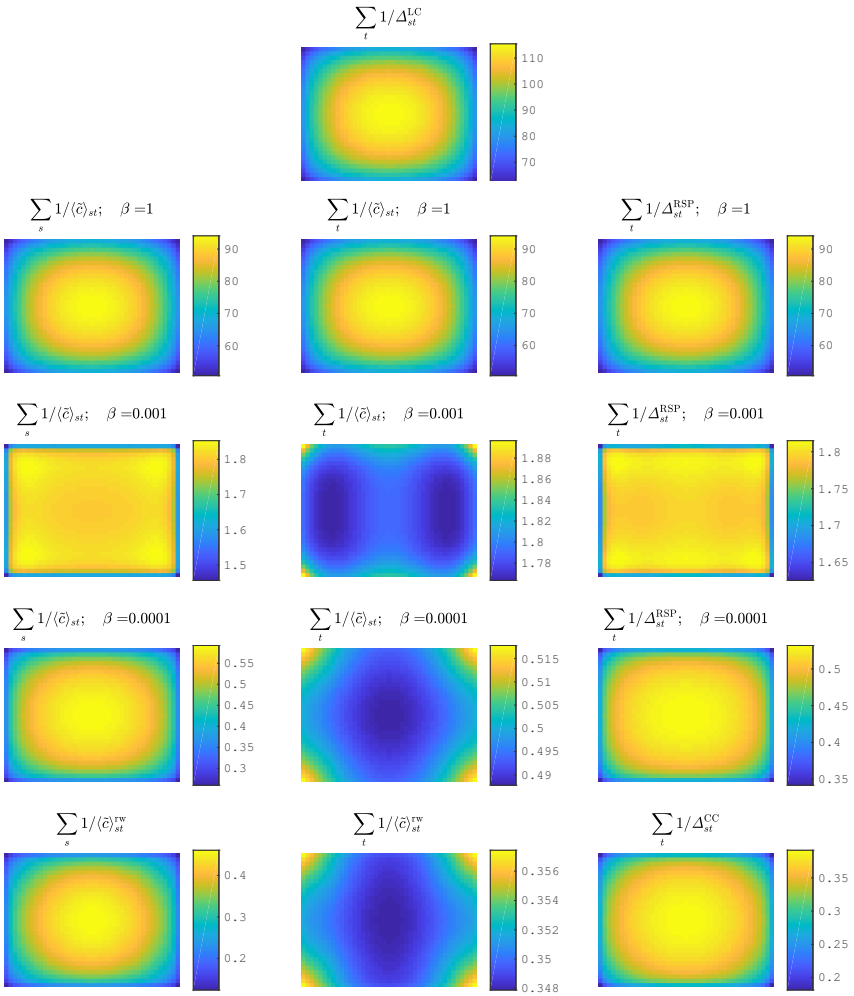


FIGURE 8.2: Harmonic RSP centrality of the inverses of incoming and outgoing expected RSP costs and RSP dissimilarities for each pixel, with different values of β. On top, the sums of least cost distances from each pixel to all others is shown. The bottom row shows the sums of the random walk expected hitting costs and commute cost distances.

Already this simple example shows an interesting aspect, which relates generally to the directed random walk distances. Namely, the middle column of Figure 8.1 shows, fairly unintuitively, that the sum of expected costs from

pixels in the center of the grid to other pixels become higher than the sum of expected costs from the corners of the landscape to elsewhere when β decreases towards zero. In other words, according to the directed random walk distances, other parts of the landscape can, on average, be reached more easily and faster from the edges and corners of the landscape than from the center.

Figure 8.2 contains the same idea as Figure 8.1 but, instead of sums of distances, here the sums of the inverses of distances at each pixel are presented as heatmaps, again considering both directed and symmetrized distances. Thus, the right-hand column of Figure 8.2 illustrates, for different values of β, the harmonic RSP centrality as defined in Equation (8.5), whereas the left-hand and middle columns show the incoming and outgoing sums of directed expected costs. A similar issue emerges in the middle column of Figure 8.2 as in the middle column of Figure 8.1, namely that with low values of β and in the natural random walk case, the sums of inverse expected costs become lower in the center than in the edges and corners of the landscape, i.e. that the method considers the corners of the landscape more central than the nodes in the center.

The same plots were also generated using the sums of free energies and free energy distances, and the results were, again, very similar to the ones presented here. The result emerging in the plots on the middle columns of Figures 8.1 and 8.2 seem counterintuitive and undesirable when considering applications. At the same time it provides an interesting result about the random walk and RSP based harmonic centrality and seems to indicate that when considering harmonic RSP centrality on a grid graph, such as the ones used here, then the symmetric distance may be more intuitive and useful than the directed distances. It is, however, worth noting, by looking at the colorbars of the plots, that the scale of the values in the middle column plots is very narrow, especially when comparing with the scales of values in the other plots. In other words, the effect of the phenomenon is not strong, but it is, nevertheless, surprising.

8.4 Effect of changing a corridor on HF

One original motivation for considering the HF metric in terms of RSP distances was that it would hopefully be sensitive to changes in the landscape that cause drastic effects on the landscape globally. With this idea in mind, we devised a simple experiment, with an artificial landscape consisting of two patches connected by a corridor. Ideally, the HF and LF metrics would be especially sensitive towards changes in the corridor, such as its widening or closing. We thus considered such a landscape, of size 31×61, with a 5-pixel

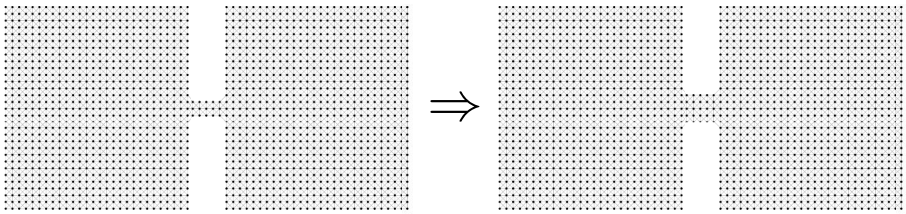


FIGURE 8.3: Landscapes used for studying the sensitivity of LF to a widening of a corridor connecting two landscape patches. The landscape on the left has a corridor of width 3 pixels, and the one on the right side has a corridor of width 5 pixels.

wide wall in the middle of the landscape. We then considered an open corridor in the middle of the wall with two different corridor widths, of 3 and 5 pixels. The grid graphs corresponding to these artificial landscapes are depicted in Figure 8.3.

8.4.1 Counterintuitive result

We computed on both landscapes of Figure 8.3 the RSP dissimilarities for all node-pairs, and the HF metric for all pixels according to the harmonic RSP centrality of Equation (8.5). The LF value for each landscape was then simply computed as the sum of the HF values on the landscape (as we considered each pixel i having quality $Q_i = 1$). We then compared the LF values between the reference landscape (with a 3-pixel-wide corridor) and the altered landscape (with a 5-pixel-wide corridors).

The results of this test are plotted in Figure 8.4A. It shows the increase, in percentage, of LF, for different values of β, when the corridor between the two patches was widened from 3 to 5 pixels. In addition, the dashed lines indicate the corresponding improvement in LF, when considering the shortest path distance, or the commute cost distance as the distance instead of the RSP dissimilarity. A bit surprisingly, and disappointingly, the effect of widening the corridor on LF was not very strong with the RSP-based functionality measure. The largest relative increase in LF is gained by the random walk based HF metric, i.e. as $\beta \to 0^+$, but with intermediate values of β the increase was always smaller than with the random-walk based metric. Even more surprisingly, the plot shows a strange bump for the value of $\beta = 10^{-4}$, for which the overall LF value actually *decreases* by widening of the corridor. The same effect was observed (though not reported here) to happen also when using the free energy

distance, $\Delta_{st}^{\mathrm{FE}}$, in place of the RSP dissimilarity, $\Delta_{st}^{\mathrm{RSP}}$, in the definition of the harmonic RSP centrality in Equation (8.5), although the effect of the "bump" was not as strong as in Figure 8.4A.

We also computed the change in LF when considering, the logarithmic forest distance, Δ^{logFor}, in place of the RSP dissimilarity. Remember that Δ^{logFor} was defined in [45] and was considered earlier in Chapter 3 when making a comparison of graph node distance measures. Recall also that the logarithmic forest distance generalizes, depending on a parameter α, the shortest path distance (when $\alpha \to 0^+$) and the resistance distance (instead of the commute distances; when $\alpha \to \infty$). Figure 8.5 shows the improvement, in percentage, of LF based on Δ^{logFor}, for a range of values of the parameter α. The plot also contains the improvement obtained with the resistance distance, Δ^{Res}, which is slightly higher than the improvement obtained with the commute cost distance, Δ^{CC}, that was plotted in Figure 8.4. From Figure 8.5 we see that, similar to the plot based on Δ^{RSP} in Figure 8.4A, the improvement in LF is also not monotonous with respect to the parameter α. Furthermore, for a range of values of α the improvement of LF based on Δ^{logFor} is actually smaller than the improvement obtained with LF based on the shortest path distance Δ^{SP}. Still, a positive aspect about considering Δ^{logFor} for defining LF in this example is that at least LF does not decrease for any value of α by widening the corridor, which happens in Figure 8.4A when considering Δ^{RSP}. However, the LF index given by Δ^{logFor} is still less sensitive to the widening of the corridor, than the index defined by the resistance distance. Moreover, the use of Δ^{logFor} in the ecological context considered in this chapter is discouraged partly by the fact that it lacks a clear interpretation in terms of movement on the graph, as opposed to the RSP distances.

One possible explanation for the undesired result with the LF metric based on RSP dissimilarity is that when the corridor is modified, the landscape graph also changes globally in such a way that affects the relative entropies of the probability distributions between node pairs. Remember that the original definition of the RSP distribution of Equation (2.10), was derived by minimization of expected cost with the relative entropy with respect to the random walk distribution constrained to a fixed level J_0. However, as discussed briefly in Section 2.3, in the end the relative entropy value J_0 is replaced by the inverse temperature β which obtains a similar role as J_0. Accordingly, when the graph is transformed, as in the above corridor example, the RSP model based on a fixed value of β will result in a different value of relative entropy on the original and the transformed graph.

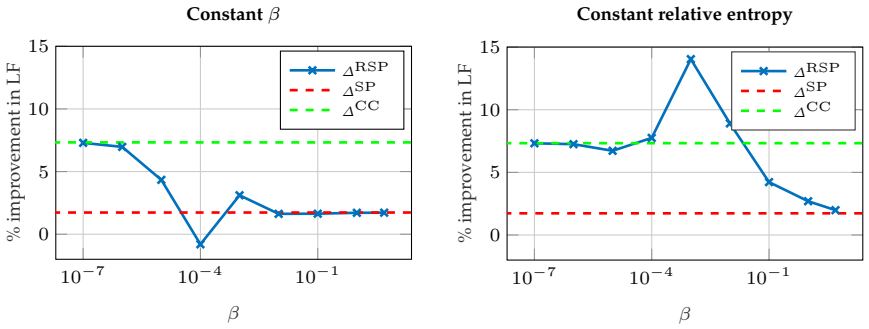


FIGURE 8.4: Increase in LF obtained by widening of corridor. The plots report the increase in LF for a range of values of β when using a fixed β for comparing LF on the two landscapes (A), and when using, on the landscape with wider corridor, a new value of β for which the mean relative entropy coincides with the mean relative entropy on the reference landscape (B).

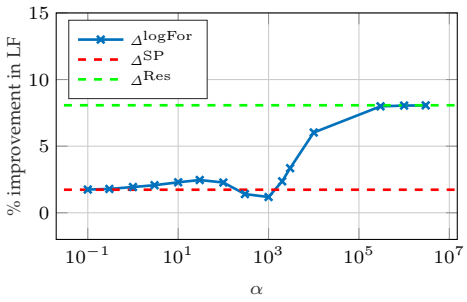


FIGURE 8.5: Increase in LF obtained by widening of corridor, when considering LF based on the logarithmic forest distance. The setting for producing the plot is otherwise same as in Figure 8.4A. Note that the behavior of the logarithmic forest distance with respect to the parameter α is opposite from the RSP dissimilarity, i.e. a value of α close to zero produces distances similar to the shortest path distances, while large values of α produce distances similar to the resistance distance.

8.4.2 Maintaining constant relative entropy

A possible solution for making the HF and LF measures more sensitive towards the change in a corridor on a landscape is to reset the value of β after the landscape has changed so that the relative entropy level of the RSP model remains constant. In order to consider the global entropy level in a rigorous manner, we extend the RSP model to consider the set of all hitting paths on G (instead of only paths between a fixed s-t-pair). This resembles the idea of the bag-of-paths (BoP) framework, considered earlier, in Section 2.6 and in Chapter 4. Recall that the BoP framework is defined, similarly to the RSP framework, according to minimization of expected cost with relative entropy constraint, but on the set $\mathcal{P}$ of all hitting paths on G, instead of the set $\mathcal{P}_{st}$ of hitting s-t-paths only. From this formulation one can compute the BoP probabilities, Π_{st} (see Equation (4.10)), also called the *coupling probabilities* in [97], i.e. the probabilities that a path drawn from the BoP distribution has s and t as its starting and ending nodes, respectively.

Here we instead make the extension of RSPs from the set $\mathcal{P}_{st}$ to the set $\mathcal{P}$ by *constraining* the coupling probabilities to stay constant, i.e. to be independent of the temperature of the system. Considering the definition of HF in Equations (8.1) and (8.3), we focus especially on the case where the coupling probabilities are set as products of probability distributions on the starting and ending nodes of paths (which correspond to the qualities Q_i in the formulation of HF in Equations (8.1) and (8.3)). For this, we consider on the nodes $i \in V$ probability distributions q_i^{s} and q_i^{t}, namely the starting and ending node probability distributions, respectively, and fix the coupling probability of an s-t-pair as $q_s^{\mathrm{s}} q_t^{\mathrm{t}}$. We then extend the definition of the random walk probabilities over the set $\mathcal{P}$, taking also into account the starting and ending probabilities, as $\mathrm{P}^{\mathrm{rw}}(\wp) = q_s^{\mathrm{s}} q_t^{\mathrm{t}} \mathrm{P}_{st}^{\mathrm{rw}}(\wp)$. Then, we may augment the BoP minimization problem of Equation (4.8) with an additional constraint on the coupling probabilities and seek for the distribution that solves

$$\underset{\mathrm{P}}{\text{Minimize}} \sum_{\wp \in \mathcal{P}} \mathrm{P}(\wp)\tilde{c}(\wp) \quad \text{subject to} \quad \begin{cases} \mathbb{J}(\mathrm{P}\|\mathrm{P}^{\mathrm{rw}}) = J_0 \\ \sum\limits_{\wp \in \mathcal{P}} \mathrm{P}(\wp) = 1 \\ \sum\limits_{\wp \in \mathcal{P}_{st}} \mathrm{P}(\wp) = q_s^{\mathrm{s}} q_t^{\mathrm{t}}, \ s,t \in V. \end{cases} \tag{8.6}$$

As such, the third constraint is not well defined, because the sum on the left side is empty (i.e. 0), when $s = t$, as normally hitting paths are not defined from a node to itself. The right side, however, in that case is $q_s^{\mathrm{s}} q_s^{\mathrm{t}}$, which is non-zero at least for some $s \in V$. One way of solving this issue is to change the definition of a path to include *zero-length* paths, i.e. sequences consisting of only

one node and no edges. Formally, we define, for all $i \in V$, the set $\mathcal{P}_{ii} = \{\wp_{ii}\}$, where $\wp_{ii} = (i)$ is the zero-length path of node i. Likewise, the set $\mathcal{P}$ of all hitting paths on G is considered to contain the zero-length paths. Moreover, we also need to define the random walk probabilities for the zero-length paths, for all $i \in V$, as $\mathrm{P}^{\mathrm{rw}}_{ii} = 1$.

It is not difficult to show that the solution of the minimization in Equation (8.6) is given by the weighted RSP distribution, namely, for a path $\wp_{st} \in \mathcal{P}_{st}$,

$$\mathrm{P}(\wp_{st}) = \begin{cases} \dfrac{q^{\mathrm{s}}_s q^{\mathrm{t}}_t \widetilde{w}(\wp_{st})}{\mathcal{Z}_{st}}, & t \neq s \\ q^{\mathrm{s}}_s q^{\mathrm{t}}_s, & t = s. \end{cases} \tag{8.7}$$

where, as before, $\widetilde{w}(\wp)$ is the unweighted path likelihood from Equation (2.13), and $\mathcal{Z}_{st}$ is the standard partition function of hitting s-t-paths of Equation (2.9). Note that here, although the path $\wp_{st}$ has starting node s and ending node t, the probability distribution P is the BoRSP distribution, defined over the set $\mathcal{P}$ of paths between all node-pairs, not the RSP distribution P_{st}, which is the conditional distribution defined over the set $\mathcal{P}_{st}$.

Finally, in this setting the global relative entropy can be shown to be simply the weighted sum of the relative entropies of the RSP distributions between all s-t-pairs, with weights q^{s}_s and q^{t}_t:

$$\mathbb{J}(\mathrm{P}\|\mathrm{P}^{\mathrm{rw}}) = \sum_{s,t \in V} q^{\mathrm{s}}_s q^{\mathrm{t}}_t \mathbb{J}(\mathrm{P}_{st}\|\mathrm{P}^{\mathrm{rw}}_{st}). \tag{8.8}$$

Note that we could as well formulate the coupling probability constraint in Equation (8.6) considering a full $(n \times n)$ matrix of coupling probabilities, whereas here only a vector $\mathbf{q}^{\mathrm{s}}$ of starting node probabilities and a vector $\mathbf{q}^{\mathrm{t}}$ of ending node probabilities are required, which fits our purpose in the context of the HF metric. Although in the current formulation of the HF metric, the only one type of quality Q_i is defined on each node, which can be considered to represent both the starting node and ending node probabilities.

As this idea can be seen as an intermediate between the RSP and BoP frameworks, we name this extension the *bag-of-randomized-shortest-paths* (BoRSP) framework. Note that in [82], Section 6, a weighted version of the BoP framework was proposed similar to the BoRSP idea. However, the difference with the current proposal is that there the weights are only used to redefine the random walk probabilities, but not as constraints of the path distribution. As a result, the coupling probabilities of the weighted BoP model are not fixed, but depend on the selection of β, similarly to the standard, unweighted BoP

model. Another similar proposal is in [97], where the BoP framework is considered in a randomized optimal transport problem setting. Also there, starting node and ending node distributions are considered, but without constraints on the coupling distribution. Instead, also there, the coupling probabilities are deduced from the model. Thus, these two other models seem to some extent similar to the model considered here, but actually all three differ and produce very different results, also in terms of distances. In a way, the two other models have a clearer intuition, and the formulation of the BoRSP model is here mostly proposed for the sole purpose of deriving a global relative entropy measure that is similar to the formulation of the HF metric. Further investigation on the BoRSP model and a more intuitive justification for it will be left for future work.

In order to solve the issue faced in the corridor experiment in Section 8.4, we thus aim to keep the global relative entropy level constant, when increasing the width of the corridor, which requires a change in the value of β. We did this by computing first on the reference landscape with the corridor of width 3, and for every intermediate value of β, the corresponding LF value and the mean relative entropy over all node pairs on the graph,

$$J_0 = \mathbb{J}(\mathrm{P}\|\mathrm{P}^{\mathrm{rw}}) = \sum_{s,t} \mathbb{J}(\mathrm{P}_{st}\|\mathrm{P}^{\mathrm{rw}}_{st})/n^2, \tag{8.9}$$

by using Equation 6.12. Remember that in the corridor experiment we consider unit qualities, corresponding to uniform starting and ending node distributions, $q_i^{\mathrm{s}} = q_i^{\mathrm{t}} = 1/n$. Then, on the modified graph, with a wider corridor, we search for a value of $\hat{\beta}$ such that the mean relative entropy is approximately equal to J_0, and then compute the LF on the modified graph using $\hat{\beta}$.

These results are presented in Figure 8.4B, which reverses the undesired result obtained with constant β shown in Figure 8.4A. Indeed, by letting β change along the change of the graph, so that the mean relative entropy is kept constant, the LF value now increases, when the corridor is widened. A further illustration of this idea is presented in Figure 8.6 which shows heatmaps of RSP dissimilarities and their inverses (corresponding to proximities in this example) from one particular pixel to all other pixels on the two landscapes. For the landscape with the 5-pixel-wide corridor, heatmaps are plotted for both the original value of $\beta = 10^{-4}$ and the updated value $\beta = 1.19 \times 10^{-4}$, which results in equal average relative entropy on the two landscapes.

Comparing the top and middle row plots in Figure 8.6 we see that although a wider corridor decreases distances to the other side of the wall, the dissimilarities to nodes on the same side of the wall actually increase, on average. This ultimately results in a decrease in LF caused by the widening of the corridor.

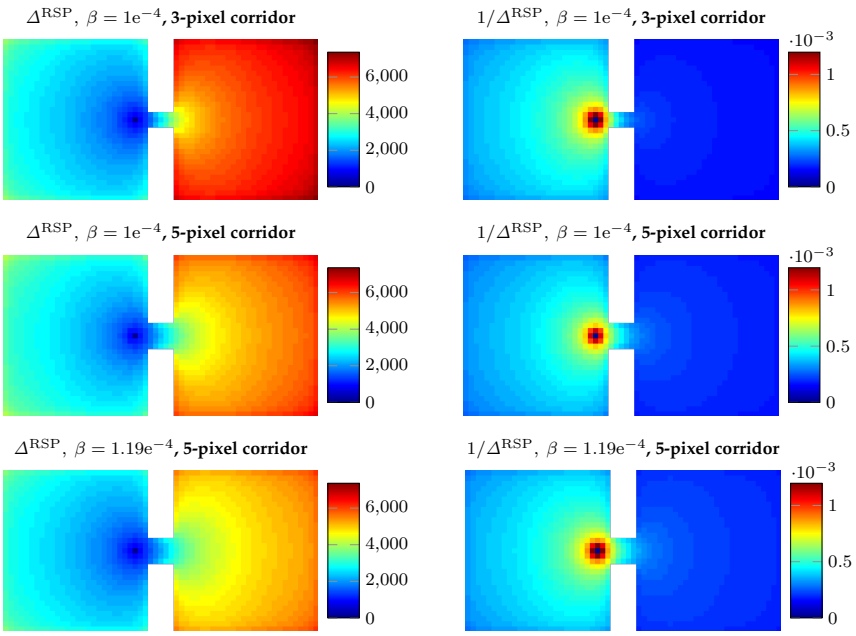


FIGURE 8.6: Heatmaps of RSP dissimilarities (left column) and their inverses (right column) from one node on the left hand patch of the landscape, next to the corridor, to all other nodes. The top row shows the landscape with the 3-pixel-wide corridor and $\beta = 10^{-4}$, the middle row shows the landscape with 5-pixel-wide corridor and also $\beta = 10^{-4}$, and the bottom row shows the landscape with 5-pixel-wide corridor and $\beta = 1.19 \times 10^{-4}$ (which results in equal average relative entropy as $\beta = 10^{-4}$ on the 3-wide-pixel corridor landscape). Note that colormaps are scaled to match in the plots within the same column meaning that a pixel of a certain color is at equal distance, or proximity, in all three plots of the same column, in order to facilitate comparisons.

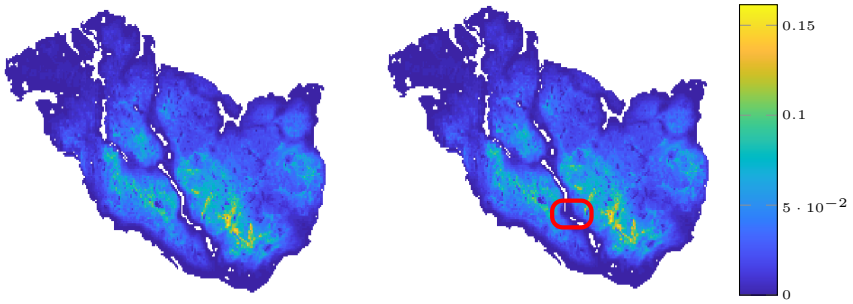


FIGURE 8.7: The real landscape (A) and its artificially modified version (B), where an imaginary "bridge" (marked with the red rectangle, for clarity) is added over the water reserve crossing the landscape. Both heatmaps depict the qualities Q_i of each pixel i.

Comparison with the bottom row plots then shows that by keeping the relative entropy constant (instead of constant β) maintains the distances on the same side of the landscape on a similar level in the 5-pixel-wide corridor landscape as in the 3-pixel-wide corridor landscape. As the distances to the other side of the wall decrease, also the LF increases, as shown by Figure 8.4

The results presented in Figures 8.4 and 8.6 provide evidence to the idea that when making comparisons of RSP distances over varying graphs, the RSP models should be defined on the different graphs so that their relative entropies are equal, instead of considering the same value of the inverse temperature β. This, however, can be computationally inconvenient, especially on large graphs, as the model cannot be computed directly so that it would result in a desired relative entropy value, but instead the desired entropy value has to sought by a binary search over different values of β. We nevertheless resort to this approach in the following for demonstrating the applicability of the HF and LF metrics in an ecological application on a real landscape.

8.5 Use case with wild mountain reindeer

The particular use case considered in this work is the study of wild mountain reindeer (*Rangifer t. tarandus*; which were also studied in Chapter 7) on a landscape in the area of Snøhetta in Norway. This wide-ranging species is especially vulnerable to anthropogenic barriers to movements, which can

block traditional migration routes and prevent access to key seasonal habitat [99]. In Norway, migrations are rapidly disappearing due to the piecemeal development of an intricate network of infrastructures including roads, railways, hydropower, and tourist resorts [167, 165, 166, 152, 208]. Recent, independent estimations of habitat loss [167] and of habitat fragmentation [168] at the national scale proved valuable, but incomplete for sound conservation planning.

In this context, we are able to use GPS data collected from individual animals for estimating two quantities required for computing the HF metric, namely the edge affinities, a_{ij}, and the habitat qualities, Q_i. The edge affinities are defined using the same method as in Section 7.4, based on Step Selection Probability Functions (SSPF) [133]. Thus, affinity a_{ij} corresponds to the Step Selection Probability, which can be interpreted as the probability of an animal to traverse edge (i, j) upon encountering it instead of remaining in location i [135]. The exact process of computation of the affinities is detailed in Appendix 7.D of Chapter 7. This probabilistic interpretation of affinities justifies well the consideration of affinities for defining the random walk behavior on the landscape.

The habitat qualities, Q_i, are estimated based on Resource Selection Probability Functions (RSPF). The only difference with the estimation process of affinities is in the definition of the term $f_i^A(x; \mathbf{s}_i)$ in (Equation 7.99) in Appendix 7.D of Chapter 7, i.e. the distribution of resources available for location s_i. Namely, for the RSPF the distribution of available resources was given by the population range (for details, see [167]), and for the SSPF we defined the area within 2 km of each location s_i as available (for details, see [168]).

8.5.1 Workflow from edge affinities to proximities between all nodes

As mentioned, the GPS data can be used for inferring edge affinities, a_{ij}, corresponding to the probability of movement over an edge upon encounter, and node qualities, Q_i, corresponding to the likelihood of observing an animal in location i based on the geographic features of the location. In order to compute the HF and LF from these, the proximities between all node-pairs must be defined. The first step is to define edge costs, c_{ij}, for which we resort to the affinities. As the affinities correspond to Step probabilities (i.e. $0 < a_{ij} \leq 1$ for all $(i, j) \in E$), we first considered edge costs according to

$$c_{ij}^{\text{odds}} = \frac{1 - a_{ij}}{a_{ij}} = \frac{1}{a_{ij}} - 1, \tag{8.10}$$

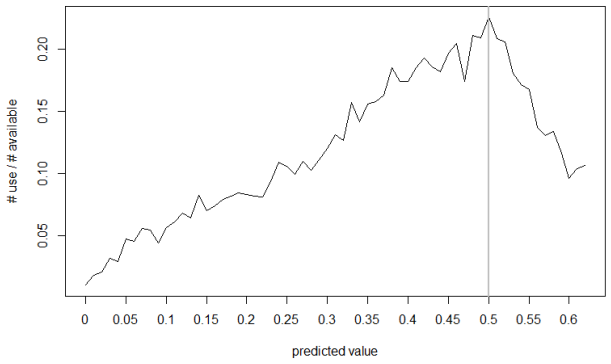


FIGURE 8.8: Ratio of used steps versus available random steps for binned predicted Step probabilities (a_{ij}). Not surprisingly, the number of actual used steps per available step increases as the predicted probability increases. The vertical gray line represents the maximum ratio, i.e. beyond this point the predicted probabilities were not increasingly used (the actual decrease is likely due to smaller sample sizes). The lack of increased use indicates the indifference of animals to a further increase in predicted probabilities. We interpreted this as for values of $a_{ij} > 0.5$ the corresponding cost is negligible (i.e. $c_{ij} = 0$).

which corresponds to the *odds against* using the step upon encounter. This is similar to the inverse, which is often used in ecological applications (e.g., [47]). However, as the affinity values here are scaled between 0 and 1, this use of the inverse results in a minimum resistance or cost of 1, which is not desirable. Instead, the odds against range from 0 to infinity, which is more convenient.

The odds against can be interpreted as the expected number of times that the animal should encounter the transition before it will take it. In gambling terminology, it can be considered as the expected payout of each bet made against the occurrence of the transition. As the cost of a path is defined as sum of edge costs along the path, the path cost can be interpreted as the expected cumulated payout.

Although in theory the Step probability ranges between 0 and 1, in practice the maximum probabilities are often far below 1 (due to e.g. stochasticity in animal movement and/or poor model fit). We therefore stretched the Step probabilities over the whole interval $[0 - 1]$ to correct for this. Figure 8.8 shows

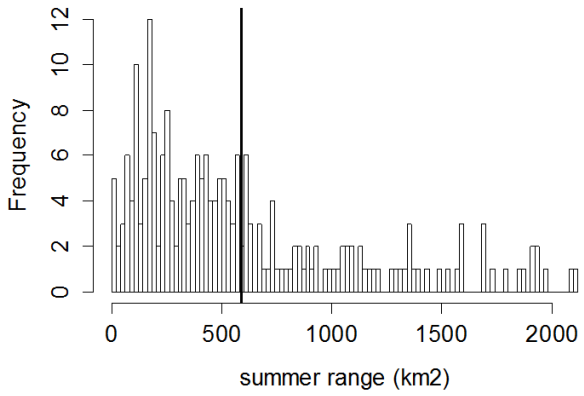


FIGURE 8.9: Histogram of the summer range sizes from reindeer (N = 224) in Southern-Norway during the summer (based on $N \approx 120$ locations per individual in July – from a sampling schedule of one location every 3 hours). We estimated the summer range sizes using 95 % contour of a kernel density estimator. The mean range size (black vertical line) was 590 km^2.

that for Step probabilities $a_{ij} > 0.5$ the animals did not increasingly use those steps, which we interpret as that there was no longer a cost associated to those transitions (i.e. $c_{ij} = 0$). Thus, we scaled the Step probabilities by multiplying with 2 and limited the scaled values to maximum of 1 (i.e. for $a_{ij} \geq 0.5$ the associated cost $c_{ij} = 0$). Adding these adjustments to the odds-against transformation in Equation (8.10), the final definition of costs from affinities can be expressed as

$$c_{ij} = \frac{1}{\min\{2a_{ij}, 1\}} - 1 \tag{8.11}$$

Another natural choice, would be to define $c_{ij} = -\log a_{ij}$. However, the choice for the former was based on initial tests, which are not reported here.

With the edge costs defined, the RSP distribution can be computed with a given value of β, and accordingly the expected costs and RSP dissimilarities. We then considered, as in Section 8.3 the RSP dissimilarities for defining proximities. We then first considered again the inverse transformation (plus one to avoid

zero division) for defining the proximities:

$$K_{st} = \frac{1}{1 + \Delta_{st}^{\mathrm{RSP}}} \tag{8.12}$$

Preliminary results using these transformation (i.e. Equations (8.11) and (8.12)) showed a fast decay of the proximity with geographic distance in core areas of the landscape, which indicated the use of a too large cost-scale (i.e. an overestimation of 'unit' cost). Ideally, cost values would be derived from individual fitness or vital rates. Future research needs to address the link between landscape resistance or cost and individual fitness. However, for now we focus on adjusting the scale based on individual space use. We decided to alter the cost scale by adjusting the RSP dissimilarities with a factor, f, resulting in the final transformation of RSP dissimilarities into proximities:

$$K_{st} = \frac{1}{1 + f\Delta_{st}^{\mathrm{RSP}}} \tag{8.13}$$

The scale correction factor controls the 'half-life' of the proximity with increasing expected path costs (*e.g.* for $f = 1/20$ the proximity $K_{st} = 0.5$ for an expected commute cost $\Delta_{st}^{\mathrm{RSP}} = 20$). We estimated f from observed reindeer space use as the diameter of a circular area corresponding to the average home range sizes of reindeer in Norway (which is $489km^2$, see Figure 8.9). Thus, we used a scale correction factor $f = 1/50$ (from $2\sqrt{\frac{489}{\pi}}$ km divided by 0.5 km for the number of pixels). To sum up, the proximities K_{st} were obtained from edge affinities, a_{ij}, by first applying Equation (8.11), then computing the RSP dissimilarities $\Delta_{st}^{\mathrm{RSP}}$ with a given value of β and finally by transforming them to proximities according to Equation (8.13) with the scaling factor $f = 1/50$.

8.5.2 Effect of opening a corridor in a real landscape

Here we demonstrate the use of the HF and LF metrics on a real landscape by a similar experiment as the experiment presented in Section 8.4 with the artificial landscape with a corridor of varying width. The original landscape considered here, and its artificially manipulated version, are depicted in Figure 8.7 as heatmaps of quality values Q_i of each pixel i. The landscapes contain 18 930 pixels (i.e. nodes in the corresponding graph).

The landscape in Figure 8.7B varies from the one in Figure 8.7A in that an artificial imaginary bridge is added to the landscape over the body of water crossing through the landscape from southeast to northwest. Studying and

comparing the effect of such an infrastructural change in the landscape in terms of the HF and LF metrics could give indication and advice on decision-making in development plans. Again, for this goal, the HF and LF metrics would desirably have a strong response to the introduction of the corridor in the central part of the landscape

We computed the LF value on both the original and the modified landscape according to the left side equality in Equation (8.3), using the qualities Q_i and the proximities K_{st} as defined in Section (8.5.1). We then computed again the change, in percentage, in LF between the two landscape versions. This was first done using a fixed value of β on both landscapes for computing LF based on RSP dissimilarities. The result is plotted in Figure 8.10A showing that the unexpected effect seen in the artificial corridor example in Section 8.4 is even stronger in the real world case. Namely, for most values of β, the LF based on RSP dissimilarity changes less than the LF index based on both the shortest path and commute cost distance. Also similar to the corridor experiment of Section 8.4, with the value $\beta = 3 \times 10^{-5}$, LF even decreases by addition of the imaginary bridge.

Presumably, the seemingly weak (and even negative) response of the RSP-based LF to the bridge is again caused by the fact that the relative entropies change when considering the same value of β on both landscapes. We therefore again computed the change in LF when maintaining a fixed relative entropy, measured according to Equation (8.8), by using different values of β in the two landscapes. Because of the large size of the landscape, causing long computation time and heavy memory-usage, we decided to resort to a faster solution of approximation of the relative entropies, where the quality-weighted sum of relative entropies was computed from all nodes of the graph to only a subset of target nodes. The target nodes were selected as the 2000 nodes with the highest quality values, Q_i, on the landscape graph. A value $\hat{\beta}$ was then sought on the modified landscape using the same set of target nodes in order to have a reasonably close relative entropy estimate compared to the estimate on the original graph.

The result, plotted in Figure 8.10B, shows, similar to the experiment in Section 8.4, that by maintaining constant relative entropy level during the addition of the bridge in the landscape the RSP-based LF increases more with most intermediate values of β, when compared to the change in LF with the shortest path or commute cost distance. The increase in the RSP-based LF seen in Figure 8.10B is actually surprisingly large, especially with higher values of β.

The above result might be also slightly distorted by possible inaccuracy in the approximation of the relative entropy on the modified landscape. Indeed, approximating the relative entropy by using only a subset of node-pairs is

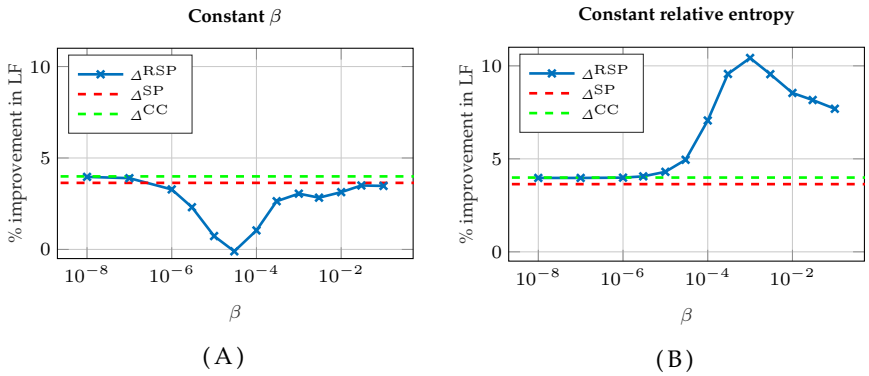


FIGURE 8.10: Increase in LF obtained by adding an imaginary "bridge" to the studied real landscape. The plots report the improvement in LF for a range of values of β when using a fixed β for comparing LF on the two landscapes (A) and when using, on the landscape with wider corridor, a new value of β for which the mean relative entropy coincides with the mean relative entropy on the reference landscape (B).

not trivial. In particular, as the estimate used here is based on the top-quality nodes, that may bias the approximation, as the top-quality nodes are often situated close to each other. This possibly leads to an overestimation of the altered $beta$-value used for computing LF in the modified landscape for Figure 8.10B. The result is still as desired, i.e. that the RSP-based LF seems sensitive to the addition of the bridge, but an incorrect estimate of the relative entropies may in other situations cause other undesired effects. Especially, if the approximation of relative entropy in the original landscape is an underestimate of the exact relative entropy, that may lead to an underestimate of β, which in turn may result to a lower value for LF on the modified landscape, thus defeating the purpose of the effort for maintaining constant relative entropy to begin with. The result should still be validated by brute-force computation of the full relative entropies.

8.6 Discussion

This chapter presented a method for estimating the functionality of geographical landscapes based on RSP distances, but also brought up some surprising

and partly unintuitive features of the method. The main motivation for considering habitat functionality in terms of RSPs derives from the seeming suitability of the assumptions behind the RSP model for modelling animal movement. Moreover, the method would desirably (and was originally assumed to) be more responsive to some critical changes in the landscape than the LF metric based on shortest path or commute cost distances. However, the initial formulation of the HF and LF metrics did not achieve this goal, as indicated by the artificial corridor experiment of Section 8.4 and the experiment on the real landscape of Section 8.5.2. On one hand this raises the question about whether the assumptions of the RSP model and RSP distances in the end are suitable for the problem at hand. On the other hand, if the assumptions behind the model are indeed valid, then perhaps such surprises could give new, unexpected insights into the movement behavior of animals.

We proposed, as one possible solution, that the undesired results could be avoided by making comparisons over two different graphs by constraining the relative entropy of the RSP models to remain constant, instead of considering the same inverse temperature on both graphs. This makes more sense in terms of both information theory as well as in ecology. Namely, the relative entropy is a more direct measure of information processing capabilities of an agent (see Section 6.3.3) than the temperature. This change of view indeed seems give a result that was initially desired from the HF and LF metrics. The idea of maintaining constant relative entropy was also given a theoretical justification by formulating the BoRSP framework, which enables a sensible quantification of the global relative entropy on the graph. The drawback of the method of maintaining constant relative entropy is the computational cost of finding a value of β corresponding to a desired relative entropy level.

Directions for future investigations in this context would be to study yet more graph-based distance measures for defining functionality metrics. On the other hand, it could be more fruitful to make an effort for laying out axioms for how such a functionality metric should preferably behave. The work presented in the chapter is based on still on-going research, which hopefully, eventually, will result in coherent answers to the questions raised here.

Chapter 9

Conclusion and discussion

This thesis has presented many new measures and methods for network and graph-based data analysis, mostly in the context of the randomized shortest paths framework. The Gibbs-Boltzmann probability distribution over paths defined within the RSP framework can be used to generalize two common paradigms for network analysis, one based on shortest paths between nodes, and the other based on random walks on the network. Accordingly, the measures and methods derived from the RSP framework generalize traditional measures on networks, which already makes them appealing to study and use. In addition, the methods were demonstrated and evaluated in different examples and experiments, often showing that they obtain good performance or otherwise have interesting qualitative features.

The methods studied in the thesis have often been developed with generality in mind. Some of the methods were born out of interesting theoretical considerations and were not designed specifically for an application. Even in those cases the applicability of the methods has also been demonstrated with illustrations and verification with artificial and real problem settings and data. For instance, the RSP betweenness centrality measures developed in Chapter 5 emerged as natural side products of the general idea behind the RSP framework. In other words, they were not designed with a fixed application in mind. Instead, the RSP betweenness measures are interesting especially on a theoretical level considering the historical context of graph node centrality measures, and also because they arise, and can be computed, very naturally from the RSP framework.

Some of the methods, however, were developed in the thesis for particular, mostly ecological applications in Chapters 7 and 8. In those cases, the methodology has also been developed with an effort to understand its place in a more general context through abstraction. Specifically, in Chapter 7 the development of maximum likelihood estimation methods for RSPs and in Chapter 8

the definition of the Habitat Functionality metric, are both aimed at particular problems in an ecological setting, but are considered first as general graph-theoretic or network analysis problems. This widens the application domain of the methods but also helps in understanding their behavior based on existing literature in contexts that may not be as apparent from the application-specific formulation.

The balance between generality and specificity can be hard to control when developing methods. With specificity the results in one application can improve, but it would be also desirable to have methods that work for a broad variety of problems. For obtaining such functionality in a general setting, a method should be scrutinized and grounded, both in a theoretical framework as well as with experiments starting from simple ones and moving on to more complex ones. This is done, for instance, with graph node distance measures in the experiments in Chapter 3 and in the examples in Section 2.4.7, as well as with the harmonic RSP centrality in Chapter 8. They are all efforts for trying to obtain insight, by examples, into how the methods work, which can give guidance in their use and can sometimes help in understanding results obtained with the methods.

The main contributions of the thesis are in the development of new distance and centrality measures. Especially, the novel idea, presented in Chapter 3, of considering the free energy related to a probability distribution over paths between two nodes has proved to be fruitful in many respects. Most notably it lead to the definition of the free energy distance, Δ^{FE}, which, as explained and demonstrated in Sections 2.4.6 and 3.3, as well as in [82, 192, 193, 110, 103], is a good candidate for measuring distance between nodes of a graph by taking into account the amount of connections between the nodes and the cluster structure of the network. This holds both in the theoretical context as well as based on experimental evidence. Many other interesting features of free energy were also presented in Chapter 6.

The work also contains results that leave open questions. Many aspects about the RSP based distance measures are still left unknown, like the question of why the RSP dissimilarity in some cases does not satisfy the triangle inequality, which has not been considered separately in the thesis. Another open problem is still the decision of which value should be used for the inverse temperature parameter β in other scenarios besides the one dealt with in Chapter 7, in particular in applications related to clustering or classification. For now, in such applications the parameter is normally set using a held-out tuning dataset. However, especially in clustering, the selection of the value would in an ideal situation arise somehow naturally based on the structure of the graph.

The selection of the value of β also relates to the lost-in-space (LIS) phenomenon [142], which has been one main driver behind the recent proposals in the literature of new distance measures on graphs. Remember, from Section 2.4.5 that the LIS phenomenon states that the random walk distance measures, presented in Section 2.4.2, become meaningless when used on even moderately large graphs. Also, remember that in [103] the logarithmic Laplace transformed hitting time distance, which has a seemingly equivalent definition to the free energy distance, was proved to avoid the LIS phenomenon with some values of the parameter β. The same result should then hold for the free energy distance, and it should give indication about which value of β should be considered in order for the free energy distance to remain global (i.e. avoid the LIS phenomenon) and still reflect the amount of connectivity between node-pairs, which is the main advantage of the random walk distances compared to the shortest path distance on smaller graphs.

Another open issue, which is implicitly raised already in Section 2.4.7, is that in the end the decision of which distance or centrality measure is better than another one is a matter of taste and depends on the point of view. Namely, as discussed with the example graph depicted in Figure 2.1, the ranking of distances in that graph from the central node to the peripheral nodes can have multiple choices that can be considered valid, depending on the application. Perhaps there is more room for defining a broader typology for distance measures on graphs that could make decisions on which measure to consider for a given application easier.

Similar issues arise in the context of the Habitat and Landscape Functionality metrics proposed and studied in Chapter 8. A seemingly functional definition for the HF and LF metrics based on the RSP dissimilarity was shown to behave unexpectedly and undesiredly in experiments. A solution was found in Section 8.4.2 for circumventing the undesired behavior by maintaining the relative entropy level of the system constant when the system changes, instead of keeping constant temperature. This example nevertheless showed the value of carefully designed experiments that test the presumptions of how a method is thought to work. Moreover, when designing methods for quantifying the functionality of a whole graph, it might be possible to state some first principles or axioms that such a method at least should have in a given application context.

The thesis shows that the RSP framework is a fruitful paradigm for graph based data analysis, and that it can help in alleviating problems and limitations of more traditional paradigms. There are still application domains where the RSP framework could still potentially be used. For instance, the RSP distances could be considered in terms of network embedding. As another example, the RSP model could be used for modeling spreading processes of e.g. news

or other information on online networks, or for different kinds of tracking tasks, such as the browsing behavior of customers on online shopping sites. In addition to the unused potential of RSPs, there are still many open questions on the side of theory related to the framework, that, on the other hand, may make the framework even more useful in network analysis tasks than so far. As a result, the research on the RSP framework and similar ideas will hopefully continue to flourish in the near future.

Bibliography

[1] T. Akamatsu. "Cyclic flows, Markov process and stochastic traffic assignment". In: *Transportation Research B* 30.5 (1996), pp. 369–386.

[2] A. H. Al-Mohy, N. J. Higham, and S. D. Relton. "Computing the Fréchet derivative of the matrix logarithm and estimating the condition number". In: *SIAM Journal on Scientific Computing* 35.4 (2013), pp. C394–C410.

[3] M. Alamgir and U. von Luxburg. "Phase transition in the familiy of p-resistances". In: *Neural Information Processing Systems (NIPS)*. 2011.

[4] M. Alamgir and U. Von Luxburg. "Shortest path distance in random k-nearest neighbor graphs". In: *Proceedings of the 29th International Coference on International Conference on Machine Learning*. Omnipress. 2012, pp. 1251–1258.

[5] R. Albert, H. Jeong, and A.-L. Barabási. "Error and attack tolerance of complex networks". In: *Nature* 406.6794 (2000), pp. 378–382.

[6] D. Aldous and J. Fill. *Reversible Markov chains and random walks on graphs*. 2002.

[7] E. Alpaydin. *Introduction to machine learning*. 2nd edition. The MIT Press, 2010.

[8] A. Arenas, A. Fernandez, S. Fortunato, and S. Gomez. "Motif-based communities in complex networks". In: *Journal of Physics A: Mathematical and Theoretical* 41 (2008), p. 224001.

[9] K. Asadi and M. L. Littman. "An alternative softmax operator for reinforcement learning". In: *Proceedings of the International Conference on Machine Learning (ICML)*. 2017, pp. 243–252.

[10] M. G. Azar, V. Gómez, and B. Kappen. "Dynamic policy programming with function approximation". In: *International Conference on Artificial Intelligence and Statistics (AISTAT)*. 2011, pp. 119–127.

[11] M. G. Azar, V. Gómez, and H. J. Kappen. "Dynamic policy programming". In: *Journal of Machine Learning Research* 13.Nov (2012), pp. 3207–3245.

[12] R. Bapat. "Resistance distance in graphs". In: *Mathematics Student-India* 68.1-4 (1999), pp. 87–98.

[13] A. L. Barabasi. *Network science*. Cambridge University Press, 2016.

[14] A.-L. Barabási and R. Albert. "Emergence of scaling in random networks". In: *science* 286.5439 (1999), pp. 509–512.

[15] A. Barrat, M. Barthelemy, and A. Vespignani. *Dynamical processes on complex networks*. Cambridge university press, 2008.

[16] M. Barthelemy. "Betweenness centrality in large complex networks". In: *The European Physical Journal B-Condensed Matter and Complex Systems* 38.2 (2004), pp. 163–168.

[17] F. Bavaud and G. Guex. "Interpolating between random walks and shortest paths: a path functional approach". In: *SocInfo 2012*. Ed. by K. A. et al. Vol. 7710. Lecture Notes in Computer Science. Springer, 2012, pp. 68–81.

[18] F. Bavaud. "Euclidean distances, soft and spectral clustering on weighted graphs". In: *Machine Learning and Knowledge Discovery in Databases*. Springer, 2010, pp. 103–118.

[19] A. Bavelas. "Communication patterns in task-oriented groups." In: *The Journal of the Acoustical Society of America* (1950), pp. 725–730.

[20] M. Benzi and C. Klymko. "A matrix analysis of different centrality measures". In: *arXiv preprint arXiv:1312.6722* (2013).

[21] C. Bishop. *Pattern recognition and machine learning*. Springer, 2006.

[22] M. Bladt and M. Sørensen. "Statistical inference for discretely observed Markov jump processes". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67.3 (2005), pp. 395–410.

[23] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. "Fast unfolding of communities in large networks". In: *Journal of Statistical Mechanics: Theory and Experiment* (2008), P10008.

[24] M. Boguná, F. Papadopoulos, and D. Krioukov. "Sustaining the internet with hyperbolic mapping". In: *Nature communications* 1 (2010), p. 62.

[25] P. Boldi and S. Vigna. "Axioms for Centrality". In: *Internet Mathematics* 10.3-4 (2014), pp. 222–262.

[26] D. Boley, A. Buendia, and G. Golnari. "Random Walk Laplacian and Network Centrality Measures". In: *arXiv preprint arXiv:1808.02912* (2018).

[27] B. Bollobás. *Modern graph theory*. Vol. 184. Springer Science & Business Media, 2013.

[28] P. Bonacich. "Factoring and weighting approaches to status scores and clique identification". In: *Journal of Mathematical Sociology* 2.1 (1972), pp. 113–120.

[29] I. Borg and P. Groenen. *Modern multidimensional scaling: Theory and applications*. Springer, 1997.

[30] S. P. Borgatti and M. G. Everett. "A Graph-theoretic perspective on centrality". In: *Social Networks* 28.4 (2006), pp. 466 –484.

[31] G. E. P. Box, J. S. Hunter, and W. G. Hunter. *Statistics for experimenters : design, innovation, and discovery.* Wiley-Interscience, 2005.

[32] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[33] E. Bozzo and M. Franceschet. "Resistance distance, closeness, and betweenness". In: *Social Networks* 35.3 (2013), pp. 460–469.

[34] U. Brandes. "On variants of shortest-path betweenness centrality and their generic computation". In: *Social Networks* 30.2 (2008), pp. 136–145.

[35] U. Brandes and T. Erlebach, eds. *Network analysis: methodological foundations*. Springer-Verlag, 2005.

[36] U. Brandes and D. Fleischer. "Centrality measures based on current flow". In: *Proceedings of the 22nd Annual Symposium on Theoretical Aspects of Computer Science (STACS)*. 2005, pp. 533–544.

[37] U. Brandes, D. Delling, M. Gaertler, R. Görke, M. Hoefer, Z. Nikoloski, and D. Wagner. "On finding graph clusterings with maximum modularity". In: *Graph-Theoretic Concepts in Computer Science*. Springer. 2007, pp. 121–132.

[38] U. Brandes, D. Delling, M. Gaertler, R. G?rke, M. Hoefer, Z. Nikoloski, and D. Wagner. "On modularity clustering". In: *IEEE Transactions on Knowledge and Data Engineering* 20 (2008), pp. 172–188. ISSN: 1041-4347.

[39] Z. Burda, J. Duda, J.-M. Luck, and B. Waclaw. "Localization of the maximal entropy random walk". In: *Physical review letters* 102.16 (2009), p. 160602.

[40] S. Cantrell, C. Cosner, and S. Ruan. *Spatial ecology*. CRC Press, 2009.

[41] T. Chakraborty, A. Dalmia, A. Mukherjee, and N. Ganguly. "Metrics for community analysis: A survey". In: *ACM Computing Surveys (CSUR)* 50.4 (2017), p. 54.

[42] A. K. Chandra, P. Raghavan, W. L. Ruzzo, R. Smolensky, and P. Tiwari. "The Electrical Resistance of a Graph Captures its Commute and Cover Times". In: *Annual ACM Symposium on Theory of Computing* (1989), pp. 574–586.

[43] P. Chebotarev and E. Shamis. "The matrix-forest theorem and measuring relations in small social groups". In: *Automation and Remote Control* 58.9 (1997), pp. 1505–1514.

[44] P. Chebotarev and E. Shamis. "The forest metrics for graph vertices". In: *Electronic Notes in Discrete Mathematics* 11 (2002), pp. 98–107.

[45] P. Chebotarev. "A class of graph-geodetic distances generalizing the shortest-path and the resistance distances". In: *Discrete Applied Mathematics* 159.5 (2011), pp. 295–302.

[46] P. Chebotarev. "The walk distances in graphs". In: *Discrete Applied Mathematics* 160.10-11 (2012), pp. 1484–1500.

[47] C.-L. B. Chetkiewicz, C. C. St. Clair, and M. S. Boyce. "Corridors for conservation: integrating pattern and process". In: *Annu. Rev. Ecol. Evol. Syst.* 37 (2006), pp. 317–342.

[48] F. Chung and L. Lu. *Complex Graphs and Networks*. American Mathematical Society, 2006.

[49] J. Cook. *Basic properties of the soft maximum*. Unpublished manuscript available from www.johndcook.com/blog/2010/01/13/soft-maximum. 2011.

[50] B. A. Craig and P. P. Sendi. "Estimation of the transition matrix of a discrete-time Markov chain". In: *Health economics* 11.1 (2002), pp. 33–42.

[51] N. Cressie and C. K. Wikle. *Statistics for spatio-temporal data*. John Wiley & Sons, 2015.

[52] M. R. Dale. *Applying Graph Theory in Ecological Research*. Cambridge University Press, 2017.

[53] J.-C. Delvenne and A.-S. Libert. "Centrality measures and thermodynamic formalism for complex networks". In: *Physical Review E* 83 (4 2011), p. 046117.

[54] R. Devooght, A. Mantrach, I. Kivimäki, H. Bersini, A. Jaimes, and M. Saerens. "Random walks based modularity: application to semi-supervised learning". In: *Proceedings of the 23rd international conference on World wide web*. 2014, pp. 213–224.

[55] M. M. Deza and E. Deza. *Encyclopedia of Distances*. Springer, 2009.

[56] P. J. Diggle. *Statistical analysis of spatial and spatio-temporal point patterns*. Chapman and Hall/CRC, 2013.

[57] L. Donetti and M. Munoz. "Detecting network communities: A new systematic and efficient algorithm". In: *Journal of Statistical Mechanics: Theory and Experiment* P10012 (2004).

[58] S. N. Dorogovtsev and J. F. Mendes. "Evolution of networks". In: *Advances in physics* 51.4 (2002), pp. 1079–1187.

[59] P. G. Doyle and J. L. Snell. *Random Walks and Electric Networks*. The Mathematical Association of America, 1984.

[60] J. Duch and A. Arenas. "Community detection in complex networks using extremal optimization". In: *Physical Review E* 72 (2005).

[61] M. Dunham. *Data Mining: Introductory and Advanced Topics*. Prentice Hall, 2003.

[62] C. Elkan. "Using the Triangle Inequality to Accelerate k-Means". In: *Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003), August 21-24, 2003, Washington, DC, USA*. Ed. by T. Fawcett and N. Mishra. AAAI Press, 2003, pp. 147–153.

[63] P. Erdős and A. Rényi. "On Random Graphs I". In: *Publicationes Mathematicae Debrecen* 6 (1959), pp. 290–294.

[64] E. Estrada. *The structure of complex networks: theory and applications*. Oxford University Press, 2012.

[65] E. Estrada. "Complex networks in the Euclidean space of communicability distances". In: *Physical Review E* 85.6 (2012), p. 066122.

[66] E. Estrada. "The communicability distance in graphs". In: *Linear Algebra and its Applications* 436.11 (2012), pp. 4317–4328.

[67] E. Estrada. *The structure of complex networks*. Oxford University Press, 2012.

[68] E. Estrada, D. J. Higham, and N. Hatano. "Communicability betweenness in complex networks". In: *Physica A: Statistical Mechanics and its Applications* 388.5 (2009), pp. 764–774.

[69] E. Estrada and J. A. Rodriguez-Velazquez. "Subgraph centrality in complex networks". In: *Physical Review E* 71.5 (2005), p. 056103.

[70] J. van Etten. "gdistance: Distances and routes on geographical grids". In: *URL http://CRAN. R-project. org/package= gdistance. R package version* (2012), pp. 1–1.

[71] L. Euler. "Solutio problematis ad geometriam situs pertinentis". In: *Commentarii Academiae Scientiarum Imperialis Petropolitanae* 8 (1736), pp. 128–140.

[72] J. D. Forester, H. K. Im, and P. J. Rathouz. "Accounting for animal movement in estimation of resource selection functions: sampling and data analysis". In: *Ecology* 90.12 (2009), pp. 3554–3565.

[73] M.-J. Fortin, P. M. James, A. MacKenzie, S. J. Melles, and B. Rayfield. "Spatial statistics, spatial regression, and graph theory in ecology". In: *Spatial Statistics* 1 (2012), pp. 100–109.

[74] S. Fortunato. "Community detection in graphs". In: *Physics Reports* 486.3-5 (2010), pp. 75 –174.

[75] S. Fortunato, V. Latora, and M. Marchiori. "Method to find community structures based on information centrality". In: *Physical Review E* 70.5 (2004).

[76] S. Fortunato and D. Hric. "Community detection in networks : A user guide". In: *Physics Reports* 659 (2016), pp. 1–44.

[77] F. Fouss, A. Pirotte, J.-M. Renders, and M. Saerens. "Random-walk computation of similarities between nodes of a graph, with application to collaborative recommendation". In: *IEEE Transactions on Knowledge and Data Engineering* 19.3 (2007), pp. 355–369.

[78] F. Fouss, K. Françoisse, L. Yen, A. Pirotte, and M. Saerens. "An experimental investigation of kernels on graphs for collaborative recommendation and semisupervised classification". In: *Neural Networks* 31 (2012), pp. 53–72.

[79] F. Fouss, M. Saerens, and M. Shimbo. *Algorithms and models for network data and link analysis*. Cambridge University Press, 2016.

[80] R. Fox, A. Pakman, and N. Tishby. "G-learning: taming the noise in reinforcement learning via soft updates". In: *Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence (UAI 2016)*. 2001, pp. 202–211.

[81] K. Françoisse, I. Kivimäki, A. Mantrach, F. Rossi, and M. Saerens. "A bag-of-paths framework for network data analysis". In: *ArXiv preprint arXiv:1302.6766* (2013).

[82] K. Françoisse, I. Kivimäki, A. Mantrach, F. Rossi, and M. Saerens. "A bag-of-paths framework for network data analysis". In: *Neural Networks* 90 (2017), pp. 90–111.

[83] L. Freeman. "A set of measures of centrality based on betweenness". In: *Sociometry* 40.1 (1977), pp. 35–41.

[84] L. Freeman. "Centrality in social networks conceptual clarification". In: *Social Networks* 1.3 (1978-1979), pp. 215 –239.

[85] L. C. Freeman, S. P. Borgatti, and D. R. White. "Centrality in valued graphs: A measure of betweenness based on network flow". In: *Social Networks* 13.2 (1991), pp. 141 –154.

[86] P. Galpern, M. Manseau, and A. Fall. "Patch-based graphs of landscape connectivity: a guide to construction, analysis and application for conservation". In: *Biological conservation* 144.1 (2011), pp. 44–55.

[87] S. Garcia-Diez, E. Vandenbussche, and M. Saerens. "A continuous-state version of discrete randomized shortest-paths, with application to path planning". In: *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*. Dec. 2011, pp. 6570 –6577.

[88] F. Gobel and A. A. Jagers. "Random walks on graphs". In: *Stochastic Processes and their Applications* 2 (1974), pp. 311–336.

[89] K.-I. Goh, B Kahng, and D Kim. "Universal behavior of load distribution in scale-free networks". In: *Physical Review Letters* 87.27 (2001), pp. 278701–278701.

[90] G. Golnari. "Markov Tensor Theory and Cascade, Reachability, and Routing in Complex Networks". PhD thesis. University of Minnesota Digital Conservancy, 2017.

[91] G. Golnari and D. Boley. *Continuum of all-pair shortest-path to all-path via random walk*. Technical report TR 13-016. USA: University of Minnesota, Department of Computer Science and Engineering, 2013.

[92] G. Golnari, Z. Zhang, and D. Boley. "Random Walk Fundamental Tensor and its Applications to Network Analysis". In: *CoRR* abs/1801.08583 (2018). arXiv: 1801.08583. URL: http://arxiv.org/abs/1801.08583.

[93] J. C. Gower and P. Legendre. "Metric and Euclidean properties of dissimilarity coefficients". In: *Journal of classification* 3.1 (1986), pp. 5–48.

[94] C. Grinstead and J. L. Snell. *Introduction to probability, 2nd ed*. The Mathematical Association of America, 1997.

[95] G. Guex. "Interpolating between random walks and optimal transportation routes: Flow with multiple sources and targets". In: *Physica A: Statistical Mechanics and its Applications* 450 (2016), pp. 264–277.

[96] G. Guex and F. Bavaud. "Flow-Based Dissimilarities: Shortest Path, Commute Time, Max-Flow and Free Energy". In: *Data Science, Learning by Latent Structures, and Knowledge Discovery*. Springer, 2015, pp. 101–111.

[97] G. Guex, I. Kivimäki, and M. Saerens. "Randomized Optimal Transport on a Graph: Framework and New Distance Measures". In: *Network Science* (2018). Accepted for publication.

[98] R. Guimera and L. N. Amaral. "Functional cartography of complex metabolic networks". In: *Nature* 433 (2005), pp. 895–899.

[99] A. Gunn. *Rangifer tarandus. The IUCN Red List of Threatened Species 2016: e.T29742A22167140*. 2016. URL: http://www.iucnredlist.org/details/29742/0.

[100] L. S. Hall, P. R. Krausman, and M. L. Morrison. "The habitat concept and a plea for standard terminology". In: *Wildlife Society Bulletin* (1997), pp. 173–182.

[101] I. Hanski and O. Ovaskainen. "The metapopulation capacity of a fragmented landscape". In: *Nature* 404.6779 (2000), pp. 755–758.

[102] D. A. Harville. *Matrix algebra from a statistician's perspective*. Springer-Verlag, 1997.

[103] T. Hashimoto, Y. Sun, and T. Jaakkola. "From random walks to distances on unweighted graphs". In: *Advances in Neural Information Processing Systems 28: Proceedings of the NIPS '15 conference*. MIT Press, 2015.

[104] M. Herbster. "A Triangle Inequality for p-Resistance". In: *NIPS Workshop 2010: Networks Across Disciplines: Theory and Applications*. 2010.

[105] N. J. Higham. *Functions of matrices: theory and computation*. SIAM, 2008.

[106] T. Horvath, T. Gartner, and S. Wrobel. "Cyclic pattern kernels for predictive graph mining". In: *Proceedings of the SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2004)*. 2004, pp. 158–167.

[107] X. Huang, Y. Ariki, and M. Jack. *Hidden Markov models for speech recognition*. Edinburgh University Press, 1990.

[108] F. Iannelli, A. Koher, D. Brockmann, P. Hovel, and I. Sokolov. "Effective distances for epidemics spreading on complex networks". In: *Phusical Review E* 95.1 (2017), p. 012313.

[109] O. Ivanciuc, T.-S. Balaban, and A. T. Balaban. "Design of topological indices. Part 4. Reciprocal distance matrix, related local vertex invariants and topological indices". In: *Journal of Mathematical Chemistry* 12.1 (1993), pp. 309–318.

[110] V. Ivashkin and P. Chebotarev. "Do logarithmic proximity measures outperform plain ones in graph clustering?" In: *International Conference on Network Analysis*. Springer. 2016, pp. 87–105.

[111] E. T. Jaynes. "Information theory and statistical mechanics". In: *Physical Review* 106 (1957), pp. 620–630.

[112] H. J. Kappen, V. Gómez, and M. Opper. "Optimal control as a graphical model inference problem". In: *Machine learning* 87.2 (2012), pp. 159–182.

[113] J. N. Kapur and H. K. Kesavan. *Entropy optimization principles with applications*. Academic Press, 1992.

[114] L. Katz. "A new status index derived from sociometric analysis". In: *Psychometrika* 18.1 (1953), pp. 39–43.

[115] I. Kivimäki, M. Shimbo, and M. Saerens. "Developments in the theory of randomized shortest paths with a comparison of graph node distances". In: *Physica A: Statistical Mechanics and its Applications* 393 (2014), pp. 600–616.

[116] I. Kivimäki, A. Panchenko, A. Dessy, D. Verdegem, P. Francq, C. Fairon, H. Bersini, and M. Saerens. "A graph-based approach to skill extraction from text". In: *Graph-Based Methods for Natural Language Processing* (2013), pp. 79–87.

[117] I. Kivimäki, B. Lebichot, J. Saramäki, and M. Saerens. "Two betweenness centrality measures based on Randomized Shortest Paths". In: *Scientific Reports* 6.19668 (2016).

[118] D. J. Klein and M. Randic. "Resistance distance". In: *Journal of Mathematical Chemistry* 12.1 (1993), pp. 81–95.

[119] D. Klein. "Centrality measure in graphs". In: *Journal of mathematical chemistry* 47.4 (2010), pp. 1209–1223.

[120] D. Klein and H.-Y. Zhu. "Distances and volumina for graphs". In: *Journal of mathematical chemistry* 23.1 (1998), pp. 179–195.

[121] D. J. Klein. "Resistance-distance sum rules". In: *Croatica chemica acta* 75.2 (2002), pp. 633–649.

[122] D. E. Knuth. "Two notes on notation". In: *The American Mathematical Monthly* 99.5 (1992), pp. 403–422.

[123] E. Kolaczyk. *Statistical analysis of network data: methods and models*. Springer, 2009.

[124] D. Koschützki, K. Lehmann, L. Peeters, S. Richter, D. Tenfelde-Podehl, and O. Zlotowski. "Centrality Indices". In: *Network Analysis*. Ed. by U.

Brandes and T. Erlebach. Vol. 3418. Lecture Notes in Computer Science. Berlin/Heidelberg: Springer, 2005, pp. 16–61.

[125] A. Lancichinetti and S. Fortunato. "Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities". In: *Phys. Rev. E* 80 (1 July 2009), p. 016118.

[126] A. Lancichinetti and S. Fortunato. "Community detection algorithms: a comparative analysis". In: *Physical review E* 80.5 (2009), p. 056117.

[127] A. Lancichinetti, S. Fortunato, and F. Radicchi. "Benchmark graphs for testing community detection algorithms". In: *Physical Review E* 78.4 (2008), p. 046110.

[128] B. Lebichot and M. Saerens. "A bag-of-paths node criticality measure". In: *Neurocomputing* 275 (2018), pp. 224–236.

[129] B. Lebichot, I. Kivimäki, K. Françoisse, and M. Saerens. "Semisupervised Classification Through the Bag-of-Paths Group Betweenness". In: *Neural Networks and Learning Systems, IEEE Transactions on* 25.6 (2014), pp. 1173–1186.

[130] B. Lebichot, G. Guex, I. Kivimäki, and M. Saerens. "A Constrained Randomized Shortest-Paths Framework for Optimal Exploration". In: *arXiv preprint arXiv:1807.04551* (2018).

[131] J. A. Lee and M. Verleysen. *Nonlinear dimensionality reduction*. Springer Science & Business Media, 2007.

[132] S. R. Lele. "A new method for estimation of resource selection probability function". In: *Journal of Wildlife Management* 73.1 (2009), pp. 122–127.

[133] S. R. Lele and J. L. Keim. "Weighted distributions and estimation of resource selection probability functions". In: *Ecology* 87.12 (2006), pp. 3021–3028.

[134] S. R. Lele, J. L. Keim, and P. Solymos. *ResourceSelection: Resource Selection (Probability) Functions for Use-Availability Data*. R package version 0.3-2. 2017. URL: https://CRAN.R-project.org/package=ResourceSelection.

[135] S. R. Lele, E. H. Merrill, J. Keim, and M. S. Boyce. "Selection, use, choice and occupancy: clarifying concepts in resource selection studies". In: *Journal of Animal Ecology* 82.6 (2013), pp. 1183–1191.

[136] T. Lewis. *Network Science: Theory and Applications*. Wiley, 2009.

[137] Y. Li, Z.-L. Zhang, and D. Boley. "The Routing Continuum from Shortest-Path to All-Path: A Unifying Theory". In: *Proceedings of the 2011 31st International Conference on Distributed Computing Systems*. ICDCS '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 847–856.

[138] Y. Li, Z.-L. Zhang, and D. Boley. "From shortest-path to all-path: The routing continuum theory and its applications". In: *IEEE Transactions on Parallel & Distributed Systems* 1 (2013), p. 1.

[139] D. Liben-Nowell and J. Kleinberg. "The link-prediction problem for social networks". In: *Journal of the American Society for Information Science and Technology* 58.7 (2007), pp. 1019–1031.

[140] L. Lovász. "Random walks on graphs: A survey". In: *Combinatorics, Paul Erdős is eighty* 2.1 (1993), pp. 1–46.

[141] L. Lü and T. Zhou. "Link prediction in complex networks: A survey". In: *Physica A: Statistical Mechanics and its Applications* 390.6 (2011), pp. 1150 –1170.

[142] U. von Luxburg, A. Radl, and M. Hein. "Hitting and commute times in large random neighborhood graphs." In: *Journal of Machine Learning Research* 15.1 (2014), pp. 1751–1798.

[143] L. v. d. Maaten and G. Hinton. "Visualizing data using t-SNE". In: *Journal of machine learning research* 9.Nov (2008), pp. 2579–2605.

[144] S. A. Macskassy and F. Provost. "Classification in Networked Data: A Toolkit and a Univariate Case Study". In: *J. Mach. Learn. Res.* 8 (May 2007), pp. 935–983. ISSN: 1532-4435.

[145] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval*. Vol. 1. Cambridge University Press Cambridge, 2008.

[146] A. Mantrach, L. Yen, J. Callut, K. Francoisse, M. Shimbo, and M. Saerens. "The sum-over-paths covariance kernel: A novel covariance measure between nodes of a directed graph". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 32.6 (2010), pp. 1112–1126.

[147] P. Metzner, E. Dittmer, T. Jahnke, and C. Schütte. "Generator estimation of Markov jump processes". In: *Journal of Computational Physics* 227.1 (2007), pp. 353–375.

[148] C. D. Meyer. *Matrix analysis and applied linear algebra*. Vol. 71. Siam, 2000.

[149] G. Milligan and M. Cooper. "An examination of procedures for determining the number of clusters in a data set". In: *Psychometrika* 50.2 (1985), pp. 159–179.

[150] K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

[151] B. Nadler, N. Srebro, and X. Zhou. "Semi-supervised learning with the graph Laplacian: The limit of infinite unlabelled data". In: *Neural Information Processing Systems (NIPS)*. 2009.

[152] C. Nellemann, I. Vistnes, P. Jordhøy, O. Strand, and A. Newton. "Progressive impact of piecemeal infrastructure development on wild reindeer". In: *Biological Conservation* 113.2 (2003), pp. 307–317.

[153] J. R. Newman. *The World of Mathematics*. Phoenix Edition Series Volume 1. Dover Publications, 2000.

[154] M. E. Newman. *Networks : An Introduction*. Oxford University Press, 2010.

[155] M. E. J. Newman. "Erratum: Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality [Phys. Rev. E 64, 016132 (2001)]". In: *Phys. Rev. E* 73 (Mar. 2006), p. 039906.

[156] M. Newman. "Analysis of weighted networks". In: *Physical Review E* 70 (2004), p. 056131.

[157] M. Newman. "A measure of betweenness centrality based on random walks". In: *Social Networks* 27.1 (2005), pp. 39–54.

[158] M. Newman. "Modularity and community structure in networks". In: *Proceedings of the National Academy of Sciences (USA)* 103 (2006), pp. 8577–8582.

[159] M. Newman and M. Girvan. "Fast algorithm for detecting community structure in networks". In: *Physical Review E* 69 (2004), p. 066133.

[160] M. Newman and M. Girvan. "Finding and evaluating community structure in networks". In: *Physical Review E* 69 (2004), p. 026113.

[161] C. H. Nguyen and H. Mamitsuka. "New resistance distances with global information on large graphs". In: *Artificial Intelligence and Statistics*. 2016, pp. 639–647.

[162] A. Noack and R. Rotta. "Multi-level Algorithms for Modularity Clustering". In: *International Symposium on Experimental Algorithms (SEA)*. Vol. 5526. Lecture Notes in Computer Science. Springer, 2009, pp. 257–268.

[163] G. Odor. "Global information loss and criticality in resistance matrices". MA thesis. Central European University, 2017.

[164] L. Page, S. Brin, R. Motwani, and T. Winograd. "The PageRank citation ranking: Bringing order to the web". In: *Technical Report 1999-0120, Computer Science Department, Stanford University* (1999).

[165] M. Panzacchi, B. Van Moorter, and O. Strand. "A road in the middle of one of the last wild reindeer migration routes in Norway: crossing behaviour and threats to conservation". In: *Rangifer* 33.2 (2013), pp. 15–26.

[166] M. Panzacchi, B. V. Moorter, P. Jordhøy, and O. Strand. "Learning from the past to predict the future: using archaeological findings and GPS data to quantify reindeer sensitivity to anthropogenic disturbance in Norway". In: *Landscape Ecology* 28.5 (May 2013), pp. 847–859.

[167] M. Panzacchi, B. Van Moorter, O. Strand, L. E. Loe, and E. Reimers. "Searching for the fundamental niche using individual-based habitat selection modelling across populations". In: *Ecography* 38.7 (2015), pp. 659–669.

[168] M. Panzacchi, B. Van Moorter, O. Strand, M. Saerens, I. Kivimäki, C. C. St. Clair, I. Herfindal, and L. Boitani. "Predicting the continuum between corridors and barriers to animal movements using Step Selection Functions and Randomized Shortest Paths". In: *Journal of Animal Ecology* 85.1 (2016), pp. 32–42.

[169] R. Pastor-Satorras and A. Vespignani. "Epidemic spreading in scale-free networks". In: *Physical review letters* 86.14 (2001), p. 3200.

[170] L. Peliti. *Statistical Mechanics in a Nutshell*. In a Nutshell. Princeton University Press, 2011.

[171] J. I. Perotti and O. V. Billoni. "Smart random walkers: the cost of knowing the path". In: *Physical Review E* 86.1 (2012), p. 011120.

[172] F. Qin, A. Auerbach, and F. Sachs. "Maximum likelihood estimation of aggregated Markov processes". In: *Proceedings of the Royal Society of London B: Biological Sciences* 264.1380 (1997), pp. 375–383.

[173] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria, 2015. URL: https://www.R-project.org/.

[174] W. M. Rand. "Objective criteria for the evaluation of clustering methods". In: *Journal of the American Statistical association* 66.336 (1971), pp. 846–850.

[175] G. Ranjan and Z.-L. Zhang. "Geometry of complex networks and topological centrality". In: *Physica A: Statistical Mechanics and its Applications* 392.17 (2013), pp. 3833–3845.

[176] F. Rossi and N. Villa-Vialaneix. "Optimizing an organized modularity measure for topographic graph clustering: A deterministic annealing approach". In: *Neurocomputing* 73.7-9 (2010), pp. 1142 –1163.

[177] J. Rubin, O. Shamir, and N. Tishby. "Trading value and information in MDPs". In: *Decision making with imperfect decision makers*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 57–74.

[178] S. J. Russell and P. Norvig. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited, 2016.

[179] D. G. Saari. "Explaining All Three-Alternative Voting Outcomes". In: *Journal of Economic Theory* 87.2 (1999), pp. 313–355.

[180] G. Sabidussi. "The centrality index of a graph". In: *Psychometrika* 31.4 (1966), pp. 581–603.

[181] M. Saerens, F. Fouss, L. Yen, and P. Dupont. "The principal components analysis of a graph, and its relationships to spectral clustering". In: *European Conference on Machine Learning*. Springer. 2004, pp. 371–383.

[182] M. Saerens, Y. Achbany, F. Fouss, and L. Yen. "Randomized shortest-path problems: Two related models". In: *Neural Computation* 21.8 (2009), pp. 2363–2404.

[183] S. Saura and L. Pascual-Hortal. "A new habitat availability index to integrate connectivity in landscape conservation planning: comparison with existing indices and application to a case study". In: *Landscape and Urban Planning* 83.2 (2007), pp. 91–103.

[184] S. E. Schaeffer. "Graph Clustering". In: *Computer Science Review* 1.1 (2007), pp. 27–64.

[185] I. J. Schoenberg. "Metric spaces and positive definite functions". In: *Transactions of the American Mathematical Society* 44.3 (1938), pp. 522–536.

[186] B Schölkopf and A. Smola. "Learning with Kernels: support vector machines, regularization, optimization, and beyond". In: *Adaptive computation and machine learning* (2002).

[187] G. Seber. *A matrix handbook for statisticians*. Wiley, 2008.

[188] R. Sedgewick. *Algorithms, 4th ed*. Addison-Wesley, 2011.

[189] J. Shawe-Taylor and N. Cristianini. *Kernel methods for pattern analysis*. Cambridge University Press, 2004.

[190] C. R. Shelton and G. Ciardo. "Tutorial on structured continuous-time Markov processes". In: *Journal of Artificial Intelligence Research* 51 (2014), pp. 725–778.

[191] T. Silva and L. Zhao. *Machine learning in complex networks*. Springer, 2016.

[192] F. Sommer, F. Fouss, and M. Saerens. "Comparison of graph node distances on clustering tasks". In: *International Conference on Artificial Neural Networks*. Springer. 2016, pp. 192–201.

[193] F. Sommer, F. Fouss, and M. Saerens. "Modularity-Driven Kernel k-means for Community Detection". In: *International Conference on Artificial Neural Networks*. Springer. 2017, pp. 423–433.

[194] K. Stephenson and M. Zelen. "Rethinking centrality: Methods and examples". In: *Social Networks* 11.1 (1989), pp. 1 –37.

[195] A. Strehl and J. Ghosh. "Cluster ensembles—a knowledge reuse framework for combining multiple partitions". In: *Journal of machine learning research* 3.Dec (2002), pp. 583–617.

[196] A. Tahbaz and A. Jadbabaie. "A one-parameter family of distributed consensus algorithms with boundary: from shortest paths to mean hitting times". In: *Proceedings of IEEE Conference on Decision and Control*. 2006, pp. 4664–4669.

[197] R. Tamassia. *Handbook of graph drawing and visualization*. Chapman and Hall/CRC, 2013.

[198] J. M. Tang and Y. Saad. "A probing method for computing the diagonal of a matrix inverse". In: *Numerical Linear Algebra with Applications* 19.3 (2012), pp. 485–501.

[199] M. Tasgin and H. Bingol. "Community detection in complex networks using genetic algorithm". In: *Condensed Matter, 0604419* (2006).

[200] J. B. Tenenbaum, V. d. Silva, and J. C. Langford. "A Global Geometric Framework for Nonlinear Dimensionality Reduction". In: *Science* 290.5500 (2000), pp. 2319–2323.

[201] M. Thelwall. *Link Analysis: An Information Science Approach*. Elsevier, 2004.

[202] S. Theodoridis and K. Koutroumbas. *Pattern Recognition*. 4th. Academic Press, 2009.

[203] E. Theodorou, D Krishnamurthy, and E. Todorov. "From information theoretic dualities to path integral and kullback-leibler control: Continuous and discrete time formulations". In: *The Sixteenth Yale Workshop on Adaptive and Learning Systems*. 2013.

[204] E. A. Theodorou and E. Todorov. "Relative entropy and free energy dualities: Connections to path integral and kl control". In: *Proceedings of the 51st IEEE Conference on Decision and Control (CDC 2012)*. IEEE. 2012, pp. 1466–1473.

[205] E. Todorov. "Linearly-solvable Markov decision problems". In: *Advances in Neural Information Processing Systems 19 (NIPS 2006)*. MIT Press, 2006, pp. 1369–1375.

[206] J. Travers and S. Milgram. "An Experimental Study of the Small World Problem". In: *Sociometry* 32.4 (1969), pp. 425–443.

[207] F. Tutzauer. "Entropy as a measure of centrality in networks characterized by path-transfer flow". In: *Social networks* 29.2 (2007), pp. 249–265.

[208] I. Vistnes, C. Nellemann, P. Jordhøy, and O. Strand. "Wild reindeer: impacts of progressive infrastructure development on distribution and range use". In: *Polar Biology* 24.7 (July 2001), pp. 531–537.

[209] U. von Luxburg. "A tutorial on spectral clustering". In: *Statistics and Computing* 17.4 (2007), pp. 395–416.

[210] U. Von Luxburg, A. Radl, and M. Hein. "Getting lost in space: Large sample analysis of the resistance distance". In: *Advances in Neural Information Processing Systems*. 2010, pp. 2622–2630.

[211] S. Wasserman and K. Faust. *Social network analysis: methods and applications*. Cambridge University Press, 1994.

[212] D. J. Watts and S. H. Strogatz. "Collective dynamics of 'small-world' networks". In: *Nature* 393.6684 (1998), pp. 440–442.

[213] S. White and P. Smyth. "Algorithms for Estimating Relative Importance in Networks". In: *Proceedings of the ninth ACM SIGKDD International Conference on Knowledge Discovery and Data mining (KDD 2003)*. 2003, pp. 266–275.

[214] P. Wilczek. "Novel centrality measures and distance-related topological indices in network data mining". In: *Silesian Journal of Pure and Applied Mathematics* 7 (2017).

[215] L. Yen, A. Mantrach, M. Shimbo, and M. Saerens. "A family of dissimilarity measures between nodes generalizing both the shortest-path and the commute-time distances". In: *Proceedings of the 14th SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2008)*. 2008, pp. 785–793.

[216] L. Yen, F. Fouss, C. Decaestecker, P. Francq, and M. Saerens. "Graph nodes clustering based on the commute-time kernel". In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer. 2007, pp. 1037–1045.

[217] L. Yen, F. Fouss, C. Decaestecker, P. Francq, and M. Saerens. "Graph nodes clustering with the sigmoid commute-time kernel: A comparative study". In: *Data & Knowledge Engineering* 68.3 (2009), pp. 338–361.

[218] K. Yeung and W. Ruzzo. "Details of the adjusted Rand index and clustering algorithms". In: *Bioinformatics* 17(9) (2001), pp. 763–774.

[219] X. Zhu and A. Goldberg. *Introduction to Semi-Supervised Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning Series. Morgan & Claypool, 2009.