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

Ilkka Kivimäki

ICTEAM institute, Université catholique de Louvain, Louvain-la-Neuve, Belgium

Masashi Shimbo

Nara Institute of Science and Technology, Ikoma, Japan

Marco Saerens

ICTEAM institute, Université catholique de Louvain, Louvain-la-Neuve, Belgium

Abstract

There have lately been several suggestions for parametrized distances on a graph that generalize the shortest path distance and the commute time or resistance distance. The need for developing such distances has risen from the observation that the above-mentioned common distances in many situations fail to take into account the global structure of the graph. In this article, we develop the theory of one family of graph node distances, known as the randomized shortest path dissimilarity, which has its foundation in statistical physics. We show that the randomized shortest path dissimilarity can be easily computed in closed form for all pairs of nodes of a graph. Moreover, we come up with a new definition of a distance measure that we call the free energy distance. The free energy distance can be seen as an upgrade of the randomized shortest path dissimilarity as it defines a metric, in addition to which it satisfies the graph-geodetic property. The derivation and computation of the free energy distance are also straightforward. We then make a comparison between a set of generalized distances that interpolate between the shortest path distance and the commute time, or resistance distance. This comparison focuses on the applicability of the distances in graph node clustering and classification. The comparison, in general, shows that the parametrized distances perform well in the tasks. In particular, we see that the results obtained with the free energy distance are among the best in all the experiments.

Keywords:
graph node distances, free energy, randomized shortest paths, shortest path distance, commute time distance, resistance distance, clustering, semisupervised classification

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 [1, 2, 3, 4, 5, 6, 7, 8, 9]. 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.

The most common ways of defining a distance on a graph are to consider either the lengths of the shortest paths between nodes, leading to the definition of the *shortest path (SP) distance*, or the expected lengths of random walks on the graph, which can be used to derive the *commute time (CT) distance* [10]. The CT distance is known to equal the *resistance distance* [11, 12] up to a constant factor [13]. In this paper, we examine generalized distances on graphs that interpolate, depending on a parameter, between the shortest path distance and the commute time or resistance distance.

The paper contains several separate contributions: First, we develop the theory of one generalized distance, the *randomized shortest path (RSP) dissimilarity* [14, 15]. 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. 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 (FE) distance* actually coincides with the *potential distance*, proposed in recent literature in a more ad hoc manner [16]. However, our new derivation gives a nice theoretical background for this distance. Finally, we make a comparison of the behavior and performance of different generalized graph node 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.

The paper is structured as follows: In Section 2, we define the terms and notation used in the paper. In our framework, we consider graphs where the edges are assigned weights and costs, which can be independent of each other. In Section 3, we recall the definitions of the common distances on graphs. We also present a surprising result related to the generalization of the commute time distance considering costs, namely that the distance based on costs equals the commute time distance, up to a constant factor. In Section 4, we revisit the definition of the RSP dissimilarity [14, 15]. We then derive the closed form algorithm, mentioned above, for computing it, and then formulate the definition of the FE distance. In Section 5, we present other parametrized distances on graphs interpolating between the SP and CT distances that have been defined in recent literature. Section 6 contains the comparison of the RSP dissimilarity,

the FE distance and the generalized distances defined in Section 5. Finally, Section 7 sums up the content of the article.

2. Terminology and notation

We first go through the terminology and notation used in this paper. We denote by $G = (V, E)$ a graph G consisting of a node set $V = \{1, 2, \ldots, n\}$ and an edge set $E = \{(i, j)\}$. Nodes i and j such that $(i, j) \in E$ are called *adjacent* or *connected*. Each graph can be represented as an adjacency matrix $\mathbf{A}$, where the elements a_{ij} are called *affinities*, or *weights*, interchangeably. For unweighted graphs $a_{ij} = 1$ if $(i, j) \in E$, for weighted graphs $a_{ij} > 0$ if $(i, j) \in E$ and in both cases $a_{ij} = 0$ if $(i, j) \notin E$. The affinities can be interpreted as representing the degree of similarity between connected nodes. A *path*, or *walk*, interchangeably, on the graph G is a sequence of nodes $\wp = (i_0, \ldots, i_T)$, where $T \geq 0$ and $(i_\tau, i_{\tau+1}) \in E$ for all $\tau = 0, \ldots, T-1$. The *length* of the path, or walk, $\wp$, is then T. Note that throughout this article we include zero-length paths $(i), i \in V$ in the definition of a path, although in some contexts it may be more appropriate to disallow this by setting $T \geq 1$ in the definition. Moreover, we define *absorbing*, or *hitting* paths as paths which contain the terminal node only once. Thus a path $\wp$ is an absorbing path if $\wp = (i_0, \ldots i_T)$, where $i_T \neq i_\tau$ for all $\tau = 0, \ldots, T-1$.

In addition to affinities, the edges of a graph can be assigned *costs*, c_{ij}, such that $0 < c_{ij} < \infty$ if $(i, j) \in E$. The cost of a path $\wp$ is the sum of the costs along the path[1] $\widetilde{c}(\wp) = \sum_{(i,j)\in\wp} c_{ij}$. In principle, we do not define costs for unconnected pairs of nodes, but when making matrix computations, we assign the corresponding matrix elements a very large number (compared to other costs). When there is no natural cost assigned to the edges, a common convention is to define the costs as reciprocals of the affinities $c_{ij} = 1/a_{ij}$. This applies both for unweighted and weighted graphs. This way the edge weights and costs are analogous to conductance and resistance, respectively, in an electric network. In the experiments, in Section 6 of this paper, we always use this conversion for determining costs from affinities. However, in the theory that we present in Sections 3-4, we consider that the costs can also be assigned independently of the affinities, allowing a more general setting. This can be useful in many applications because links can often have a two-sided nature, on one hand based on the structure of the graph and on the other hand based on internal features of the edges. One such example can be a toll road network, where the affinities represent the proximities of places and the costs represent toll costs of traversing a road. This interpretation is especially useful in graph analysis based on a probabilistic framework, wherein the emphasis of this paper also lies. Experiments that take advantage of the possible independence between affinities and costs are left for further work.

[1] Throughout the article we will use the tilde ($\sim$) to differentiate quantities related to paths from quantities related to edges.

We denote by $\mathbf{e}$ the $n \times 1$ vector whose each element is 1. For an $n \times n$ square matrix $\mathbf{A}$, let $\mathbf{Diag}(\mathbf{A})$ denote the $n \times n$ diagonal matrix whose diagonal elements are the diagonal elements of $\mathbf{A}$ and by $\mathbf{diag}(\mathbf{A})$ the $n \times 1$ vector of the diagonal elements of $\mathbf{A}$. Likewise, for an $n \times 1$ vector $\mathbf{v}$, $\mathbf{Diag}(\mathbf{v})$ denotes the $n \times n$ diagonal 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; these should not to be confused with the matrix exponential and matrix logarithm which are not used in this article. Furthermore, we use $\mathbf{A} \circ \mathbf{B}$ and $\mathbf{A} \div \mathbf{B}$ for elementwise product and division, respectively, of $n \times m$ matrices $\mathbf{A}$ and $\mathbf{B}$.

3. The shortest path and commute time distances

The most common distance measure between two nodes of a graph is the *shortest path (SP) distance*. As introduced earlier in Section 1, in our framework, we consider costs associated to the edges of a graph. Hence, we define the SP distance between two nodes as the *minimal cost* of a path between the nodes. This applies for both unweighted and weighted undirected graphs. Also, recall that edge costs can be independent of the affinities a_{ij}. Thus, our definition of the SP distance does not necessarily depend on the affinities, either, but only on the costs. In addition, we define the *unweighted SP distance* between two nodes as the *minimal length* of a path between the nodes.[2]

The SP distance can be used, for example, for estimating the geodesic distance between points when assuming that the graph points lie on a manifold. One popular method to use this idea is the Isomap algorithm [18] for nonlinear dimensionality reduction. One major drawback of the SP distance is that it does not take into account the global structure of the network. In particular, it does not consider the number of connections that exist between nodes, only the length of the shortest one.

Another interesting and well-known graph distance measure is the *commute time (CT) distance* [10] which is defined between two nodes as the *expected length* of paths that a random walker moving along the edges of the graph has to take from one node to the other and back. The transition probability of the walker moving from a node i to an adjacent node j is given naturally as

$$p_{ij}^{\mathrm{ref}} = \frac{a_{ij}}{\sum_k a_{ik}}, \tag{1}$$

where the superscript "ref" emphasizes that these probabilities will later be considered as reference probabilities, when defining new path probabilities. We refer to a random walk based on these probabilities as a *natural* random walk.

[2]Some authors, e.g. Chebotarev in [17], instead call the SP distance based on the edge weights the *weighted* SP distance and use the term SP distance only for the distance based on the number of edges on paths. However, there the costs (or resistances) are fixed as the reciprocals of affinities, unlike in our approach.

The CT distance is well known to be proportional to the *resistance distance* [13] which is defined as the effective resistance of a network when it is considered as an electric circuit where the poles of a unit volt battery have been attached to the nodes between which the distance is being measured [11, 12].

We can also define a generalization of the commute time distance that considers costs of paths instead of their lengths. More precisely, we define the *commute cost (CC) distance* as the *expected cost* of the paths that a random walker will take when moving from a node to another and back according to the transition probabilities p_{ij}^{ref} [19]. An interesting, somewhat unintuitive result in this context is that *in an undirected graph, the commute cost distance is proportional to the commute time distance*. We provide the proof for this novel result in Appendix A. Here, it is important to remember that the costs are independent of the weights and vice versa. Thus the same applies between the costs and the reference transition probabilities of the random walker. This result means that the commute time, commute cost and resistance distances are all the same up to a constant factor. In other words, in most practical applications they will give the exact same results, because in practice the interest lies in the ratios of pairwise distances instead of the distances themselves.

A nice thing about the commute time, commute cost and resistance distances, when compared to the SP distance, is that they take into account the number of different paths connecting pairs of nodes. As a result, these distances have been used in different applications of network science with beneficial results. However, it has been noted that in a large graph these distances are affected largely by the stationary distribution of the natural random walk on the graph [20]. It has recently been shown [21, 22] that in certain models, as the size of a graph grows, the resistance distance (and thus the CT and CC distances as well) between two nodes become only dependent on trivial local properties of the nodes. More specifically, the resistance distance between two nodes approaches the sum of the reciprocals of the degrees of these two nodes. This result is presented in [22] for random geometric graphs where nodes and edges are added to the graph according to the underlying density. The authors claim that the result can be shown to hold also for other models, such as power law graphs in cases where the minimum degree of the graph grows with the number of nodes.

An intuitive explanation of this phenomenon is that in very large graphs a random walker has too many paths to follow and the chance of the walker finding its destination node becomes more dependent on the number of edges (instead of paths, per se) that lead to that node. This undesirable phenomenon serves as one motivation for defining new graph node distances that choose an alternative between the SP and CT distances. This idea already appeared in the development of the RSP dissimilarity [14, 15], with the main motivations in path planning and simply in proposing a distance interpolating between the SP and CT distances. In the following, we first recapitulate the definition of the RSP dissimilarity and then develop the theory behind it. After this we will review other generalized graph node distances and compare their use in data analysis tasks.

4. Advances in the randomized shortest paths framework

The RSP dissimilarity was defined in [14] inspired by [23] and its theory has been extended further in [15] and [24]. It is based on the interpretation of random walks in terms of statistical physics. The definition involves a parameter[3] β which is analogous to the inverse temperature of a thermodynamical system. The RSP dissimilarity is shown to converge to the SP distance as $\beta \to \infty$ and to the CT 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, meaning that it is only a semimetric. In the experimental part of the paper, 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. [25]. 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 semimetric. Furthermore, it has been already shown that using the RSP dissimilarity with its intermediate parameter values provides good results in graph node clustering [14].

4.1. The randomized shortest path dissimilarity

We shall first recall the definition of the RSP dissimilarity [14, 15]. It is defined by considering a random walker choosing an *absorbing*, or *hitting*, path from a starting node s to a destination node t, meaning that node t can appear in the path only once, as the ending node. Let $\mathcal{P}_{st}$ denote the set of such paths and let $\wp = (i_1 = s, \ldots, i_T = t) \in \mathcal{P}_{st}$. The *reference probability* of the path $\wp$ is $\widetilde{\mathrm{P}}^{\mathrm{ref}}_{st}(\wp) = p^{\mathrm{ref}}_{i_1 i_2} \cdots p^{\mathrm{ref}}_{i_{T-1} i_T}$. It simply corresponds to the likelihood of the paths, i.e., the product of the transition probabilities along the path.

In the RSP model, the randomness of the walker is constrained by fixing the relative entropy between the distribution over paths according to the reference probabilities and the distribution over paths that the walker actually chooses from. With this constraint, the walker then chooses the path from the probability distribution that minimizes the *expected cost*[4]

$$\overline{c}(\widetilde{\mathrm{P}}_{st}) = \sum_{\wp \in \mathcal{P}_{st}} \widetilde{\mathrm{P}}_{st}(\wp)\widetilde{c}(\wp)$$

of going from node s to node t. Thus, the relative entropy constraint controls the *exploration* of the walker, whereas the minimization of expected cost controls

[3] The authors in the referred work use θ in place of β.

[4] Notice the difference in notation between the expected cost (denoted with a bar as $\overline{c}$) and the cost of a particular path (denoted with a tilde as $\widetilde{c}$).

its *exploitation*. Formally, the walker moves according to the distribution

$$\widetilde{\mathrm{P}}_{st}^{\mathrm{RSP}} = \arg\min_{\widetilde{\mathrm{P}}_{st}} \overline{c}(\widetilde{\mathrm{P}}_{st}) \;\text{ subject to }\; \begin{cases} J(\widetilde{\mathrm{P}}_{st} \| \widetilde{\mathrm{P}}_{st}^{\mathrm{ref}}) = J_0 \\ \sum_{\wp \in \mathcal{P}_{st}} \widetilde{\mathrm{P}}_{st}(\wp) = 1 \end{cases} \tag{2}$$

where $J(\widetilde{\mathrm{P}}_{st} \| \widetilde{\mathrm{P}}_{st}^{\mathrm{ref}}) = \sum_{\wp \in \mathcal{P}_{st}} \widetilde{\mathrm{P}}_{st}(\wp) \log(\widetilde{\mathrm{P}}_{st}(\wp) / \widetilde{\mathrm{P}}_{st}^{\mathrm{ref}}(\wp))$ is the relative entropy of the distribution with respect to $\widetilde{\mathrm{P}}^{\mathrm{ref}}$, which is constrained to a value J_0. The minimization is shown [14, 15] to result in a Boltzmann distribution

$$\widetilde{\mathrm{P}}_{st}^{\mathrm{RSP}}(\wp) = \frac{\widetilde{\mathrm{P}}_{st}^{\mathrm{ref}}(\wp) \exp\left(-\beta \widetilde{c}(\wp)\right)}{\sum_{\wp \in \mathcal{P}_{st}} \widetilde{\mathrm{P}}_{st}^{\mathrm{ref}}(\wp) \exp\left(-\beta \widetilde{c}(\wp)\right)}, \tag{3}$$

where the inverse temperature parameter β controls the influence of the cost on the walker's selection of a path. When applying the model, the user is assumed to provide β as an input parameter instead of the relative entropy J_0.

After deriving the optimal distribution for minimizing the expected cost, the RSP dissimilarity between the nodes s and t is defined as this expected cost after symmetrization, formally

$$\Delta_{st}^{\mathrm{RSP}} = \left(\overline{c}(\widetilde{\mathrm{P}}_{st}^{\mathrm{RSP}}) + \overline{c}(\widetilde{\mathrm{P}}_{ts}^{\mathrm{RSP}}) \right) / 2. \tag{4}$$

The authors in [14] develop an algorithm for computing the expected cost $\overline{c}(\widetilde{\mathrm{P}}_{st}^{\mathrm{RSP}})$, which is not at all a trivial task. In the next section we develop a new, more efficient algorithm for computing the expected costs and thus the matrix $\mathbf{\Delta}^{\mathrm{RSP}}$ of the RSP dissimilarities between all pairs of nodes of a graph in closed form. After that, in Section 4.3, we will consider a distance based on the minimized *Helmholtz free energy* [26], instead of minimized expected cost. The free energy of a thermodynamical system with temperature $T = 1/\beta$ and state transition probabilities $\widetilde{\mathrm{P}}_{st}$ has the form[5]

$$\phi(\widetilde{\mathrm{P}}_{st}) = \overline{c}(\widetilde{\mathrm{P}}_{st}) + J(\widetilde{\mathrm{P}}_{st} \| \widetilde{\mathrm{P}}_{st}^{\mathrm{ref}}) / \beta. \tag{5}$$

4.2. Algorithm for faster computation of $\mathbf{\Delta}^{\mathrm{RSP}}$

We now show how to compute the RSP dissimilarity and then 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 [14] 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.

[5] Conventionally, the Helmholtz free energy is defined with the entropy of $\widetilde{\mathrm{P}}_{st}$ in place of the relative entropy. Regardless of this, we use the term as we have presented it.

The computation of the expected cost $\overline{c}(\widetilde{\mathrm{P}}_{st}^{\mathrm{RSP}})$ is based on considering the denominator of the right side of Equation (3) and denoting

$$z_{st}^{\mathrm{h}} = \sum_{\wp \in \mathcal{P}_{st}} \widetilde{\mathrm{P}}_{st}^{\mathrm{ref}}(\wp) \exp(-\beta \widetilde{c}(\wp)). \tag{6}$$

This quantity is in statistical physics called the *partition function* of a thermodynamical system. In our case, the system consists of the hitting (hence the superscript "h" in z_{st}^{h}) paths of $\mathcal{P}_{st}$. The partition function is essential for deriving different quantities related to the RSP framework. Indeed, by manipulating the expected cost of travelling from node s to node t we see that

$$\begin{aligned} \overline{c}(\widetilde{\mathrm{P}}_{st}^{\mathrm{RSP}}) &= \sum_{\wp \in \mathcal{P}_{st}} \widetilde{\mathrm{P}}_{st}^{\mathrm{RSP}}(\wp)\widetilde{c}(\wp) = \frac{\sum_{\wp \in \mathcal{P}_{st}} \widetilde{\mathrm{P}}_{st}^{\mathrm{ref}}(\wp) \exp(-\beta \widetilde{c}(\wp))\widetilde{c}(\wp)}{z_{st}^{\mathrm{h}}} \\ &= -\frac{1}{z_{st}^{\mathrm{h}}} \frac{\partial z_{st}^{\mathrm{h}}}{\partial \beta} = -\frac{\partial \log z_{st}^{\mathrm{h}}}{\partial \beta} \end{aligned} \tag{7}$$

meaning that the expected cost can be obtained by taking the derivative of the logarithm of the partition function.

Let us denote by $\mathbf{C}$ the matrix of costs on edges, c_{ij}, and by $\mathbf{P}^{\mathrm{ref}}$ the transition probability matrix of the natural random walk associated to the graph G containing the elements p_{ij}^{ref}. The latter can be computed from the adjacency matrix as $\mathbf{P}^{\mathrm{ref}} = \mathbf{D}^{-1}\mathbf{A}$, where $\mathbf{D} = \mathbf{Diag}(\mathbf{Ae})$. In order to compute the partition function, we define a new matrix

$$\mathbf{W} = \mathbf{P}^{\mathrm{ref}} \circ \exp(-\beta \mathbf{C}). \tag{8}$$

This matrix is substochastic and thus it can be interpreted as a new transition matrix defining a *killed*, or an *evaporating* random walk on the graph [27]. 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 [28], where the walker can end up from each node of the graph with a non-zero probability.

Remember now that we only consider hitting paths, meaning that we want to set the destination node t absorbing. For this, we define a new matrix by setting the row t of $\mathbf{W}$ to zero: $\mathbf{W}_t^{\mathrm{h}} = \mathbf{W} - \mathbf{e}_t(\mathbf{w}_t^{\mathrm{r}})^{\mathsf{T}}$, where $\mathbf{e}_t$ is a vector containing 1 in element t and 0 elsewhere and $\mathbf{w}_t^{\mathrm{r}}$ is row t (hence the superscript "r") of matrix $\mathbf{W}$ as a column vector, i.e. $\mathbf{w}_t^{\mathrm{r}} = (\mathbf{e}_t^{\mathsf{T}}\mathbf{W})^{\mathsf{T}}$. Thus, expressed elementwise, $(\mathbf{W}_t^{\mathrm{h}})_{ij} = (\mathbf{W})_{ij}$, if $i \neq t$, and 0 otherwise.

The powers of this matrix, $(\mathbf{W}_t^{\mathrm{h}})^{\tau}$, contain in element (s,t) the probability that a killed random walk of exactly τ steps leaving from node s ends up in node t when obeying the transition probabilities assigned by $\mathbf{W}_t^{\mathrm{h}}$. This can also be expressed as

$$\left[(\mathbf{W}_t^{\mathrm{h}})^{\tau}\right]_{st} = \sum_{\wp \in \mathcal{P}_{st}(\tau)} \widetilde{\mathrm{P}}_{st}^{\mathrm{ref}}(\wp) \exp(-\beta \widetilde{c}(\wp)),$$

where $\mathcal{P}_{st}(\tau)$ denotes the set of paths whose length is τ going from node s to node t. Then by summing over all walk lengths[6] τ we can cover all hitting paths from node s to t and write the partition function as a power series

$$\begin{aligned} z_{st}^{\mathrm{h}} &= \sum_{\wp \in \mathcal{P}_{st}} \widetilde{\mathrm{P}}_{st}^{\mathrm{ref}}(\wp) \exp(-\beta \widetilde{c}(\wp)) = \sum_{\tau=0}^{\infty} \sum_{\wp \in \mathcal{P}_{st}(\tau)} \widetilde{\mathrm{P}}_{st}^{\mathrm{ref}}(\wp) \exp(-\beta \widetilde{c}(\wp)) \\ &= \sum_{\tau=0}^{\infty} \left[(\mathbf{W}_t^{\mathrm{h}})^{\tau} \right]_{st} = \left[(\mathbf{I} - \mathbf{W}_t^{\mathrm{h}})^{-1} \right]_{st}, \end{aligned}$$

The series converges to the matrix $\mathbf{Z}_t^{\mathrm{h}} = (\mathbf{I} - \mathbf{W}_t^{\mathrm{h}})^{-1}$, as the spectral radius of $\mathbf{W}_t^{\mathrm{h}}$ is less than one, $\rho(\mathbf{W}_t^{\mathrm{h}}) < 1$. The matrix $\mathbf{Z}_t^{\mathrm{h}}$ is the *fundamental matrix* corresponding to the killed Markov Chain with the transition matrix $\mathbf{W}_t^{\mathrm{h}}$. In the original reference [14], the authors then derive a way of computing the matrix $\mathbf{Z}_t^{\mathrm{h}}$ just by using the simpler matrix

$$\mathbf{Z} = (\mathbf{I} - \mathbf{W})^{-1}. \tag{9}$$

This is the fundamental matrix of the killed, but *non-absorbing* Markov Chain with the transition matrix $\mathbf{W}$, and its computation only requires a simple matrix inversion.

With this in mind, the authors in [14] use the Sherman-Morrison rule for a rank-one update of a matrix for deriving the form (Equation (20) in [14])

$$\mathbf{Z}_t^{\mathrm{h}} = \mathbf{Z} - \frac{\mathbf{Z}\mathbf{e}_t (\mathbf{w}_t^{\mathrm{r}})^{\mathsf{T}} \mathbf{Z}}{1 + (\mathbf{w}_t^{\mathrm{r}})^{\mathsf{T}} \mathbf{Z}\mathbf{e}_t}. \tag{10}$$

They then use this form of $\mathbf{Z}_t^{\mathrm{h}}$ 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, note that we are only interested in computing the element (s, t) of the matrix $\mathbf{Z}_t^{\mathrm{h}}$, which we can do by using (10):

$$\begin{aligned} z_{st}^{\mathrm{h}} &= \mathbf{e}_s^{\mathsf{T}} \mathbf{Z}_t^{\mathrm{h}} \mathbf{e}_t = z_{st} - \frac{z_{st} (\mathbf{w}_t^{\mathrm{r}})^{\mathsf{T}} \mathbf{Z}\mathbf{e}_t}{1 + (\mathbf{w}_t^{\mathrm{r}})^{\mathsf{T}} \mathbf{Z}\mathbf{e}_t} = z_{st} \left(1 - \frac{1 + (\mathbf{w}_t^{\mathrm{r}})^{\mathsf{T}} \mathbf{Z}\mathbf{e}_t - 1}{1 + (\mathbf{w}_t^{\mathrm{r}})^{\mathsf{T}} \mathbf{Z}\mathbf{e}_t} \right) \\ &= \frac{z_{st}}{\mathbf{e}_t^{\mathsf{T}} \mathbf{e}_t + \mathbf{e}_t^{\mathsf{T}} \mathbf{W}\mathbf{Z}\mathbf{e}_t} = \frac{z_{st}}{\mathbf{e}_t^{\mathsf{T}} (\mathbf{I} + \mathbf{W}\mathbf{Z}) \mathbf{e}_t} = \frac{z_{st}}{\mathbf{e}_t^{\mathsf{T}} \mathbf{Z}\mathbf{e}_t} = \frac{z_{st}}{z_{tt}}, \end{aligned} \tag{11}$$

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 z_{st}^{h} of hitting paths from node s to node t without having to compute $\mathbf{Z}_t^{\mathrm{h}}$ for each t separately. This can be done simply using the matrix $\mathbf{Z}$ as

$$\mathbf{Z}^{\mathrm{h}} = \mathbf{Z}\mathbf{D}_{\mathrm{h}}^{-1}, \tag{12}$$

[6] Although in [14] only paths of length ≥ 1 are considered, we also include paths of length 0, i.e. the paths that consist of only one node and no links, into the set of allowed paths; see [16] for a discussion related to this.

where $\mathbf{D}_\mathrm{h} = \mathbf{Diag}(\mathbf{Z})$ is the diagonal matrix with elements z_{ii} on its diagonal. The advantage compared to the original algorithm presented in [14] is that $\mathbf{Z}^\mathrm{h}$ can be used for computing the RSP dissimilarities between all pairs of nodes instead of having to compute the matrix $\mathbf{Z}_t^\mathrm{h}$ separately for each destination t. This reduces the computational complexity of the algorithm and makes it much more straightforward to implement than before.

Note that a similar derivation for computing z_{st}^h as the above one is also presented in Appendix A of [16]. It is also showed in the same reference 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.

Now we can finally derive the new matrix formula for the RSP dissimilarities between all pairs of nodes using Equations (7) and (12). The expected cost is given by

$$\overline{c}(\widetilde{\mathrm{P}}_{st}^{\mathrm{RSP}}) = -\frac{\partial \log z_{st}^\mathrm{h}}{\partial\beta} = -\frac{\partial \log(z_{st}/z_{tt})}{\partial\beta} = -\frac{\partial \log z_{st}}{\partial\beta} + \frac{\partial \log z_{tt}}{\partial\beta}. \tag{13}$$

The first term can be computed by

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

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

Thus, we can write Equation (13) as

$$\overline{c}(\widetilde{\mathrm{P}}_{st}^{\mathrm{RSP}}) = \frac{\mathbf{e}_s^\mathsf{T}\mathbf{Z}\left(\mathbf{C}\circ\mathbf{W}\right)\mathbf{Z}\mathbf{e}_t}{z_{st}} - \frac{\mathbf{e}_t^\mathsf{T}\mathbf{Z}\left(\mathbf{C}\circ\mathbf{W}\right)\mathbf{Z}\mathbf{e}_t}{z_{tt}} \tag{14}$$

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 (14), and contains the expected costs of non-hitting random walks. We can now use it to write out the matrix form of computing all the expected costs of hitting walks 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 $\mathbf{\Delta}^{\mathrm{RSP}}$ is defined by symmetrizing $\overline{\mathbf{C}}$: $\mathbf{\Delta}^{\mathrm{RSP}} = (\overline{\mathbf{C}}+\overline{\mathbf{C}}^\mathsf{T})/2$.

4.3. 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 recently in Equation (38) of [16] 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 [29], the authors define a ranking method called the free-energy rank (in the spirit of the well-known PageRank [30]) 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 [28], 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. Recall that the RSP dissimilarity was defined by considering a distribution of random walks between two nodes that minimizes the expected cost $\overline{c}(\widetilde{\mathrm{P}}_{st})$ subject to a relative entropy constraint. 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 that minimizes the free energy according to Equation (5). The minimization of free energy can be simply written as

$$\begin{aligned}\widetilde{\mathrm{P}}_{st}^{\mathrm{FE}} = \underset{\widetilde{\mathrm{P}}_{st}}{\arg\min} \sum_{\wp\in\mathcal{P}_{st}} \widetilde{\mathrm{P}}_{st}(\wp)\widetilde{c}(\wp) + \frac{1}{\beta}\sum_{\wp\in\mathcal{P}_{st}} \widetilde{\mathrm{P}}_{st}(\wp)\log(\widetilde{\mathrm{P}}_{st}(\wp)/\widetilde{\mathrm{P}}_{st}^{\mathrm{ref}}(\wp)),\\ \text{subject to } \sum_{\wp\in\mathcal{P}_{st}} \widetilde{\mathrm{P}}_{st}(\wp) = 1.\end{aligned}$$

It is not difficult to see that this problem becomes equivalent to the minimization problem (2) involved in the definition of the RSP probabilities and thus the optimal solution is again the Boltzmann distribution (3), in other words, $\widetilde{\mathrm{P}}_{st}^{\mathrm{FE}} = \widetilde{\mathrm{P}}_{st}^{\mathrm{RSP}}$. We define the free energy distance between nodes s and t as the symmetrized minimum free energy between these two nodes, in other words

$$\Delta_{st}^{\mathrm{FE}} = \left(\phi(\widetilde{\mathrm{P}}_{st}^{\mathrm{FE}}) + \phi(\widetilde{\mathrm{P}}_{ts}^{\mathrm{FE}})\right)/2 \tag{15}$$

In order to show that the FE distance coincides with the potential distance defined in Equation (38) of [16], we remind that the FE probability (which is

equal to the RSP probability) of a path $\wp$ can be written as (see Equations (3) and (6))

$$\widetilde{\mathrm{P}}_{st}^{\mathrm{FE}}(\wp) = \frac{\widetilde{\mathrm{P}}_{st}^{\mathrm{ref}}(\wp)\exp(-\beta\widetilde{c}(\wp))}{z_{st}^{\mathrm{h}}}.$$

Using then the fact that $\sum_{\wp\in\mathcal{P}_{st}}\widetilde{\mathrm{P}}_{st}^{\mathrm{FE}}(\wp) = 1$, we can write out the expression for the relative entropy:

$$\begin{aligned}
J(\widetilde{\mathrm{P}}_{st}^{\mathrm{FE}}\|\widetilde{\mathrm{P}}_{st}^{\mathrm{ref}}) &= \sum_{\wp\in\mathcal{P}_{st}}\widetilde{\mathrm{P}}_{st}^{\mathrm{FE}}(\wp)\log\left(\widetilde{\mathrm{P}}_{st}^{\mathrm{FE}}(\wp)/\widetilde{\mathrm{P}}_{st}^{\mathrm{ref}}(\wp)\right)\\
&= \sum_{\wp\in\mathcal{P}_{st}}\widetilde{\mathrm{P}}_{st}^{\mathrm{FE}}(\wp)\log\left(\frac{\widetilde{\mathrm{P}}_{st}^{\mathrm{ref}}(\wp)\exp(-\beta\widetilde{c}(\wp))}{z_{st}^{\mathrm{h}}}\right) - \sum_{\wp\in\mathcal{P}_{st}}\widetilde{\mathrm{P}}_{st}^{\mathrm{FE}}(\wp)\log\left(\widetilde{\mathrm{P}}_{st}^{\mathrm{ref}}(\wp)\right)\\
&= \sum_{\wp\in\mathcal{P}_{st}}\widetilde{\mathrm{P}}_{st}^{\mathrm{FE}}(\wp)\log\left(\widetilde{\mathrm{P}}_{st}^{\mathrm{ref}}(\wp)\right) - \beta\sum_{\wp\in\mathcal{P}_{st}}\widetilde{\mathrm{P}}_{st}^{\mathrm{FE}}(\wp)\widetilde{c}(\wp)\\
&\quad - \log(z_{st}^{\mathrm{h}}) - \sum_{\wp\in\mathcal{P}_{st}}\widetilde{\mathrm{P}}_{st}^{\mathrm{FE}}(\wp)\log\left(\widetilde{\mathrm{P}}_{st}^{\mathrm{ref}}(\wp)\right)\\
&= -\beta\overline{c}(\widetilde{\mathrm{P}}_{st}^{\mathrm{FE}}) - \log(z_{st}^{\mathrm{h}})
\end{aligned}$$

When combining this result with Equation (5), the FE becomes

$$\phi(\widetilde{\mathrm{P}}_{st}^{\mathrm{FE}}) = -\frac{1}{\beta}\log(z_{st}^{\mathrm{h}})$$

which after the symmetrization (15) equals the potential distance defined in Equation (38) of [16]. Thanks to the derivation of the new algorithm for the RSP dissimilarities in Section 4.2, the matrix of free energies between all pairs of nodes

$$\mathbf{\Phi} = -1/\beta\log\mathbf{Z}^{\mathrm{h}}$$

can also be computed straightforwardly by performing Equations (8), (9) and (12). Finally, the matrix of all FE distances is obtained, again, by symmetrization as $\mathbf{\Delta}^{\mathrm{FE}} = (\mathbf{\Phi} + \mathbf{\Phi}^{\mathsf{T}})/2$.

Thus, we have shown that the potential distance derived within the bag-of-paths framework in [16], 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 [16]. Most importantly, it was shown that the distance obeys the triangle inequality (in Inequality (37) of [16]) and is thus a metric, as opposed to the RSP dissimilarity. The distance also converges to the

SP distance when $\beta \to \infty$ and to the CT distance[7] when $\beta \to 0^+$ (see Appendices D and E in [16]). In addition, in Appendix C of [16], it is shown to be *graph geodetic* [17, 31], meaning that $\Delta_{st}^{\mathrm{FE}} = \Delta_{sk}^{\mathrm{FE}} + \Delta_{kt}^{\mathrm{FE}}$ 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 distance measure between graph nodes with nice properties. Interestingly, as can be seen from Equation (5), 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 $\log z_{st}^{\mathrm{h}}$ already appeared in [24] as a potential inducing a drift for a random walker in a continuous-state extension of the RSP framework.

5. 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 [17, 32, 33, 34]. In this paper, we focus on the logarithmic forest distances [17]. 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 [35]. Indeed they show that with certain values of the parameter p, the p-resistance distance avoids this pitfall.

5.1. The SP-CT combination distance

The graph distance families presented and reviewed in this paper 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:

$$\Delta_{st}^{\mathrm{SP-CT}} = \lambda \Delta_{st}^{\mathrm{SP}} + (1-\lambda)\Delta_{st}^{\mathrm{CT}}, \tag{16}$$

where $\lambda \in [0, 1]$.[8] 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.

[7] To the CT distance divided by 2, to be precise.

[8] 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.

5.2. Logarithmic forest distances

The logarithmic forest distance [17] has its foundation in the matrix-forest theorem and another family of distances developed earlier by Chebotarev called simply the forest distance [32, 33].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* [32] 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 [17] 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

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

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

$$\boldsymbol{\Delta}^{\text{logFor}} = \frac{1}{2}(\mathbf{m}\mathbf{e}^{\mathsf{T}} + \mathbf{e}\mathbf{m}^{\mathsf{T}}) - \mathbf{M}, \tag{17}$$

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 [36].

The definition provides a metric which also satisfies the graph-geodetic property (see Section 4.3). 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[9]. 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.

5.3. p-resistance distance

Alamgir and von Luxburg [35] 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. This forms

[9]More accurately, the logarithmic forest distance converges to the *unweighted* SP distance (see Section 3) but we nevertheless include it in our comparison.

a *unit flow from s to t*, $(i_{kl})_{s\to t}$, where the currents i_{kl} are assigned on all the edges $(k,l) \in E$ of the graph. In short, this means that for all k, l such that $(k,l) \in E$ the currents i_{kl} satisfy the following three conditions: (1) $i_{kl} = -i_{lk}$, (2) $\sum_l i_{sl} = 1$ and $\sum_k i_{kt} = -1$ and (3) $\sum_l i_{kl} = 0$ for $s \neq k \neq t$. 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

$$\Delta_{st}^{p\mathrm{Res}} = \min_{(i_{kl})_{s\to t}} \left\{ \sum_{(k,l)\in E} r_{kl}|i_{kl}|^p \;\middle|\; (i_{kl})_{s\to t} \text{ is a unit flow from } s \text{ to } t \right\}. \quad (18)$$

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 [35] show that there exists a value $1 < p < 2$ for which the p-resistance distance avoids the problem of the traditional resistance distance with large graphs, introduced in Section 3.

Another, almost identical form of the p-resistance distance has been proposed by Herbster in [37]. Also, in a closely related work [38], 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 6.1-6.2, but not the others.

5.4. Other graph node distances

In the comparisons in Section 6, 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 [16], 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 [16] to provide good results in clustering and semisupervised learning.

In [22], 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 5 to the work of Chebotarev, we want to highlight the *walk distance* [34], which has some interesting, quite unintuitive properties, especially when considering the peripheral nodes of a network. Finally, we mention the *communicability distance* [39, 40], 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 [41] 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 [27].

6. Experiments

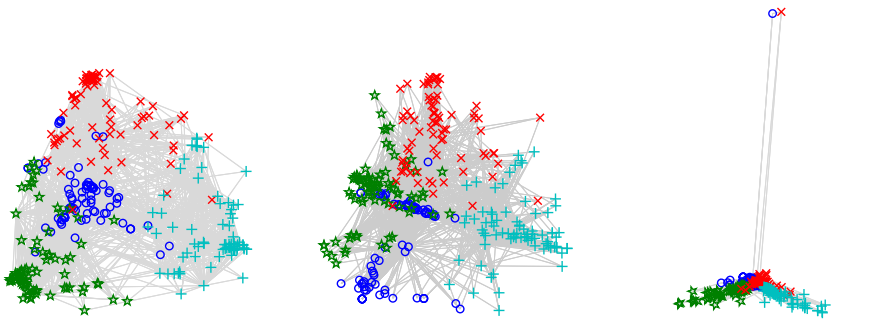
In this Section, we compare the different distance families presented in the previous Sections, namely the RSP dissimilarity, the FE distance, the p-resistance distance, the logarithmic forest distance and the SP-CT combination distance. First, we consider small artificial graphs and study the behavior 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 [34], 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.

6.1. Visualization

In order to show that the different distance families are appealing to use, we show a graph visualization example using an artificially generated graph. The graph is generated with the benchmark algorithm of Lancichinetti, Fortunato and Radicchi [42], which we refer to as the LFR algorithm. The algorithm can generate 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 12, the maximum degree to 160 and the number of overlapping nodes, belonging to two communities, to 5.

The visualization is achieved with classical multidimensional scaling (CMDS) [36] using the set of distances between all pairs of nodes provided by the different distance families. We show the 2-dimensional representation according to the coordinates obtained with CMDS using the SP distance, the CT distance, the

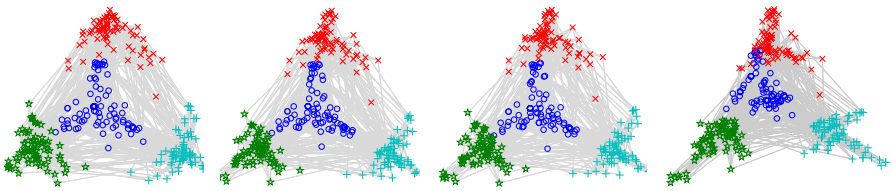
RSP dissimilarity, the FE distance and the parametrized distances introduced in Sections 5.1-5.3. We also plot the edges of the graph with grey lines. For all the parametrized distances, we selected one intermediate parameter value, just in order to differentiate the plots from the plots obtained with the SP and CT distances. The visualizations are presented in Figure 1 and the particular parameter values are given in the captions.



(a) SP distance

(b) SP-CT combination distance, $\lambda = 0.1$

(c) CT distance



(d) RSP dissimilarity, $\beta = 0.05$

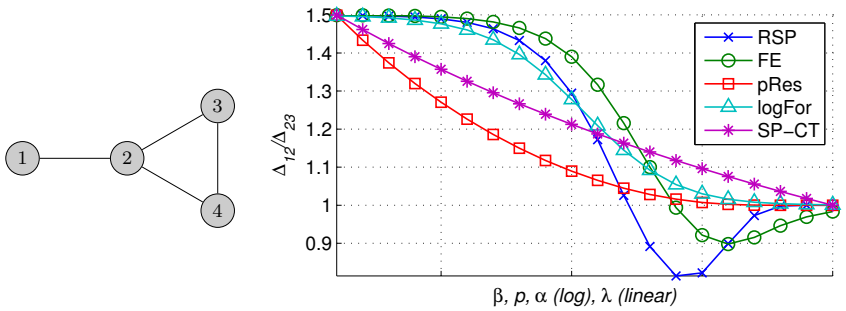
(e) FE distance, $\beta = 0.05$

(f) Logarithmic forest distance, $\alpha = 0.8$

(g) p-resistance distance, $p = 1.5$

Figure 1: Visualizations with CMDS of a graph generated with the LFR algorithm consisting of four communities represented by red crosses, blue circles, green pentagrams and cyan plus-signs.

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 3. In fact, these nodes have, among all the nodes, the two largest sums of CT



(a) (b)

Figure 2: The extended triangle graph (a) and the ratio of distances Δ_{12}/Δ_{23} (b) 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 $\lambda \in [0, 1]$.

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.

6.2. Comparisons with small graphs

In the first example, we use the simple graph depicted in Figure 2(a) 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. We plot the results in Figure 2(b) 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 $\Delta_{12}^{\mathrm{CT}} > \Delta_{23}^{\mathrm{CT}}$ 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.

The real interest in Figure 2(b) 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

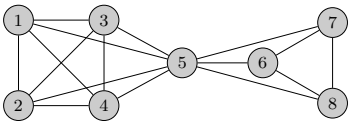


Figure 3: A graph with a 4-clique and a 3-clique and a hub node between them.

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. 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 2(b), 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 gen-

eralized 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 6.3 and 6.4, 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 [4, 8].

6.3. 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 [43]. 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* [44]. 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 [36, 45, 46] as

$$\mathbf{K} = -\frac{1}{2}\mathbf{H}\mathbf{\Delta}\mathbf{H}, \tag{19}$$

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 [47]. 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 (19), we also use the *sigmoid commute time kernel*

proposed by Yen et al. [43]. 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 $(\mathbf{K}^{\sigma\mathrm{CT}})_{st} = 1/(1 + \exp(-a\, l^+_{st}/\sigma))$. 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 paper. 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[10]. A more detailed description of the collection that we use can be found in [43]. 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 5.3 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) [48] 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.

The ranges of parameter values and the optimal values for each method are reported in Table 1. Note that with the SP-CT combination distance, the

[10] `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	$\mathbf{K}_{\mathrm{LogF}}$	$[10^{-2}, 500]$	$\alpha = 0.95$
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 1: The notation of the similarity matrices and the optimal parameter values obtained on the tuning data set.

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.

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 2. 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.

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

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

Table 2: Clustering performances (Normalized Mutual Information, multiplied by 100) for each similarity matrix on the nine Newsgroup subsets

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.

6.4. 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 [49], 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 6.3 but also for another collection of networks, the WebKB data set [50]. 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 6.3, 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 4 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.

We perform a ranking of the distance families using Copeland's voting method [51], 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. 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

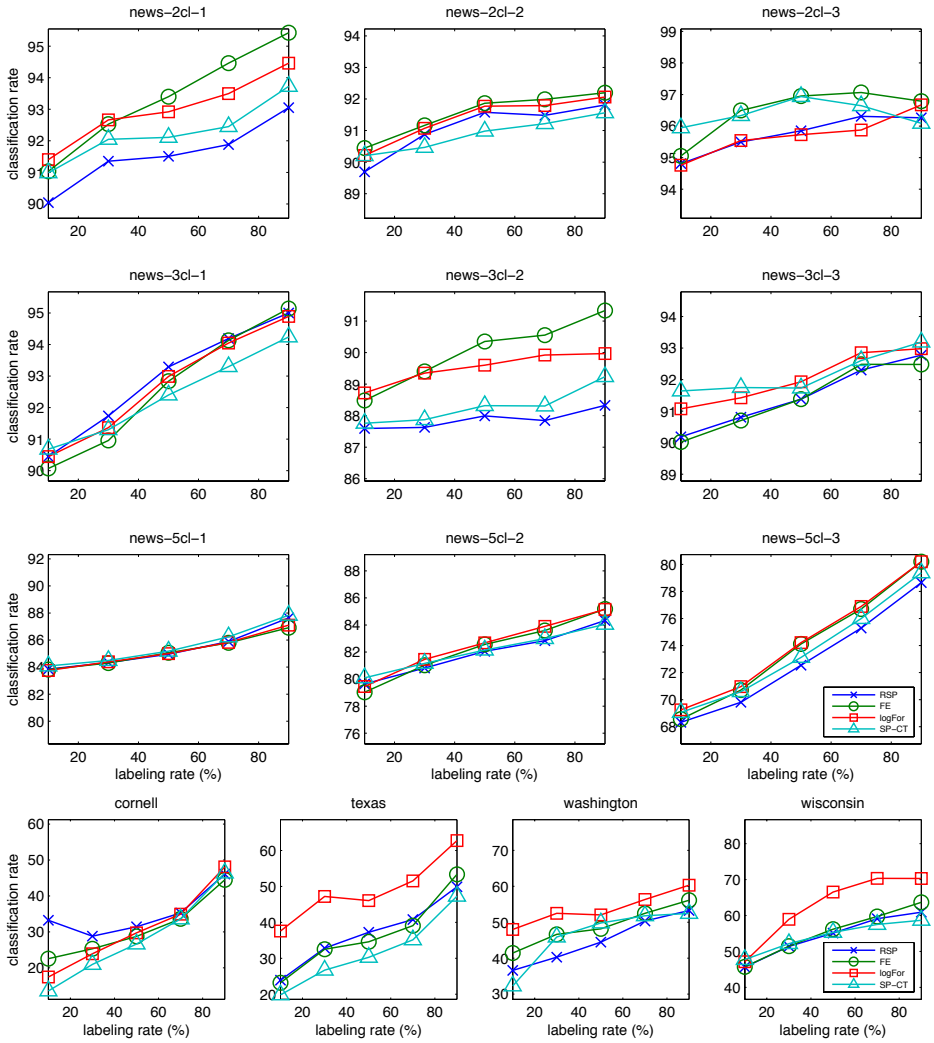


Figure 4: The results of the propagating 1-NN algorithm with the nine Newsgroups graphs (top) and the four WebKB co-citation graphs (below)

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.

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
$\mathbf{K}_{\mathrm{LogF}}$	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: 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).

7. Conclusion

In this article, we concentrated on graph node distances that generalize the SP and CT distances. We first presented the setting of the paper 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 the article 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 paper 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. Also, we plan to extend even further the RSP framework for defining different graph node betweenness and robustness measures as well as temporal versions of the distances.

Appendix A. The CC distance is proportional to the CT distance

For deriving this result, we refer to earlier literature. First, we call to mind a well-known result [13] that the commute time distance can be computed in terms of the pseudo-inverse of the graph Laplacian as

$$\Delta_{st}^{\mathrm{CT}} = (l_{ss}^{+} + l_{tt}^{+} - 2l_{st}^{+}) \sum_{i,j=1}^{n} a_{ij}. \tag{A.1}$$

In addition, the authors in [52] derive a formula for computing the *average first passage cost*, o_{st}, i.e. the expected cost of random walks from a node another. The formula (see [52], Appendix B, Equation (18)) is given as $o_{st} = \sum_{i=1}^{n}(l_{si}^{+} - l_{st}^{+} - l_{ti}^{+} + l_{tt}^{+}) \sum_{j=1}^{n} a_{ij}c_{ij}$. From this we can obtain the commute cost distance $\Delta_{st}^{\mathrm{CC}}$ by symmetrization:

$$\begin{aligned}\Delta_{st}^{\mathrm{CC}} = o_{st} + o_{ts} &= \sum_{i=1}^{n}(l_{si}^{+} - l_{st}^{+} - l_{ti}^{+} + l_{tt}^{+} + l_{ti}^{+} - l_{ts}^{+} - l_{si}^{+} + l_{ss}^{+}) \sum_{j=1}^{n} a_{ij}c_{ij} \\ &= (l_{ss}^{+} + l_{tt}^{+} - 2l_{st}^{+}) \sum_{i,j=1}^{n} a_{ij}c_{ij},\end{aligned}$$

which holds because the graph is assumed undirected. Comparing this result with Equation (A.1) we see that the distances only differ from each other by a multiplying factor. Moreover, we see that this factor is

$$\frac{\Delta_{st}^{\mathrm{CC}}}{\Delta_{st}^{\mathrm{CT}}} = \frac{\sum_{i,j=1}^{n} a_{ij}c_{ij}}{\sum_{i,j=1}^{n} a_{ij}} = \frac{\mathbf{e}^{\mathsf{T}}(\mathbf{A} \circ \mathbf{C})\mathbf{e}}{\mathbf{e}^{\mathsf{T}}\mathbf{A}\mathbf{e}}.$$

[1] S. Wasserman, K. Faust, Social Network Analysis: Methods and Applications, Cambridge University Press, 1994.

[2] M. Thelwall, Link Analysis: An Information Science Approach, Elsevier, 2004.

[3] F. Chung, L. Lu, Complex Graphs and Networks, American Mathematical Society, 2006.

[4] D. Liben-Nowell, J. Kleinberg, The link-prediction problem for social networks, Journal of the American Society for Information Science and Technology 58 (7) (2007) 1019–1031.

[5] E. D. Kolaczyk, Statistical Analysis of Network Data, Springer, 2009.

[6] T. G. Lewis, Network Science : Theory and Applications, Wiley, 2009.

[7] M. E. Newman, Networks : An Introduction, Oxford University Press, 2010.

[8] L. Lü, T. Zhou, Link prediction in complex networks: A survey, Physica A: Statistical Mechanics and its Applications 390 (6) (2011) 1150 – 1170.

[9] E. Estrada, The structure of complex networks: theory and applications, Oxford University Press, 2012.

[10] F. Gobel, A. A. Jagers, Random walks on graphs, Stochastic Processes and their Applications 2 (1974) 311–336.

[11] D. J. Klein, M. Randic, Resistance distance, Journal of Mathematical Chemistry 12 (1) (1993) 81–95.

[12] P. G. Doyle, J. L. Snell, Random Walks and Electric Networks, The Mathematical Association of America, 1984.

[13] A. K. Chandra, P. Raghavan, W. L. Ruzzo, R. Smolensky, P. Tiwari, The electrical resistance of a graph captures its commute and cover times, Annual ACM Symposium on Theory of Computing (1989) 574–586.

[14] L. Yen, A. Mantrach, M. Shimbo, 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.

[15] M. Saerens, Y. Achbany, F. Fouss, L. Yen, Randomized shortest-path problems: Two related models, Neural Computation 21 (8) (2009) 2363–2404.

[16] K. Françoisse, I. Kivimäki, A. Mantrach, F. Rossi, M. Saerens, A bag-of-paths framework for network data analysis, arXiv preprint arXiv:1302.6766.

[17] P. Chebotarev, A class of graph-geodetic distances generalizing the shortest-path and the resistance distances, Discrete Applied Mathematics 159 (5) (2011) 295–302.

[18] J. B. Tenenbaum, V. d. Silva, J. C. Langford, A Global Geometric Framework for Nonlinear Dimensionality Reduction, Science 290 (5500) (2000) 2319–2323.

[19] H. M. Taylor, S. Karlin, An Introduction to Stochastic Modeling, 3rd Edition, Academic Press, 1998.

[20] M. Brand, A random walks perspective on maximizing satisfaction and profit, Proceedings of the 2005 SIAM International Conference on Data Mining.

[21] A. Radl, U. von Luxburg, M. Hein, The resistance distance is meaningless for large random geometric graphs, in: Proc. Workshop on Analyzing Networks and Learning with Graphs, 2009.

[22] U. von Luxburg, A. Radl, M. Hein, Getting lost in space: large sample analysis of the commute distance, Proceedings of the 23th Neural Information Processing Systems conference (NIPS 2010) (2010) 2622–2630.

[23] T. Akamatsu, Cyclic flows, markov process and stochastic traffic assignment, Transportation Research B 30 (5) (1996) 369–386.

[24] S. Garcia-Diez, E. Vandenbussche, 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, 2011, pp. 6570 –6577.

[25] C. Elkan, Using the triangle inequality to accelerate k-means, in: T. Fawcett, N. Mishra (Eds.), Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003), August 21-24, 2003, Washington, DC, USA, AAAI Press, 2003, pp. 147–153.

[26] L. Peliti, Statistical Mechanics in a Nutshell, In a Nutshell, Princeton University Press, 2011.

[27] F. Fouss, K. Françoisse, L. Yen, A. Pirotte, M. Saerens, An experimental investigation of kernels on graphs for collaborative recommendation and semisupervised classification, Neural Networks 31 (2012) 53–72.

[28] F. Bavaud, G. Guex, Interpolating between random walks and shortest paths: a path functional approach, in: K. A. et al. (Ed.), SocInfo 2012, Vol. 7710 of Lecture Notes in Computer Science, Springer, 2012, pp. 68–81.

[29] J.-C. Delvenne, A.-S. Libert, Centrality measures and thermodynamic formalism for complex networks, Physical Review E 83 (2011) 046117.

[30] L. Page, S. Brin, R. Motwani, T. Winograd, The pagerank citation ranking: Bringing order to the web, Technical Report 1999-0120, Computer Science Department, Stanford University.

[31] D. Klein, H.-Y. Zhu, Distances and volumina for graphs, Journal of mathematical chemistry 23 (1) (1998) 179–195.

[32] P. Chebotarev, E. Shamis, The matrix-forest theorem and measuring relations in small social groups, Automation and Remote Control 58 (9) (1997) 1505–1514.

[33] P. Chebotarev, E. Shamis, The forest metrics for graph vertices, Electronic Notes in Discrete Mathematics 11 (2002) 98–107.

[34] P. Chebotarev, The walk distances in graphs, Discrete Applied Mathematics 160 (10-11) (2012) 1484–1500.

[35] M. Alamgir, U. von Luxburg, Phase transition in the familiy of p-resistances, in: Neural Information Processing Systems (NIPS), 2011.

[36] I. Borg, P. Groenen, Modern multidimensional scaling: Theory and applications, Springer, 1997.

[37] M. Herbster, A triangle inequality for p-resistance, in: NIPS Workshop 2010: Networks Across Disciplines: Theory and Applications, 2010.

[38] Y. Li, Z.-L. Zhang, 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, IEEE Computer Society, Washington, DC, USA, 2011, pp. 847–856.

[39] E. Estrada, The communicability distance in graphs, Linear Algebra and its Applications 436 (11) (2012) 4317–4328.

[40] E. Estrada, Complex networks in the euclidean space of communicability distances, Physical Review E 85 (6) (2012) 066122.

[41] F. Bavaud, Euclidean distances, soft and spectral clustering on weighted graphs, in: Machine Learning and Knowledge Discovery in Databases, Springer, 2010, pp. 103–118.

[42] A. Lancichinetti, S. Fortunato, Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities, Phys. Rev. E 80 (2009) 016118.

[43] L. Yen, F. Fouss, C. Decaestecker, P. Francq, M. Saerens, Graph nodes clustering with the sigmoid commute-time kernel: A comparative study, Data and Knowledge Engineering 68 (3) (2009) 338 – 361.

[44] J. Shawe-Taylor, N. Cristianini, Kernel methods for pattern analysis, Cambridge University Press, 2004.

[45] I. J. Schoenberg, Metric spaces and positive definite functions, Transactions of the American Mathematical Society 44 (3) (1938) 522–536.

[46] J. C. Gower, P. Legendre, Metric and euclidean properties of dissimilarity coefficients, Journal of classification 3 (1) (1986) 5–48.

[47] M. Saerens, F. Fouss, L. Yen, P. Dupont, The principal components analysis of a graph, and its relationships to spectral clustering, Proceedings of the 15th European Conference on Machine Learning (ECML 2004). Lecture Notes in Artificial Intelligence, vol. 3201, Springer-Verlag, Berlin (2004) 371–383.

[48] A. Strehl, J. Ghosh, C. Cardie, Cluster ensembles - a knowledge reuse framework for combining multiple partitions, Journal of Machine Learning Research 3 (2002) 583–617.

[49] X. Zhu, A. Goldberg, Introduction to Semi-Supervised Learning, Synthesis Lectures on Artificial Intelligence and Machine Learning Series, Morgan & Claypool, 2009.

[50] S. A. Macskassy, F. Provost, Classification in networked data: A toolkit and a univariate case study, J. Mach. Learn. Res. 8 (2007) 935–983.

[51] D. G. Saari, Explaining all three-alternative voting outcomes, Journal of Economic Theory 87 (2) (1999) 313–355.

[52] F. Fouss, A. Pirotte, J.-M. Renders, M. Saerens, Random-walk computation of similarities between nodes of a graph, with application to collaborative recommendation, IEEE Transactions on Knowledge and Data Engineering 19 (3) (2007) 355–369.