Université catholique de Louvain
École polytechnique de Louvain

Role Extraction in Networks

Thomas P. Cason

Thesis submitted in partial fulfillment of the requirements
for the degree of *Docteur en sciences de l'ingénieur*.

Dissertation committee:

Pierre-Antoine Absil
Vincent D. Blondel
Moritz Diehl
Rodolphe Sepulchre
Paul Van Dooren
Michel Verleysen, Chair

September 2012

Acknowledgment

First of all, I would like to thank my supervisors, Pierre-Antoine Absil and Paul Van Dooren, for their careful interest and useful advises.

I would also like to thank all the members of my thesis committee for their feedback which have helped me in improving this thesis.

I would particularly like to thank Vincent Traag and Jean-Charles Delvenne whose research and knowledge have inspired me several of my most interesting ideas.

I would finally like to thank all the people who made of this thesis an enjoyable experience while we were either working together, sharing an office, or having a break.

And last but not least, I am extremely grateful to my wife for her constant support and understanding throughout these last years.

Contents

List of Notation

$\mathbb{N}$	Set of natural numbers, *i.e.* $\mathbb{N} = \{1, 2, \cdots\}$
$\mathbb{N}_{\leq m}$	Set of natural numbers smaller of equal to m
$\mathbb{R}_{\geq 0}$	Set of non-negative real numbers
A, B, M	Matrices (usually square)
δ_{ij}	Kronecker delta, *i.e.* $\delta_{ij} := 1$ if $i = j$ and 0 otherwise
$I_{m,n}$	Eye matrix, *i.e.* $I_{m,n} := [\delta_{ij}]_{i,j=1}^{m,n}$ (note that $I_m := I_{m,m}$)
$\mathbf{0}_{m,n}$	Zeros matrix, *i.e.* $\mathbf{0}_{m,n} := [0]_{i,j=1}^{m,n}$
$\mathbf{1}_{m,n}$	Ones matrix, *i.e.* $\mathbf{1}_{m,n} := [1]_{i,j=1}^{m,n}$
$A \odot B$	Hadamard product, *i.e.* $[A \odot B]_{i,j} := A_{ij} B_{ij}$
$A \otimes B$	Kronecker product, *i.e.* $A \otimes B := \left[A_{ij} \left[B_{kl}\right]_{k,l=1}^{m_B,n_B}\right]_{i,j=1}^{m_A,n_A}$
$\langle A, B \rangle_F$	Frobenius matrix inner product, *i.e.* $\langle A, B \rangle_F := \mathrm{tr}\left(A^T B\right)$
$\|A\|_F$	Frobenius matrix norm of A, *i.e.* $\|A\|_F := \sqrt{\langle A, A \rangle_F}$
$\rho(A)$	Spectral radius of A 21
$V_{m,n}$	Stiefel manifold, *i.e.* $V_{m,n} := \left\{M \in \mathbb{R}^{m \times n} \text{ s.t. } M^T M = I_n\right\}$
$O(m)$	Orthogonal group of order m, *i.e.* $O(m) := V_{m,m}$
$\mathcal{P}(m)$	Set of permutation matrices, *i.e.* $\mathcal{P}(m) = O(m) \cap \{0,1\}^{m \times m}$
$\mathcal{P}(m,n)$	Set of selection matrices of size $m \times n$ 43
$_p(m,n)$	Set of ordered m-selections in $\mathbb{N}_{\leq n}$
$G = (N, E)$	Graph whose set of nodes is N and set of edges is E .. 9
$G(A) = (N(A), E(A))$	Graph whose adjacency matrix is A 12
$C(i)$	Set of children of node i 10
$P(i)$	Set of parents of node i 10
$\Gamma(i)$	Set of neighbors of node i 10
$\Gamma^r(i)$	Set of neighbors of r^{th} degree of node i 10
$\Gamma^{\leq r}(i)$	Set of neighbors of $\leq r^{\text{th}}$ degree of node i 10
$\wp\left(i \overset{\ell}{\to} j\right)$	Set of walks of length ℓ from the node i to the node j . 11

List of Figures

List of Algorithms

Chapter 1

Introduction

> *And thou shalt make unto it a grate like* networke *of brasse: also upon that grate shalt thou make foure brasen rings upon the foure corners thereof.*
>
> [Tyn60, Exodus xxvii 4]

This is how (from the contraction of the words *net* and *work*) was born the word *network* in 1560 in The Geneva Bible in which it denoted a net-like handwork of threads and wires. Over time, its definition extended to any collection or arrangement of items to resembles a net or anything reticulated or decussated [Joh55], *e.g.*, biological and ecological networks (food webs, spider webs, vascular system in animals and plants, the nervous system), transport networks (rivers, canals, sewers, rail, road and air transport, water, gas and electricity distribution, trade), information and communication networks (World Wide Web, internet, emails and phone network), social networks (collaboration, friendship, or any group of people interconnected by a particular relationship) [COT+07, AB02, New03, Spa85, OED11].

Graph[1] theory began in 1736 when Euler solved the *Königsberg bridge problem* [Eul36, BLW99]. The city of Königsberg (now Kaliningrad) was divided in four distinct territories by the river Pregel which were connected by seven bridges that spanned the river. It is said that the people of Königsberg used to entertain themselves by trying (though they always failed) to find a route around the city which would cross each

[1] Networks are conveniently represented by graphs (*cf.* Definition 2.1)

of the seven bridges exactly once, and possibly return to its starting point. In [Eul36], Euler proved this was impossible, and gave a necessary and sufficient condition for an arbitrary graph to admit a tour of the above kind.

Graphs allow to represent real problems in a very abstract fashion which, though easily stated, raises non trivial mathematical problems. In the past few years, several large networks have become omnipresent in everyday life, which made their analysis a major concern [Str01, DM02, AB02, New03]. Indeed, amongst all problems that arise in graph theory, let us mention below a few of them as examples.

Network *overloads* are problems that most of us have already experienced, either in traffic jams on a road network or power outages on their electric power distribution network. Moreover, major power outages may even be caused by a small perturbation like on the 4^{th} of November 2006, when over fifteen million European customers were left without electricity for two hours after the German electricity company EON switched off an electricity line over the river Ems to allow the cruise ship Norwegian Pearl to pass through safely causing a cascading failure in the electric power distribution network [UCT07].

Network *exploration* while searching for a particular piece of information is an other problem that most internet users are familiar with. The problem of finding relevant websites based on a list of keywords of interest appeared with the creation of the World Wide Web in 1989 [BL89]. And, even though Vannevar Bush already dreamed of a futuristic search engine called the Memex in 1945 [BW45], efficient ones only became real after 1998 with the PageRank link analysis algorithm [BP98] and the HITS algorithm [Kle99] that respectively gave birth to the search engine of Google and Teoma [LM06].

Finally, let us draw attention to network *visualization*, a problem related to role extraction in networks. This problem recently came into the spotlight while considering network that has an underlying community structure, *i.e.* network in which one can identify groups of nodes, called communities, that are significantly more densely connected internally than with the rest of the network. Community detection has been studied for a long time, and has recently been extensively studied in network sciences by Girvan and Newman [GN02, New04] and others [CV07, Chap. 5], [Sch07, POM09, For10]. Community detection has

been applied successfully to several real world example, but is nevertheless useless when one considers networks that have no underlying community structure. In this thesis, we consider an extension of the community detection problem that tries to find groups of nodes that (almost) play the same *role* in the network.

Regular Equivalence Relation

The notion of regularity for equivalence relations (*cf.* Definition 2.10) is one of the many analysis tools used in graph theory to reduce large (potentially incoherent) graphs into smaller comprehensible structures. This notion comes from the field of social networks and was first introduced by White and Reitz in [WR83] while extending the notion of structural equivalence [LW71, Sai79]. This notion has then been extensively studied by Borgatti, Everett, Winship, and others in [WM84, Eve85, Dor87, Win88, Fau88, EB88, Bor89, BE89, BBE89, EB91, BE92a, BE92b, EB93, BE93, BE94, EB94, LBJE02, JBLE03]. Regular equivalence relations partition the nodes of a graph in blocks of nodes that play the same role in the graph. Unfortunately, the notion of regular equivalence relations is very sensitive to small perturbations of the graph under consideration. This thesis focuses on two analysis tools that remedy this sensitivity:

- node-to-node self-similarity measures, and
- Blockmodeling,

which, as explained hereafter, regularly require one to solve non-trivial trace maximization problems.

Node-to-Node Similarity Measures

Node-to-node similarity measures assign real values to each pair of nodes that tells how the first node is similar to the second one based on particular criteria. When both nodes belong to the same graph, one talks about node-to-node self-similarity measures. Node-to-node similarity measures are born in the field of biological sciences and were first introduced by Jaccard in [Jac01] and followed by other new definitions

introduced in [BB32, Sim43, Sor48, CH69]. Later, node-to-node similarity measures came back in the spotlight in the field of computer sciences first applied to information retrieval [SM83, Sal89] and later applied to ranking techniques [BP98, Kle99] which inspired several other similarity measures [RSM+02, MGMR02, JW02, AA03, BGH+04, HH05, Zag05, LHN06, ZLZ09]. An important class of these node-to-node self-similarity measures are designed to give information on how close two nodes are from being equivalent for the *structural equivalence relation* or for the *isomorphic regular equivalence relation* (*cf.* p. 18).

In [BGH+04], Blondel *et al.* introduce a basic node-to-node similarity measure that generalizes the *Hubs and Authorities Algorithm*, introduced by Kleinberg in [Kle99]. This similarity recursively requires that the similarity score between node i and node j (possibly belonging to two different graphs) is large, if the similarity scores between the children of the node i and the children of the node j are large, or if the similarity scores between the parents of node i and the parents of node j are large. This leads them to define a node-to-node similarity measure as an extremal point of a so called *reinforcement loop* (*cf.* p. 30). Moreover, Blondel *et al.* also showed that the similarity matrices associated to their similarity measure are the solutions of trace maximization problems defined on the smooth continuous set of matrices of norm 1.

In Chapter 3, we introduce the notion of *weak and strong compatibility* which links node-to-node self-similarity measures to equivalence relations. We present several existing node-to-node similarity measures, consider their strengths and weaknesses, and propose alternative definitions which improve the earlier. We emphasize that several reinforcement functions of the node-to-node similarity measures are affine functions. We finally classify all similarity measures according to their properties. This chapter is based on ideas that were presented at the *19th International Symposium on Mathematical Theory of Networks and Systems*, held in Budapest, Hungary [CABVD10].

Blockmodeling

Blockmodeling consists in finding equivalence relations in a graph that would be regular if few modifications were made to the original graph, or, in other words, consists in partitioning the nodes of the original

graph in blocks of nodes that almost play the same role in the graph. *E.g.*, a food web is an abstract representation of a group of animals interconnected by the relation "who eats who". This network can be composed of thousands of animals, and hence may not be easy to read. However, those networks can usually be summarized by partitioning the animals in different groups and saying that the animals in the group of herbivores are eaten by animals in the group of primary predators themselves eaten by animals in the group of secondary predators.

The notion Blockmodeling has first been introduced in 1976 by White, Boorman, and Breiger in [WBB76], and has later been extensively studied in social sciences, *cf.* [WF94, DBF04]. More recently, Reichardt and White in [RW07] extended the ideas of Newman's modularity function which are used for community detection [GN02], while other techniques were introduced by Cooper and Barahona in [CB10].

Blockmodeling is a non-trivial problem conveniently implemented by maximizing Blockmodel quality measures which assign high scores to relevant partitions. The Blockmodeling Algorithms presented in this thesis are equivalent to trace maximization problems defined on the discrete set of all partitions, and are NP-hard problems.

In Chapter 4, we define the notion of relevance of a Blockmodel with respect to a given quality measure, we present and analyze existing Blockmodel quality measures, along with their strengths and weaknesses, and we propose novel Blockmodel quality measures. We show that the maximization of the quality measures that we consider can be translated in terms of trace-maximization problems, we propose an Algorithm to solve these problems and test it on a toy example and on the *Baydry* network, a food web observed in the Florida bay.

Trace Maximization on Manifolds

Node-to-Node Similarity Measures and Blockmodeling regularly require one to solve non-trivial trace maximization problems. The solutions of theses optimization problems may not have a closed form expression or may be very expensive to compute and hence require efficient algorithms in order to estimate or compute them. Much research has been done in constrained optimization and several general techniques may be used to solve those problems. Nevertheless, when one talks about matrix con-

strained optimization, the complexity of the problem may dramatically increase according to the number of variables to deal with. However, the constraints may significantly reduce the dimension of the feasible set. Geometric optimization consists in looking at a constrained optimization problem in an unconstrained set in terms of an unconstrained optimization problem in a constrained set. Moreover, if the feasible set is a manifold (*i.e.* may be *locally smoothly* mapped onto $\mathbb{R}^d$, where d is the dimension of the manifold) classical unconstrained optimization methods defined on Euclidean spaces can often be generalized. Recent research has been done in that area using well known machinery from differential geometry in order to build high order generalizations of classical methods [AMS08].

In [FNVD06, FNVD08, FVD07], Fraikin *et al.* consider trace maximization problems associated to the node-to-node similarity measure introduced by Blondel *et al.*, they characterize their critical points, and discuss their optimal values. They present algorithms to solve those problems, along with numerical experiments, and apply them to graph matching problems.

In Chapter 5, we extend the work of Fraikin *et al.* to other trace maximization problems associated to the node-to-node similarity measure introduced by Blondel *et al.*, we characterize their critical points, and discuss their optimal values. We implemented algorithms dedicated to optimization on manifold described in [AMS08] to solve those problems. This chapter is based on a joint work with Pierre-Antoine Absil and Paul Van Dooren [CAVD11] which has been presented at the *Conference in Numerical Analysis – Recent Approaches to Numerical Analysis: Theory, Methods and Applications*, held in Kalamata, Greece.

In Chapter 6, we consider restrictions of the feasible set of the trace maximization problem associated to the node-to-node similarity measure introduced by Blondel *et al.* to sets of low-rank matrices, in order to find a low-rank approximation of the similarity matrix associated to the node-to-node similarity measure introduced by Blondel *et al.*. We first characterize the stationary points of the associated optimization problems and further consider iterative algorithms to find one of them. We analyze the convergence properties of our algorithms, and finally compare our method in terms of speed and accuracy to the full rank algorithm proposed in [BGH$^+$04]. This chapter is based on a joint work

with Pierre-Antoine Absil and Paul Van Dooren [CAVD10] which has been presented at the *16th Conference of the International Linear Algebra Society*, held in Pisa, Italy.

Chapter 2

Preliminaries

In this chapter, we introduce basic concepts that we will extensively use in the following ones. We first consider some general definitions and properties of *graph theory*. Then, we focus on the more specific concept of *role*. We finally give some nice properties of *non-negative matrices*, among which the Perron-Frobenius Theorem.

2.1 Graph and Matrix Representation

Graphs allow to represent real problems in an abstract fashion. This section introduces basic concepts and useful properties of graph theory.

Definition 2.1 (Graph [Die91, p. 2]) *Let $N = \mathbb{N}_{\leq m}$ be a non-empty set of indexes and $E \subseteq N \times N$ be a relation on N. The pair $G = (N, E)$ is what we further call a graph (of order m). An element of N is called a node. An element of E is called an edge.*

A convenient way to represent a graph is by drawing,

- for each node of the graph, an item numbered according to its index, and,
- for each edge of the graph, an arrow from its first element to its second one.

E.g., see Figure 2.1.

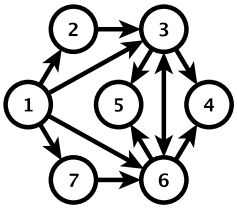


Figure 2.1: Toy directed graph

Let $G = (N, E)$ be a graph. The graph G is *undirected* if the relation E is symmetric. For readability, the arrows representing edges in the graphical representation of an undirected graph are usually replaced by undirected links. *E.g.*, see Figure 2.2.

The node $j \in N$ is a *child* (resp. a *parent*) of the node $i \in N$ if (i, j) (resp. (j, i)) belongs to E. The node j is a *neighbor* of the node i if the node j is either a child or a parent of the node i. The set of children, parents and neighbors of the node i are respectively denoted $C(i)$, $P(i)$ and $\Gamma(i)$. *E.g.*, in Figure 2.1, the set of children (resp. parents) of node 3 is $\{4, 5, 6\}$ (resp. $\{1, 2, 6\}$). Given $r \in \mathbb{N}$, the node j is a *child* (resp. a *parent* or a *neighbor*) of r^{th} degree of the node i if

- either $r = 1$ and the node j is a *child* (resp. a *parent* or a *neighbor*) of the node i,
- or $r \geq 2$, there exists a node $k \in C(i)$ (resp. $k \in P(i)$ or $k \in \Gamma(i)$) such that the node j is a *child* (resp. a *parent* or a *neighbor*) of $r - 1^{\text{th}}$ degree of the node k.

The set of children, parents and neighbors of r^{th} degree of the node i are respectively denoted $C^r(i)$, $P^r(i)$ and $\Gamma^r(i)$. *E.g.*, in Figure 2.1, the set of children (resp. parents) of 2^{nd} degree of node 3 is $\{3\}$ (resp. $\{1, 3, 7\}$). Moreover, the set of children, parents and neighbors of degree less than or equal to r^{th} of the node i are respectively denoted $C^{\leq r}(i)$, $P^{\leq r}(i)$ and $\Gamma^{\leq r}(i)$.

Walking in a graph is one of the basic tools extensively used to characterize the underlying properties of the graph under consideration.

Definition 2.2 (Walk [Die91, p. 9]) *Let $G = (N, E)$ be a graph. A walk of length ℓ in G from the node $i \in N$ to the node $j \in N$ is a non-empty alternating sequence $n_1, e_1, n_2, e_2, \cdots, e_\ell, n_{\ell+1}$ of nodes and edges in G such that $n_1 = i$, $n_{\ell+1} = j$, and $e_t = (n_t, n_{t+1})$ for all $t \in \{1, \cdots, \ell\}$. The nodes n_1 and $n_{\ell+1}$ are respectively called the origin and termination of the walk.*

Note that a node or an edge can appear more than once in a walk. The set of walks of length ℓ from the node i to the node j is denoted $\wp\left(i \xrightarrow{\ell} j\right)$.

A graph may be represented without equivocation by the set of its nodes and edges or by its graphical representation, nevertheless these representations are sometimes not handy. We now introduce the notion of *matrix representation* which will allow us to represent a graph by a matrix, called its *adjacency matrix*.

Definition 2.3 (Matrix representation) *Let R be a binary relation between $\mathbb{N}_{\leq m}$ and $\mathbb{N}_{\leq n}$ (*i.e. *$R \subseteq \mathbb{N}_{\leq m} \times \mathbb{N}_{\leq n}$). The matrix representation of R, denoted $M(R)$, is a matrix in $\{0,1\}^{m \times n}$ defined as*

$$[M(R)]_{i,j} := \begin{cases} 1, & \text{if } (i,j) \in R, \text{ and} \\ 0, & \text{otherwise.} \end{cases}$$

And, the vector representation of $i \in \mathbb{N}_{\leq m}$, denoted $v(i)$, is a vector in $\{0,1\}^m$ defined as

$$[v(i)]_j := \begin{cases} 1\ , & \text{if } i = j, \text{ and} \\ 0\ , & \text{otherwise.} \end{cases}$$

Note that, given a function $f : \mathbb{N}_{\leq m} \to \mathbb{N}_{\leq n}$, we have $v(f(i)) = M(f)\ v(i)$.

Definition 2.4 (Adjacency Matrix [Die91, p. 23]) *Let $G = (N, E)$ be a graph with $N = \mathbb{N}_{\leq m}$ for some m. The matrix representation of the set of edges, $M(E)$, is called the adjacency matrix of G.*

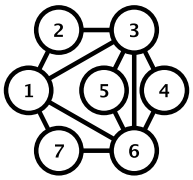


Figure 2.2: Toy undirected graph

E.g., the adjacency matrix of the graph $G := (N, E)$ defined with $N = \{1, \cdots, 7\}$ and

$$E = \left\{ \begin{array}{l} (1,2),(1,3),(1,6),(1,7),(2,1),(2,3),(3,1), \\ (3,2),(3,4),(3,5),(3,6),(4,3),(4,6),(5,3), \\ (5,6),(6,1),(6,3),(6,4),(6,5),(6,7),(7,1),(7,6) \end{array} \right\},$$

graphically represented in Figure 2.2, is

$$A = \begin{bmatrix} 0&1&1&0&0&1&1 \\ 1&0&1&0&0&0&0 \\ 1&1&0&1&1&1&0 \\ 0&0&1&0&0&1&0 \\ 0&0&1&0&0&1&0 \\ 1&0&1&1&1&0&1 \\ 1&0&0&0&0&1&0 \end{bmatrix} .$$

Conversely, given a matrix $A \in \{0,1\}^{m \times m}$, the graph representation of A, denoted $G(A) = (N(A), E(A))$, is the graph of order m whose adjacency matrix is A. In particular, we have $N(A) = \mathbb{N}_{\leq m}$ and $M(E(A)) = A$. Note that the graph $G(A)$ is undirected if and only if its adjacency matrix A is symmetric.

Proposition 2.1 *Let $A \in \{0,1\}^{m \times m}$ be an adjacency matrix of the graph $G(A) = (N(A), E(A))$. The number of walks of length ℓ in $G(A)$ from node i to node j, given by $\left|\wp\left(i \overset{\ell}{\rightarrow} j\right)\right|$, is equal to the (i,j) entry of the ℓ^{th} power of A,* i.e. $[A^\ell]_{i,j}$. *[Big93, Lemma 2.5]*

Proof: This is obviously true for $\ell = 0$ since the number of paths of length 0 from the node i to the node j is either equal to 1 if $i = j$, or equal to 0 otherwise, and $A^0 = I$. Let us assume it is true for ℓ and

prove it for $\ell + 1$.

$$[A^{\ell+1}]_{i,j} = \sum_k A_{ik}[A^\ell]_{k,j} = \sum_{k \in C(i)} [A^\ell]_{k,j}$$

$$= \sum_{k \in C(i)} \left| \wp\left(k \overset{\ell}{\to} j\right) \right| = \left| \wp\left(i \overset{\ell+1}{\to} j\right) \right| .$$

□

E.g., let us consider the graph represented in Figure 2.2, and its adjacency matrix A. There are 7 walks of length 3 form node 2 to node 3, which are $(2,1,2,3)$, $(2,1,6,3)$, $(2,3,1,3)$, $(2,3,2,3)$, $(2,3,4,3)$, $(2,3,5,3)$, and $(2,3,6,3)$, and $[A^3]_{2,3} = 7$.

If there exists a relabeling of the nodes of a graph G that makes it equal to another graph G', then G and G' have the same intrinsic properties, and are called *isomorphic graphs*.

Definition 2.5 (Isomorphism [Die91, p. 3]) *Let $G = (N, E)$ and $G' = (N', E')$ be two graphs. The function $\sigma : N \to N'$ is called an* isomorphism *from G to G' if*

- *σ is bijective, and*
- *for all $i, j \in N$, $(i, j) \in E$ if and only if $(\sigma(i), \sigma(j)) \in E'$.*

If there exists an isomorphism from G to G', then we say that G and G' are isomorphic graphs, or $G \sim G'$. If there exists an isomorphism from $G = (N, E)$ to $G' = (N', E')$ that maps node $i \in N$ onto node $j \in N'$, then we say that i and j are isomorphic nodes or $i \sim j$.

Let $A, B \in \{0,1\}^{m \times m}$. The matrix representation of any isomorphism from $G(A)$ to $G(B)$ is a matrix $P \in \mathcal{P}(m)$ such that $P^T A P = B$.

A *weighted graph* is a graph in which the edges are weighted proportionally to their relative importance.

Definition 2.6 (Weighted-graph [BM76, pp. 15-16]) *Let $N = \mathbb{N}_{\leq m}$ be a set of indexes and $w : N \times N \to \mathbb{R}$ a function. The pair $G_w = (N, w)$ is what we further call a weighted graph (of order m). An element of N is called a node. An element of $E := w^{-1}(\mathbb{R}_{\neq 0})$ is called an edge.*

- *The weighted adjacency matrix of G_w is a matrix $A \in \mathbb{R}^{m\times m}$ defined as*
$$A_{ij} := w(i,j) \ .$$
- *Given a matrix $A \in \mathbb{R}^{m\times m}$, the weighted graph representation of A, denoted $G_w(A) = (N(A), w_A)$ is the weighted graph of order n, whose weighted adjacency matrix is A.*

2.2 Image Graph and Role Coloring

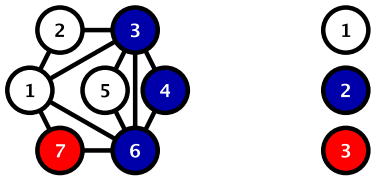
Role coloring a graph consists in coloring its nodes according to their role such that it emphasizes their underlying interconnection, and eventually summarizes the original graph into a smaller compressible structure called the image graph. In this section, we first introduce the notion of regular partition and regular equivalence relation, we further formally define the notion of role and image graph, and we finally emphasize different types of regular equivalence relations.

An indexed partition split the nodes of a graph into distinct indexed blocks of nodes.

Definition 2.7 (Indexed Partition) *Let $G = (N, E)$ be a graph, and $n \in \mathbb{N}$. A function $\sigma : N \to \mathbb{N}_{\leq n}$ is an indexed partition of G in n blocks. The set $\sigma^{-1}(i)$ is called the i^{th} block.*

Note that a block of σ can be empty. *E.g.*, let us consider G, the graph of Figure 2.2. The function $\sigma : \mathbb{N}_{\leq 7} \to \mathbb{N}_{\leq 3}$, defined as

$$\begin{array}{l}\sigma(1) = 1,\\ \sigma(2) = 1,\\ \sigma(3) = 2,\\ \sigma(4) = 2,\\ \sigma(5) = 1,\\ \sigma(6) = 2, \text{ and}\\ \sigma(7) = 3,\end{array}$$



is an indexed partition of G in 3 blocks. The blocks of the indexed partition are $\sigma^{-1}(1) = \{1, 2, 5\}$, $\sigma^{-1}(2) = \{3, 4, 6\}$, and $\sigma^{-1}(3) = \{7\}$, respectively colored in the image above in white, blue, and red.

Definition 2.8 (Regular Partition) *Let $G = (N, E)$ be a graph, and $\sigma : N \to \mathbb{N}_{\leq n}$ be an indexed partition of G. The partition σ is regular with respect to G if, for all $i, j \in N$, $\sigma(i) = \sigma(j)$ implies*

$$\{\sigma(k) \text{ s.t. } k \in C(i)\} = \{\sigma(\ell) \text{ s.t. } \ell \in C(j)\} \ ,$$

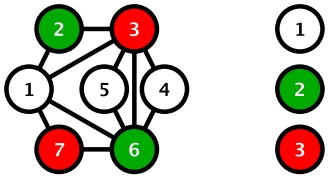
and

$$\{\sigma(k) \text{ s.t. } k \in P(i)\} = \{\sigma(\ell) \text{ s.t. } \ell \in P(j)\} \ .$$

Or in other words, if i and j belong to the same block, then the set of blocks which the parents (resp. children) of i belong to, is equal to the set of blocks which the parents (resp. children) of j belong to.

E.g., let us consider G, the graph of Figure 2.2. The function $\sigma : \mathbb{N}_{\leq 7} \to \mathbb{N}_{\leq 3}$ defined as

$$\begin{array}{l} \sigma(1) = 1, \\ \sigma(2) = 2, \\ \sigma(3) = 3, \\ \sigma(4) = 1, \\ \sigma(5) = 1, \\ \sigma(6) = 2, \text{ and} \\ \sigma(7) = 3, \end{array}$$



is a regular indexed partition of G in 3 blocks. Indeed, $\sigma(3) = \sigma(7) = 3$, and the sets of the indexes of the blocks of their respective neighbors are also equal,

$$\sigma(\{\Gamma(3)\}) = \sigma(\{1, 2, 4, 5, 6\}) = \{1, 2\} = \sigma(\{1, 6\}) = \sigma(\{\Gamma(7)\}) \ .$$

The blocks of this regular indexed partition are $\sigma^{-1}(1) = \{1, 4, 5\}$, $\sigma^{-1}(2) = \{2, 6\}$, and $\sigma^{-1}(3) = \{3, 7\}$, and respectively colored in the image above in white, green, and red.

An *equivalence relation* is a particular kind of relation that is reflexive, symmetric and transitive.

Definition 2.9 (Equivalence Relation [STT77, p. 74]) *Let N be a non empty set. The relation $R \subseteq N \times N$ is an* equivalence relation *on N if*

- *R is reflexive: $(i, i) \in R$, $\forall i \in N$,*

- *R is symmetric: if $(i,j) \in R$, then $(j,i) \in R$, $\forall i,j \in N$, and*
- *R is transitive: if $(i,j),(j,k) \in R$, then $(i,k) \in R$, $\forall i,j,k \in N$.*

Let now i be an element of N. The set

$$[i] := \{j \in N \text{ s.t. } (i,j) \in R\}$$

is called the equivalence class *associated to i.*

Definition 2.10 (Regular Equivalence Relation) [WR83, Def. 11], [EB91, pp. 1,2] *Let $G = (N,E)$ be a graph. An equivalence relation $R \subseteq N \times N$ is regular with respect to G if, for all $i,j \in N$, $[i] = [j]$, implies*

$$\{[k] \text{ s.t. } k \in C(i)\} = \{[\ell] \text{ s.t. } \ell \in C(j)\} \ ,$$

and

$$\{[k] \text{ s.t. } k \in P(i)\} = \{[\ell] \text{ s.t. } \ell \in P(j)\} \ .$$

In other words, let the nodes be colored according to their equivalence classes, then R is called regular if, whenever the nodes i and j are colored alike, the set of colors which the parents (resp. children) of i are colored with is equal to the set of colors which the parents (resp. children) of j are colored with.

We now show that an indexed partition of the nodes of a graph naturally induces an equivalence relation on the nodes of this graph. Moreover, if that indexed partition is regular, then the induced equivalence relation is also regular.

Lemma 2.2 *Let $\sigma : N \to \mathbb{N}_{\leq n}$ be an indexed partition of $G = (N,E)$. Then, R_σ, the relation induced by σ, defined as*

$$R_\sigma = \{(i,j) \in N \times N \text{ s.t. } \sigma(i) = \sigma(j)\} \ ,$$

is an equivalence relation whose equivalence classes are the n blocks, $\sigma^{-1}(1), \cdots, \sigma^{-1}(n)$. Moreover, if σ is regular then R_σ is regular.

Proof: this property can be directly derived from [WR83, Lemma 1]. □

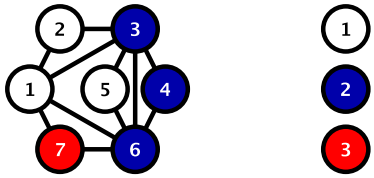
An image graph induced by an indexed partition is a graph whose nodes are the indexes associated to the blocks of the indexed partition, and whose edges each indicate whether there exists an edge between nodes belonging to the blocks it connects or not.

Definition 2.11 (Image Graph [LBJE02, Sec. 2.4]) *Let $G = (N, E)$ be a graph, and $\sigma : N \to \mathbb{N}_{\leq n}$ be an indexed partition of G. The graph $G' = (N', E')$ is called the* image graph *of G induced by σ if*

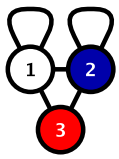
- $N' = \mathrm{codom}(\sigma) = \mathbb{N}_{\leq n}$, *and*
- $E' = \{(\sigma(i), \sigma(j)) \text{ s.t. } (i, j) \in E\}$.

E.g., let us consider G, the graph of Figure 2.2, and the indexed partition of G defined as

$$\begin{array}{l} \sigma(1) = 1, \\ \sigma(2) = 1, \\ \sigma(3) = 2, \\ \sigma(4) = 2, \\ \sigma(5) = 1, \\ \sigma(6) = 2, \text{ and} \\ \sigma(7) = 3. \end{array}$$



The image graph of G induced by σ is $G' = \left(\{1,2,3\}, \{1,2,3\}^2 \setminus \{(3,3)\}\right)$, *i.e.*



Let $G = (N, E)$ be a graph, σ be an indexed partition on G, and $G' = (N', E')$ be the image graph of G induced by σ. If σ is regular, then the nodes of G' are called *roles*. We say that the node $i \in N$ plays the role $j = \sigma(i) \in N'$. In Chapter 3, we consider node to node similarity measures that quantify how close two nodes are to playing the same role according to an equivalence relation.

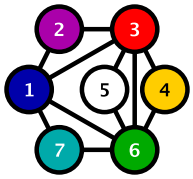
Let us consider $G = (N, E)$, the graph of Figure 2.2. Figure 2.3 shows all equivalence relations on N regular with respect to G, along with the respective image graph induced by an arbitrary indexed partition whose blocks are their equivalence classes. These regular colorings emphasize the underlying interconnection between the nodes of the graph. Each image graph is a summarized version of how one could observe the original graph.

Let A be an adjacency matrix. We list below some particular regular equivalence relations on $N(A)$.

1. **The minimal regular equivalence relation** on $N(A)$, denoted $R_{\min}(A)$, is the set of reflexive pairs, *i.e.*

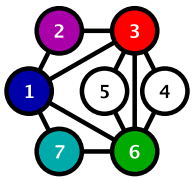
$$R_{\min}(A) := \{(i,i) \text{ s.t. } i \in N(A)\} \ .$$

And, the equivalence class associated to node i (denoted $[i]_{\min}$) is the singleton $\{i\}$, or in other words, all nodes are colored in different colors.



2. **The structural equivalence relation** on $N(A)$, denoted $R_{\text{struct}}(A)$, is the set of pairs of nodes $(i,j) \in N(A) \times N(A)$, s.t. i and j have the same children and parents, *i.e.*

$$R_{\text{struct}}(A) = \left\{ \begin{array}{r} (i,j) \in N(A)^2 \text{ s.t. } C(i) = C(j) \ , \\ \text{and } P(i) = P(j) \end{array} \right\} \ . \tag{2.1}$$



If the nodes i and j belong to the same structural equivalence

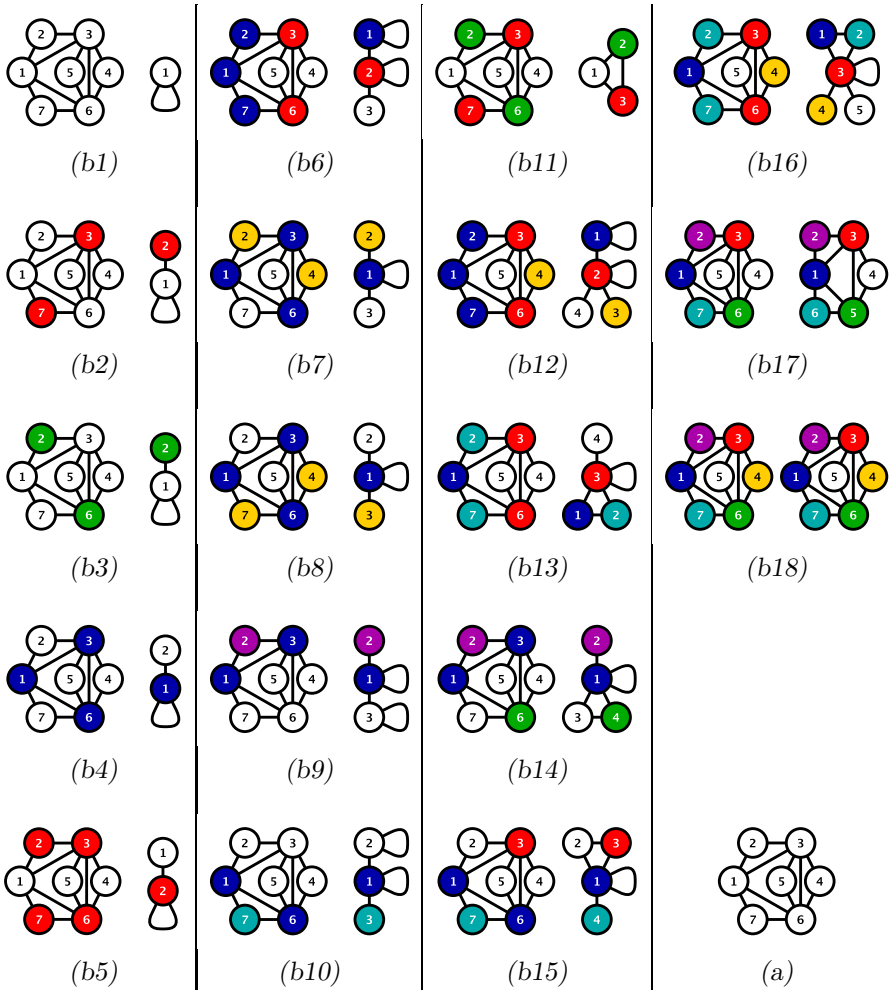


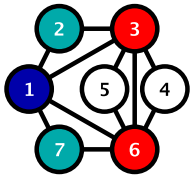
Figure 2.3: (a) $G = (N, E)$ *is an undirected graph. Amongst the relations induced by an indexed partition on* N*, eighteen are regular with respect to* G*. Let us denote them* R_k *with* $k = 1, \cdots, 18$*. (b"k") The graph on the left-hand-side is* $G = (N, E)$ *and the graph on the right-hand-side is denoted* $G' = (N', E')$*. The nodes of* N *are colored according to the equivalence class of* R_k *they belong to. The nodes of* N' *are colored according to* $\sigma : N \to N'$*, an arbitrary indexed partition that induces* R_k*,* i.e. *for all* $i \in N$*,* $\sigma(i) = j$ *if and only if the color of* $\sigma(i)$ *is the color of* j*. The graph* $G' = (N', E')$ *is the image graph of* G *induced by* σ*.*

class, *i.e.* $(i,j) \in R_{\text{struct}}(A)$, then the nodes i and j are termed *structurally equivalent*.

3. **The isomorphic regular equivalence relation** on $N(A)$, denoted $R_{\text{iso}}(A)$, is the set of pairs of nodes $(i,j) \in N(A) \times N(A)$, s.t. i is isomorphic to j, *i.e.*

$$R_{\text{iso}}(A) = \left\{(i,j) \in N(A)^2 \text{ s.t. } i \sim j\right\} \ .$$

The isomorphic regular equivalence relation is also known as automorphic coloration (see [BE94]).

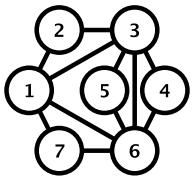


4. **The maximal regular equivalence relation** on $N(A)$, denoted $R_{\max}(A)$. This relation is defined by an induction. Let $R_1 = N(A) \times N(A)$ be an equivalence relation[1] and $[i]_1 (= N(A))$ denotes the equivalence class of R_1 associated to node i. In other words, all nodes are colored with the same color. We further recursively define (for $t = 1, 2, \cdots$)

$$R_{t+1} = \left\{ \begin{array}{ll} (i,j) & \text{s.t. } \{[k]_t \text{ s.t. } k \in C(i)\} = \{[\ell]_t \text{ s.t. } \ell \in C(j)\} \\ & \text{and } \{[k]_t \text{ s.t. } k \in P(i)\} = \{[\ell]_t \text{ s.t. } \ell \in P(j)\} \end{array} \right\} \ ,$$

where $[i]_t$ denotes the equivalence class of R_t associated to node i. Since $N(A)$ is finite, there must be a T such that $R_{T+1} = R_T$. In [WR83, Th. 3C], White *et al.* prove that all regular equivalence relations are subsets of R_T, also called the maximal regular equivalence relation or $R_{\max}(A)$. Note that if all nodes have at least one child and one parent, then $R_{\max}(A) = R_1$.

[1] There exist graphs for which the equivalence relation R_1 is not regular, *e.g.*, it is the case for the graph whose adjacency matrix is $\left[\begin{smallmatrix} 0 & 1 \\ 0 & 0 \end{smallmatrix}\right]$.



2.3 Non-negative Matrices

A matrix is termed non-negative (resp.positive) if all its entries are non-negative (resp.positive). These matrices have nice properties (see [HJ90, Chap. 8],[Mey00, Chap. 8]) in particular concerning their eigenvalues of maximum modulus and their associate non-negative eigenvectors. In this section, we first introduce several definitions and then present several of these nice properties in Theorem 2.4.

We now define the *spectral radius* of a matrix which can be seen as the largest multiplication factor of the norm of any vector different from zero when one multiplies it by that matrix.

Definition 2.12 (Spectral Radius [HJ90, Def. 5.6.8]) *Let A be a matrix in $\mathbb{R}^{m\times m}$. The spectral radius of A, denoted $\rho(A)$, is defined as*

$$\max\{|\lambda| \text{ s.t. } Au = \lambda u \text{ with } \|u\| \neq 0\} \ .$$

Definition 2.13 (Reducible [HJ90, Def. 6.2.21]) *The matrix $A \in \mathbb{R}^{m\times m}$ is said to be reducible if*

- *either $m = 1$ and $A = 0$,*
- *or $m \geq 2$ and there exists a permutation $P \in \mathcal{P}(m)$, and a $1 \leq k \leq m-1$, such that*

$$PAP^T = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix} \ ,$$

with $A_{11} \in \mathbb{R}^{k\times k}$, $A_{12} \in \mathbb{R}^{k\times(m-k)}$, and $A_{22} \in \mathbb{R}^{(m-k)\times(m-k)}$.

The matrix $A \in \mathbb{R}^{m\times m}$ is said to be irreducible if it is not reducible[HJ90, Def. 6.2.22]. Note that an adjacency matrix A is irreducible if and only if $G(A)$ is strongly connected[2].

Definition 2.14 (Primitive [HJ90, Def. 8.5.0]) *The matrix $A \in \mathbb{R}^{m\times m}$ is called primitive if A is non-negative, A is irreducible, and A has exactly one eigenvalue of maximum modulus.*

Theorem 2.3 ([HJ90, Th. 8.5.2]) *Let A be a non-negative matrix. A is primitive if and only if there exists a $k \geq 1$ such that A^k is positive.*

Notice that, in particular, positive matrices are primitive. In 1907, Oscar Perron found remarkable properties of positive matrices, which are actually also applicable to primitive matrices as we will further mention in Theorem 2.4. The name of Frobenius is associated with generalizations of Perron's results about positive matrices to non-negative matrices. However, as we will mention in Theorem 2.4, the results of Frobenius originally only concerned irreducible matrices [Gan00, XIII. 2. Th. 1].

Theorem 2.4 *Let A be a matrix in $\mathbb{R}^{m\times m}$. If A is non-negative, then*

- *$\rho(A)$ is an eigenvalue of A, [HJ90, Th. 8.3.1],*
- *there exists at least one vector $u \in \mathbb{R}^m$ such that $u \geq 0$, $\|u\| \neq 0$, and $Au = \rho(A)u$, [HJ90, Th. 8.3.1],*
- *if $Au = \lambda u$ and $u > 0$ then $\lambda = \rho(A)$, [HJ90, Cor. 8.1.30],*
- *$\rho(A) = \max_{\{u|u\geq 0, \|u\|\neq 0\}} \min_{\{i|u_i\neq 0\}} \frac{[Au]_i}{u_i}$, [HJ90, Cor. 8.3.3],*

*moreover, (**Frobenius**) if A is irreducible, then*

- *$\rho(A) > 0$, ($\rho(A)$ is called the Perron root of A) [HJ90, Th. 8.4.4 (a)],*

[2] A graph is strongly connected if, for all nodes i, j, there exists a walk from i to j [Cha85, p. 149].

- *there exists exactly one vector* $u \in \mathbb{R}^m$ *such that* $u > 0$, $\|u\| = 1$, *and* $Au = \rho(A)u$, *(*u *is called the Perron vector of* A*) [HJ90, Th. 8.4.4 (c)],*
- *the algebraic multiplicity and geometric multiplicity of* $\rho(A)$ *are equal,* i.e. $m_a(\rho(A)) = m_g(\rho(A)) = 1$, *[HJ90, Th. 8.4.4 (d)],*
- *if* A *has* $k \geq 2$ *eigenvalues of maximum modulus, then each non-zero eigenvalue of* A *lies on a circle centered at* 0 *in* $\mathbb{C}$ *that passes through exactly* k *eigenvalues of* A*, all equally spaced around the circle, [HJ90, Remark 8.4.7],*

*moreover, (**Perron**) if* A *is primitive, then*

- $\rho(A)$ *is the unique eigenvalue of maximum modulus of* A*,*
- $\lim_{k\to\infty} \left[\rho(A)^{-1}A\right]^k = uv^T$ *with* $Au = \rho(A)u$, $A^Tv = \rho(A)v$, $u > 0$, $v > 0$, *and* $u^Tv = 1$, *[HJ90, Th. 8.5.1].*

Chapter 3

Node-to-Node Similarity and Self-Similarity Measures

In this chapter, we first emphasize the difference between the concepts of *node-to-node similarity measure* and *node-to-node self-similarity measure*, introduce the notion of *weak and strong compatibility*, and present the *reinforcement loop algorithm*. Then, in Section 3.1 (resp. Section 3.2), we present several existing node-to-node similarity (resp. self-similarity) measures, consider their strengths and weaknesses, and propose alternative definitions which improve the earlier, and finally classify all similarity (resp. self-similarity) measures according to their properties.

A *node-to-node similarity measure* tells how the nodes of a graph are similar to the ones of another graph based on particular criteria. More precisely, given two adjacency matrices, $A \in \mathbb{R}^{m \times m}$ and $B \in \mathbb{R}^{n \times n}$, a node-to-node similarity measure assigns a real value to any pair of nodes $(i, j) \in N(A) \times N(B)$ that tells how the node i is similar to the node j, and conveniently stores this node-to-node similarity score in the (i, j) entry of the so-called *similarity matrix*, denoted $S(A, B)$.

A *node-to-node self-similarity measure* tells how the nodes of a graph are similar to themselves based on particular criteria. More precisely, given an adjacency matrix, $A \in \mathbb{R}^{m \times m}$, a node-to-node self-similarity

measure assigns a real value to any pair of nodes $(i, j) \in N(A) \times N(A)$ that tells how the node i is similar to the node j, and conveniently stores this node-to-node self-similarity score in the (i, j) entry of the so-called *self-similarity matrix*, denoted $S(A)$. Note that any node-to-node similarity measure $S(\cdot, \cdot)$ trivially induces a node-to-node self-similarity measure $S'(\cdot)$ defined as $S'(A) := S(A, A)$.

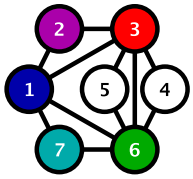
Weak and Strong Compatibility

An equivalence relation on the nodes of a graph dichotomously indicates if two nodes are equivalent or not (or, equivalently, the matrix representation of an equivalence relation can be seen as a trivial binary self-similarity matrix) but it does not carry any information on how close two nodes are from being equivalent. An important class of node-to-node self-similarity measures remedies this by assigning to the pair of nodes (i, j)

- either the value 1 if the node i is equivalent to the node j,
- or a real value (usually between 0 and 1) whose closeness to 1 indicates how the node i is close to being equivalent to the node j.

E.g., let A be the adjacency matrix of the graph represented in Figure 2.2. The matrix representation of its structural equivalence relation is

$$M(R_{\text{struct}}(A)) = \begin{bmatrix} \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{1} & \mathbf{1} & 0 & 0 \\ 0 & 0 & 0 & \mathbf{1} & \mathbf{1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} \end{bmatrix}.$$



Here are two self-similarity matrices built using two node-to-node self-similarity measures that will be further defined in Section 3.2:

$$S^{\text{corr.}}(A) = \begin{bmatrix} \mathbf{1} & -0.09 & -0.54 & 0.54 & 0.54 & -0.54 & -0.09 \\ -0.09 & \mathbf{1} & -0.3 & 0.3 & 0.3 & 0.4 & 0.3 \\ -0.54 & -0.3 & \mathbf{1} & -0.3 & -0.3 & -0.4 & 0.4 \\ 0.54 & 0.3 & -0.3 & \mathbf{1} & \mathbf{1} & -0.3 & 0.3 \\ 0.54 & 0.3 & -0.3 & \mathbf{1} & \mathbf{1} & -0.3 & 0.3 \\ -0.54 & 0.4 & -0.4 & -0.3 & -0.3 & \mathbf{1} & -0.3 \\ -0.09 & 0.3 & 0.4 & 0.3 & 0.3 & -0.3 & \mathbf{1} \end{bmatrix},$$

and

$$S^{\cap\Gamma^1}_{w=S_{ii}}(A) = \begin{bmatrix} \mathbf{1} & 0.25 & 0.5 & 0.5 & 0.5 & 0.5 & 0.25 \\ 0.5 & \mathbf{1} & 0.5 & 0.5 & 0.5 & 1 & 0.5 \\ 0.4 & 0.2 & \mathbf{1} & 0.2 & 0.2 & 0.6 & 0.4 \\ 1 & 0.5 & 0.5 & \mathbf{1} & \mathbf{1} & 0.5 & 0.5 \\ 1 & 0.5 & 0.5 & \mathbf{1} & \mathbf{1} & 0.5 & 0.5 \\ 0.4 & 0.4 & 0.6 & 0.2 & 0.2 & \mathbf{1} & 0.2 \\ 0.5 & 0.5 & 1 & 0.5 & 0.5 & 0.5 & \mathbf{1} \end{bmatrix} .$$

Note that a pair of nodes whose self-similarity score is equal to 1 are not necessarily equivalent.

This leads us to introduce the notion of weak and strong compatibility.

Definition 3.1 (Weak and Strong Compatibility) *Let $R(\cdot)$ be a function that maps the binary adjacency matrix A onto $R(A)$ a relation on $N(A)$. A node-to-node self-similarity measure $S(\cdot)$ is* weakly compatible *with $R(\cdot)$ if, for any adjacency matrix A, for all nodes $i, j \in N(A)$, we have*

$$(i,j) \in R(A) \quad \Rightarrow \quad [S(A)]_{i,j} = 1 \ .$$

A node-to-node self-similarity measure $S(\cdot)$ is strongly compatible *with $R(\cdot)$ if, for any adjacency matrix A, for all nodes $i, j \in N(A)$, we have*

$$(i,j) \in R(A) \quad \Leftrightarrow \quad [S(A)]_{i,j} = 1 \ .$$

We also introduce specific terms for the weak and strong compatibility with regular equivalence relation introduced on page 18.

Definition 3.2 *Let $R_{\min}(\cdot)$, $R_{\text{struct}}(\cdot)$, $R_{\text{iso}}(\cdot)$, and $R_{\max}(\cdot)$ be functions that map the binary adjacency matrix A onto*

- *$R_{\min}(A)$, the minimal regular equivalence relation on $N(A)$,*
- *$R_{\text{struct}}(A)$, the structural regular equivalence relation on $N(A)$,*
- *$R_{\text{iso}}(A)$, the isomorphic regular equivalence relation on $N(A)$, and*
- *$R_{\max}(A)$, the maximal regular equivalence relation on $N(A)$,*

$\mathcal{A}$ be a set of adjacency matrices, and $S(\cdot)$ be a node-to-node self-similarity measure. If, for each adjacency matrix $A \in \mathcal{A}$, $S(A)$ is weakly (resp. strongly) compatible with

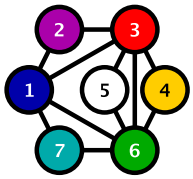
- $R_{\min}(A)$, *then* $S(\cdot)$ *is termed* weakly *(resp.* strongly*)* minimal *on* $\mathcal{A}$.
- $R_{\text{struct}}(A)$, *then* $S(\cdot)$ *is termed* weakly *(resp.* strongly*)* structural *on* $\mathcal{A}$.
- $R_{\text{iso}}(A)$, *then* $S(\cdot)$ *is termed* weakly *(resp.* strongly*)* isomorphic *on* $\mathcal{A}$.
- $R_{\max}(A)$, *then* $S(\cdot)$ *is termed* weakly *(resp.* strongly*)* maximal *on* $\mathcal{A}$.

Omitting to specify the set of adjacency matrices $\mathcal{A}$ *means that this set is the set of all adjacency matrices,* i.e.

$$\mathcal{A} = \bigcup_{n \in \mathbb{N}} \{0,1\}^{n \times n} \ .$$

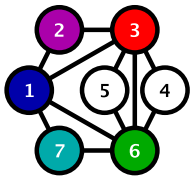
E.g., let us consider A the adjacency matrix of the graph represented in Figure 2.2, and $R_{\min}(A)$, $R_{\text{struct}}(A)$, $R_{\text{iso}}(A)$, and $R_{\max}(A)$ be respectively the minimal, the structural, the isomorphic, and the maximal regular equivalence relation on $N(A)$.

- $S_{\min}(A) := \begin{bmatrix} \mathbf{1} & S_{12} & S_{13} & S_{14} & S_{15} & S_{16} & S_{17} \\ S_{21} & \mathbf{1} & S_{23} & S_{24} & S_{25} & S_{26} & S_{27} \\ S_{31} & S_{32} & \mathbf{1} & S_{34} & S_{35} & S_{36} & S_{37} \\ S_{41} & S_{42} & S_{43} & \mathbf{1} & S_{45} & S_{46} & S_{47} \\ S_{51} & S_{52} & S_{53} & S_{54} & \mathbf{1} & S_{56} & S_{57} \\ S_{61} & S_{62} & S_{63} & S_{64} & S_{65} & \mathbf{1} & S_{67} \\ S_{71} & S_{72} & S_{73} & S_{74} & S_{75} & S_{76} & \mathbf{1} \end{bmatrix}$.



  $S_{\min}(A)$ is weakly compatible with $R_{\min}(A)$. Moreover, if all unspecified entries of $S_{\min}(A)$ are different from 1, then $S_{\min}(A)$ is strongly compatible with $R_{\min}(A)$.

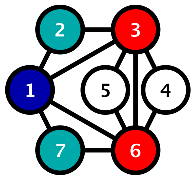
- $S_{\text{struct}}(A) := \begin{bmatrix} \mathbf{1} & S_{12} & S_{13} & S_{14} & S_{15} & S_{16} & S_{17} \\ S_{21} & \mathbf{1} & S_{23} & S_{24} & S_{25} & S_{26} & S_{27} \\ S_{31} & S_{32} & \mathbf{1} & S_{34} & S_{35} & S_{36} & S_{37} \\ S_{41} & S_{42} & S_{43} & \mathbf{1} & \mathbf{1} & S_{46} & S_{47} \\ S_{51} & S_{52} & S_{53} & \mathbf{1} & \mathbf{1} & S_{56} & S_{57} \\ S_{61} & S_{62} & S_{63} & S_{64} & S_{65} & \mathbf{1} & S_{67} \\ S_{71} & S_{72} & S_{73} & S_{74} & S_{75} & S_{76} & \mathbf{1} \end{bmatrix}$.



  $S_{\text{struct}}(A)$ is weakly compatible with $R_{\text{struct}}(A)$. Moreover, if all

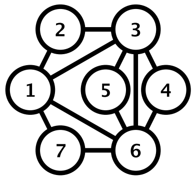
unspecified entries of $S_{\text{struct}}(A)$ are different from 1, then $S_{\text{struct}}(A)$ is strongly compatible with $R_{\text{struct}}(A)$.

- $S_{\text{iso}}(A) := \begin{bmatrix} \mathbf{1} & S_{12} & S_{13} & S_{14} & S_{15} & S_{16} & S_{17} \\ S_{21} & \mathbf{1} & S_{23} & S_{24} & S_{25} & S_{26} & \mathbf{1} \\ S_{31} & S_{32} & \mathbf{1} & S_{34} & S_{35} & \mathbf{1} & S_{37} \\ S_{41} & S_{42} & S_{43} & \mathbf{1} & \mathbf{1} & S_{46} & S_{47} \\ S_{51} & S_{52} & S_{53} & \mathbf{1} & \mathbf{1} & S_{56} & S_{57} \\ S_{61} & S_{62} & \mathbf{1} & S_{64} & S_{65} & \mathbf{1} & S_{67} \\ S_{71} & \mathbf{1} & S_{73} & S_{74} & S_{75} & S_{76} & \mathbf{1} \end{bmatrix}.$



  $S_{\text{iso}}(A)$ is weakly compatible with $R_{\text{iso}}(A)$. Moreover, if all unspecified entries of $S_{\text{iso}}(A)$ are different from 1, then $S_{\text{iso}}(A)$ is strongly compatible with $R_{\text{iso}}(A)$.

- $S_{\max}(A) := \begin{bmatrix} \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} \\ \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} \\ \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} \\ \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} \\ \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} \\ \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} \\ \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} \end{bmatrix}.$



  $S_{\max}(A)$ is weakly and strongly compatible with $R_{\max}(A)$.

In [LHN06], Leicht *et al.* say that a self-similarity measure is a *structural* self-similarity measure if it is only based on the pattern of edges between nodes. They also say that a self-similarity measure is a *regular* self-similarity measure if similar nodes are connected to other nodes that are themselves similar. These terms vaguely extend the notions of structural and regular equivalences. In [CABVD10], we propose a more rigorous definition of what they call structural self-similarity measure, and introduce the notion of *topological* self-similarity measure. It states that a self-similarity measure is termed topological if the self-similarity scores S_{ij} equals 0 whenever the node i and the node j do not belong to the same connected component. Notice that any weakly or strongly isomorphic self-similarity measure is not a topological self-similarity measure.

Reinforcement Loops

Several node-to-node self-similarity and similarity measures are defined with reinforcement loops (*cf.* Algorithm 1) that let the similarity scores

percolate through the whole network and hopefully converge towards a normalized fixed point of the reinforcement function. The role of the normalizing function is to scale the iterates such that one can compare them at different time steps. Usually, this normalizing function is set to $\text{normalize}(S) := S/\left\|S\right\|_F$.

Algorithm 1 Reinforcement Loop

Require: $S_0 \in \mathbb{R}^{m\times n}$, an initial similarity matrix,
$f : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$, a reinforcement function, and
$\text{normalize} : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$, a normalizing function
$t_{\max} \in \mathbb{N}$, the maximal number of iterations.

for $t = 1, 2, \cdots, t_{\max}$ **do**

$S_t \leftarrow \text{normalize}\left(f\left(S_{t-1}\right)\right)$.

end for

Unless specified, normalize is further set to $\text{normalize}(S) := S/\left\|S\right\|_F$.

Let us now consider that the normalizing function of Algorithm 1 can be written as $\text{normalize}(S) := S/\left\|S\right\|$ for a given matrix norm $\|\cdot\|$. If a matrix S is a fixed point of Algorithm 1, then $\left(\frac{\|f(S)\|}{\|S\|}, S\right)$ is an eigenpair of the following generalized eigenvalue problem

$$\lambda\ S = f\left(S\right)\ .$$

In [Nin08, sec. 2.1.4], Ninove claims that it is easy to prove using Brouwer's fixed point theorem that, if the map f is continuous, there exists at least one solution to that eigenvalue problem. In [Nin08, Th. 2.6], Ninove also characterizes a class of iterated maps of the type

$$\text{normalize}\left(f\left(S\right)\right) := \frac{f\left(S\right)}{\|f(S)\|}\ ,$$

over the non-negative orthant that has a unique fixed point, that is moreover positive, and globally converges to it on the cone. This result is due to Krause [Kra86] who also stated several other similar particularities for concave maps in [Kra86, Kra01]. Finally, in [Nin08, chap. 3], Ninove considers the case of affine iterated maps on the non-negative orthant, *i.e.*

$$\text{vec}(\text{normalize}\left(f\left(S\right)\right)) := \frac{A\ \text{vec}(S) + b}{\|A\ \text{vec}(S) + b\|}\ ,$$

with $A, S, b \geq 0$. Ninove discusses the existence and uniqueness of a fixed point for those iterative maps, and characterised this unique solution for particular A and b as the Perron vector of a rank-one modification of the matrix A satisfying a maximizing property.

3.1 Node-to-Node Similarity Measures

The node-to-node similarity scores between the node i and the node j are essentially (and we hereafter assume that they always are) based on the structure of the networks around the nodes i and j, *e.g.*

> *If the similarity between the neighborhood of the node i and the one of the node j is large, then the similarity between the node i and the node j is large.*

Since relabeling the nodes does not influence the structure of a graph, node-to-node similarity measures are hence invariant under relabeling, which means that, given a node-to-node similarity measure $S(\cdot,\cdot)$, for all adjacency matrices $A \in \{0,1\}^{m\times m}$ and $B \in \{0,1\}^{n\times n}$, and for all permutation matrices $P \in \mathcal{P}(m)$ and $Q \in \mathcal{P}(n)$, we have,

$$S(A,B) = P\ S(P^T A P, Q^T B Q)\ Q^T\ .$$

As a consequence of this property, if the node $i \in N(A)$ is isomorphic to the node $i' \in N(A)$, then their similarity scores to any other node $j \in N(B)$ are equal, *i.e.*

$$[S(A,B)]_{i,j} = [S(A,B)]_{i',j}\ , \quad \text{and} \quad [S(B,A)]_{j,i} = [S(B,A)]_{j,i'}\ .$$

Lemma 3.1 *Let $S(\cdot,\cdot)$ be a node-to-node similarity measure. The node-to-node self-similarity measure $S'(\cdot)$ defined as $S'(A) := S(A,A)$ can not be strongly structural.*

Proof: One can easily build a graph $G(A)$ in which two nodes $i, i' \in N(A)$ are isomorphic, but not structurally equivalent, *e.g.* a cycle graph. Since the node i is structurally equivalent to himself, we have $S(A,A)_{i',i} = S(A,A)_{i,i} = 1$, hence $S'(\cdot)$ is not strongly structural. □

Hence the node-to-node self-similarity measures induced by node-to-node similarity measures ($S'(A) := S(A, A)$) will not be strongly structural. On the other hand, we will further introduce several node-to-node similarity measures whose induced node-to-node self-similarity measures are weakly or strongly isomorphic. The weak isomorphic node-to-node self-similarity measures can usually be computed in polynomial time, conversely to the strong isomorphic node-to-node self-similarity measures. Indeed, mapping the nodes that have a similarity score of 1 in a strong isomorphic similarity measure between the nodes of two isomorphic asymmetric[1] graphs directly solves the graph matching problem (see [For96]) for which we do not know any polynomial algorithm.

In this section, we consider the following node-to-node similarity measures:

- We first introduce basic similarity measures among which the one introduced by Blondel *et al.* in [BGH+04] extends the definition of hub and authority score introduced by Kleinberg in [Kle99], while the others are introduced to help to introduce other similarity measures.

  $S^{\text{Blondel}}(A, B)$ . . . . . p. 33

  $S_\alpha^{\Pi\Sigma\text{-}r}(A, B)$ . . . . . p. 35

  $S_\alpha^{\Sigma\Pi\text{-}r}(A, B)$ . . . . . p. 37

  $S_\alpha^{\text{P}\text{-}r}(A, B)$ . . . . . p. 40

  $S^{\text{Edit}}(A, B)$ . . . . . p. 42

- We further consider similarity measures that are diagonally scaled version of the basic definitions. Notice that they are all at least weakly isomorphic.

  $S^{\text{Blondel}'}(A, B)$ . . . . . p. 45

  $S_\alpha^{\text{Cooper}\text{-}r}(A, B)$ . . . . . p. 46

  $S_\alpha^{\text{Cooper}'\text{-}r}(A, B)$ . . . . . p. 46

[1] A graph is termed *asymmetric* if none of its nodes are isomorphic.

- We finally consider similarity measures that involve computation with the stochastically scaled adjacency matrix. Notice that $S_\alpha^{\text{P-Rank}}(A,B)$ extends the definition of Page-Rank score introduced by Brin and Page in [BP98].

3.1.1 Basic Definitions

Blondel *et al.* Similarity Measure

A first basic node-to-node similarity measure has been introduced by Blondel *et al.* in [BGH+04] in which they recursively require that, given two adjacency matrices $A \in \{0,1\}^{m\times m}$ and $B \in \{0,1\}^{n\times n}$,

> *the similarity score between $i \in N(A)$ and $j \in N(B)$ is large,*
> *if the similarity score between $k \in C(i)$ and $l \in C(j)$ is large, or*
> *if the similarity score between $k' \in P(i)$ and $l' \in P(j)$ is large.*

This leads them to define $S^{\text{Blondel}}(A,B)$ as an extremal point of Algorithm 1 with $S_0^{\text{Blondel}} := \mathbf{1}_{m,n}$, and

$$[f^{\text{Blondel}}(S)]_{i,j} := \left[ASB^T + A^TSB\right]_{i,j} = \sum_{\substack{k\in C(i)\\ l\in C(j)}} S_{kl} + \sum_{\substack{k\in P(i)\\ l\in P(j)}} S_{kl} \ ,$$

whose complexity is of the order of $4\left(|E(A)|\,n + |E(B)|\,m\right)$ operations[2] per evaluation. *E.g.*, let us consider A the adjacency matrix of the graph represented in Figure 2.1. The node-to-node similarity measure S^{Blondel} yields,

$$S^{\text{Blondel}}(A,A) = \begin{bmatrix} 0.28 & 0.1 & 0.17 & 0 & 0 & 0.17 & 0.1 \\ 0.1 & 0.07 & 0.12 & 0.04 & 0.04 & 0.12 & 0.07 \\ 0.17 & 0.12 & 0.31 & 0.13 & 0.13 & 0.31 & 0.12 \\ 0 & 0.04 & 0.13 & 0.14 & 0.14 & 0.13 & 0.04 \\ 0 & 0.04 & 0.13 & 0.14 & 0.14 & 0.13 & 0.04 \\ 0.17 & 0.12 & 0.31 & 0.13 & 0.13 & 0.31 & 0.12 \\ 0.1 & 0.07 & 0.12 & 0.04 & 0.04 & 0.12 & 0.07 \end{bmatrix} .$$

[2] An operation is one of the basic arithmetic operations, *i.e.* $+$, $-$, $\times$ and $\div$

The Blondel *et al.* similarity generalizes the Hubs and Authorities Algorithm, introduced by Kleinberg in [Kle99]. The Hubs and Authorities Algorithm give a hub score and authority score to every node in a graph, based on the following recursive requirement :

- the hub score of a node is large if the authority score of the nodes it points to is large, and
- the authority score of a node is large if the hub score of the nodes pointing to it is large.

Kleinberg uses its Hubs and Authorities Algorithm on the World Wide Web in order to identify web-pages with relevant content (that are supposed to be the authorities) and the web-pages with relevant links (that are supposed to be the hubs). The hub and authority scores are conveniently stored in a hub and authority vector, respectively denoted h and a, whose i^{th} entries are respectively the hub and authority scores associated with the i^{th} node, and are defined as follows.

Hubs and Authorities Algorithm

Input: $A \in \{0,1\}^{m\times m}$, an adjacency matrix, and
$t_{\max} \in \mathbb{N}$, the maximal number of iterations.

Output: $h \in \mathbb{R}^n_{\geq 0}$, the hub vector, and
$a \in \mathbb{R}^n_{\geq 0}$, the authority vector.

$h_i \leftarrow 1$, and $a_i \leftarrow 1$, for all $i \in N(A)$.

for $t = 1, 2, \cdots, t_{\max}$ **do**

$$h'_i \leftarrow \sum_{k\in C(i)} a_k \ , \quad \text{and} \quad a'_i \leftarrow \sum_{k\in P(i)} h_k \ , \quad \forall i \in N(A).$$

$$h \leftarrow \frac{h'}{\|h'\|_F} \ , \quad \text{and} \quad a \leftarrow \frac{a'}{\|a'\|_F} \ .$$

endfor

The hub an authority scores are easily computed from the Blondel *et al.* similarity scores that compare the nodes of $G(A)$ with the nodes of the graph whose adjacency matrix is $B = \left[\begin{smallmatrix} 0 & 1 \\ 0 & 0 \end{smallmatrix}\right]$, or in other words, a graph with a hub node that points towards an authority node, *i.e.*

.

Indeed, the Hubs and Authorities scores can be written as $S^{\text{H-A}} = [h|a]$, which is an extremal point of Algorithm 1 with $S_0^{\text{H-A}} \leftarrow \mathbf{1}_{m,2}$,

$$f^{\text{H-A}}(S) := A\ S\ \left[\begin{smallmatrix} 0 & 1 \\ 0 & 0 \end{smallmatrix}\right]^T + A^T\ S\ \left[\begin{smallmatrix} 0 & 1 \\ 0 & 0 \end{smallmatrix}\right] \ ,$$

and the normalizing function defined as

$$\left[\text{normalize}^{\text{H-A}}(S)\right]_{i,j} := \frac{S_{ij}}{\|S_{i\cdot}\|_F} \ .$$

Note that the scaling of $S^{\text{H-A}}$ and S^{Blondel} are different, though, one can prove that

$$S^{\text{H-A}}(A) = \text{normalize}^{\text{H-A}}\left(S^{\text{Blondel}}\left(A, \left[\begin{smallmatrix} 0 & 1 \\ 0 & 0 \end{smallmatrix}\right]\right)\right) \ .$$

E.g., let us consider A the adjacency matrix of the graph represented in Figure 2.1. The node-to-node similarity measure $S^{\text{H-A}}(A)$ yields,

$$S^{\text{H-A}}(A) = \begin{bmatrix} 0.59 & 0 \\ 0.2 & 0.23 \\ 0.53 & 0.52 \\ 0 & 0.41 \\ 0 & 0.41 \\ 0.53 & 0.52 \\ 0.20 & 0.23 \end{bmatrix} \ .$$

Similarity Measures Based on Walk Comparison

Another basic possibility consists in defining the similarity between $i \in N(A)$ and $j \in N(B)$ by comparing the vectors $[A^r]_{i,\cdot}$ with $[B^r]_{j,\cdot}$, and $[A^r]_{\cdot,i}$ with $[B^r]_{\cdot,j}$, or in other words to compare the number of walks of length r that begin (resp. end) in i with the one that begin (resp. end) in j. We first introduce an unscaled version of $S_\alpha^{\text{Cooper-}r}$ [3], and further introduce two new measures that improve that unscaled similarity measure.

[3] $S_\alpha^{\text{Cooper-}r}$ is a similarity measure that we will further introduce in Section 3.1.2

Let us first introduce $S_\alpha^{\Pi\Sigma\text{-}r}$ (the unscaled version of $S_\alpha^{\text{Cooper-}r}$) that defines the similarity score between the node i and the node j as the scalar product between the two vectors $\mathbf{x}_i(A)$ and $\mathbf{x}_j(B)$, *i.e.*

$$[S_\alpha^{\Pi\Sigma\text{-}r}(A,B)]_{i,j} := \langle \mathbf{x}_i(A), \mathbf{x}_j(B) \rangle_F$$

where $\mathbf{x}_i(A)$ is the i^{th} row of the matrix with $|N(A)|$ rows and $2r$ columns defined as $\left[\ X_{\alpha,r}(A) \ \middle|\ X_{\alpha,r}\left(A^T\right)\ \right]$ with

$$X_{\alpha,r}(A) := \begin{bmatrix} \alpha A \mathbf{1}_{m,1} & \alpha^2 A^2 \mathbf{1}_{m,1} & \cdots & \alpha^r A^r \mathbf{1}_{m,1} \end{bmatrix} ,$$

with $\alpha > 0$ a damping factor. Note that $X_{\alpha,r}(A)$ is clearly directly linked with $\left[A^1\right]_{i,\cdot}$, $\left[A^2\right]_{i,\cdot}$, $\ldots$, $[A^r]_{i,\cdot}$, since

$$[A^\ell \mathbf{1}_{m,1}]_i = \sum_k [A^\ell]_{i,k} .$$

The vector $\mathbf{x}_i(A)$ is then a row vector with $2r$ columns,

- whose ℓ^{th} entry in the r first ones contains the number of walks of length ℓ with node i as origin, and
- whose ℓ^{th} entry in the r last ones contains the number of walks of length ℓ with node i as termination.

Let us give an interpretation of this similarity measure. The more the neighborhood of the node $i \in N(A)$ is similar to the one of the node $j \in N(B)$, the more the number of walks of length ℓ with node i as origin (resp. termination) is close to the one with node j as origin (resp. termination), the more the vector $\mathbf{x}_i(A)$ is aligned with $\mathbf{x}_j(B)$, the larger the scalar product between $\mathbf{x}_i(A)$ and $\mathbf{x}_j(B)$, and the larger $[S_\alpha^{\Pi\Sigma\text{-}r}(A,B)]_{i,j}$. The complexity of computing $X_{\alpha,r}(A)$ is of the order of $2rm\,|E(A)|$ operations, and thus the complexity of computing $S_\alpha^{\Pi\Sigma\text{-}r}$ is of the order of $2r(mn + |E(A)| + |E(B)|)$ operations. This computation is cheap and simple, but much information gets lost when computing the product between the powers of A and $\mathbf{1}_{m,1}$. Notice that $S_\alpha^{\Pi\Sigma\text{-}r}(A,B)$ can be bounded as follows

$$\begin{aligned} [S_\alpha^{\Pi\Sigma\text{-}r}(A,B)]_{i,j} &:= \langle \mathbf{x}_i(A), \mathbf{x}_j(B) \rangle_F \\ &\leq \sum_{k=1}^{r} \alpha^k \left\| A^k \right\|_\infty \beta^k \left\| B^k \right\|_\infty + \alpha^k \left\| \left(A^T\right)^k \right\|_\infty \beta^k \left\| \left(B^T\right)^k \right\|_\infty . \end{aligned}$$

We know form [Arv02, Eq. 1.12], [Arv02, Th. 1.7.3] or [Gel41] that

$$\rho(A) \leq \left\|A^k\right\|_\infty^{1/k} \quad , \quad \text{and} \quad \rho(A) = \lim_{k\to\infty} \left\|A^k\right\|_\infty^{1/k} \quad .$$

And eventually, if r tends to infinity, we can bound the similarity matrix $S_\alpha^{\Pi\Sigma\text{-}r}(A,B)$ by a geometric series that converges linearly with rate $2\alpha\beta\rho(A)\rho(B)$ if

$$\alpha\beta < \frac{1}{2\rho(A)\rho(B)} \ .$$

Let us now introduce a first new more informative measure that defines the similarity score between the node i and the node j as the sum for k equals 1 to r of the maximal scalar products between $\left[\alpha^k A^k\right]_{i,\cdot}$ and $\left[\alpha^k B^k\right]_{j,\cdot}$ among all the permutations of their entries, plus the corresponding term for $\left[\alpha^k A^k\right]_{\cdot,i}$ and $\left[\alpha^k B^k\right]_{\cdot,j}$, *i.e.*

$$\begin{aligned}[S_\alpha^{\Sigma\Pi\text{-}r}(A,B)]_{i,j} := \sum_{k=1}^{r} & \max_{\substack{P\in\mathcal{P}(m)\\ Q\in\mathcal{P}(n)}} \left\langle \mathbf{x}_{i,k}(A)P\ I_{m,n}, \mathbf{x}_{j,k}(B)Q\right\rangle_F \\ & + \max_{\substack{P\in\mathcal{P}(m)\\ Q\in\mathcal{P}(n)}} \left\langle \mathbf{x}_{i,k}\left(A^T\right)P\ I_{m,n}, \mathbf{x}_{j,k}\left(B^T\right)Q\right\rangle_F \ ,\end{aligned}$$

where $\mathbf{x}_{i,k}(A)$ is the i^{th} row of the matrix $\alpha^k A^k$ with $\alpha > 0$ a damping factor. Note that the vector $\mathbf{x}_i(A)$ in the definition of $S_\alpha^{\Pi\Sigma\text{-}r}$ is directly linked to the vectors $\mathbf{x}_{i,k}(A)$ and $\mathbf{x}_{i,k}\left(A^T\right)$ since

$$\mathbf{x}_i(A) = \left[\ \cdots \quad \sum_j \left[\mathbf{x}_{i,k}(A)\right]_j \quad \cdots \ \middle|\ \cdots \quad \sum_j \left[\mathbf{x}_{i,k}\left(A^T\right)\right]_j \quad \cdots\ \right] \ .$$

The interpretation of $S_\alpha^{\Sigma\Pi\text{-}r}$ is similar to the one of $S_\alpha^{\Pi\Sigma\text{-}r}$, *i.e.* the more the neighborhood of the node $i \in N(A)$ is similar to the one of the node $j \in N(B)$, the more one could align the vector $\mathbf{x}_{i,k}(A)$ with the vector $\mathbf{x}_{j,k}(B)$ by permuting their entries, the larger their maximal scalar product among all the permutations of their entries, and the larger $[S_\alpha^{\Sigma\Pi\text{-}r}(A,B)]_{i,j}$. The difference between $S_\alpha^{\Sigma\Pi\text{-}r}$ and $S_\alpha^{\Pi\Sigma\text{-}r}$ is that $S_\alpha^{\Sigma\Pi\text{-}r}$ compares the vector of the number of walks of length ℓ between the node i and the node k, for all $k \in N(A)$ to the one of the number of walks of length ℓ between the node j and the node l, for all $l \in N(B)$, whereas $S_\alpha^{\Pi\Sigma\text{-}r}$ compares the sum of all the entries of those vectors which

dismisses valuable information. Notice that $S_\alpha^{\Sigma\Pi\text{-}r}(A,B)$ can be bounded as follows

$$\begin{aligned}
&[S_\alpha^{\Sigma\Pi\text{-}r}(A,B)]_{i,j} \\
&\quad\leq \sum_{k=1}^{r} \|\mathbf{x}_{i,k}(A)\|_F \, \|\mathbf{x}_{j,k}(B)\|_F + \left\|\mathbf{x}_{i,k}\left(A^T\right)\right\|_F \left\|\mathbf{x}_{j,k}\left(B^T\right)\right\|_F \\
&\quad\leq \sum_{k=1}^{r} \alpha^k \left\|A^k\right\|_F \beta^k \left\|B^k\right\|_F + \alpha^k \left\|\left(A^T\right)^k\right\|_F \beta^k \left\|\left(B^T\right)^k\right\|_F
\end{aligned}$$

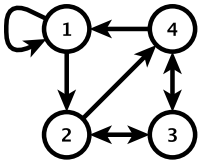
We know form [Arv02, Eq. 1.12], [Arv02, Th. 1.7.3] or [Gel41] that

$$\rho(A) \leq \left\|A^k\right\|_F^{1/k} \quad , \quad \text{and} \quad \rho(A) = \lim_{k\to\infty} \left\|A^k\right\|_F^{1/k} \quad .$$

And eventually, if r tends to infinity, we can bound the similarity matrix $S_\alpha^{\Sigma\Pi\text{-}r}(A,B)$ by a geometric series that converges linearly with rate $2\alpha\beta\rho(A)\rho(B)$ if

$$\alpha\beta < \frac{1}{2\rho(A)\rho(B)} \ .$$

E.g., let A be the adjacency matrix of the graph whose graphical representation is the following,



.

Let us now compute $\left[S_{\alpha=1}^{\Pi\Sigma\text{-}3}(A,A)\right]_{i,j}$ and $\left[S_{\alpha=1}^{\Sigma\Pi\text{-}3}(A,A)\right]_{i,j}$ for $i,j \in \{1,2\}$. Since

$$A = \begin{bmatrix} 1&1&0&0 \\ 0&0&1&1 \\ 0&1&0&1 \\ 1&0&1&0 \end{bmatrix} \quad , \quad A^2 = \begin{bmatrix} 1&1&1&1 \\ 1&1&1&1 \\ 1&0&2&1 \\ 1&2&0&1 \end{bmatrix} \quad , \quad A^3 = \begin{bmatrix} 2&2&2&2 \\ 2&2&2&2 \\ 2&3&1&2 \\ 2&1&3&2 \end{bmatrix} \ ,$$

and $\alpha = 1$, we have

$$\mathbf{x}_1(A) = \left[\begin{array}{cccc|cccc} 2 & 4 & 8 & 2 & 4 & 8 \end{array}\right] ,$$

$$\mathbf{x}_{1,1}(A) = \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix} ,$$
$$\mathbf{x}_{1,2}(A) = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} ,$$
$$\mathbf{x}_{1,3}(A) = \begin{bmatrix} 2 & 2 & 2 & 2 \end{bmatrix} ,$$

$$\mathbf{x}_{1,1}(A^T) = \begin{bmatrix} 1 & 0 & 0 & 1 \end{bmatrix} ,$$
$$\mathbf{x}_{1,2}(A^T) = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} ,$$
$$\mathbf{x}_{1,3}(A^T) = \begin{bmatrix} 2 & 2 & 2 & 2 \end{bmatrix} ,$$

$$\mathbf{x}_2(A) = \left[\begin{array}{cccc|cccc} 2 & 4 & 8 & 2 & 4 & 8 \end{array}\right] ,$$

$$\mathbf{x}_{2,1}(A) = \begin{bmatrix} 0 & 0 & 1 & 1 \end{bmatrix} ,$$
$$\mathbf{x}_{2,2}(A) = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} ,$$
$$\mathbf{x}_{2,3}(A) = \begin{bmatrix} 2 & 2 & 2 & 2 \end{bmatrix} ,$$

$$\mathbf{x}_{2,1}(A^T) = \begin{bmatrix} 1 & 0 & 1 & 0 \end{bmatrix} ,$$
$$\mathbf{x}_{2,2}(A^T) = \begin{bmatrix} 1 & 1 & 0 & 2 \end{bmatrix} , \text{ and}$$
$$\mathbf{x}_{2,3}(A^T) = \begin{bmatrix} 2 & 2 & 3 & 1 \end{bmatrix} .$$

And finally, we have $\left[S_{\alpha=1}^{\Pi\Sigma\text{-}3}(A,A)\right]_{i,j} = \langle \mathbf{x}_i(A), \mathbf{x}_j(A) \rangle_F = 168$, for all $i, j \in \{1, 2\}$, whereas

$$\left[S_{\alpha=1}^{\Sigma\Pi\text{-}3}(A,A)\right]_{1,1} = \left[S_{\alpha=1}^{\Sigma\Pi\text{-}3}(A,A)\right]_{1,2} = \left[S_{\alpha=1}^{\Sigma\Pi\text{-}3}(A,A)\right]_{2,1}$$
$$= 2 + 4 + 16 + 2 + 4 + 16 = 24 , \text{ and}$$

$$\left[S_{\alpha=1}^{\Sigma\Pi\text{-}3}(A,A)\right]_{2,2} = 2 + 4 + 16 + 2 + 6 + 18 = 28 .$$

Notice that $S_{\alpha=1}^{\Sigma\Pi\text{-}3}$ better discriminates the nodes than $S_{\alpha=1}^{\Pi\Sigma\text{-}3}$.

We now show that computing the optimal P and Q in the definition of $S_\alpha^{\Pi\Sigma\text{-}r}$ is equivalent to a sorting problem.

Proposition 3.2 *Given $u \in \mathbb{R}_{\geq 0}^m$ and $v \in \mathbb{R}_{\geq 0}^n$. We have*

$$\max_{\substack{P \in \mathcal{P}(m) \\ Q \in \mathcal{P}(n)}} \langle uP\ I_{m,n}, vQ \rangle_F = \langle \mathrm{sort}(u)\ I_{m,n}, \mathrm{sort}(v) \rangle_F$$

Proof: This proposition is a special case of Lemma 4.2 in [FNVD08] with $k = \min(m, n)$ and $u_i, v_j \geq 0$ for all i and j. □

Computing each optimal P in the definition of $S_\alpha^{\Sigma\Pi\text{-}r}$ only requires to sort $\mathbf{x}_{i,k}(A)$ and $\mathbf{x}_{j,k}(B)$, whose complexity are respectively of the order of $m \log(m)$ and $n \log(n)$ operations [CLRS01, II. 6.]. The complexity of computing the powers of A and B are respectively of the order of $m\,|E(A)|$ and $n\,|E(B)|$ operations. And, eventually, the complexity of computing $S_\alpha^{\Sigma\Pi\text{-}r}$ is of the order of

$$2rmn[m \log(m) + n \log(n)] + rm\,|E(A)| + rn\,|E(B)|$$

operations. As we will further see in Section 3.1.2, this similarity measure is more informative than $S_\alpha^{\Pi\Sigma\text{-}r}$.

Let us finally introduce an even more informative new measure that defines the similarity score between the node i and the node j as the maximal scalar product between two matrices $X_i(A)$ and $X_j(B)$ among all the permutations of their columns, *i.e.*

$$[S_\alpha^{\text{P-}r}(A,B)]_{i,j} := \max_{\substack{P\in\mathcal{P}(m)\\ Q\in\mathcal{P}(n)}} \langle X_i(A)P\ I_{m,n}, X_j(B)Q\rangle_F \ ,$$

where $X_i(A) := \begin{bmatrix} Y_i(A) \\ Y_i(A^T) \end{bmatrix}$ is a matrix with $2(r+1)$ rows and $|N(A)|$ columns defined using

$$Y_i(A) := \begin{bmatrix} [\alpha^0 A^0]_{i,\cdot} \\ [\alpha^1 A^1]_{i,\cdot} \\ \vdots \\ [\alpha^r A^r]_{i,\cdot} \end{bmatrix} \ .$$

The interpretation of $S_\alpha^{\text{P-}r}$ is similar to the one of $S_\alpha^{\Sigma\Pi\text{-}r}$, *i.e.* the more the neighborhood of the node $i \in N(A)$ is similar to the one of the node $j \in N(B)$, the more one could align the matrix $X_i(A)$ with the matrix $X_j(B)$ by permuting their columns, the larger their maximal scalar product among all the permutations of their columns, and the larger $[S_\alpha^{\text{P-}r}(A,B)]_{i,j}$. The difference between $S_\alpha^{\text{P-}r}$ and $S_\alpha^{\Sigma\Pi\text{-}r}$ is that $S_\alpha^{\text{P-}r}$ compares the numbers of walks of length 0 to r between the node i and the node k, for all $k \in N(A)$ to the ones of the numbers of walks of length 0 to r between the node j and the node ℓ, for all $\ell \in N(B)$, whereas $S_\alpha^{\Sigma\Pi\text{-}r}$ compares those respective numbers of walks for all length from 1 to r separately, which dismisses valuable information. Notice that $S_\alpha^{\text{P-}r}(A,B)$ can be bounded as follows

$$\begin{aligned}
[S_\alpha^{\text{P-}r}(A,B)]_{i,j} &:= \max_{\substack{P\in\mathcal{P}(m)\\ Q\in\mathcal{P}(n)}} \left\langle PI_{m,n}Q^T, X_i(A)^T X_j(B)\right\rangle_F \\
&\leq \left\|X_i(A)^T X_j(B)\right\|_{\bullet 1} \\
&\leq \sum_{k=1}^{r} \alpha^k \left\|A^k\right\|_{\bullet 1} \beta^k \left\|B^k\right\|_{\bullet 1} + \alpha^k \left\|\left(A^T\right)^k\right\|_{\bullet 1} \beta^k \left\|\left(B^T\right)^k\right\|_{\bullet 1} \ ,
\end{aligned}$$

where $\|A\|_{\bullet 1} := \sum_{ij} |A_{ij}|$ denotes the entrywise matrix norm with exponent 1 of the matrix A. We know form [Arv02, Eq. 1.12], [Arv02, Th. 1.7.3] or [Gel41] that

$$\rho(A) \leq \left\| A^k \right\|_{\bullet 1}^{1/k} , \quad \text{and} \quad \rho(A) = \lim_{k \to \infty} \left\| A^k \right\|_{\bullet 1}^{1/k} .$$

And eventually, if r tends to infinity, we can bound the similarity matrix $S_\alpha^{\text{P-}r}(A, B)$ by a geometric series that converges linearly with rate $2\alpha\beta\rho(A)\rho(B)$ if

$$\alpha\beta < \frac{1}{2\rho(A)\rho(B)} .$$

We now show that computing the optimal P and Q in the definition of $S_\alpha^{\Sigma\Pi\text{-}r}$ is equivalent to solving an assignment problem.

Proposition 3.3 *Given $U \in \mathbb{R}^{r \times m}$ and $V \in \mathbb{R}^{r \times n}$. Then, solving*

$$\max_{\substack{P \in \mathcal{P}(m) \\ Q \in \mathcal{P}(n)}} \langle UP\ I_{m,n}, VQ \rangle_F$$

is equivalent to finding the solution of an assignment problem (see [Mun57] for details about the assignment problem).

Proof: let $\underline{m} := \min(m, n)$ and $\overline{m} := \max(m, n)$. The scalar product $\langle UP\ I_{m,n}, VQ \rangle_F$ can always be rewritten as a scalar product between a matrix $M \in \mathbb{R}^{\overline{m} \times \underline{m}}$, and a selection matrix $P' \in \mathcal{P}(\overline{m}, \underline{m})$, *i.e.*

$$\langle UP\ I_{m,n}, VQ \rangle_F = \langle M, P' \rangle_F ,$$

where

$$\text{if } m \geq n\text{: } M := U^T V \text{ and } P' := P I_{m,n} Q^T, \text{ and}$$
$$\text{if } m < n\text{: } M := V^T U \text{ and } P' := Q I_{n,m} P^T.$$

We can now rewrite the maximization problem as

$$\max_{P' \in \mathcal{P}(\underline{m}, \overline{m})} \langle M, P' \rangle_F = \max_{\left[P'\ P'_\perp\right] \in \mathcal{P}(\overline{m})} \left\langle \left[M\ \mathbf{0}_{\overline{m}, \overline{m}-\underline{m}}\right], \left[P'\ P'_\perp\right] \right\rangle_F ,$$

which is an assignment problem with $\left[M\ \mathbf{0}_{\overline{m}, \overline{m}-\underline{m}}\right]$ as cost matrix. □

Hence, computing the optimal P and Q in the definition of $S_\alpha^{\Sigma\Pi\text{-}r}$ is

equivalent to solving an assignment problem of order $max(m,n)$, whose complexity is of the order of $[max(m,n)]^3$ operations. And, eventually, the complexity of computing $S_\alpha^{\text{P-}r}$ is of the order of

$$mn\ \left([max(m,n)]^3 + 2(r+1)\right) + rm\,|E(A)| + rn\,|E(B)|$$

operations. *E.g.*, let A be the adjacency matrix of the graph presented on page 38, and let us now compute $\left[S_{\alpha=1}^{\text{P-3}}(A,A)\right]_{i,j}$ for $i,j \in \{1,2\}$. We have, $\alpha = 1$, $r = 3$,

$$X_1(A) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \end{bmatrix}, \quad \text{and} \quad X_2(A) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 2 \\ 2 & 2 & 3 & 1 \end{bmatrix}.$$

The cost matrix of the non-diagonal terms is

$$[X_1(A)]^T X_2(A) = \left([X_2(A)]^T X_1(A)\right)^T = \begin{bmatrix} 11 & 10 & 10 & \mathbf{11} \\ \mathbf{12} & 10 & 10 & 10 \\ 13 & 12 & \mathbf{11} & 12 \\ 10 & \mathbf{10} & 9 & 9 \end{bmatrix},$$

in which we highlighted a maximal assignment, that yields

$$\left[S_{\alpha=1}^{\text{P-3}}(A,A)\right]_{1,2} = \left[S_{\alpha=1}^{\text{P-3}}(A,A)\right]_{2,1} = 12 + 10 + 11 + 11 = 44\ ,$$

whereas

$$\left[S_{\alpha=1}^{\text{P-3}}(A,A)\right]_{1,1} = \|X_1(A)\|_F^2 = 14 \cdot 1^2 + 8 \cdot 2^2 = 46\ ,\ \text{and}$$
$$\left[S_{\alpha=1}^{\text{P-3}}(A,A)\right]_{2,2} = \|X_2(A)\|_F^2 = 13 \cdot 1^2 + 7 \cdot 2^2 + 3^2 = 50\ .$$

Similarity Measures Based on Graph Superposition

A last basic definition consists in defining the similarity between i and j as the maximal number of edges of $G(A)$ one could match with the ones of $G(B)$ when i is mapped on j, *i.e.*

$$[S^{\text{Edit}}(A,B)]_{i,j} = \max_{\substack{P \in \mathcal{P}(m,n), \\ \text{s.t. } P_{ij}=1}} \left\langle A, PBP^T \right\rangle_F\ . \tag{3.1}$$

where $\mathcal{P}(m,n)$ is the set of selection matrices defined as

$$\mathcal{P}(m,n) := \{PI_{m,n}Q \text{ s.t. } P \in \mathcal{P}(m),\ Q \in \mathcal{P}(n)\}\ .$$

The scalar product in Equation (3.1) counts the number of edges in A "aligned" with the ones in PBP^T. There are $\frac{\max(m,n)!}{\max(m-n,n-m)!}$ selection matrices in $\mathcal{P}(m,n)$, and there is no known algorithm that finds the optimal P in the definition of S^{Edit} in polynomial time. One can further observe that the maximization of the scalar product in Equation (3.1) is equivalent to the minimization of a norm, *i.e.*

$$\begin{aligned}
&\underset{\substack{P\in\mathcal{P}(m,n),\\ \text{s.t. } P_{ij}=1}}{\operatorname{argmax}} \left\langle A, PBP^T\right\rangle_F \qquad (3.2)\\
&\quad = \underset{\substack{P\in\mathcal{P}(m,n),\\ \text{s.t. } P_{ij}=1}}{\operatorname{argmin}} \langle A, A\rangle_F + \left\langle PBP^T, PBP^T\right\rangle_F - 2\left\langle A, PBP^T\right\rangle_F\\
&\quad = \underset{\substack{P\in\mathcal{P}(m,n),\\ \text{s.t. } P_{ij}=1}}{\operatorname{argmin}} \left\|A - PBP^T\right\|_F^2\ .
\end{aligned}$$

The norm counts the number of edges to insert or delete in order to make $G(A)$ isomorphic to $G(B)$ which is directly related to the so called *graph edit distance* [Bun97, ZTW+09].

Definition 3.3 (Edit Operation [ZTW+09, Sec. 2.1]) *An edit operation on a graph $G(A)$ is an insertion or deletion of a node or edge. A node can be deleted only if no edge is connected to the node.*

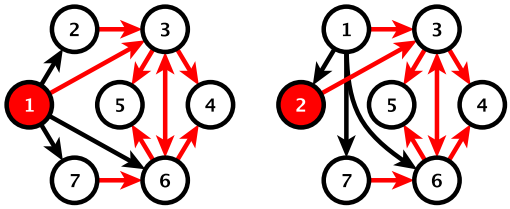
Definition 3.4 (Graph Edit Distance [ZTW+09, Def. 2.1]) *The graph edit distance (GED) between $G(A)$ and $G(B)$, denoted $d(G(A), G(B))$, is the minimal number of edit operations needed to make $G(A)$ isomorphic to $G(B)$.*

One can rewrite the graph edit distance as

$$d(G(A),G(B)) = \min_{P\in\mathcal{P}(m,n)} \left\|A - PBP^T\right\|_F^2 + |m-n|\,, \qquad (3.3)$$

where $|m-n|$ accounts for the number of nodes to insert or delete, and the minimization accounts for the number of edges to insert or delete.

Finding the optimal P is known to be a hard problem. *E.g.*, let us consider A the adjacency matrix of the graph represented in Figure 2.1, and compute the self-similarity matrix $S^{\text{Edit}}(A, A)$. In particular, for the similarity score between the node 1 and the node 2, one can rearrange the nodes of $G(A)$ respectively around the node 1 and around the node 2 as follows



and observe that this optimal arrangement aligns 9 edges.

$$S^{\text{Edit}}(A, A) = \begin{bmatrix} 12 & 9 & 6 & 6 & 6 & 6 & 9 \\ 9 & 12 & 8 & 8 & 8 & 8 & 12 \\ 6 & 8 & 12 & 7 & 7 & 12 & 8 \\ 6 & 8 & 7 & 12 & 12 & 7 & 8 \\ 6 & 8 & 7 & 12 & 12 & 7 & 8 \\ 6 & 8 & 12 & 7 & 7 & 12 & 8 \\ 9 & 12 & 8 & 8 & 8 & 8 & 12 \end{bmatrix}$$

Notice that, for any adjacency matrix A, the diagonal entries of $S^{\text{Edit}}(A, A)$ are all equal to the maximal entry $|E(A)|$.

The node-to-node similarity measures S^{Blondel}, $S^{\Pi\Sigma\text{-}r}$, $S^{\Sigma\Pi\text{-}r}$, $S^{\text{P}\text{-}r}$, and S^{Edit} tend to give higher similarity scores to highly connected nodes. As a result of this, when considering a node-to-node self-similarity measure induced by a node-to-node similarity measure, two distinct nodes could be more self-similar than one node is self-similar to itself. *E.g.*, in the earlier example illustrating the definition of S^{Blondel}, we had

$$[S^{\text{Blondel}}(A, A)]_{1,3} = 0.17 > 0.07 = [S^{\text{Blondel}}(A, A)]_{2,2} \ .$$

This effect is commonly avoided by weighting the definitions.

1. **The diagonal scalings.** The similarity score between the node $i \in N(A)$ and the node $j \in N(B)$ is scaled by a function that depends of the self-similarity of the node i, and the one of the node j, *i.e.*

$$[S_w]_{i,j} = \frac{[S(A, B)]_{i,j}}{w([S(A, A)]_{i,i}, [S(B, B)]_{j,j})} \ .$$

The diagonal scalings are hereafter studied in Section 3.1.2.

2. **The stochastic scalings.** The out-edges or in-edges of a node are (usually uniformly) weighted so that their sum is equal to 1. In other words, the similarity scores are either computed by replacing A and B by their so called row-stochastic scalings or column-stochastic scalings, *i.e.*

$$[\,\mathrm{rss}(A)\,]_{i,j} := \begin{cases} \dfrac{A_{ij}}{\sum_j A_{ij}}, \text{ if } \sum_j A_{ij} \neq 0 \\ 0, \text{ otherwise,} \end{cases} \tag{3.4}$$

or

$$[\,\mathrm{css}(A)\,]_{i,j} := \begin{cases} \dfrac{A_{ij}}{\sum_i A_{ij}}, \text{ if } \sum_i A_{ij} \neq 0 \\ 0, \text{ otherwise,} \end{cases} \tag{3.5}$$

or computed based on a row-stochastic scaling or column-stochastic scaling of the adjacency matrix of the graph $G(A \otimes B)$ [4] where $\otimes$ denotes the Kronecker product. The stochastic scalings are hereafter studied in Section 3.1.3.

3.1.2 Diagonal Scalings

As mentioned in the introduction of Section 3.1.1, the node-to-node similarity measures $S^{\Pi\Sigma\text{-}r}$, $S^{\Sigma\Pi\text{-}r}$, and $S^{\mathrm{P}\text{-}r}$ tend to give higher similarity scores to highly connected nodes. As a result of this, when considering a node-to-node self-similarity measure induced by a node-to-node similarity measure, two distinct nodes could be more self-similar than one node is self-similar to itself. This effect is commonly avoided by scaling the similarity score between the node $i \in N(A)$ and the node $j \in N(B)$ by a function that depends of the self-similarity of the node i, and the one of the node j. *E.g.*, one can define

$$\left[S^{\text{Blondel}'}(A,B)\right]_{i,j} := \frac{[S^{\text{Blondel}}(A,B)]_{i,j}}{\max\left([S^{\text{Blondel}}(A,A)]_{i,i}, [S^{\text{Blondel}}(B,B)]_{j,j}\right)} \tag{3.6}$$

which yields

$$S^{\text{Blondel}'}(A,A) = \begin{bmatrix} 1 & 0.34 & 0.55 & 0 & 0 & 0.55 & 0.34 \\ 0.34 & 1 & 0.39 & 0.27 & 0.27 & 0.39 & 1 \\ 0.55 & 0.39 & 1 & 0.42 & 0.42 & 1 & 0.39 \\ 0 & 0.27 & 0.42 & 1 & 1 & 0.42 & 0.27 \\ 0 & 0.27 & 0.42 & 1 & 1 & 0.42 & 0.27 \\ 0.55 & 0.39 & 1 & 0.42 & 0.42 & 1 & 0.39 \\ 0.34 & 1 & 0.39 & 0.27 & 0.27 & 0.39 & 1 \end{bmatrix},$$

[4] $G(A \otimes B)$ is called the graph product of $G(A)$ and $G(B)$.

where A is the adjacency matrix of the graph of Figure 2.1. Notice that, as mentioned in [BGH$^+$04, Th. 7], the self-similarity matrices associated to the similarity measure of Blondel *et al.* are positive semidefinite, and hence, in a given graph, no distinct nodes are more self-similar than any node is self-similar to itself.

In [CB10], Cooper *et al.* define the similarity score between the node $i \in N(A)$ and the node $j \in N(B)$ as the cosine of the angle between the two vectors $\mathbf{x}_i(A)$ and $\mathbf{x}_j(B)$, *i.e.*

$$[S_\alpha^{\text{Cooper-}r}(A,B)]_{i,j} := \frac{\langle \mathbf{x}_i(A), \mathbf{x}_j(B) \rangle_F}{\|\mathbf{x}_i(A)\|_F \, \|\mathbf{x}_j(B)\|_F}$$

where $\mathbf{x}_i(A)$ is the i^{th} row of the matrix with $|N(A)|$ rows and $2r$ columns defined as $\left[\; X_{\alpha,r}(A) \;\middle|\; X_{\alpha,r}\left(A^T\right) \; \right]$ with

$$X_{\alpha,r}(A) := \begin{bmatrix} \alpha A\mathbf{1} & \alpha^2 A^2\mathbf{1} & \cdots & \alpha^r A^r\mathbf{1} \end{bmatrix} ,$$

with α a damping factor. This similarity measure can be rewritten as a scaling of $S^{\Pi\Sigma\text{-}k}(A,B)$ by a geometric mean of the diagonal entries, *i.e.*

$$[S_\alpha^{\text{Cooper-}r}(A,B)]_{i,j} = \frac{[S_\alpha^{\Pi\Sigma\text{-}r}(A,B)]_{i,j}}{\sqrt{[S_\alpha^{\Pi\Sigma\text{-}r}(A,A)]_{i,i} \cdot [S_\alpha^{\Pi\Sigma\text{-}r}(B,B)]_{j,j}}} . \tag{3.7}$$

A disadvantage of this approach is that if A and B are regular (*i.e.* all nodes have the same in and out degree), then all the entries of the similarity matrices $S_\alpha^{\text{Cooper-}r}(A,B)$ and $S^{\text{Blondel}'}(A,B)$ are equal to 1. We now introduce two new modified versions of $S_\alpha^{\text{Cooper-}r}(A,B)$ that better discriminate the nodes of regular graphs.

The first new similarity measure, denoted $S_\alpha^{\text{Cooper}'\text{-}r}$, is obtained by replacing the basic similarity measure $S^{\Pi\Sigma\text{-}k}$ by the other basic similarity measure $S^{\Sigma\Pi\text{-}k}$ in $S_\alpha^{\text{Cooper-}r}$, *i.e.*

$$\left[S_\alpha^{\text{Cooper}'\text{-}r}(A,B)\right]_{i,j} := \frac{[S_\alpha^{\Sigma\Pi\text{-}r}(A,B)]_{i,j}}{\sqrt{[S_\alpha^{\Sigma\Pi\text{-}r}(A,A)]_{i,i} \cdot [S_\alpha^{\Sigma\Pi\text{-}r}(B,B)]_{j,j}}} . \tag{3.8}$$

The second new similarity measure, denoted $S_\alpha^{\text{Cooper}''\text{-}r}$, is defined as a scaling of the basic similarity measures $S_\alpha^{\text{P-}r}(A,B)$ by a geometric mean of the corresponding diagonal entries, *i.e.*

$$\left[S_\alpha^{\text{Cooper}''\text{-}r}(A,B)\right]_{i,j} := \frac{[S_\alpha^{\text{P-}r}(A,B)]_{i,j}}{\sqrt{[S_\alpha^{\text{P-}r}(A,A)]_{i,i} \cdot [S_\alpha^{\text{P-}r}(B,B)]_{j,j}}} , \tag{3.9}$$

which can be rewritten as

$$\left[S_\alpha^{\text{Cooper}''\text{-}r}\right]_{i,j} := \frac{\max_{\substack{P\in\mathcal{P}(m)\\ Q\in\mathcal{P}(n)}} \left\langle X_i(A)P\ I_{m,n}, X_j(B)Q\right\rangle_F}{\|X_i(A)\|_F \|X_j(B)\|_F}$$

where $X_i(A) := \begin{bmatrix} Y_i(A) \\ Y_i\left(A^T\right) \end{bmatrix}$ is a matrix with $2(r+1)$ rows and $N(A)$ columns defined with

$$Y_i(A) := \begin{bmatrix} \left[\alpha^0 A^0\right]_{i,\cdot} \\ \left[\alpha^1 A^1\right]_{i,\cdot} \\ \vdots \\ \left[\alpha^r A^r\right]_{i,\cdot} \end{bmatrix} .$$

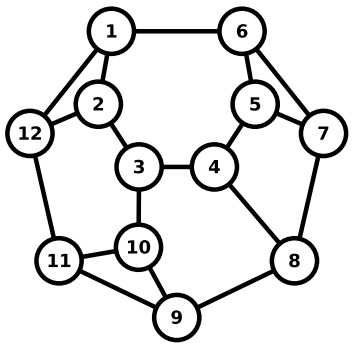


Figure 3.1: The Frucht graph is the smallest undirected simple graph that is asymmetric and cubic. A graph is termed simple *if, for each node i, the pair (i,i) does not belong to the set of edges and, for all nodes i,j, the pair (i,j) appears at most once in the set of edges,* asymmetric *if none of its nodes are isomorphic, and* cubic *if the in- and out-degrees of its nodes are* 3.

E.g., let us compare some of these similarity measures for the Frucht graph represented in Figure 3.1, and with adjacency matrix A_F. All of its nodes are somehow similar since they all have three neighbors, though none of them are isomorphic. This graph is an interesting benchmark since it will allows us to see how well the different induced node-to-node self-similarity measures discriminate its non-isomorphic nodes. The node-to-node similarity measure $S_\alpha^{\text{Cooper-}r}$, though relevant for the graphs

considered in [CB10], and the node-to-node similarity measure $S^{\text{Blondel}'}$ yields,

$$S_\alpha^{\text{Cooper-}r}(A_F, A_F) = S^{\text{Blondel}'}(A_F, A_F) = \mathbf{1}_{12,12} \ , \quad \forall r,$$

whereas $S_\alpha^{\text{Cooper}'\text{-}r}$ and $S_\alpha^{\text{Cooper}''\text{-}r}$ yield

$$S_{\alpha=1/3}^{\text{Cooper}'\text{-}6}(A_F, A_F) = \begin{bmatrix} 1 & 0.99 & 0.99 & 0.97 & 0.97 & 0.99 & 0.97 & 0.97 & 0.99 & 0.99 & 0.99 & 0.99 \\ 0.99 & 1 & 0.99 & 0.97 & 0.97 & 0.99 & 0.97 & 0.97 & 0.99 & 0.99 & 0.99 & 0.99 \\ 0.99 & 0.99 & 1 & 0.97 & 0.97 & 0.99 & 0.97 & 0.97 & 0.99 & 0.99 & 0.99 & 0.99 \\ 0.97 & 0.97 & 0.97 & 1 & 0.99 & 0.97 & 0.99 & 0.99 & 0.97 & 0.97 & 0.97 & 0.97 \\ 0.97 & 0.97 & 0.97 & 0.99 & 1 & 0.97 & 0.99 & 0.99 & 0.97 & 0.97 & 0.97 & 0.97 \\ 0.99 & 0.99 & 0.99 & 0.97 & 0.97 & 1 & 0.97 & 0.97 & 0.99 & 0.99 & 0.99 & 0.99 \\ 0.97 & 0.97 & 0.97 & 0.99 & 0.99 & 0.97 & 1 & 0.99 & 0.97 & 0.97 & 0.97 & 0.97 \\ 0.97 & 0.97 & 0.97 & 0.99 & 0.99 & 0.97 & 0.99 & 1 & 0.97 & 0.97 & 0.97 & 0.97 \\ 0.99 & 0.99 & 0.99 & 0.97 & 0.97 & 0.99 & 0.97 & 0.97 & 1 & 0.99 & 0.99 & 0.99 \\ 0.99 & 0.99 & 0.99 & 0.97 & 0.97 & 0.99 & 0.97 & 0.97 & 0.99 & 1 & 0.99 & 0.99 \\ 0.99 & 0.99 & 0.99 & 0.97 & 0.97 & 0.99 & 0.97 & 0.97 & 0.99 & 0.99 & 1 & 0.99 \\ 0.99 & 0.99 & 0.99 & 0.97 & 0.97 & 0.99 & 0.97 & 0.97 & 0.99 & 0.99 & 0.99 & 1 \end{bmatrix} ,$$

and

$$S_{\alpha=1/3}^{\text{Cooper}''\text{-}5}(A_F, A_F) = \begin{bmatrix} 1 & 0.99 & 0.97 & 0.97 & 0.98 & 0.99 & 0.98 & 0.97 & 0.99 & 0.99 & 0.99 & 0.99 \\ 0.99 & 1 & 0.97 & 0.97 & 0.98 & 0.99 & 0.98 & 0.97 & 0.99 & 0.99 & 0.99 & 0.99 \\ 0.97 & 0.97 & 1 & 0.98 & 0.96 & 0.97 & 0.96 & 0.98 & 0.97 & 0.97 & 0.97 & 0.97 \\ 0.97 & 0.97 & 0.98 & 1 & 0.97 & 0.97 & 0.97 & 0.99 & 0.97 & 0.97 & 0.97 & 0.97 \\ 0.98 & 0.98 & 0.96 & 0.97 & 1 & 0.98 & 0.99 & 0.97 & 0.98 & 0.98 & 0.98 & 0.98 \\ 0.99 & 0.99 & 0.97 & 0.97 & 0.98 & 1 & 0.98 & 0.97 & 0.99 & 0.99 & 0.99 & 0.99 \\ 0.98 & 0.98 & 0.96 & 0.97 & 0.99 & 0.98 & 1 & 0.98 & 0.98 & 0.98 & 0.98 & 0.98 \\ 0.97 & 0.97 & 0.98 & 0.99 & 0.97 & 0.97 & 0.98 & 1 & 0.97 & 0.97 & 0.97 & 0.97 \\ 0.99 & 0.99 & 0.97 & 0.97 & 0.98 & 0.99 & 0.98 & 0.97 & 1 & 0.99 & 0.99 & 0.99 \\ 0.99 & 0.99 & 0.97 & 0.97 & 0.98 & 0.99 & 0.98 & 0.97 & 0.99 & 1 & 0.99 & 0.99 \\ 0.99 & 0.99 & 0.97 & 0.97 & 0.98 & 0.99 & 0.98 & 0.97 & 0.99 & 0.99 & 1 & 0.99 \\ 0.99 & 0.99 & 0.97 & 0.97 & 0.98 & 0.99 & 0.98 & 0.97 & 0.99 & 0.99 & 0.99 & 1 \end{bmatrix} .$$

Note that $r = 6$ (resp. $r = 5$) is the smallest r such that $S_\alpha^{\text{Cooper}'\text{-}r}(A_F, A_F)$ (resp. $S_\alpha^{\text{Cooper}''\text{-}r}(A_F, A_F)$) has ones only on the diagonal. Nevertheless, the node-to-node similarity measure $S_\alpha^{\text{Cooper}'\text{-}r}$ cannot always discriminate non-isomorphic nodes. Indeed, let us consider the graph G whose graphical representation is the following,

,

and A its adjacency matrix. The node-to-node similarity measure $S_\alpha^{\text{Cooper}'\text{-}r}$ yields

$$S_\alpha^{\text{Cooper}'\text{-}r}(A, A) = \mathbf{1}_{3,3} \ , \quad \forall r,$$

whereas $S_\alpha^{\text{Cooper}''\text{-}r}$ yields

$$S_{\alpha=1/3}^{\text{Cooper}''\text{-}1}(A, A) = \begin{bmatrix} 1 & 0.83 & 1 \\ 0.83 & 1 & 0.83 \\ 1 & 0.83 & 1 \end{bmatrix} .$$

The nodes 1 and 3 are isomorphic, hence

$$\left[S_{\alpha=1/3}^{\text{Cooper}''\text{-}1}\right]_{1,1} = \left[S_{\alpha=1/3}^{\text{Cooper}''\text{-}1}\right]_{1,3} = \left[S_{\alpha=1/3}^{\text{Cooper}''\text{-}1}\right]_{3,1} = \left[S_{\alpha=1/3}^{\text{Cooper}''\text{-}1}\right]_{3,3} = 1 \ ,$$

whereas the node 2 is not isomorphic to the nodes 1 and 3 and

$$\left[S^{\text{Cooper}''\text{-}1}_{\alpha=1/3}\right]_{1,1} \neq \left[S^{\text{Cooper}''\text{-}1}_{\alpha=1/3}\right]_{1,2} = \left[S^{\text{Cooper}''\text{-}1}_{\alpha=1/3}\right]_{2,1} \neq \left[S^{\text{Cooper}''\text{-}1}_{\alpha=1/3}\right]_{2,2} \ .$$

We finally introduce $S^{\text{Edit}'}$, a similarity measure for which we can prove that its induced self-similarity measure is strongly isomorphic, but hence, in general, not computable in polynomial time:

$$\left[S^{\text{Edit}'}(A,B)\right]_{i,j} := \frac{[S^{\text{Edit}}(A,B)]_{i,j}}{\sqrt{[S^{\text{Edit}}(A,A)]_{i,i} \cdot [S^{\text{Edit}}(B,B)]_{j,j}}} \ . \tag{3.10}$$

Lemma 3.4 *The node-to-node self-similarity measure induced by $S^{\text{Edit}'}(\cdot,\cdot)$ is strongly isomorphic.*

Proof. By definition, for all $i,j \in N(A)$, we have

$$[S^{\text{Edit}}(A,A)]_{i,j} = \max_{\substack{P\in\mathcal{P}(m),\\ \text{s.t. } P_{ij}=1}} \left\langle A, PAP^T \right\rangle_F \ . \tag{3.11}$$

Let us now prove that, for all $i,j \in N(A)$, we have

$$i \sim j \quad \Leftrightarrow \quad \left[S^{\text{Edit}'}(A,A)\right]_{i,j} = 1 \ .$$

$\Rightarrow$: if i is isomorphic to j, then there exists a permutation $P^\star \in \mathcal{P}(m)$, such that $P^\star_{ij} = 1$ and $A = P^\star A P^{\star T}$. The scalar product in Equation (3.11) is bounded by $\|A\|_F^2$ since,

$$\left\langle A, PAP^T \right\rangle_F \overset{CS}{\leq} \|A\|_F \left\|PAP^T\right\|_F = \|A\|_F^2 \ ,$$

and we have

$$[S^{\text{Edit}}(A,A)]_{i,j} = \left\langle A, P^\star A P^{\star T} \right\rangle_F = \|A\|_F^2 \ ,$$
$$[S^{\text{Edit}}(A,A)]_{i,i} = \left\langle A, I_m A I_m^T \right\rangle_F = \|A\|_F^2 \ , \quad \text{and}$$
$$[S^{\text{Edit}}(A,A)]_{j,j} = \left\langle A, I_m A I_m^T \right\rangle_F = \|A\|_F^2 \ .$$

And eventually, we have

$$\left[S^{\text{Edit}'}(A,A)\right]_{i,j} = \frac{[S^{\text{Edit}}(A,A)]_{i,j}}{\sqrt{[S^{\text{Edit}}(A,A)]_{i,i} \cdot [S^{\text{Edit}}(A,A)]_{j,j}}} = 1 \ .$$

$\Leftarrow$: if $\left[S^{\text{Edit}'}(A,A)\right]_{i,j} = 1$, then

$$[S^{\text{Edit}}(A,A)]_{i,j} = \sqrt{[S^{\text{Edit}}(A,A)]_{i,i} \cdot [S^{\text{Edit}}(A,A)]_{j,j}} \ ,$$

which yields

$$\max_{\substack{P \in \mathcal{P}(m,m), \\ \text{s.t. } P_{ij}=1}} \left\langle A, PAP^T \right\rangle_F = \|A\|_F \cdot \|A\|_F \ .$$

Due to the Cauchy-Schwarz inequality, this can only happen if A is aligned with PAP^T for some P, which, since A and A are binary, yields $A = PAP^T$. □

3.1.3 Stochastic Scalings

As mentioned in the introduction of Section 3.1.1, the node-to-node similarity measures $S^{\Pi\Sigma\text{-}r}$, $S^{\Sigma\Pi\text{-}r}$, and $S^{\text{P-}r}$ tend to give higher similarity scores to highly connected nodes. As a result of this, when considering a node-to-node self-similarity measure induced by a node-to-node similarity measure, two distinct nodes could be more self-similar than one node is self-similar to itself. This effect is commonly avoided by weighting the out-edges or in-edges of a node (usually uniformly) so that their sum is equal to 1. The similarity scores are then either computed by replacing A and B by their so called row-stochastic scalings or column-stochastic scalings, or computed based on a row-stochastic scaling or column-stochastic scaling of $G(A \otimes B)$, the graph product of A and B. One can possibly define the similarity score between the node i and the node j as the Page-Rank [BP98] of a node (i,j) in the graph product of A and B, or equivalently as $[S_\alpha^{\text{P-Rank}}(A,B)]_{i,j}$, the (i,j) entry of the limit point of Algorithm 1 with $S_0^{\text{P-Rank}} = \mathbf{1}_{m,n}$, and

$$\text{vec}(f^{\text{P-Rank}}(S)) := \text{rss}(\alpha\ B \otimes A + (1-\alpha)\mathbf{1})^{\,T}\text{vec}(S)$$

with α, a damping factor. The complexity of computing $f^{\text{P-Rank}}(S)$ is of the order of $|E(A)|\,|E(B)|+mn$ operations. One can also interpret $S_t^{\text{P-Rank}}$ as an extensive quantity[5] that is distributed on the nodes $G(B \otimes A)$. At each step, $[S_t^{\text{P-Rank}}]_{i,j}$, the quantity present on the node (i,j), is divided

[5] An extensive quantity of a system is directly proportional to the amount of material in the system.

in two parts, respectively $\alpha[S_t^{\text{P-Rank}}]_{i,j}$ and $(1-\alpha)[S_t^{\text{P-Rank}}]_{i,j}$. The first one flows across the outgoing edges incident to the node (i,j), whereas the second one flows uniformly towards all nodes of $G(B \otimes A)$. This definition yields

$$S_{\alpha=0.99}^{\text{P-Rank}}(A,A) = \begin{bmatrix} 0.076 & 0.076 & 0.076 & 0.076 & 0.076 & 0.076 & 0.076 \\ 0.076 & 0.08 & 0.103 & 0.088 & 0.088 & 0.103 & 0.08 \\ 0.076 & 0.103 & 0.266 & 0.203 & 0.203 & 0.266 & 0.103 \\ 0.076 & 0.088 & 0.203 & 0.188 & 0.188 & 0.203 & 0.088 \\ 0.076 & 0.088 & 0.203 & 0.188 & 0.188 & 0.203 & 0.088 \\ 0.076 & 0.103 & 0.266 & 0.203 & 0.203 & 0.266 & 0.103 \\ 0.076 & 0.08 & 0.103 & 0.088 & 0.088 & 0.103 & 0.08 \end{bmatrix} ,$$

where A is the adjacency matrix of the graph of Figure 2.1. Notice that, in this case,

$$[S_{\alpha=0.99}^{\text{P-Rank}}(A,A)]_{1,3} = 0.076 < 0.08 = [S_{\alpha=0.99}^{\text{P-Rank}}(A,A)]_{2,2} \ ,$$

but two non-isomorphic nodes can still be more self-similar than a node is self-similar to itself,

$$[S_{\alpha=0.99}^{\text{P-Rank}}(A,A)]_{3,4} = 0.203 > 0.076 = [S_{\alpha=0.99}^{\text{P-Rank}}(A,A)]_{1,1} \ .$$

though the overweight of strongly connected nodes in $S_{\alpha=0.99}^{\text{P-Rank}}(A,A)$ is smaller than the one in $S^{\text{Blondel}}(A,A)$. Notice that the definition $S_\alpha^{\text{P-Rank}}$ generalizes the definition of the Page-Rank since $S_\alpha^{\text{P-Rank}}(A,1)$ clearly is the Page-Rank vector of $G(A)$.

In [MGMR02], Melnik *et al.* introduce a similarity measure defined as the extremal point of Algorithm 1 with $[S_0^{\text{Melnik}}]_{i,j} := 1$, and

$$\begin{aligned}[f^{\text{Melnik}}(S)]_{i,j} := S_{ij} &+ \sum_{\substack{k\in P(i)\\ l\in P(j)}} \frac{S_{kl}}{|C(k)|\,|C(l)| + |P(k)|\,|P(l)|} \\ &+ \sum_{\substack{k\in C(i)\\ l\in C(j)}} \frac{S_{kl}}{|C(k)|\,|C(l)| + |P(k)|\,|P(l)|} \ .\end{aligned} \tag{3.12}$$

Vectorizing this formulation yields $\text{vec}(S_0^{\text{Melnik}}) = \text{vec}(\mathbf{1})$, and

$$\text{vec}(f^{\text{Melnik}}(S)) = \left(I_{mn} + \text{rss}\left(B\otimes A + B^T \otimes A^T\right)^T\right) \text{vec}(S) \ . \tag{3.13}$$

This iterative scheme is the power method [Saa92, IV.1.1] applied to the matrix $I_{mn} + \text{rss}\left(B\otimes A + B^T\otimes A^T\right)^T$ whose unique dominant eigenvalue is $1+\rho\left(\text{rss}\left(B\otimes A + B^T\otimes A^T\right)^T\right)$. If the geometric multiplicity of this dominant eigenvalue is equal to its algebraic multiplicity, then the iterates converge towards the normalized projection of $\text{vec}(S_0^{\text{Melnik}})$

onto the eigenspace associated to this dominant eigenvalue. Moreover, if its algebraic multiplicity is equal to 1 (which is the case if the matrix is irreducible), then the iterates converge towards the Perron vector which, since $B \otimes A + B^T \otimes A^T$ is symmetric, is trivially given by the normalized vector of the degrees of the nodes of $G\left(B \otimes A + B^T \otimes A^T\right)$. And eventually, $[S^{\text{Melnik}}]_{i,j}$, the similarity score between the node i and the node j, is simply proportional to the degree of the node (i,j) in $G(B \otimes A + B^T \otimes A^T)$. Thus, the complexity of computing S^{Melnik} is of the order of $|E(A)|\,|E(B)|$ operations. *E.g.*, let us consider A the adjacency matrix of the graph represented in Figure 2.1. We have

$$S^{\text{Melnik}}(A,A) = 0.018 \times \begin{bmatrix} 16 & 4 & 12 & 0 & 0 & 12 & 4 \\ 4 & 2 & 6 & 2 & 2 & 6 & 2 \\ 12 & 6 & 18 & 6 & 6 & 18 & 6 \\ 0 & 2 & 6 & 4 & 4 & 6 & 2 \\ 0 & 2 & 6 & 4 & 4 & 6 & 2 \\ 12 & 6 & 18 & 6 & 6 & 18 & 6 \\ 4 & 2 & 6 & 2 & 2 & 6 & 2 \end{bmatrix} .$$

3.1.4 Generalized Affine Transformation

In [CABVD10], we emphasized that the reinforcement functions of the node-to-node similarity measures mentioned in this section are all affine functions, *i.e.*

$$[f(S_t)]_{i,j} = \sum_k \sum_l C_{ijkl}\,[S_t]_{k,l} + D_{ij} \ ,$$

which can be rewritten in its vectorized form,

$$\text{vec}(f(S_t)) = \begin{bmatrix} \text{vec}(C_{11\cdot\cdot})^T \\ \text{vec}(C_{21\cdot\cdot})^T \\ \vdots \\ \text{vec}(C_{nn\cdot\cdot})^T \end{bmatrix} \text{vec}(S_t) + \text{vec}(D) \ .$$

The linear term $C_{ijkl}\,[S_t]_{k,l}$ accounts for the reinforcement of the similarities at each iteration and usually propagates them from neighbors to neighbors. The constant term D_{ij} influences the fixed point of the iteration based on *a priory* knowledge on local similarities.

S^{Blondel}, $S_\alpha^{\text{P-Rank}}$, and S^{Melnik} are initialized with $[S_0]_{i,j} = 1$ whereas the

affine transformation coefficients are given by $D_{ij} = 0$, and

$$C_{ijkl}^{\text{Blondel}} = A_{ik}B_{jl} + A_{ki}B_{lj} \ ,$$

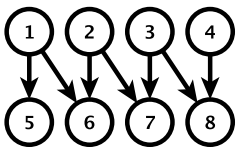
$$C_{ijkl}^{\text{P-Rank}} = \frac{\alpha \ A_{ik}B_{jl} + (1-\alpha)}{\alpha \ |C(k)|\,|C(l)| + (1-\alpha) \ mn} \ ,$$

$$C_{ijkl}^{\text{Melnik}} = \delta_{ik}\delta_{jl} + \frac{A_{ik}B_{jl} + A_{ki}B_{lj}}{|C(k)|\,|C(l)| + |P(k)|\,|P(l)|} \ .$$

In [BND05], Blondel *et al.* consider the non-negative solution of an affine eigenvalue problem with non-negative parameters, *i.e.* $C_{ijkl}, D_{ij} \geq 0$.

3.1.5 Similarity for Weakly Connected Graphs

Except for S^{Edit}, all similarity measures mentioned until now sometimes perform poorly when applied to weakly connected or disconnected graphs. *E.g.*, let us consider the matrix A, the adjacency matrix of the graph whose graphical representation is the following,



.

There is no non-trivial isomorphism from $G(A)$ to itself, though we have

$$S_{\alpha=1}^{\text{Cooper}''\text{-}2}(A,A) = \begin{bmatrix} 1 & 1 & 1 & 0.87 & 0.58 & 0.5 & 0.5 & 0.5 \\ 1 & 1 & 1 & 0.87 & 0.58 & 0.5 & 0.5 & 0.5 \\ 1 & 1 & 1 & 0.87 & 0.58 & 0.5 & 0.5 & 0.5 \\ 0.87 & 0.87 & 0.87 & 1 & 0.67 & 0.58 & 0.58 & 0.58 \\ 0.58 & 0.58 & 0.58 & 0.67 & 1 & 0.87 & 0.87 & 0.87 \\ 0.5 & 0.5 & 0.5 & 0.58 & 0.87 & 1 & 1 & 1 \\ 0.5 & 0.5 & 0.5 & 0.58 & 0.87 & 1 & 1 & 1 \\ 0.5 & 0.5 & 0.5 & 0.58 & 0.87 & 1 & 1 & 1 \end{bmatrix} \ ,$$

which has non-diagonal elements equal to 1. In fact all path methods fail on this example since the entries of the powers of the adjacency matrix, A^k, are all equal to 0 for $k \geq 2$.

This effect can be avoided by combining (for instance) $S(A,B)$, the similarity scores between the nodes of $G(A)$ and $G(B)$, with $S(\mathbf{1}-A, \mathbf{1}-B)$, the similarity scores between the nodes of the complement graphs[6]

[6]Let A be an adjacency matrix. The complement graph of $G(A)$ is defined as $\bar{G}(A) := (N(A), N(A)^2 \setminus E(A))$, and one can show its adjacency matrix is $\bar{A} = \mathbf{1} - A$.

of $G(A)$ and $G(B)$. Indeed, if the neighborhood of the node i in $G(A)$ is similar to the neighborhood of the node j in $G(B)$ then the neighborhood of the node i in the complement of $G(A)$ is probably similar to the neighborhood of the node j in the complement of $G(B)$. Moreover if there is no walk from the node i to some node k in $G(A)$ then there is a walk from the node i to the node k in $G(\mathbf{1} - A)$.

More generally, this effect can be avoided by combining the similarity scores (or scores of the reinforcement function) between the nodes of the graph $G(\phi_i(A))$ and the ones of the graph $G(\phi_i(B))$ with $i = 1, \cdots, k$ where the ϕ_i's are label-invariant mappings acting on adjacency matrices *i.e.* $\phi_i(A) = P\ \phi_i\big(P^T A P\big)\ P^T$. And eventually, similarity measure may perform better on weakly connected or disconnected graphs when considering the following modification

- either on the similarity measure itself

$$\tilde{S}(A, B, \{\cdots, \phi_i, \cdots\}, p) = M_p\left(\cdots, S(\phi_i(A), \phi_i(B)), \cdots\right) \ ,$$

- or on the reinforcement function (when applicable)

$$\tilde{f}(A, B, \{\cdots, \phi_i, \cdots\}, p) = M_p\left(\cdots, f(\phi_i(A), \phi_i(B)), \cdots\right) \ ,$$

where $M_p(x_1, \cdots, x_k)$ is power mean with exponent p of the positive real numbers $(x_1, \cdots, x_k)$ as defined in Equation (3.23). The ϕ_i's can include sum, product, boolean operations, transposition, complement of adjacency matrices, *e.g.* $\phi_1(A) = \mathbf{1} - A$ (adjacency matrix of the complement of $G(A)$), $\phi_2(A) = A \vee A^T$ (the adjacency matrix of the graph whose edges are $E(A) \cup E(A^T)$), *etc.* Applied to the example introduced on page 53 yields

$$\tilde{S}_{\alpha=1}^{\text{Cooper}''\text{-}3}(A, A, \{\phi_1, \phi_2\}, 0) = \begin{bmatrix} 1 & 0.98 & 0.98 & 0.95 & 0.88 & 0.80 & 0.79 & 0.85 \\ 0.98 & 1 & 0.99 & 0.93 & 0.87 & 0.81 & 0.80 & 0.84 \\ 0.98 & 0.99 & 1 & 0.93 & 0.87 & 0.81 & 0.80 & 0.84 \\ 0.95 & 0.93 & 0.93 & 1 & 0.93 & 0.76 & 0.75 & 0.81 \\ 0.88 & 0.87 & 0.87 & 0.93 & 1 & 0.89 & 0.88 & 0.92 \\ 0.80 & 0.81 & 0.81 & 0.76 & 0.89 & 1 & 0.99 & 0.97 \\ 0.79 & 0.80 & 0.80 & 0.75 & 0.88 & 0.99 & 1 & 0.96 \\ 0.85 & 0.84 & 0.84 & 0.81 & 0.92 & 0.97 & 0.96 & 1 \end{bmatrix} .$$

3.1.6 Classification

We now detail in Table 3.1 a number of properties of all node-to-node similarity measures that have been listed in this section.

A first important property is the so-called *symmetry*. $S(\cdot,\cdot)$ is called *symmetric* if $S(A,B) = S(B,A)^T$, for all adjacency matrices A and B. In other words, a similarity measure is symmetric if $[S(A,B)]_{i,j}$, the similarity score between the node $i \in N(A)$ and the node $j \in N(B)$, is equal to $[S(B,A)]_{j,i}$, the similarity score between the node $j \in N(B)$ and the node $i \in N(A)$.

Let $\mathcal{I}$ be a subset of $\mathbb{N}_{\leq m}$. The submatrix $[M_{ij}]_{i,j\in\mathcal{I}}$ of a matrix $M \in \mathbb{R}^{m\times m}$ is denoted $M_{\mathcal{I}}$.

Definition 3.5 *The radius of a similarity measure $S(\cdot,\cdot)$ is the smallest $r \in \mathbb{N}$ such that, for all adjacency matrices A, A', B and B', for all $i \in N(A)$, $j \in N(B)$, $i' \in N(A')$ and $j' \in N(B')$, if there exists*

- *an isomorphism from $G(A_{\mathcal{I}})$ to $G\left(A'_{\mathcal{I}'}\right)$ where $\mathcal{I} = \Gamma^{\leq r}(i)$, and $\mathcal{I}' = \Gamma^{\leq r}(i')$, which maps the node i onto the nodes i', and*
- *an isomorphism from $G(B_{\mathcal{J}})$ to $G\left(B'_{\mathcal{J}'}\right)$ where $\mathcal{J} = \Gamma^{\leq r}(j)$, and $\mathcal{J}' = \Gamma^{\leq r}(j')$, which maps the node j onto the nodes j',*

then $[S(A,B)]_{i,j} = [S(A',B')]_{i',j'}$.

3.1.7 Summary

We review here all node-to-node similarity measures that we have presented in this section.

- $S^{\text{Blondel}}(A,B)$ [BGH+04] is equal to the extremal point of Algorithm 1 with

  $$S_0^{\text{Blondel}} := \mathbf{1}_{m,n} \ , \quad \text{and} \quad [f^{\text{Blondel}}(S)]_{i,j} := \left[ASB^T + A^TSB\right]_{i,j} \ .$$

  Advantage:

  - Its definition is simple and extends the Hub and Authority scores. This makes its mathematical analysis easy and interesting.

  Disadvantage:

  - It tends to give higher similarity scores to highly connected nodes.
  - It gives poor information about node-to-node similarity, even though, Blondel *et al.* unconvincingly tried to apply it to synonyms extraction [BGH+04].

- $[S_\alpha^{\Pi\Sigma\text{-}r}(A,B)]_{i,j} := \langle \mathbf{x}_i(A), \mathbf{x}_j(B) \rangle_F$ with

$$\mathbf{x}_i(A) := \left[\; X_{\alpha,r}(A) \;\middle|\; X_{\alpha,r}\left(A^T\right) \; \right]_{i,\cdot} , \quad \text{and}$$

$$X_{\alpha,r}(A) := \begin{bmatrix} \alpha A \mathbf{1}_{m,1} & \alpha^2 A^2 \mathbf{1}_{m,1} & \cdots & \alpha^r A^r \mathbf{1}_{m,1} \end{bmatrix} .$$

  We only defined it to introduce $S_\alpha^{\text{Cooper-}r}$ which is its diagonally scaled version. It should not be used without scaling, but the scaling used to define $S_\alpha^{\text{Cooper-}r}$ is not the only possibility one could consider.

- $[S_\alpha^{\Sigma\Pi\text{-}r}(A,B)]_{i,j} := \sum_{k=1}^{r} \max_{\substack{P\in\mathcal{P}(m)\\ Q\in\mathcal{P}(n)}} \langle \mathbf{x}_{i,k}(A) P \; I_{m,n}, \mathbf{x}_{j,k}(B) Q \rangle_F$

$$+ \max_{\substack{P\in\mathcal{P}(m)\\ Q\in\mathcal{P}(n)}} \left\langle \mathbf{x}_{i,k}\left(A^T\right) P \; I_{m,n}, \mathbf{x}_{j,k}\left(B^T\right) Q \right\rangle_F ,$$

  where $\mathbf{x}_{i,k}(A) := \left[\alpha^k A^k\right]_{i,\cdot}$.

  We only defined it to introduce $S_\alpha^{\text{Cooper}'\text{-}r}$ which is its diagonally scaled version. It should not be used without scaling, but the scaling used to define $S_\alpha^{\text{Cooper}'\text{-}r}$ is not the only possibility one could consider.

- $[S_\alpha^{\text{P-}r}(A,B)]_{i,j} := \max_{\substack{P\in\mathcal{P}(m)\\ Q\in\mathcal{P}(n)}} \langle X_i(A) P \; I_{m,n}, X_j(B) Q \rangle_F$, where

$$X_i(A) := \begin{bmatrix} Y_i(A) \\ Y_i\left(A^T\right) \end{bmatrix} , \quad \text{and} \quad Y_i(A) := \begin{bmatrix} \left[\alpha^0 A^0\right]_{i,\cdot} \\ \left[\alpha^1 A^1\right]_{i,\cdot} \\ \vdots \\ \left[\alpha^r A^r\right]_{i,\cdot} \end{bmatrix} .$$

  We only defined it to introduce $S_\alpha^{\text{Cooper}''\text{-}r}$ which is its diagonally scaled version. It should not be used without scaling, but the scaling used to define $S_\alpha^{\text{Cooper}''\text{-}r}$ is not the only possibility one could consider.

- $[S^{\text{Edit}}(A,B)]_{i,j} := \max_{\substack{P\in\mathcal{P}(m,n),\\ \text{s.t. } P_{ij}=1}} \left\langle A, PBP^T \right\rangle_F .$

  We only defined it to introduce $S_\alpha^{\text{Edit}'r}$ which is its diagonally scaled version. It should not be used without scaling, but the scaling used to define $S_\alpha^{\text{Edit}'r}$ is not the only possibility one could consider.

- $\left[S^{\text{Blondel}'}(A,B)\right]_{i,j} := \dfrac{[S^{\text{Blondel}}(A,B)]_{i,j}}{\max\left([S^{\text{Blondel}}(A,A)]_{i,i}, [S^{\text{Blondel}}(B,B)]_{j,j}\right)}$

  It is a diagonally scaled version of S^{Blondel} which has at least the advantage to turn S^{Blondel} into a weakly isomorphic similarity measure.

- $[S_\alpha^{\text{Cooper-}r}(A,B)]_{i,j} := \dfrac{[S_\alpha^{\Pi\Sigma\text{-}r}(A,B)]_{i,j}}{\sqrt{[S_\alpha^{\Pi\Sigma\text{-}r}(A,A)]_{i,i}\cdot[S_\alpha^{\Pi\Sigma\text{-}r}(B,B)]_{j,j}}}$ [CB10].

  Advantage:

  - Its values are easily interpretable.
  - It is weakly isomorphic.

  Disadvantage:

  - It is not strongly isomorphic and poorly discriminates non-isomorphic nodes in regular graphs such as the Frucht graph.

- $\left[S_\alpha^{\text{Cooper}'\text{-}r}(A,B)\right]_{i,j} := \dfrac{[S_\alpha^{\Sigma\Pi\text{-}r}(A,B)]_{i,j}}{\sqrt{[S_\alpha^{\Sigma\Pi\text{-}r}(A,A)]_{i,i}\cdot[S_\alpha^{\Sigma\Pi\text{-}r}(B,B)]_{j,j}}}.$

  Advantage:

  - It is weakly isomorphic.
  - Even if it is not strongly isomorphic, it discriminates perfectly the non-isomorphic nodes in the Frucht graph.

  Disadvantage:

  - Its values are less easily interpretable than the ones of $S_\alpha^{\text{Cooper-}r}$.

- $\left[S_\alpha^{\text{Cooper}''\text{-}r}(A,B)\right]_{i,j} := \dfrac{[S_\alpha^{\text{P-}r}(A,B)]_{i,j}}{\sqrt{[S_\alpha^{\text{P-}r}(A,A)]_{i,i}\cdot[S_\alpha^{\text{P-}r}(B,B)]_{j,j}}}.$

  Advantage:

  - It is weakly isomorphic.

  - It discriminates perfectly the non-isomorphic nodes in the Frucht graph. We did not find any example that proves it is not strongly isomorphic. It is actually comparable to techniques used for solving the graph isomorphism problem introduced in [SS08].

  Disadvantage:

  - Its values are less easily interpretable than the ones of $S_\alpha^{\text{Cooper-}r}$.

- $\left[S^{\text{Edit}'}(A,B)\right]_{i,j} := \dfrac{[S^{\text{Edit}}(A,B)]_{i,j}}{\sqrt{[S^{\text{Edit}}(A,A)]_{i,i} \cdot [S^{\text{Edit}}(B,B)]_{j,j}}}$.

  It proves that a similarity measure can be strongly isomorphic, but it is not supposed to be used in practice since it is often computationally expensive.

- $S_\alpha^{\text{P-Rank}}(A,B)$ is equal to the extremal point of Algorithm 1 with

$$S_0^{\text{P-Rank}} := \mathbf{1}_{m,n} \ , \quad \text{and}$$
$$\text{vec}(f^{\text{P-Rank}}(S)) := \text{rss}(\alpha\ B \otimes A + (1-\alpha)\mathbf{1})^{\,T}\text{vec}(S) \ .$$

  It is to the Page-Rank score what S^{Blondel} is to the Hub and Authority scores. It is inspired from S^{Melnik} while being less trivial than S^{Melnik} when the algebraic multiplicity of the spectral radius eigenvalue is 1.

- $S^{\text{Melnik}}(A,B)$ [MGMR02] is equal to the extremal point of Algorithm 1 with

$$S_0^{\text{Melnik}} = \mathbf{1}_{m,n} \ , \quad \text{and}$$
$$\text{vec}(f^{\text{Melnik}}(S)) = \text{vec}(S) + \text{rss}\big(B \otimes A + B^T \otimes A^T\big)^{\,T}\ \text{vec}(S) \ .$$

  It gives poor information about node-to-node similarity when the algebraic multiplicity of the spectral radius eigenvalue is 1.

3.2 Node-to-Node Self-Similarity Measures

As previously mentioned, any node-to-node similarity measure $S(\cdot,\cdot)$ trivially induces a node-to-node self-similarity measure $S'(\cdot)$ defined as $S'(A) := S(A,A)$ which we studied in Section 3.1 when considering the

corresponding node-to-node similarity measure. The other node-to-node self-similarity measures yield similarity scores between the node i and the node j that are essentially (and we hereafter assume that they always are) based on the structure of the networks in between the nodes i and j, *e.g.*

> *If i and j have many common children or parents,*
> *then the similarity between i and j is large.*

or

> *If there are many walks from i to j,*
> *then the similarity between i and j is large.*

Hence, since relabeling the nodes does not influence the structure of a graph, they are usually invariant under relabeling, which means that, given a node-to-node self-similarity measure $S(\cdot)$, we have, for every adjacency matrix A, and for all permutation matrices P,

$$S(A) = P\ S(P^T A P)\ P^T\ .$$

Several node-to-node self-similarity measures that we will further introduce are weakly or strongly structural (*cf.* Definition of p. 27).

In this section, we consider the following node-to-node self-similarity measures:

- We first introduce basic similarity measures which help to introduce other similarity measures.

  $S^{\cap\Gamma^r}(A)$ ..... p. 61

  $S^{\overset{r}{\rightarrow}}(A)$ ..... p. 61

  $S^{\text{Zhou}}(A)$ ..... p. 62

  $S^{\text{Leicht}}(A)$ ..... p. 62

  $S^{\text{Estrada}}(A)$ ..... p. 62

- We further consider a similarity measure that scales $S^{\cap\Gamma^1}(A)$ by an interesting upper bound. Notice that it is strongly isomorphic.

  $S^{\text{Jaccard}}(A)$ ..... p. 64

- We further consider similarity measures that are diagonally scaled versions of the basic definitions. Notice that most of them are strongly isomorphic.

- We further consider similarity measures that scale the basic definitions by their expected similarity scores in a randomized model called the null model.

- We finally consider similarity measures that involve computation with the stochastically scaled adjacency matrix.

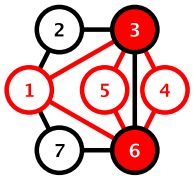
3.2.1 Basic Definitions

A first basic definition consists in defining the similarity between i and j as the number of nodes in the intersection of $\Gamma^r(i)$ and $\Gamma^r(j)$, *i.e.*

$$\left[S^{\cap\Gamma^r}(A)\right]_{i,j} := \left|\Gamma^r(i) \cap \Gamma^r(j)\right| \ . \tag{3.14}$$

For directed weighted networks, this definition may be extended as follows $\left[S^{\cap\Gamma^r}(A)\right]_{i,j} := \frac{1}{2}\left[A^r \left(A^T\right)^r\right]_{i,j} + \frac{1}{2}\left[\left(A^T\right)^r A^r\right]_{i,j}$. The complexity of computing $S^{\cap\Gamma^r}(A)$ is thus of the order of $2rm\left|E(A)\right|$ operations. For A, the adjacency matrix of the graph represented in Figure 2.2, we have

$$S^{\cap\Gamma^1}(A) = \begin{bmatrix} 4&1&2&2&2&2&1\\ 1&2&1&1&1&2&1\\ 2&1&5&1&1&\mathbf{3}&2\\ 2&1&1&2&2&1&1\\ 2&1&1&2&2&1&1\\ 2&2&3&1&1&5&1\\ 1&1&2&1&1&1&2 \end{bmatrix}.$$



As emphasized above, the similarity score between the node 3 and the node 6 is 3 since there are 3 nodes in the intersection between $\Gamma^1(3)$ and $\Gamma^1(6)$, namely the nodes 1, 4 and 5.

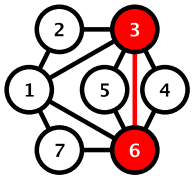
Another basic definition consists in defining the similarity between i and j as the number of undirected walks of length r from i to j, *i.e.*

$$\left[S^{\overset{r}{\rightarrow}}(A)\right]_{i,j} := \left|\wp\left(i \overset{r}{\rightarrow} j\right)\right| \ . \tag{3.15}$$

with $\wp\left(i \overset{r}{\rightarrow} j\right)$ denotes the set of walks of length r starting at node i and ending at node j.

For directed weighted networks, these definitions may be directly extended as follows $\left[S^{\overset{r}{\rightarrow}}(A)\right]_{i,j} := \left[A^r\right]_{i,j}$. The complexity of computing $S^{\overset{r}{\rightarrow}}(A)$ is thus of the order of $rm\left|E(A)\right|$ operations. For A, the adjacency matrix of the graph represented in Figure 2.2, we obtain:

$$S^{\overset{1}{\rightarrow}}(A) = \begin{bmatrix} 0&1&1&0&0&1&1\\ 1&0&1&0&0&0&0\\ 1&1&0&1&1&\mathbf{1}&0\\ 0&0&1&0&0&1&0\\ 0&0&1&0&0&1&0\\ 1&0&1&1&1&0&1\\ 1&0&0&0&0&1&0 \end{bmatrix}.$$



For undirected graphs, we have $S^{\overset{2r}{\rightarrow}}(A) = S^{\cap\Gamma^r}(A)$.

The node-to-node similarity measures $S^{\cap\Gamma^r}$ and $S^{\overset{r}{\rightarrow}}$ count the number of times a simple particular structure is observed between i and j. However, many more similarity measures may be built by counting the number of times a more complex structure is observed between i and j, *e.g.*

$$S_{ij}(A) = \text{the size of the largest clique that contains } i \text{ and } j \ ,$$

(a clique is a set of nodes such that every node in the set is connected to all the other nodes of the set) or, any of their combinations.

- The measure introduced by Zhou *et al.* in [ZLZ09] is designed for undirected graphs and defined as

  $$S^{\text{Zhou}}(A) := S^{\overset{2}{\rightarrow}}(A) + \alpha S^{\overset{3}{\rightarrow}}(A) \ .$$

  The purpose of this similarity consists in having fewer pair of nodes with the same similarity scores, conversely to $S^{\overset{2}{\rightarrow}}(A)$ in which the similarity scores are mostly 0, occasionally 1 and rarely 2 or more.

- In [LHN06], Leicht *et al.* consider that the similarity between i and j should be large if the similarity between the neighbors of i and j is large, or if i is equal to j. This leads them to introduce a measure $S^{\text{Leicht}}(A)$ designed for undirected graphs and defined as the extremal point of Algorithm 1 with $\text{normalize}^{\text{Leicht}}(S) := S$, $[S_0^{\text{Leicht}}]_{i,j} := \delta_{ij}$, and

  $$[f^{\text{Leicht}}(S)]_{i,j} := [I + \alpha AS]_{i,j} = \delta_{ij} + \alpha \sum_{k \in \Gamma(i)} S_{kj} \ ,$$

  where α is a damping factor. The complexity of computing $f^{\text{Leicht}}(S)$ is of the order of $|E(A)|\, m$ operations. In [LHN06], Leicht *et al.* show that, if $\alpha < \rho(A)$, then S^{Leicht} is a fixed point of f^{Leicht}, or in other words $S^{\text{Leicht}} = I + \alpha A S^{\text{Leicht}}$, which yields

  $$S_\alpha^{\text{Leicht}}(A) = [I - \alpha A]^{-1} = I + \alpha A + \alpha^2 A^2 + \alpha^3 A^3 + \cdots \ , \quad (3.16)$$

  or in other words

  $$S_\alpha^{\text{Leicht}}(A) = S^{\overset{0}{\rightarrow}}(A) + \alpha S^{\overset{1}{\rightarrow}}(A) + \alpha^2 S^{\overset{2}{\rightarrow}}(A) + \alpha^3 S^{\overset{3}{\rightarrow}}(A) + \cdots \ . \quad (3.17)$$

- Finally, in [EH10], Estrada and Higham propose the following definition

$$S^{\text{Estrada}}(A) := \exp(A) = \frac{I}{0!} + \frac{A}{1!} + \frac{A^2}{2!} + \frac{A^3}{3!} + \cdots \ , \tag{3.18}$$

or in other words

$$S^{\text{Estrada}}(A) = \frac{S^{\overset{0}{\rightarrow}}(A)}{0!} + \frac{S^{\overset{1}{\rightarrow}}(A)}{1!} + \frac{S^{\overset{2}{\rightarrow}}(A)}{2!} + \frac{S^{\overset{3}{\rightarrow}}(A)}{3!} + \cdots \ . \tag{3.19}$$

The node-to-node similarity measures $S^{\cap\Gamma^r}$, and $S^{\overset{r}{\rightarrow}}$ tend to give higher similarity scores to highly connected nodes. As a result of this, two distinct nodes could be more similar than one node is similar to itself. *E.g.*, in the earlier examples illustrating the definitions of $S^{\cap\Gamma^r}$ and $S^{\overset{r}{\rightarrow}}$, we had

$$\left[S^{\cap\Gamma^1}\right]_{3,6} = 3 > 2 = \left[S^{\cap\Gamma^1}\right]_{2,2} \quad \text{and} \quad \left[S^{\overset{1}{\rightarrow}}\right]_{3,6} = 1 > 0 = \left[S^{\overset{1}{\rightarrow}}\right]_{2,2} .$$

This effect is commonly avoided by weighting the definitions.

1. **The upper bound scalings.** The similarity score between the node i and the node j is scaled by any relevant upper bound of the similarity score between the node i and the node j, *i.e.*

$$[S_w]_{i,j} = \frac{S_{ij}}{U_{ij}} \ .$$

2. **The diagonal scalings.** The similarity score between the node i and the node j is scaled by a function that depends of the similarity score between the node i and itself, and possibly also of the similarity score between the node j and itself, *i.e.*

$$[S_w]_{i,j} = \frac{S_{ij}}{w(S_{ii}, S_{jj})} \ .$$

3. **The null model scalings.** The similarity score between the node i and the node j is scaled by the expected value of the similarity score between the node i and the node j in a *null model*, *i.e.*

$$[S_w]_{i,j} = \frac{S_{ij}}{\mathbb{E}(S_{ij})} \ .$$

4. **The stochastic scalings.** The out-edges or in-edges of a node are (usually uniformly) weighted so that their sum is equal to 1. The similarity scores are then computed by replacing A by its so called row-stochastic scaling or column-stochastic scaling, *i.e.*

$$[\,\mathrm{rss}(A)\,]_{i,j} := \begin{cases} \dfrac{A_{ij}}{\sum_j A_{ij}}, \text{ if } \sum_j A_{ij} \neq 0 \\ 0, \text{ otherwise,} \end{cases}$$

or

$$[\,\mathrm{css}(A)\,]_{i,j} := \begin{cases} \dfrac{A_{ij}}{\sum_i A_{ij}}, \text{ if } \sum_i A_{ij} \neq 0 \\ 0, \text{ otherwise.} \end{cases}$$

3.2.2 Upper Bound Scalings

As mentioned in the introduction of Section 3.2.1, the node-to-node similarity measures $S^{\cap\Gamma^r}$, and $S^{\overset{r}{\to}}$ tend to give higher similarity scores to highly connected nodes. This effect is commonly avoided by weighting the similarity score between the node i and the node j by any relevant upper bound of the similarity score between the node i and the node j.

A non trivial upper bound of $\left[S^{\cap\Gamma^r}(A)\right]_{i,j} (:= |\Gamma^r(i) \cap \Gamma^r(j)|)$ is found when one replaces the "$\cap$" by a "$\cup$". The *Jaccard Index* is a similarity measure designed for undirected graph that was defined in [Jac01] over a hundred years ago as

$$[S^{\text{Jaccard}}(A)]_{i,j} := \frac{|\Gamma(i) \cap \Gamma(j)|}{|\Gamma(i) \cup \Gamma(j)|}\ , \tag{3.20}$$

extended to directed weighted networks as

$$[S^{\text{Jaccard}}(A)]_{i,j} := \frac{\left[S^{\cap\Gamma^1}(A)\right]_{i,j}}{\left[S^{\cap\Gamma^1}(A)\right]_{i,i} + \left[S^{\cap\Gamma^1}(A)\right]_{j,j} - \left[S^{\cap\Gamma^1}(A)\right]_{i,j}}\ .$$

The *Jaccard Index* is used, among other, by biologists (including Jaccard) to compare biological entities and their habitats. Indeed, let the biological entities and their habitats be the nodes of a bipartite graph in which the i^{th} biological entity is linked to the j^{th} habitat if i lives in j.

Let now i_1 and i_2 be two biological entities. If $S_{i_1 i_2}$ is close to 1 (resp. 0), then i_1 and i_2 are usually found in the same (resp. different) places.

Let us notice that, for undirected graphs, the *Jaccard Index* can be rewritten as

$$[S_\alpha(A)]_{i,j} = \lim_{t \to \alpha} \frac{1}{1 + t\frac{|\Gamma(i)\Delta\Gamma(j)|}{|\Gamma(i)\cap\Gamma(j)|}} \ , \quad \text{with } \alpha = 1, \tag{3.21}$$

and where Δ denotes the symmetric difference. One can further notice that

$$[S_{\alpha=0}(A)]_{i,j} = \begin{cases} 1, \text{ if } |\Gamma(i) \cap \Gamma(j)| \neq 0, \\ 0, \text{ otherwise,} \end{cases}$$

whereas

$$[S_{\alpha=\infty}(A)]_{i,j} = \begin{cases} 1, \text{ if } |\Gamma(i)\Delta\Gamma(j)| = 0, \\ 0, \text{ otherwise,} \end{cases}$$

which is exactly the matrix representation of the structural equivalence relation for undirected graphs as defined in Equation (2.1). Note that this reasoning holds also for directed graphs.

3.2.3 Diagonal Scalings

As mentioned in the introduction of Section 3.2.1, the node-to-node similarity measures $S^{\cap\Gamma^r}$, and $S^{\overset{r}{\to}}$ tend to give higher similarity scores to highly connected nodes. This effect is commonly avoided by weighting the similarity score between the node i and the node j by a function that depends of the similarity score between the node i and itself, and possibly also of the similarity score between the node j and itself, *i.e.*

$$[S_w]_{i,j} = \frac{S_{ij}}{w(S_{ii}, S_{jj})} \ .$$

Several similarity measures scale S_{ij} by a *power mean* of S_{ii} and S_{jj}, *i.e.*

$$[S_p]_{i,j} = \frac{S_{ij}}{M_p\left(S_{ii}, S_{jj}\right)} \ . \tag{3.22}$$

Given $p \in \bar{\mathbb{R}}$. The power mean with exponent p of the positive real numbers $(x_1, \cdots, x_n)$ is defined as

$$M_p\left(x_1, \cdots, x_n\right) := \lim_{t \to p} \left(\frac{1}{n} \sum_{i=1}^{n} (x_i)^t \right)^{1/t} \ . \tag{3.23}$$

For particular values of p, the power mean simply reduces to other well known means:

- M_0 is the geometric mean[7], *i.e.* $M_0(x_1, \cdots, x_n) = \left(\prod_{i=1}^{n} x_i\right)^{1/n}$,
- M_1 is the arithmetic mean, *i.e.* $M_1(x_1, \cdots, x_n) = \frac{1}{n}\sum_{i=1}^{n} x_i$,
- M_∞ is the maximum, *i.e.* $M_\infty(x_1, \cdots, x_n) = \max(x_1, \cdots, x_n)$,
- M_{-1} is the harmonic mean, *i.e.* $M_{-1}(x_1, \cdots, x_n) = \frac{n}{\sum_{i=1}^{n} \frac{1}{x_i}}$, and
- $M_{-\infty}$ is the minimum, *i.e.* $M_{-\infty}(x_1, \cdots, x_n) = \min(x_1, \cdots, x_n)$.

For these particular values of p, the scaling of $\left[S^{\cap\Gamma^1}\right]_{i,j}$ $(:= |\Gamma(i) \cap \Gamma(j)|)$ by a power mean of exponent p of the associated diagonal terms yields common named similarity measure. Note that the diagonal terms reduce to

$$\left[S^{\cap\Gamma^1}\right]_{i,i} = |\Gamma(i) \cap \Gamma(i)| = |\Gamma(i)| \ .$$

- The case $p = 0$ is known as the *Salton* (or *Cosine*) *Index* [SM83]

$$\left[S^{\cap\Gamma^1}_{p=0}\right]_{i,j} := \frac{|\Gamma(i) \cap \Gamma(j)|}{\sqrt{|\Gamma(i)| \cdot |\Gamma(j)|}} \ . \tag{3.24}$$

  In [Ley08], Leydesdorff considers the *Jaccard* and *Salton Index* to analyze author co-citation networks. Notice that the *Pearson's Correlation*, (*e.g.* presented in [AJR03]) is a slightly modified version of $S^{\cap\Gamma^1}_{p=0}$ that is defined as $[S^{\text{corr.}}]_{i,j} = \text{corr}(A_{i\cdot}, A_{j\cdot})$ with

$$\text{corr}(u, v) := \frac{\text{covar}(u, v)}{\sqrt{\text{covar}(u, u) \cdot \text{covar}(v, v)}}$$

[7] $M_0(x_1, \cdots, x_n) := \lim_{t\to 0} \left(\frac{1}{n}\sum_{i=1}^{n}(x_i)^t\right)^{1/t} = \exp\left(\lim_{t\to 0} \frac{\log\left(\frac{1}{n}\sum_{i=1}^{n}(x_i)^t\right)}{t}\right)$
$\overset{\text{Hospital}}{=} \exp\left(\lim_{t\to 0} \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i)^t \log(x_i)}{\frac{1}{n}\sum_{i=1}^{n}(x_i)^t}\right) = \exp\left(\frac{\sum_{i=1}^{n}\log(x_i)}{n}\right) = \sqrt[n]{\prod_{i=1}^{n} x_i}$.

and

$$\mathrm{covar}(u,v) := \sum_{k=1}^{n} \left[u_k - M_1(u)\right]\left[v_k - M_1(v)\right] \ .$$

- The case $p = 1$ is known as the *Sørensen-Dice Index* [Sor48]

$$\left[S^{\cap\Gamma^1}_{p=1}\right]_{i,j} := \frac{|\Gamma(i) \cap \Gamma(j)|}{\frac{|\Gamma(i)|+|\Gamma(j)|}{2}} \ . \tag{3.25}$$

  As the *Jaccard Index*, the *Sørensen-Dice Index* is used, among others, by biologists to analyze ecological structures [Dah60]. Notice that the $S^{\cap\Gamma^1}_{p=1}$ can be rewritten as $S_{\alpha=\frac{1}{2}}$ (*cf.* Equation (3.21)).

- The case $p = \infty$ is known as the *Braun-Blanquet Index* [BB32]

$$\left[S^{\cap\Gamma^1}_{p=\infty}\right]_{i,j} := \frac{|\Gamma(i) \cap \Gamma(j)|}{\max(|\Gamma(i)|, |\Gamma(j)|)} \tag{3.26}$$

- The case $p = -1$ is known as the *Second Kulczynski Index*

$$\left[S^{\cap\Gamma^1}_{p=-1}\right]_{i,j} := \frac{|\Gamma(i) \cap \Gamma(j)|}{\frac{2}{\frac{1}{|\Gamma(i)|}+\frac{1}{|\Gamma(j)|}}} \tag{3.27}$$

- The case $p = -\infty$ is known as the *Simpson* (or *overlap*) *Index*

$$\left[S^{\cap\Gamma^1}_{p=-\infty}\right]_{i,j} := \frac{|\Gamma(i) \cap \Gamma(j)|}{\min(|\Gamma(i)|, |\Gamma(j)|)} \tag{3.28}$$

  The *topological overlap index* introduced by Ravasz *et al.* in [RSM+02] is a slightly modified version of $S^{\cap\Gamma^1}_{p=-\infty}$ that is equivalent to

$$[S^{\text{Ravasz}}]_{i,j} := \frac{\left[S^{\cap\Gamma^1} + S^{\overset{1}{\rightarrow}}\right]_{i,j}}{\min\left(\left[S^{\cap\Gamma^1} + S^{\overset{1}{\rightarrow}}\right]_{i,i}, \left[S^{\cap\Gamma^1} + S^{\overset{1}{\rightarrow}}\right]_{j,j}\right)} \tag{3.29}$$

  for graphs without loops.

Notice that, for all $p_1, p2 \in \mathbb{R} \cup \{\infty, -\infty\}$, we clearly have

$$p_1 < p2 \quad \Rightarrow \quad \left[S^{\cap\Gamma^1}_{p=p2}\right]_{i,j} \leq \left[S^{\cap\Gamma^1}_{p=p1}\right]_{i,j} \ ,$$

and moreover

$$[S^{\text{Jaccard}}]_{i,j} \leq \left[S^{\cap\Gamma^1}_{p=\infty}\right]_{i,j} .$$

Notice also that the order of the similarity sores is not necessarily preserved when one considers different values of p , *i.e.*

$$\left[S^{\cap\Gamma^1}_{p=p1}\right]_{i,j} \leq \left[S^{\cap\Gamma^1}_{p=p1}\right]_{k,l} \not\Rightarrow \left[S^{\cap\Gamma^1}_{p=p2}\right]_{i,j} \leq \left[S^{\cap\Gamma^1}_{p=p2}\right]_{k,l} .$$

For undirected graphs, we propose to extend $S^{\cap\Gamma^1}_p$ as

$$\left[S^{\cap\Gamma^1}_p(A)\right]_{i,j} := \frac{\left[S^{\cap\Gamma^1}\right]_{i,j}}{M_p\left(\left[S^{\cap\Gamma^1}\right]_{i,i}, \left[S^{\cap\Gamma^1}\right]_{j,j}\right)} \tag{3.30}$$

Let us finally mention the node-to-node similarity measure introduced in [SM83, p. 203] that asymmetrically scales S_{ij} by S_{ii},

$$\left[S^{\cap\Gamma^1}_{w=S_{ii}}(A)\right]_{i,j} := \frac{\left[S^{\cap\Gamma^1}\right]_{i,j}}{\left[S^{\cap\Gamma^1}\right]_{i,i}} = \frac{|\Gamma(i)\cap\Gamma(j)|}{|\Gamma(i)|} . \tag{3.31}$$

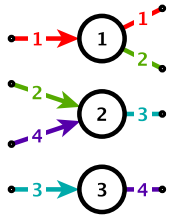
3.2.4 Null Model Scalings

As mentioned in the introduction of Section 3.2.1, the node-to-node similarity measures $S^{\cap\Gamma^r}$, and $S^{\overset{r}{\rightarrow}}$ tend to give higher similarity scores to highly connected nodes. As a result of this, two distinct nodes could be more similar than one node is similar to itself. This effect is commonly avoided by weighting the similarity score $[S(A,A)]_{i,j}$ by the expected similarity score of the random variable $[S(X,X)]_{i,j}$ where X is a random variable defined over a set of adjacency matrices $\mathcal{A}$ that contains A.

Let the matrices $B_{\text{out}} \in \{0,1\}^{n\times m}$ (resp. $B_{\text{in}} \in \{0,1\}^{n\times m}$) be the *out incidence matrix* (resp. *in incidence matrix*), or in other words, $[B_{\text{out}}]_{i,j} = 1$ if the i^{th} node is the origin of the j^{th} edge, 0 otherwise, and $[B_{\text{in}}]_{i,j} = 1$ if the i^{th} node is the termination of the j^{th} edge, 0 otherwise. It is easy to see that the adjacency matrix associated with B_{out} and B_{in} is given by $A = B_{\text{out}}{B_{\text{in}}}^T$. *E.g.*, let us consider the graph whose graphical representation is the following,

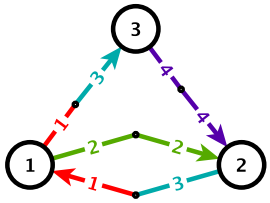
We have $B_{\text{out}} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$, $B_{\text{in}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$, and $A = B_{\text{out}} {B_{\text{in}}}^T = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$.

Let us cut the edges of $G(A)$ in half-edges (an origin and a termination) and rewire all the origins with all the terminations, and define a rewiring matrix $R \in \mathcal{P}(m)$ such that $R_{ij} = 1$ if the origin of the i^{th} edge is rewired with the termination of the j^{th} edge. One can show that the adjacency matrix of the rewired graph is $A_R := B_{\text{out}} R {B_{\text{in}}}^T$. *E.g.*, in the above example, cutting the edges yields



,

while rewiring them with $R = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ yields



.

The *configuration model* is the null model obtained when the random variable $X = B_{\text{out}} R {B_{\text{in}}}^T$, with R uniformly distributed over the set of permutation matrices, $\mathcal{P}(m)$. If $G(A)$ is undirected (*i.e.* $A = A^T$), then one may require that R rewires the edges symmetrically, *i.e.* $A_R = {A_R}^T$.

In [LHN06], Leicht *et al.* scale $S^{\cap \Gamma^1}$ by a quantity proportional to its expected value in the configuration model, *i.e.*

$$\left[S^{\text{Leicht}'}\right]_{i,j} := \frac{|\Gamma(i) \cap \Gamma(j)|}{|\Gamma(i)| \cdot |\Gamma(j)|} \ . \tag{3.32}$$

Indeed, the expected number of walks of length 1 from i to j in a configuration model can be computed as follows. For any of the $|\Gamma(i)|$ edges emerging from vertex i, there are $|E|$ possible terminations of which

$|\Gamma(j)|$ ends at node j. Hence, for each edge emerging from i there is a probability $\frac{|\Gamma(j)|}{|E|}$ to terminate in j, and the number of expected walks of length 1 from i to j is $\frac{|\Gamma(i)||\Gamma(j)|}{|E|}$. And the expected number of walks of length 2 from i to j (equal to the number of common neighbors in undirected graphs) in a configuration model can be computed as

$$\sum_{k=1}^{|N|} \frac{|\Gamma(i)|\,|\Gamma(k)|}{|E|} \frac{|\Gamma(k)|\,|\Gamma(j)|}{|E|} = \frac{|\Gamma(i)|\,|\Gamma(j)|}{|E|^2} \sum_{k=1}^{|N|} |\Gamma(k)|^2$$

$$= \frac{|\Gamma(i)|\,|\Gamma(j)|}{|E|} \frac{\sum_{k=1}^{|N|} |\Gamma(k)|^2 / N}{\sum_{k=1}^{|N|} |\Gamma(k)| / N}$$

In [LHN06], Leicht *et al.* propose to scale

$$S^{\text{Leicht}} = [I - \alpha A]^{-1} = I + \alpha A + \alpha^2 A^2 + \alpha^3 A^3 + \cdots \ ,$$

by dividing the term $\left[A^l\right]_{i,j}$, the number of walks of length l from i to j, by a number asymptotically equal to the number of expected walks in the configuration model. They hence define

$$\left[S^{\text{Leicht}''}\right]_{i,j} := \sum_{l=0}^{\infty} W_{l,ij} \ \left[A^l\right]_{i,j}$$

with $W_{0,ij} = \delta_{ij}$, and, for $l \geq 1$,

$$W_{l,ij} = \frac{1}{\frac{|\Gamma(i)||\Gamma(j)|}{|E|}\ \rho^{l-1}} \ , \quad \text{with } \rho \text{ the spectral radius of } A \ .$$

Notice first that this choice for $W_{0,ij}$ counterintuitively does not match the purpose of the weighted definition. Indeed, there is no more walk of length 0 between two nodes in the original graph than there is in any graph in the configuration model, *i.e.* either $i = j$ and there is one walk of length 0, or $i \neq j$ and there is none. Notice also that $W_{1,ij}$ is exactly equal to the inverse of the number of expected walks in the configuration model. Moreover, since A is non-negative, one may write, using Theorem 2.4 (Perron-Frobenius), that

$$A^l \sim \rho^l \mathbf{v}\mathbf{w}^T \ .$$

Or, in other words, independently from the origin and termination, there are asymptotically ρ times more walks of length $l+1$ than of length l. This reasoning holds for the graphs considered in the associate configuration model since their adjacency matrices are row and column permutations of A. Hence, the expected number of walks of length l from i to j in the configuration model asymptotically grows as $\frac{|\Gamma(i)||\Gamma(j)|}{|E|}\ \rho^{l-1}$, hence the expression of the $W_{l,ij}$. Note that $S^{\text{Leicht}''}$ can be rewritten as follows

$$S^{\text{Leicht}''} = I_m + |E(A)|\ D\ A\ \left(I_m - \frac{A}{\rho}\right)^{-1}\ D\ ,$$

where $D = \text{diag}\left(\frac{1}{|\Gamma(1)|}, \cdots, \frac{1}{|\Gamma(m)|}\right)$.

3.2.5 Stochastic Scalings

As mentioned in the introduction of Section 3.2.1, the node-to-node similarity measures $S^{\cap\Gamma^r}$, and $S^{\overset{r}{\rightarrow}}$ tend to give higher similarity scores to highly connected nodes. As a result of this, two distinct nodes could be more similar than one node is similar to itself. This effect is commonly avoided by weighting (usually uniformly) the out-edges (or in-edges) of a node so that their sum is equal to 1. The similarity scores are then computed by replacing A by its so called row-stochastic scaling or column-stochastic scaling.

In [ZLZ09], Zhou *et al.* propose

$$\left[S^{\text{Zhou}'}\right]_{i,j} := \sum_{k\in\Gamma(i)\cap\Gamma(j)} \frac{1}{|\Gamma(i)|\cdot|\Gamma(k)|} = \left[\,\text{rss}(A)^{\,2}\right]_{i,j}\ . \tag{3.33}$$

Note that the similarity score $\left[S^{\text{Zhou}'}\right]_{i,j}$ can be interpreted as the probability to end at node j when one starts a 2-step random walk at node i. Some other measures lie in between $S^{\text{Zhou}'}$ and $S^{\cap\Gamma^1}$:

- Zhou *et al.* also propose

$$\left[S^{\text{Zhou}''}\right]_{i,j} := \sum_{k\in\Gamma(i)\cap\Gamma(j)} \frac{1}{|\Gamma(k)|} = \left[A\ \text{rss}(A)\,\right]_{i,j}\ , \tag{3.34}$$

- In [AA03], Adamic and Adar consider a social network in which individuals are linked to properties based on information collected on their homepages. The more individuals have properties in common, the more they are similar. Moreover, properties unique to a few individuals are weighted more than commonly occurring properties. This leads them to choose a definition in between $S^{\text{Zhou}''}$ and $S^{\cap\Gamma^1}$:

$$[S^{\text{Adamic}}]_{i,j} := \sum_{k\in\Gamma(i)\cap\Gamma(j)} \frac{1}{\log|\Gamma(k)|} \ . \tag{3.35}$$

For all nodes i and j with $i \neq j$, we have

$$\left[S^{\text{Zhou}'}\right]_{i,j} \leq \left[S^{\text{Zhou}''}\right]_{i,j} \leq [S^{\text{Adamic}}]_{i,j} \leq \left[S^{\cap\Gamma^1}\right]_{i,j} \ .$$

In [JW02], Jeh *et al.* define a similarity measure as the extremal point of Algorithm 1 with:

$$[S_0^{\text{Jeh}}]_{i,j} := \delta_{ij} \ , \quad \text{and} \quad [S_t^{\text{Jeh}}]_{i,j} := \delta_{ij} + \frac{(1-\delta_{ij})\ \alpha}{|\Gamma(i)|\ |\Gamma(j)|} \sum_{\substack{k\in\Gamma(i)\\ l\in\Gamma(j)}} \left[S_{t-1}^{\text{Jeh}}\right]_{k,l} \ ,$$

or in matrix formulation

$$S_0^{\text{Jeh}} := I \ , \quad \text{and} \quad S_t^{\text{Jeh}} := I + \alpha \ \text{off}\left(\text{rss}(A)\, S_{t-1}^{\text{Jeh}}\, \text{rss}(A)^{\,T}\right) \ .$$

where α is a damping factor chosen sufficiently small. S^{Jeh} first initialize the self-similarity scores S_{ii} to 1, and further compute the crossed-similarity scores S_{ij} $(i \neq j)$ in the loop. Vectorizing the matrix formulation yields

$$\text{vec}(S_t^{\text{Jeh}}) = \text{vec}(I) + \alpha\ \text{vec}(\mathbf{1} - I) \odot \left(\left[\,\text{rss}(A)\ \otimes\ \text{rss}(A)\,\right]\ \text{vec}\left(S_{t-1}^{\text{Jeh}}\right)\right) \ .$$

The complexity of computing $f^{\text{Jeh}}(S)$ is of the order of $2\,|E(A)|\,m$ operations. One can notice that after one step

$$[S_1^{\text{Jeh}}]_{i,j} = \left[S^{\text{Leicht}'}\right]_{i,j} = \frac{|\Gamma(i)\cap\Gamma(j)|}{|\Gamma(i)|\,|\Gamma(j)|} \ , \quad \forall i \neq j.$$

The *P-rank* measure introduced by Zhao *et al.* in [ZHS09] extends

S^{Jeh} to directed networks using Algorithm 1 with $\left[S^{\text{Zhao}}_{\lambda,0}\right]_{i,j} := \delta_{ij}$ and

$$\left[S^{\text{Zhao}}_{\lambda,t}\right]_{i,j} := \lambda \left(\delta_{ij} + \frac{(1-\delta_{ij})\ \alpha}{|C(i)|\ |C(j)|} \sum_{\substack{k \in C(i) \\ l \in C(j)}} \left[S^{\text{Zhao}}_{\lambda,t-1}\right]_{k,l} \right) \tag{3.36}$$

$$+(1-\lambda) \left(\delta_{ij} + \frac{(1-\delta_{ij})\ \alpha}{|P(i)|\ |P(j)|} \sum_{\substack{k \in P(i) \\ l \in P(j)}} \left[S^{\text{Zhao}}_{\lambda,t-1}\right]_{k,l} \right) ,$$

3.2.6 Classification

We now detail in Table 3.2 (resp. Table 3.3) a number of properties of all undirected (resp. directed) node-to-node similarity measures that have been listed in this section.

A first important property is the so-called *symmetry*. $S(\cdot)$ is called *symmetric* if $S(A) = S(A)^T$, for every adjacency matrix A. In other words, a self-similarity measure is symmetric if $[S(A)]_{i,j}$, the self-similarity score between the node $i \in N(A)$ and the node $j \in N(A)$, is equal to $[S(A)]_{j,i}$, the self-similarity score between the node j and the node i.

Let $\mathcal{I}$ be a subset of $\mathbb{N}_{\leq m}$. The submatrix $[M_{ij}]_{i,j \in \mathcal{I}}$ of a matrix $M \in \mathbb{R}^{m \times m}$ is denoted $M_{\mathcal{I}}$.

Definition 3.6 *The radius of a self-similarity measure $S(\cdot)$ is the smallest $r \in \mathbb{N}$ such that, for all adjacency matrices A, A', for all $i, j \in N(A)$, $i', j' \in N(A')$, if the following statement in point 1 implies the one in point 2.*

1. *There exists an isomorphism from $G\left(A_{\mathcal{I}}\right)$ to $G\left(A'_{\mathcal{I}'}\right)$ where $\mathcal{I} = \Gamma^{\leq r}(i) \cup \Gamma^{\leq r}(j)$, and $\mathcal{I}' = \Gamma^{\leq r}(i') \cup \Gamma^{\leq r}(j')$, which maps the nodes i and j respectively onto the nodes i' and j'.*

2. $[S(A,B)]_{i,j} = [S(A',B')]_{i',j'}$.

3.2.7 Summary

We review here all node-to-node self-similarity measures that we have presented in this section.

Self-Similarity Measures for Undirected Graphs

- $\left[S^{\cap\Gamma^r}(A)\right]_{i,j}\left(=[S^{\overset{2r}{\rightarrow}}]_{i,j}\right) := |\Gamma^r(i)\cap\Gamma^r(j)|$.

  It tends to give higher similarity scores to highly connected nodes. It is usually only defined to help to introduce other similarity measures.

- $\left[S^{\overset{r}{\rightarrow}}(A)\right]_{i,j} := \left|\wp\left(i \overset{r}{\rightarrow} j\right)\right|$.

  It has the same advantages and disadvantages as $S^{\cap\Gamma^r}$.

- $S^{\text{Zhou}} := S^{\overset{2}{\rightarrow}} + \alpha S^{\overset{3}{\rightarrow}}$ [ZLZ09],

  Advantage:

  - The number of different scores in $S^{\text{Zhou}}(A)$ is usually larger than the ones in $S^{\overset{2}{\rightarrow}}(A)$.

  Disadvantage:

  - It tends to give higher similarity scores to highly connected nodes.
  - It is not weakly structural.

- S^{Leicht} [LHN06] is equal to the extremal point of Algorithm 1 with

$$[S_0^{\text{Leicht}}]_{i,j} = \delta_{ij}, \text{ and } [S_t^{\text{Leicht}}]_{i,j} = \delta_{ij} + \alpha \sum_{k\in\Gamma(i)} \left[S_{t-1}^{\text{Leicht}}\right]_{k,j} ,$$

  or $S^{\text{Leicht}} = S^{\overset{0}{\rightarrow}} + \alpha S^{\overset{1}{\rightarrow}} + \alpha^2 S^{\overset{2}{\rightarrow}} + \alpha^3 S^{\overset{3}{\rightarrow}} + \cdots$.

  Advantage:

  - Its definition is simple which makes its mathematical analysis easy and interesting.

- It is meaningful if similar means interconnected.
- Usually, nearly all scores are different.

Disadvantage:

- It is not weakly structural.

- $S^{\text{Estrada}} = \frac{S^{\overset{0}{\rightarrow}}}{0!} + \frac{S^{\overset{1}{\rightarrow}}}{1!} + \frac{S^{\overset{2}{\rightarrow}}}{2!} + \frac{S^{\overset{3}{\rightarrow}}}{3!} + \cdots$, [EH10]
  It has the same advantages and disadvantages as S^{Leicht}.

- $\left[S^{\text{Jaccard}}(A)\right]_{i,j} := \frac{|\Gamma(i) \cap \Gamma(j)|}{|\Gamma(i) \cup \Gamma(j)|}$ [Jac01].
  Advantage:
  - It is strongly structural.
  - Usually, there are several different scores in contrast with $S^{\cap\Gamma^1}$.

- $\left[S_p^{\cap\Gamma^1}(A)\right]_{i,j} = \frac{|\Gamma(i) \cap \Gamma(j)|}{M_p\left(|\Gamma(i)|, |\Gamma(j)|\right)}$, where
  - $p = 0$ yields *Salton* (or *Cosine*) *Index* [SM83],
  - $p = 1$ yields *Sørensen-Dice Index* [Sor48],
  - $p = \infty$ yields *Braun-Blanquet Index* [BB32],
  - $p = -1$ yields *Second Kulczynski Index*,
  - $p = -\infty$ yields *Simpson* (or *overlap*) *Index*,

  They have the same advantages and disadvantages as S^{Jaccard}, except $S_{p=-\infty}^{\cap\Gamma^1}$ which is only weakly structural. The higher (resp. smaller) is p, the smaller (resp. higher) are the entries in the columns and rows to which belong the highest (resp. smallest) diagonal entries. The most commonly used are $S_{p=0}^{\cap\Gamma^1}$ and $S_{p=\infty}^{\cap\Gamma^1}$, or eventually S^{Jaccard}.

- $\left[S^{\text{corr.}}\right]_{i,j} := \frac{\text{covar}(A_{i\cdot}, A_{j\cdot})}{\sqrt{\text{var}(A_{i\cdot}) \cdot \text{var}(A_{j\cdot})}}$ [AJR03].
  It has the same advantages and disadvantages as $S_{p=0}^{\cap\Gamma^1}$.

- $[S^{\text{Ravasz}}]_{i,j} := \dfrac{[S^{\cap\Gamma^1} + S^{\overset{1}{\rightarrow}}]_{i,j}}{\min\left([S^{\cap\Gamma^1} + S^{\overset{1}{\rightarrow}}]_{i,i}, [S^{\cap\Gamma^1} + S^{\overset{1}{\rightarrow}}]_{j,j}\right)}$ [RSM+02].

  It has the same advantages and disadvantages as $S^{\cap\Gamma^1}_{p=-\infty}$, except it is not weakly structural.

- $\left[S^{\cap\Gamma^1}_{w=S_{ii}}(A)\right]_{i,j} := \dfrac{|\Gamma(i) \cap \Gamma(j)|}{|\Gamma(i)|}$ [SM83, p. 203].

  It has the same advantages and disadvantages as $S^{\cap\Gamma^1}_{p=0}$, except it is asymmetric.

- $\left[S^{\text{Leicht}'}\right]_{i,j} := \dfrac{|\Gamma(i) \cap \Gamma(j)|}{|\Gamma(i)| \cdot |\Gamma(j)|}$ [LHN06].

  It tends to give smaller similarity scores to highly connected nodes. We would recommend to use $S^{\cap\Gamma^1}_{p=0}$ instead.

- $\left[S^{\text{Leicht}''}\right]_{i,j} := \delta_{ij} + \sum_{l=1}^{\infty} \dfrac{1}{\frac{|\Gamma(i)||\Gamma(j)|}{|E|}\ \rho(A)^{l-1}}\ \left[A^l\right]_{i,j}$ [LHN06].

  It has the same advantages and disadvantages as S^{Leicht} except for the simplicity argument.

- $\left[S^{\text{Zhou}'}\right]_{i,j} := \sum_{k\in\Gamma(i)\cap\Gamma(j)} \dfrac{1}{|\Gamma(i)| \cdot |\Gamma(k)|}$ [ZLZ09].

  It has a interpretation in terms of probability of presence of a random walker in the graph under consideration.

- $\left[S^{\text{Zhou}''}\right]_{i,j} := \sum_{k\in\Gamma(i)\cap\Gamma(j)} \dfrac{1}{|\Gamma(k)|}$ [ZLZ09].

  It is in between $S^{\text{Zhou}'}$ and $S^{\cap\Gamma^1}$.

- $[S^{\text{Adamic}}]_{i,j} := \sum_{k\in\Gamma(i)\cap\Gamma(j)} \dfrac{1}{\log|\Gamma(k)|}$ [AA03].

  It is also in between $S^{\text{Zhou}'}$ and $S^{\cap\Gamma^1}$.

- S^{Jeh} [JW02] is equal to the extremal point of Algorithm 1 with

$$[S_0^{\text{Jeh}}]_{i,j} := \delta_{ij}, \text{ and } [S_t^{\text{Jeh}}]_{i,j} := \delta_{ij} + \frac{(1-\delta_{ij})\ \alpha}{|\Gamma(i)|\ |\Gamma(j)|} \sum_{\substack{k\in\Gamma(i)\\ l\in\Gamma(j)}} \left[S_{t-1}^{\text{Jeh}}\right]_{k,l}\ .$$

It has the same advantages and disadvantages as $S^{\text{Leicht}''}$.

Note that α is a damping factor.

Self-Similarity Measures for Directed Graphs

Note that these definitions have the same advantages and disadvantages as their respective undirected counterparts.

- $\left[S^{\cap\Gamma^r}(A)\right]_{i,j} := \frac{1}{2}\left[A^r\left(A^T\right)^r\right]_{i,j} + \frac{1}{2}\left[\left(A^T\right)^r A^r\right]_{i,j}$
- $\left[S^{\overset{r}{\rightarrow}}(A)\right]_{i,j} := \left[A^r\right]_{i,j}$,
- $[S^{\text{Jaccard}}(A)]_{i,j} := \frac{\left[S^{\cap\Gamma^1}\right]_{i,j}}{\left[S^{\cap\Gamma^1}\right]_{i,i} + \left[S^{\cap\Gamma^1}\right]_{j,j} - \left[S^{\cap\Gamma^1}\right]_{i,j}}$,
- $\left[S_p^{\cap\Gamma^1}(A)\right]_{i,j} := \frac{\left[S^{\cap\Gamma^1}\right]_{i,j}}{M_p\left(\left[S^{\cap\Gamma^1}\right]_{i,i}, \left[S^{\cap\Gamma^1}\right]_{j,j}\right)}$,
- $\left[S_{w=S_{ii}}^{\cap\Gamma^1}(A)\right]_{i,j} := \frac{\left[S^{\cap\Gamma^1}\right]_{i,j}}{\left[S^{\cap\Gamma^1}\right]_{i,i}}$ [SM83, p. 203], and
- S_λ^{Zhao} [ZHS09] is equal to the extremal point of Algorithm 1 with $\left[S_{\lambda,0}^{\text{Zhao}}\right]_{i,j} := \delta_{ij}$ and the reinforcement step of Equation (3.36).

	Sym.	*WI*	*SI*	*Radius*	*Complexity*
$S^{\text{Blondel}}(A,B)$	✓	✗	✗	$t_{\max}$	$4\,(\lvert E(A)\rvert\, n + \lvert E(B)\rvert\, m)\;\; t_{\max}$
$S_\alpha^{\Pi\Sigma\text{-}r}(A,B)$	✓	✗	✗	r	$2r\;\; (mn + \lvert E(A)\rvert + \lvert E(B)\rvert)$
$S_\alpha^{\Sigma\Pi\text{-}r}(A,B)$	✓	✗	✗	r	$2rmn\;\; [m\log(m) + n\log(n)]$ $+rm\,\lvert E(A)\rvert + rn\,\lvert E(B)\rvert$
$S_\alpha^{\text{P}\text{-}r}(A,B)$	✓	✗	✗	r	$mn\;\; \left(\overline{m}^3 + 2(r+1)\right)$ $+rm\,\lvert E(A)\rvert + rn\,\lvert E(B)\rvert$
$S^{\text{Edit}}(A,B)$	✓	✗	✗	∞	$\frac{\overline{m}!\;\underline{E}^2}{(\overline{m}-\underline{m})!}$
$S^{\text{Blondel}'}(A,B)$	✓	✓	✗	$t_{\max}$	cf. $S^{\text{Blondel}}(A,B)$
$S_\alpha^{\text{Cooper}\text{-}r}(A,B)$	✓	✓	✗	r	cf. $S_\alpha^{\Pi\Sigma\text{-}r}(A,B)$
$S_\alpha^{\text{Cooper}'\text{-}r}(A,B)$	✓	✓	✗	r	cf. $S_\alpha^{\text{P}\text{-}r}(A,B)$
$S_\alpha^{\text{Cooper}''\text{-}r}(A,B)$ *with* $r < d(A)$	✓	✓	✗	r	cf. $S_\alpha^{\text{P}\text{-}r}(A,B)$
$S_\alpha^{\text{Cooper}''\text{-}r}(A,B)$ *with* $r \geq d(A)$	✓	✓	?	r	cf. $S_\alpha^{\text{P}\text{-}r}(A,B)$
$S^{\text{Edit}'}(A,B)$	✓	✓	✓	∞	cf. $S^{\text{Edit}}(A,B)$
$S^{\text{P-Rank}}(A,B)$	✓	✗	✗	$t_{\max}$	$(\lvert E(A)\rvert\,\lvert E(B)\rvert + mn)\;\; t_{\max}$
$S^{\text{Melnik}}(A,B)$	✓	✗	✗	1	$\lvert E(A)\rvert\,\lvert E(B)\rvert$

Table 3.1: ✓ *(resp.* ✗*) means that the property of the corresponding column is true (resp. false) for the similarity of the corresponding row. "Sym." means that the similarity is symmetric* (cf. *page 55). "WI" and "SI" respectively means that the similarity is weakly and strongly isomorphic (see Definition 3.2) for strongly connected graphs. The "Radius" of a similarity is defined in Definition 3.5,* $t_{\max}$ *is the maximum number of iterations in Algorithm 1,* $\overline{m} := \max(m,n)$, $\underline{m} := \min(m,n)$, $\underline{E} := \min(\lvert E(A)\rvert, \lvert E(B)\rvert)$, *and* $d(A)$ *is the diameter of the graph* $G(A)$, i.e. *the length of the longest shortest path between any nodes of* $G(A)$.

	Sym.	*WS*	*SS*	*Radius*	$\frac{\text{Complexity}}{m\lvert E(A)\rvert}$
$S^{\cap\Gamma^{\forall}}(A)$	✓	✗	✗	r	$2r-1$
$S^{\overset{r}{\to}}(A)$	✓	✗	✗	$\lceil r/2 \rceil$	$r-1$
$S^{\text{Zhou}}(A)$	✓	✗	✗	2	cf. $S^{\cap\Gamma^3}(A)$
$S^{\text{Leicht}}(A)$	✓	✗	✗	$\lceil t_{\max}/2 \rceil$	$t_{\max}$
$S^{\text{Estrada}}(A)$	✓	✗	✗	$\lceil t_{\max}/2 \rceil$	$t_{\max}$
$S^{\text{Jaccard}}(A)$	✓	✓	✓	1	cf. $S^{\cap\Gamma^1}(A)$
$S^{\cap\Gamma^1}_{p\in]-\infty,\infty]}(A)$	✓	✓	✓	1	cf. $S^{\cap\Gamma^1}(A)$
$S^{\text{corr.}}(A)$	✓	✓	✓	1	cf. $S^{\cap\Gamma^1}(A)$
$S^{\text{Ravasz}}(A)$	✓	✗	✗	1	cf. $S^{\cap\Gamma^1}(A)$
$S^{\cap\Gamma^1}_{p=-\infty}(A)$	✓	✓	✗	1	cf. $S^{\cap\Gamma^1}(A)$
$S^{\cap\Gamma^1}_{w=S_{ii}}(A)$	✗	✓	✗	1	cf. $S^{\cap\Gamma^1}(A)$
$S^{\text{Leicht}'}(A)$	✓	✗	✗	1	cf. $S^{\cap\Gamma^1}(A)$
$S^{\text{Leicht}''}(A)$	✓	✗	✗	$\lceil t_{\max}/2 \rceil$	cf. $S^{\text{Leicht}}(A)$
$S^{\text{Zhou}'}(A)$	✗	✗	✗	2	cf. $S^{\cap\Gamma^1}(A)$
$S^{\text{Zhou}''}(A)$	✓	✗	✗	2	cf. $S^{\cap\Gamma^1}(A)$
$S^{\text{Adamic}}(A)$	✓	✗	✗	2	cf. $S^{\cap\Gamma^1}(A)$
$S^{\text{Jeh}}(A)$	✓	✗	✗	$t_{\max}$	$2t_{\max}$

*Table 3.2: ✓ (resp. ✗) means that the property of the corresponding column is true (resp. false) for the undirected node-to-node similarity of the corresponding row. "Sym." means that the self-similarity is symmetric (*cf. *page 73). "WS" and "SS" respectively means that the similarity is weakly and strongly structural (see Definition 3.2). The "Radius" of a self-similarity is defined in Definition 3.6, $t_{\max}$ is the maximum number of iterations in Algorithm 1.*

	Sym.	*WS*	*SS*	*Radius*	$\frac{\text{Complexity}}{m\lvert E(A)\rvert}$
$S^{\cap\Gamma^r}(A)$	✓	✗	✗	r	$2r-1$
$S^{\overset{r}{\to}}(A)$	✓	✗	✗	$\lceil r/2 \rceil$	$r-1$
$S^{\text{Jaccard}}(A)$	✓	✓	✓	1	cf. $S^{\cap\Gamma^1}(A)$
$S^{\cap\Gamma^1}_{p\in]-\infty,\infty]}(A)$	✓	✓	✓	1	cf. $S^{\cap\Gamma^1}(A)$
$S^{\cap\Gamma^1}_{p=-\infty}(A)$	✓	✗	✗	1	cf. $S^{\cap\Gamma^1}(A)$
$S^{\cap\Gamma^1}_{w=S_{ii}}(A)$	✗	✓	✗	1	cf. $S^{\cap\Gamma^1}(A)$
$S^{\text{Zhao}}_{\lambda}(A)$	✓	✗	✗	$t_{\max}$	$4t_{\max}$

Table 3.3: ✓ *(resp.* ✗*) means that the property of the corresponding column is true (resp. false) for the directed node-to-node similarity of the corresponding row. "Sym." means that the self-similarity is symmetric (*cf. *page 73). "WS" and "SS" respectively means that the similarity is weakly and strongly structural (see Definition 3.2). The "Radius" of a self-similarity is defined in Definition 3.6,* $t_{\max}$ *is the maximum number of iterations in Algorithm 1.*

Chapter 4

Graph Blockmodeling

In this chapter, we first introduce the concept of Blockmodeling and further define the notion of relevance of a Blockmodel with respect to a given quality function. We then present and analyze existing Blockmodel quality measures, along with their strengths and weaknesses, and propose a novel Blockmodel quality measure. We show that the maximization of the quality measures that we consider can be translated in terms of trace-maximization problems. Finally, we present and investigate an heuristic algorithm that tries to find the most relevant Blockmodels with respect to the different quality functions.

The goal of Blockmodeling is to reduce a large, potentially incoherent graph to a small comprehensible structure. More precisely, given a graph $G(A) = (N(A), E(A))$, Blockmodeling consists in finding

- σ, an indexed partition of $G(A)$, and
- $G(B) := (\mathrm{codom}(\sigma), E(B))$, a graph defined on the indexes of the blocks of σ,

that best satisfies the two following concurrent objectives,

- σ has a small number of blocks, and
- it requires one to make few modification to $E(A)$ in order to turn σ into a regular indexed partition with $G(B)$ as image graph.

The pair (B, σ) is actually called a Blockmodel of $G(A)$. The graph $G(B)$ is a small image graph that hopefully emphasizes relevant role interconnections that reveal the underlying structure in $G(A)$. An illustration of a Blockmodel is shown in Figure 4.1. The order of a Blockmodel is the order of its image graph. $\mathcal{BM}(n)$ is the set of all Blockmodels of order n. In this chapter, an indexed partition $\sigma : N(A) \to \mathbb{N}_{\leq n}$ will also be represented by its matrix representation, *i.e.*, $S := M(\sigma)$ is a matrix with $|N(A)|$ rows and n columns, such that $S(i, j) = 1$ if $\sigma(i) = j$, 0 otherwise. *E.g.*, let us consider G, the graph of Figure 2.2. The indexed partition of the Blockmodel presented in Figure 4.1 can equivalently be represented by the function σ or its matrix representation S given by

$$\begin{array}{ll} \sigma(1) = 1, & \sigma(7) = 2, \\ \sigma(2) = 2, & \sigma(8) = 1, \\ \sigma(3) = 3, & \sigma(9) = 4, \\ \sigma(4) = 4, & \sigma(10) = 1, \\ \sigma(5) = 4, & \sigma(11) = 4, \\ \sigma(6) = 3, & \sigma(12) = 4, \end{array} \quad \text{and} \quad S = \begin{bmatrix} 1&0&0&0 \\ 0&1&0&0 \\ 0&0&1&0 \\ 0&0&0&1 \\ 0&0&0&1 \\ 0&0&1&0 \\ 0&1&0&0 \\ 1&0&0&0 \\ 0&0&0&1 \\ 1&0&0&0 \\ 0&0&0&1 \\ 0&0&0&1 \end{bmatrix}.$$

4.1 Relevant Blockmodel

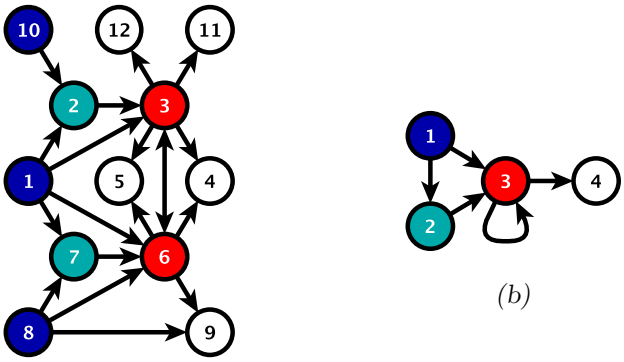
Blockmodel quality measures assign a real value to any Blockmodel that determines its quality. More precisely, given an adjacency matrix $A \in \mathbb{R}^{m \times m}$, a Blockmodel quality measure,

$$Q_A : \bigcup_n \mathcal{BM}(n) \to \mathbb{R} : (B, S) \mapsto Q_A(B, S) \ ,$$

should be high if the Blockmodel (B, S) highly satisfies (according to particular criteria) the concurrent objectives of Blockmodeling, namely

- S has a small number of blocks, and
- it requires one to make few modification to $E(A)$ in order to turn S into a regular indexed partition with $G(B)$ as image graph.

Blockmodeling tries to find a Blockmodel of high quality which, in practice, is usually done as follows.



(a) (b)

Figure 4.1: Blockmodeling consists in finding an indexed partition σ of the graph in (a), denoted $G(A)$, and an image graph $G(B) := (\mathrm{codom}(\sigma), E(B))$ defined on the indexes of the blocks of σ that hopefully emphasizes relevant role interconnections that reveal the underlying structure in $G(A)$. This figure shows an example of Blockmodel of $G(A)$, the indexed partition of $G(A)$ is defined, as emphasized by the colors, by $\sigma^{-1}(1) = \{1, 8, 10\}$, $\sigma^{-1}(2) = \{2, 7\}$, $\sigma^{-1}(3) = \{3, 6\}$, and $\sigma^{-1}(4) = \{4, 5, 9, 11, 12\}$, whereas the image graph $G(B)$ is shown in (b). This Blockmodel emphasizes correctly the underlying structure in $G(A)$, since it only requires to make few modification to $E(A)$ in order to turn σ into a regular indexed partition with $G(B)$ as image graph. Indeed, one only needs to remove the edge $(8, 9)$ and to add an edge from the node 10 to any node in $\sigma^{-1}(3)$ which are colored in red.

1. Fix the number of blocks to a suitable small value, *e.g.*

$$n \in \left\{2, 3, 4, \cdots, 8, \cdots, \frac{m}{10}\right\} .$$

2. Find a Blockmodel of order n that maximizes a given quality function, *i.e.* find

$$(B^*, S^*) \in \underset{(B,S)\in\mathcal{B}m(n)}{\mathrm{argmax}} \; Q_A(B, S) .$$

As a consequence of this, Blockmodel quality measures are usually designed to compare Blockmodels of the same order. Eventhough they are usually inadequate for comparing Blockmodels of different order, one can conveniently scale the quality of a Blockmodel of order n by its expected maximal value with respect to a null model associated to A, which may be formalized as follows.

Problem 4.1 *Given*

- $A \in \mathbb{R}^{m \times m}$*, an adjacency matrix,*
- $\mathcal{A}$*, a set of adjacency matrices generated by a null model*[1] *associated with* A*, and*
- *a Blockmodel quality function*

$$Q_A : \bigcup_n \mathcal{BM}(n) \to \mathbb{R} : (B, S) \mapsto Q_A(B, S) \ .$$

Find a Blockmodel (B^*, S^*) *in*

$$\underset{(B,S)\in\Omega}{\operatorname{argmax}} \ Q_A(B, S) - Q_{\mathbb{E}}(\mathcal{A}, \operatorname{order}(B, S)) \ ,$$

with

$$Q_{\mathbb{E}}(\mathcal{A}, n) = \mathbb{E}_{A' \in \mathcal{A}} \left[\max_{(B', S') \in \mathcal{BM}(n) \cap \Omega} Q_{A'} \left(B', S' \right) \right] \ ,$$

and where Ω *is a set of admissible Blockmodels.*

The set of admissible Blockmodels, Ω, is often chosen equal to the set of Blockmodels of order n, that is reasonably small, since

- the purpose of Blockmodeling is to summarize the structure of a large incomprehensible graph in a small comprehensible image graph,
- the number of coloration in n colors of a set of order m is given by $S(m, n)$, the Stirling number of the second kind [Sti30, AS64, Maz09], explicitly computed as

$$S(m, n) = \frac{1}{n!} \sum_{j=0}^{n} (-1)^j \binom{n}{j} (n - j)^m \ ,$$

  which yields $S(m, 1) = S(m, m) = 1$, and

[1] Null models are pattern-generating models that deliberately ignore a mechanism of interest, and allow for randomization tests of data.

$S(m,n)$	$n=2$	$n=3$	$n=4$	$\max_n S(m,n)$	$n=m-1$
$m=2$	1			1	1
$m=3$	3	1		3	3
$m=4$	7	6	1	7	6
$m=5$	15	25	10	25	10
$m=6$	31	90	65	90	15
$m=7$	63	301	350	350	21
$m=8$	127	966	1701	1701	28
$m=9$	255	3025	7770	7770	36
$m=10$	511	9330	34105	42525	45
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$m=20$	$5.2\ 10^5$	$5.8\ 10^8$	$4.5\ 10^{10}$	$1.5\ 10^{13}$	210
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$m=100$	$6.3\ 10^{29}$	$8.5\ 10^{46}$	$6.6\ 10^{58}$	$7.7\ 10^{114}$	4950.

The Stirling numbers of the second kind grow exponentially with m. However, when n is close to 1 (or m), the rate of growth remains small in comparison with the rate of growth of the midrange values of n, *i.e.* $1 \ll n \ll m$.

- if Ω is equal to the set of Blockmodels of a fixed order n, then the expected value in the maximization in Problem 4.1 is a constant and may hence be ignored.

For all the reasons cited above, we set $\Omega = \mathcal{BM}(n)$ from now on, unless explicitly stated.

4.2 Blockmodel Quality Measures

In this section, we present and analyze existing Blockmodel quality measures, along with their strengths and weaknesses, and finally propose a novel Blockmodel quality measure.

In this section, we consider the following node-to-node similarity measures:

4.2.1 Similarity Clustering

Given $A \in \mathbb{R}^{m \times m}$, an adjacency matrix and $Sim(\cdot)$, a node-to-node self-similarity measure, the quality of a Blockmodel may be evaluated by giving reward

- if the nodes i and j in $N(A)$ belong to the same block of the indexed partition of the Blockmodel and $[Sim(A)]_{i,j}$ and $[Sim(A)]_{j,i}$ are large, or
- if the nodes i and j in $N(A)$ belong to different block of the indexed partition of the Blockmodel and $[Sim(A)]_{i,j}$ and $[Sim(A)]_{j,i}$ are small.

Let $G_w(W) = (N, w)$ be a weighted graph whose adjacency matrix is W. Let N_1, N_2 be two subsets of vertices. The size of the cut between N_1 and N_2 is defined as the sum of the weights of the edges starting from a node in N_1 and ending at a node in N_2 (see [SM97, AG06]), *i.e.*

$$Cut_W(N_1 \to N_2) = \sum_{i \in N_1, j \in N_2} W_{ij} \ .$$

And, the size of the cut of N_1 is defined as $Cut_W(N_1) := Cut(N_1 \to N \setminus N_1)$. Let now σ be a partition of N. The size of the cut into σ is defined as the sum of the sizes of the cuts of its clusters, *i.e.*

$$Cut_W(\sigma) := \sum_i Cut\left(\sigma^{-1}(i)\right) \ .$$

Let $G\left(Sim(A)\right)$ be the weighted graph whose adjacency matrix is $Sim(A)$. The lower the size of the cut of $G\left(Sim(A)\right)$ into the partition

σ of the Blockmodel is, the better is its quality. Hence, a natural Blockmodel quality measure may be defined as the sum of the weights of the uncut edges in the partition σ, *i.e.*

$$Q^{\text{Uncut}}_{Sim(A)}(S) := \sum_{i,j} [Sim(A)]_{i,j} - Cut_{Sim(A)}(\sigma) = \operatorname{tr}\big(S^T Sim(A) S\big) \ .$$

Let the quality function of Problem 4.1 be given by $Q_A(B,S) = Q^{\text{Uncut}}_{Sim(A)}(S)$, which reduces that problem to the following.

Problem 4.2 *Given $M = Sim(A) \in \mathbb{R}^{m\times m}$, and $n \in \mathbb{N}$, find a partition S^* in*

$$\operatorname*{argmax}_{S\in\mathcal{P}(m,n)} \operatorname{tr}\big(S^T M S\big) \ .$$

The quality $Q^{\text{Uncut}}_{Sim(A)}(S)$ tends to give higher quality scores to partitions with one large cluster and several small ones. This effect is commonly avoided using weights. In [CB10], Cooper *et al.* build a Blockmodel based on a normalized cut of $G\,(Sim(A))$. The sum of the weights of the uncut edges in $\sigma^{-1}(i)$ (the i^{th} cluster) – which is equal to $Cut_{Sim(A)}\left(\sigma^{-1}(i) \to N\right) - Cut_{Sim(A)}\left(\sigma^{-1}(i)\right)$ – is weighted by the size of the cut between $\sigma^{-1}(i)$ and all vertices – which is equal to $Cut_{Sim(A)}\left(\sigma^{-1}(i) \to N\right)$ – , *i.e.*

$$\begin{aligned} Q^{\text{N-Uncut}}_{Sim(A)}(S) &:= \sum_{i=1}^{n} \frac{Cut_{Sim(A)}\left(\sigma^{-1}(i) \to N\right) - Cut_{Sim(A)}\left(\sigma^{-1}(i)\right)}{Cut_{Sim(A)}\left(\sigma^{-1}(i) \to N\right)} \\ &= \operatorname{tr}\big(\operatorname{rss}\big(S^T Sim(A) S\big)\big) \ . \end{aligned}$$

Let now the quality function of Problem 4.1 be given by $Q_A(B,S) = Q^{\text{N-Uncut}}_{Sim(A)}(S)$, which reduces that problem to the following.

Problem 4.3 *Given $M = Sim(A) \in \mathbb{R}^{m\times m}$, and $n \in \mathbb{N}$, find a partition S^* in*

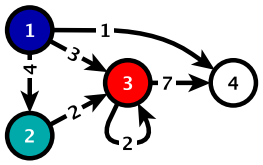
$$\operatorname*{argmax}_{S\in\mathcal{P}(m,n)} \operatorname{tr}\big(\operatorname{rss}\big(S^T M S\big)\big) \ .$$

Note that the image graph does not influence the quality functions in Problems 4.2 and 4.3. We hereafter propose two methods to derive an image graph based on a given partition. Problems 4.2 and 4.3 are known

as the matrix clustering problems, and have been studied extensively in the literature specially in the image segmentation domain, and can now be performed with any of a variety of methods. The number of clusters is usually considered as constant, which is equivalent to choosing $\Omega = \mathcal{Bm}(n)$ (with n the number of clusters) in Problem 4.1. *E.g.*, in [CB10], Cooper *et al.* use a spectral algorithm based on a Multiple Normalized Cut [SM97, AG06]. Note that, among the partitions in 4 blocks, the one of the graph $G(A)$ in Figure 4.1 is the unique maximizer of $Q^{\text{N-Uncut}}_{Sim(A)}(S)$ with $Sim(A) = S^{\text{Cooper-2}}_{\alpha=0.5}(A, A)$.

In [CB10], Cooper *et al.* do not look for an unweighted image graph, they simply give a weighted image graph in which the weights are equal to the sum of edges between the nodes of the corresponding blocks of the partition σ, *e.g.*, let us consider the graph $G(A)$ in Figure 4.1 and its associated indexed partition, σ. The weighted adjacency matrix of the image graph proposed by Cooper is

$$B_w = S^T A S = \begin{bmatrix} 0 & 4 & 3 & 1 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 2 & 7 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad i.e. \quad$$



.

We now propose two methods to derive a binary image graph based on a partition σ.

The first method consists in saying that there is an edge between the node r and the node s in the image graph if the number of edges from $\sigma^{-1}(r)$ to $\sigma^{-1}(r)$ in $G(A)$ is higher than its expected value in a null model. Or in other words, given $\mathcal{A}$, a set of adjacency matrices generated by a null model associated with A, we have

$$B_{rs} = \begin{cases} 1, \text{ if } \displaystyle\sum_{(i,j)\in E_{rs}} A_{ij} > \sum_{(i,j)\in E_{rs}} \mathbb{E}_{A'\in\mathcal{A}} A'_{ij}, \\ \\ 0, \text{ otherwise,} \end{cases}$$

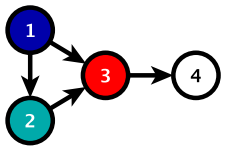
with $E_{rs} = \sigma^{-1}(r) \times \sigma^{-1}(s)$. Note that, as further highlighted in Lemma 4.1, this B is actually the sparsest one in

$$\underset{\tilde{B}\in\{0,1\}^{n\times n}}{\text{argmax}} \ \text{tr}\Big(S^T M_A^T S \ \tilde{B}\Big) \ ,$$

with $M_A = A - \mathbb{E}_{A' \in \mathcal{A}} A'$. Let us now choose the configuration model (see Section 3.2.4) as null model for A. We then have

$$\mathbb{E}_{A' \in \mathcal{A}} A'_{ij} = \frac{|C(i)|\,|P(j)|}{|E|} .$$

E.g., let us consider the graph $G(A)$ in Figure 4.1 and its associated indexed partition, σ. The image graph is directly derived by computing $S^T M_A S$, *i.e.*

$$S^T M_A S = \begin{bmatrix} 0 & \mathbf{2.3} & \mathbf{0.1} & -2.4 \\ 0 & -0.4 & \mathbf{1.2} & -0.8 \\ 0 & -1.9 & -1.3 & \mathbf{3.2} \\ 0 & 0 & 0 & 0 \end{bmatrix} , \quad \text{and } G(B) \text{ is } \quad 

.

The second method consists in saying that there is an edge between the node r and the node s in the image graph if the number of nodes in $\sigma^{-1}(r)$ that do not have a child in $\sigma^{-1}(s)$ plus the number of nodes in $\sigma^{-1}(s)$ that do not have a parent in $\sigma^{-1}(r)$ is smaller than the sum of the numbers of the ones that do, *i.e.* if

$$\begin{aligned} &\left|\left\{i \in \sigma^{-1}(r) \text{ s.t. } s \notin \sigma(C(i))\right\}\right| + \left|\left\{i \in \sigma^{-1}(s) \text{ s.t. } r \notin \sigma(P(i))\right\}\right| \\ &\quad < \left|\left\{i \in \sigma^{-1}(r) \text{ s.t. } s \in \sigma(C(i))\right\}\right| + \left|\left\{i \in \sigma^{-1}(s) \text{ s.t. } r \in \sigma(P(i))\right\}\right| , \end{aligned}$$

then $B_{rs} = 1$, otherwise $B_{rs} = 0$. And since the number of nodes in $\sigma^{-1}(r)$ that do not plus the one that do is equal to the total number of nodes in $\sigma^{-1}(r)$, we can rewrite this statement as

$$B_{rs} = \begin{cases} 1, \text{ if } \left[S^T \left(R_C - \frac{\mathbf{1}}{2}\right) + \left(R_P - \frac{\mathbf{1}}{2}\right)^T S\right]_{r,s} > 0 , \\ 0, \text{ otherwise,} \end{cases}$$

where R_C (resp. R_P) is the matrix representation of the function $\sigma(C(\cdot))$ (resp. $\sigma(P(\cdot))$) which can also be computed as follows

$$R_C = Is_{\neq 0}(AS) , \quad (\text{resp. } R_P = Is_{\neq 0}(A^T S)) ,$$

with

$$[Is_{\neq 0}(M)]_{i,j} := \begin{cases} 1, \text{ if } M_{ij} \neq 0 \\ 0, \text{ otherwise.} \end{cases}$$

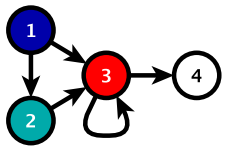
Note that, as further highlighted in Lemma 4.1, this B is actually the sparsest one in

$$\underset{\tilde{B}\in\{0,1\}^{n\times n}}{\operatorname{argmax}} \operatorname{tr}\left(\left(S^T\left(R_C-\frac{\mathbf{1}}{2}\right)+\left(R_P-\frac{\mathbf{1}}{2}\right)^T S\right)^T\left(\tilde{B}-c\mathbf{1}\right)\right), \quad (4.1)$$

for any $c \in \mathbb{R}$ [2]. *E.g.*, let us consider the graph $G(A)$ in Figure 4.1 and its associated indexed partition, σ. This method yields

$$S^T\left(R_C-\frac{\mathbf{1}}{2}\right)+\left(R_P-\frac{\mathbf{1}}{2}\right)^T S=\frac{1}{2}\cdot\begin{bmatrix}-6 & \mathbf{5} & \mathbf{3} & -4\\ -5 & -4 & \mathbf{4} & -7\\ -5 & -4 & \mathbf{4} & \mathbf{7}\\ -8 & -7 & -7 & -10\end{bmatrix},$$

and the graphical representation of the image graph is the following



.

4.2.2 Edge Alignment

The quality of a Blockmodel (B, σ) may be evaluated by giving

- reward if, for a pair of nodes $(i, j) \in N(A)^2$, the node i is connected to the node j in the same way the role of the node i is connected to the role of the node j, or in other words, if $A_{ij} = B_{\sigma_i\sigma_j}$, and

- penalty if, for a pair of nodes $(i, j) \in N(A)^2$, the node i is connected to the node j in the opposite way the role of the node i is connected to the role of the node j, or in other words, if $A_{ij} \neq B_{\sigma_i\sigma_j}$.

[2] Note that the term "$-c\mathbf{1}$" in Equation (4.1) may appear unnecessary but will actually be useful in Section 4.2.3 since Problem 4.6 is equivalent to solving Equation (4.1) with $c = 1/2$.

Let us use now the ideas introduced by Reichardt *et al.* in [RW07] and [RB06] and propose the following quality measure

$$\begin{aligned} Q_A^{\text{EA}}(B,\sigma) = \frac{1}{|E|}\sum_{i,j} & a_{ij}A_{ij}B_{\sigma_i\sigma_j} - b_{ij}(1-A_{ij})B_{\sigma_i\sigma_j} \\ & -c_{ij}A_{ij}(1-B_{\sigma_i\sigma_j}) \\ & +d_{ij}(1-A_{ij})(1-B_{\sigma_i\sigma_j}) \\ = \frac{1}{|E|}\sum_{i,j} & \left[(a_{ij}+b_{ij}+c_{ij}+d_{ij})A_{ij} - (b_{ij}+d_{ij})\right] B_{\sigma_i\sigma_j} \\ & -(c_{ij}+d_{ij})A_{ij} + d_{ij} \end{aligned} \tag{4.2}$$

with a_{ij}, b_{ij}, c_{ij} and d_{ij} positive parameters that do not depend on B and σ. Clearly, for each pair of nodes $(i,j) \in N(A)^2$,

- if $A_{ij} = B_{\sigma_i\sigma_j} = 1$, then the contribution of this pair to the quality measure is a_{ij}, and
- if $A_{ij} = B_{\sigma_i\sigma_j} = 0$, then the contribution of this pair to the quality measure is d_{ij},

which are both positive contributions conformally with the requirements. On the other hand, for each pair of nodes $(i,j) \in N(A)^2$,

- if $A_{ij} = 1$ and $B_{\sigma_i\sigma_j} = 0$, then the contribution of this pair to the quality measure is $-b_{ij}$, and
- if $A_{ij} = 0$ and $B_{\sigma_i\sigma_j} = 1$, then the contribution of this pair to the quality measure is $-c_{ij}$,

which are both negative contributions conformally with the requirements. The last terms of the quality, *i.e.* $-(c_{ij}+d_{ij})A_{ij} + d_{ij}$ does not depend on B and σ, and may hence be omitted since quality functions are essentially designed to compare Blockmodels among themselves. Without loss of generality, one can prove that this is $c_{ij} = d_{ij} = 0$, and rewrite the quality of the Blockmodel as

$$Q_A^{\text{EA}}(B,\sigma) = \frac{1}{|E|}\sum_{i,j}\left[(a_{ij}+b_{ij})\,A_{ij} - b_{ij}\right] B_{\sigma_i\sigma_j}\ . \tag{4.3}$$

Moreover, it is often desirable to choose the parameters a_{ij}, b_{ij}, c_{ij}, and d_{ij} such that

- all the entries of A have an equal contribution to the quality measure, which implies that $a_{ij} + b_{ij} + c_{ij} + d_{ij} = 1$, and
- the expected value over all image graphs of the contributions of present edges in $G(A)$ balance the ones of absent edges in $G(A)$, which implies that

$$\sum_{i,j} (a_{ij} + c_{ij})\, A_{ij} = \sum_{i,j} (b_{ij} + d_{ij})\ (1 - A_{ij}) \ ,$$

which simplify, using the previous arguments, to

$$a_{ij} + b_{ij} = 1 \ , \quad \text{and} \quad \sum_{i,j} A_{ij} = \sum_{i,j} b_{ij} \ .$$

Finally, one can require that the expected value of the quality function with respect to a null model associated to A is equal to 0. Choosing the configuration model (see Section 3.2.4) as null model for A yields

$$b_{ij} = \mathbb{E}_{A' \in \mathcal{A}} \left[A'_{ij} \right] = \frac{|C\,(i)\,|\ |P\,(j)\,|}{|E|} \ .$$

This quality measure may be written using matrices as follows

$$Q^{\text{EA}}_A(B, S) = \frac{1}{|E|}\ \text{tr}\big(S^T M_A^T S B\big) \ , \tag{4.4}$$

with $[M_A]_{i,j} := A_{ij} - |C\,(i)\,||P\,(j)\,|/|E|$. Notice that, with this particular choice of parameters, the matrix M_A is the Newman modularity matrix, and, moreover, if the image graph is an identity matrix, this quality measure is the Newman modularity [New06].

Notice that, if one sets $b_{ij} = d_{ij}$ and $a_{ij} = c_{ij}$ in Equation (4.2), then the quality function is then

$$Q^{\text{EA}'}_A(B, \sigma) = \frac{1}{|E|} \sum_{i,j} \left[2\,(a_{ij} + b_{ij})\, A_{ij} - 2 b_{ij} \right] \left(B_{\sigma_i \sigma_j} - \frac{1}{2} \right) \ , \tag{4.5}$$

which, using the earlier mentioned desirable requirements $2\,(a_{ij} + b_{ij}) = 1$, and $2b_{ij} = |C\,(i)|\,|P\,(j)|\,/\,|E|$, can be rewritten as the sum of $Q^{\text{EA}}_A(B, S)$ and a term that does not depend on the Blockmodel, *i.e.*

$$Q^{\text{EA}'}_A(B, S) = Q^{\text{EA}}_A(B, S) - \frac{1}{2|E|} \sum_{i,j} [M_A]_{i,j} \ .$$

Moreover, the desirable requirements also impose $\sum_{i,j} [M_A]_{i,j} = 0$, and eventually, we have $Q_A^{\text{EA}'}(B,S) = Q_A^{\text{EA}}(B,S)$.

Let now the quality function of Problem 4.1 be given by $Q_A(B,S) = Q_A^{\text{EA}}(B,S) = Q_A^{\text{EA}'}(B,S)$, which reduces that problem to the following.

Problem 4.4 *Given $M = M_A \in \mathbb{R}^{m\times m}$, and $n \in \mathbb{N}$, find a Blockmodel (B^*, S^*) in*

$$\underset{(B,S)\in\mathcal{B}m(n)}{\text{argmax}} \ \text{tr}\big(S^T M^T S \ B\big) \ ,$$

which can also be written as

$$\underset{(B,S)\in\mathcal{B}m(n)}{\text{argmax}} \ \text{tr}\big(S^T M^T S \ \ (B - \mathbf{1}/2)\big) \ .$$

We now show that, for a given S, one can directly find the set of image graphs B that maximize the quality function $Q_A^{\text{EA}}(B,S)$.

Lemma 4.1 *Given $C \in \mathbb{R}^{n\times n}$ and a $c \in \mathbb{R}$. The set*

$$\big\{B \in \{0,1\}^{n\times n} \ \text{s.t.} \ B_{ij} = 1, \ \text{if} \ C_{ij} > 0, \ \text{and} \ B_{ij} = 0, \ \text{if} \ C_{ij} < 0.\big\}$$

is equal to the set

$$\underset{B\in\{0,1\}^{n\times n}}{\text{argmax}} \ \text{tr}\big(C^T \, (B - c\mathbf{1})\big) \ .$$

Moreover,

$$\max_{B\in\{0,1\}^{n\times n}} \text{tr}\big(C^T \, (B - c\mathbf{1})\big) = \sum_{i,j} \max\left(C_{ij}, 0\right) - c \ C_{ij} \ .$$

Proof: the objective function can be rewritten as follows

$$\sum_{i,j} C_{ij} B_{ij} - c \sum_{i,j} C_{ij}.$$

The second term is a constant, and the first term is clearly maximal when, for all i, j, B_{ij} is equal to

- 1 if $C_{ij} > 0$, since $C_{ij} \cdot 1 > C_{ij} \cdot 0$,

- 0 or 1 if $C_{ij} = 0$, since $C_{ij} \cdot 1 = C_{ij} \cdot 0$, and
- 0 if $C_{ij} < 0$, since $C_{ij} \cdot 1 > C_{ij} \cdot 0$.

The maximal value is reached for any of these B. □

In view of Lemma 4.1, for a given S, the matrix $B(S) := Is_{>0}\left(S^T M_A S\right)$ is the sparsest matrix in

$$\underset{B\in\{0,1\}^{n\times n}}{\operatorname{argmax}} \; Q_A^{\text{EA}}(B,S) \;, \quad \text{also equal to} \quad \underset{B\in\{0,1\}^{n\times n}}{\operatorname{argmax}} \; Q_A^{\text{EA}'}(B,S) \;,$$

and

$$Q_A^{\text{EA}}(B(S),S) = \frac{1}{|E|}\sum_{i,j} \max\left(\left[S^T M_A S\right]_{i,j}, 0\right) \;, \quad \text{also equal to}$$

$$\begin{aligned} Q_A^{\text{EA}'}(B(S),S) &= \frac{1}{|E|}\sum_{i,j} \max\left(\left[S^T M_A S\right]_{i,j}, 0\right) - \frac{1}{2}\left[S^T M_A S\right]_{i,j} \\ &= \frac{1}{2|E|}\sum_{i,j} \left|S^T M_A S\right|_{ij} \end{aligned}$$

In [RW07], Reichardt *et al.* maximize $Q_A^{\text{EA}}(B,S)$ that they fortunately transform in $\frac{1}{|E|}\sum_{i,j}\left|S^T M_A S\right|_{ij}$ in order to get rid of the B, though they did not mention that doing so they were choosing the optimal choice for B for any given S.

And Problem 4.4 may be reduced to the following.

Problem 4.5 *Given $M = M_A \in \mathbb{R}^{m\times m}$, and $n \in \mathbb{N}$, find a partition S^* in*

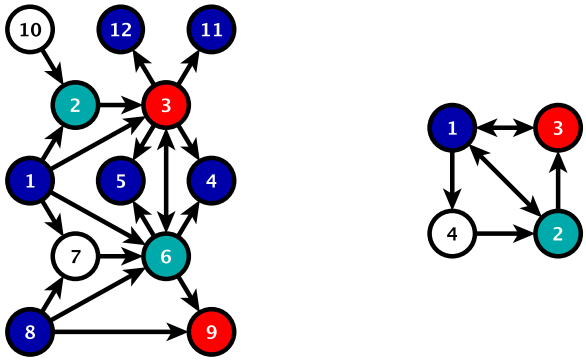
$$\underset{S\in\mathcal{P}(m,n)}{\operatorname{argmax}} \sum_{i,j} \left|S^T M S\right|_{ij} \;,$$

which can also be written as

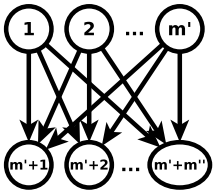
$$\underset{S\in\mathcal{P}(m,n)}{\operatorname{argmax}} \sum_{i,j} \max\left(\left[S^T M S\right]_{i,j}, 0\right) \;.$$

This method is sometimes relevant as shown in [RW07], but can nevertheless yield unexpected results. *E.g.*, the Blockmodel with highest

quality of the graph $G(A)$ in Figure 4.1 is given by the following partition and image graph.[3]



Those unexpected results are even worse when considering the graph whose graphical representation and adjacency matrix are respectively



, and $A = \begin{bmatrix} \mathbf{0}_{m',m'} & \mathbf{1}_{m',m''} \\ \mathbf{0}_{m'',m'} & \mathbf{0}_{m'',m''} \end{bmatrix}$

with $m', m'' \in \mathbb{N}$. Although, one could naturally think that a clear relevant Blockmodel for this graph would have the first m' nodes in the first partition and the m'' last ones in the second partition along with an image graph whose adjacency matrix would be $B = \left[\begin{smallmatrix} 0 & 1 \\ 0 & 0 \end{smallmatrix}\right]$, for this graph, we have $M_A = \mathbf{0}_{m'+m'',m'+m''}$, and any Blockmodel yields $Q_A^{\text{EA}}(B,S) = 0$. More generally, this effect affects all nodes without children or parents which will yield a row or a column filled with zeros in M_A. As a consequence, a node without children (resp. parents) can be assigned to a role with or without children (resp. parents) since the contribution due to the children (resp. parents) of the earlier is always zero.

Let us consider now another quality measure $Q_A^{\text{EA}''}(B,S)$ which we will prove is strongly related to $Q_A^{\text{EA}}(B,S)$ and will help us to emphasize this negative effect of $Q_A^{\text{EA}}(B,S)$. Let us consider that one performs a undirected random walk in the graph $G(A)$, *i.e.* when one ends up on the i^{th} node, one can choose, with an equal probability, to cross

[3] This result has been found using exhaustive search, *i.e.* all colorings have been tried.

an outgoing or incoming edge (respectively downstream and upstream). The quality $Q_A^{\mathrm{EA}''}$ is equal to the probability that one step of this random walk is coherent with the Blockmodel minus the expected value of this probability in the configuration model. More precisely, $Q_A^{\mathrm{EA}''}$ is equal to the difference between

- the sum for all pair of nodes (i, j) of the product between probability of being on node i, and the probability to walk an edge either from i to j or from j to i such that their roles are connected similarly in the image graph, and
- its expected value in the configuration model,

which can be mathematically written as follows

$$\begin{aligned} Q_A^{\mathrm{EA}''}(B,S) = \sum_{i,j} \pi_A^i \cdot \left(\pi_A^{i\to\cdot}\ \pi_A^{i\to j}\ \pi_B^{\sigma_i\to\sigma_j} + \pi_A^{i\leftarrow\cdot}\ \pi_A^{i\leftarrow j}\ \pi_B^{\sigma_i\leftarrow\sigma_j} \right) \\ -\mathbb{E}_{A'\in\mathfrak{a}} \sum_{i,j} \pi_{A'}^i \cdot \left(\pi_{A'}^{i\to\cdot}\ \pi_{A'}^{i\to j}\ \pi_B^{\sigma_i\to\sigma_j} + \pi_{A'}^{i\leftarrow\cdot}\ \pi_{A'}^{i\leftarrow j}\ \pi_B^{\sigma_i\leftarrow\sigma_j} \right) \end{aligned} \tag{4.6}$$

where

- π_A^i is the probability to be on the node i during an undirected random walk on $G(A)$,
- $\pi_A^{i\to\cdot}$ (resp. $\pi_A^{i\leftarrow\cdot}$) is the probability to choose an edge with the node i as origin (resp. termination) when uniformly choosing one edge among all edges with the node i as origin or termination,
- $\pi_A^{i\to j}$ (resp. $\pi_A^{i\leftarrow j}$) is the probability to choose an edge with the node i as origin (resp. termination) and the node j as termination (resp. origin) when uniformly choosing one edge among all edges with the node i as origin (resp. termination),
- $\pi_B^{\sigma_i\to\sigma_j}$ (resp. $\pi_B^{\sigma_i\leftarrow\sigma_j}$) is the probability that there is an edge from σ_i to σ_j (resp. from σ_i to σ_j) in $G(B)$.

In an undirected random walk, π_A^i is proportional to the degree of the node [Tre05], and hence

$$\pi_A^i = \pi_{A'}^i = \frac{|C(i)| + |P(i)|}{2\,|E(A)|}\ , \quad \forall A' \in \mathfrak{a}\ .$$

Moreover, we clearly have

$$\pi_A^{i\to\cdot} = \pi_{A'}^{i\to\cdot} = \frac{|C(i)|}{|C(i)| + |P(i)|} , \quad \forall A' \in \mathcal{A} ,$$

and

$$\pi_A^{i\leftarrow\cdot} = \pi_{A'}^{i\leftarrow\cdot} = \frac{|P(i)|}{|C(i)| + |P(i)|} , \quad \forall A' \in \mathcal{A} ,$$

and

$$\pi_A^{i\to j} = \frac{A_{ij}}{|C(i)|} , \quad \text{and} \quad \pi_A^{i\leftarrow j} = \frac{\left[A^T\right]_{ij}}{|P(i)|}$$

and

$$\pi_B^{\sigma_i\to\sigma_j} = B_{\sigma_i\sigma_j} , \quad \text{and} \quad \pi_B^{\sigma_i\leftarrow\sigma_j} = \left[B^T\right]_{\sigma_i\sigma_j} ,$$

and one can rewrite Equation (4.6) as follows

$$\begin{aligned} Q_A^{\text{EA}''}(B,S) = \frac{1}{2\,|E(A)|} \sum_{i,j} & A_{ij}B_{\sigma_i\sigma_j} + \left[A^T\right]_{ij}\left[B^T\right]_{\sigma_i\sigma_j} \\ & - |C(i)|\ \mathbb{E}_{A'\in\mathcal{A}}\left[\pi_{A'}^{i\to j}\right]\ B_{\sigma_i\sigma_j} \\ & - |P(i)|\ \mathbb{E}_{A'\in\mathcal{A}}\left[\pi_{A'}^{i\leftarrow j}\right]\ \left[B^T\right]_{\sigma_i\sigma_j} . \end{aligned} \tag{4.7}$$

The expected value of the probability that an edge with node i as origin has node j as termination in the configuration model is proportional to the degree of j, and we have

$$\mathbb{E}_{A'\in\mathcal{A}}\left[\pi_{A'}^{i\to j}\right] = \frac{|P(j)|}{|E(A)|} , \quad \text{and} \quad \mathbb{E}_{A'\in\mathcal{A}}\left[\pi_{A'}^{i\leftarrow j}\right] = \frac{|C(i)|}{|E(A)|} .$$

And finally, Equation (4.7) becomes

$$\begin{aligned} Q_A^{\text{EA}''}(B,S) &= \frac{1}{2\,|E(A)|} \sum_{i,j} \left(A_{ij} - \tfrac{|C(i)||P(j)|}{|E(A)|}\right)\ B_{\sigma_i\sigma_j} \\ &\qquad + \left(\left[A^T\right]_{ij} - \tfrac{|P(i)||C(i)|}{|E(A)|}\right)\ \left[B^T\right]_{\sigma_i\sigma_j} \\ &= \frac{1}{2\,|E(A)|}\text{tr}\big(S^T M_A S B^T + S^T M_A^T S B\big) = Q_A^{\text{EA}}(B,S) . \end{aligned}$$

Let us now interpret the problem of nodes without children or parents in view of this new interpretation. If one ends up on a node without

children (resp. parents) during the undirected random walk on $G(A)$ or $G(A')$ with $A' \in \mathcal{A}$, one can only choose an incoming (resp. outgoing) edge to continue the undirected random walk and further check if it is compatible with the Blockmodel, and hence it does not matter if the node without children (resp. parents) is assigned to a role with or without children (resp. parents).

The choice of the configuration model as null model plays an important role in this unexpected effect. Indeed, if the reconfiguration of the edges in the null model would allow to assign children or parents to nodes that initially had none, then their expected connections would not be zero in the null model. Let us now choose another null model whose null set is composed of the adjacency matrices with the same number of edges as the initial graph, *i.e.*

$$\mathcal{A} = \left\{ A' \in \mathbb{R}^{m \times m} \text{ s.t. } \left|E(A')\right| = |E(A)| \right\} \ .$$

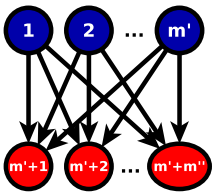
This choice yields

$$b_{ij} = \mathbb{E}_{A' \in \mathcal{A}} \left[A'_{ij} \right] = \frac{|E|}{m^2} \ ,$$

and its associated quality measure may be written as follows

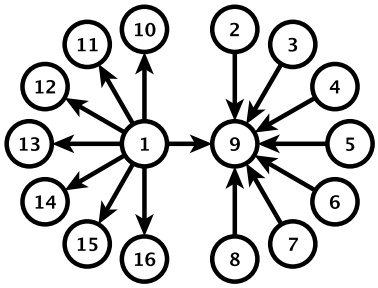
$$Q_A^{\widetilde{\text{EA}}}(B, S) = \frac{1}{|E|} \operatorname{tr}\Big(S^T \tilde{M}_A^T S B \Big) \ , \tag{4.8}$$

with $[\tilde{M}_A]_{i,j} := A_{ij} - |E| / m^2$. *E.g.*, when considering the graph of the example on page 95, the Blockmodel with highest quality is as expected given by the following partition.



Note that the Blockmodel of the graph $G(A)$ in Figure 4.1 is also the unique maximizer of $Q_A^{\widetilde{\text{EA}}}(B, S)$.

Though performing better on our example, the quality $Q_A^{\widetilde{\text{EA}}}$ still suffers from counter-intuitive effects when applied to graphs with widely spread node degrees. *E.g.*, let us consider the graph whose graphical representation is the following,



.

This graph is clearly bipartite, but the unique maximizer of $Q_A^{\text{EA}\tilde{}}$ among the Blockmodels of order 2 is composed of the partition $\sigma^{-1}(1) = \{1, 9\}$, and $\sigma^{-1}(2) = \{2, 3, \cdots, 8, 10, 11, \cdots, 16\}$ and of the image graph whose adjacency matrix is $B = [\begin{smallmatrix} 1 & 1 \\ 1 & 0 \end{smallmatrix}]$.[4] In fact, one can see that the quality measure $Q_A^{\text{EA}\tilde{}}$ tends to group the nodes with high degree in the same partition.

4.2.3 Role Satisfaction

The quality of a Blockmodel (B, σ) may be evaluated by giving

- reward if, for a pair $(i, s) \in N(A) \times N(B)$, the node i is connected to one or more nodes whose roles are s while the role of the node i is connected to the role s, or
- reward if, for a pair $(i, s) \in N(A) \times N(B)$, the node i is not connected to any node whose roles is s while the role of the node i is not connected to the role s, and
- penalty if, for a pair $(i, s) \in N(A) \times N(B)$, the node i is connected to one or more nodes whose roles are s while the role of the node i is not connected to the role s, or
- penalty if, for a pair $(i, s) \in N(A) \times N(B)$, the node i is not connected to any node whose roles is s while the role of the node i is connected to the role s,

or, in other words, give a

[4] This result has been found using exhaustive search, *i.e.* all colorings have been tried.

- reward if $[R_C]_{i,s} = B_{\sigma_i,s}$ or $[R_P]_{i,s} = [B^T]_{\sigma_i,s}$, and
- penalty if $[R_C]_{i,s} \neq B_{\sigma_i,s}$ or $[R_P]_{i,s} \neq [B^T]_{\sigma_i,s}$,

where R_C (resp. R_P) is the matrix representation of the function $\sigma(C(\cdot))$ (resp. $\sigma(P(\cdot))$) which can also be computed as follows

$$R_C = Is_{\neq 0}(AS) \ , \quad (\text{resp. } R_P = Is_{\neq 0}(A^T S)) \ .$$

Let us extend the ideas introduced in Section 4.2.2 and propose the following quality measure

$$\begin{aligned} Q_A^{\text{RS}}(B,S) \ &= \ \sum_{i,s} a_{is}[R_C]_{i,s}B_{\sigma_i,s} + b_{is}(1-[R_C]_{i,s})(1-B_{\sigma_i,s}) \\ &\qquad -c_{is}[R_C]_{i,s}(1-B_{\sigma_i,s}) - d_{is}(1-[R_C]_{i,s})B_{\sigma_i,s} \\ &\qquad +a'_{is}[R_P]_{i,s}[B^T]_{\sigma_i,s} + b'_{is}(1-[R_P]_{i,s})(1-[B^T]_{\sigma_i,s}) \\ &\qquad -c'_{is}[R_P]_{i,s}(1-[B^T]_{\sigma_i,s}) - d'_{is}(1-[R_P]_{i,s})[B^T]_{\sigma_i,s} \\ &= \ \sum_{i,s} (a_{is}+b_{is}+c_{is}+d_{is})\ [R_C]_{i,s}\ B_{\sigma_i,s} \\ &\qquad -(b_{is}+d_{is})\ B_{\sigma_i,s} - (b_{is}+c_{is})\ [R_C]_{i,s} + b_{is} \\ &\qquad +(a'_{is}+b'_{is}+c'_{is}+d'_{is})\ [R_P]_{i,s}\ [B^T]_{\sigma_i,s} \\ &\qquad -(b'_{is}+d'_{is})\ [B^T]_{\sigma_i,s} - (b'_{is}+c'_{is})\ [R_P]_{i,s} + b'_{is} \end{aligned}$$

with a_{ij}, b_{ij}, c_{ij}, d_{ij}, a'_{ij}, b'_{ij}, c'_{ij} and d'_{ij} positive parameters that do not depend on B and σ. Clearly, for each pair $(i,s) \in N(A) \times N(B)$,

- if $[R_C]_{i,s} = B_{\sigma_i,s} = 1$, then the contribution of this pair to the quality measure is a_{is},
- if $[R_C]_{i,s} = B_{\sigma_i,s} = 0$, then the contribution of this pair to the quality measure is b_{is},
- if $[R_P]_{i,s} = [B^T]_{\sigma_i,s} = 1$, then the contribution of this pair to the quality measure is a'_{is}, and
- if $[R_P]_{i,s} = [B^T]_{\sigma_i,s} = 0$, then the contribution of this pair to the quality measure is b'_{is},

which are all positive contributions conformally with the requirements. On the other hand, for each pair $(i,s) \in N(A) \times N(B)$,

- if $[R_C]_{i,s} = 1$ and $B_{\sigma_i,s} = 0$, then the contribution of this pair to the quality measure is $-c_{is}$,
- if $[R_C]_{i,s} = 0$ and $B_{\sigma_i,s} = 1$, then the contribution of this pair to the quality measure is $-d_{is}$,
- if $[R_P]_{i,s} = 1$ and $[B^T]_{\sigma_i,s} = 0$, then the contribution of this pair to the quality measure is $-c'_{is}$, and
- if $[R_P]_{i,s} = 0$ and $[B^T]_{\sigma_i,s} = 1$, then the contribution of this pair to the quality measure is $-d'_{is}$,

which are all negative contributions conformally with the requirements. A natural choice consists in weighting all contributions equally by setting $a_{is} = b_{is} = c_{is} = d_{is} = a'_{is} = b'_{is} = c'_{is} = d'_{is} = \frac{1}{4}$ which yields the following quality function

$$Q_A^{\text{RS}}(B,S) = \operatorname{tr}\left(\left(S^T\left(R_C - \tfrac{\mathbf{1}_{m,n}}{2}\right) + \left(R_P - \tfrac{\mathbf{1}_{m,n}}{2}\right)^T S\right)^T \left(B - \tfrac{\mathbf{1}_{n,n}}{2}\right)\right),$$

which is exactly the quality function of Equation (4.1) of Section 4.2.1 with $c = 1/2$. Moreover, since R_C, R_P, and B are binary matrices and a selection matrix S satisfies $S\mathbf{1}_{n,n} = \mathbf{1}_{m,n}$, we clearly have

$$\operatorname{tr}\Big(\left(S^T R_C\right)^T \mathbf{1}_{n,n}\Big) = \operatorname{tr}\big(R_C^T \mathbf{1}_{m,n}\big) = \operatorname{tr}\big(R_C^T R_C\big)\ ,$$
$$\operatorname{tr}\Big(\left(S^T \mathbf{1}_{m,n}\right)^T B\Big) = \operatorname{tr}\big(\mathbf{1}_{m,n}^T \cdot SB\big) = \operatorname{tr}\big((SB)^T \cdot SB\big)\ ,$$

and

$$\operatorname{tr}\Big(\left(R_P^T S\right)^T \mathbf{1}_{n,n}\Big) = \operatorname{tr}\big(R_C^T \mathbf{1}_{m,n}\big) = \operatorname{tr}\big(R_P^T R_P\big)\ ,$$
$$\operatorname{tr}\Big(\left(\mathbf{1}_{m,n}^T S\right)^T B\Big) = \operatorname{tr}\big(\mathbf{1}_{m,n}^T \cdot SB^T\big) = \operatorname{tr}\Big(\left(SB^T\right)^T \cdot SB^T\Big)\ ,$$

which allows us to rewrite our quality function as follows

$$Q_A^{\text{RS}}(B,S) = \frac{1}{2}\left(\|\mathbf{1}_{m,n}\|_F^2 - \|R_C - SB\|_F^2 - \left\|R_P - SB^T\right\|_F^2\right)\ .$$

Let now the quality function of Problem 4.1 be given by $Q_A(B,S) = Q_A^{\text{RS}}(B,S)$, which reduces that problem to the following.

Problem 4.6 *Given $A \in \{0,1\}^{m\times m}$, and $n \in \mathbb{N}$, find a Blockmodel (B^*, S^*) in*

$$\underset{(B,S)\in\mathcal{B}m(n)}{\operatorname{argmax}} \operatorname{tr}\left(\left(S^T\left(R_C - \tfrac{\mathbf{1}_{m,n}}{2}\right) + \left(R_P - \tfrac{\mathbf{1}_{m,n}}{2}\right)^T S\right)^T \left(B - \tfrac{\mathbf{1}_{n,n}}{2}\right)\right),$$

with $R_C = Is_{\neq 0}(AS)$, and $R_P = Is_{\neq 0}(A^TS)$, which can also be written as

$$\underset{(B,S)\in\mathcal{B}m(n)}{\operatorname{argmax}} \left(\|\mathbf{1}_{m,n}\|_F^2 - \|R_C - SB\|_F^2 - \left\|R_P - SB^T\right\|_F^2\right).$$

One can show again in view of Lemma 4.1 that, for a given S, the matrix

$$B(S) := Is_{>0}\left(S^T\left(R_C - \frac{\mathbf{1}_{m,n}}{2}\right) + \left(R_P - \frac{\mathbf{1}_{m,n}}{2}\right)^T S\right)$$

is the sparsest matrix in $\underset{B\in\{0,1\}^{n\times n}}{\operatorname{argmax}} Q_A^{\mathrm{RS}}(B,S)$, and

$$Q_A^{\mathrm{RS}}(B(S),S) = \sum_{r,s}\left|S^T\left(R_C - \frac{\mathbf{1}_{m,n}}{2}\right) + \left(R_P - \frac{\mathbf{1}_{m,n}}{2}\right)^T S\right|_{r,s}.$$

And Problem 4.6 may be reduced to the following.

Problem 4.7 *Given $A \in \{0,1\}^{m\times m}$, and $n \in \mathbb{N}$, find a partition S^* in*

$$\underset{S\in\mathcal{P}(m,n)}{\operatorname{argmax}} \sum_{r,s}\left|S^T\left(R_C - \frac{\mathbf{1}_{m,n}}{2}\right) + \left(R_P - \frac{\mathbf{1}_{m,n}}{2}\right)^T S\right|_{r,s}.$$

Note that, among the Blockmodels of order 4, the one of the graph $G(A)$ in Figure 4.1 is the unique maximizer of $Q_A^{\mathrm{RS}}(B,S)$. Moreover, among the Blockmodels of order 2, the bipartite coloration of the bipartite graphs of pages 95 and 99 along with their respective image graphs are the unique maximizer of $Q_A^{\mathrm{RS}}(B,S)$.

4.3 Numerical Methods for Blockmodeling

There exists no known polynomial method that solves Problem 4.2, nor Problem 4.3, nor Problem 4.5, nor Problem 4.7. In this section, we

present an algorithm that finds Blockmodels that are relevant with respect to a given quality function. We further test this algorithm with the different quality functions that we introduced in the previous section.

4.3.1 The Algorithm

Our algorithm is described in Algorithm 2. It first requires the following ingredients:

- $A \in \{0,1\}^{m\times m}$ is the adjacency matrix of the graph under consideration,
- $\mathcal{R}_A$ is a subset of $N(A) \times N(A)$ such that (i,j) belongs to $\mathcal{R}_A$ if one thinks that the role of the node i could be the same as the role of the node j. In our algorithm, we define this set as follows

$$\mathcal{R}_A := \left\{ (i,j) \text{ s.t. } j \neq i \text{ and } \begin{pmatrix} j \in \Gamma(i), \\ \text{or } C(i) \cap C(j) \neq \varnothing, \\ \text{or } P(i) \cap P(j) \neq \varnothing. \end{pmatrix} \right\} .$$

  This choice assumes that two different nodes that are either neighbors or have common children or parents, may possibly play the same role. For a uniform degree distribution, this $\mathcal{R}_A$ has of the order of $\frac{|E(A)|^2}{|N(A)|}$ elements.

- $\mathcal{R}_A(\sigma)$ is a function that, for a given partition σ, returns a subset of $\mathrm{Im}(\sigma) \times \mathrm{Im}(\sigma)$ such that (i,j) belongs to $\mathcal{R}_A(\sigma)$ if one thinks that the role i could be set to be equal to the role of j. In our algorithm, we define this set as follows

$$\mathcal{R}_A(\sigma) := \left\{ \begin{matrix} (i,j) \text{ s.t. } j \neq i \text{ and } \exists k \in \sigma^{-1}(i) \text{ and } \ell \in \sigma^{-1}(j) \\ \text{with } \ell \in \Gamma(k), \text{ or } C(\ell) \cap C(k) \neq \varnothing, \\ \text{or } P(\ell) \cap P(k) \neq \varnothing. \end{matrix} \right\} .$$

  The reasons of this choice are similar to the ones of $\mathcal{R}_A$. Let us notice that, for the partition $\sigma_0 := (N(A) \to \mathbb{N}_{\leq m} : i \mapsto i)$, we have

$$\mathcal{R}_A = \mathcal{R}_A(\sigma_0) .$$

- $n(\sigma)$ is a function that, for a given partition σ, returns the number of non empty blocks of a partition σ,

- $Q_{\mathrm{B}}(\sigma)$ is a function that, for a given partition, $\sigma \in \cup_{n\ p}(m,n)$, returns the maximal value over all B of a Blockmodel quality function $Q(B, M(\sigma))$, or, being given the value $\varnothing$ returns $-\infty$, *i.e.*

$$Q_{\mathrm{B}}(\sigma) := \begin{cases} \max_B Q(B, M(\sigma)), \text{ if } \sigma \in \cup_{n\ p}(m,n), \\ -\infty, \text{ if } \sigma = \varnothing, \end{cases}$$

  where $M(\sigma)$ denotes the matrix representation of σ.

Lines 1 and 2 of Algorithm 2 initialize the following variables:

- Σ^{Best} is a function that, being given $n \in \mathbb{N}_{\leq m}$, returns either the partition with n non-empty blocks with the highest value for $Q_{\mathrm{B}}(\sigma)$ computed so far in the algorithm, or $\varnothing$ if none has been encountered yet. Σ^{Best} is initialized as follows

$$\Sigma^{\mathrm{Best}} : \mathbb{N}_{\leq m} \to \left\{\varnothing \cup (\cup_{n\ p}(m,n))\right\} : n \mapsto \varnothing\ .$$

- σ_0 is the initial partition and is defined such that each node is alone in its own block, *e.g.*

$$\sigma_0 : N(A) \to \mathbb{N}_{\leq m} : i \mapsto i\ .$$

Lines 3 to 34 of Algorithm 2 loop until convergence. At each step of the loop, the following actions are successively tried:

- a split step (line 5 to 10),
- a move step (line 11 to 20),
- a merge step (line 21 to 30), or
- a deflate step (line 31 to 33),

until one of them changes the current partition. If no step succeeds, the algorithm has converged.

Let us now describe these different steps.

Algorithm 2 Louvain Method for Blockmodeling

Require: A, $\mathcal{R}_A$, $\mathcal{R}_A(\sigma)$, $n(\sigma)$, $Q_{\mathrm{B}}(\sigma)$ (see p. 103 for details).

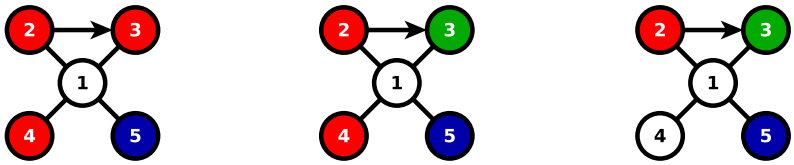
$\Sigma^{\mathrm{Best}} \leftarrow (\mathbb{N}_{\leq m} \to \left\{\varnothing \cup (\cup_{n\ p}(m,n))\right\} : n \mapsto \varnothing)$;
$t \leftarrow 0$; $\sigma_0 \leftarrow (N(A) \to \mathbb{N}_{\leq m} : i \mapsto i)$; $\Sigma^{\mathrm{Best}}(m) \leftarrow \sigma_0$;
repeat
  $t \leftarrow t+1$; $\sigma_t \leftarrow \sigma_{t-1}$; success $\leftarrow$ false ;
  if $n(\sigma_{t-1}) \leq m-1$ and $Q_{\mathrm{B}}(\sigma_{t-1}) \geq Q_{\mathrm{B}}\left(\Sigma^{\mathrm{Best}}\left(n\left(\sigma_{t-1}\right)+1\right)\right)$ **then**
    $\sigma_{\mathrm{new}} \leftarrow$ `split_one_block_of`(σ_{t-1}) ;
    if $Q_{\mathrm{B}}(\sigma_{\mathrm{new}}) > Q_{\mathrm{B}}(\Sigma^{\mathrm{Best}}(n(\sigma_{\mathrm{new}})))$ **then**
      $\sigma_t \leftarrow \sigma_{\mathrm{new}}$; $\Sigma^{\mathrm{Best}}(n(\sigma_{\mathrm{new}})) \leftarrow \sigma_{\mathrm{new}}$; success $\leftarrow$ true ;
    end if
  end if
  if success = false **then**
    $\mathcal{R}' \leftarrow \mathcal{R}_A$;
    while $\sigma_t \neq \sigma_{t-1}$ and $\mathcal{R}' \neq \varnothing$ **do**
      $(i,j) \leftarrow$ `get_element_of`$(\mathcal{R}')$; $\mathcal{R}' \leftarrow \mathcal{R}' \setminus \{(i,j)\}$;
      $\sigma_{\mathrm{new}} \leftarrow \sigma_{t-1}$; $\sigma_{\mathrm{new}}(i) \leftarrow \sigma_{t-1}(j)$;
      if $Q_{\mathrm{B}}(\sigma_{\mathrm{new}}) > Q_{\mathrm{B}}(\Sigma^{\mathrm{Best}}(n(\sigma_{\mathrm{new}})))$ **then**
        $\sigma_t \leftarrow \sigma_{\mathrm{new}}$; $\Sigma^{\mathrm{Best}}(n(\sigma_{\mathrm{new}})) \leftarrow \sigma_{\mathrm{new}}$; success $\leftarrow$ true ;
      end if
    end while
  end if
  if success = false and $n(\sigma_{t-1}) \geq 2$ **then**
    $\mathcal{R}' \leftarrow \mathcal{R}_A(\sigma_{t-1})$;
    while $\sigma_t \neq \sigma_{t-1}$ and $\mathcal{R}' \neq \varnothing$ **do**
      $(\sigma_i, \sigma_j) \leftarrow$ `get_element_of`$(\mathcal{R}')$; $\mathcal{R}' \leftarrow \mathcal{R}' \setminus \{(\sigma_i, \sigma_j)\}$;
      $\sigma_{\mathrm{new}} \leftarrow \sigma_{t-1}$; $\sigma_{\mathrm{new}}(k|\sigma_{t-1}(k) = \sigma_i) \leftarrow \sigma_j$;
      if $Q_{\mathrm{B}}(\sigma_{\mathrm{new}}) > Q_{\mathrm{B}}(\Sigma^{\mathrm{Best}}(n(\sigma_{\mathrm{new}})))$ **then**
        $\sigma_t \leftarrow \sigma_{\mathrm{new}}$; $\Sigma^{\mathrm{Best}}(n(\sigma_{\mathrm{new}})) \leftarrow \sigma_{\mathrm{new}}$; success $\leftarrow$ true ;
      end if
    end while
  end if
  if success = false and $n(\sigma_{t-1}) \geq 2$ and $\Sigma^{\mathrm{Best}}(n(\sigma_{t-1})-1) \neq \varnothing$ **then**
    $\sigma_t \leftarrow \Sigma^{\mathrm{Best}}(n(\sigma_{t-1})-1)$; success $\leftarrow$ true ;
  end if
until success = false
return Σ^{Best}

`split_one_block_of`(σ) is a function that randomly choose one block of the partition σ with more than two elements and split it in two non empty blocks. `get_element_of`$(\mathcal{R})$ is a function that returns a random element of the set $\mathcal{R}$.

The split step

A split step (line 5 to 10 of Algorithm 2) is performed if the quality score of the last partition, $Q_{\mathrm{B}}(\sigma_{t-1})$, is higher than the quality score of the best partition with one more non empty block than σ_{t-1}, *i.e.* $Q_{\mathrm{B}}(\Sigma^{\mathrm{Best}}(n(\sigma_{t-1})+1))$. A split step consists in trying to randomly choose a block of σ_{t-1} containing more than two nodes and split it in two non-empty blocks. If the quality score of the split partition is higher than the best partition with the same number of non empty blocks, then σ_t is set to that split partition. *E.g.*, let the partitions σ_{t-1}, σ_{new}, and σ_{best} be defined as follows

σ_{t-1} σ_{new} σ_{best}



$Q_{\mathrm{B}}^{\mathrm{EA}}(\sigma_{t-1}) = 0.40$, $Q_{\mathrm{B}}^{\mathrm{EA}}(\sigma_{\mathrm{new}}) = 0.43$, $Q_{\mathrm{B}}^{\mathrm{EA}}(\sigma_{\mathrm{best}}) = 0.26$,
$Q_{\mathrm{B}}^{\mathrm{EA}\tilde{}}(\sigma_{t-1}) = 0.57$, $Q_{\mathrm{B}}^{\mathrm{EA}\tilde{}}(\sigma_{\mathrm{new}}) = 0.60$, $Q_{\mathrm{B}}^{\mathrm{EA}\tilde{}}(\sigma_{\mathrm{best}}) = 0.32$,
$Q_{\mathrm{B}}^{\mathrm{RS}}(\sigma_{t-1}) = 0.92$, $Q_{\mathrm{B}}^{\mathrm{RS}}(\sigma_{\mathrm{new}}) = 0.96$, $Q_{\mathrm{B}}^{\mathrm{RS}}(\sigma_{\mathrm{best}}) = 0.76$,
$Q_{\mathrm{B}}^{\text{N-Uncut}}(\sigma_{t-1}) = 1.032$, $Q_{\mathrm{B}}^{\text{N-Uncut}}(\sigma_{\mathrm{new}}) = 1.049$, $Q_{\mathrm{B}}^{\text{N-Uncut}}(\sigma_{\mathrm{best}}) = 1.030$,

where σ_{new} is the result of a split step applied to σ_{t-1} where the block $\{2,3,4\}$ is split in the two blocks $\{2,4\}$ and $\{3\}$, and

$$\sigma_{\mathrm{best}} = \Sigma^{\mathrm{Best}}\left(n\left(\sigma_{t-1}\right)+1\right) \ .$$

In this case, the split step would succeed for all quality functions since the quality of σ_{new} is always higher than the quality of σ_{best}.

Note that in the case of Q_{B} is built with the Blockmodel quality functions Q_A^{EA} or $Q_A^{\mathrm{EA}\tilde{}}$, one can prove that the value of Q_{B} of the split partition is always higher than the best partition with the same number of non empty blocks.

The move step

A move step (line 11 to 20 of Algorithm 2) is performed if the split step has failed. A move step consists in trying for each element $(i,j) \in \mathcal{R}_A$ to

change the last partition σ_{t-1} by moving the node i into the block of the node j, until an improvement of Σ^{Best} is found. *E.g.*, let the partitions σ_{t-1}, and σ_{new} be defined as follows

σ_{t-1} σ_{new}

$Q_{\text{B}}^{\text{EA}}(\sigma_{t-1}) = 0.26,$ $Q_{\text{B}}^{\text{EA}}(\sigma_{\text{new}}) = 0.43,$
$Q_{\text{B}}^{\text{EA}}(\sigma_{t-1}) = 0.32,$ $Q_{\text{B}}^{\text{EA}}(\sigma_{\text{new}}) = 0.60,$
$Q_{\text{B}}^{\text{RS}}(\sigma_{t-1}) = 0.76,$ $Q_{\text{B}}^{\text{RS}}(\sigma_{\text{new}}) = 0.96,$
$Q_{\text{B}}^{\text{N-Ncut}}(\sigma_{t-1}) = 1.030,$ $Q_{\text{B}}^{\text{N-Ncut}}(\sigma_{\text{new}}) = 1.049,$

where σ_{new} is the result of a move step applied to σ_{t-1} where node 4 has been moved from block $\{1, 4\}$ to block $\{2, 4\}$. In this case, the move step would succeed for all quality functions since the quality of σ_{new} is always higher than the quality of σ_{t-1}.

The merge step

A merge step (line 21 to 30 of Algorithm 2) is performed if the split step, and the move step have failed. A merge step consists in trying for each element $(i, j) \in \mathcal{R}_A(\sigma_{t-1})$ to change the last partition σ_{t-1} by merging the block i with the block j, until an improvement of Σ^{Best} is found. *E.g.*, let the partitions σ_{t-1}, and σ_{new} be defined as follows

σ_{t-1} σ_{new}

$Q_{\text{B}}^{\text{EA}}(\sigma_{\text{new}}) = 0.43,$ $Q_{\text{B}}^{\text{EA}}(\sigma_{t-1}) = 0.40,$
$Q_{\text{B}}^{\text{EA}}(\sigma_{\text{new}}) = 0.60,$ $Q_{\text{B}}^{\text{EA}}(\sigma_{t-1}) = 0.57,$
$Q_{\text{B}}^{\text{RS}}(\sigma_{\text{new}}) = 0.96,$ $Q_{\text{B}}^{\text{RS}}(\sigma_{t-1}) = 0.92,$
$Q_{\text{B}}^{\text{N-Ncut}}(\sigma_{\text{new}}) = 1.049,$ $Q_{\text{B}}^{\text{N-Ncut}}(\sigma_{t-1}) = 1.032,$

where σ_{new} is the result of a merge step applied to σ_{t-1} where the block $\{2, 4\}$ and $\{3\}$ are merged together.

The deflate step

A deflate step (line 31 to 33 of Algorithm 2) is performed if the split step, the move step, and the merge step have failed. A deflate step consists in setting σ_t to $\Sigma^{\text{Best}}(n(\sigma_{t-1}) - 1)$ if the latter is not equal to $\varnothing$.

Notice that the move step and the merge step are partially inspired by the *Louvain Method for Community Detection*, described in [BGLL08]. Indeed, in that algorithm, the first phase consists in trying to improve the modularity function by moving nodes from their block to one of their neighbors until no improvement is made. Each attempt of this phase is very similar to our move step, except that in our algorithm the set of possible moves is given by $\mathcal{R}_A$ instead of trying to move nodes to their neighboring blocks. If no move has been made during that first phase, the *Louvain Method for Community Detection* stops, otherwise it proceeds with its second phase which consists in building the weighted image graph in which the weights are equal to the sum of edges between the nodes of the corresponding blocks found at the end of the first phase. The algorithm then applies the first phase to the weighted graph built in the second phase. This second phase is quite similar to our merge step since

- it is done when no move step can improve the objective function, and
- by performing the first move step of the first phase on the weighted image graph, the *Louvain Method for Community Detection* tries to improve the modularity function by merging two blocks as the merge step does in our algorithm for the Blockmodel quality function.

Notice the major differences between the *Louvain Method for Community Detection* and our algorithm are that

- their improvement condition is stated over partitions of any number of blocks while our condition compares the Blockmodel quality scores of partitions with the same number of blocks, and

- once the nodes are merged in a node in a image graph during the second phase of the *Louvain Method for Community Detection*, it is not possible to improve the modularity function by moving a single node of the original graph since they are for ever merged into a single node in the image graph while in our algorithm a move step can still happen after a merge step.

4.3.2 Convergence Analysis

Notice that Algorithm 2 always converges since

- each success in a split step, a move step or a merge step increases a value of the image of Σ^{Best}, and hence, because the feasible set is finite, one may not perform these steps indefinitely without reaching convergence, and
- each success in a deflate step, does not decrease Σ^{Best} and decreases the number of non-empty blocks of the current partition, and hence, because the number of possible numbers of non empty block is finite, one may not either perform this step indefinitely without reaching convergence.

Moreover, even if the split step, the move step and the merge step cannot increase Σ^{Best} anymore, the deflate step will succeed until it reaches the partition with one non-empty block which is hence a global attractor of Algorithm 2.

4.3.3 Experiments and Results on an Artificial Network

We now present the results of Algorithm 2 on the graph shown in Figure 4.2, for different Blockmodel quality functions,

- $Q^{\text{N-Ucut}}_{Sim(A)}$ in Figure 4.3,
- Q^{EA}_{A} in Figure 4.4,
- $Q^{\text{EA}^{\sim}}_{A}$ in Figure 4.5, and
- Q^{RS}_{A} in Figure 4.6.

Algorithm 2 with the Blockmodel quality function $Q^{\text{N-Uncut}}_{Sim(A)}$ fails to detect the 5 expected blocks, though there are only 2 misplaced nodes (*i.e.* nodes 10 and 20) over 40. Actually, the 5 blocks partition found by the algorithm has a quality score of 1.627 which is higher than the quality of the expected one (1.617). One can also observe that the 5 blocks partition does not achieve the highest value in the graph (b) of Figure 4.3.

Algorithm 2 with the Blockmodel quality function Q^{EA}_{A} fails to detect the 5 blocks and seems to confound nodes without parents and nodes without children confirming what we emphasized in Section 4.2.2. Hence the highest value in the graph (b) of Figure 4.4 is achieved for a 4-Blocks partition. Notice that the partition $\Sigma^{\text{Best}}(4)$ (resp. $\Sigma^{\text{Best}}(5)$) in Figure 4.5 have a quality score for Q^{EA}_{A} of 0.5209 (resp. 0.5280), which are respectively higher than the one of partition $\Sigma^{\text{Best}}(4)$ (resp. $\Sigma^{\text{Best}}(5)$) in Figure 4.4, *i.e.* 0.5183 (resp. 0.5205).

Algorithm 2 with the Blockmodel quality function $Q^{\text{EA}\sim}_{A}$ succeeds to detect the 5 blocks which, moreover, achieves the highest value in the graph (b) of Figure 4.5. Notice that the other best partitions of less than 5 blocks almost simply merge several blocks of the 5 blocks one.

Algorithm 2 with the Blockmodel quality function Q^{RS}_{A} succeeds to detect the 5 blocks which, nevertheless, does not achieve the highest value in the graph (b) of Figure 4.5.

Finally, Figure 4.7 shows the number of iterations before convergence of Algorithm 2 with the Blockmodel quality function $Q^{\text{EA}\sim}_{A}$ versus the size of the graph under consideration. The number of iterations before convergence seems to be linearly related to the size of the graph under consideration.

4.3.4 Experiments and Results on the *Baydry* Network

We now present in Figure 4.9 the results of Algorithm 2 on the Baydry network for the Blockmodel quality functions $Q^{\text{EA}\sim}_{A}$. The Baydry network is a food web whose nodes represent the living entities in the Florida bay and whose edges are such that there is an edge from the node i to the node j if the entity i is eaten by the entity j [Bayb, Bayc, Baya]. The adjacency matrix of the graph that represents this network is shown in

Figure 4.8. Notice that in this Figure the nodes have been relabelled to emphasize the block structure of the network. In our experiment, we have considered that the edges of the graph are not weighted although they were in the original data, and we have ignored six non living entities of the network which are water POC, benthic POC, DOC, input, output, and respiration.

In Figure 4.9, we observe that the Blockmodel with five block seems to be a good Blockmodel since it has an almost maximal quality score while it has a small number of blocks. The five blocks of this Blockmodel are completely described in Figure 4.8 and we now analyze theses blocks and give an overview of what type of entities are present in each block. The first block is composed of phytoplanktons, bacteria, seagrasses, macroalgae, microfauna and the green turtle. The second block is composed of zooplanktons, gastropods, macrobenthos, crustaceans, amphipods, isopods, shrimps, several crabs, and a tiny fish, the sailfin molly. The third block is composed of

- macroinvertebrates: several corals, sea stars, sea urchins, sand dollars, and sea cucumbers, marine worms, lobsters, several crabs, and
- small to medium-sized fishes: sardines, anchovies, flying fishes, killifishes, silversides, horsefishes, pipefishes, seahorses, mojarras, pinfishes, mullets, blennies, gobies, and other demersal fishes.

The fourth block is composed of

- macroinvertebrates: several corals, sea anemones, jellyfishes
- medium-sized to large fishes: rays, bonefishes, lizardfishes, catfishes, eels, needlefishes, snooks, jacks, pompanos, snappers, spadefishes, parrotfishes, flatfishes, filefishes, pufferfishes, and other pelagic fishes
- medium-sized birds: herons & egrets, ibis, roseate spoonbills, several ducks, gruiformes, small shorebirds, gulls & terns, kingfishers, and
- other large animals: loggerhead turtles, hawksbill turtles, manatees.

The fifth block is composed of

- roots,
- large fishes: sharks, tarpons, groupers, mackerels, barracudas,
- medium-sized to large birds: loons, greebs, pelicans, cormorants, big herons & egrets, several ducks, and
- other large animals: raptors, crocodiles, dolphins.

The green turtle seems to be an outlier in the first block. Nevertheless, in the data, the green turtle eats and is eaten by nearly nobody which justifies its position in a block whose entities eat nearly nobody and are only mainly eaten by the entities of the second block. This may come from the fact that we have considered that the graph was not weighted and that we underestimated the importance of those few connections from and toward the green turtle. Similarly, roots seems also to be an outlier in the fifth block. But again roots eat and are eaten by nobody.

Globally, by looking at our results, one can summarize the Baydry network as follows:

- Block 1 is composed of very simple micro-organisms that are mainly eaten by the ones in Block 2,
- Block 2 is composed of very simple macro-organisms that are mainly eaten by the ones in Block 3, Block 4, and Block 5,
- Block 3 is composed of macroinvertebrates and small to medium-sized fishes that are mainly eaten by the ones in Block 4, and Block 5,
- Block 4 is composed of medium-sized to large fishes, medium-sized birds and other large animals that are mainly eaten by the ones in Block 5,
- Block 5 is composed of large fishes, medium-sized to large birds, and other large animals that are mainly not eaten by anyone,

as shown in the image graph in (c) in Figure 4.9.

4.3.5 Complexity Analysis

We now focus on the complexity of Algorithm 2 with Q_A^{EA}. Let us first notice that, at any time t in Algorithm 2, the partitions in the image of Σ^{Best} or σ_{t-1} have all been previously set equal to either σ_0 or some σ_{new}. We hereafter present an efficient way to compute the qualities $Q_{\mathrm{B}}(\sigma_0)$ and any $Q_{\mathrm{B}}(\sigma_{\mathrm{new}})$, and as a consequence, at a time t in Algorithm 2, the qualities of the partitions in the image of Σ^{Best} and σ_{t-1} have simply to be remembered from when they were first computed.

Let us first describe how we compute $Q_{\mathrm{B}}(\sigma_0)$. The matrix representation of σ_0 is the identity matrix, *i.e.* $S_0 := M(\sigma_0) = I_m$, and we have

$$\begin{aligned} Q_{\mathrm{B}}(\sigma_0) &= \max_{B\in\{0,1\}^{m\times m}} Q_A^{\mathrm{EA}}(B, I_m) \\ &= \max_{B\in\{0,1\}^{m\times m}} \frac{1}{|E|}\mathrm{tr}\left(\left(A - \frac{k^{\mathrm{out}}k^{\mathrm{in}T}}{|E(A)|}\right)^T B\right) , \end{aligned}$$

with k^{out}, k^{in}, column vectors of $\mathbb{R}^m$, such that $k^{\mathrm{out}}(i)$ (resp. $k^{\mathrm{in}}(i)$) is the number of children (resp. parents) of node i. In view of Lemma 4.1, the maximum is achieved for

$$B = Is_{>0}\left(A - \frac{k^{\mathrm{out}}k^{\mathrm{in}T}}{|E(A)|}\right) .$$

Computing this B requires $E(A)$ operations. Indeed,

- if $A_{ij} = 0$, then $B_{ij} = 0$, and there is nothing to compute, and
- if $A_{ij} = 1$, then $B_{ij} = Is_{>0}\left(A_{ij} - \frac{k^{\mathrm{out}}(i)(k^{\mathrm{in}}(j))^T}{|E(A)|}\right)$.

Notice that this B is at least as empty as A is. This yields that $Q_{\mathrm{B}}(\sigma_0)$ is equal to a scalar product between a full matrix and a sparse matrix that has at most $|E(A)|$ non-zero elements, which requires less than $2\,|E(A)|$ operations. Finally, computing $Q_{\mathrm{B}}(\sigma_0)$ requires less than $3\,|E(A)|$ operations.

Let us now describe how we compute $Q_{\mathrm{B}}(\sigma_{\mathrm{new}})$. Any σ_{new}, built at time t in lines 6, 15, or 25 of Algorithm 2, is built by starting from σ_{t-1}

and moving k nodes from one block to another one. We now show how to compute the value of $Q_{\mathrm{B}}(\sigma_{\mathrm{new}})$ by incrementing the value of $Q_{\mathrm{B}}(\sigma_{t-1})$, that we know by induction. We assume first that, during the step $t-1$, we also computed the following quantities:

- $S_{t-1}^T A S_{t-1}$, a matrix in $\mathbb{N}^{n\times n}$, where $S_{t-1} := M(\sigma_{t-1})$ is the matrix representation of σ_{t-1} and
- B_{t-1}, the sparsest adjacency matrix in $\underset{B\in\{0,1\}^{n\times n}}{\operatorname{argmax}} Q_A^{\mathrm{EA}}(B, S_{t-1})$.

We hereafter also describe how to compute $S_{\mathrm{new}}^T A S_{\mathrm{new}}$ and B_{new} by respectively incrementing $S_{t-1}^T A S_{t-1}$ and B_{t-1}.

Let us first describe the computation of $S_{\mathrm{new}}^T A S_{\mathrm{new}}$. We have

$$\begin{aligned} S_{\mathrm{new}}^T A S_{\mathrm{new}} &= (S_{t-1}+\Delta)^T A (S_{t-1}+\Delta) \\ &= S_{t-1}^T A S_{t-1} + \Sigma^T A \Delta + \Delta^T A \Sigma \ , \end{aligned} \tag{4.9}$$

with $\Delta := S_{\mathrm{new}} - S_{t-1}$ and $\Sigma := \frac{S_{\mathrm{new}}+S_{t-1}}{2}$. Notice that Δ is a matrix in $\{-1,0,1\}^{m,n}$ such that only two of its columns contain k non-zero entries, whereas Σ is a matrix in $\left\{0,\frac{1}{2},1\right\}^{m,n}$ that contains $m+k$ non-zero elements, where k is the number of moved nodes. In Equation (4.9),

- $S_{t-1}^T A S_{t-1}$ has already been computed at time $t-1$ in Algorithm 2,
- $\Sigma^T(A\Delta)$ requires $4mk+4(m+k)$ operations, and is an $n\times n$ matrix that has only 2 non-empty columns, and
- $(\Delta^T A)\Sigma$ requires $4mk+4(m+k)$ operations, and is an $n\times n$ matrix that has only 2 non-empty rows.

Considering the sum of all terms, computing $S_{\mathrm{new}}^T A S_{\mathrm{new}}$ requires $8mk + 8(m+k) + 4n$ operations.

Let us now describe the computation of B_{new}. We have

$$\begin{aligned} B_{\mathrm{new}} &= Is_{>0}\left((S_{t-1}+\Delta)^T\left(A - \tfrac{k^{\mathrm{out}}k^{\mathrm{in}T}}{|E(A)|}\right)(S_{t-1}+\Delta)\right) \\ &= Is_{>0}\begin{pmatrix} S_{t-1}^T\left(A - \frac{k^{\mathrm{out}}k^{\mathrm{in}T}}{|E(A)|}\right)S_{t-1} + \Sigma^T A\Delta \\ +\Delta^T A\Sigma - \Sigma^T\frac{k^{\mathrm{out}}k^{\mathrm{in}T}}{|E(A)|}\Delta - \Delta^T\frac{k^{\mathrm{out}}k^{\mathrm{in}T}}{|E(A)|}\Sigma \end{pmatrix} \end{aligned} \tag{4.10}$$

In Equation (4.10),

- $\Sigma^T A\Delta + \Delta^T A\Sigma$ has already been computed while computing $S_{\text{new}}^T A S_{\text{new}}$, and is an $n \times n$ matrix that has 2 non-empty rows and 2 non-empty columns,
- $-(\Sigma^T k^{\text{out}})({k^{\text{in}}}^T \Delta)/\left|E(A)\right|$ requires $4k+2(m+k)+2n$ operations, and is an $n \times n$ matrix that has only 2 non-empty columns, and
- $-(\Delta^T k^{\text{out}})({k^{\text{in}}}^T \Sigma)/\left|E(A)\right|$ requires $4k+2(m+k)+2n$ operations, and is an $n \times n$ matrix that has only 2 non-empty rows.

The sum of these term $C := \Sigma^T A\Delta + \Delta^T A\Sigma - \Sigma^T \frac{k^{\text{out}} {k^{\text{in}}}^T}{|E(A)|}\Delta - \Delta^T \frac{k^{\text{out}} {k^{\text{in}}}^T}{|E(A)|}\Sigma$ is an $n \times n$ matrix that has 2 non-empty rows and 2 non-empty columns. And finally, whenever $C = 0$, we have $[B_{\text{new}}]_{ij} = \left[Is_{>0}\left(S_{t-1}^T\left(A - \frac{k^{\text{out}} {k^{\text{in}}}^T}{|E(A)|}\right)S_{t-1}\right)\right]_{ij} = [B_{t-1}]_{ij}$. For the other $4(n-1)$ non-zero entries of C, we have to compute

$$\begin{aligned} D_{ij} &:= [B_{\text{new}} - B_{t-1}]_{ij} \\ &= Is_{>0}\left(\left[S_{t-1}^T A S_{t-1} - S_{t-1}^T \frac{k^{\text{out}} {k^{\text{in}}}^T}{|E(A)|} S_{t-1} + C\right]_{ij}\right) \\ &\quad - Is_{>0}\left(\left[S_{t-1}^T A S_{t-1} - S_{t-1}^T \frac{k^{\text{out}} {k^{\text{in}}}^T}{|E(A)|} S_{t-1}\right]_{ij}\right) . \end{aligned}$$

In this equation,

- $S_{t-1}^T A S_{t-1}$ has already been computed at time $t-1$ in Algorithm 2, and
- $([S_{t-1}]_{.i})^T \frac{k^{\text{out}} {k^{\text{in}}}^T}{|E(A)|} [S_{t-1}]_{.j}$ is a scalar that requires of the order of m/n operations.

And finally, when one knows $S_{\text{new}}^T A S_{\text{new}}$, computing B_{new} requires $8k + 4(n+k) + 4n + 8(n-1) + O(m)$ operations.

Let us finally describe the computation of $Q_{\mathrm{B}}(\sigma_{\mathrm{new}})$. We have

$$\begin{aligned} Q_{\mathrm{B}}(\sigma_{\mathrm{new}}) &= \mathrm{tr}\Bigg((S_{t-1}+\Delta)^T \left(A - \tfrac{k^{\mathrm{out}} k^{\mathrm{in}^T}}{|E(A)|} \right)^T (S_{t-1}+\Delta)\ B_{\mathrm{new}} \Bigg) \\ &= Q_{\mathrm{B}}(\sigma_{t-1}) + \mathrm{tr}\big(C^T\ B_{\mathrm{new}}\big) + \mathrm{tr}\big(S_{t-1}^T A^T S_{t-1}\ D\big) \\ &\quad -\mathrm{tr}\Bigg(S_{t-1}^T \left(\tfrac{k^{\mathrm{out}} k^{\mathrm{in}^T}}{|E(A)|} \right)^T S_{t-1}\ D \Bigg) \end{aligned}$$

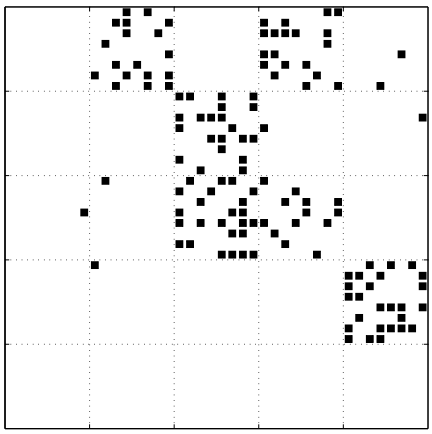
In this equation,

- $Q_{\mathrm{B}}(\sigma_{t-1})$ has already been computed at time $t-1$ in Algorithm 2,
- $\mathrm{tr}\big(C^T\ B_{\mathrm{new}}\big)$ is a scalar product between the matrix C that contains $4(n-1)$ non-zero entries and the matrix B_{new}, and hence requires $8(n-1)$ operations,
- $\mathrm{tr}\big(S_{t-1}^T A^T S_{t-1}\ D\big)$ is a scalar product between the matrix D that contains less than $4(n-1)$ non-zero entries and the matrix $S_{t-1}^T A S_{t-1}$ which has already been computed at time $t-1$ in Algorithm 2, and hence requires $8(n-1)$ operations,
- $-\mathrm{tr}\Bigg(S_{t-1}^T \left(\frac{k^{\mathrm{out}} k^{\mathrm{in}^T}}{|E(A)|} \right)^T S_{t-1}\ D \Bigg)$ is a scalar product between the matrix D that contains less than $4(n-1)$ non-zero entries and the matrix $S_{t-1}^T \left(\frac{k^{\mathrm{out}} k^{\mathrm{in}^T}}{|E(A)|} \right) S_{t-1}$ whose entries that are multiplied by non-zero entries of D have already been computed while computing B_{new}, hence requires $8(n-1)$ operations.

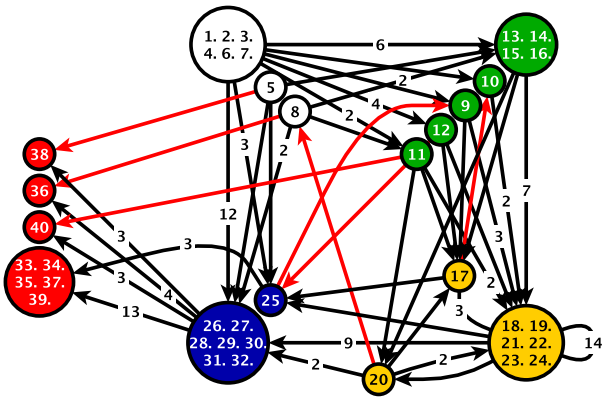
And finally, when one knows B_{new}, computing $Q_{\mathrm{B}}(\sigma_{\mathrm{new}})$ requires $24(n-1)+3$ operations.

In the end, computing $S_{\mathrm{new}}^T A S_{\mathrm{new}}$, B_{new}, and $Q_{\mathrm{B}}(\sigma_{\mathrm{new}})$ by incrementing $S_{t-1}^T A S_{t-1}$, B_{t-1}, and $Q_{\mathrm{B}}(\sigma_{t-1})$ requires $4mk + O(m)$ operations.

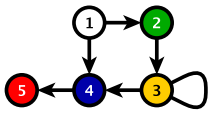
Similar analyzes for $Q_A^{\mathrm{EA}\sim}$ and Q_A^{RS} yield that these quality measures require of the order of the same number of operations as Q_A^{EA}. Conversely, since one has to compute a similarity matrix $Sim(A) \in \mathbb{R}^{m \times m}$ for $Q_{Sim(A)}^{\text{N-Uncut}}$, the complexity of that quality is at least of the order of m^2 operations.



(a)



(b)



(c)

Figure 4.2: Toy graph with 40 nodes, 128 edges and an underlying 5-block structure. The adjacency matrix of the graph is shown in (a). The graphical representation of the graph is shown in (b). Note that some nodes have been merged in the graphical representation for readability. The black edges are the ones coherent with the underlying 5-block structure, contrarily to the red ones. The graphical representation of the expected 5-role image graph is shown in (c).

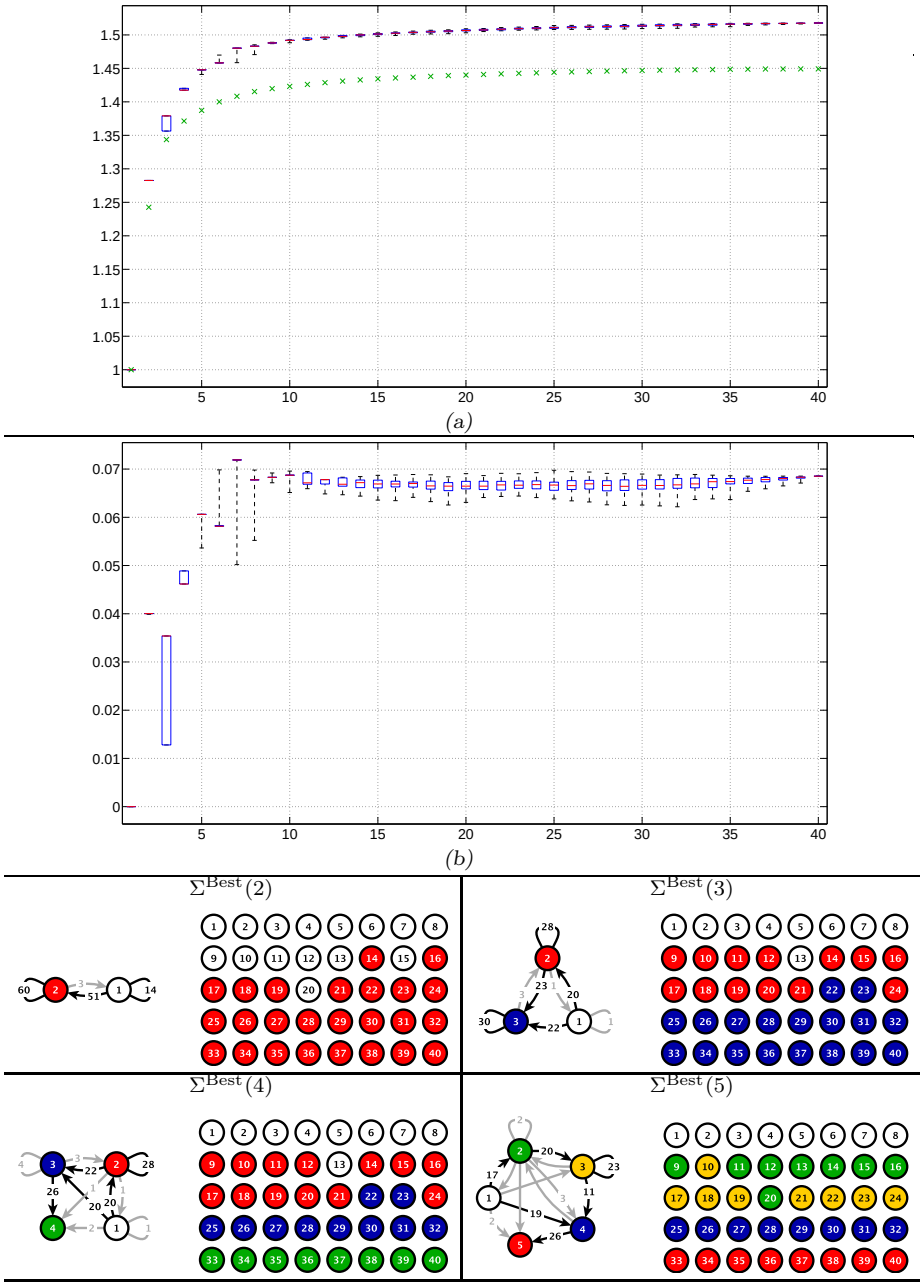


Figure 4.3: Results of Algorithm 2 on the graph shown in Figure 4.2 for the Blockmodel quality functions $Q^{\text{N-Uncut}}_{Sim(A)}(S)$ with $Sim(A) = S^{\text{Cooper-3}}_{\alpha=\rho(A)}(A, A)$. In (a), the abscissas are the elements of the domain of Σ^{Best}, i.e. $\mathbb{N}_{\leq 40}$ or the number of non-empty blocks of the image of Σ^{Best}. The box-plot in the ordinate associated to the abscissa n shows the range of $Q_{\text{B}}(\Sigma^{\text{Best}}(n))$ over 64 runs of Algorithm 2, and the green cross shows its expected value in the configuration model. (b) shows the difference between the box-plots and the green crosses in (a). $\Sigma^{\text{Best}}(i)$ shows the best partition with i non-empty blocks found with the associated image graph. The edges in black are the ones of the sparsest B in (4.1).

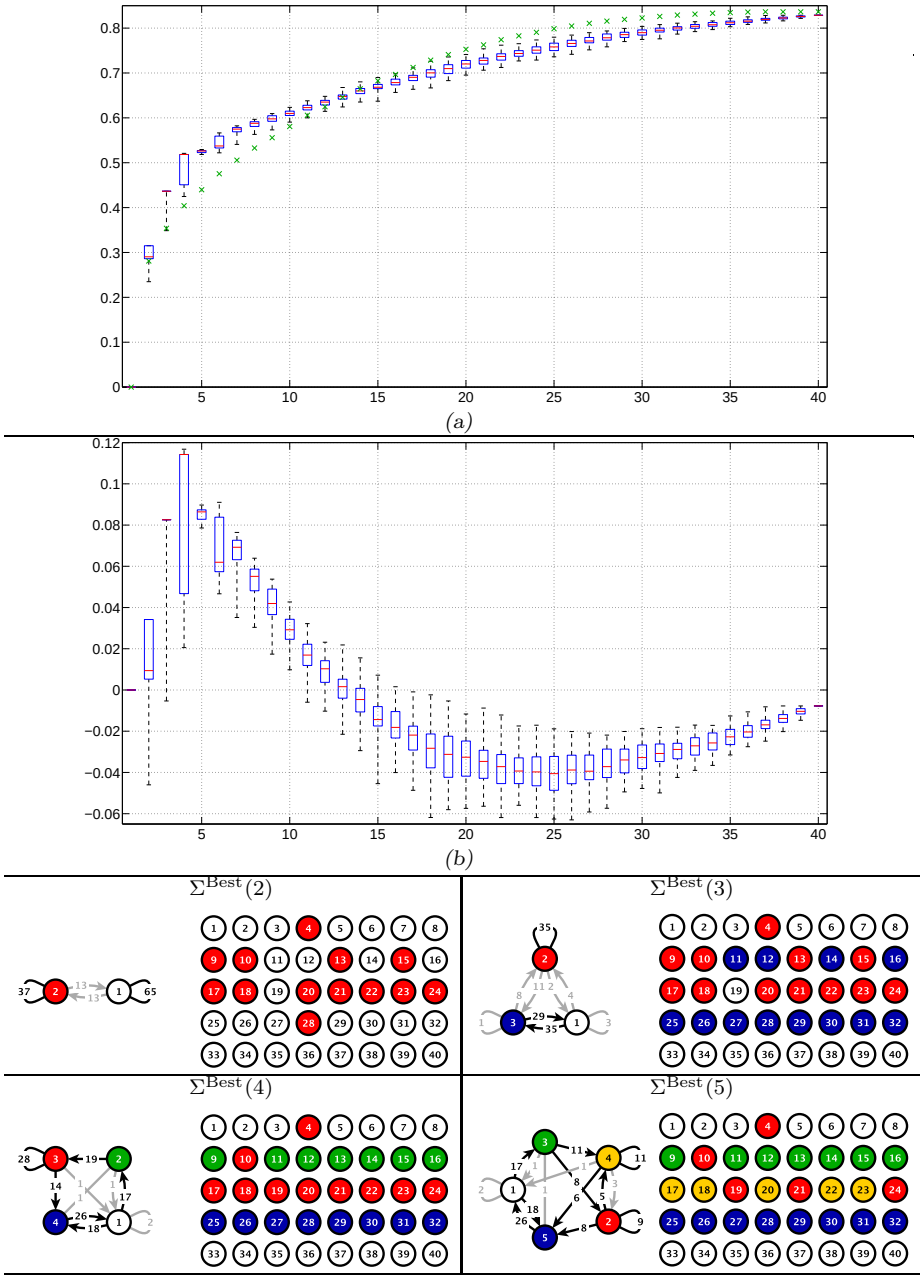


Figure 4.4: Results of Algorithm 2 on the graph shown in Figure 4.2 for the Blockmodel quality functions Q_A^{EA}. In (a), the abscissas are the elements of the domain of Σ^{Best}, i.e. $\mathbb{N}_{\leq 40}$ or the number of non-empty blocks of the image of Σ^{Best}. The box-plot in the ordinate associated to the abscissa n shows the range of $Q_{\mathrm{B}}(\Sigma^{\mathrm{Best}}(n))$ over 64 runs of Algorithm 2, and the green cross shows its expected value in the configuration model. (b) shows the difference between the box-plots and the green crosses in (a). $\Sigma^{\mathrm{Best}}(i)$ shows the best partition with i non-empty blocks found with the associated image graph. The edges in black are the ones of the sparsest B in $\mathrm{argmax}_B\, Q_A^{\mathrm{EA}}(B, \Sigma^{\mathrm{Best}}(i))$.

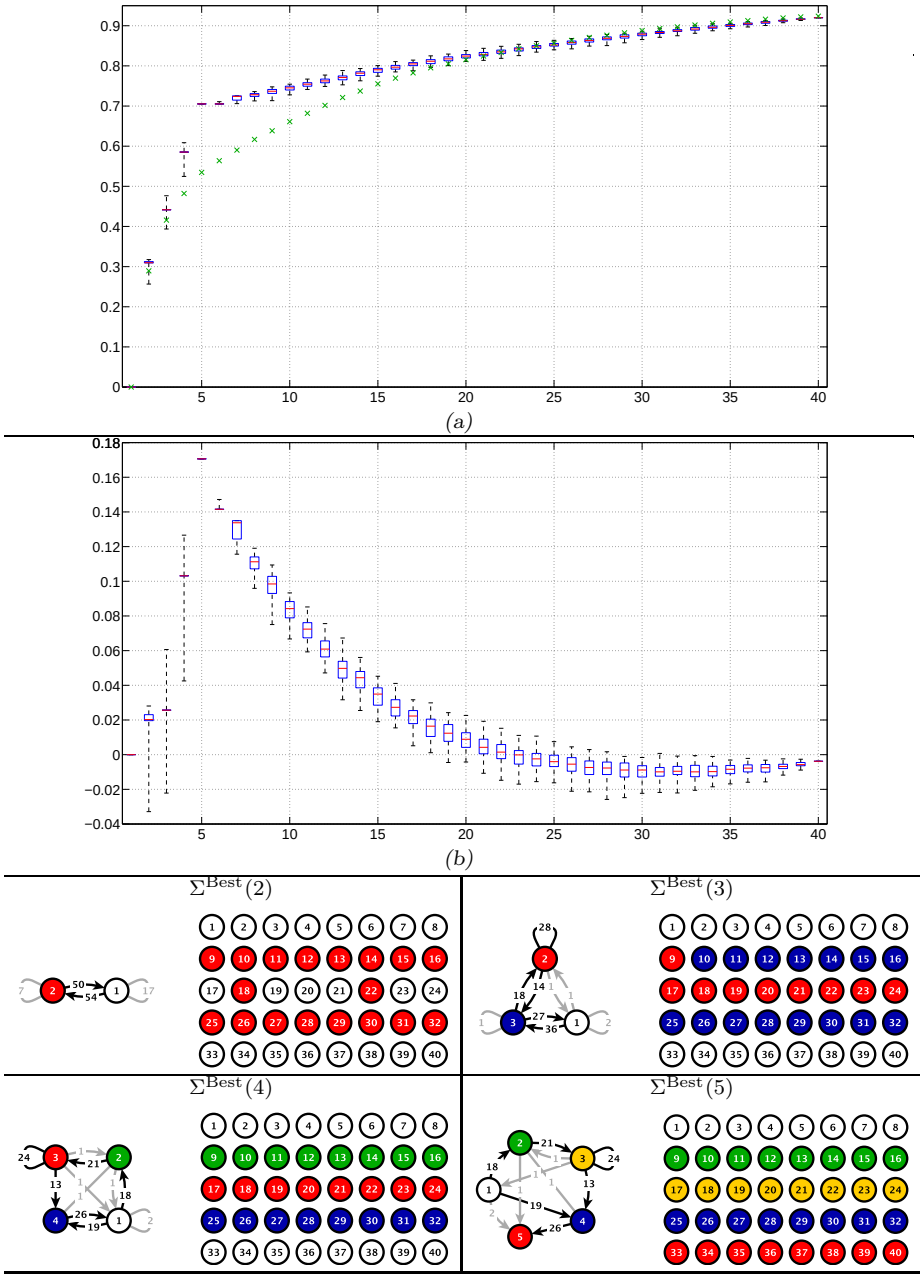


Figure 4.5: Results of Algorithm 2 on the graph shown in Figure 4.2 for the Blockmodel quality functions $Q_A^{\mathrm{EA}\sim}$. In (a), the abscissas are the elements of the domain of Σ^{Best}, i.e. $\mathbb{N}_{\leq 40}$ or the number of non-empty blocks of the image of Σ^{Best}. The box-plot in the ordinate associated to the abscissa n shows the range of $Q_{\mathrm{B}}(\Sigma^{\mathrm{Best}}(n))$ over 64 runs of Algorithm 2, and the green cross shows its expected value in the configuration model. (b) shows the difference between the box-plots and the green crosses in (a). $\Sigma^{\mathrm{Best}}(i)$ shows the best partition with i non-empty blocks found with the associated image graph. The edges in black are the ones of the sparsest B in $\mathrm{argmax}_B\, Q_A^{\mathrm{EA}\sim}(B, \Sigma^{\mathrm{Best}}(i))$.

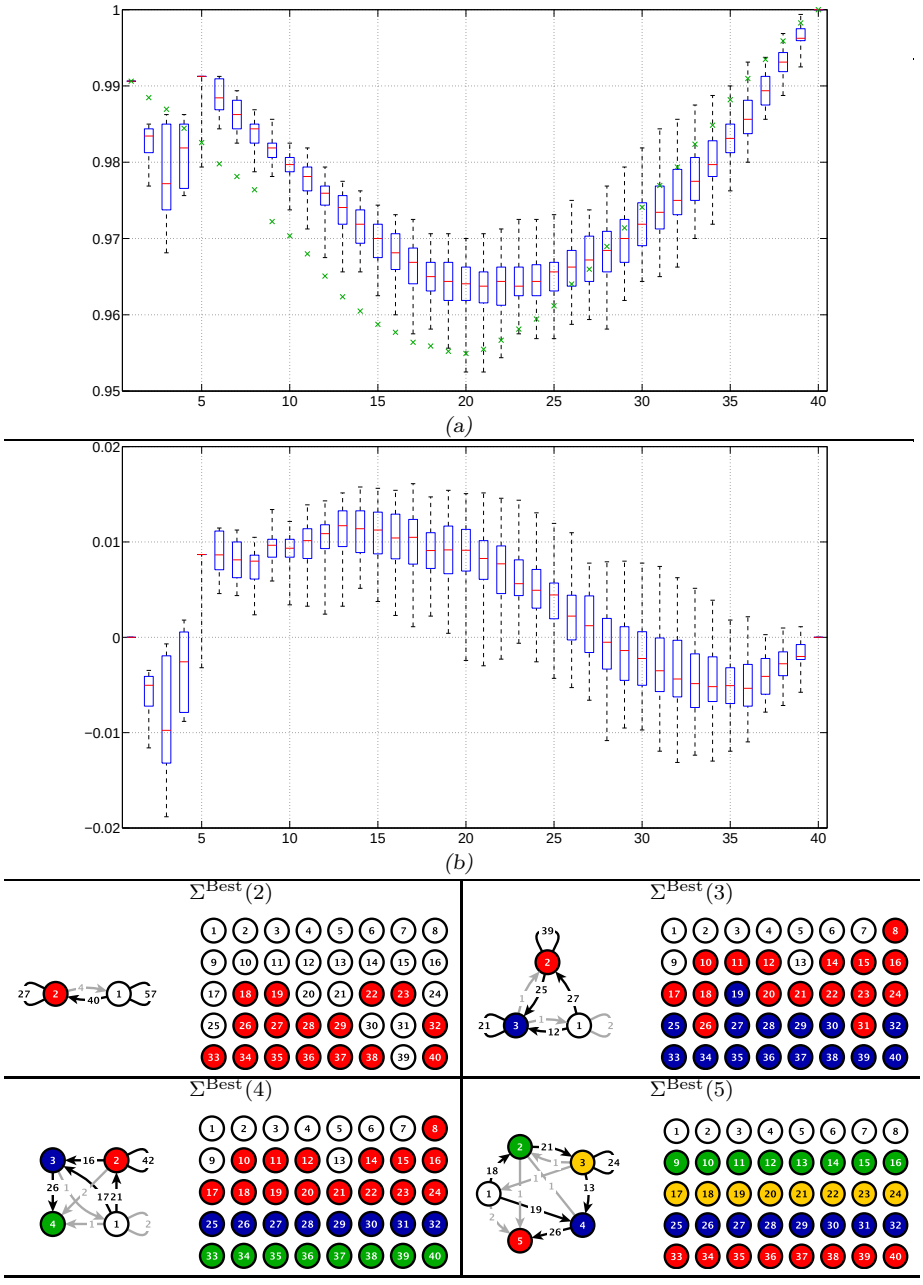


Figure 4.6: Results of Algorithm 2 on the graph shown in Figure 4.2 for the Blockmodel quality functions Q_A^{RS}. In (a), the abscissas are the elements of the domain of Σ^{Best}, i.e. $\mathbb{N}_{\leq 40}$ or the number of non-empty blocks of the image of Σ^{Best}. The box-plot in the ordinate associated to the abscissa n shows the range of $Q_{\mathrm{B}}(\Sigma^{\mathrm{Best}}(n))$ over 64 runs of Algorithm 2, and the green cross shows its expected value in the configuration model. (b) shows the difference between the box-plots and the green crosses in (a). $\Sigma^{\mathrm{Best}}(i)$ shows the best partition with i non-empty blocks found with the associated image graph. The edges in black are the ones of the sparsest B in $\mathrm{argmax}_B\, Q_A^{\mathrm{RS}}(B, \Sigma^{\mathrm{Best}}(i))$.

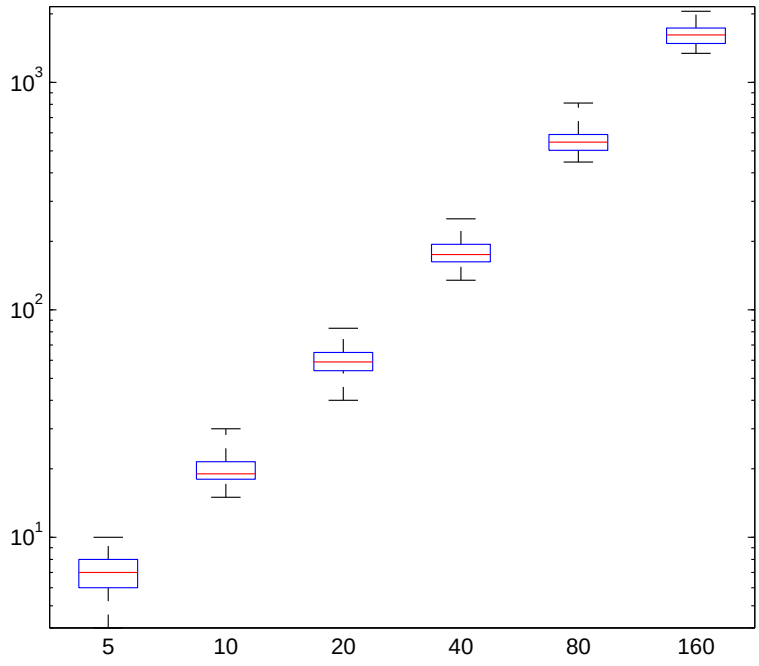


Figure 4.7: Box-plot over 64 runs of the number of iterations before convergence of Algorithm 2 with the Blockmodel quality function $Q_A^{\mathrm{EA}\sim}$ versus the size of the graph under consideration.

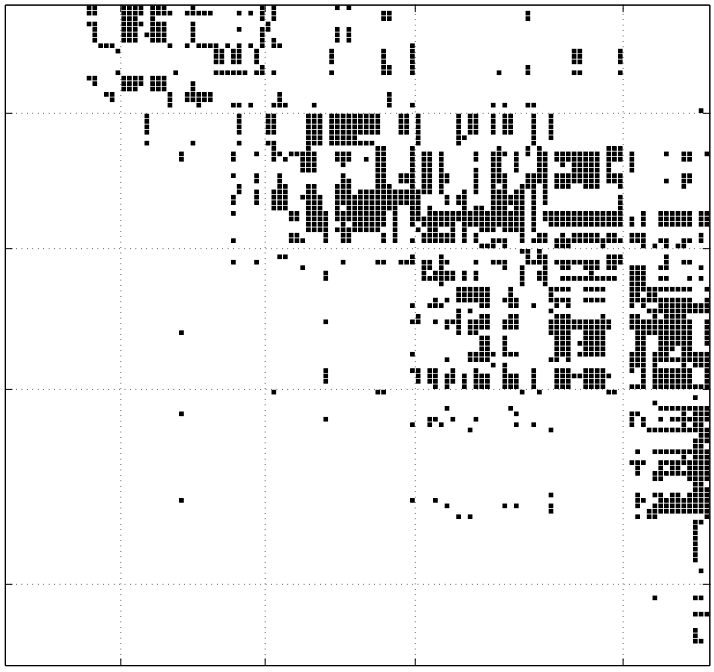
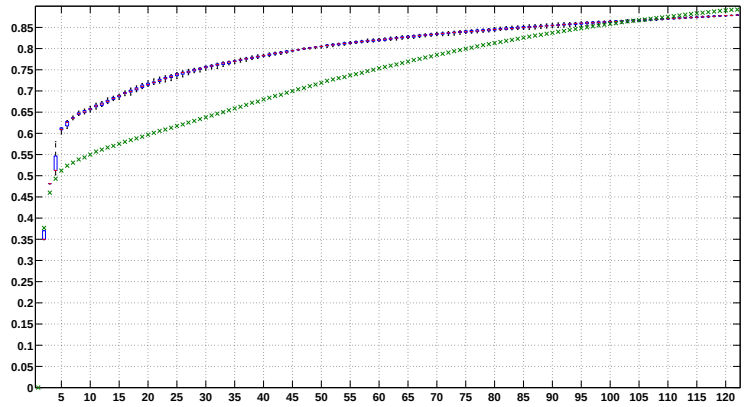
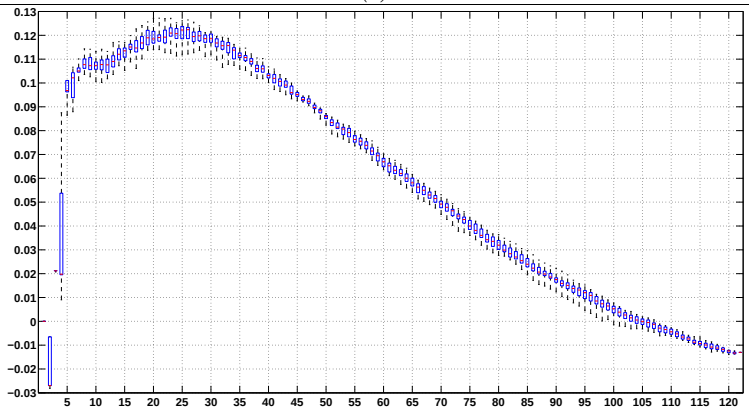


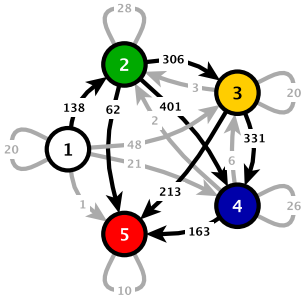
Figure 4.8: Adjacency matrix of the Baydry *network without weights in which the nodes have been rearranged in blocks as follows.* ***Block 1:*** *1. 2μm Spherical Phytoplankton, 2. Synedococcus, 3. Oscillatoria, 4. Small Diatoms (< 20μm), 5. Big Diatoms (> 20μm), 6. Dinoflagellates, 7. Other Phytoplankton, 8. Benthic Phytoplankton, 9. Thalassia, 10. Halodule, 11. Syringodium, 12. Drift Algae, 13. Epiphytes, 14. Free Bacteria, 15. Water Flagellates, 16. Water Cilitaes, 17. Benthic Flagellates, 18. Benthic Ciliates, 19. Meiofauna, 20. Green Turtle –* ***Block 2:*** *21. Acartia Tonsa, 22. Oithona nana, 23. Paracalanus, 24. Other Copepoda, 25. Meroplankton, 26. Other Zooplankton, 27. Sponges, 28. Bivalves, 29. Detritivorous Gastropods, 30. Epiphytic Gastropods, 31. Predatory Gastropods, 32. Detritivorous Polychaetes, 33. Suspension Feeding Polych, 34. Macrobenthos, 35. Benthic Crustaceans, 36. Detritivorous Amphipods, 37. Herbivorous Amphipods, 38. Isopods, 39. Herbivorous Shrimp, 40. Predatory Shrimp, 41. Pink Shrimp, 42. Thor Floridanus, 43. Detritivorous Crabs, 44. Omnivorous Crabs, 45. Sailfin Molly –* ***Block 3:*** *46. Coral, 47. Echinoderma, 48. Predatory Polychaetes, 49. Lobster, 50. Predatory Crabs, 51. Callinectus sapidus, 52. Stone Crab, 53. Sardines, 54. Anchovy, 55. Bay Anchovy, 56. Toadfish, 57. Halfbeaks, 58. Other Killifish, 59. Goldspotted killifish, 60. Rainwater killifish, 61. Silverside, 62. Other Horsefish, 63. Gulf Pipefish, 64. Dwarf Seahorse, 65. Mojarra, 66. Pinfish, 67. Mullet, 68. Blennies, 69. Code Goby, 70. Clown Goby, 71. Other Demersal Fishes –* ***Block 4:*** *72. Other Cnidaridae, 73. Rays, 74. Bonefish, 75. Lizardfish, 76. Catfish, 77. Eels, 78. Brotalus, 79. Needlefish, 80. Snook, 81. Jacks, 82. Pompano, 83. Other Snapper, 84. Gray Snapper, 85. Grunt, 86. Porgy, 87. Scianids, 88. Spotted Seatrout, 89. Red Drum, 90. Spadefish, 91. Parrotfish, 92. Flatfish, 93. Filefishes, 94. Puffer, 95. Other Pelagic Fishes, 96. Small Herons & Egrets, 97. Ibis, 98. Roseate Spoonbill, 99. Herbivorous Ducks, 100. Omnivorous Ducks, 101. Gruiformes, 102. Small Shorebirds, 103. Gulls & Terns, 104. Kingfisher, 105. Loggerhead Turtle, 106. Hawksbill Turtle, 107. Manatee –* ***Block 5:*** *108. Roots, 109. Sharks, 110. Tarpon, 111. Grouper, 112. Mackerel, 113. Barracuda, 114. Loon, 115. Greeb, 116. Pelican, 117. Comorant, 118. Big Herons & Egrets, 119. Predatory Ducks, 120. Raptors, 121. Crocodiles, 122. Dolphin*



(a)



(b)



(c)

Figure 4.9: Results of Algorithm 2 on the Baydry network for the Blockmodel quality functions Q_A^{EA}. In (a), the abscissas are the elements of the domain of Σ^{Best}, i.e. $\mathbb{N}_{\leq 122}$ or the number of non-empty blocks of the image of Σ^{Best}. The box-plot in the ordinate associated to the abscissa n shows the range of $Q_{\mathrm{B}}(\Sigma^{\mathrm{Best}}(n))$ over 16 runs of Algorithm 2, and the green cross shows its expected value in the configuration model. (b) shows the difference between the box-plots and the green crosses in (a). (c) shows the best partition with 5 non-empty blocks found with the associated image graph. The edges in black are the ones of the sparsest B in $\operatorname{argmax}_B Q_A^{\mathrm{EA}}(B, \Sigma^{\mathrm{Best}}(5))$.

Chapter 5

Trace Maximization Problems

5.1 Introduction

When comparing two matrices A and B it is often natural to allow for a class of transformations acting on these matrices. For instance, when comparing adjacency matrices A and B of two graphs with an equal number of nodes, one can allow symmetric permutations $P^T AP$ on one matrix in order to compare it to B, since this is merely a relabeling of the nodes of A. The so-called comparison then consists in finding the best match between A and B under this class of transformations.

A more general class of transformations would be that of unitary similarity transformations Q^*AQ, where Q is a unitary matrix. This leaves the eigenvalues of A unchanged but rotates its eigenvectors, which will of course play a role in the comparison between A and B. If A and B are of different order, say m and n, one may want to consider their restriction on a lower dimensional subspace:

$$U^*AU \quad \text{and} \quad V^*BV, \tag{5.1}$$

with U and V belonging to $V_{m,k}$ and $V_{n,k}$ respectively, and where $V_{m,k} = \left\{U \in \mathbb{C}^{m\times k} : U^*U = I_k\right\}$ denotes the *compact Stiefel manifold*. This yields two square matrices of equal dimension $k \leq \min(m, n)$, which can again be compared.

But one still needs to define a measure of comparison between these restrictions of A and B which clearly depends on U and V. Fraikin et *al.* [FNVD08] propose in this context to maximize the inner product between the *isometric projections*, U^*AU and V^*BV, namely:

$$\arg \max_{\substack{U^*U=I_k \\ V^*V=I_k}} \langle U^*AU, V^*BV \rangle := \Re \operatorname{tr}((U^*AU)^* (V^*BV)),$$

where $\Re$ denotes the real part of a complex number. They show this is also equivalent to

$$\arg \max_{\substack{X=VU^* \\ U^*U=I_k \\ V^*V=I_k}} \langle XA, BX \rangle = \Re \operatorname{tr}(A^*X^*BX),$$

and eventually show how this problem is linked to the notion of graph similarity introduced by Blondel et *al.* in [BGH$^+$04]. The graph similarity matrix S introduced in that paper also proposes a way of comparing two matrices A and B via the fixed point of a particular iteration. But it is shown in [FVD07] that this is equivalent to the optimization problem

$$\arg \max_{\|S\|_F=1} \langle SA, BS \rangle = \Re \operatorname{tr}((SA)^*BS)$$

or also

$$\arg \max_{\|S\|_F=1} \langle S, BSA^* \rangle = \Re \operatorname{tr}((SA)^*BS).$$

Notice that S also belongs to a Stiefel manifold, since $\operatorname{vec}(S) \in V_{mn,1}$.

In this chapter, we use a distance measure rather than an inner product to compare two matrices. As squared distance measure between two matrices M and N, we will use

$$\operatorname{dist}^2(M, N) = \|M - N\|_F^2 = \operatorname{tr}((M-N)^*(M-N)).$$

We will analyze distance minimization problems that are essentially the counterparts of the similarity measures defined above. These are

$$\arg \min_{\substack{U^*U=I_k \\ V^*V=I_k}} \operatorname{dist}^2(U^*AU, V^*BV),$$

$$\arg \min_{\substack{X=VU^* \\ U^*U=I_k \\ V^*V=I_k}} \operatorname{dist}^2(XA, BX),$$

and

$$\arg \min_{\substack{X=VU^* \\ U^*U=I_k \\ V^*V=I_k}} \operatorname{dist}^2(X, BXA^*),$$

for the problems involving two isometries U and V. Notice that these three distance problems are not equivalent although the corresponding inner product problems are equivalent.

Similarly, we will analyze the two problems

$$\arg \min_{\|S\|_F=1} \operatorname{dist}^2(SA, BS) = \operatorname{tr}((SA - BS)^*(SA - BS))$$

and

$$\arg \min_{\|S\|_F=1} \operatorname{dist}^2(S, BSA^*) = \operatorname{tr}((S - BSA^*)^*(S - BSA^*)),$$

for the problems involving a single matrix S. Again, these are not equivalent in their distance formulation although the corresponding inner product problems are equivalent.

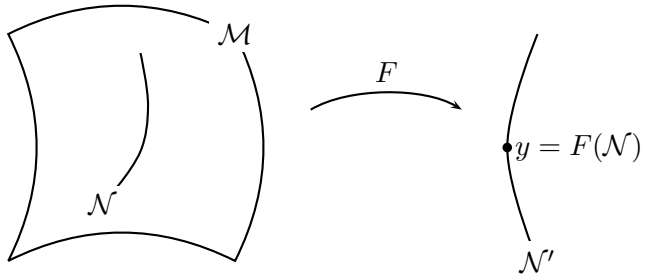
We will develop optimality conditions for those matrix comparison problems, indicate their relations with existing problems from the literature and give an analytic solution for particular matrices A and B.

5.2 Preliminaries on Riemannian Optimization

All those problems are defined on feasible sets that have a *manifold* structure. Roughly speaking, this means that the feasible set is locally smoothly identified with $\mathbb{R}^d$, where d is the dimension of the manifold. Optimization on a manifold generalizes optimization in $\mathbb{R}^d$ while retaining the concept of smoothness. In this section, we recall the essential background on optimization on manifolds, and refer the reader to [AMS08] for details.

A well known and largely used class of manifolds is the class of embedded submanifolds. The submersion theorem gives a useful sufficient condition to prove that a subset of a manifold $\mathcal{M}$ is an embedded submanifold of $\mathcal{M}$. If there exists a smooth mapping $F : \mathcal{M} \to \mathcal{N}'$ between two manifolds of dimension d_m and $d'_n (< d_m)$ and $y \in \mathcal{N}'$ such that the

rank of F is equal to d'_n at each point of $\mathcal{N} := F^{-1}(y)$, then $\mathcal{N}$ is a embedded submanifold of $\mathcal{M}$ and the dimension of $\mathcal{N}$ is $d_m - d'_n$.



Example The unitary group $\mathrm{U}(n) = \{Q \in \mathbb{C}^{n\times n} : Q^*Q = I_n\}$ is an embedded submanifold of $\mathbb{C}^{n\times n}$. Indeed, consider the function

$$F : \mathbb{C}^{n\times n} \to \mathcal{S}_{\mathrm{Her}}(n) : Q \mapsto Q^*Q - I_n$$

where $\mathcal{S}_{\mathrm{Her}}(n)$ denotes the set of Hermitian matrices of order n. Clearly, $\mathrm{U}(n) = F^{-1}(0_n)$. It remains to show for all $\hat{H} \in \mathcal{S}_{\mathrm{Her}}(k)$, there exists an $H \in \mathbb{C}^{n\times n}$ such that $\mathrm{D}F(Q) \cdot H = Q^*H + H^*Q = \hat{H}$. It is easy to see that $\mathrm{D}F(Q) \cdot (Q\hat{H}/2) = \hat{H}$, and according to the submersion theorem, it follows that $\mathrm{U}(n)$ is an embedded submanifold of $\mathbb{C}^{n\times n}$. The dimension of $\mathbb{C}^{n\times n}$ and $\mathcal{S}_{\mathrm{Her}}(n)$ are $2n^2$ and n^2 respectively. Hence $\mathrm{U}(n)$ is of dimension n^2.

In our problems, embedding spaces are matrix-Euclidean spaces $\mathbb{C}^{m\times k} \times \mathbb{C}^{n\times k}$ and $\mathbb{C}^{n\times m}$ which have a trivial manifold structure since $\mathbb{C}^{m\times k} \times \mathbb{C}^{n\times k} \simeq \mathbb{R}^{2mnk^2}$ and $\mathbb{C}^{n\times m} \simeq \mathbb{R}^{2mn}$. For each problem, we further analyze whether or not the feasible set is an embedded submanifold of their embedding space.

When working with a function on a manifold $\mathcal{M}$, one may be interested in having a local linear approximation of that function. Let M be an element of $\mathcal{M}$ and $\mathfrak{F}_M(\mathcal{M})$ denote the set of smooth real-valued functions defined on a neighborhood of M.

Definition 5.1 *A tangent vector ξ_M to a manifold $\mathcal{M}$ at a point M is a mapping from $\mathfrak{F}_M(\mathcal{M})$ to $\mathbb{R}$ such that there exists a curve γ on $\mathcal{M}$ with $\gamma(0) = M$, satisfying*

$$\xi_M f = \left.\frac{\mathrm{d}f(\gamma(t))}{\mathrm{d}t}\right|_{t=0}, \qquad \forall f \in \mathfrak{F}_M(\mathcal{M}).$$

Such a curve γ is said to realize the tangent vector ξ_M.

So, the only thing we need to know about a curve γ in order to compute the first-order variation of a real-valued function f at $\gamma(0)$ along γ is the tangent vector ξ_M realized by γ. The tangent space to $\mathcal{M}$ at M, denoted by $T_M\mathcal{M}$, is the set of all tangent vectors to $\mathcal{M}$ at M and it admits a structure of vector space over $\mathbb{R}$. When considering an embedded submanifold in a Euclidean space $\mathcal{E}$, any tangent vector ξ_M of the manifold is equivalent to a vector E of the Euclidean space. Indeed, let $\hat{f}$ be any differentiable continuous extension of f on $\mathcal{E}$, we have

$$\xi_M f := \left.\frac{\mathrm{d}f(\gamma(t))}{\mathrm{d}t}\right|_{t=0} = \mathrm{D}\hat{f}(M)\cdot E, \tag{5.2}$$

where E is $\dot{\gamma}(0)$ and D is the directional derivative operator

$$\mathrm{D}\hat{f}(M)\cdot E = \lim_{t\to 0}\frac{\hat{f}(M+tE)-\hat{f}(M)}{t}.$$

The tangent space reduces to a linear subspace of the original space $\mathcal{E}$.
Example Let $\gamma(t)$ be a curve on the unitary group $\mathrm{U}(n)$ passing through Q at $t=0$, *i.e.* $\gamma(t)^*\gamma(t) = I_n$ and $\gamma(0) = Q$. Differentiating with respect to t yields

$$\dot{\gamma}(0)^*Q + Q^*\dot{\gamma}(0) = 0_n.$$

One can see from Equation (5.2) that the tangent space to $\mathrm{U}(n)$ at Q is contained in

$$\left\{E \in \mathbb{C}^{n\times n} : E^*Q + Q^*E = 0_n\right\} = \left\{Q\Omega \in \mathbb{C}^{n\times n} : \Omega^* + \Omega = 0_n\right\}. \tag{5.3}$$

Moreover, this set is a vector space over $\mathbb{R}$ of dimension n^2, and hence is the tangent space itself.

Let g_M be an inner product defined on the tangent plane $T_M\mathcal{M}$. The gradient of f at M, denoted $\operatorname{grad} f(M)$, is defined as the unique element of the tangent plane $T_M\mathcal{M}$, that satisfies

$$\xi_M f = g_M(\operatorname{grad} f(M), \xi_M), \qquad \forall \xi_M \in T_M\mathcal{M}.$$

The gradient, together with the inner product, fully characterizes the local first order approximation of a smooth function defined on the manifold. In the case of an embedded manifold of a Euclidean space $\mathcal{E}$, since

$T_M\mathcal{M}$ is a linear subspace of $T_M\mathcal{E}$, an inner product $\hat{g}_M$ on $T_M\mathcal{E}$ generates by restriction an inner product g_M on $T_M\mathcal{M}$. The orthogonal complement of $T_M\mathcal{M}$ with respect to $\hat{g}_M$ is called the normal space to $\mathcal{M}$ at M and denoted by $(T_M\mathcal{M})^\perp$. The gradient of a smooth function $\hat{f}$, defined on the embedding manifold may be decomposed into its orthogonal projection on the tangent and normal space, respectively

$$\mathrm{P}_M \mathrm{grad}\, \hat{f}(M) \qquad \text{and} \qquad \mathrm{P}_M^\perp \mathrm{grad}\, \hat{f}(M),$$

and it follows that the gradient of f (the restriction of $\hat{f}$ on $\mathcal{M}$) is the projection on the tangent space of the gradient of $\hat{f}$

$$\mathrm{grad}\, f(M) = \mathrm{P}_M \mathrm{grad}\, \hat{f}(M).$$

If $T_M\mathcal{M}$ is endowed with an inner product g_M for all $M \in \mathcal{M}$ and g_M varies smoothly with M, then $\mathcal{M}$ is termed a *Riemannian manifold*.
Example Let A and B be two Hermitian matrices. We define

$$\hat{f} : \mathbb{C}^{n\times n} \to \mathbb{R} : Q \mapsto \Re\,\mathrm{tr}(Q^*AQB),$$

and f its restriction on the unitary group $\mathrm{U}(n)$. We have

$$\mathrm{D}\hat{f}(Q)\cdot E = 2\Re\,\mathrm{tr}(E^*AQB).$$

We endow the tangent space $T_Q\mathbb{C}^{n\times n}$ with an inner product

$$\hat{g} : T_Q\mathbb{C}^{n\times n} \times T_Q\mathbb{C}^{n\times n} \to \mathbb{R} : E, F \mapsto \Re\,\mathrm{tr}(E^*F),$$

and the gradient of $\hat{f}$ at Q is then given by $\mathrm{grad}\, \hat{f}(Q) = 2AQB$. One can further define an orthogonal projection on $T_Q\mathrm{U}(n)$

$$\mathrm{P}_Q E := E - Q\,\mathrm{Her}(Q^*E),$$

where $\mathrm{Her}(\cdot)$ stands for

$$\mathrm{Her}(\cdot) : X \mapsto (X + X^*)/2,$$

and the gradient of f at Q is given by $\mathrm{grad}\, f(Q) = \mathrm{P}_Q \mathrm{grad}\, \hat{f}(Q)$.

Those relations are useful when one wishes to analyze optimization problems, and will hence be further developed for the problems we are interested in.

5.3 The Matrix Comparison Problems and their Geometry

Below we look at the various problems introduced earlier and focus on the first problem to make these ideas more explicit.

Problem 5.1 *Given $A \in \mathbb{C}^{m\times m}$ and $B \in \mathbb{C}^{n\times n}$, let*

$$\hat{f} : \mathbb{C}^{m\times k} \times \mathbb{C}^{n\times k} \to \mathbb{C} : (U, V) \mapsto \hat{f}(U, V) = \mathrm{dist}^2(U^*AU, V^*BV),$$

find the minimizer of

$$f : V_{m,k} \times V_{n,k} \to \mathbb{C} : (U, V) \mapsto f(U, V) = \hat{f}(U, V),$$

where

$$V_{m,k} = \left\{ U \in \mathbb{C}^{m\times k} : U^*U = I_k \right\}$$

denotes the compact Stiefel manifold.

Let $A = (A_1, A_2)$ and $B = (B_1, B_2)$ be pairs of matrices. We define the following useful operations:

- an entrywise product, $A \diamond B = (A_1B_1, A_2B_2)$,
- a contraction product, $A \star B = A_1B_1 + A_2B_2$, and
- a conjugate-transpose operation, $A^* = (A_1^*, A_2^*)$.

The definitions of the binary operations, $\diamond$ and $\star$, are (for readability) extended to single matrices when one has to deal with pairs of identical matrices. Let, for instance, $A = (A_1, A_2)$ be a pair of matrices and B be a single matrix, we define

$$\begin{aligned} A \diamond B &= (A_1, A_2) \diamond B = (A_1, A_2) \diamond (B, B) = (A_1B, A_2B) \\ A \star B &= (A_1, A_2) \star B = (A_1, A_2) \star (B, B) = A_1B + A_2B. \end{aligned}$$

The feasible set of Problem 5.1 is given by the cartesian product of two compact Stiefel manifolds, namely $\mathcal{M} = V_{m,k} \times V_{n,k}$ and is hence a manifold itself (*cf.* [AMS08]). Moreover, we can prove that $\mathcal{M}$ is an embedded submanifold of

$$\mathcal{E} := \mathbb{C}^{m\times k} \times \mathbb{C}^{n\times k}.$$

Indeed, consider the function

$$F : \mathcal{E} \to \mathcal{S}_{\mathrm{Her}}(k) \times \mathcal{S}_{\mathrm{Her}}(k) : M \mapsto M^* \diamond M - (I_k, I_k)$$

where $\mathcal{S}_{\mathrm{Her}}(k)$ denotes the set of Hermitian matrices of order k. Clearly, $\mathcal{M} = F^{-1}(0_k, 0_k)$. It remains to show that each point $M \in \mathcal{M}$ is a regular value of F which means that F has full rank, *i.e.* for all $\hat{Z} \in \mathcal{S}_{\mathrm{Her}}(k) \times \mathcal{S}_{\mathrm{Her}}(k)$, there exists $Z \in \mathcal{E}$ such that $\mathrm{D}F(M) \cdot Z = \hat{Z}$. It is easy to see that $\mathrm{D}F(M) \cdot (M \diamond \hat{Z}/2) = \hat{Z}$, and according to the submersion theorem, it follows that $\mathcal{M}$ is an embedded submanifold of $\mathcal{E}$.

The tangent space to $\mathcal{E}$ at a point $M = (U, V) \in \mathcal{E}$ is the embedding space itself (*i.e.* $T_M\mathcal{E} \simeq \mathcal{E}$), whereas the tangent space to $\mathcal{M}$ at a point $M = (U, V) \in \mathcal{M}$ is given by

$$\begin{aligned} T_M\mathcal{M} &:= \{\dot{\gamma}(0) : \gamma, \text{ a differentiable curve on } \mathcal{M} \text{ with } \gamma(0) = M\} \\ &= \{\xi = (\xi_U, \xi_V) : \mathrm{Her}(\xi^* \diamond M) = 0\} \\ &= \left\{ M \diamond \begin{pmatrix} \Omega_U \\ \Omega_V \end{pmatrix} + M_\perp \diamond \begin{pmatrix} K_U \\ K_V \end{pmatrix} : \Omega_U, \Omega_V \in \mathcal{S}_{\mathrm{s-Her}}(k) \right\}, \end{aligned}$$

where $M_\perp = (U_\perp, V_\perp)$ with $U_\perp$ and $V_\perp$ any orthogonal complement of respectively U and V, and where $\mathcal{S}_{\mathrm{s-Her}}(k)$ denotes the set of skew-Hermitian matrices of order k. We endow the tangent space $T_M\mathcal{E}$ with an inner product:

$$\hat{g}_M(\cdot, \cdot) : T_M\mathcal{E} \times T_M\mathcal{E} \to \mathbb{C} : \xi, \zeta \mapsto \hat{g}_M(\xi, \zeta) = \Re \operatorname{tr}(\xi^* \star \zeta),$$

and define its restriction on the tangent space $T_M\mathcal{M} (\subset T_M\mathcal{E})$:

$$g_M(\cdot, \cdot) : T_M\mathcal{M} \times T_M\mathcal{M} \to \mathbb{C} : \xi, \zeta \mapsto g_M(\xi, \zeta) = \hat{g}_M(\xi, \zeta).$$

One may now define the normal space to $\mathcal{M}$ at a point $M \in \mathcal{M}$:

$$\begin{aligned} T_M^\perp\mathcal{M} &:= \{\xi : \hat{g}_M(\xi, \zeta) = 0, \ \forall \zeta \in T_M\mathcal{M}\} \\ &= \{M \diamond (H_U, H_V) : H_U, H_V \in \mathcal{S}_{\mathrm{Her}}(k)\}, \end{aligned}$$

where $\mathcal{S}_{\mathrm{Her}}(k)$ denotes the set of Hermitian matrices of order k.

Problem 5.2 *Given $A \in \mathbb{C}^{m\times m}$ and $B \in \mathbb{C}^{n\times n}$, let*

$$\hat{f} : \mathbb{C}^{n\times m} \to \mathbb{C} : X \mapsto \hat{f}(X) = \mathrm{dist}^2\,(XA, BX)$$

find the minimizer of

$$f : \mathcal{M} \to \mathbb{C} : X \mapsto f(X) = \hat{f}(X),$$

where $\mathcal{M} = \{VU^* \in \mathbb{C}^{n\times m} : (U, V) \in V_{m,k} \times V_{n,k}\}$.

$\mathcal{M}$ is a smooth and connected manifold. Indeed, let $\Sigma := \left[\begin{smallmatrix} I_k & 0 \\ 0 & 0 \end{smallmatrix}\right]$ be an element of $\mathcal{M}$. Every $X \in \mathcal{M}$ is congruent to Σ by the congruence action $((\tilde{U}, \tilde{V}), X) \mapsto \tilde{V}^* X \tilde{U}$, $(\tilde{U}, \tilde{V}) \in \mathrm{U}(m) \times \mathrm{U}(n)$, where $\mathrm{U}(n) = \{U \in \mathbb{C}^{n\times n} : U^*U = I_n\}$ denotes the unitary group of degree n. The set $\mathcal{M}$ is an orbit of this smooth complex algebraic Lie group action of $\mathrm{U}(m)\times\mathrm{U}(n)$ on $\mathbb{C}^{n\times m}$ and therefore a smooth manifold [HM94, App. C]. $\mathcal{M}$ is the image of the connected subset $\mathrm{U}(m) \times \mathrm{U}(n)$ of the continuous (and in fact smooth) map $\pi : \mathrm{U}(m) \times \mathrm{U}(n) \to \mathbb{C}^{n\times m}, \pi(\tilde{U}, \tilde{V}) = \tilde{V}^* X \tilde{U}$, and hence is also connected.
The tangent space to $\mathcal{M}$ at a point $X = VU^* \in \mathcal{M}$ is

$$\begin{aligned} T_X\mathcal{M} &:= \{\dot{\gamma}(0) : \gamma \text{ curve on } \mathcal{M} \text{ with } \gamma(0) = X\} \\ &= \{\xi_V U^* + V\xi_U^* : \mathrm{Her}(V^*\xi_V) = \mathrm{Her}(U^*\xi_U) = 0_k\} \\ &= \{V\Omega U^* + VK_U^*U_\perp^* + V_\perp K_V U^* : \Omega \in \mathcal{S}_{\mathrm{s-Her}}(k)\}\,. \end{aligned}$$

We endow the tangent space $T_X\mathbb{C}^{n\times m} \simeq \mathbb{C}^{n\times m}$ with an inner product:

$$\hat{g}_X(\cdot,\cdot) : T_X\mathbb{C}^{n\times m} \times T_X\mathbb{C}^{n\times m} \mapsto \mathbb{C} : \xi, \zeta \to \hat{g}_X(\xi, \zeta) = \Re\,\mathrm{tr}(\xi^*\zeta),$$

and define its restriction on the tangent space $T_X\mathcal{M} (\subset T_X\mathcal{E})$:

$$g_X(\cdot,\cdot) : T_X\mathcal{M} \times T_X\mathcal{M} \mapsto \mathbb{C} : \xi, \zeta \to g_X(\xi, \zeta) = \hat{g}_X(\xi, \zeta).$$

One may now define the normal space to $\mathcal{M}$ at a point $X \in \mathcal{M}$:

$$\begin{aligned} T_X^\perp\mathcal{M} &:= \{\xi : \hat{g}_X(\xi, \zeta) = 0,\ \forall \zeta \in T_X\mathcal{M}\} \\ &= \{VHU^* + V_\perp KU_\perp^* : H \in \mathcal{S}_{\mathrm{Her}}(k)\}\,. \end{aligned}$$

Problem 5.3 *Given $A \in \mathbb{C}^{m\times m}$ and $B \in \mathbb{C}^{n\times n}$, let*

$$\hat{f} : \mathbb{C}^{n\times m} \to \mathbb{C} : X \mapsto \hat{f}(X) = \mathrm{dist}^2\left(X, BXA^*\right)$$

find the minimizer of

$$f : \mathcal{M} \to \mathbb{C} : X \mapsto f(X) = \hat{f}(X),$$

where $\mathcal{M} = \{VU^* \in \mathbb{C}^{n\times m} : (U, V) \in V_{m,k} \times V_{n,k}\}$.

Since they have the same feasible set, developments obtained for Problem 5.2 hold also for Problem 5.3.

Problem 5.4 *Given $A \in \mathbb{C}^{m\times m}$ and $B \in \mathbb{C}^{n\times n}$, let*

$$\hat{f} : \mathbb{C}^{n\times m} \to \mathbb{C} : S \mapsto \hat{f}(S) = \mathrm{dist}^2(SA, BS)$$

find the minimizer of

$$f : \mathcal{M} \to \mathbb{C} : X \mapsto f(X) = \hat{f}(X),$$

where $\mathcal{M} = \{S \in \mathbb{C}^{n\times m} : \|S\|_F = 1\}$.

The tangent space to $\mathcal{M}$ at a point $S \in \mathcal{M}$ is

$$T_S\mathcal{M} = \{\xi : \Re\,\mathrm{tr}(\xi^* S) = 0\}\,.$$

We endow the tangent space $T_S\mathbb{C}^{n\times m} \simeq \mathbb{C}^{n\times m}$ with an inner product:

$$\hat{g}_S(\cdot,\cdot) : T_S\mathbb{C}^{n\times m} \times T_S\mathbb{C}^{n\times m} \mapsto \mathbb{C} : \xi, \zeta \to \hat{g}_S(\xi, \zeta) = \Re\,\mathrm{tr}(\xi^*\zeta),$$

and define its restriction on the tangent space $T_S\mathcal{M} (\subset T_S\mathcal{E})$:

$$g_S(\cdot,\cdot) : T_S\mathcal{M} \times T_S\mathcal{M} \mapsto \mathbb{C} : \xi, \zeta \to g_S(\xi, \zeta) = \hat{g}_S(\xi, \zeta).$$

One may now define the normal space to $\mathcal{M}$ at a point $S \in \mathcal{M}$:

$$T_S^{\perp}\mathcal{M} := \{\xi : \hat{g}_S(\xi, \zeta) = 0,\ \forall \zeta \in T_S\mathcal{M}\} = \{\alpha S : \alpha \in \mathbb{R}\}$$

Problem 5.5 *Given $A \in \mathbb{C}^{m\times m}$ and $B \in \mathbb{C}^{n\times n}$, let*

$$\hat{f} : \mathbb{C}^{n\times m} \to \mathbb{C} : S \mapsto \hat{f}(S) = \mathrm{dist}^2(S, BSA^*)$$

find the minimizer of

$$f : \mathcal{M} \to \mathbb{C} : X \mapsto f(X) = \hat{f}(X),$$

where $\mathcal{M} = \{S \in \mathbb{C}^{n\times m} : \|S\|_F = 1\}$.

Since they have the same feasible set, developments obtained for Problem 5.4 also hold for Problem 5.5.

5.4 Optimality conditions

Our problems are optimization problems of smooth functions defined on a compact domain $\mathcal{M}$, and therefore there always exists an optimal solution $M \in \mathcal{M}$ where the first order optimality condition is satisfied,

$$\operatorname{grad} f(M) = 0. \tag{5.4}$$

We study the stationary points of Problem 5.1 in detail, and we show how the other problems can be tackled.

Analysis of Problem 5.1

We first analyze this optimality condition for Problem 5.1. For any $(W, Z) \in T_M\mathcal{E}$, we have

$$\begin{aligned} &\mathrm{D}\hat{f}(U,V)\cdot(W,Z) \\ &\quad= 2\Re\operatorname{tr}\begin{pmatrix}(W^*AU + U^*AW - Z^*BV - V^*BZ)^*\\ (U^*AU - V^*BV)\end{pmatrix} \\ &\quad= \hat{g}_{(U,V)}\left(2\begin{pmatrix}AU\Delta_{AB}^* + A^*U\Delta_{AB}\\ BV\Delta_{BA}^* + B^*V\Delta_{BA}\end{pmatrix}, (W,Z)\right), \end{aligned} \tag{5.5}$$

with $\Delta_{AB} := U^*AU - V^*BV =: -\Delta_{BA}$, and hence the gradient of $\hat{f}$ at a point $(U, V) \in \mathcal{E}$ is

$$\operatorname{grad}\hat{f}(U,V) = 2\begin{pmatrix}AU\Delta_{AB}^* + A^*U\Delta_{AB}\\ BV\Delta_{BA}^* + B^*V\Delta_{BA}\end{pmatrix}. \tag{5.6}$$

Since the normal space $T_M^\perp\mathcal{M}$ is the orthogonal complement of the tangent space $T_M\mathcal{M}$, one can, for any $M \in \mathcal{M}$, decompose any $E \in \mathcal{E}$ into its orthogonal projections on $T_M\mathcal{M}$ and $T_M^\perp\mathcal{M}$:

$$\mathrm{P}_M E := E - \mathrm{P}_M^\perp E \quad \text{and} \quad \mathrm{P}_M^\perp E := M \diamond \operatorname{Her}(M^* \diamond E). \tag{5.7}$$

The gradient of f at a point $(U, V) \in \mathcal{M}$ is

$$\operatorname{grad} f(U,V) = \mathrm{P}_M \operatorname{grad}\hat{f}\,(M). \tag{5.8}$$

For our problem, the first order optimality condition (5.4) yields, by means of (5.6), (5.7) and (5.8)

$$\begin{pmatrix} AU\Delta_{AB}^* + A^*U\Delta_{AB} \\ BV\Delta_{BA}^* + B^*V\Delta_{BA} \end{pmatrix} = \begin{pmatrix} U \\ V \end{pmatrix} \diamond \operatorname{Her}\begin{pmatrix} U^*AU\Delta_{AB}^* + U^*A^*U\Delta_{AB} \\ V^*BV\Delta_{BA}^* + V^*B^*V\Delta_{BA} \end{pmatrix}. \tag{5.9}$$

Observe that f is constant on the equivalence classes

$$[(U,V)] = \{(U,V) \diamond Q : Q \in \mathrm{U}(k)\},$$

and that any point of $[(U,V)]$ is a stationary point of f whenever (U,V) is.

We consider the special case where U^*AU and V^*BV are simultaneously diagonalizable by a unitary matrix at all stationary points (U,V), *i.e.* eigendecomposition of U^*AU and V^*BV are respectively WD_AW^* and WD_BW^*, with $W \in \mathrm{U}(k)$ and $D_A = \operatorname{diag}\left(\theta_1^A, \cdots, \theta_k^A\right)$, $D_B = \operatorname{diag}\left(\theta_1^B, \cdots, \theta_k^B\right)$. This happens when A and B are both Hermitian. Indeed, in that case, applying $\begin{pmatrix} U^* \\ V^* \end{pmatrix} \diamond$ on the left of (5.9) yields

$$U^*AU\, V^*BV = V^*BV\, U^*AU,$$

which implies that U^*AU and V^*BV have the same eigenvectors. The cost function at stationary points simply reduces to $\sum_{i=1}^{k} \left|\theta_i^A - \theta_i^B\right|^2$ and the minimization problem roughly consists in finding the isometric projections U^*AU, V^*BV such that their eigenvalues are pairwise as near as possible.
More precisely, the first order optimality condition becomes

$$\left[\begin{pmatrix} A \\ B \end{pmatrix} \diamond \begin{pmatrix} U \\ V \end{pmatrix} \diamond W - \begin{pmatrix} U \\ V \end{pmatrix} \diamond W \diamond \begin{pmatrix} D_A \\ D_B \end{pmatrix}\right] \diamond \begin{pmatrix} D_A - D_B \\ D_B - D_A \end{pmatrix} = 0, \tag{5.10}$$

that is,

$$\left[\begin{pmatrix} A \\ B \end{pmatrix} \diamond \begin{pmatrix} \bar{U}_i \\ \bar{V}_i \end{pmatrix} - \begin{pmatrix} \bar{U}_i \\ \bar{V}_i \end{pmatrix} \diamond \begin{pmatrix} \theta_i^A \\ \theta_i^B \end{pmatrix}\right] \diamond \begin{pmatrix} \theta_i^A - \theta_i^B \\ \theta_i^B - \theta_i^A \end{pmatrix} = 0, \quad i = 1, \ldots, k \tag{5.11}$$

where $\bar{U}_i$ and $\bar{V}_i$ denotes the i^{th} column of $\bar{U} = UW$ and $\bar{V} = VW$ respectively. This implies that for all $i = 1, \ldots, k$ either $\theta_i^A = \theta_i^B$ or $\left(\theta_i^A, \bar{U}_i\right)$ and $\left(\theta_i^B, \bar{V}_i\right)$ are eigenpairs of respectively A and B.

Definition 5.2 *Let A and A_U be two square matrices respectively of order m and k, where $m \geq k$. A_U is said to be imbeddable in A if there exists a matrix $U \in V_{m,k}$ such that $U^*AU = A_U$.*

For Hermitian matrices, [FP57] gives us the following result.

Theorem 5.1 *Let A and A_U be Hermitian matrices and $\alpha_1 \leq \alpha_2 \leq \cdots \leq \alpha_m$ and $\theta_1^A \leq \theta_2^A \leq \cdots \leq \theta_k^A$ their respective eigenvalues. Then a necessary and sufficient condition for A_U to be imbeddable in A is that*

$$\theta_i^A \in [\alpha_i, \alpha_{i-k+m}], \quad i = 1, \ldots, k.$$

The necessity part of this theorem is well known as the *Cauchy interlacing theorem* [Par80, p. 202]. The sufficiency part is less easy to prove *cf.* [FP57, Th. 1][PS08, Th. 1 and 2].

Definition 5.3 *For S_1 and S_2, two non-empty subsets of a metric space together with the distance d, we define*

$$e_d(S_1, S_2) = \inf_{\substack{s_1 \in S_1 \\ s_2 \in S_2}} d(s_1, s_2)$$

When considering two non-empty subsets $[\alpha_1, \alpha_2]$ and $[\beta_1, \beta_2]$ of $\mathbb{R}$, one can easily see that

$$e_d([\alpha_1, \alpha_2], [\beta_1, \beta_2]) = \max(0, \alpha_1 - \beta_2, \beta_1 - \alpha_2).$$

Theorem 5.2 *Let $\alpha_1 \leq \alpha_2 \leq \cdots \leq \alpha_m$ and $\beta_1 \leq \beta_2 \leq \cdots \leq \beta_n$ be the eigenvalues of Hermitian matrices respectively A and B. The solution of Problem 5.1 is*

$$\sum_{i=1}^{k} \left(e_d([\alpha_i, \alpha_{i-k+m}], [\beta_i, \beta_{i-k+n}])\right)^2$$

with d the Euclidean norm.

Proof Recall that when A and B are Hermitian matrices, U^*AU and V^*BV are jointly diagonalizable by a unitary matrix at all stationary points (U, V). Since the distance is invariant by joint unitary transformation, the value of the minimum is not affected if we restrict (U, V) to be such that U^*AU and V^*BV are diagonal. Problem 5.1 reduces to minimize

$$f(U,V) = \sum_{i=1}^{k} \left|\theta_i^A - \theta_i^B\right|^2,$$

where $U^*AU = \operatorname{diag}(\theta_1^A, \ldots, \theta_k^A)$ and $V^*BV = \operatorname{diag}(\theta_1^B, \ldots, \theta_k^B)$. It follows from Theorem 5.1 that the minimum of Problem 5.1 is

$$\min_{\theta_i^A, \theta_i^B} \min_{\pi} \sum_{i=1}^{k} \left(\theta_{\pi(i)}^A - \theta_i^B\right)^2 \tag{5.12}$$

such that

$$\theta_1^A \leq \theta_2^A \leq \cdots \leq \theta_k^A, \quad \theta_1^B \leq \theta_2^B \leq \cdots \leq \theta_k^B, \tag{5.13}$$

$$\theta_i^A \in [\alpha_i, \alpha_{i-k+m}], \quad \theta_i^B \in [\beta_i, \beta_{i-k+n}], \tag{5.14}$$

and $\pi(\cdot)$ is a permutation of $1, \ldots, k$.

Let $\theta_1^A, \ldots, \theta_k^A$, and $\theta_1^B, \ldots, \theta_k^B$ satisfy (5.13). Then, the identity permutation $\pi(i) = i$ is optimal for problem (5.12). Indeed, if π is not the identity, then there exists i and j such that $i < j$ and $\pi(i) > \pi(j)$, and we have

$$\begin{aligned}
&\left(\theta_i^A - \theta_{\pi(i)}^B\right)^2 + \left(\theta_j^A - \theta_{\pi(j)}^B\right)^2 - \left[\left(\theta_j^A - \theta_{\pi(i)}^B\right)^2 + \left(\theta_i^A - \theta_{\pi(j)}^B\right)^2\right] \\
&\quad = 2\left(\theta_j^A - \theta_i^A\right)\left(\theta_{\pi(i)}^B - \theta_{\pi(j)}^B\right) \leq 0.
\end{aligned}$$

Since the identity permutation is optimal, our minimization problem simply reduces to

$$\sum_{i=1}^{k} \min_{(5.13)(5.14)} \left(\theta_i^A - \theta_i^B\right)^2.$$

We now show that (5.13) can be relaxed. Indeed, assume there is an optimal solution that does not satisfy the ordering condition, *i.e.* there

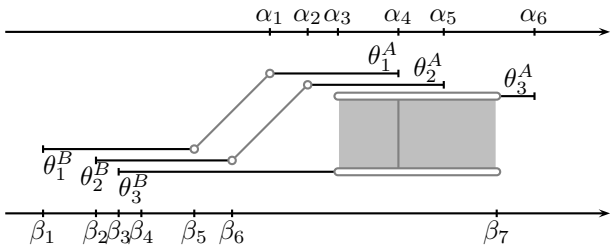


Figure 5.1: Let the α_i and β_i be the eigenvalues of the Hermitian matrices A and B, and $k = 3$. Problem 5.1 is then equivalent to $\sum_{i=1}^{3} \min_{\theta_i^A, \theta_i^B} \left(\theta_i^A - \theta_i^B\right)^2$ such that $\theta_i^A \in [\alpha_i, \alpha_{i+3}]$, $\theta_i^B \in [\beta_i, \beta_{i+4}]$. The two first terms of this sum have strictly positive contributions whereas the third one can be reduced to zero within a continuous set of values for θ_3^A and θ_3^B in $[\alpha_3, \beta_7]$.

exist i and j, $i < j$ such that $\theta_j^A \leq \theta_i^A$. One can see that the following inequalities hold

$$\alpha_i \leq \alpha_j \leq \theta_j^A \leq \theta_i^A \leq \alpha_{i-k+m} \leq \alpha_{j-k+m}.$$

Since θ_i^A belongs to $[\alpha_j, \alpha_{j-k+m}]$ and θ_j^A belongs to $[\alpha_i, \alpha_{i-k+m}]$, one can switch i and j and build an ordered solution that does not change the cost function and hence remains optimal.

It follows that $\sum_{i=1}^{k} \min_{(5.13)(5.14)} \left(\theta_i^A - \theta_i^B\right)^2$ is equal to $\sum_{i=1}^{k} \min_{(5.14)} \left(\theta_i^A - \theta_i^B\right)^2$. This result is precisely what we were looking for. □

Figure 5.1 gives an example of an optimal matching.

Analysis of Problem 5.2

For all $Y \in T_X \mathbb{C}^{n \times m} \simeq \mathbb{C}^{n \times m}$, we have

$$\mathrm{D}\hat{f}(X) \cdot Y = 2\Re \operatorname{tr}(Y^* \left(XAA^* - B^*XA - BXA^* + B^*BX\right)), \quad (5.15)$$

and hence the gradient of $\hat{f}$ at a point $X \in \mathbb{C}^{n\times m}$ is

$$\operatorname{grad} \hat{f}(X) = 2(XAA^* - B^*XA - BXA^* + B^*BX). \tag{5.16}$$

Since the normal space $T_X^\perp\mathcal{M}$ is the orthogonal complement of the tangent space $T_X\mathcal{M}$, one can, for any $X = VU^* \in \mathcal{M}$, decompose any $E \in \mathbb{C}^{n\times m}$ into its orthogonal projections on $T_X\mathcal{M}$ and $T_X^\perp\mathcal{M}$:

$$\begin{aligned} \mathrm{P}_X E &= E - V \operatorname{Her}(V^*EU)\, U^* - (I_n - VV^*)\, E\, (I_m - UU^*), \text{ and} \\ \mathrm{P}_X^\perp E &= V \operatorname{Her}(V^*EU)\, U^* + (I_n - VV^*)\, E\, (I_m - UU^*). \end{aligned} \tag{5.17}$$

For any $Y \in T_X\mathcal{M}$, (5.15) hence yields

$$\mathrm{D}\hat{f}(X) \cdot Y = \mathrm{D}f(X) \cdot Y = g_X\left(\mathrm{P}_X \operatorname{grad} \hat{f}(X), Y\right),$$

and the gradient of f at a point $X = VU^* \in \mathcal{M}$ is

$$\operatorname{grad} f(X) = \mathrm{P}_X \operatorname{grad} \hat{f}\,(X). \tag{5.18}$$

Analysis of Problem 5.3

This problem is very similar to Problem 5.2. We have

$$\mathrm{D}\hat{f}(X) \cdot Y = 2\Re \operatorname{tr}(Y^*\,(X - B^*XA - BXA^* + B^*BXA^*A)), \tag{5.19}$$

for all $Y \in T_X\mathbb{C}^{n\times m} \simeq \mathbb{C}^{n\times m}$, and hence the gradient of $\hat{f}$ at a point $X \in \mathbb{C}^{n\times m}$ is

$$\operatorname{grad} \hat{f}(X) = 2(X - B^*XA - BXA^* + B^*BXA^*A).$$

The feasible set is the same as in Problem 5.2. Hence the orthogonal decomposition (5.17) holds, and the gradient of f at a point $X = VU^* \in \mathcal{M}$ is

$$\operatorname{grad} f(X) = \mathrm{P}_X \operatorname{grad} \hat{f}\,(X).$$

Analysis of Problem 5.4

For all $T \in T_S\mathbb{C}^{n\times m} \simeq \mathbb{C}^{n\times m}$, we have

$$\mathrm{D}\hat{f}(S) \cdot T = 2\Re \operatorname{tr}(T^*\,(SAA^* - B^*SA - BSA^* + B^*BS)), \tag{5.20}$$

and hence the gradient of $\hat{f}$ at a point $S \in \mathbb{C}^{n\times m}$ is

$$\operatorname{grad} \hat{f}(S) = 2(SAA^* - B^*SA - BSA^* + B^*BS). \tag{5.21}$$

Since the normal space, $T_S^{\perp}\mathcal{M}$, is the orthogonal complement of the tangent space, $T_S\mathcal{M}$, one can, for any $S \in \mathcal{M}$, decompose any $E \in \mathbb{C}^{n\times m}$ into its orthogonal projections on $T_S\mathcal{M}$ and $T_S^{\perp}\mathcal{M}$:

$$\mathrm{P}_S E = E - S\ \Re\operatorname{tr}(S^*E) \quad \text{and} \quad \mathrm{P}_S^{\perp} E = S\ \Re\operatorname{tr}(S^*E). \tag{5.22}$$

For any $T \in T_S\mathcal{M}$, (5.20) then yields

$$\mathrm{D}\hat{f}(S)\cdot T = \mathrm{D}f(S)\cdot T = g_S\left(\mathrm{P}_S \operatorname{grad} \hat{f}(S), T\right),$$

and the gradient of f at a point $S \in \mathcal{M}$ is $\operatorname{grad} f(S) = \mathrm{P}_S \operatorname{grad} \hat{f}\,(S)$.

For our problem, (5.4) yields, by means of (5.21) and (5.22)

$$\lambda\, S = (SA - BS)A^* - B^*(SA - BS)$$

where $\lambda = \operatorname{tr}((SA - BS)^*(SA - BS)) \equiv \hat{f}(S)$. Its equivalent vectorized form is

$$\lambda \operatorname{vec}(S) = \left(A^T \otimes I - I \otimes B\right)^* \left(A^T \otimes I - I \otimes B\right) \operatorname{vec}(S).$$

Hence, the stationary points of Problem 5.4 are given by the eigenvectors of $\left(A^T \otimes I - I \otimes B\right)^* \left(A^T \otimes I - I \otimes B\right)$. The cost function f simply reduces to the corresponding eigenvalue and the minimal cost is then the smallest eigenvalue.

Analysis of Problem 5.5

This problem is very similar to Problem 5.4. A similar approach yields

$$\lambda\, S = (S - BSA^*) - B^*(S - BSA^*)A$$

where $\lambda = \operatorname{tr}((S - BSA^*)^*(S - BSA^*))$. Its equivalent vectorized form is

$$\lambda \operatorname{vec}(S) = \left(I \otimes I - \bar{A} \otimes B\right)^* \left(I \otimes I - \bar{A} \otimes B\right) \operatorname{vec}(S),$$

where $\bar{A}$ denotes the complex conjugate of A.

Hence, the stationary points of Problem 5.5 are given by the eigenvectors of $\left(I \otimes I - \bar{A} \otimes B\right)^* \left(I \otimes I - \bar{A} \otimes B\right)$, and the cost function f again simply reduces to the corresponding eigenvalue and the minimal cost is then the smallest eigenvalue.

5.5 Iterative Methods and Numerical Experiments

The solutions of the problems mentioned above may not have a closed form expression or may be very expensive to compute. Iterative optimization methods build a sequence of iterates that hopefully converges fast enough towards the optimal solution. We further implemented several optimization algorithms defined on manifolds (see [AMS08] for details) and applied them to Problem 5.1.

The complexity of classical algorithms can significantly increase according to the number of variables to deal with. An interesting approach when one works on a feasible set that has a manifold structure amounts to use classical methods defined on Euclidean spaces and apply them to the tangent space to the manifold (see [AMS08]). The link between the tangent space and the manifold is naturally done using specific geometric tools such as the so called *retraction* and *vector transport*. Let M be a point of a manifold $\mathcal{M}$, a retraction R_M is a smooth function that maps the tangent vector $\xi_M \in T_M\mathcal{M}$ to a point on $\mathcal{M}$ such that

- 0_M (the zero element of $T_M\mathcal{M}$) is mapped onto M, and
- there is no *distortion* around the origin, which means that

$$DR_M(0_M) = \mathrm{id}_{T_M\mathcal{M}}$$

  where $\mathrm{id}_{T_M\mathcal{M}}$ denotes the identity mapping on $T_M\mathcal{M}$.

Let now η_M be a tangent vector to $\mathcal{M}$ at M. A vector transport $\mathcal{T}_{\eta_M}$ is a smooth mapping defined as follows

$$\mathcal{T}_{\eta_M} : T_M\mathcal{M} \mapsto T_{R_M\eta_M}\mathcal{M} : \xi_M \to \mathcal{T}_{\eta_M}(\xi_M)$$

and satisfying the following properties:

- $\mathcal{T}_{0_M}\xi_M = \xi_M$ for all $\xi_M \in T_M\mathcal{M}$, and
- $\mathcal{T}_{\eta_M}(a\xi_M + b\zeta_M) = a\mathcal{T}_{\eta_M}(\xi_M) + b\mathcal{T}_{\eta_M}(\zeta_M)$ for all $\xi_M, \zeta_M \in T_M\mathcal{M}$, $a, b, \in \mathbb{R}$.

For solving Problem 5.1, we choose the following retraction and vector transport:

$$R_{(U,V)}(\xi_U, \xi_V) := (\mathrm{qf}(U + \xi_U), \mathrm{qf}(V + \xi_V))$$

where $\mathrm{qf} : \mathbb{R}^{m\times k} \to \mathbb{R}^{m\times k} : M \mapsto \mathrm{qf}(M)$ denotes the Q factor of the QR decomposition of the argument, and

$$\mathcal{T}_{\xi_M}(\zeta_M) := P_{R_M \xi_M} \hat{\zeta}_M$$

where $\hat{\zeta}_M$ is the same vector as ζ_M see as a vector in the embedding space rather than a vector in $T_M\mathcal{M}$.

We first implemented a steepest descent method based on the line search algorithm described in [AMS08, Algorithm 1]. Steepest descent consists in choosing the opposite of the gradient as search direction. In our numerical experiments, we further set the step size in [AMS08, Algorithm 1] using the so-called *Armijo* scheme whose parameters are arbitrarily set to $\beta = 0.5$ and $\sigma = 0.8$.

We then implemented a conjugate gradient method based on [AMS08, Algorithm 13] with the Fletcher-Reeves scheme. In our numerical experiments, we further set the step size in [AMS08, Algorithm 13] using the so-called *Armijo* scheme whose parameters are arbitrarily set to $\beta = 0.5$ and $\sigma = 0.8$.

We further implemented the Newton method as described in [AMS08, Algorithm 5]. This method uses higher-order derivatives of f. Note that, on a manifold, the concept of directional derivative is generalized by the so-called affine connection [AMS08, Section 5.2]. Notice also that this Newton method is a plain Newton method without globalization enhancement, *i.e.* without modifications that would guarantee convergence to critical points [DS96, chap. 6].

We then enhanced the Newton method with a trust region scheme as described in [AMS08, Algorithm 10]. Note that, in our numerical experiments, we solve the trust-region sub-problem as explained in [AMS08, Proposition 7.3.1].

Finally, we compared our methods with two classical optimization methods, the sequential quadratic programming method and the interior point method, which are implemented in the Optimization toolbox of Matlab.

Figure 5.5 shows,

- for geometric methods, the norm of the gradient of the cost function of Problem 5.1, and
- for classical methods, the norm of the gradient of the Lagrangian of Problem 5.1

at each iterate of the different Numerical methods. Figure 5.5 shows the difference between the value of the cost function of Problem 5.1 and its optimal value at each iterate of the different Numerical methods.

We observe that the rate of convergence of the steepest descent method and of the conjugate gradient are linear as the ones of the classical optimization methods whereas the Newton methods (with and without trust region) present a quadratic rate of convergence. However, as explained in [AMS08, Chapter 7], the Newton method without trust region does not distinguish among local minima, saddle points, and local maxima. Moreover, there exist examples for which the Newton method does not converge. The Newton method with trust region remedies this undesirable behaviour while retaining the super-linear local convergence properties of the Newton method.

One can finally notice that the rate of convergence of geometric optimization methods is higher than the ones of classical methods while the steepest descent method, the conjugate gradient method and the classical methods did not used derivatives of order higher than 1.

5.6 Relation to the Crawford Number

The field of values of a square matrix A is defined as the set of complex numbers

$$\mathcal{F}(A) := \{x^*Ax : x^*x = 1\},$$

and is known to be a closed convex set [HJ91]. The Crawford number is defined as the distance from that compact set to the origin

$$\mathrm{Cr}(A) := \min\{|\lambda| : \lambda \in \mathcal{F}(A)\},$$

and can be computed *e.g.* with techniques described in [HJ91]. One could define the *generalized Crawford number* of two matrices A and B

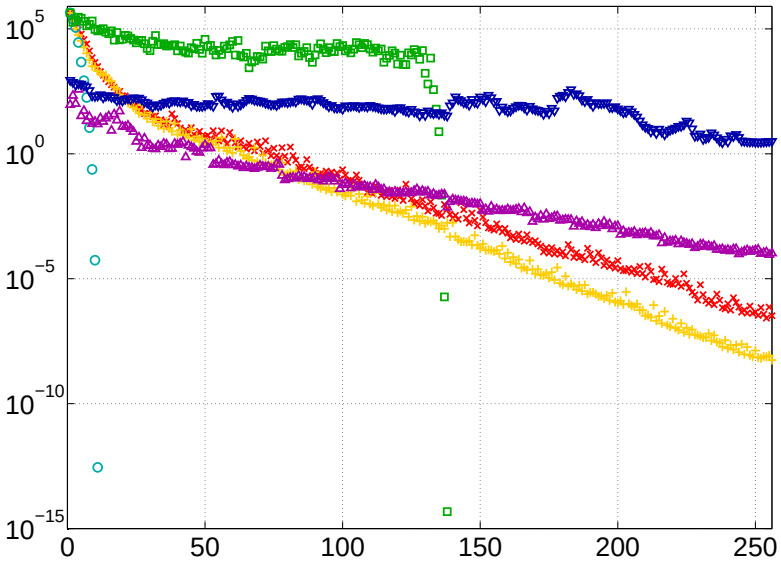
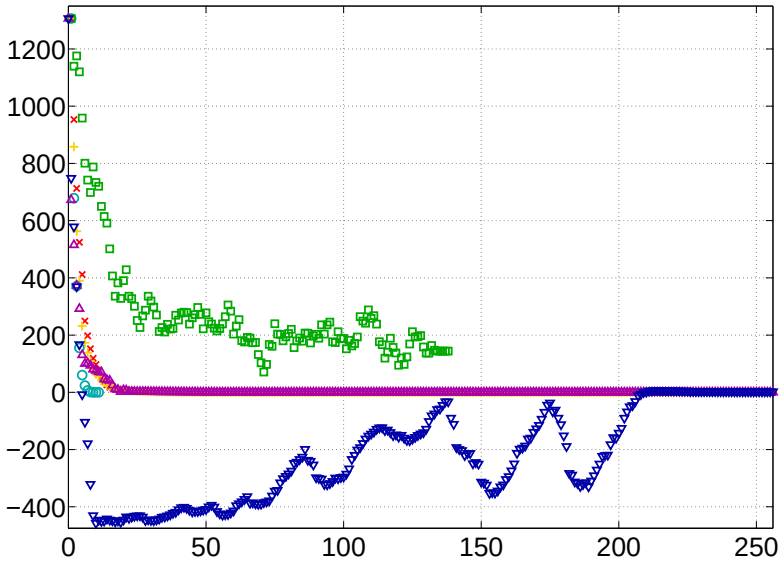


Figure 5.2: This figure shows in abscissas the number of iterations and in ordinate the norm of the gradient of the cost function of Problem 5.1 for geometric methods and the norm of the gradient of the Lagrangian of Problem 5.1 for classical methods at each iterate of the different Numerical methods. The matrices A and B are defined as follows $A = P^T \mathrm{diag}(1, 2, \cdots, 16)P$ and $B = Q^T \mathrm{diag}(17, 18, \cdots, 29)Q$, with P and Q, two random orthogonal matrices.



Geometric optimization methods:

Steepest Descent	*Newton without Trust Region*
Conjugate Gradient	*Newton with Trust Region*

Classical optimization methods:

SQP (`fmincon`)	*Interior Point (`fmincon`)*

Figure 5.3: This figure shows in abscissas the number of iterations and in ordinate the difference between the value of the cost function of Problem 5.1 and its optimal value (512) at each iterate of the different Numerical methods. The matrices A and B are defined as follows $A = P^T \mathrm{diag}(1, 2, \cdots, 16)P$ and $B = Q^T \mathrm{diag}(17, 18, \cdots, 29)Q$, with P and Q, two random orthogonal matrices.

as the distance between $\mathcal{F}(A)$ and $\mathcal{F}(B)$, *i.e.*

$$\mathrm{Cr}(A,B) := \min\left\{|\lambda-\mu| : \lambda\in\mathcal{F}(A), \mu\in\mathcal{F}(B)\right\}.$$

Clearly, $\mathrm{Cr}(A,0) = \mathrm{Cr}(A)$ which thus generalizes the concept. Moreover this is a special case of our problem since

$$\mathrm{Cr}(A,B) = \min_{U^*U=V^*V=1} \|U^*AU - V^*BV\|_F .$$

One can say that Problem 5.1 is a k-dimensional extension of this problem.

Chapter 6

Low Rank Approximation

In this chapter, we analyze an algorithm to compute a low-rank approximation of the similarity matrix S^{Blondel} introduced by Blondel *et al.* in [BGH$^+$04]. This problem can be reformulated as an optimization problem of a continuous function $\Phi(S) = \text{tr}\big(S^T \mathcal{M}^2(S)\big)$ where S is constrained to have unit Frobenius norm, and $\mathcal{M}^2$ is a non-negative linear map. We restrict the feasible set to the set of matrices of unit Frobenius norm with either k nonzero identical singular values or at most k nonzero (not necessarily identical) singular values. We first characterize the stationary points of the associated optimization problems and further consider iterative algorithms to find one of them. We analyze the convergence properties of our algorithm and prove that accumulation points are stationary points of $\Phi(S)$. We finally compare our method in terms of speed and accuracy to the full rank algorithm proposed in [BGH$^+$04].

6.1 Introduction

Node-to-node similarity measures compare the nodes of a graph G_A with the nodes of another graph G_B according to some similarity criterion, and have been applied to many practical problems such as comparing chemical structures [Bal85], navigating in complex networks like the World Wide Web [Kle99], and analyzing different kinds of biological data [HS03].

These node-to-node similarity measures are conveniently stored in the so-called similarity matrix, S^{Blondel}, whose (i, j) entry tells how the node i is similar to the node j. In [BGH+04], Blondel *et al.* define a node-to-node similarity measure as a fixed point of an iterative process, and prove that their measure is equivalent to the solution of an eigenvalue problem of a dimension that is the product of the number of nodes in both graphs. For large graphs, computing this similarity measure can hence be quite expensive. In [FNVD08], Fraikin *et al.* approach the similarity matrix defined by Blondel *et al.* by a rank-k matrix with k identical singular values. Note that if one of the two graphs to compare is undirected, the similarity matrix, S^{Blondel}, is a matrix of rank 1 [BGH+04, Theorem 10] which can be exactly approximated with $k = 1$. We propose to reduce the computational cost of the Blondel *et al.* similarity by using a low-rank iterative scheme that experimentally converges towards their approximation.

In this chapter, we propose two low-rank iterative schemes that converge towards two approximations of the Blondel *et al.* similarity matrix with respectively either k nonzero identical singular values or at most k nonzero (not necessarily identical) singular values, and further analyze the convergence properties of our algorithms.

This chapter is organized as follows. Section 6.2 introduces the notations further used in the article. Section 6.3 recalls the similarity matrix defined by Blondel *et al.* Section 6.4 shows that it is the solution of an optimization problem. Sections 6.5 and 6.6 analyze different low-rank approximations of this optimization problem. Section 6.7 analyzes the complexity of our algorithms and section 6.8 presents experimental results. And finally, section 6.9 gives our conclusions.

6.2 Notations

Throughout, G_A and G_B stand for graphs with respectively m and n nodes. These graphs are conveniently represented by $A \in \mathbb{R}^{m \times m}$ and $B \in \mathbb{R}^{n \times n}$, their respective adjacency matrices. And $C_A(i)$ and $P_A(i)$ denote respectively the set of children and parents of node i in G_A.

We further consider the following sets of matrices of Frobenius

norm 1:

$$\mathcal{S}(m,n) := \mathrm{Norm}(1,m,n) = \{S \in \mathbb{R}^{m\times n} : \|S\|_F = 1\} \ ,$$

$$\mathcal{S}_k(m,n) := \left\{ U\hat{I}_k V^T \in \mathbb{R}^{m\times n} : \begin{array}{l} U \in V_{k,m},\ V \in V_{k,n}, \\ \hat{I}_k = I_k / \|I_k\|_F = I_k/\sqrt{k} \end{array} \right\} \text{, and}$$

$$\mathcal{S}_{\leq k}(m,n) := \left\{ UDV^T \in \mathbb{R}^{m\times n} : \begin{array}{l} U \in V_{k,m},\ V \in V_{k,n}, \\ D \text{ diagonal, } \|D\|_F = 1 \end{array} \right\} .$$

That is, $\mathcal{S}_k(m,n)$ is the set of all $m \times n$ matrices with unit Frobenius norm and k nonzero equal singular values, and $\mathcal{S}_{\leq k}(m,n)$ is the set of all $m \times n$ matrices with unit Frobenius norm and rank less than or equal to k. $\mathrm{Diag}(k,m,n)$ denotes a set of diagonal matrices defined as follows

$$\mathrm{Diag}(k,m,n) := \left\{ D \in \mathbb{R}^{m\times n} : \ D \text{ diagonal, and } D_{ii} = 0 \text{ for all } i > k \right\}.$$

$\mathcal{Skew}(k)$ denotes the set of skew-symmetric matrices of order k. $\mathcal{Sym}(k)$ denotes the set of symmetric matrices of order k. $\mathbf{1}$ is the matrix whose entries are all equal to 1.

Let $\mathcal{H}$ be a Hilbert space and $\mathcal{S}$, a non empty algebraic subset of $\mathcal{H}$. A vector $\xi \in \mathcal{H}$ is an *analytic admissible direction* for $\mathcal{S}$ at $S \in \mathcal{S}$ if there exists an analytic curve $\gamma(t) : \mathbb{R} \mapsto \mathcal{H}$ with $\gamma(0) = S$, and $\gamma(t) \in \mathcal{S}$, for all $t \geq 0$, such that

$$\lim_{t\to 0} \frac{\gamma(t) - \gamma(0)}{t} = \xi.$$

According to [OW04, Proposition 2], the *contingent cone* (see, e.g., [Aub00]) to $\mathcal{S}$ at S, denoted $C_S\mathcal{S}$, is equal to the set of all analytic admissible directions for $\mathcal{S}$ at S, *i.e.*

$$C_S\mathcal{S} = \left\{ \dot{\gamma}(0) : \begin{array}{l} \gamma \text{ is an analytic curve with } \gamma(0) = S, \\ \text{and } \gamma(t) \in \mathcal{S}, \text{ for all } t \geq 0. \end{array} \right\} .$$

The normal cone to $\mathcal{S}$ at S, denoted $N_S\mathcal{S}$, is defined as

$$N_S\mathcal{S} := \{\zeta \in \mathcal{H} : \langle \zeta, \xi \rangle_F \leq 0, \forall \xi \in C_S\mathcal{S}\} \ .$$

6.3 The Similarity Matrix

Node-to-node similarity measures compare the nodes of a graph G_A with the nodes of another graph G_B according to some similarity criterion.

In [BGH+04], Blondel *et al.* introduce a recursive requirement which states that the similarity between node i and node j should be large if the similarity between the neighbors of node i and the neighbors of node j is large. More specifically, they define a similarity measure by means of Algorithm 3.

Algorithm 3

Require: $G(A)$ and $G(B)$, two graphs respectively of order m and n.

$S^0 \leftarrow \mathbf{1}/\left\|\mathbf{1}\right\|_F \in \mathbb{R}^{m\times n}$

for $t = 1, 2, \cdots, t_{\max}$ **do**

$$S^t \leftarrow \frac{\mathcal{M}(S^{t-1})}{\left\|\mathcal{M}(S^{t-1})\right\|_F}, \text{ with } [\mathcal{M}(S)]_{ij} := \sum_{\substack{k\in C_A(i)\\ l\in C_B(j)}} S_{kl} + \sum_{\substack{k\in P_A(i)\\ l\in P_B(j)}} S_{kl}.$$

end for

$S^{\text{Blondel}} \leftarrow S^t$

where $t_{\max}$ is an even number that is "sufficiently large".

In this algorithm, they first initialize all similarity scores to the same value, and further update them in the *reinforcement loop*, that can be justified as follows:

- $[\mathcal{M}(S^{t-1})]_{ij} : S^t_{ij}$, the similarity score between node i and node j at step t, is the sum of all (k, l) entries of S^{t-1} such that node k is a child of node i in G_A and node l is a child of node j in G_B, plus the sum of all (k, l) entries of S^{t-1} such that node k is a parent of node i in G_A and node l is a parent of node j in G_B. Doing so, the similarity score between two nodes increases if they have many highly similar children or parents.
- $\mathcal{M}(S^{t-1})/\left\|\mathcal{M}(S^{t-1})\right\|_F$: since they are not interested in the absolute value of S^t_{ij} but only in the relative score of two different pairs, they normalize the whole similarity matrix S^t to avoid over- or under-flow.

One can rewrite $\mathcal{M}(S)$ in terms of matrix operations over S, *i.e.*

$$[\mathcal{M}(S)]_{ij} = \sum_{k,l} A_{ik} S_{kl} B_{jl} + \sum_{k,l} A_{ki} S_{kl} B_{lj} = \left[ASB^T + A^T SB\right]_{ij}. \quad (6.1)$$

Since $\mathrm{vec}\big(ASB^T\big) = (B \otimes A)\,\mathrm{vec}(S)$, equation (6.1) can be rewritten in its so-called vector form:

$$\mathrm{vec}(\mathcal{M}(S)) = M\ \mathrm{vec}(S) := \big(B \otimes A + B^T \otimes A^T\big)\ \mathrm{vec}(S).$$

In [BGH+04], Blondel *et al.* show that Algorithm 3 is in fact the power method applied to the matrix M. This matrix is non-negative and hence, according to Perron-Frobenius Theorem, there exists a real positive eigenvalue ρ, called the Perron root, such that any other eigenvalue λ satisfies $|\lambda| \leq \rho$. Since M is symmetric, its eigenvalues are real and hence M can have at most two extremal eigenvalues (*i.e.* of maximum modulus), ρ and possibly $-\rho$. As a direct consequence, M^2 has only one extremal eigenvalue, namely ρ^2 (but possibly of multiplicity higher than one), and the even iterates of the reinforcement loop in Algorithm 3 converge towards $S^{2\infty}$, the normalized orthogonal projection of S^0 onto

$$E_{\rho^2}(\mathcal{M}^2) := \big\{S \quad \text{s.t.} \quad \rho^2 S = \mathcal{M}^2(S)\big\}\ ,$$

the eigenspace of $\mathcal{M}^2$ associated to ρ^2, with respect to the Frobenius inner product (see [BGH+04] for details about the proof of convergence). Notice that, since $S^{2\infty}$ is a fixed point of the even iterates of the reinforcement loop in Algorithm 3, one can write

$$\rho^2 S^{2\infty} = \mathcal{M}^2(S^{2\infty})\ .$$

6.4 From Similarity to Optimization

In this section, we show that the similarity matrix defined by Blondel *et al.* is the solution of an optimization problem.

One can first observe that the iteration in Algorithm 3 is such that

$$S^t \in \underset{\|S\|_F=1}{\mathrm{argmax}} \big\langle S, \mathcal{M}(S^{t-1})\big\rangle_F\ .$$

This result is easy to prove using to the Cauchy-Schwarz inequality

$$\begin{aligned}\big\langle S^t, \mathcal{M}(S^{t-1})\big\rangle_F &= \left\langle \frac{\mathcal{M}(S^{t-1})}{\|\mathcal{M}(S^{t-1})\|_F}, \mathcal{M}(S^{t-1})\right\rangle_F = \big\|\mathcal{M}(S^{t-1})\big\|_F \\ &= \|S\|_F \big\|\mathcal{M}(S^{t-1})\big\|_F \overset{CS}{\geq} \big\langle S, \mathcal{M}(S^{t-1})\big\rangle_F\ , \qquad \text{since } \|S\|_F = 1.\end{aligned}$$

Moreover, one can prove that $S^{2\infty}$ is a solution of

$$\operatorname*{argmax}_{S\in\mathcal{S}(m,n)} \Phi(S) \ , \quad \text{where} \quad \Phi(S) := \big\langle S, \mathcal{M}^2(S)\big\rangle_F = \operatorname{tr}\big(S^T\, \mathcal{M}^2(S)\big) \ , \tag{6.2}$$

and $\mathcal{M}^2(S) = \mathcal{M}(\mathcal{M}(S))$ is defined in equation (6.1). Indeed, since the Perron root is equal to the spectral radius, *i.e.*

$$\rho = \max_{\|S\|_F=1} \|\mathcal{M}(S)\|_F \ , \tag{6.3}$$

and the map $\mathcal{M}(\cdot)$ is self-adjoint (*i.e.* $\langle S_1, \mathcal{M}(S_2)\rangle_F = \langle \mathcal{M}(S_1), S_2\rangle_F$), we have

$$\max_{\|S\|_F=1} \big\langle S, \mathcal{M}^2(S)\big\rangle_F = \max_{\|S\|_F=1} \|\mathcal{M}(S)\|_F^2 = \rho^2 = \big\langle S^{2\infty}, \mathcal{M}^2\left(S^{2\infty}\right)\big\rangle_F \ . \tag{6.4}$$

The problem (6.2) maximizes a continuous function Φ on a compact domain. Hence, according to first order optimality conditions, if $S^{2\infty}$ is a maximizer of (6.2) then $S^{2\infty}$ is a stationary point of (6.2). The concept of stationary point in the context of (6.2) is recalled in Definition 6.1 below.

The *gradient* of a differentiable function Φ at a point S, denoted $\operatorname{grad}\Phi(S)$, is defined as the unique vector that satisfies

$$\langle \xi, \operatorname{grad}\Phi(S)\rangle_F = \mathrm{D}\Phi(S)[\xi] \ , \tag{6.5}$$

for all admissible directions ξ.

Definition 6.1 *A point $S \in \mathcal{S}(m,n)$ is a* stationary point *of* (6.2) *if* $\operatorname{grad}\Phi(S)$ *belongs to $N_S\mathcal{S}(m,n)$, the normal cone to $\mathcal{S}(m,n)$ at S,* i.e.

$$\operatorname{grad}\Phi(S) \in N_S\mathcal{S}(m,n) := \{\zeta : \langle \zeta, \xi\rangle_F \le 0, \forall \xi \in C_S\mathcal{S}(m,n)\} \ . \tag{6.6}$$

For example, the tangent cone to $\mathcal{S}(m,n) = \{S : \|S\|_F = 1\}$ at a point S is given by

$$C_S\mathcal{S}(m,n) = \{\xi : \langle \xi, S\rangle_F = 0\} \ ,$$

and the normal cone to $\mathcal{S}(m,n)$ is

$$N_S\mathcal{S}(m,n) := \{\zeta : \langle \zeta, \xi\rangle_F \le 0, \forall \xi \in C_S\mathcal{S}(m,n)\} = \{\alpha S : \alpha \in \mathbb{R}\} \ .$$

Since the linear map $\mathcal{M}$ is self-adjoint (*i.e.* $\langle S_1, \mathcal{M}(S_2)\rangle_F = \langle \mathcal{M}(S_1), S_2\rangle_F$) one can write

$$\mathrm{D}\Phi(S)[\xi] = \left\langle \xi, \mathcal{M}^2(S)\right\rangle_F + \left\langle S, \mathcal{M}^2(\xi)\right\rangle_F = \left\langle \xi, 2\mathcal{M}^2(S)\right\rangle_F \ , \tag{6.7}$$

and the gradient of Φ at a point S is then $2\mathcal{M}^2(S)$. And clearly, one can observe that

$$\operatorname{grad}\Phi(S^{2\infty}) = 2\mathcal{M}^2(S^{2\infty}) = 2\rho^2 S^{2\infty} \in N_{S^{2\infty}}\mathcal{S}(m,n) = \left\{\alpha S^{2\infty} : \alpha \in \mathbb{R}\right\} \ , \tag{6.8}$$

and it follows directly that $S^{2\infty}$ is a stationary point of (6.2).

When S is large, Algorithm 3 becomes relatively expensive in terms of computational cost. Hence one can think of modifying the problem in order to find an approximation of S at lower cost. This chapter considers two kinds of low-rank approximations of the similarity matrix S, either by matrices of norm 1 with k nonzero identical singular values or by matrices of norm 1 with at most k nonzero (not necessarily identical) singular values. We first characterize the stationary points of the associated optimization problems and further consider iterative algorithms to find one of them.

6.5 Approximation with Identical Singular Values

We first consider the following approximations for the feasible set of (6.2).

Problem 6.1 *Solve* (6.2) *with* $\mathcal{S}(m,n)$ *replaced by* $\mathcal{S}_k(m,n)$*, the set of rank* k *matrices of norm* 1 *with* k *identical singular values,* i.e.

$$\mathcal{S}_k(m,n) := \left\{ U\hat{I}_k V^T \in \mathbb{R}^{m\times n} : \begin{array}{l} U \in V_{k,m},\ V \in V_{k,n}, \\ \hat{I}_k = I_k/\left\|I_k\right\|_F = I_k/\sqrt{k} \end{array} \right\} \ . \tag{6.9}$$

Note that if the extremal eigenvalues (*i.e.* eigenvalues of maximum modulus) of $\mathcal{M}$ are positive, then Problem 6.1 is equivalent to the problem considered in [FNVD08].

6.5.1 The Feasible Set of Problem 6.1 and its Stationary Points

Problem 6.1 is defined on a feasible set that has a *manifold* structure [CAVD11]. In this case, the tangent cone is called the tangent space since it admits a structure of vector space (see [AMS08, sec. 3.5] for details). The tangent space to the Stiefel manifold $V_{k,m}$ at a point U is given by

$$C_U V_{k,m} := \left\{\xi : \xi^T U + U^T \xi = 0_k\right\} \ ;$$

see, e.g., [AMS08, sec. 3.5.7]. Equivalently,

$$C_U V_{k,m} = \left\{U\Omega + U_\perp K : \Omega \in \mathcal{Skew}(k),\ K \in \mathbb{R}^{m-k\times k}\right\} \ . \tag{6.10}$$

where $U_\perp$ denotes any matrix such that $\begin{bmatrix} U & U_\perp \end{bmatrix}^T \begin{bmatrix} U & U_\perp \end{bmatrix} = I_m$. For Problem 6.1, it follows that the tangent space to $\mathcal{S}_k(m,n)$ at a point $S = U\hat{I}_k V^T \in \mathcal{S}_k(m,n)$ is given by

$$\begin{aligned} C_S\mathcal{S}_k(m,n) &:= \{\dot\gamma(0) : \gamma \text{ curve on } \mathcal{S}_k(m,n) \text{ with } \gamma(0) = S\} \\ &= \left\{ \begin{array}{l} U\Omega V^T + UK_V^T V_\perp^T + U_\perp K_U V^T \text{ s.t.} \\ \qquad \Omega \in \mathcal{Skew}(k),\ K_U \in \mathbb{R}^{m-k\times k},\ K_V \in \mathbb{R}^{n-k\times k} \end{array} \right\} . \end{aligned}$$

And subsequently, the normal space to $\mathcal{S}_k(m,n)$ at a point $S = U\hat{I}_k V^T \in \mathcal{S}_k(m,n)$ is given by

$$\begin{aligned} N_S\mathcal{S}_k(m,n) &:= \{\zeta : \langle \zeta, \xi\rangle_F \leq 0, \forall \xi \in C_S\mathcal{S}_k(m,n)\} \\ &= \left\{UHV^T + U_\perp K V_\perp^T \text{ s.t. } H \in \mathcal{Sym}(k),\ K \in \mathbb{R}^{m-k\times n-k}\right\}. \end{aligned}$$

As mentioned in Definition 6.1, a matrix S is defined to be a stationary point of (6.2) if $\operatorname{grad}\Phi(S)$ $(= 2\mathcal{M}^2(S))$ belongs to the normal cone to the feasible set at S. Hence, $S = U\hat{I}_k V^T \in \mathcal{S}_k(m,n)$ is a stationary point of Problem 6.1 if and only if

$$\mathcal{M}^2(S) \in N_S\mathcal{S}_k(m,n) = \left\{ \begin{array}{l} UHV^T + U_\perp K V_\perp^T \text{ s.t.} \\ \qquad H \in \mathcal{Sym}(k),\ K \in \mathbb{R}^{m-k\times n-k} \end{array} \right\} . \tag{6.11}$$

6.5.2 Algorithm for Problem 6.1 and its Convergence Analysis

We now propose Algorithm 4 and further prove that it converges towards stationary points of Problem 6.1 (see Theorem 6.5).

Algorithm 4

Require: $G(A)$ and $G(B)$, two graphs respectively of order m and n.

$S^0 \leftarrow \mathbf{1}/\left\|\mathbf{1}\right\|_F$

for $t = 1, 2, \cdots, t_{\max}$ **do**

Compute $S^t \in \mathcal{S}_k(m,n)$ according to

$$S^t \ (= U^t \hat{I}_k [V^t]^T) \ \leftarrow f(S^{t-1}) := \underset{\tilde{S} \in \mathcal{S}_k(m,n)}{\operatorname{argmax}} \left\langle \tilde{S}, \mathcal{M}^2(S^{t-1}) \right\rangle_F . \tag{6.12}$$

end for

$S^{\text{Blondel}} \leftarrow S^t$

where $\mathcal{M}^2(S) = \mathcal{M}(\mathcal{M}(S))$ is defined in equation (6.1).

Notice that Algorithm 4 is a particular case of [JNRS10, Algorithm 1] in which the convex function, $f(x)$, and the compact set, $\mathcal{Q}$, are respectively $\Phi(S)$ and $\mathcal{S}_k(m,n)$.

Let $\mathcal{M}^2(S^{t-1})$ have an ordered singular value decomposition (SVD)

$$\mathcal{M}^2(S^{t-1}) = \begin{bmatrix} P_1 & P_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix} , \tag{6.13}$$

with $P_1 \in \mathbb{R}^{m \times k}$, $P_2 \in \mathbb{R}^{m \times (m-k)}$, $Q_1 \in \mathbb{R}^{n \times k}$, $Q_2 \in \mathbb{R}^{n \times (n-k)}$, $\Sigma_1 \in \mathbb{R}^{k \times k}$ and $\Sigma_2 \in \mathbb{R}^{(m-k) \times (n-k)}$. Let $\nu(S^{t-1})$ denote the singular value gap,

$$\nu(S^{t-1}) := \sigma_{\min}(\Sigma_1) - \sigma_{\max}(\Sigma_2) = \sigma_k(\mathcal{M}^2(S^{t-1})) - \sigma_{k+1}(\mathcal{M}^2(S^{t-1})). \tag{6.14}$$

When $\nu(S^{t-1}) \neq 0$ (i.e., $\nu(S^{t-1}) > 0$), the next iterate S^t is uniquely defined by (6.12) and is equal to $P_1 \hat{I}_k Q_1^T$, as we will show in Lemma 6.1 below. When $\nu(S^{t-1}) = 0$, however, S^t is no longer

uniquely defined by (6.12); in this case, S^t is chosen arbitrarily in $\underset{\tilde{S}\in\mathcal{S}_k(m,n)}{\operatorname{argmax}} \left\langle \tilde{S}, \mathcal{M}^2(S^{t-1}) \right\rangle_F$. In practice, in our numerical experiments, we systematically choose $S^t := P_1 \hat{I}_k Q_1^T$, where P_1 and Q_1 are returned by the SVD function.

We first state a few intermediate results in order to prove convergence of Algorithm 4 to a stationary point of Problem 6.1.

Lemma 6.1 *Let $M \in \mathbb{R}^{m\times n}$ and its ordered singular value decomposition*

$$M = \begin{bmatrix} P_1 & P_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix} = P\Sigma Q^T \tag{6.15}$$

with $P_1 \in \mathbb{R}^{m\times k}$, $P_2 \in \mathbb{R}^{m\times(m-k)}$, $Q_1 \in \mathbb{R}^{n\times k}$, $Q_2 \in \mathbb{R}^{n\times(n-k)}$, $\Sigma_1 \in \mathbb{R}^{k\times k}$ and $\Sigma_2 \in \mathbb{R}^{(m-k)\times(n-k)}$. Then

$$\max_{S=U\hat{I}_k V^T \in \mathcal{S}_k(m,n)} \langle S, M \rangle_F = \operatorname{tr}\Big(\hat{I}_k \Sigma_1\Big). \tag{6.16}$$

Moreover, if $\nu := \sigma_{\min}(\Sigma_1) - \sigma_{\max}(\Sigma_2) > 0$, then the maximizing solution S is unique and equals $P_1 \hat{I}_k Q_1^T$.

Proof: We have

$$\operatorname{tr}\Big(\hat{I}_k U^T M V\Big) \le \sum_{i=1}^{k} \sigma_i(\hat{I}_k U^T M V) \le \sum_{i=1}^{k} \sigma_i(\hat{I}_k)\ \sigma_i(\Sigma_1) \le \operatorname{tr}\Big(\hat{I}_k \Sigma_1\Big) \tag{6.17}$$

according to [HJ91, Formulas 3.1.10b, and Lemma 3.3.1]. The upper bound is reached for $U = P_1$, and $V = Q_1$. The uniqueness of $P_1 \hat{I}_k Q_1^T$ is a known result discussed, *e.g.*, in [HJ91, Theorem 3.1.1 and 3.1.1'] □

Theorem 6.2 *Let $\mathcal{M}^2(S)$ have an ordered singular value decomposition*

$$\mathcal{M}^2(S) = \begin{bmatrix} P_1 & P_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix} = P\Sigma Q^T\ , \tag{6.18}$$

with $P_1 \in \mathbb{R}^{m\times k}$, $P_2 \in \mathbb{R}^{m\times(m-k)}$, $Q_1 \in \mathbb{R}^{n\times k}$, $Q_2 \in \mathbb{R}^{n\times(n-k)}$, $\Sigma_1 \in \mathbb{R}^{k\times k}$ and $\Sigma_2 \in \mathbb{R}^{(m-k)\times(n-k)}$, and $S^+ := f(S)$, with f the function defined in Algorithm 4. Then

$$\Phi(S^+) - \Phi(S) \ge \big\|\mathcal{M}(S^+ - S)\big\|_F^2 + \nu\ \sqrt{k}\ \big\|S^+ - S\big\|_F^2 \tag{6.19}$$

with Φ the function defined in equation (6.2), and $\nu = \sigma_{\min}(\Sigma_1) - \sigma_{\max}(\Sigma_2)$. In particular, Algorithm 4 is an ascent iteration for Φ.

Proof. Adding and subtracting S to S^+ and using the self-adjointness of $\mathcal{M}$ yields

$$\Phi(S^+ - S + S) = \left\|\mathcal{M}(S^+ - S)\right\|_F^2 + 2\left\langle S^+ - S, \mathcal{M}^2(S)\right\rangle_F + \Phi(S) \ . \quad (6.20)$$

According to Lemma 6.1, $S^+ = P_1 \hat{I}_k Q_1^T$. Hence, using the ordered singular value decomposition of $\mathcal{M}^2(S)$, the second term of the right-hand side of equation (6.20) becomes

$$\left\langle S^+ - S, \mathcal{M}^2(S)\right\rangle_F = \mathrm{tr}\Big(\Big(\hat{I}_k - Q_1^T S^T P_1\Big)\,\Sigma_1\Big) - \mathrm{tr}\big(Q_2^T S^T P_2 \Sigma_2\big) \ . \quad (6.21)$$

Moreover, one can observe that

$$\Big(\hat{I}_k - Q_1^T (U \hat{I}_k V^T)^T P_1\Big)_{ii} \geq \frac{1}{\sqrt{k}}\ \left(1 - \left\|V^T (Q_1)_{\cdot i}\right\|_F \left\|U^T (P_1)_{\cdot i}\right\|_F\right) \geq 0 \ . \quad (6.22)$$

As a consequence, the first term of the right hand side of (6.21) is bounded below by

$$\mathrm{tr}\Big(\Big(\hat{I}_k - Q_1^T S^T P_1\Big)\,\Sigma_1\Big) \geq \mathrm{tr}\Big(\hat{I}_k - Q_1^T S^T P_1\Big)\ \sigma_{\min}(\Sigma_1) \ , \quad (6.23)$$

and the second term of the right hand side of (6.21) is bounded below by

$$-\mathrm{tr}\big(Q_2^T S^T P_2 \Sigma_2\big) \geq -\mathrm{tr}\big(\big|Q_2^T S^T P_2 I_{m-k,n-k}\big|\big)\ \sigma_{\max}(\Sigma_2) \ . \quad (6.24)$$

Using (6.23) and (6.24) with (6.21) yields

$$\begin{aligned}\left\langle S^+ - S, \mathcal{M}^2(S)\right\rangle_F \geq\ & \mathrm{tr}\Big(\hat{I}_k - Q_1^T S^T P_1\Big)\ \sigma_{\min}(\Sigma_1) \\ & -\mathrm{tr}\big(\big|Q_2^T S^T P_2 I_{m-k,n-k}\big|\big)\ \sigma_{\max}(\Sigma_2) \ .\end{aligned} \quad (6.25)$$

Adding and subtracting $\sigma_{\max}(\Sigma_2)$ to $\sigma_{\min}(\Sigma_1)$ along with $Q_1^T S^T P_1 \leq |Q_1^T S^T P_1|$ yields

$$\begin{aligned}\left\langle S^+ - S, \mathcal{M}^2(S)\right\rangle_F \geq\ & \mathrm{tr}\Big(\hat{I}_k - Q_1^T S^T P_1\Big)\ (\sigma_{\min}(\Sigma_1) - \sigma_{\max}(\Sigma_2)) \\ & + \Big[\mathrm{tr}\Big(\hat{I}_k\Big) - \mathrm{tr}\big(\big|Q^T S^T P I_{m,n}\big|\big)\Big]\ \sigma_{\max}(\Sigma_2) \ .\end{aligned} \quad (6.26)$$

One can observe that

$$\operatorname{tr}\big(\big|Q^T S^T P I_{m,n}\big|\big) = \operatorname{tr}\Big(Q^T V \hat{I}_k U^T P \tilde{I}_{m,n}\Big) = \operatorname{tr}\Big(\hat{I}_k U^T P \tilde{I}_{m,n} Q^T V\Big)$$

$$= \sum_{i=1}^{k} \frac{1}{\sqrt{k}} (U_{\cdot i})^T P \tilde{I}_{m,n} Q^T V_{\cdot i} \leq \sum_{i=1}^{k} \frac{1}{\sqrt{k}} \left\|P^T U_{\cdot i}\right\|_F \left\|Q^T V_{\cdot i}\right\|_F = \sqrt{k} = \operatorname{tr}\Big(\hat{I}_k\Big)$$

where $(\tilde{I}_{m,n})_{ij} = (I_{m,n})_{ij}\ \operatorname{sign}\Big((Q_{\cdot j})^T V \hat{I}_k U^T P_{\cdot i}\Big)$. And equation (6.26) reduces to

$$\left\langle S^+ - S, \mathcal{M}^2(S)\right\rangle_F \geq \operatorname{tr}\Big(\hat{I}_k - U^T P_1 \hat{I}_k Q_1^T V\Big)\ \left(\sigma_{\min}(\Sigma_1) - \sigma_{\max}(\Sigma_2)\right)\ . \tag{6.27}$$

Finally, one can see that

$$\left\|S^+ - S\right\|_F^2 = 2\ \frac{1}{\sqrt{k}}\ \operatorname{tr}\Big(\hat{I}_k - U^T P_1 \hat{I}_k Q_1^T V\Big)\ . \tag{6.28}$$

Combining (6.28), (6.27), and (6.20) gives the desired result. □

Lemma 6.3 *If S is a fixed point of Algorithm 4, then S is a stationary point of Problem 6.1.*

Proof. Let $S = U\hat{I}_k V^T$ be a fixed point. Then, in view of Lemma 6.1, the ordered singular value decomposition of $\mathcal{M}^2(U\hat{I}_k V^T) = U^+\Sigma_1[V^+]^T + U_\perp \Sigma_2 V_\perp^T$ with $U\hat{I}_k V^T = U^+\hat{I}_k[V^+]^T$. Since $U\hat{I}_k V^T$ and $U^+\hat{I}_k[V^+]^T$ are two singular value decompositions of the same matrix, there must be a square orthogonal matrix Q such that $U^+ = UQ$ and $V^+ = VQ$.

Hence,

$$\mathcal{M}^2(U\hat{I}_k V^T) = U\ Q\Sigma_1 Q^T\ V^T + U_\perp \Sigma_2 V_\perp^T\ , \tag{6.29}$$

and we conclude that $S = U\hat{I}_k V^T$ is a stationary point of Problem 6.1 since equation (6.29) satisfies the stationarity condition (6.11). □

Lemma 6.4 *Let S be a nonstationary point of Problem 6.1. There exists an $\epsilon > 0$ such that for all $\|S_\epsilon\|_F < \epsilon$, $S + S_\epsilon$ is not a stationary point.*

Proof: The critical points of an analytic (non-constant) function form a closed set with empty interior (actually an analytic set). □

The next theorem states the main convergence result for Algorithm 4. Note that the existence of an accumulation point is guaranteed by the fact that the iteration evolves on the compact set $\mathcal{S}_k(m,n)$. In practice, in our experiments, the sequences of iterates always had a single accumulation point S', with $\nu(S') \neq 0$; by virtue of the next theorem, it thus follows that S' is a stationary point of Problem 6.1. Moreover, since the iteration is an ascent iteration for Φ, convergence to stationary points that are not local maxima is not expected to occur in practice.

Theorem 6.5 *Let S', with $\nu(S') \neq 0$ (6.14), be an accumulation point of the sequence $\{S_i\}$ constructed by Algorithm 4. Then S' is a stationary point of Problem 6.1.*

Proof. This proof relies heavily on [Pol71, Sec. 1.3 Th. 3]. Let S' be such an accumulation point, *i.e.* $S_i \to S'$ for $i \in K$ where $K \in \mathbb{N}$ is an infinite index set. Assume that $S' = U'\hat{I}_k V'^T$ is not a stationary point of the iteration. Let $\bar{B}_\epsilon(S')$ denote the closed ball $\{S \in \mathbb{R}^{m\times n} : \|S - S'\| \leq \epsilon\}$. One can choose an $\epsilon > 0$ using Lemma 6.4 such that all $S \in \bar{B}_\epsilon(S')$ are nonstationary points with nonzero gap. In view of Lemma 6.3, these points are not fixed points either, *i.e.*, $\|S^+ - S\|_F^2 > 0$. In view of Theorem 6.2, $\Phi(S^+) - \Phi(S) > 0$ for all $S \in \bar{B}_\epsilon(S')$.

In order to proceed, we need to show that $\Phi(S^+) - \Phi(S)$ is actually bounded away from zero on $\bar{B}_\epsilon(S')$, i.e., (6.30). To this end, it is sufficient to show that $S \mapsto \Phi(S^+) - \Phi(S)$ is continuous for all $S \in \bar{B}_\epsilon(S')$. The function $S \mapsto \Phi(S)$ is continuous in view of the definition of Φ in (6.2). To conclude the continuity argument, we show that the function $S \mapsto S^+$ is also continuous at all points where the gap is nonzero. To see this, observe that $S \mapsto S^+$ is the composition of the function $S \mapsto \mathcal{M}^2(S)$ and of the function $M \mapsto P_1\hat{I}_k Q_1^T = \frac{1}{k}P_1 Q_1^T$ defined from Lemma 6.1. The first function is continuous; it remains to show that the second one is. Consider $M_\star \in \mathcal{M}^2(\bar{B}_\epsilon(S'))$. Let L_M be an orthonormal basis of the dominant k-dimensional left singular subspace of M such that $M \mapsto L_M$ is continuous at $M_\star$; this is possible in view of [Ste73, Th. 6.4]. Let R_M be chosen likewise for the right singular subspace. We then have $P_1 = L_M\tilde{P}_1$ and $Q_1 = R_M\tilde{Q}_1$, where $\tilde{P}_1, \tilde{Q}_2 \in O_k$. Thus $M = L_M\tilde{P}_1\Sigma_1\tilde{Q}_1^T R_M^T + P_2\Sigma_2 Q_2^T$ and

$L_M^T M R_M = \tilde{P}_1 \Sigma_1 \tilde{Q}_1^T = \tilde{P}_1 \tilde{Q}_1^T \, \tilde{Q}_1 \Sigma_1 \tilde{Q}_1^T$. Hence, in view of the polar decomposition, $\tilde{P}_1 \tilde{Q}_1^T = L_M^T M R_M ((L_M^T M R_M)^T L_M^T M R_M)^{-1/2}$. Finally, $P_1 Q_1^T = L_M \tilde{P}_1 \tilde{Q}_1^T R_M^T = L_M L_M^T M R_M ((L_M^T M R_M)^T L_M^T M R_M)^{-1/2} R_M^T$. This function is continuous at $M = M_\star$. (Note that the argument of the square root is positive-definite locally in view of the nonzero gap assumption.) Since $M_\star$ is arbitrary, the claim follows. We have thus shown that

$$\delta := \min_{\|S - S'\|_F \le \epsilon} \Phi(S^+) - \Phi(S) > 0 \; . \tag{6.30}$$

Since $S_i \to S'$ for $i \in K$, there exists a $k \in K$ such that for all $i \ge k$, $i \in K$,

$$\left\| S_i - S' \right\|_F^2 \le \epsilon \qquad \text{and thus} \qquad \Phi(S_{i+1}) - \Phi(S_i) \ge \delta \; . \tag{6.31}$$

Hence, for any two consecutive S_i, S_{i+j} points of the subsequence, with $i, i+j \ge k$, $i, i+j \in K$, we must have

$$\Phi(S_{i+j}) - \Phi(S_i) \ge \Phi(S_{i+1}) - \Phi(S_i) \ge \delta > 0 \; . \tag{6.32}$$

But, since Φ is continuous, the sequence $\{\Phi(S_i)\}_{i \in K}$ must converge. This is contradicted by (6.32), which implies S' has to be a stationary point of Problem 6.1. □

6.6 Approximation of rank at most k

We now consider the following approximations for the feasible set of (6.2).

Problem 6.2 *Solve* (6.2) *with* $\mathcal{S}(m,n)$ *replaced by* $\mathcal{S}_{\le k}(m,n)$*, the set of matrices of norm* 1 *with rank at most* k*,* i.e.

$$\mathcal{S}_{\le k}(m,n) := \left\{ U D V^T \in \mathbb{R}^{m \times n} : \begin{array}{r} U \in V_{k,m}, \; V \in V_{k,n}, \\ D \text{ diagonal}, \; \|D\|_F = 1 \end{array} \right\} \; . \tag{6.33}$$

Note that $\mathcal{S}_{\le k}(m,n)$ is an algebraic set since $\operatorname{rank}(S) \le k$ is equivalent to saying that all minors of S of order $k+1$ are equal to zero.

6.6.1 The Feasible Set of Problem 6.2 and its Tangent Cone

Problem 6.2 is defined on a feasible set that does not have a *manifold* structure. Indeed, as we will further see, the tangent cone $C_S\mathcal{S}_{\leq k}(m,n)$ is no longer a tangent space when $\text{rank}(S) < k$.

Theorem 6.6 *Let $S \in \mathcal{S}_{\leq k}(m,n)$ be of rank $r \leq k$ and let*

$$S = \begin{bmatrix} U_r & U_{r\perp} \end{bmatrix} \begin{bmatrix} D_r & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_r^T \\ V_{r\perp}^T \end{bmatrix}$$

be and ordered singular value decomposition, with $U_r \in \mathbb{R}^{m\times r}$, $U_{r\perp} \in \mathbb{R}^{m\times(m-r)}$, $V_r \in \mathbb{R}^{n\times r}$, $V_{r\perp} \in \mathbb{R}^{n\times(n-k)}$, $D_r \in \text{Diag}(k,k,k)$.

The tangent cone to $\mathcal{S}_{\leq k}(m,n)$ at S is given by

$$C_S\mathcal{S}_{\leq k}(m,n) = \left\{ \begin{array}{l} \begin{bmatrix} U_r & U_{r\perp} \end{bmatrix} \begin{bmatrix} A & B \\ C & R_{k-r} \end{bmatrix} \begin{bmatrix} V_r^T \\ V_{r\perp}^T \end{bmatrix} : \\ \qquad B,\ C \text{ arbitrary},\ \text{tr}(AD_r) = 0, \\ \qquad \text{rank}(R_{k-r}) \leq k-r \end{array} \right\} \tag{6.34}$$

Proof: Let us first define the surjection

$$\psi : \hat{M} \to \mathcal{S}_{\leq k}(m,n) : (U,D,V) \mapsto UDV^T \tag{6.35}$$

where $\hat{M} := O_m \times (\text{Diag}(k,m,n) \cap \text{Norm}(1,m,n)) \times O_n$. Let $\hat{\gamma}(t) : \mathbb{R} \mapsto \mathbb{R}^{m\times m} \times \mathbb{R}^{m\times n} \times \mathbb{R}^{n\times n}$ be an analytic curve with $\hat{\gamma}(0) = (U,D,V)$, and $\hat{\gamma}(t) \in \hat{M}$, for all $t \geq 0$, and γ be a curve defined as

$$\gamma(t) : \mathbb{R} \mapsto \mathbb{R}^{m\times n} : t \to \gamma(t) := \psi(\hat{\gamma}(t)) \ .$$

Clearly, $\gamma(0) = UDV^T$, and $\gamma(t) \in \mathcal{S}_{\leq k}(m,n)$, for all $t \geq 0$. Moreover, since ψ is analytic, γ is also analytic, and we have

$$\dot{\gamma}(0) = \text{D}\psi(U,D,V)\left[\dot{\hat{\gamma}}(0)\right] \ .$$

This holds for any analytic curve $\hat{\gamma}$, so one can write

$$C_S\mathcal{S}_{\leq k}(m,n) \supseteq \bigcup_{(U,D,V)\in\psi^{-1}(S)} \text{D}\psi(U,D,V)\left[C_{(U,D,V)}\hat{M}\right]. \tag{6.36}$$

The next step, which will be fulfilled in (6.48), is to show that the right-hand side of (6.36) is equal to the right-hand side of (6.34).

One can show that $\hat{M}$ is a manifold, and its tangent space at a point $\hat{S} = (U, D, V) \in \psi^{-1}(S)$ is given by

$$C_{\hat{S}}\hat{M} = \left\{ (\Omega_U U, \xi_D, \Omega_V V) : \begin{array}{r} \Omega_U^T + \Omega_U = 0_m\,,\ \Omega_V^T + \Omega_V = 0_n, \\ \xi_D \in \mathrm{Diag}(k,m,n)\,,\ \mathrm{tr}\big(\xi_D^T D\big) = 0 \end{array} \right\}. \tag{6.37}$$

Let us choose $\xi := (\xi_U, \xi_D, \xi_V) = (\Omega_U U, \xi_D, \Omega_V V) \in C_{\hat{S}}\hat{M}$, and write the differential of ψ at $\hat{S}$ in that direction

$$\mathrm{D}\psi(\hat{S})\,[\xi] = \xi_U D V^T + U \xi_D V^T + U D \xi_V^T = \Omega_U S + U \xi_D V^T + S \Omega_V^T\,. \tag{6.38}$$

Let us first change the variables and rewrite Ω_U and Ω_V as follows

$$\Omega_U = \begin{bmatrix} U_r & U_{r\perp} \end{bmatrix} \tilde{\Omega}_U \begin{bmatrix} U_r^T \\ U_{r\perp}^T \end{bmatrix}, \quad \text{and} \quad \Omega_V = \begin{bmatrix} V_r & V_{r\perp} \end{bmatrix} \tilde{\Omega}_V \begin{bmatrix} V_r^T \\ V_{r\perp}^T \end{bmatrix}. \tag{6.39}$$

The conditions on Ω_U and Ω_V can directly be translated in terms of $\tilde{\Omega}_U$ and $\tilde{\Omega}_V$, *i.e.*

$$\tilde{\Omega}_U^T + \tilde{\Omega}_U = 0_m, \quad \text{and} \quad \tilde{\Omega}_V^T + \tilde{\Omega}_V = 0_n\,.$$

Moreover, since (U, D, V) is in $\psi^{-1}(S)$, there must be $P \in O_k \cap \{-1,0,1\}^{k\times k}$ and $Q \in O_k \cap \{-1,0,1\}^{k\times k}$ such that $\begin{bmatrix} P^T & 0 \\ 0 & I_{m-k} \end{bmatrix} D \begin{bmatrix} Q & 0 \\ 0 & I_{n-k} \end{bmatrix} = \begin{bmatrix} D_r & 0 \\ 0 & 0 \end{bmatrix}$. Let us change the variable and rewrite ξ_D as follows

$$\xi_D = \begin{bmatrix} P & 0 \\ 0 & I_{m-k} \end{bmatrix} \tilde{\xi}_D \begin{bmatrix} Q^T & 0 \\ 0 & I_{m-k} \end{bmatrix}. \tag{6.40}$$

The conditions on ξ_D translate to

$$\tilde{\xi}_D \in \mathrm{Diag}(k,m,n), \qquad \text{and} \qquad \mathrm{tr}\Big(\tilde{\xi}_D^T \begin{bmatrix} D_r & 0 \\ 0 & 0 \end{bmatrix}\Big) = 0.$$

Finally, let us define $U_P := U \begin{bmatrix} P & 0 \\ 0 & I_{m-k} \end{bmatrix}$, $V_Q := V \begin{bmatrix} Q & 0 \\ 0 & I_{m-k} \end{bmatrix}$, and $d_1, \cdots, d_s$, $r_1, \cdots, r_s$ such that

$$D_r = \begin{bmatrix} d_1 I_{r_1} & & \\ & \ddots & \\ & & d_s I_{r_s} \end{bmatrix}$$

with $d_1 > d_2 > \cdots > d_s > 0$ and $r_1 + r_2 + \cdots + r_s = r$. Since

$$S = U_P \begin{bmatrix} D_r & 0 \\ 0 & 0 \end{bmatrix} V_Q^T = \begin{bmatrix} U_r & U_{r\perp} \end{bmatrix} \begin{bmatrix} D_r & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_r^T \\ V_{r\perp}^T \end{bmatrix} ,$$

one can show that

$$\begin{bmatrix} U_r & U_{r\perp} \end{bmatrix}^T U_P = R_U := \begin{bmatrix} R_1 & & & \\ & \ddots & & \\ & & R_s & \\ & & & R_{\perp U} \end{bmatrix} \tag{6.41}$$

and

$$\begin{bmatrix} V_r & V_{r\perp} \end{bmatrix}^T V_Q = R_V := \begin{bmatrix} R_1 & & & \\ & \ddots & & \\ & & R_s & \\ & & & R_{\perp V} \end{bmatrix} \tag{6.42}$$

where $R_1, \cdots, R_s$, $R_{\perp U}$, and $R_{\perp V}$ are square orthogonal matrices respectively of order $r_1, \cdots, r_s$, $m - r$ and $n - r$.

By using the equations (6.39), (6.40), (6.41), and (6.42), one can rewrite (6.38) as

$$\mathrm{D}\psi(\hat{S})\,[\xi] = \begin{bmatrix} U_r & U_{r\perp} \end{bmatrix} \left(\tilde{\Omega}_U \begin{bmatrix} D_r & 0 \\ 0 & 0 \end{bmatrix} + R_U \tilde{\xi}_D R_V^T + \begin{bmatrix} D_r & 0 \\ 0 & 0 \end{bmatrix} \tilde{\Omega}_V^T \right) \begin{bmatrix} V_r^T \\ V_{r\perp}^T \end{bmatrix} . \tag{6.43}$$

Let us write $\tilde{\Omega}_U$, $\tilde{\xi}_D$, and $\tilde{\Omega}_V$ as follows

$$\tilde{\Omega}_U = \begin{bmatrix} \omega_U & -K_U^T \\ K_U & \omega_{\perp U} \end{bmatrix} , \quad \tilde{\xi}_D = \begin{bmatrix} \tilde{\xi}_{r_1} & & & \\ & \ddots & & \\ & & \tilde{\xi}_{r_s} & \\ & & & \tilde{\xi}_\perp \end{bmatrix} , \quad \text{and} \quad \tilde{\Omega}_V = \begin{bmatrix} \omega_V & -K_V^T \\ K_V & \omega_{\perp V} \end{bmatrix} .$$

The conditions on $\tilde{\Omega}_U$, $\tilde{\xi}_D$, and $\tilde{\Omega}_V$ translate to $\omega_U^T + \omega_U = \omega_V^T + \omega_V = 0_r$, K_U, K_V, arbitrary matrices, $\tilde{\xi}_\perp \in \mathrm{Diag}(k - r, m - r, n - r)$, and $\tilde{\xi}_{r_i} \in \mathrm{Diag}(r_i, r_i, r_i)$ such that

$$\mathrm{tr}\left(\begin{bmatrix} \tilde{\xi}_{r_1} & & & \\ & \ddots & & \\ & & \tilde{\xi}_{r_s} & \\ & & & \end{bmatrix}^T D_r \right) = 0 . \tag{6.44}$$

One can further change the variables and rewrite ω_U and ω_V as follows

$$\omega_U = \omega_1 + \omega_2 , \quad \text{and} \quad \omega_V = \omega_1 - \omega_2 . \tag{6.45}$$

The previous conditions translate to $\omega_1^T + \omega_1 = \omega_2^T + \omega_2 = 0_r$. Equation (6.43) then becomes

$$\mathrm{D}\psi(\hat{S})\,[\xi] = \begin{bmatrix} U_r & U_{r\perp} \end{bmatrix} \begin{bmatrix} A & D_r K_V^T \\ K_U D_r & R_{\perp U}\ \tilde{\xi}_\perp\ {R_{\perp V}}^T \end{bmatrix} \begin{bmatrix} V_r^T \\ V_{r\perp}^T \end{bmatrix} , \qquad (6.46)$$

with

$$A = \omega_1 D_r - D_r \omega_1 + \begin{bmatrix} R_1 \tilde{\xi}_{r_1} R_1^T & & \\ & \ddots & \\ & & R_s \tilde{\xi}_{r_s} R_s^T \end{bmatrix} + \omega_2 D_r + D_r \omega_2 \ . \qquad (6.47)$$

One can first see that $\mathrm{Skew}A = \omega_2 D_r + D_r \omega_2$. Moreover, the skew-symmetric part of A can be made equal to any skew-symmetric matrix Ω by choosing ω_2 such that

$$[\omega_2]_{ij} = \frac{\Omega_{ij}}{[D_r]_{ii} + [D_r]_{jj}} \ .$$

One can further see that

$$\mathrm{Sym}(A) = \omega_1 D_r - D_r \omega_1 + \begin{bmatrix} R_1 \tilde{\xi}_{r_1} R_1^T & & \\ & \ddots & \\ & & R_s \tilde{\xi}_{r_s} R_s^T \end{bmatrix} \ .$$

Moreover, the symmetric part of A can be made equal to any symmetric matrix H with $\mathrm{tr}(HD_r) = 0$ by choosing ω_1 and $R_1, \cdots, R_s$, $\tilde{\xi}_{r_1}, \cdots, \tilde{\xi}_{r_s}$ according to the constraints: block-partition H as

$$H = \begin{bmatrix} H_{11} & \cdots & H_{1s} \\ \vdots & & \vdots \\ H_{s1} & \cdots & H_{ss} \end{bmatrix} ,$$

with $H_{ij} \in \mathbb{R}^{r_i \times r_j}$ and choose

$$\omega_1 = \begin{bmatrix} 0 & \frac{H_{12}}{d_2 - d_1} & \cdots & \frac{H_{1s}}{d_s - d_1} \\ \frac{H_{21}}{d_1 - d_2} & 0 & \ddots & \vdots \\ \vdots & \ddots & 0 & \frac{H_{s-1,s}}{d_s - d_{s-1}} \\ \frac{H_{s1}}{d_1 - d_s} & \cdots & \frac{H_{s,s-1}}{d_{s-1} - d_s} & 0 \end{bmatrix} \quad \text{and} \quad R_i \tilde{\xi}_{r_i} R_i^T = H_{ii} \ ,$$

to get $\mathrm{Sym}(A) = H$. The condition $\mathrm{tr}(HD_r) = 0$ comes from (6.44).

These observations yield that A can be made equal to any arbitrary matrix as long as

$$\operatorname{tr}(AD_r) = 0 \ .$$

Returning to (6.46), let us define $B := D_r K_V^T$, $C := K_U D_r$, and $R_{k-r} := R_{\perp U}\ \tilde{\xi}_\perp\ R_{\perp V}{}^T$. Clearly, B and C can be set to any matrix by choosing $K_V^T = D_r^{-1}B$ and $K_U = CD_r^{-1}$, and, since $\tilde{\xi}_\perp \in \operatorname{Diag}(k-r, m-r, n-r)$, R_{k-r} can be any arbitrary matrix of rank less or equal to $k-r$ by choosing $R_{\perp U}$, $\tilde{\xi}_\perp$, $R_{\perp V}{}^T$ equal to its ordered singular value decomposition. In view of (6.36), we conclude that

$$C_S\mathcal{S}_{\le k}(m,n) \supseteq \left\{ \begin{array}{l} \begin{bmatrix} U_r & U_{r\perp} \end{bmatrix} \begin{bmatrix} A & B \\ C & R_{k-r} \end{bmatrix} \begin{bmatrix} V_r^T \\ V_{r\perp}^T \end{bmatrix} : \\ \qquad B,\, C \text{ arbitrary},\ \operatorname{tr}(AD_r) = 0, \\ \qquad \operatorname{rank}(R_{k-r}) \le k-r \end{array} \right\} \tag{6.48}$$

The "$\subseteq$" part follows directly from [BGBMN91, Theorem 1], that is, if $t \mapsto S(t)$ is a matrix-valued curve, then there exists a decomposition $S(t) = U(t)D(t)V(t)^T$, where $U(\cdot)$ and $V(\cdot)$ are orthonormal and analytic and $D(\cdot)$ is diagonal and analytic. □

6.6.2 Characterization of the Stationary Points of Problem 6.2

Let now $S \in \mathcal{S}_{\le k}(m,n)$ be of rank $r \le k$, with an ordered singular value decomposition given by

$$S = \begin{bmatrix} U_r & U_{r\perp} \end{bmatrix} \begin{bmatrix} D_r & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_r^T \\ V_{r\perp}^T \end{bmatrix} \ .$$

Since $\operatorname{grad}\Phi(S) = 2\mathcal{M}^2(S)$, S is a stationary point of Problem 6.2 if and only if

$$\mathcal{M}^2(S) \in N_S\mathcal{S}_{\le k}(m,n) = \left\{ \begin{array}{l} \begin{bmatrix} U_r & U_{r\perp} \end{bmatrix} \begin{bmatrix} \alpha D_r & 0 \\ 0 & R_\perp \end{bmatrix} \begin{bmatrix} V_r^T \\ V_{r\perp}^T \end{bmatrix} : \\ \qquad \alpha \in \mathbb{R},\ R_\perp = 0 \text{ if } r < k,\ R_\perp \text{ arbitrary if } r = k \end{array} \right\}$$

$$= \left\{ \alpha S + U_{r\perp} R_\perp V_{r\perp}^T : \alpha \in \mathbb{R},\ R_\perp = 0 \text{ if } r < k,\ R_\perp \text{ arbitrary if } r = k \right\} \tag{6.49}$$

6.6.3 Algorithm for Problem 6.2 and its Convergence Analysis

We now propose Algorithm 5 to find a stationary point of Problem 6.2.

Algorithm 5

Require: $G(A)$ and $G(B)$, two graphs respectively of order m and n.

$S^0 \leftarrow \mathbf{1}/\left\|\mathbf{1}\right\|_F$

for $t = 1, 2, \cdots, t_{\max}$ **do**

Compute $S^t \in \mathcal{S}_{\leq k}(m,n)$ according to

$$S^t \ (= U^t D^t [V^t]^T) \ \leftarrow f(S^{t-1}) := \underset{\tilde{S} \in \mathcal{S}_{\leq k}(m,n)}{\operatorname{argmax}} \left\langle \tilde{S}, \mathcal{M}^2(S^{t-1}) \right\rangle_F . \tag{6.50}$$

end for

$S^{\text{Blondel}} \leftarrow S^t$

where $\mathcal{M}^2(S) = \mathcal{M}(\mathcal{M}(S))$ is defined in equation (6.1).

Notice that Algorithm 5 is a particular case of [JNRS10, Algorithm 1] in which the convex function, $f(x)$, and the compact set, $\mathcal{Q}$, are respectively $\Phi(S)$ and $\mathcal{S}_{\leq k}(m,n)$.

When $\nu(S^{t-1}) \neq 0$ (6.14) or $\sigma_k(\mathcal{M}^2(S^{t-1})) = 0$, the next iterate S^t is uniquely defined by (6.50) and is equal to $\frac{1}{\|\Sigma_1\|_F} P_1 \Sigma_1 Q_1^T$, as we will show in Lemma 6.7 below. Otherwise, (i.e., when $\sigma_k(\mathcal{M}^2(S^{t-1})) = \sigma_{k+1}(\mathcal{M}^2(S^{t-1})) > 0$), S^t is no longer uniquely defined by (6.50); in this case, S^t is chosen arbitrarily in $\underset{\tilde{S} \in \mathcal{S}_{\leq k}(m,n)}{\operatorname{argmax}} \left\langle \tilde{S}, \mathcal{M}^2(S^{t-1}) \right\rangle_F$. This case was never observed in our numerical experiments, but if it did, a possible choice would have been $S^t := \frac{1}{\|\Sigma_1\|_F} P_1 \Sigma_1 Q_1^T$, where P_1, Σ_1 and Q_1 are returned by the SVD function.

We first state a few intermediate results in order to prove convergence of Algorithm 5 to the stationary points of Problem 6.2 (see Theorem 6.11).

Lemma 6.7 *Let $M \in \mathbb{R}^{m\times n}$ and its ordered singular value decomposition*

$$M = \begin{bmatrix} P_1 & P_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix} = P\Sigma Q^T \tag{6.51}$$

with $P_1 \in \mathbb{R}^{m\times k}$, $P_2 \in \mathbb{R}^{m\times(m-k)}$, $Q_1 \in \mathbb{R}^{n\times k}$, $Q_2 \in \mathbb{R}^{n\times(n-k)}$, $\Sigma_1 \in \mathbb{R}^{k\times k}$ and $\Sigma_2 \in \mathbb{R}^{(m-k)\times(n-k)}$. Then

$$\max_{S=UDV^T\in\mathcal{S}_{\leq k}(m,n)} \langle S, M\rangle_F = \operatorname{tr}\Big(\hat{\Sigma}_1\Sigma_1\Big). \tag{6.52}$$

where $\hat{\Sigma}_1 := \Sigma_1 / \|\Sigma_1\|_F$.

Moreover, if $\nu := \sigma_{\min}(\Sigma_1) - \sigma_{\max}(\Sigma_2) > 0$ or if $\|\Sigma_2\|_F = 0$, then the maximizing solution S is unique and equals $P_1\hat{\Sigma}_1 Q_1^T$.

Proof. We have

$$\begin{aligned} \operatorname{tr}\big(DV^TMU\big) &\leq \textstyle\sum_{i=1}^k \sigma_i(DV^TMU) \leq \sum_{i=1}^k \sigma_i(D)\ \sigma_i(V^TMU) \\ &\leq \textstyle\sum_{i=1}^k \sigma_i(D)\ \sigma_i(\Sigma_1) \leq \operatorname{tr}\Big(\hat{\Sigma}_1\Sigma_1\Big) \end{aligned} \tag{6.53}$$

according to [HJ91, Formulas 3.1.10b, and Lemma 3.3.1]. The upper bound is reached for $U = P_1$, $D = \hat{\Sigma}_1$ and $V = Q_1$. The uniqueness of $P_1\hat{\Sigma}_1 Q_1^T$ is a well known result discussed, *e.g.*, in [HJ91, Theorem 3.1.1 and 3.1.1'] □

Theorem 6.8 *Let $S \in \mathcal{S}_{\leq k}(m,n)$ and $\mathcal{M}^2(S)$ have an ordered singular value decomposition*

$$\mathcal{M}^2(S) = \begin{bmatrix} P_1 & P_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix} = P\Sigma Q^T\ , \tag{6.54}$$

with $P_1 \in \mathbb{R}^{m\times k}$, $P_2 \in \mathbb{R}^{m\times(m-k)}$, $Q_1 \in \mathbb{R}^{n\times k}$, $Q_2 \in \mathbb{R}^{n\times(n-k)}$, $\Sigma_1 \in \mathbb{R}^{k\times k}$ and $\Sigma_2 \in \mathbb{R}^{(m-k)\times(n-k)}$, and $S^+ := f(S)$, with f the function defined in Algorithm 5. Then

$$\Phi(S^+) - \Phi(S) \geq \big\|\mathcal{M}(S^+ - S)\big\|_F^2\ , \tag{6.55}$$

with Φ the function defined in equation (6.2), and $\nu = \sigma_{\min}(\Sigma_1) - \sigma_{\max}(\Sigma_2)$. Moreover, if $\nu = \sigma_{\min}(\Sigma_1) - \sigma_{\max}(\Sigma_2) > 0$, then the inequality becomes an equality if and only if S is a fixed point of the iteration, i.e. *$S^+ = S$.*

Proof. Adding and subtracting S to S^+ and using the self-adjointness of $\mathcal{M}$ yield

$$\Phi(S^+ - S + S) = \Phi(S) + \left\|\mathcal{M}(S^+ - S)\right\|_F^2 + 2\left\langle S^+ - S, \mathcal{M}^2(S)\right\rangle_F \ . \quad (6.56)$$

One can observe that $\left\langle S^+ - S, \mathcal{M}^2(S)\right\rangle_F$ is positive since S^+ maximizes the scalar product with $\mathcal{M}^2(S)$, and hence

$$\left\langle S^+, \mathcal{M}^2(S)\right\rangle_F \geq \left\langle S, \mathcal{M}^2(S)\right\rangle_F \ . \quad (6.57)$$

The equation (6.56) combined with (6.57) gives the desired result.
Moreover, if $\nu = \sigma_{\min}(\Sigma_1) - \sigma_{\max}(\Sigma_2) > 0$, Lemma 6.7 ensures that the maximizing solution is unique. Hence, unless $S^+ = S$, the last term of the right hand side is strictly positive. □

Lemma 6.9 *If S is a fixed point of Algorithm 5 then S is a stationary point of Problem 6.2.*

Proof. Let $S = UDV^T$ be a fixed point, *i.e.* $S^+ = U^+D^+[V^+]^T = UDV^T$ and the ordered singular value decomposition of $\mathcal{M}^2(UDV^T)$ is

$$\mathcal{M}^2(UDV^T) = \begin{bmatrix} U^+ & U_\perp \end{bmatrix} \begin{bmatrix} \alpha D^+ & \\ & \Sigma_2 \end{bmatrix} \begin{bmatrix} [V^+]^T \\ V_\perp^T \end{bmatrix} \ , \quad (6.58)$$

with $\sigma_{\min}(\alpha D^+) \geq \sigma_{\max}(\Sigma_2)$. Remark that if the rank of $U^+D^+[V^+]^T$ is lower than k, then $\sigma_{\min}(\alpha D^+) = 0$ and $\Sigma_2 = 0_{m-k,n-k}$.
Let us now remind that the gradient of Φ at a point S is $2\mathcal{M}^2(S)$. One can verify that this expression is in the normal cone given by equation (6.49) and is hence a stationary point. □

Notice that all stationary points are not fixed points since Σ_2 has to be such that $\sigma_{\min}(\alpha D^+) \geq \sigma_{\max}(\Sigma_2)$.

Lemma 6.10 *Let S be a nonstationary point of Problem 6.2. There exists an $\epsilon > 0$ such that for all $\|S_\epsilon\|_F < \epsilon$, $S + S_\epsilon$ is not a stationary point.*

Proof. Since $S = U_r D_r V_r^T$ is not a stationary point, we have

$$2\ \mathcal{M}^2(U_r D_r V_r^T) = \begin{bmatrix} U_r & U_{r\perp} \end{bmatrix} \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} V_r^T \\ V_{r\perp}^T \end{bmatrix} \ , \quad (6.59)$$

with either, if $r = k$,

$$\delta_{\max} := \max\left(\left\|M_{11} - \frac{\|M_{11}\|_F}{\|D_r\|_F} D_r\right\|_F^2, \|M_{12}\|_F^2, \|M_{21}\|_F^2\right) > 0\ , \quad (6.60)$$

or, if $r < k$,

$$\delta_{\max} := \max\left(\left\|M_{11} - \frac{\|M_{11}\|_F}{\|D_r\|_F} D_r\right\|_F^2, \|M_{12}\|_F^2, \|M_{21}\|_F^2, \|M_{22}\|_F^2\right) > 0\ . \quad (6.61)$$

Since $\mathcal{M}^2(\cdot)$ is a continuous mapping, for all $\delta > 0$ there always exists $\epsilon(\delta) > 0$ such that for all $\|S_\epsilon\|_F < \epsilon(\delta)$, we have $\left\|\mathcal{M}^2(S + S_\epsilon) - \mathcal{M}^2(S)\right\|_F < \delta$. A reasoning similar to the one held for Lemma 6.4, when one chooses $\delta \leq \delta_{\max}$, yields the desired result. □

The next theorem states the main convergence result for Algorithm 5. Note that the existence of an accumulation point is guaranteed by the fact that the iteration evolves on the compact set $\mathcal{S}_{\leq k}(m, n)$. In practice, in our experiments, the sequences of iterates always had a single accumulation point S', with $\nu(S') \neq 0$ or $\sigma_k(\mathcal{M}^2(S')) = 0$; by virtue of the next theorem, it thus follows that S' is a stationary point of Problem 6.2. Moreover, since the iteration is an ascent iteration for Φ, convergence to stationary points that are not local maxima is not expected to occur in practice.

Theorem 6.11 *Let S', with $\nu(S') > 0$ (6.14) or $\sigma_k(\mathcal{M}^2(S')) = 0$, be an accumulation point of the sequence $\{S_i\}$ constructed by 5. Then S' is a stationary point of Problem 6.2.*

Proof. A reasoning similar to the one held for Theorem 6.5, along with Theorem 6.8 and Lemmas 6.10, 6.9, yields the desired result. In this parallelism Theorem 6.2 is replaced by Theorem 6.8, which lacks the term $\nu\sqrt{k}\,\|S^+ - S\|_F^2$. However, Theorem 6.8 still ensures that $\Phi(S^+) - \Phi(S) > 0$ holds whenever S is not a fixed point of the algorithm. The continuity argument for S_+ still holds since we now simply have $S_+ = L_M L_M^T M R_M R_M^T$. Thus the reasoning in the proof of Theorem 6.5 still holds for Problem 6.2, and the result follows. □

6.7 Complexity Analysis

The adjacency matrices A and B respectively contain $|E(A)|$ and $|E(B)|$ non-zero elements.

Let us first consider the complexity of one step of Algorithm 3, *i.e.*

$$S^t \leftarrow \frac{AS^{t-1}B^T + A^T S^{t-1} B}{\|AS^{t-1}B^T + A^T S^{t-1} B\|_F}$$

Assuming that S^{t-1} is a dense matrix, the products AS^{t-1} and $A^T S^{t-1}$ require less than $2\,|E(A)|\,n$ flops each, while the subsequent products $(AS^{t-1})B^T$ and $(A^T S^{t-1})B$ require less than $2\,|E(B)|\,m$ flops each. The sum and the calculation of the Frobenius norm requires $2mn$ flops, while the scaling requires one division and nm multiplications. Then, the total complexity per iteration step is of the order of $4(|E(A)|\,n + |E(B)|\,m)$ flops.

Let us now consider the complexity of one step of Algorithm 4 and 5. In these algorithms, the rank of S is at most k. Hence, in practice, we do not really work with $S \in \mathbb{R}^{m\times n}$ itself but with its singular value factorization $(U, D, V) \in \mathbb{R}^{m\times k} \times \mathrm{Diag}(k,k,k) \times \mathbb{R}^{n\times k}$. When k is small, the space required to store the factors of S (i.e. $mk + k + nk$ elements) is smaller than the one required to store S itself (i.e. mn elements). Similarly, in practice, we do not really compute $\mathcal{M}^2(S) \in \mathbb{R}^{m\times n}$ itself but its singular value factorization. We now show how we compute the factors of the singular value decomposition of $\mathcal{M}^2(S)$. Notice first that $\mathcal{M}^2(UDV^T)$ can be written as $U_A\,(I_4 \otimes D)\,V_B^T$ with

$$U_A := \begin{bmatrix} U^TAA \\ U^TAA^T \\ U^TA^TA \\ U^TA^TA^T \end{bmatrix}^T, \quad (I_4 \otimes D) := \begin{bmatrix} D & & & \\ & D & & \\ & & D & \\ & & & D \end{bmatrix}, \text{ and } V_B := \begin{bmatrix} V^TBB \\ V^TBB^T \\ V^TB^TB \\ V^TB^TB^T \end{bmatrix}^T. \tag{6.62}$$

One can further compute $Q_A \in \mathbb{R}^{m\times 4k}$ and $R_A \in \mathbb{R}^{4k\times 4k}$ (resp. $Q_B \in \mathbb{R}^{n\times 4k}$ and $R_B \in \mathbb{R}^{4k\times 4k}$), the factors of the QR decomposition of U_A (resp. V_B), *i.e.* $Q_AR_A = U_A$, with $Q_A^TQ_A = I_{4k}$ and $[R_A]_{ij} = 0$ for all $i > j$, and rewrite equation (6.62) as $\mathcal{M}^2(UDV^T) = Q_AR_A\,(I_4 \otimes D)\,R_B^TQ_B^T$. We further compute $(\bar{U}, \bar{D}, \bar{V}) \in \mathbb{R}^{4k\times 4k} \times \mathrm{Diag}(4k,4k,4k) \times \mathbb{R}^{n\times 4k}$, the factors of the singular value decomposition of $R_A\,(I_4 \otimes D)\,R_B^T$. Finally, one can write $\mathcal{M}^2(UDV^T) = Q_A\bar{U}\,\bar{D}\,\bar{V}^TQ_B^T$, and the factors of its singular value decomposition are given by $(Q_A\bar{U}, \bar{D}, Q_B\bar{V})$. And, eventually, Algorithm 5 can be rewritten as follows.

1: $(U^0, D^0, V^0) \leftarrow SVD_k(\mathbf{1}/\left\|\mathbf{1}\right\|_F)$;

2: **for** $t = 1, 2, \cdots$

3: $\quad U' \leftarrow [AU^{t-1}D^{t-1}, A^TU^{t-1}D^{t-1}] \in \mathbb{R}^{m\times 2k}$;

4: $\quad U'' \leftarrow [AU', A^TU'] \in \mathbb{R}^{m\times 4k}$;

5: $\quad V' \leftarrow [BV^{t-1}, B^TV^{t-1}] \in \mathbb{R}^{n\times 2k}$;

6: $\quad V'' \leftarrow [BV', B^TV'] \in \mathbb{R}^{n\times 4k}$;

7: $\quad (Q_U, R_U) \leftarrow QR(U'') \in \mathbb{R}^{m\times 4k} \times \mathbb{R}^{4k\times 4k}$;

8: $\quad (Q_V, R_V) \leftarrow QR(V'') \in \mathbb{R}^{n\times 4k} \times \mathbb{R}^{4k\times 4k}$;

9: $\quad (U''', D''', V''') \leftarrow SVD_k(R_UR_V^T) \in \mathbb{R}^{m\times k} \times \mathbb{R}^{k\times k} \times \mathbb{R}^{n\times k}$;

10: $\quad (U^t, D^t, V^t) \leftarrow (Q_UU''', \frac{D'''}{\|D'''\|}, Q_VV''')$;

11: **end**

We now consider the complexity of computing the singular value decomposition of $\mathcal{M}^2(UDV^T)$ with UDV^T a matrix of rank k. The products AU, A^TU, AU', and A^TU' require less than $2\left|E(A)\right|k$ flops each, whereas the products BV, B^TV, BV', and B^TV' require less than $2\left|E(B)\right|k$ flops each. The QR factorization of a matrix $M \in \mathbb{R}^{m\times k}$ requires $4mk^2 - \frac{4}{3}k^3$ flops [Hig08, p. 337] and subsequently computing Q_AR_A and Q_BR_B require receptively less than $4m(4k)^2 - \frac{4}{3}(4k)^3$ and $4n(4k)^2 - \frac{4}{3}(4k)^3$ flops. The product $R_A\left(I_4 \otimes D\right)R_B^T$ require less than $2(4k)^3$ flops. The complexity of computing the singular value decomposition of $\bar{U}\bar{D}\bar{V} \in \mathbb{R}^{4k\times 4k}$ up to a given precision is of the order of k^3 flops. Finally, the products $Q_A\bar{U}$ and $Q_B\bar{V}$ require less than $2m(4k)^2$ and $2n(4k)^2$ flops each. Hence, in total, computing the singular value decomposition of $\mathcal{M}^2(UDV^T)$ requires less than $8\left|E(A)\right|k + 8\left|E(B)\right|k + 96mk^2 + 96nk^2 + O(k^3)$ flops. Let now $\mathcal{M}^2(S^{t-1})$ admit the following ordered singular value decomposition

$$\mathcal{M}^2(S^{t-1}) = \begin{bmatrix} P_1 & P_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix}$$

with $P_1 \in \mathbb{R}^{m\times k}$, $Q_1 \in \mathbb{R}^{n\times k}$, and $\Sigma_1 \in \mathbb{R}^{k\times k}$. According to Lemmas 6.1 and 6.7, one step of Algorithm 4 and 5 consists in choosing S^t respectively equal to $P_1\hat{I}_kQ_1^T$ and $P_1\hat{\Sigma}_1Q_1^T$. The number of operations required

to compute one step of Algorithm 4 and 5 is then equal to the one required to compute the singular value decomposition of $\mathcal{M}^2(UDV^T)$ which costs

$$8\,|E(A)|\,k + 8\,|E(B)|\,k + 96mk^2 + 96nk^2 + O(k^3) \text{ flops.}$$

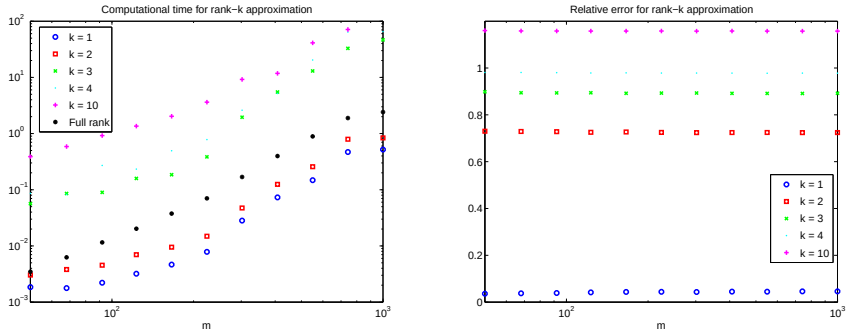
Let us remind that one step of Algorithm 4 and 5 are low-rank approximations of two steps of 3 which costs $8(|E(A)|\,n + |E(B)|\,m)$ flops.

6.8 Experiments

We look at the performances of our method to compute self-similarity matrices. This means that A and B are equal. In other words, the self-similarity matrix expresses how a node of a graph is similar to other nodes of the same graph. We ran several experiments to compute low-rank approximations of self-similarity matrices on random graphs. Results about the average computational time, and the average relative error with respect to the full rank self-similarity matrices for exactly k identical nonzero eigenvalues and for at most k nonzero eigenvalues are shown respectively in Figure 6.1 and Figure 6.2. Results about the speed of convergence are shown in Figure 6.3. As expected, we clearly notice that the smaller the rank of the approximation k, the smaller the computational time. We further notice that, when the order of the graph m increases, the algorithm for low rank approximation converges faster than the full rank algorithm. As far as the relative error is concerned, we observe that it does not vary much with m, the order of this graph. For exactly k identical nonzero eigenvalues, this error increases when the rank of the approximation increases. Indeed, rank 1 approximation have a relative error about 0.05 whereas higher rank approximation are about 0.7 up to 1.2 ! This reveals that the equal eigenvalues assumption is not adequate for this class of graphs. For at most k nonzero eigenvalues, the results are much more satisfactory since the error decreases when the rank of the approximation increases.

6.9 Conclusions

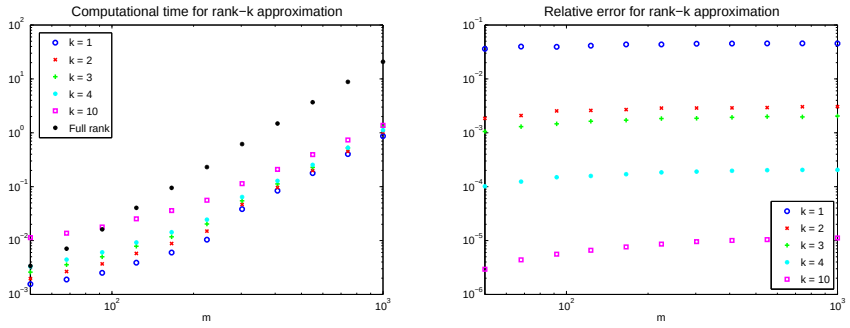
In this chapter, we have considered two optimization problems (Problem 6.1 and Problem 6.2) whose solutions are low-rank approximations



(a) (b)

Figure 6.1: We compute rank-k approximations with exactly k identical nonzero eigenvalues of the self-similarity matrix of a connected Erdós-Rényi graph with probability $10/m$*, where* m *is the order of this graph. The graph is built such that the average number of outgoing edges of a node is* 10*. The algorithm stops when* $\|\Delta_S\|_F \leq 10^{-6} \|S\|_F$*. The full rank results are obtained using Algorithm 3 which was investigated in [BGH+04]. (a) shows the average computational time versus* m*, the order of this graph, (b) shows the average relative error of the rank-k approximations of the self-similarity matrix of a connected random graph versus* m*.*

of the similarity matrix S^{Blondel} introduced by Blondel *et al.* in [BGH+04]. The cost functions of Problem 6.1 and Problem 6.2 are the same as the one presented in Equation (6.2) whereas their feasible sets are respectively set to $\mathcal{S}_k(m,n)$ and $\mathcal{S}_{\leq k}(m,n)$ instead of $\mathcal{S}(m,n)$. We have first characterized the stationary points of Problem 6.1 and Problem 6.2. Then we have considered Algorithms 4 and 5 and proved that their accumulation points are stationary points of respectively Problem 6.1 and Problem 6.2. Next, we have analyzed the complexity of one step of Algorithms 4 and 5 and compared them to the complexity of Algorithm 3 used to compute the original similarity matrix S^{Blondel} introduced by Blondel *et al.* We have further performed numerical experiments and considered the performances of Algorithms 4 and 5. Finally, we have concluded that Problem 6.1 is not adequate to find a low rank approximation of the optimization problem presented in Equation (6.2) since the relative error of the approximations of rank bigger than 2 is about 100%. On the other hand, the solution of Problem 6.2 appropriately approaches the solution of the optimization problem presented in Equation (6.2). As expected, we have observed that the relative error of



(a) (b)

Figure 6.2: We compute rank-k approximations with at most k nonzero eigenvalues of the self-similarity matrix of a connected random graph. The graph is built such that the average number of outgoing edges of a node is 10*. The algorithm stops when* $\|\Delta_S\|_F \leq 10^{-6} \|S\|_F$*. The full rank results are obtained using Algorithm 3 which was investigated in [BGH+04]. (a) shows the average computational time versus m, the order of this graph, (b) shows the average relative error of the rank-k approximations of the self-similarity matrix of a connected random graph versus m.*

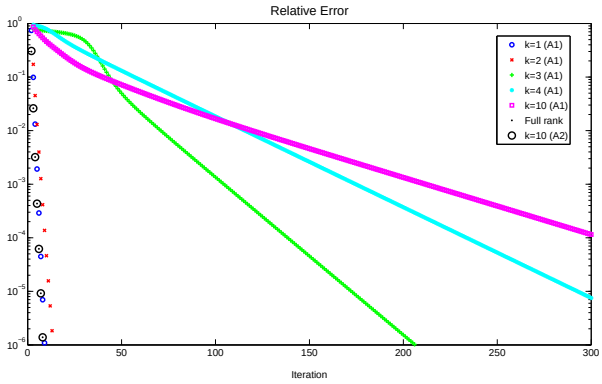


Figure 6.3: The (A1) (resp. (A2)) lines refer to the experiment of figure 6.1 (resp. 6.2) which shows results for Algorithm 4 (resp. 5). The graph shows the relative distance between an iterate and the extremal point of the corresponding experiment ($\frac{\|S^t - S^\infty\|}{\|S^\infty\|}$) *versus t, the number of iterations.*

approximation decreases when the rank of the approximation increases,

and the ratio between the time until convergence of Algorithm 5 and the time until convergence of Algorithm 3 decreases as m and n (the size of the problem) grow.

Chapter 7

Conclusion

In this thesis, we focused on two analysis tools designed to remedy the sensitivity of regular equivalence relations in graphs, *i.e.*

- node-to-node self-similarity measure, and
- Blockmodeling,

and presented several associated trace maximization problems.

In Chapter 3, we introduced the notion of *weak and strong compatibility* which links node-to-node self-similarity measures to equivalence relations, we presented several existing node-to-node similarity measures, considered their strengths and weaknesses, and proposed alternative definitions which improve the existing ones, we emphasized that several reinforcement functions of the node-to-node similarity measures are affine functions, and finally we classified all similarity measures according to their properties. A difficulty that several existing node-to-node similarity measures encounter is that they are either easy to analyze and disconnected from specific applications, such as S^{Blondel}, or very specifically designed for a very particular application and not really interesting in terms of analysis. We hope that this thesis will help to emphasize this gap when one has to design new node-to-node similarity measures. Moreover, further work could be carried out in order to prove that the similarity measure $S_{\alpha}^{\text{Cooper}''\text{-}r}(A, A)$ is or is not strongly isomorphic with r either infinite or bounded by a polynomial expression in m, the order of the graph $G(A)$.

In Chapter 4, we defined the notion of relevance of a Blockmodel with respect to a given quality function, we presented and analyzed existing Blockmodel quality measures, along with their strengths and weaknesses, and we proposed a novel Blockmodel quality measure. We showed that the maximization of the quality measures that we considered can be translated in terms of trace-maximization problems, we proposed an Algorithm to solve these problems and tested it on a toy example with 40 nodes. The science of network visualization has extensively studied community detection while very little work has been done in Blockmodel detection. We think it would possible to benefit from the analysis of community detection in order to solve problems that arise in Blockmodel detection. *E.g.*, further work could first be carried out in order to

- apply Algorithm 2 on other real world examples, and
- find better quality measures, since all the measures we have presented (including the best one Q_A^{EA}) suffer from counter-intuitive phenomenons.

In Chapter 5, we extended the work of Fraikin *et al.* to other trace maximization problems associated to the node-to-node similarity measure introduced by Blondel *et al.*, we characterized their critical points, and discussed their optimal values. We implemented algorithms dedicated to optimization on manifold described in [AMS08] to solve those problems. This chapter emphasized cases for which geometric optimization methods were more efficient than classical methods. Further work could be carried out[1] in geometric optimization in order to tackle a wider class of optimization problems, *e.g.* including feasible set with boundaries, piecewise discontinuous or piecewise differentiable objective functions, etc.

In Chapter 6, we considered restrictions of the feasible set of the trace maximization problem associated to the node-to-node similarity measure introduced by Blondel *et al.* to sets of low-rank matrices, in order to find a low-rank approximation of the similarity matrix associated to the node-to-node similarity measure introduced by Blondel *et al.*. We

[1] and actually is carried out by several of the Ph.D. students supervised by Pierre-Antoine Absil

first characterized the stationary points of the associated optimization problems and further considered iterative algorithms to find one of them. We analyzed the convergence properties of our algorithms, and finally compared our method in terms of speed and accuracy to the full rank algorithm proposed in [BGH$^+$04].

Bibliography

[AA03] L. Adamic and E. Adar. Friends and neighbors on the web. *Social Networks*, 25(3):211–230, 2003.

[AB02] R. Albert and A.-L. Barabási. Statistical mechanics of complex networks. *Rev. Mod. Phys.*, 74:47–97, June 2002.

[AG06] A. Azran and Z. Ghahramani. Spectral Methods for Automatic Multiscale Data Clustering. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 190–197, 2006.

[AJR03] P. Ahlgren, B. Jarneving, and R. Rousseau. Requirements for a cocitation similarity measure, with special reference to Pearson's correlation coefficient. *Journal of the American Society for Information Science and Technology*, 54(6):550–560, 2003.

[AMS08] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, New Jersey, January 2008.

[Arv02] W. Arveson. *A Short Course on Spectral Theory*. Graduate Texts in Mathematics. Springer, 2002.

[AS64] M. Abramowitz and I.A. Stegun. *Handbook of mathematical functions with formulas, graphs, and mathematical tables*. Number v. 55, no. 1972 in Applied mathematics series. U.S. Govt. Print. Off., 1964.

[Aub00] Jean-Pierre Aubin. *Applied functional analysis*. Pure and Applied Mathematics (New York). Wiley-Interscience,

New York, second edition, 2000. With exercises by Bernard Cornet and Jean-Michel Lasry, Translated from the French by Carole Labrousse.

[Bal85] A. T. Balaban. Applications of graph theory in chemistry. *Journal of Chemical Information and Computer Sciences*, (3):334–343, 1985.

[Baya] http://vlado.fmf.uni-lj.si/pub/networks/data/bio/foodweb/baydry.paj.

[Bayb] http://www.cbl.umces.edu/ atlss/fbay001.html.

[Bayc] http://www.cbl.umces.edu/ atlss/fbay704.html.

[BB32] J. Braun-Blanquet. *The Study of Plant Communities*. Plant Sociology. McGraw-Hil, New York, 1932.

[BBE89] S. P. Borgatti, J. Boyd, and M. Evertt. Iterated roles: Mathematics and application. *Social Networks*, 11(2):159 – 172, 1989.

[BE89] S. P. Borgatti and M. G. Everett. The class of all regular equivalences: Algebraic structure and computation. *Social Networks*, 11(1):65 – 88, 1989.

[BE92a] S. P. Borgatti and M. G. Everett. Notions of position in social network analysis. *Sociological Methodology*, 22:1–35, 1992.

[BE92b] S. P. Borgatti and M. G. Everett. Regular blockmodels of multiway, multimode matrices. *Social Networks*, 1992.

[BE93] S. P. Borgatti and M. G. Everett. Two algorithms for computing regular equivalence. *Social Networks*, 15(4):361 – 376, 1993.

[BE94] S. P. Borgatti and M. G. Everett. Ecological and perfect colorings. *Social Networks*, 16(1):43 – 55, 1994.

[BGBMN91] A. Bunse-Gerstner, R. Byers, V. Mehrmann, and N. K. Nichols. Numerical computation of an analytic singular value decomposition of a matrix valued function. *Num. Math.*, 60(1):1–39, 1991.

[BGH+04] V. D. Blondel, A. Gajardo, M. Heymans, P. Senellart, and P. Van Dooren. A measure of similarity between graph vertices: applications to synonym extraction and Web searching. *SIAM Review*, 46(4):647–666, 2004.

[BGLL08] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.

[Big93] N. Biggs. *Algebraic graph theory*. Cambridge mathematical library. Cambridge University Press, 1993.

[BL89] Tim Berners-Lee. Information Management: A Proposal. Technical report, CERN, March 1989.

[BLW99] N. L. Biggs, E. K. Lloyd, and R. J. Wilson. *Graph theory, 1736-1936*. Clarendon Press, 1999.

[BM76] J. A. Bondy and U. S. R. Murty. *Graph theory with applications*. American Elsevier Pub. Co., 1976.

[BND05] V. D. Blondel, L. Ninove, and P. Van Dooren. An affine eigenvalue problem on the nonnegative orthant. *Linear Algebra and its Applications*, 404:69 – 84, 2005.

[Bor89] S. P. Borgatti. *Regular Equivalence in Graphs, Hypergraphs, and Matrices*. University of California, Irvine, 1989.

[BP98] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1-7):107–117, 1998.

[Bun97] H. Bunke. On a relation between graph edit distance and maximum common subgraph. *Pattern Recogn. Lett.*, 18:689–694, August 1997.

[BW45] Vannevar Bush and Jingtao Wang. As we may think. *Atlantic Monthly*, 176:101–108, 1945.

[CABVD10] T Cason, P.-A. Absil, V. Blondel, and P. Van Dooren. A unified framework for affine graph similarity. Budapest,

Hungary, 2010. 19th International Symposium on Mathematical Theory of Networks and Systems (MTNS 2010).

[CAVD10] T Cason, P.-A. Absil, and P. Van Dooren. Low-rank approximation of graph similarity matrices. Pisa, Italy, 2010. 16th Conference of the International Linear Algebra Society (ILAS).

[CAVD11] T. P. Cason, P. A. Absil, and P. Van Dooren. Comparing two matrices by means of isometric projections. In P. Van Dooren, S. P. Bhattacharyya, R. H. Chan, V. Olshevsky, and A. Routray, editors, *Numerical Linear Algebra in Signals, Systems and Control*, volume 80 of *Lecture Notes in Electrical Engineering*, pages 77–93. Springer Netherlands, 2011. 10.1007/978-94-007-0602-6_4.

[CB10] K. Cooper and M. Barahona. Role-based similarity in directed networks. December 2010.

[CH69] Alan H. Cheetham and Joseph E. Hazel. Binary (presence-absence) similarity coefficients. *Journal of Paleontology*, 43(5):1130–1136, 1969.

[Cha85] G. Chartrand. *Introductory graph theory*. Popular Science Series. Dover, 1985.

[CLRS01] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT electrical engineering and computer science series. MIT Press, 2001.

[COT+07] Luciano Da F Costa, Osvaldo N Oliveira, Gonzalo Travieso, Francisco A Rodrigues, Paulino R Villas Boas, Lucas Antiqueira, Matheus P Viana, and Luis E C Da Rocha. Analyzing and modeling real-world phenomena with complex networks: A survey of applications. *Physics*, physics.so(3):103, 2007.

[CV07] G. Caldarelli and A. Vespignani. *Large Scale Structure and Dynamics of Complex Networks: From Information Technology to Finance and Natural Science*. Complex Systems and Interdisciplinary Science. World Scientific, 2007.

[Dah60] E. Dahl. Some measures of uniformity in vegetation analysis. *Ecology*, 41(4):805–808, 1960.

[DBF04] P. Doreian, V. Batagelj, and A. Ferligoj. *Generalized Blockmodeling*. Structural Analysis in the Social Sciences. Cambridge University Press, 2004.

[Die91] R. Diestel. *Graph Theory*, volume 173 of *Graduate texts in mathematics*. Springer, New York, 1991.

[DM02] S. N. Dorogovtsev and J. F. F. Mendes. Evolution of networks. In *Adv. Phys*, pages 1079–1187, 2002.

[Dor87] P. Doreian. Measuring regular equivalence in symmetric structures. *Social Networks*, 9(2):89 – 107, 1987.

[DS96] J. E. Dennis and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics, 1996.

[EB88] M. G. Everett and S. Borgatti. Calculating role similarities: An algorithm that helps determine the orbits of a graph. *Social Networks*, 10(1):77 – 91, 1988.

[EB91] M. G. Everett and S. P. Borgatti. Role coloring a graph. *Math. Soc. Sci.*, 21:183–188, 1991.

[EB93] M. G. Everett and S. P. Borgatti. An extension of regular colouring of graphs to digraphs, networks and hypergraphs. *Social Networks*, 15(3):237 – 254, 1993.

[EB94] M. G. Everett and S. P. Borgatti. Regular equivalence: General theory. *The Journal of Mathematical Sociology*, 19(1):29–52, 1994.

[EH10] Ernesto Estrada and Desmond Higham. Network properties revealed through matrix functions. *SIAM Review*, 52(4):696–714, 2010.

[Eul36] L. Euler. Solutio problematis ad geometriam situs pertinentis. *Commentarii academiae scientiarum Petropolitanae*, 8:128–140, 1736.

[Eve85] M. G. Everett. Role similarity and complexity in social networks. *Social Networks*, 7(4):353 – 359, 1985.

[Fau88] K. Faust. Comparison of methods for positional analysis: Structural and general equivalences. *Social Networks*, 10(4):313 – 341, 1988.

[FNVD06] C. Fraikin, Y. Nesterov, and P. Van Dooren. A gradient-type algorithm optimizing the coupling between matrices. *Proceedings of the 13-th ILAS conference in Amsterdam*, 2006.

[FNVD08] C. Fraikin, Y. Nesterov, and P. Van Dooren. Optimizing the coupling between two isometric projections of matrices. *SIAM J. Matrix Anal. Appl.*, 30(1):324–345, 2008.

[For96] S. Fortin. The graph isomorphism problem. Technical report, Department of Computing Science, University of Alberta, Canada, 1996.

[For10] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75 – 174, 2010.

[FP57] K. Fan and G. Pall. Imbedding conditions for Hermitian and normal matrices. *Canad. J. Math.*, 9:298–304, 1957.

[FVD07] C. Fraikin and P. Van Dooren. Graph matching with type constraints on nodes and edges. In Andreas Frommer, Michael W. Mahoney, and Daniel B. Szyld, editors, *Web Information Retrieval and Linear Algebra Algorithms*, number 07071 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany, 2007. Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI).

[Gan00] F.R. Gantmakher. *The Theory of Matrices*. Number v. 2. AMS Chelsea, 2000.

[Gel41] I. Gelfand. Normierte ringe. *Rec. Math. [Mat. Sbornik] N.S.*, pages 3–24, 1941.

[GN02] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, June 2002.

[HH05] P. Holme and M. Huss. Role-similarity based functional prediction in networked systems: application to the yeast proteome. *Journal of the Royal Society Interface*, (4):327–33, 2005.

[Hig08] Nicholas J. Higham. *Functions of Matrices: Theory and Computation*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2008.

[HJ90] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, 1990.

[HJ91] R. Horn and C. R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, New York, 1991.

[HM94] U. Helmke and J. B. Moore. *Optimization and Dynamical Systems*. Communications and Control Engineering Series. Springer-Verlag London Ltd., London, 1994. With a foreword by R. Brockett.

[HS03] M. Heymans and A. K. Singh. Deriving phylogenetic trees from the similarity analysis of metabolic pathways. *Bioinformatics*, (1):i138–i146, 2003.

[Jac01] P. Jaccard. Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bulletin del la Socit Vaudoise des Sciences Naturelles*, 37:547–579, 1901.

[JBLE03] Johnson, Borgatti, Luczkovich, and Everett. Network role analysis in the study of food webs: An application of regular role coloration. *Journal of Social Structure*, 2(3), 2003.

[JNRS10] Michel Journée, Yurii Nesterov, Peter Richtárik, and Rodolphe Sepulchre. Generalized power method for sparse principal component analysis. *J. Mach. Learn. Res.*, 11:517–553, March 2010.

[Joh55] S. Johnson. *Dictionary of the English language*. Arno Press, 1755.

[JW02] Glen Jeh and Jennifer Widom. Simrank: a measure of structural-context similarity. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on*

Knowledge discovery and data mining, pages 538–543, New York, NY, USA, 2002. ACM.

[Kle99] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, 1999.

[Kra86] Ulrich Krause. Perron's stability theorem for non-linear mappings. *Journal of Mathematical Economics*, 15(3):275 – 282, 1986.

[Kra01] U. Krause. Concave perron-frobenius theory and applications. *Nonlinear Analysis*, pages 1457–1466, 2001.

[LBJE02] J. J. Luczkovich, S. P. Borgattiz, J. C. Johnsony, and M. G. Everett. Defining and measuring trophic role similarity in food webs using regular equivalence. 2002.

[Ley08] L. Leydesdorff. On the normalization and visualization of author co-citation data: Salton's cosine versus the jaccard index. *Journal of the American Society for Information Science and Technology*, 59:77–85, 2008.

[LHN06] E. A. Leicht, P. Holme, and M. E. J. Newman. Vertex similarity in networks. *Physical Review*, (2), 2006.

[LM06] A.N. Langville and C.D. Meyer. *Google Page Rank and Beyond*. Princeton University Press, 2006.

[LW71] F. Lorrain and H. C. White. Structural equivalence of individuals in social networks. *Journal of Mathematical Sociology*, pages 49–80, 1971.

[Maz09] D.R. Mazur. *Combinatorics: A Guided Tour*. MAA textbooks. Mathematical Association of America, 2009.

[Mey00] C.D. Meyer. *Matrix analysis and applied linear algebra: solutions manual*. Number v. 2. Society for Industrial and Applied Mathematics, 2000.

[MGMR02] S. Melnik, H. Garcia-Molina, and E. Rahm. Similarity flooding: a versatile graph matching algorithm and its application to schema matching. In *Data Engineering, 2002. Proceedings. 18th International Conference on*, pages 117–128, 2002.

[Mun57] J. Munkres. Algorithms for the Assignment and Transportation Problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38, 1957.

[New03] M. E. J. Newman. The structure and function of complex networks. *SIAM REVIEW*, 45:167–256, 2003.

[New04] M. E. J. Newman. Detecting community structure in networks. *The European Physical Journal B - Condensed Matter and Complex Systems*, 38(2):321–330, March 2004.

[New06] M. E. J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006.

[Nin08] L. Ninove. *Dominant vectors of nonnegative matrices: application to information extraction in large graphs.* PhD thesis, Universit catholique de Louvain, 2008.

[OED11] Online etymology dictionary. Nov 2011.

[OW04] Donal B. O'Shea and Leslie C. Wilson. Limits of tangent spaces to real surfaces. *Amer. J. Math.*, 126(5):951–980, 2004.

[Par80] Beresford N. Parlett. *The symmetric eigenvalue problem.* Prentice-Hall Inc., Englewood Cliffs, N.J., 1980. Prentice-Hall Series in Computational Mathematics.

[Pol71] E. Polak. *Computational Methods in Optimization - A Unified Approach*, volume 77 of *Mathematics in science and engineering.* Academic Press, New York, 1971.

[POM09] Mason A. Porter, Jukka-Pekka Onnela, and Peter J. Mucha. Communities in networks, 2009.

[PS08] B. Parlett and G. Strang. Matrices with prescribed ritz values. *Linear Algebra and its Applications*, 428(7):1725 – 1739, 2008.

[RB06] J. Reichardt and S. Bornholdt. Statistical mechanics of community detection. *Phys Rev E Stat Nonlin Soft Matter Phys*, 74, 2006.

[RSM+02] E. Ravasz, A. L. Somera, D. A. Mongru, Z. N. Oltvai, and A.-L. Barabási. Hierarchical Organization of Modularity in Metabolic Networks. *Science*, 297:1551–1555, 2002.

[RW07] J. Reichardt and D. R. White. Role models for complex networks. *The European Physical Journal B - Condensed Matter and Complex Systems*, 60(2):217–224, 2007.

[Saa92] Y. Saad. *Numerical methods for large eigenvalue problems*. Algorithms and architectures for advanced scientific computing. Manchester University Press, 1992.

[Sai79] Lee Douglas Sailer. Structural equivalence: Meaning and definition, computation and application. *Social Networks*, 1(1):73 – 90, 19781979.

[Sal89] G. Salton. *Automatic text processing: the transformation, analysis, and retrieval of information by computer*. Computer Science Series. Addison-Wesley, 1989.

[Sch07] Satu Elisa Schaeffer. Graph clustering. *Computer Science Review*, 1(1):27–64, 2007.

[Sim43] George Gaylord Simpson. Mammals and the nature of continents. *American Journal of Science*, 241(1):1–31, 1943.

[SM83] Gerard Salton and Michael J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York u.a., 1983.

[SM97] J. Shi and J. Malik. Normalized cuts and image segmentation. In *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, CVPR '97, pages 731–, Washington, DC, USA, 1997. IEEE Computer Society.

[Sor48] T. Sorensen. A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on danish commons. *K. Dan. Vidensk. Selsk. Biol. Skr.*, 5:1–34, 1948.

[Spa85] I.G. Sparkes. *Dictionary of collective nouns and group terms*. Gale Research Co., 1985.

[SS08] S. Sorlin and C. Solnon. A parametric filtering algorithm for the graph isomorphism problem. *Constraints*, 13:518–537, 2008. 10.1007/s10601-008-9044-1.

[Ste73] G. W. Stewart. Error and perturbation bounds for subspaces associated with certain eigenvalue problems. *SIAM Rev.*, 15:727–764, 1973.

[Sti30] J. Stirling. *Methodus differentialis, sive tractatus de summatione et interpolatione serierum infinitarum*. Strahan, 1730.

[Str01] Steven H. Strogatz. Exploring complex networks. *Nature*, 410(6825):268–276, March 2001.

[STT77] I. Stewart, D. Tall, and D.O. Tall. *The Foundations of Mathematics*. Oxford science publications. Oxford University Press, 1977.

[Tre05] E. Tressler. Random walks on graphs and directed graphs. Course Notes, September 2005.

[Tyn60] W. Tyndale. *The 1560 Geneva Bible*. Bible Museum, 1560.

[UCT07] Final Report on the disturbances of 4 November 2006 with recommendations on standards, procedures, tools and rules - UCTE. *Oil, Gas & Energy Law (OGEL)*, 5(1), 2007.

[WBB76] Harrison C. White, Scott A. Boorman, and Ronald L. Breiger. Social structure from multiple networks. *American Journal of Sociology*, 81:730–780, 1976.

[WF94] S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications*. Structural Analysis in the Social Sciences. Cambridge University Press, 1994.

[Win88] Christopher Winship. Thoughts about roles and relations: An old document revisited. *Social Networks*, 1988.

[WM84] Christopher Winship and Michael Mandel. *Roles and Positions: A Critique and Extension of the Blockmodeling Approach*. Jossey-Bass, 1984.

[WR83] D. R. White and K. P. Reitz. Graph and semigroup homomorphisms on networks of relations. 1983.

[Zag05] L. Zager. *Graph Similarity and Matching*. PhD thesis, Massachusetts Institute of Technology, may 2005.

[ZHS09] Peixiang Zhao, Jiawei Han, and Yizhou Sun. P-rank: a comprehensive structural similarity measure over information networks. In *CIKM*, pages 553–562. ACM, 2009.

[ZLZ09] Tao Zhou, Linyuan Lu, and Yi-Cheng Zhang. Predicting missing links via local information. Technical Report arXiv:0901.0553, Jan 2009.

[ZTW+09] Z. Zeng, A. K. H. Tung, J. Wang, J. Feng, and L. Zhou. Comparing stars: on approximating graph edit distance. *Proc. VLDB Endow.*, 2:25–36, August 2009.