Università Ca' Foscari di Venezia

Dipartimento di Scienze Ambientali, Informatica e Statistica
Dottorato di Ricerca in Informatica

Ph.D. Thesis: Cycle 27

Algorithms for Graph Compression: Theory and Experiments

Farshad Nourbakhsh

Supervisor
Marcello Pelillo

PhD Coordinator
Riccardo Focardi

September, 2014

Author's Web Page: http://www.dais.unive.it/~farshad

Author's e-mail: farshad@dais.unive.it

Author's address:

Dipartimento di Informatica
Università Ca' Foscari di Venezia
Via Torino, 155
30172 Venezia Mestre – Italia
tel. +39 041 2348411
fax. +39 041 2348419
web: http://www.dais.unive.it

Università
Ca'Foscari
Venezia

Scuola Dottorale di Ateneo
Graduate School

Dottorato di ricerca
in Informatica
Ciclo XXVII
Anno di discussione 2014

Algorithms for Graph Compression: Theory and Experiments

SETTORE SCIENTIFICO DISCIPLINARE DI AFFERENZA: INF/01
Tesi di Dottorato di Farshad Nourbakhsh, matricola 955963

Coordinatore del Dottorato	Tutore del Dottorando
Prof. Riccardo Focardi	Prof. Marcello Pelillo

To the future readers of this thesis, for their patience.

Abstract

Graphs and networks are everywhere, from social networks to the World Wide Web. Since the last decade, massive graphs has become the center attention of an intense research activity, both industrial and academic research centers. So these graphs are a new challenge for the storage, geometric visual representation and retrieve information. These information retrieving need an efficient techniques to compress the graph.

Compressing data consists in changing its representation in a way to require fewer bits. Depending on the reversibility of this encoding process we might have a lossy or lossless compression.

In the first part of thesis, we have addressed this problem by a two steps clustering strategy and the solution takes advantage of the strong notion of edge density Regularity introduced by Endre Szemerédi.

In the second chapter, we address the problem of encoding a graph of order n into a graph of order $\mathsf{k} < \mathsf{n}$ in a way to minimize reconstruction error. This encoding is characterized in terms of a particular factorization of the adjacency matrix of the original graph. The factorization is determined as the solution of a discrete optimization problem, which is for convenience relaxed into a continuous, but equivalent, one. Our formulation does not require to have the full graph, but it can factorize the graph also in the presence of partial information. We propose a multiplicative update rule for the optimization task resembling the ones introduced for non-negative matrix factorization, and convergence properties are proven. Experiments are conducted to assess the effectiveness of the proposed approach.

Our main contribution are summarized as:

i) we link matrix factorization with graph compression by proposing a factorization that can be used to reduce the order of a graph and can be employed also

in the presence of incomplete observations. We show that the same technique can be used to compress a kernel, by retaining a kernel as the reduced representation; Moreover, we consider a general setting, where the observations of the original graph/kernel are incomplete;

ii) we cast the discrete problem of finding the best factorization into a continuous optimization problem for which we formally prove the equivalence between the discrete and continuous formulations;

iii) we provide a novel algorithm to approximately find the proposed factorization, which resembles the NMF algorithm in [48] (under ℓ_2 divergence) and the Baum-Eagon dynamics [5]. Additionally, we formally prove convergence properties for our algorithm and we believe that this theoretical contribution can be helpful for devising other factorization algorithms working on the domain of stochastic matrices (rather than simply non-negative matrices);

iv) finally, we establish a relation between clustering and our graph compression model and show that existing clustering approaches in the literature can be regarded as particular, *constrained* variants of our matrix factorization.

Acknowledgments

I would like to express my gratitude to my supervisor Professor Marcello Pelillo, for his scientific guidance and support during my Ph.D. studies. I am grateful to my co-authors with particular reference to Samuel Rota Bulò for his helpfulness and scientific support. I would like to thank Dr. Hannu Reittu for his scientific help.

Thanks to my external reviewers Prof. Francisco Escolano and Prof. Carlo Sansone for the time they spent on carefully reading the thesis and for their useful comments and suggestions.

Thanks to the Department of Computer Science of the University CaFoscari of Venice for financing my studies with a three years grant, and thanks to all the people working there like Professor Riccardo Focardi,Teresa Scantamburlo, Francesco Lettich, Gian-Luca Rossi, Nicola Rebagliati, Luca Rossi, member of CVPR groupd and etc.

Special thanks to my ex professors and colleagues whom I have enjoyed my past and I'm in this stage now because of their support, starting from Professor A.G Ramakrishnan, Professor Angel Sappa, Professor Ernest Valveny, Peeta Basa Pati, Thotreingam Kasar, Debprakash Patnaik, Naila Murray, David Vázquez Bermúdez, Antonio Clavelli, Montse Culleré, Nima Hatami, Nima Dashtban and others who I can't remember now.

Special thanks go also to my family and my only sister for their unconditional support, and special thanks to Michele Destro who helped and assisted me to have an easier life in Mestre. Finally, many thanks to my close friend Dr Mohammad Rouhani who used to listen to me since last six years.

Farshad Nourbakhsh

Contents

I Introduction **1**

1 Motivations **3**

1.0.1 Graph Theory in Math and Computer Science 3

1.0.2 Extremal Graph Theory 4

1.0.3 State of the Art 5

1.0.4 General Scientific Background 7

II Preliminary Definitions **9**

2 Notations **11**

2.1 Szemerédis Regularity Lemma (SRL) 11

2.2 Matrix Factorization Approach (MFA) 12

III The Szemerédi's Regularity Lemma and Graph Compression **15**

3 The Szemerédi's Regularity Lemma **17**

3.1 Extremal Graph Theory 17

3.1.1 What is the Extremal Graph Theory? 17

3.2 The Regularity Lemma 19

3.2.1 The Lemma 19

3.2.2 The role of exceptional pairs 21

3.3 Checking regularity 21

3.3.1 The reduction mechanism 22

3.4 Extension to the Regularity Lemma 25

3.5 Algorithms for the Regularity Lemma 27

3.6 The first Algorithm 28

3.7 Checking regularity in a bipartite graph 29

3.7.1 The algorithm 32

3.8 Algorithm Implementation 35

3.8.1 Connection between Pairwise Clustering and Regularity Lemma 35
3.8.2 Reduced Graph and Key Lemma 36
3.8.3 The Proposed Method 37
3.9 Experiments 39
3.9.1 Dominant Sets 39
3.9.2 Experimental Results 42
3.10 Discussion and future work 44

IV A Matrix Factorization Approach to Graph Compression with Partial Information 45

4 A Matrix Factorization and Graph Compression 47
4.1 Introduction 47
4.2 Preliminaries 51
4.3 Matrix factorization for graph compression 51
4.4 A tight relaxation of the graph compression objective 54
4.5 Graph compression algorithm 57
4.5.1 Update rule for **R**. 57
4.5.2 Update rule for **Y**. 58
4.5.3 Summary of the algorithm. 62
4.5.4 Graph reconstruction with incomplete observations 62
4.6 Clustering as graph compression 64
4.7 Experiments 65
4.7.1 Synthetic experiments 65
4.7.2 Real-world experiments 68
4.7.3 Clustering 71
4.8 Discussion and future work 73

V Appendix 77

A Determining the Number of Dominant-Set Clusters and Community Detection 79
A.0.1 Dominant set based determination of number of clusters . . . 80
A.1 Proposed Method 84
A.1.1 Detect Number of Cliques 84

A.1.2 Experimental Results: 86
A.2 Discussion and future work 89

Conclusions **91**
C.1 Contributions 91
C.2 Impact and Future Works 93

Bibliography **95**

List of Figures

3.1 This experiment shows the classification result of two circles with Dominantset. 42

4.1 Graph compression 63
4.2 Results obtained for different types of synthetic experiments. See Section 4.7.1 for details. 74
4.3 Evolution of the objective $f(\mathbf{Y}, \mathbf{R})$ during the execution of our graph compression algorithm. We report the evolution of 6 runs with different values of k on random graphs of order $\mathsf{n} = 400$ that can be potentially compressed to order k (see, Section 4.7.1 for details about the graph generation procedure). Markers have been sparsified for a better visualization. 75
4.4 Qualitative results of K-PCA for Iris and eColi dataset. See Section 4.7.2 for details. 76
4.5 Clustering results obtained on different machine learning data-sets by our approach and BE [72]. For details see Section 4.7.3. 76

A.1 Two steps of proposed method from left to right column. 87
A.2 Graph transduction method is applied to show the final result 88
A.3 Shows the performance of proposed method on noisy data 89

List of Tables

2.1 Table of Notations 13

3.1 Results obtained on the UCI benchmark datasets. Each row represents a dataset, while the columns represent (left to right): the number of elements in the corresponding dataset, the classification accuracy obtained by the proposed two-phase strategy and that obtained using the plain dominant-set algorithm, respectively, the size of the reduced graphs, the compression rate and the number of classes. 43

3.2 Result obtained on Breast dataset of UCI benchmark. Ech row represents a ϵ, accuracy and compression rate 44

4.1 Experimental results on Erdós-Renyi random graphs for the task of predicting the state of non-observed edges. We consider three different scenarios: purely random graph, unstructured graph and structured graph. We report the average AUC value (and standard deviation) obtained with 10 graph samples per setting. 68

4.2 Clustering results obtained by Kernel K-means and Spectral Clustering (Normalized Cut) on different data-sets, after having compressed the data (graph/kernel) at different rates. 69

4.3 Experimental results on real-world data-sets for the task of predicting the state of non-observed edges. We report the average AUC (and standard deviation) obtained with 10 random training/test set (80%/20%) splits. 71

A.1 The result of proposed method on different data-sets with compression to SC and KM methods as clustering 90

I

Introduction

1

Motivations

TAKING THE FIRST
FOOTSTEP with a good thought,
the second with a good word, and
the third with a good deed, I
entered paradise.

Zarathustra (c.628 - c.551)

1.0.1 Graph Theory in Math and Computer Science

In mathematics and computer science, graph theory is the study of graphs. Mathematical structures are used to model pairwise relations between objects and are called graphs. A graph contains vertices or nodes and lines called edges that connect them. A graph may be undirected, directed, weighted or unweighted. Graphs are one of the prime objects of study in discrete mathematics.

In physical, biological, social and, information systems, graphs can be used to model many types of relations and processes. So, many practical problems can be represented by graphs.

As far as, our interest is computer science, it can be observed that graphs are used to represent networks of communication, data organization, computational devices and the flow of computation. The link structure of a website, biology, computer chip design, social network and many other fields can be given as examples.

With this short overview, it is clear that to develop methods to handle graphs, is in the main attention of computer science society. By transforming graph in this

major, we are involved in-memory manipulation of graphs as a main issue. So one of the hottest topics in this field is **graph compression** in structured data. Some well known domains of application of graph theory are: linguistics, the study of molecules in chemistry and physics (e.g., the topology of atoms) and sociology (e.g., rumor spreading). Another important area of application is concerned with social networks like friendship graphs and collaboration graphs.

As mentioned before, Graph compression plays a fundamental/crucial role to handle huge amounts of structured data, as far as, our hardware devices are not strong enough to obtain a meaningful result in appropriate time and memory space.

Graph compression can be used as a technique for a preprocessing step or directly, based on the method of compression. Among the vast amount of techniques in this field, Extremal Graph theory provides a tool to deal with Graph Compression. Moreover, among the vast amount of literature in this field, we present a coarse review of compression methods in this chapter.

1.0.2 Extremal Graph Theory

Let us start with a question How much of something can we have, given a certain constraint? . This is a basic question for any extremal problem. Indeed the above question can be applied in philosophy and science in the same manner.

In 1941, the Hungarian mathematician P. Turán provided an answer the question of how many edges a vertex graph can have, given that it has no k clique. That can be represented as the most basic result of extremal graph theory (This graph is now known as a Turán graph). T. S. Motzkin and E. G. Straus were able to provide a proof for Turáns theorem based on the clique number of a graph.

The Turan's work was developed to a branch of graph theory named as extremal graph theory by researchers like P. Erdös, B. Bollobás, M. Simonovits and E. Szemerédi in early 20th-century. In a core definition, extremal graph theory explores how the intrinsic structure of graphs ensures certain types of properties (e.g., cliques, colorings and spanning sub-graphs) under appropriate conditions (e.g., edge density and minimum degree).

Szemerédis regularity lemma is certainly one of the best known among extremal graph theory. Szemerédi shows that the monochromatic arithmetic progression occurs in each color class that is not trivially sparse. Hence, it explains that every graph can be partitioned into regular pairs that are random-like bipartite graphs and a few leftover edges. Szemerédis result was introduced as a tool for the proof of the Erdös-Turán conjecture on arithmetic progressions in dense sets of integers.

Another way to illustrate the connection between extremal graph theory and Szemerédi Lemma is, given a graph G, an invariant , a property P and a value m, the least value of m such that $(G)m$ and P is verified. The graphs in which $(G) = m$ are called extremal for the property P . In this scenario, the Szemerédi's Regularity Lemma searches for the conditions which assure the existence of a graph partition that respects a notion of edge density regularity. More than this, Szemerédi demonstrates that every graph, with a sufficiently high number of vertices, has a Regular Partition. As well as for the importance of the results, the value of the Szemerédi's Regularity Lemma is also due to its utility as demonstration tool.

1.0.3 State of the Art

A problem linked to graph compression that has focused the attention of researchers is the network and sociometric literature for the last few decades. In general, researchers have attacked to the graph compression in two different perspectives: the structural way, to produce a simplified graph, or the information theoric way that makes a graph with fewer bits memory capacity. Paper [[79] and [14]] are an example for structural and information theoric category respectively.

This [79] is a method based on graph structure (or link structure) that is very efficient to improve the performance of search engine and the work presented in [14] is,it is an information theory approach to compress a graph with respect into structural entropy by applying two stage encoding.

In [42] the authors have proposed a method based on link compression by applying web-graph algorithm [1]. In this way, they produce smaller representation of the graph that has less number of edges with more similarity between adjacency nodes.

In [85] the authors have presented two series of methods for graph simplification by introducing the notion of supernode and superedge. Their method is a trade off between minimizing approximation error and maximizing the compression in

the same time by defining a distance measure. They reported that weighted graph method is suitable for a compression with a little loss but the second method (generalized weighted graph compression) is better to be used as a preprocessing step.

In [28] the authors have presented two notions which are importance of node and similarity to handle topological compression. The first notion is based on ranking web search. Additionally, two measures have been defined to detect nodes to compress the graph. These measure are degree and shortest path. In the second notion case, a similarity measure or relation is applied to combine similar vertices that leads to substantially less cluttered graphs at a minimum information loosing.

In [24] the authors consider the compression problem as minimizing the order of the cliques. They have shown that the existence of a partition with a small order in enough dense graph. Their compression algorithm is based on the binary search to find cliques with specific order. After partitioning edges of the graph to a collection of edge-disjoint cliques,they replace them with new edge and vertices to obtain the compressed graph.

The work proposed in [44] suggests a method based on the edge-length and terminal vertices as the notion of compressing for undirected graph. In their method, they compute shortest paths between every pairs of terminals and remove every non terminals and edges that do not participate to any terminal to terminal and finally they set all not terminal vertices to the defined length.

Graph compression method has been used in other domain in computer science like graph mining, [40]. Compression-based graph miners discovers those sub graphs that maximize the amount of compression that a particular substructure provides a graph. In their paper, they have investigated the difference between compression based graph mining and frequency graph mining in different data-sets.

The work proposed in [33] suggests a method proposed a method to compress a graph based on a right number of group or compression rate that is obtained by Rissanens Minimum Description Length (MDL) criterion method (see [32]). After that, an expectation maximization algorithm based on maximum likelihood method with Bernoulli input data is applied to partition the algorithm. This method reaches a local minimum sequentially reallocating elements to decrease the target function,

until stopping criteria.

1.0.4 General Scientific Background

In this section we would like to show the link between chapters of this thesis and to give an overview of the concept of graph compression.

In the chapter III, Szemerédi's Regularity Lemma (SRL) is used for graph compression and in the chapter IV , an iterative optimization method is introduced based on a Matrix Factorization Approach (MFA) to compress a graph.

SRL is a fundamental combinatorical result with a wide range of applications in mathematics [32]. However, it has not been generally recognized as a principle with wide practical relevance. SRL introduces the term of *regularity* that refers in this context to a certain kind of "pseudo-randomness up to ϵ-accuracy". The intuitive idea is that the regular decomposition presents a structure, which can be seen as a kind of compressed version of original graph. Moreover, taking the regular structure and "blowing it up" by cloning its nodes multiple times and drawing truly random edges with probabilities prescribed by the structure, one obtains a graph that is, in a formally specified sense, very similar to the original graph [26]. This important result,sometimes called the "Key Lemma", in a sense proves the relevance of a regular decomposition. Note also that a regular decomposition can be interpreted as a stochastic model of the original graph. The importance of regular decomposition is getting evidence from many directions. Particularly remarkable in this sense are the results of Alon et al. [3], who have shown that the (effective) testability of a graph property is essentially equivalent to that this property is based on a regular decomposition of the graph, which is applied as main core for our proposed graph compression. Finally, we can say that SRL condenses a very general principle of separating structure and randomness as well as compression by defining the notion of regularity.

In the chapter IV , we compress the graph in a way to minimize reconstruction error. This encoding is characterized in terms of a particular factorization of the adjacency matrix of the original graph. Instead of cambinatorial optimization in SRL method, the factorization is determined as the solution of a discrete optimization problem. We propose a multiplicative update rule for the optimization task to compress the graph by grouping them to the similar categories.

In summary the main difference of proposed methods are, finding regular decomposition can be considered as one type of graph clustering, but it differs from clustering in the usual sense: in classical graph clustering, vertices that are well connected with each other are grouped together like our proposed method in forth chapter, whereas in a regular decomposition, members of a group have stochastically similar relations to all other groups. In fact, the internal edge structure within a group is not considered at all in SRL.

II

Preliminary Definitions

2

Notations

Your task is not to seek for love, but merely to seek and find all the barriers within yourself that you have built against it.

Jalal Ad-Din Rumi quotes

2.1 Szemerédis Regularity Lemma (SRL)

Szemerédi's main result is widely used in the context of dense extremal graph theory. As usual, a graph is said to be dense if its number of edges is about quadratic in the number of vertices. In particular, if we consider a bipartite graph $G = (V, E)$, where V is the vertex set and $E \subseteq V \times V$, the edge density of G is the number

$$\mathbf{d}(G) = \frac{||G||}{\binom{|G|}{2}} \tag{2.1}$$

where $||G|| = |E|$ and $|G| = |V|$.

As we can see, the edge density of a graph is the proportion between the number of edges that exist in the graph and the number of possible edges in a graph with $|V|$ vertices (not considering self-loops).

If we consider $X, Y \subset V$, $X \bigcap Y = \emptyset$, the edge density can be defined as

$$\mathbf{d}(X, Y) = \frac{e(X, Y)}{|X|.|Y|} \tag{2.2}$$

where

$$\mathbf{e}(X, Y) = |(x, y) \in E | x \in X \wedge y \in Y| \tag{2.3}$$

The Szemerédi's Lemma deals with vertex subsets that shows a sort of regular behavior. In particular, Szemerédi introduced the definition of regular pair to describe that pair of sufficiently large subsets that are characterized by a quite uniform edge distribution.

Definition 1. *(Regularity Condition). Let $\epsilon > 0$. Given a graph $G = (V, E)$ and two disjoint vertex sets $A \subset V$, $B \subset V$, we say that the pair A, B is ϵ-regular if for every $X \subset A$ and $Y \subset B$ satisfying*

$$|X| > \epsilon|A| \text{ and } |Y| > \epsilon|B| \tag{2.4}$$

we have

$$|d(X, Y) - d(A, B)| < \epsilon \tag{2.5}$$

If we are considering a graph G with a vertex set V partitioned into pairwise disjoint classes $C_0, C_1, ..., C_k$, this partition is said *equitable* if all the classes $C_i, 1 \leq i \leq k$ have the same cardinality. The so called *exceptional* set C_0 has usually a only technical purpose: it enables the construction of same size subsets and it can also be empty, if necessary.

If a graph G has been partitioned in $k + 1$ subsets and they allow the presence of a sufficient number of ϵ-regular pairs, we may speak of a regular partition of G. More precisely:

Definition 2. *(ϵ-regular Partition). Let $G = (V, E)$ a graph and $C_0, C_1, ..., C_k$ a partition of its vertex set. The partition is said ϵ-regular*
if all the following conditions hold.

- $|C_0 < \epsilon|V|$
- $|C_1| = |C_2| = ... = |C_k|$
- *all but at most ϵk^2 of the pairs (C_i, C_j) are $\epsilon - regular$.*

2.2 Matrix Factorization Approach (MFA)

We present here the notation and basic definitions adopted throughout the thesis.

Table 2.1: Table of Notations

General

$\boldsymbol{x}, \boldsymbol{y}$	$\triangleq$	*vectors* with bold lowercase letters
$\mathbf{A}, \mathbf{B}$	$\triangleq$	*matrices* with uppercase typewitter-style letters
$\mathcal{X}, \mathcal{Y}$	$\triangleq$	*sets* with calligraphic-style uppercase letters
$\mathbf{A}_{ij}$	$\triangleq$	The (i,j)th element of matrix $\mathbf{A}$
v_i	$\triangleq$	The ith element of a vector $\boldsymbol{v}$
i, j, k, h	$\triangleq$	*indices* with lowercase letters
n, m	$\triangleq$	*constants* with lowercase serif-style letters
$\mathbb{1}_P$	$\triangleq$	The indicator function giving 1 or 0
$\mathbb{R}$	$\triangleq$	The sets of real numbers
$\mathbb{R}_+$	$\triangleq$	The sets of non-negative real numbers
$[\mathsf{n}]$	$\triangleq$	The set $\{1, \ldots, \mathsf{n}\}$

Linear Algebra

x^+	$\triangleq$	The pseudo-inverse of a scalar $x \in \mathbb{R}$
x^+	$=$	$\begin{cases} 1/\mathrm{x}, & \text{if } x \neq 0 \\ 0, & \text{otherwise} \end{cases}$
$\mathbf{I}$	$\triangleq$	The *identity* matrix
$\mathbf{1}_\mathsf{k}$	$\triangleq$	The vector of all 1s of size k
$\mathbf{E}$	$\triangleq$	The matrix of all 1s
$\mathbf{A}$ and $\mathbf{B}$	$\triangleq$	$\mathsf{n} \times \mathsf{m}$ matrix
$\mathbf{A}^\top$	$\triangleq$	The *transposition* of $\mathbf{A}$
$\mathbf{A}^{-1}$	$\triangleq$	The *inverse* of a square matrix $\mathbf{A}$
$\mathcal{S} = \{\mathbf{X} \in \mathbb{R}_+^{\mathsf{k}\times\mathsf{n}} : \mathbf{X}^\top \mathbf{1}_\mathsf{k} = \mathbf{1}_\mathsf{n}\}$	$\triangleq$	The set of *left-stochastic* $\mathsf{k} \times \mathsf{n}$ matrices
$\mathcal{S}_{01} = \mathcal{S} \cap \{0,1\}^{\mathsf{k}\times\mathsf{n}}$	$\triangleq$	left-stochastic binary matrices
$\mathbf{A} \succcurlyeq \mathbf{B}$	$\triangleq$	$(\mathbf{A} - \mathbf{B})$ is positive semidefinite $\sum_{ij\in[\mathsf{n}]}(\mathbf{A}_{ij} - \mathbf{B}_{ij})x_i x_j \geq 0$ for all $\boldsymbol{x} \in \mathbb{R}^\mathsf{n}$
$\mathrm{diag}\mathbf{A}$	$\triangleq$	column-vector holding the diagonal of $\mathbf{A}$
$\mathrm{tr}(\mathbf{A}) = \sum_{i=1}^{\min\{\mathsf{n},\mathsf{m}\}} a_{ii}$	$\triangleq$	The *trace* of $\mathbf{A}$
$\mathrm{rk}(\mathbf{A})$	$\triangleq$	The *rank* of $\mathbf{A}$
Λ_A	$\triangleq$	$\mathsf{n} \times \mathsf{n}$ diagonal matrix
$(\mathbf{A})_+$	$\triangleq$	the matrix $\mathbf{A}$ with its negative entries set to 0
$\|\mathbf{A}\|_F = \sqrt{\mathrm{tr}(\mathbf{A}^\top \mathbf{A})}$	$\triangleq$	The *Frobenius* norm of $\mathbf{A}$
$\|\mathbf{A}\|_\mathcal{O} = \sqrt{\sum_{(i,j)\in\mathcal{O}} \mathbf{A}_{ij}^2}$	$\triangleq$	The *Frobenious* norm of $\mathbf{A}$ restricted to $\mathcal{O}$
$\mathrm{vec}(\mathbf{A}) = (\boldsymbol{a}_1^\top, \boldsymbol{a}_2^\top, \ldots, \boldsymbol{a}_n^\top)^\top$	$\triangleq$	The column-wise *vectorization* of $\mathbf{A}$
$\boldsymbol{a}_i \in \mathbb{R}^\mathsf{k}$	$\triangleq$	the ith column of $\mathbf{A}$

III

The Szemerédi's Regularity Lemma and Graph Compression

3

The Szemerédi's Regularity Lemma

Drink wine. This is life eternal.
This is all that youth will give you.
It is the season for wine, roses and
drunken friends. Be happy for this
moment. This moment is your life

Omar Khayyam

3.1 Extremal Graph Theory

Extremal Graph Theory is a branch of the mathematical field of Graph Theory on the studies of Paul Turán and Paul Erdos.

Extremal Graph Theory had its huge spread in the seventies with the work of a large group of mathematicians among which we can find Béla Bollobás, Endre Szemerédi, Miklós Simonovits and Vojtéch Rodl. In the last three decades, Extremal Graph Theory was a field so rich of results that it remains up to this days one of the most active fields of mathematics. Let's now see some more technical definitions.

3.1.1 What is the Extremal Graph Theory?

Extremal Graph Theory concerns the relations between graph invariants. In particular, Extremal Graph Theory studies how the intrinsic structure of graphs ensures certain types of properties (*e.g.*, cliques, colourings and spanning sub-graphs) under appropriate conditions (*e.g.*, edge density and minimum degree).

A graph invariant is a map that takes graphs as arguments and assigns equal values to isomorphic graphs. There is a wide range of possible graph invariants: the

simplest are the number of vertices (called order) and the number of edges. Other more complex graph invariants are connectivity, minimum degree, maximum degree, chromatic number and diameter.

Extremal Graph Theory is based on the concept of *extremal* graph. Béla Bollobás, in his book [8], gave the following definition of Extremal Graph

> *[. . .] given a property $\mathscr{P}$ and an invariant for a class $\mathscr{H}$ of graphs, we wish to determine the least value m for which every graph G in $\mathscr{H}$ with $\mu(G) > m$ has property $\mathscr{P}$. Those graph G in $\mathscr{H}$ without the property $\mathscr{P}$ graph for the problem.*

In other words, the extremal graphs are on the boundary between the set of graphs showing a particular property and the set of graphs for which this property is not valid. As said before an invariant maps graphs into the integers set $\mathbb{N}$. So an extremal G is a graph evaluated in the maximum $m \in \mathbb{N}$ such that for any $n \in \mathbb{N}, n > m$, the property is true.

Bollobás [8] suggest the following simple examples to clarify the notion of extremal.

- Consider a graph $G = (V, E)$ where, as usual, V is the vertex set and $E \subseteq V \times V$ is the edge set. The property analyzed is the one which states that every graph of order $|V| = n$ and size $|E| \geq n$ contains a cycle. In this case, the extremal graphs for the considered property are all the trees of order n.
- Consider a graph $G = (V, E)$ of order $|V| = 2u$, $u \in \mathbb{N}$. If the minimum degree is at least $u + 1$, then G contains a triangle. In this case, the extremal of the problem is the graphs K_{uu}, *i.e.* the complete bipartite graph of order u.

Despite the simplicity of the previous examples, the reader should easily imagine how the study of extremal graph could have led to interesting results.

In a more general form, the Main extremal problem is (see [8]):given a graph $F = (V, E)$ determine $e(n, F)$, the maximum number of edges in a graph of order n not containing F as a sub-graph or, observing the problem from another point of view (see [19]), searching exactly which edge density is needed to force a given sub-graph.

R. Diestel, in his book [19], presents a deep analysis of the Extremal Graph Theory problems. An Extremal Problem is characterized by searching for a graph

G to have a chosen graph H as sub-graph or one of its topological copy. A topological copy of a graph H is a graph obtained from H by replacing some edges by pairwise disjoint paths ([[3], [19]]).

In particular, Diestel distinguishes two main groups of Extremal Problems [19]:

- In the first category there are all the problems in which we are interested in making sure, by global assumptions, that a graph G contains a given graph H (or a topological copy) as sub-graph.
- In the second class, on the other hand, there are all the problems in which we are asking what global assumptions might imply the existence of a given graph H as sub-graph.

In this context, the Szemerédi's Regularity Lemma is one of the most powerful mathematical tool to demonstrate most of the extremal existence theorems.

Moreover, in the last ten years, many researchers have been involved in the study of the algorithmic aspects of the lemma, which has led to an algorithmic reformulation of many existence extremal problems.

3.2 The Regularity Lemma

The Regularity Lemma was developed in 1976, Szemerédi's studies about the Ramsey properties of arithmetic progression [81]. Indeed, even though this thesis is related to the graph theoretical utilization of the theorem, the lemma was born and is still used to solve problem in the field of number theory and combinatorial geometry.

As far as graph theory is concerned, the Szemerédi's Lemma is considered one of the most important tools in the proof of other technical results.

In its first formulation, the Lemma was only a technical step during the proof of a major result and it was only related to bipartite graph.

3.2.1 The Lemma

In his paper [82], Szemerédi proposed a result that is particular appealing for its first-sight simplicity and for its generality. His main result, indeed, can be applied to every graph, with the only requirement of a sufficiently large number of vertices. The partition described in the Szemerédi's Lemma highlights the connection between the

field of extremal graph theory and the one of *Random Graphs*. A random graph is defined as a graph in which every edge is independently generated with probability p during a random process

In particular, as Komlos and Simonovits [43] say, a ϵ-regular partitioned graph can be approximated by a generalized random graph. According to the authors, given a $r \times r$ symmetric matrix (p_{ij}) with $0 \leq p_{ij} \leq 1$, and positive integers $n_1, n_2, \cdots, n_r$ a generalized random graph R_n for $n = n_1 + n_2 + \cdots + n_r$ is obtained partitioning n vertices into classes C_i of size n_i and joining the vertices $x \in V_i$, $y \in V_j$ with probability p_{ij} , independently for all pairs x, y.

The Lemma states that a graph can be partitioned in such a way that the number of edges inside each subset is very small and the edges running between two subsets are about equally distributed. From a probabilistic point of view, a graph that admit a Szemerédi's partition (*i.e.* a graph with a sufficiently large vertex set) can be approximated by a graph with uniform distribution generated edge set.

Theorem 1. *(Szemerédi's Lemma). For every positive real ϵ and for every positive integer m, there are positive integers N and M with the following property: for every graph G with $|V| \geq N$ there is an ϵ-regular partition of G into $k+1$ classes such that $m \leq k \leq M$*

The Lemma 1, as stated in [81], says that if we fix in the first place a density bound ϵ, and a lower bound of the number of subsets in the partition, we can also find

- a lower bound of the required number of vertices, *i.e.* to the order of the graph
- an upper bound of the partition cardinality.

This also means that the values N and M deeply depend on the constants ϵ and m. Indeed many authors [[19], [43], [3]]) use to stress this dependence writing the bounds as functions of ϵ and m: $N(\epsilon, m)$ and $M(\epsilon, m)$.

The lower bound of the number of classes, m, is necessary to make the classes C_i sufficiently small. In this way, we can assure the presence of a very small number of intra-class edges.

Finally, it should be noted that a singleton partition is *epsilon*-regular for every value of ϵ and m.

There also exists an alternative form of the Lemma 1. In this case, the requirement about the size of the vertex-set in the partition is relaxed. Indeed, in this case, the sizes of two vertex-sets can differ by one.

Theorem 2. *(Szemerédi's Lemma: alternative form). For every positive* $\epsilon > 0$ *there exists an* $M(\epsilon)$ *such that the vertex set of any n-graph G can be partitioned into k sets* $C_1, C_2, ...C_k$ *for some* $k \leq M(\epsilon)$*, so that*

- $|C_i| \leq \epsilon n$ *for every i*
- $||C_i| - |C_j|| \leq 1$ *for all i, j*
- (C_i, C_j) *is* ϵ*-regular in G for all but at most* ϵk^2 *pairs* (i, j)

In this case, the first condition on the result assures that the classes C_i are sufficiently small with respect to ϵ. In this way, it is possible to relax the dependence of M on m.

3.2.2 The role of exceptional pairs

Theorems 1 and 2 don't require that all the pairs in a partition have to be regular because at most ϵk^2 pairs to be exceptional. Following the Szemerédi [82] point, many researchers studied if the Lemma could be strengthened, avoiding the presence of irregular pairs.

Forcing the lemma in that way would have invalidate the result generality. Indeed, Alon et al. [3] show a counterexample

> *[...] showing that irregular pairs are necessary is a bipartite graph with vertex classes* $A = a_1, ..., a_n$ *and* $B = b_1, ..., b_n$ *in which* (a_i, b_j) *is an edge iff* $i \leq j$

3.3 Checking regularity

In fact, the problem of checking if a given partition is ϵ-regular is a quite time consuming process. In fact, as pointed out in [3], constructing a ϵ-regular partition is easier than checking if a given one responds to the ϵ-regularity criterion.

Alon et Al. [3], prove the following results in the favor of this point of view.

Theorem 3. *The following decision problem is co-NP-complementary.*

Instance: An input graph G, an integer $k \geq 1$ and a parameter $\epsilon > 0$, and a partition of the set of vertex of G into $k+1$ parts.

Problem: Decide if the given partition is ϵ-regular.

As defined in [15], the complexity class co-NP-complementary collects all that problems which complement belongs to the NP-complete class:

A problem Q co-NP-complementary iff Q NP-complementary.

In other worlds, a co-NP-complementary problem is one for which there are efficiently verifiable proofs of its no-instance, *i.e.* its counterexample.

In this case, the proof of co-NP-completeness goes over multiple steps of reduction, first proving the complexity of some wider-use problems, and finally restricting these to a useful particular case. The formal proof, as given in [3] is technical. In the following, it will be only sketched. In particular, the authors follow the next steps:

- starting from the well-known NP-complete CLIQUE problem and showing, by the reduction mechanism, the complexity of some auxiliaries. Subjects of reductions are the HALF SIZE CLIQUE problem and the COMPLETE BIPARTITE SUB-GRAPH $K_{k,k}$ problem.
- once proved the complexity of $K_{k,k}$, characterizing a more strict version of the same, namely a half-size complete bipartite isomorphism.
- at last, reducing the half-size complete bipartite isomorphism to a non-ϵ-regularity problem for bipartite graphs (see Theorem 6), which implies Theorem 3.

3.3.1 The reduction mechanism

The standard way to prove NP-completeness is to find a polynomial reduction function that reduces a known NP complete problem to the new one. In particular, a problem Q_1 is polynomial time reducible to a problem $Q_2, Q_1 \leq_P Q_2$ if there exists

a polynomial time computation function

f : {instances of Q_1} $\rightarrow$ {instances of Q_2} such that for all x

$$\mathbf{x} \in \{\text{instances of } Q_1\}\text{if and only if } f(x) \in \{\text{instances of } Q_2\}. \quad (3.1)$$

In our case, Alon et al. [3] show via reduction that the following problems are NP-complete:

Theorem 4. *(The $K_{k,k}$ Problem). Given a positive integer k and a bipartite graph G with vertex classes A,B such that $|A| = |B| = n$, determine if G contains the complete bipartite graph with k vertices in each vertex class.*

Lemma 1. *(HALF SIZE CLIQUE Problem) The following decision problem is NP-complete: Given a graph G on n vertices where n is odd, decide if G contains a sub-graph isomorphic to the complete $\frac{n+1}{2}$-vertex sub-graph $K_{\frac{n+1}{2}}$.*

Reducing CLIQUE to HALF SIZE CLIQUE

The CLIQUE Decision Problem (K_k) asks if, given a graph G with n vertices and an integer k, G contains a k-vertices complete sub-graph. The problem can be reduced to the HALF SIZE PROBLEM (Lemma 1): a new graph G^* can be constructed as following. Consider $G = (V, E)$ and a integer k. G^* is the graph obtained from G and defined as:

- $G^* = G + K_{|V|+1-2k}$, *i.e.* the graph obtained from the disjoint union of G and $K_{|V|+1-2k}$ and joining every vertex of G to every vertex of $K_{|V|+1-2k}$, if $k \leq \frac{|V|+1}{2}$;
- G^* is the disjoint union of G and $E^{2k-|V|-1}$, which is the graph with $2k-|V|-1$ isolated vertices, if $k > \frac{|V|+1}{2}$.

In any case, the order of G^* is sufficiently high to have a half size clique of order $\frac{n+1}{2}$. In fact, if G has a K_k sub-graph and $k \leq \frac{|V|+1}{2}$, then G^* has a clique of dimension $k + |V| + 1 - 2k = |V| + 1 - k \geq |V| + 1 - \frac{|V|+1}{2} = \frac{|V|+1}{2}$.

Otherwise, if G has a K_k sub-graph and $k < \frac{|V|+1}{2}$, G^* has a K_k sub-graph and the order of G^* is $2k - 1$.

The procedure of transforming G into G^* is polynomial, so also HALF SIZE CLIQUE belongs to the class NP-C.

Reducing HALF SIZE CLIQUE to $K_{k,k}$

The second step of reduction transforms a half size clique problem instance to a $K_{k,k}$ one. In particular, consider a graph $G = (V, E)$ to be the input of half size clique. Consider also the bipartite graph $H = (A \cup B, F)$ defined as:

- $A = \{\alpha_{ij} | 1 \leq i, j \leq n\}$
- $B = \{\beta_{ij} | 1 \leq i, j \leq n\}$
- $F = \{(\alpha_{ij}\ ,\ \beta_{kl})\ \mid\ i = k \text{ or } (i, j) \in E \text{ and } l \neq i \text{ and } j \neq k\}$

Alon et al. [3] prove that G has a clique of size $\frac{n+1}{2}$ if and only if H has a sub-graph isomorphic to $K_{(\frac{n+1}{2})^2,(\frac{n+1}{2})^2}$. In fact, the vertex set of $K_{\frac{n+1}{2}}$ induces a complete bipartite sub-graph with colour classes of order $(\frac{n+1}{2})^2$ in H. On the other hand, if H has a sub-graph isomorphic to $K_{(\frac{n+1}{2})^2,(\frac{n+1}{2})^2}$ with classes $A^{'}$ and $B^{'}$, Alon et al. [3] prove that the only possible sub-graph in G induced by the vertices indexed in $A^{'}$ and $B^{'}$ is a clique of size $\frac{n+1}{2}$.

A stricter version of $K_{k,k}$

The previous reductions have proved that the $K_{k,k}$ Problem is NP-Complete. Also a more specific formulation of the problem belongs to the same complexity class, by the following Theorem.

Theorem 5. *The following problem is NP-Complete. Given a bipartite graph $G = (A \cup B, E)$ where $|A| = |B| = n$ and $|E| = \frac{n^2}{2} - 1$ decide if G contains a sub-graph isomorphic to $K_{\frac{n}{2},\frac{n}{2}}$*

The condition about the number of edges in G permits the connection to the edge-density and consequently to the non-ϵ-regularity.

Regularity in bipartite graphs

The final reduction in [3] involves the stricter version of $K_{k,k}$ and the problem of checking regularity in a bipartite graph $G = (A \cup B, E)$. Formally:

Theorem 6. *Theorem 6 The following problem is co-NP-complementary. Given $\epsilon > 0$ and a bipartite graph G with vertex classes A,B such that $|A| = |B| = n$, determine if G is ϵ-regular.*

The co-NP-Completeness of ϵ-regularity in a bipartite sub-graph proceeds from Theorem 5. Indeed, a bipartite graph G with n vertices in each colour class and $\frac{n^2}{2} - 1$ vertices contains a $K_{\frac{k}{2},\frac{k}{2}}$ if and only if it is not *epsilon*-regular for $\epsilon = \frac{1}{2}$. So, if $G = (A \cup B, E)$ is not ϵ-regular (*i.e.*there exist $X \subset A, B \subset Y, |X| \geq \frac{1}{2}n, |Y| \geq \frac{1}{2}n$ that testify the irregularity) and knowing that

$$\mathbf{d}(A, B) = \frac{e(A, B)}{|A|.|B|} = \frac{1}{2} - \frac{1}{n^2} \tag{3.2}$$

we can see that

$$|d(X, Y) - d(A, B)| = |d(X, Y) - \frac{1}{2} + \frac{1}{n^2}| \geq \frac{1}{2} \tag{3.3}$$

This is possible if and only if $d(X, Y) = 1$. Hence, the sub-graph of G induced by X and Y contains $K_{\frac{n}{2},\frac{n}{2}}$.

On the other side, if G contains $K_{\frac{n}{2},\frac{n}{2}} = (X, Y, E \cap X \times Y)$ then $d(X, Y) = 1$ and

$$|d(X, Y) - d(A, B)| = \frac{1}{n^2} + \frac{1}{2}| > \frac{1}{2} \tag{3.4}$$

Hence, G is not ϵ-regular for $\epsilon = \frac{1}{2}$.

As the Szemerédi's Lemma deals with pair of vertex sets and the ϵ- regularity should be tested for each possible pair, Theorem 6 directly implies Theorem 3.

3.4 Extension to the Regularity Lemma

Recently, the interest in Szemerédi's Regularity Lemma is growing with the study of its weighted hypergraph extensions. In particular, Czygrinow and Rödl [16] propose a possible Regularity Lemma reformulation that directly extends the Szemerédi's original one.

As before, we need definitions for basic concepts as hypergraph, density, regular pair and regular partition.

A hypergraph is the common generalization of a graph. In particular in a hypergraph $H = (V, E)$, the elements of the edge set E are non-empty subsets of any cardinality of V . If all vertex subsets in E have the same cardinality l, then the hypergraph is said to be l-uniform. A graph is a particular case of hypergraph (2-uniform hypergraph). A l-uniform hypergraph is weighted if there is a non-negative weighted function $: [V]^l \to R_+ \cup 0$.

Let $V_1, \cdots, V_l$ subsets of V such that $V_i \cap V_j = \emptyset$ for each $i \neq j$. The weighted density is defined as

$$d_w(V_1, \cdots, V_l) = \frac{\sigma(v_1, \cdots, v_l) : (v_1, \cdots, v_l) \in V_1 \cdots V_l}{K|V_1| \cdots |V_l|} \tag{3.5}$$

where $K = max_{v_1,\cdots,v_l \in [V]^l}(v_1, \cdots, v_l) + 1$ acts as a normalization factor.

In this case the notion of ϵ-regularity doesn't involve pairs of subsets, but tuples. In particular an l-tuple $(V_1, \cdots, V_l), V_i \subset V$ for each $i = 1 \cdots l$ with $V_i \in V_j = \emptyset$ for each $i \neq j$ is said to be $(\epsilon,)$-regular if for every subsets $W_i \subset V_i$ with $|W_i| \geq \epsilon |V_i|$ the following inequality holds:

$$|d_w(V_1, \cdots, V_l) d_w(W_1, \cdots, W_l)| < \epsilon.$$

Finally, a partition $V_0 \cup V_1 \cup \cdots \cup V_t$ of V is $(\epsilon,)$-regular if all the conditions are satisfied,

- $|V_0| \leq \epsilon |V|$
- $|V_i| = |V_j|$ for all $i, j \in 1 \cdots, t$
- all but at most ϵt^l l-tuples $(V_{i1}, \cdots, V_{il})$with $i_1, \cdots, i_l \subset [t]^l$are $(\epsilon,)$-regular.

The generalization of Regularity Lemma for hypergraphs, as in its original formulation, states that for each $\epsilon > 0$ and every $m \in N$ there are $M, N \in N$ such that every hypergraph $H = (V, E,)$ with $|V| \geq N$ has an $(\epsilon,)$-regular partition $V_0 \cup V_1 \cup \cdots \cup V_t$, where $m \leq t \leq M$.

As happened in the normal graph case, also Czygrinow and Rödl [16] show some property of hypergraph regularity, as for example continuity of density function. Moreover, they describe some possible applications, we can find for the normal graph case, *i.e.* Max-Cut problem for hypergraph and estimation of the chromatic number. All these results seem to demonstrate that the hypergraph approach is a suitable generalization of the original Regularity Lemma. Other researchers carried out similar studies. See, for example, [70, 35, 31].

3.5 Algorithms for the Regularity Lemma

The original proof of Szemerédi's Lemma proposed by Szemerédi (see [[81], [82]]) is not algorithmic. This has not narrowed the range of possible applications of the result in the fields of extremal graph theory, number theory and combinatorics. However, it was at once plain that a possible development towards an algorithmic version of the lemma would have led to several improvements. First of all, there would be the possibility of rewriting many of the previously achieved results and proposing new algorithmic formulations (see [3] , [27]).

Example 1. *One of the possible algorithmic reformulation, proposed in [3], given a graph H, deals with the number of H-isomorphic sub-graphs in a graph G. Alon and Yuster, in [4], had previously demonstrated the following result:*

Theorem 7. *For every $\epsilon > 0$ and every integer h, there exists a $n_0 = n_0(\epsilon, h)$ such that for every graph H with h vertices and for every $n > n_0$, any graph G with n vertices and with minimum degree $d \geq \frac{X(H)-1}{X(H)}n$ contains at least*

$$\frac{(1-\epsilon)n}{h} \tag{3.6}$$

vertex disjoint copies of H.

The original proof used the Szemerédi's Lemma in order to show the existence of a subset of copies of H. The development of an algorithm to compute a Szemerédi's Regular Partition leads to the immediate reformulation of the previous result, as in [3].

Theorem 8. *For every $\epsilon > 0$ and for every integer h, there exists a positive integer $n_0 = n_0(\epsilon, h)$ such that for every graph H with h vertices and chromatic number (H), there exists a polynomial algorithm that given a graph $G = (V, E)$ with $n > n_0$ vertices and minimum degree $d \geq \frac{X(H)-1}{X(H)}n$ contains at least*

$$\frac{(1-\epsilon)n}{h} \tag{3.7}$$

vertex disjoint copies of H in G.

In this case, the Szemerédi's Regular Partition gives the necessary instrument to create one by one all the vertex-disjoint copies of H. The time complexity of the procedure remains, also in this situation, polynomial.

The Example 1 is only one of the possible applications of the algorithmic version of the Szemerédi's Lemma. Alon et al. [[3]] cite other examples related to the fields of recognition of sub-graphs, topological copies, number of isomorphic sub-graphs, creation of K_k-free graphs.

In [[3]], the authors, dealing with these algorithms, propose a new formulation of the lemma, which emphasizes the algorithmic nature of the result.

Theorem 9. *(A constructive version of the Regularity Lemma). For every $\epsilon > 0$ and every positive integer t there is an integer $Q = Q(\epsilon, t)$ such that every graph with $n > Q$ vertices has a ϵ-regular partition into $k + 1$ classes, where $t \leq k \leq Q$. For every fixed $\epsilon > 0$ and $t \geq 1$ such a partition con be found in $O(M(n))$ sequential time, where $M(n)$ is the time for multiplying two n by n matrices with $\{0, 1\}$ entries over the integers. It can also be found in time $O(logn)$ on a EREW PRAM with a polynomial number of parallel processor.*

In the next sections two of the most important algorithm to create a Szemerédi's Partition will be described. Finally, the Czygrinow and Rödl [16] extension for hypergraph will be presented.

3.6 The first Algorithm

The first algorithmic reformulation of Szemerédi's Lemma is proposed by Alon et al. [3] in 1994. In their proposed method, they investigate the neighborhood relations between vertices in the a bipartite graph, in order to recognize witnesses of not-regularity.

Given $\epsilon > 0$ and a graph $G = (V, E)$, it computes a new value $\epsilon' < \epsilon$.

ϵ' is a function of the previous value ϵ. The algorithm recognizes the following possible situations:

- If G is not ϵ-regular, the algorithm builds vertex subsets that are witnesses of not ϵ'-regularity.
- If G is ϵ'-regular, the algorithm decides that G is also ϵ-regular.
- The previous cases don't cover all the possible situations. Indeed, G could be ϵ-regular but not ϵ'-regular. In this case, the algorithm decides to categorize G in one of the previous situations.

3.7 Checking regularity in a bipartite graph

Before the detailed description of the procedure, Alon et al. [3] give proofs of the major steps of the algorithm. In particular, the following lemmas describe all possible situations: the regularity termination case and the not regularity one, which produces the witnesses. In each case, the result emphasizes their relations with the adjacency matrix of G. All the results described in this sections refer to a single pair of vertex sets. The complete algorithm, on the other hand, will deal with a partition and will include steps to improve the current partition, as stated in Lemma B.1. First of all, it could be useful to introduce some definitions. In particular, Alon et al. [3] , studying a bipartite graph H with equal colour classes $|A| = |B| = n$, define the average degree d of H as

$$d = \frac{1}{2n} \sum_{i \in A \cup B} deg(i) \tag{3.8}$$

where, $deg(i)$ is the degree of the vertex i (see [19]). Let y_1 and y_2 be two distinct vertices, such that y_1, $y_2 \in B$. Alon et al. [3] define the neighborhood deviation of y_1 and y_2 by

$$\sigma(y_1, y_2) = |N(y_1) \cap N(y_2)| - \frac{d^2}{n} \tag{3.9}$$

In the Equation (3.9) , $N(x) \in N$ is the set of neighbors of the vertex x. Finally, for a subset $Y \subset B$, the deviation of Y is

$$\sigma(Y) = \frac{\sum_{y_1, y_2 \in Y} \sigma(y_1, y_2)}{|Y|^2} \tag{3.10}$$

The first Lemma describes the possible scenarios that could be found analyzing a pair of subsets in the current partition. The pair can be usefully reduced to a bipartite graph with equal colour classes, because the Lemma never considers the neighborhood relations between vertices in the same subset.

Lemma 2. *Let H be a bipartite graph with equal colour classes $|A| = |B| = n$, and let d denote the average degree of H. Let $0 < \epsilon < \frac{1}{16}$.*

If there exists YB, $|Y| > \epsilon n$ such that $\sigma(Y) \geq \frac{\epsilon^3}{2} n$ then at least one of the following cases occurs.

1. $d < \epsilon^3 n$

2. There exists in B a set of more than $\frac{1}{8}\epsilon^4 n$ vertices whose degrees deviate from d by at least $\epsilon^4 n$.

3. There are subsets $A' \subset A$, $B' \subset B$, $|A'| \geq \frac{\epsilon^4}{4}n$, $|B'| \geq \frac{\epsilon^4}{4}n$ and $|d(A', B')d(A, B)| \geq \epsilon^4$.

Moreover, there is an algorithm whose input is a graph H with a set $Y \subset B$ as above that outputs either

- The fact that 1 holds, or
- The fact that 2 holds and a subset of more than $\frac{1}{8}\epsilon^4 n$ members of B demonstrating this fact, or
- The fact that 3 holds and two subsets A' and B' as in 3 demonstrating this fact.

The algorithm runs in sequential time $O(M(n))$, where $M(n) = O(n^2.376)$ is the time needed to multiply two n by n matrices over the integers, and can be parallel and implemented in time $O(logn)$ on a EREW PRAM with a polynomial number of parallel processors. The hypothesis regarding the vertex subset Y assures that it is possible to find a subset of the colour class B in which every vertex has a sufficient high number of neighbors in A. If the pair should result ϵ-irregular, then the witnesses of irregularity (the vertex subsets A' and B') have to be searched starting form Y . Indeed $B' \subset Y$ and, most properly, $B' \subset Y'$, where $Y' = y \in Y||deg(y) - d| < \epsilon^4 n$, i.e. the set of vertices whose degrees show a certain regularity with regard to the average degree of H. Alon et al. [3] demonstrate that it is possible to find a vertex $y_0 \in Y$ such that the number of common neighbors between y_0 and all the elements $b \in B'$ deviates from the value accepted for the regularity. Finally, Alon et al. [3] simply define the second witness as $A\prime = N(y_0)$.

On the other hand, the existence of Y doesn't assure that the current graph H is not directly ϵ-regular. The Fact 1 rephrases the ϵ-regularity conditions in terms of average degree and ϵ: if $d < \epsilon^3 n$, then (A, B) is a ϵ-regular pair. Indeed, considering the ϵ-regularity condition 1.5, and observing that

$$d(A, B) = \frac{e(A, B)}{|A||B|} = \frac{2e(A, B)}{2n}\frac{1}{n} = \frac{\sum_{i \in A \cup B} deg(i)}{2n}\frac{1}{n} = \frac{d}{n} \tag{3.11}$$

we have the following two cases:

- $d(A,B) - d(A',B^\epsilon) < \epsilon$: then

$$d(A,B) - d(A',B') = \frac{d}{n} - \frac{e(A,B)}{|A||B|} < \frac{d}{n} - 1 < \epsilon^3 - 1 < \epsilon \tag{3.12}$$

- $d(A,B)d(A',B^\epsilon) > -\epsilon$: then

$$d(A,B) - d(A',B') = \frac{e(A,B)}{|A||B|} - \frac{d}{n} > 1 - \frac{d}{n} - 1 > 1 - \epsilon^3 > -\epsilon \tag{3.13}$$

In the previous statements, the following relations have been used: $d(.,.) \in [0,1]$ and in particular $d(.,.) < 1$, the hypothesis $d < \epsilon^3 n$ and finally $\epsilon \in [0,1]$.

If both Facts 1 and 3 in Lemma 2 don't happen, then there exists another scenario: Fact 2 describes the situation in which the number of vertex in B deviating from the average degree more than $\epsilon^4 n$ is too high to permit the creation of the first witness B'. Remember that B' is always constructed as a set of vertices that respect the regularity of distribution of degrees around the average degree. On the contrary, A' is the set of vertices whose degrees deviate from the average degree with regard to the vertices selected in B'. Alon et al. [3] also prove a Lemma that relates the existence of a high number of vertices $b \in B$ which degrees deviate from the average and the existence of a vertex subset $Y \subset B$ such that $\sigma(Y)\frac{\epsilon^3}{2}n$.

Lemma 3. *Let H be a bipartite graph with equal classes $|A| = |B| = n$. Let $2n^{\frac{1}{4}} < \epsilon < \frac{1}{16}$. Assume that at most $\frac{1}{8}\epsilon^4 n$ vertices of B deviate form the average degree of H by at least $\epsilon^4 n$. Then, if H is not ϵ-regular then there exists $Y \subset B$, $|Y| \geq \epsilon n$ such that $\sigma(Y) \geq \frac{\epsilon^3}{2}n$. More interesting to the final algorithm is the corollary to the Lemma 3.*

Corollary 1. *Let H be a bipartite graph with equal classes $|A| = |B| = n$. Let $2n^{1/4} < \epsilon < \frac{1}{16}$. There is an $O(M(n))$ algorithm that verifies that H is ϵ-regular or finds two subsets $A' \subset A$, $B' \subset B$, $|A'|\frac{\epsilon^4}{4}n$, $|B'|\frac{\epsilon^4}{4}n$ and $|d(A',B') - d(A,B)| \geq \epsilon^4$.*

It is quite easy to show that the computation time of the algorithm doesn't exceed $O(M(n))$. Indeed, the computation of d, the average degree of H, takes a $O(n2)$ time. Once d is computed, the ϵ-regularity of H could be checked. Then, it is necessary to count the number of vertices in B whose degrees deviate from d by at least $\epsilon^4 n$. The operation takes a time in $O(|B|^2)$. If the number of deviating vertices is more than $\frac{\epsilon^4}{8}n$ then Lemma 3 and the observation that H is not ϵ-regular assure that there exists a subset $Y \subset B$ such that $|Y| \geq \epsilon n$ and $\sigma(Y) \geq \frac{\epsilon^3}{2}n$. Lemma 2, now, states that the required subsets A' and B' can be found in $O(M(n))$ time.

3.7.1 The algorithm

The algorithm combines the lemmas presented in the previous section and the observations introduced directly in the original paper by Szemerédi (see [82]). The procedure is divided into two main steps: in the first step all the constants needed during the next computation are set; in the second one, the partition is iteratively created. An iteration is called refinement step, because, at each iteration, the current partition is closer to a regular one. Given $\epsilon > 0$ and $t \in N$, the lower bound of the number of vertices needed to assure the regularity of the partition, $N = N(\epsilon, t)$, and the upper bound of the number of possible sets in the partition, $T = T(\epsilon, t)$, are computed. It is important to note that, with regard to the definition in Theorem 9, it is clear that $T \leq Q$. As already described in [82], let b be the least positive integer such that

$$4^b > 600(\frac{\epsilon^4}{16})^{-5}, b \geq t \tag{3.14}$$

Let $f : N \to N$ be a function defined as:

$$f(0) = b, f(i+1) = f(i)4^{f(i)} \tag{3.15}$$

f describes the growth of the cardinality of the partition over the iterations.

Indeed, as described in [82], if the partition doesn't result ϵ-regular, then every set has to be split into 4^k subsets, where k is the cardinality of the current partition. Moreover, Szemerédi also states that the index of partition $ind(P)$, as defined in B.1, is bounded and can be improved of the quantity $\frac{\epsilon^5}{20}$ during each refinement step (see Lemma 4). As described in [43], the number of refinement steps is bounded, because $ind(P)$ is also bounded. In particular, after q steps and using Equation (3.21), the current partition P_q has:

$$\frac{1}{2} \geq ind(P_q) \geq ind(P) + \frac{q\epsilon^5}{20} \tag{3.16}$$

Hence,

$$q \leq \frac{10}{\epsilon^5} \tag{3.17}$$

and the number of refinement steps is no more than $10\epsilon^5$. Therefore, the number of classes will be at most $f(\frac{10}{\epsilon^5}) + 1$

Returning to the constants setting, $T = (10(\frac{\epsilon^4}{16})^{-5})$

and $N = max\{T4^{2T}, \frac{32T}{\epsilon^5}\}$. Alon et al.[3] prove the theorem with $Q = N(\geq T)$.

The procedure

Given a graph $G = (V, E)$ with n vertices where $n \geq N$, let k be the cardinality of the partition. The following steps describe the algorithmic version of Szemerédi's Lemma.

1. **Creating the initial partition:** arbitrarily divide the set V into a equitable partition P_1 with classes $C_0, C_1, \cdots, C_b$ where $|C_i| = n/b$, $i = 1 \cdots b$ and $|C_0| < b$. Let $k_1 = b$.

2. **Verifying the Regularity:** for every pair C_r, C_s of P_i verify if it is ϵ-regular or find $X \subset C_r$, $Y \subset C_s$, $|X| \geq \frac{\epsilon^4}{16}|C_r|$, $|Y| \geq \frac{\epsilon^4}{16}|C_s|$ such that

$$|d(X, Y) - d(C_s, C_r)| \geq \epsilon^4 \tag{3.18}$$

3. **Counting the Regular Pairs:** if there are at most $\epsilon\binom{k_i}{2}$ pairs that are not verified as ϵ-regular, then halt. P_i is an ϵ-regular partition.

4. **Iterative step:** apply Lemma B.1 where $P = P_i$, $k = k_i$, $\gamma = \frac{\epsilon^4}{16}$ and obtain a partition P' with $1 + k_i 4^{k_i}$ classes.

5. Let $k_{i+1} = k_i 4^{k_i}$, $P_{i+1} = P'$, $i = i + 1$ and go to step 2.

Lemma 4. *Let $G = (V, E)$ be a graph with n vertices. Let P be an equitable partition of V into classes $C_0, C_1 \cdots C_k$, where the exceptional class is C_0 . Let $\epsilon > 0$. Let k be the least positive integer such that*

$$4^k > 600\epsilon^{-5} \tag{3.19}$$

If more than ϵk^2 pairs (C_s, C_t) in $1 \leq s < t \leq k$ are ϵ-irregular, then there is an equitable partition Q of V into $1 + k4^k$ classes, the cardinality of the exceptional class being at most

$$|C_0| + \frac{n}{4^k} \tag{3.20}$$

and such that

$$ind(Q) > ind(P) + \frac{\epsilon^5}{20} \tag{3.21}$$

The idea formalized in the Lemma 4 B.1 is that, if a partition violates the Regularity condition, then it can be refined by a new partition and, in this case, the $ind(P)$ measure can be improved. On the other hand, the new partition adds only few elements to the current exceptional set, so, in the end, its cardinality will respect the definition of equitable partition.

3.8 Algorithm Implementation

3.8.1 Connection between Pairwise Clustering and Regularity Lemma

The aim of this section is to test Szemerédi's regularity Lemma in the clustering application domain.

The notion of cluster is one of fundamental tools in machine learning which is applied in many disciplines dealing with large amounts of data such as data mining and pattern recognition. Formally, clustering is an unsupervised learning method that uses the relations between input data.

The goal of Cluster Analysis is finding appropriate common characteristics among relational data to make them as a group, called cluster. Therefore, elements in the same cluster share a high degree of similarity among them and dis-similarly with other clusters. K-means is a common example for a distance based clustering method.

The algorithm produces a partition of a n-elements data-set into k equivalence classes, whose k is previously chosen. Each element is represented as a point in a d-dimensional space. Each cluster is constructed around an initially random selected point, called centroid. The algorithm aims to minimize the error between each point in cluster and the centroid of that cluster.

Density based methods are well designed for a spatial disposition and a coordinate system, but not all the elements subjected to clustering can be represented in this way. Therefore, we need a more general method that have a wider applicability with a greater degree of complexity and not be a problem dependent.

To overcome this challenge, notion of affinity between elements are introduced. The affinity function is a measure of the relation between every possible pair of elements in the data-set which is used in Pairwise Clustering. The aim of Pairwise Clustering is to create a partition of vertex-set using the information provided by edge weights. In this case, the partition problem changes into the clustering one. Some of established methods that use Pairwise clustering are Normalized-Cut Algorithm, Dominant Sets and Path-Based Clustering (see [77, 64, 62, 25]). The Regularity Lemma is partitional method that partitions vertices by Pairwise disjoint sets. However, the Regular Partition and clustering are fundamentally different, the Regularity Lemma shows some interesting characteristics of clustering notion which are worth to investigate.

- Szemerédi's Lemma, like Pairwise clustering, maps data and its relations to a graph and creates a partition over graph's vertices.
- The Regularity Lemma is designed for unweighted graphs by Alon et al [3] but the weighted extended version is not affected by weights of edges.
- We are able to say that the relation between two regular sets is stronger and more structured than a relation between regular and irregular sets. Hence the relation between two elements extends to the level of sets.
- Like Pairwise clustering methods, every function in Regularity Lemma is based on edges and its weights.
- One of the main characteristics of Regularity Lemma is size equality of each set.
- The main difference between clustering and partitioning is, clustering methods divide a data-set to a similar items in a same group as much as possible, in the case of Regularity Lemma, only the relations between elements in different sets are subject of interest. It should be emphasis that the less intra-set edges in each partition, the better is the partition however we don't have a guarantee to reach the best partition.

3.8.2 Reduced Graph and Key Lemma

To complete the proposed method, we need to define two more concepts which are Key Lemma and Reduced graph.

It is shown that a regular partition with fixed number of vertices has a sufficiently high number of edge density. This observation permits to define Reduced Graph [43].

Definition 3. *Definition 3 (Reduced Graph). Given a graph $G = (V, E)$, a partition P of the vertex-set V into the set $V_1, V_2, \cdots V_k$ and two parameters and d, the Reduced Graph R is defined as:*

1. *the vertices are the cluster $V_1, V_2, \cdots V_k$.*
2. *V_i is joined to V_j if (V_i, V_j) is ϵ-regular with density more than d.*

The Reduced Graph gives a simple structure of the original graph and at the same time the most significant edges are considered during the reconstruction. Additionally, Reduced Graph R provides a tool to study the original graph G with less complexity. Indeed, many properties of R are directly inherited from G.

Lemma 5. *(Key Lemma). Given $d > \epsilon > 0$, a graph R and a positive integer m (m is fixed and derives directly from the Regular Partition.), construct a graph G following these steps:*

1. *replace every vertex of R by m vertices*

2. *replace the edges of R with regular pairs of density at least d.*

For the details of the demonstration of Lemma 5, see [43]. It is clear by Lemma 5 that every small sub-graph of R is also a sub-graph of G. The direct consequence of the Key Lemma is that it is possible to search for significant substructure in a Reduced Graph R in order to find common sub-graphs of R and the original graph.

Finally, combination of the Reduced Graph and the Key Lemma provides a tool for mapping substructure back to the original set of elements. The Reduced graph could contain more information about relations between subsets in the original configuration than about point to point substructures.

3.8.3 The Proposed Method

Alton's algorithmic method is designed for unweighted graph so we had to modify it for the weighted case. In the following lines, we explain the modifications. An edge-weighted graph can be interpreted as a degree of similarity which lies between 0 and 1 as no similarity and highest similarity respectively.

As it has been shown, the Szemerédi's Regularity Lemma accepts a weighted graph as input and produces a partition according to weights and edges existence in the graph.

The density between the sets of a pair in a weighted context is

$$d = d(A, B) = \frac{\sum_{i=1}^{|A|} \sum_{j=1}^{|B|} (a_i, b_j)}{|A||B|} \tag{3.22}$$

In Equation 3.22 the cardinality $|A| = |B| = n$ and $d(A, B)$ is always bounded in the interval $[0, 1]$. The zero is a full connected vertices between two sets and one

is a null case. Additionally, the number of edges crossing the partition is significant for $d(A, B)$.

The measure $d(A, B)$ can be interpreted as a combined regularity and vertex similarity with inter-subset and intra-subset relations. The measure itself doesn't have potential to handle both aspects, so by defining average weighted degree, we are able to emphasize intra-subset relations.

$$awdeg_S(i) = \frac{1}{|S|} \sum_{j \in S} w(i, j) \quad S \subseteq V \tag{3.23}$$

All elements in the current subset S are put in decreasing order by average weighted degree. In this way, the partition of S takes place simply subdividing the ordered sequence of elements into the wanted number of subsets.

After these steps, a Pairwise clustering method like dominant set can be applied to search structures in the Reduced Graph.

To end this section some technical issues have been applied:

The exceptional set is applied to have the same cardinality for each subsets which interpreted as the less significant elements in cluster. To reconstruct back the original graph, exceptional set are assigned to the closer cluster (shortest distance is the criteria to assign to a cluster in case of Euclidean points clustering).

In the world real application, the number of iterations and the vertex set cardinality required is simply too big to be considered. So, in this algorithm adaptation, iterations stop when a Regular Partition has been constructed or when the subsets size becomes smaller than a previously chosen parameter.

The next-iteration number of subsets is also intractable. So it has been decided to split every subset, from an iteration to the next one, on the basis of a user selected fixed parameter (In our case, we set it to 2).

Finally, **the relation between Dominant Sets and this graph partitioning is:**

If the exceptional set is empty and the subsets A and B cardinality is 1, we have

$$d = d(A, B) = \frac{\sum_{i=1}^{|A|} \sum_{j=1}^{|B|} w(a_i, b_j)}{|A||B|} = w(a_1, b_1) \tag{3.24}$$

3.9 Experiments

3.9.1 Dominant Sets

The edge weighted graph $G = (V, E,)$ representing a set of n elements and (i, j) is a not-negative weight function that maps edges in real values and measures the similarity between the considered vertices i and j that is obtained from the following formula.

$$w(i,j) = exp(\frac{-||F(i) - F(j)||^2{}_2}{\sigma^2}) \tag{3.25}$$

where

1. σ is a positive real number and is bounded in the interval $(0, 1]$.

2. $F(i)$ is a feature vector and $||.||_2$ is the Euclidean distance between the two values which gives dissimilarity between two considered elements.

Dominant Sets are a pairwise clustering method, developed in 2003 by Pavan and M. Pelillo ([62, 64]). The Dominant Sets framework generalizes the concept of complete sub-graph or clique to weighted graphs. Generally, a clique is defined for unweighted graphs and its characteristic of compactness is accepted as a definition of cluster.

As often happens in pairwise clustering, data to be clustered are represented by an undirected edge-weighted graph with no self-loops.

$G = (V, E,)$, where the vertex set $V = 1, \cdots, n$ corresponds to elements in the data-set, the edge set $E \subseteq V \times V$ represents neighborhood relationships and $: E \to R^+ \cup 0$ is a positive weight function that reflects the similarity between pairs of linked vertices.

The graph G is represented by a nxn non-negative weighted matrix $A = (a_{ij})$, known as weighted adjacency matrix. A is defined as:

$$a_{ij} = \begin{cases} w(i,j) & \text{if } (i,j) \in E \\ 0 & \text{otherwise.} \end{cases} \tag{3.26}$$

The assignment of edge-weights induces an assignment of weights on the vertices. These weights are used to give a formal definition of cluster.

Let $S \subseteq V$ be a non-empty subset of vertices and $i \in V$. The average weighted degree of i w.r.t. S is defined as:

$$awdeg_S(i) = \frac{1}{|S|}\sum_{j\in S} a_{ij} \tag{3.27}$$

Pavan and Pelillo [62] define also a function

$$\phi_S(i,j) = a_{ij} awdeg_S(i) \tag{3.28}$$

where $j \notin S.\phi_{\{i\}}(i,j) = a_{ij}$, for all $i,j \in V$ with $i \neq j$. $\phi_S(i,j)$ measures the similarity between nodes j and i, with respect to the average similarity between node i and all neighbors in S.

Node-weights can now be defined as follow. Let $S \subset V$ be a non- empty subset of vertices and $i \in S$. The weight of i w.r.t S is

$$w_S(i) = \begin{cases} 1 & \text{if } |S| = 1 \\ \sum_{j\in S/\{i\}} \phi_{S/\{i\}} W_{S/\{i\}}(j) & \text{otherwise.} \end{cases} \tag{3.29}$$

Moreover, the total weight of S is defined to be:

$$W(S) = \sum_{i\in S} w_S(i) \tag{3.30}$$

Intuitively, $w_S(i)$ gives a measure of the overall similarity between vertex i and the vertices of $S/\{i\}$ with respect to the overall similarity among the vertices in $S/\{i\}$.

Finally, it is possible to formally define a cluster, remembering the notion of intra-cluster homogeneity and inter-clusters unhomogeneity:

Definition 4. *A non-empty subset of vertices $S \subseteq V$ such that $W(T) > 0$ for any non-empty $T \subseteq S$, is said to be dominant if:*

1. *$w_S(i) > 0$, for all $i \in S$*
2. *$w_{S\cup\{i\}}(i) < 0$, for all $i \notin S$*

The problem of finding a Dominant Set in a graph is strictly related to solving a quadratic problem. Consider $G = (V, E,)$ and its weighted adjacency matrix A. The following quadratic problem is a generalization of the Motzkin-Straus program [55]:

$$\begin{aligned} \text{maximize} \quad & f(x) = X'AX \\ \text{s.t.} \quad & X \in \triangle \end{aligned} \tag{3.31}$$

Where $\triangle = \{\mathbf{X} \in \mathbb{R}^{\mathsf{n}}, x_i \geq 0 \quad \text{for all} \quad i \in V \quad \textit{and} \quad e^T x = 1\}$

is the standard simplex of $\mathbb{R}^{\mathsf{n}}$, $\mathbf{e}$ is a vector of appropriate length consisting of unit entries (hence $\mathbf{e}^{'} x = \sum_i x_i$), and a prime denotes transposition.

The following theorem, proved in [62], states the relation between Dominant Sets and local solution to program.

Theorem 10. *If S is a dominant subset of vertices, then its weighted characteristics vector x^S , which is the vector of $\triangle$ defined as*

$$x_i^S = \begin{cases} \frac{w_S(i)}{W(S)}, \textit{ if } i \in S \\ 0 \textit{ otherwise.} \end{cases} \tag{3.32}$$

is a strict local solution of the optimization 3.31. Conversely, if x is a strict local solution of the optimization then its support $S = \sigma(x)$ is a dominant set, provided that $w_{S\cup\{i\}}(i) \neq 0$ for all $i \notin S$.

Theorem 10 permits to find a Dominant Set by calculating a solution of a program: the individuated Dominant Set will correspond to the support set of the found solution.

The connection between Dominant Sets and local optima is important because it permits the framework to be really competitive in computational time. Local solution of quadratic problems can be found, indeed, in a very short time by the so-called Replication Dynamics, a class of continuous- and discrete-time dynamical systems arising from evolutionary game theory (see [62, 74, 64, 9, 52]).

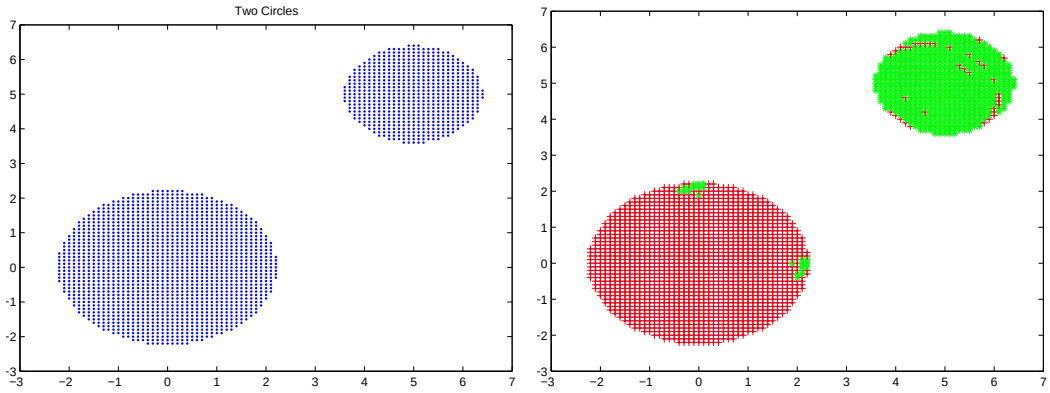
The basic idea under the Dominant Sets framework is that if we have found a cluster via Dominant Set in the original complete set of objects, it is successively possible to search for other clusters that satisfy the partitional nature of the problem. In other words, every time a new cluster is found, we search for the others in the set of all not already clustered elements, until all the objects have been partitioned. The Dominant Set framework has already been used for problems of clustering and image segmentation, as it is possible to see in [64, 62].

We could show that by extension of dominant set method, we are able to determine the number of clusters which can be used to compress the graph. The detail information is provided in appendix.

3.9.2 Experimental Results

Synthetic Experiment

We made an easy example of two filled circles with different size. The location of each pixel is used as a feature vector and similarity matrix is made based on the features.



(a) Two Circles (b) The classification result

Figure 3.1: This experiment shows the classification result of two circles with Dominantset.

Figure 3.1 shows the result of two step method based on Szemerédi reduced matrix and using Dominantset as classification. In this example the number of pixels are 1569×1569 and the size of reduced matrix is 32×32. The final accuracy is 97.5%.

Real-World Experiment

We performed some preliminary experiments aimed at assessing the potential of the approach described in the previous sections. We selected the following UCI data-sets: the IRIS database (150 elements, 3 classes), Pima database (768 elements, 2 classes), Haberman (306 elements, 2 classes), Wine (178 elements, 3 classes), Imox (192 elements, 2 classes), ecoli (272 elements, 3 classes), auto-mpg (398 elements, 2 classes), biomed (194 elements, 2 classes), breast (683 elements, 2 classes), sonar (208 elements, 2 classes), glass (214 elements, 4 classes) and Ionosphere (351 elements, 2 classes).

Table 3.1 summarizes the result obtained on these data by showing the classification accuracy obtained by our two-phase strategy for each database considered. Recall that in the second phase we used the dominant-set algorithm for cluster-

DataSet	Classif. Accuracy					
	Size	Two-Phase	Plain-DS	R.G Size	Compression Rate	Class No
IRIS	150	84%	88%	75	50%	3
Pima	768	60.42%	62.74 %	32	95.83 %	2
Haberman	306	73.53 %	73.20 %	153	50 %	2
Wine	178	67.42 %	71.35 %	89	50 %	3
Imox	192	53.65 %	77.08 %	64	66.67 %	2
ecoli	272	85.66 %	91.54 %	136	50 %	3
auto-mpg	398	81.91 %	84.17 %	132	66.83 %	2
biomed	194	72.68 %	80.93 %	64	67.01 %	2
breast	683	65.15 %	95.46 %	2	99.71 %	2
sonar	208	53.37 %	52.88 %	69	66.83 %	2
glass	214	46.26 %	52.34 %	71	66.82 %	4
Ionosphere	351	64.10%	70.09%	2	99.43%	2

Table 3.1: Results obtained on the UCI benchmark datasets. Each row represents a dataset, while the columns represent (left to right): the number of elements in the corresponding dataset, the classification accuracy obtained by the proposed two-phase strategy and that obtained using the plain dominant-set algorithm, respectively, the size of the reduced graphs, the compression rate and the number of classes.

ing the reduced graph. Further, for the sake of comparison, we present the results produced by a direct application of the dominant-set algorithm to the original similarity graph without any pre-clustering step. As can be seen, our combined approach substantially outperforms the plain algorithm. As can be seen, the cardinality of the reduced graph (or, alternatively, the size of the regularity subsets which in our implementation is a user-defined parameter) is of crucial importance as introduces a trade-off between the second-phase precision and the overall performance. Note how, using the regularity partitioning process, we were able to achieve compression rates from 50 to 99.71 while improving classification accuracy.

Table 3.2 summarizes the result of Breast benchmark of UCI dataset. In this experiment, effect of different values of ϵ in the range [01] is investigated. By increasing the value of ϵ compression rate increases and in the same time accuracy falling down.

	BREAST from UCI DataSet						
ϵ	0.11	0.16	0.21	0.26	0.31	0.36	0.41
Accuracy	92.97%	88.87%	87.41%	77.45%	65.45%	65.15 %	65.15 %
Compression	50.07 %	50.07 %	50.07 %	80.09 %	98.83 %	99.41 %	99.71%
ϵ	0.46	0.51	0.56	0.61	0.66	0.71	0.77
Accuracy	65.15 %	65.15%	64.86%	64.86%	65.15%	65.15%	64.86 %
Compression	99.71 %	99.71 %	99.71 %	99.71 %	99.71 %	99.71 %	99.71%

Table 3.2: Result obtained on Breast dataset of UCI benchmark. Ech row represents a ϵ, accuracy and compression rate

3.10 Discussion and future work

The aim of the chapter was compressing a graph with Szemerédis Regularity Lemma applied in the field of Pairwise Clustering.

The Szemerédi's Lemma is now one of the most famous theorem in Extremal Graph Theory. Two steps strategy evaluation as described, shows some interesting properties that is used as a compression device.

In the light of the studied examples of applications, the following conclusive observations emerged:

- It is justified to consider the Lemma as a preclustering strategy.
- The use of Reduced Graph as a weighted instrument to investigate the existence of significant substructures in the original graph is justified by the theoretical framework of Key Lemma.
- From an operational point of view, the Two Step Strategy is able to combine a edge-density based partitional method with a vertex similarity based one. Through the use of Reduced Graph, the clustering method becomes faster.

For the future work, we have observed that, there is an interesting potential to apply the SRL to the matching problem. Based on the earlier work of professor M. Pelillo in this field [[67], [65] and [66]], we can apply SLR method to the associate matrix of two graphs before detecting the maximum clique of it. Therefore, in this way, we are able to reduce time as far as Maximum clique detection is NP-complementary problem.

IV

A Matrix Factorization Approach to Graph Compression with Partial Information

4

A Matrix Factorization and Graph Compression

He who has overcome his fears will truly be free

Aristotle

4.1 Introduction

In the recent years, matrix factorization approaches have become an important tool in machine learning and found successful applications for tasks like, *e.g.* information retrieval, data mining and pattern recognition. Well-known techniques based on matrix factorization are singular value decomposition, principal component analysis, latent semantic analysis, non-negative matrix factorization and concept factorization. Singular Value Decomposition (SVD) [38] decomposes without loss of information any data matrix (not necessarily square) into the product of two unitary matrices (left and right factors) and a non-negative diagonal matrix (central factor). Principal Component Analysis (PCA) [41] is a statistical tool that determines an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values from linearly uncorrelated variables, known as principal components; the principal components can actually be recovered through the SVD factorization. Latent Semantic Analysis (LSA) [17] is another tool developed in the context of natural language processing that is based on the SVD factorization; this technique aims at recovering hidden concepts from a typically sparse co-occurrence matrix of, *e.g.* words and documents in the context of document analysis. Further developments of LSA gave rise to the probabilistic Latent Semantic Analysis (pLSA) [36], which is

based on a mixture decomposition of the latent class model, and to Latent Dirichlet Allocation (LDA) [7], which is a more general generative model where observations are explained by latent groups capturing the similarity structure within the data. Non-negative Matrix Factorization (NMF) [47, 60] decomposes a non-negative data matrix into the product of two non-negative factor matrices in a way to minimize the divergence between the original matrix and the factorized one; several divergence measures have been proposed including Frobenious distance, Kullback-Leibler divergence, and more in general Bregman divergences. Some authors extended NMF by integrating regularization terms acting on the factor matrices. Notably, in [11] a method called Graph-regularized NMF (GNMF) is proposed, where the geometrical manifold of the data points is used to regularize their low-dimensional representation that is encoded in the non-negative right-factor matrix; a step forward is taken with the Discriminative Orthogonal Non-negative (DON) matrix factorization approach [49], which is characterized by an additional regularization term, aimed at rendering the low-dimensional representations more discriminant. Furthermore, relations between NMF and clustering have been established in [20], where the authors show how the k-means clustering formulation and the Laplacian-based spectral clustering formulation can be cast into weighted, symmetrized instances of NMF. Similar works shed light on a variety of further connections to graph matching, clique finding, bi-clustering [21], pLSA [22] and hidden Markov models [46]. Some of those connections actually involve more complex, quadratic matrix factorization, *i.e.* exhibiting a quadratic dependency on a factor matrix. Quadratic non-negative matrix factorization methods under different types of constraints have recently been studied in more details in [89, 90]. We end our brief introduction to matrix factorization techniques with Concept Factorization (CF) [87], *i.e.* a matrix factorization method originally introduced for document clustering that can be regarded as an auto-encoder model for the data matrix, where the encoding and decoding steps are linear transformations. The resulting factorization is similar to NMF, but can also potentially deal with non-negative data matrices. Recently, a regularized variant of CF has been introduced [50], which takes into account the local, geometrical data structure in kernel space.

A matrix undergoing a factorization in learning algorithms typically arises as a collection of feature vectors, each vector being a column of the matrix, or as a representation of the similarity/dissimilarity relations among data objects. In the latter case, we can easily interpret the matrix as the adjacency matrix of a weighted graph having the data objects as vertices. The application of matrix factorization

for the analysis of graphs is mainly restricted to clustering [45, 72]. This paper aims at giving a novel viewpoint about the role of matrix factorization in the analysis of graphs and, in the specific, we show how matrix factorization can serve the purpose of compressing a graph.

Compressing data consists in changing its representation in a way to require fewer bits. Depending on the reversibility of this encoding process we might have a lossy or lossless compression. Information-theoretic works on compressing graphical structures have recently appeared [13]. However, they do not focus on preserving a graph structure as the compressed representation, which is instead what we aim at in our graph compression model. Our work is instead closer in spirit to [57], which proposes a summarization algorithm for unweighted graphs, and [84], which proposes a greedy procedure to determine a set of supernodes and superedges to approximate a weighted graph. Moreover, our work is related to the Szemerédi regularity lemma [80], a well-known result in extremal graph theory, which roughly states that a dense graph can be approximated by a bounded number of random bipartite graphs. An algorithmic version of this lemma has been used for speeding-up a pairwise clustering algorithm in [78].

A problem linked to graph compression that has focused the attention of researchers in the network and sociometric literature for the last few decades is *block-modeling* [37, 51]. Block models try to group the graph vertices into groups that preserve a *structural equivalence*, *i.e.* vertices falling in the same group should exhibit similar relations to the nodes in other groups (including self-similarity), and they differ by the way in which structural equivalence is defined. We refer to [29] for an overview of blockmodels and to [2] for recent developments on mixed-membership stochastic block models.

In this thesis we provide a novel viewpoint about the role of matrix factorization in the analysis of graphs and, in the specific, for the purpose of compressing a graph. The solution that we propose to compress a graph can be regarded as a block-model, where blocks and their relationships can be determined using a matrix factorization approach. Moreover, the model we propose extends our preliminary study in [59] to deal with incomplete observation of the graph (or kernel) to be compressed. The main contributions of the this chapter of thesis are the following:

i) we link matrix factorization with graph compression by proposing a factorization that can be used to reduce the order of a graph and can be employed also in the presence of incomplete observations. We show that the same tech-

nique can be used to compress a kernel, by retaining a kernel as the reduced representation;

ii) we cast the discrete problem of finding the best factorization into a continuous optimization problem for which we formally prove the equivalence between the discrete and continuous formulations;

iii) we provide a novel algorithm to approximately find the proposed factorization, which resembles the NMF algorithm in [48] (under ℓ_2 divergence) and the Baum-Eagon dynamics [5]. Additionally, we formally prove convergence properties for our algorithm;

iv) finally, we establish a relation between clustering and our graph compression model and show that existing clustering approaches in the literature can be regarded as particular, *constrained* variants of our matrix factorization.

The rest of the manuscript is organized as follows. Section 4.2 introduces some notation and basic definitions used in the work. Section 4.3 introduces our graph compression formulation and its applicability also in the context of kernel compression. Section 4.4 addresses the problem of computing the factorization by proposing an algorithm and by providing some convergence guarantees. Section 4.6 establishes relations between the problem of clustering and our graph compression formulation. Section 4.7 is devoted to showing the effectiveness of our graph compression algorithm on both synthetic and real-world data-sets. Finally, we provide some concluding discussion and plans for future developments.

4.2 Preliminaries

We present here the notation and basic definitions adopted throughout the thesis.

General. We denote *vectors* with bold lowercase letters (*e.g.* $\boldsymbol{x}$, $\boldsymbol{y}$), *matrices* with uppercase typewriter-style letters (*e.g.* $\mathbf{A}$, $\mathbf{B}$), *sets* with calligraphic-style uppercase letters (*e.g.* $\mathcal{X}$, $\mathcal{Y}$). The (i,j)th element of matrix $\mathbf{A}$ is denoted by $\mathbf{A}_{ij}$, whereas the ith element of a vector $\boldsymbol{v}$ is denoted by v_i. We write *indices* with lowercase letters (*e.g.* i, j, k, h) and *constants* with lowercase serif-style letters (*e.g.* n, m). We denote by $\mathbb{1}_P$ the indicator function giving 1 if proposition P is true, 0 otherwise. The sets of real and non-negative real numbers are given by $\mathbb{R}$ and $\mathbb{R}_+$ as usual. We denote by $[\mathsf{n}]$ the set $\{1,\dots,\mathsf{n}\}$.

The pseudo-inverse x^+ of a scalar $x \in \mathbb{R}$ returns $1/x$ if $x \neq 0$ and 0 otherwise. The *identity* matrix is denoted by $\mathbf{I}$. The vector of all 1s of size k is denoted by $\mathbf{1}_\mathsf{k}$, the size being omitted where unambiguous. The matrix of all 1s is denoted by $\mathbf{E}$.

Let $\mathbf{A}$ and $\mathbf{B}$ be two $\mathsf{n} \times \mathsf{m}$ matrix. The *transposition* of $\mathbf{A}$ is denoted as $\mathbf{A}^\top$. The *inverse* of a square matrix $\mathbf{A}$ is denoted by $\mathbf{A}^{-1}$. The set of *left-stochastic* $\mathsf{k} \times \mathsf{n}$ matrices is given by $\mathcal{S} = \{\mathbf{X} \in \mathbb{R}_+^{\mathsf{k}\times\mathsf{n}} : \mathbf{X}^\top\mathbf{1}_\mathsf{k} = \mathbf{1}_\mathsf{n}\}$ and we denote by $\mathcal{S}_{01} = \mathcal{S} \cap \{0,1\}^{\mathsf{k}\times\mathsf{n}}$ the set of left-stochastic binary matrices. Given $\mathbf{A}$ and $\mathbf{B}$ square, the notation $\mathbf{A} \succcurlyeq \mathbf{B}$ means that $(\mathbf{A}-\mathbf{B})$ is positive semidefinite, *i.e.* $\sum_{ij\in[\mathsf{n}]}(\mathbf{A}_{ij} - \mathbf{B}_{ij})x_i x_j \geq 0$ for all $\boldsymbol{x} \in \mathbb{R}^\mathsf{n}$

4.3 Matrix factorization for graph compression

An (edge-weighted) *graph* is a triplet $(\mathcal{V}, \mathcal{E}, \omega)$ where $\mathcal{V} = [\mathsf{n}]$ is the set of n vertices, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges and $\omega : \mathcal{E} \to \mathbb{R}$ is a function providing each edge with a weight. We refer to n as the *order* of the graph. For the sake of simplicity, we consider the graph to be complete and model missing edges as 0-weighted edges. Moreover, the graph is assumed to be undirected and therefore $\omega(i,j) = \omega(j,i)$ for all $(i,j) \in \mathcal{E}$. A graph $(\mathcal{V}, \mathcal{E}, \omega)$ can thus be encoded as a symmetric matrix $\mathbf{G} \in \mathbb{R}^{\mathsf{n}\times\mathsf{n}}$, where $\mathbf{G}_{ij} = \mathbb{1}_{(i,j)\in\mathcal{E}}\omega(i,j)$ or.

$$\mathbf{G}_{ij} = \begin{cases} \omega(i,j) & \text{if } (i,j) \in \mathcal{E} \\ 0 & \text{otherwise.} \end{cases}$$

Hereafter, we might refer to $\mathbf{G}$ both as a graph or as a matrix depending on the context.

Let $\mathbf{G} \in \mathbb{R}^{\mathsf{n}\times\mathsf{n}}$ be a graph and let $\mathsf{k} \leq \mathsf{n}$ be a constant representing the order of the new graph that we expect after the compression. The rate $\frac{\mathsf{k}}{\mathsf{n}}$ is regarded as the *graph compression rate*. To reduce the order of graph $\mathbf{G}$ from n vertices to $\mathbf{k}$ vertices, we have to determine the compressed graph (or *reduced graph*) $\mathbf{R} \in \mathbb{R}^{\mathsf{k}\times\mathsf{k}}$, and a mapping $\psi : [\mathsf{n}] \to [\mathsf{k}]$ between vertices of the original graph $\mathbf{G}$ and the reduced one $\mathbf{R}$. The mapping ψ relates each vertex i of $\mathbf{G}$ to a vertex $\psi(i)$ of $\mathbf{R}$. Since ψ is in general a *many-to-one* mapping, *i.e.* such that multiple vertices of $\mathbf{G}$ can potentially be assigned to the same vertex of $\mathbf{R}$, we have a lossless compression only when the relation

$$\mathbf{G}_{ij} = \mathbf{R}_{\psi(i)\psi(j)} \tag{4.1}$$

holds for all $i, j \in [\mathsf{n}]$. In real-world scenarios, the graph might not be fully observed, but only a subset of the entries in $\mathbf{G}$ might be available. To cope with this eventuality, we assume that only a subset $\mathcal{O} \subseteq [\mathsf{n}] \times [\mathsf{n}]$ of entries (not restricted to graph edges) is observed. Due to the symmetry of $\mathbf{G}$, we assume $\mathcal{O}$ to be symmetric as well, *i.e.* $(i, j) \in \mathcal{O} \implies (j, i) \in \mathcal{O}$. If only partial information about the graph is available, we relax the requirements for a "lossless" compression, by imposing that (4.1) should hold just for all $(i, j) \in \mathcal{O}$. From now on, we assume without loss of generality that only partial information about the graph is available and that $\mathcal{O}$ encodes the index set of the observed entries.

A many-to-one mapping ψ can be expressed as a left-stochastic binary matrix $\mathbf{X} \in \mathcal{S}_{01}$ where By expressing the many-to-one mapping ψ as a binary matrix $\mathbf{X} \in \{0, 1\}^{\mathsf{k}\times\mathsf{n}}$, having columns summing up to 1, *i.e.* $\mathbf{X}^\top \mathbf{1}_{\mathsf{k}} = \mathbf{1}_{\mathsf{n}}$, where each entry $\mathbf{X}_{kj} = \mathbb{1}_{\psi(j)=k}$ indicates whether vertex j of $\mathbf{G}$ should be assigned to vertex k of $\mathbf{R}$. On the other hand, every left-stochastic binary matrix $\mathbf{X} \in \mathcal{S}_{01}$ encodes the many-to-one mapping $\psi(i) = \arg\max_k \mathbf{X}_{kj}$. By replacing the mapping ψ with its matrix-form counterpart $\mathbf{X}$, we can compactly rewrite the condition for a lossless compression (with partial observations $\mathcal{O}$) in (4.1) in terms of the factorization

$$\mathbf{G} =_{\mathcal{O}} \mathbf{X}^\top \mathbf{R} \mathbf{X} \,. \tag{4.2}$$

In general, a lossless factorization for a graph $\mathbf{G}$ at some fixed compression rate might not exist. In this case, we resort to a least-squares approximation, *i.e.* we look for a minimizer of

$$\begin{aligned} \min \quad & g(\mathbf{X}, \mathbf{R}) = \|\mathbf{G} - \mathbf{X}^\top \mathbf{R} \mathbf{X}\|_{\mathcal{O}}^2 \\ \text{s.t.} \quad & \mathbf{X} \in \mathcal{S}_{01} \,, \quad \mathbf{R} \in \mathbb{R}^{\mathsf{k}\times\mathsf{k}} \,. \end{aligned} \tag{4.3}$$

The proposed factorization allows to reduce the complexity of the graph representation from n^2, *i.e.* the space complexity of $\mathbf{G}$, to $(\mathsf{n}+\mathsf{k}^2)$, *i.e.* the effective space complexity of $(\mathbf{X}, \mathbf{R})$. In fact, since matrix $\mathbf{X}$ is the matrix-form of the mapping ψ, it can be stored more compactly as a n-dimensional vector, the ith component begin index $\psi(i)$. Additionally, it has interesting properties which render it suitable, *e.g.* as a kernel compression strategy. Indeed, as a consequence of property 2 of Theorem 11, if $\mathbf{G}$ is a kernel and we have a complete set of observations (*i.e.* $\mathcal{O} = [\mathsf{n}] \times [\mathsf{n}]$) then also $\mathbf{R}$ is a kernel.

Theorem 11. *Let* $(\mathbf{X}, \mathbf{R})$ *be a local solution of* (4.3) *with* $\mathcal{O} = [\mathsf{n}] \times [\mathsf{n}]$ *and assume* $\mathbf{X}$ *to have full rank. Then the following properties hold for any* $s \in \{-1, +1\}$*:*

1. $s\mathbf{G} \in \mathbb{R}_+^{\mathsf{n}\times\mathsf{n}} \implies s\mathbf{R} \in \mathbb{R}_+^{\mathsf{k}\times\mathsf{k}}$,
2. $s\mathbf{G} \succcurlyeq \mathbf{0} \implies s\mathbf{R} \succcurlyeq \mathbf{0}$.

Proof. By setting to zero the derivative of (4.3) with respect to $\mathbf{R}$ we obtain the equation $\mathbf{X}\mathbf{G}\mathbf{X}^\top = \mathbf{X}\mathbf{X}^\top\mathbf{R}\mathbf{X}\mathbf{X}^\top$ which must be satisfied by any local solution. Since $\mathbf{X} \in \mathcal{S}_{01}$ has rank k, matrix $\mathbf{X}\mathbf{X}^\top$ is invertible. By left and right multiplying both sides of the previous equation by $(\mathbf{X}\mathbf{X}^\top)^{-1}$ we obtain $(\mathbf{X}\mathbf{X}^\top)^{-1}\mathbf{X}\mathbf{G}\mathbf{X}^\top(\mathbf{X}\mathbf{X}^\top)^{-1} = \mathbf{R}$. From this identity and assuming $s\mathbf{G} \succcurlyeq \mathbf{0}$ we can see that $\boldsymbol{y}^\top(s\mathbf{R})\boldsymbol{y} = \boldsymbol{z}^\top(s\mathbf{G})\boldsymbol{z} \geq 0$ for all $\boldsymbol{y} \in \mathbb{R}^{\mathsf{k}}$, where $\boldsymbol{z} = \mathbf{X}^\top(\mathbf{X}\mathbf{X}^\top)^{-1}\boldsymbol{y}$. Hence, Property 1 holds for any $\mathbf{X} \in \mathcal{S}_{01}$. Moreover also Property 2 holds, since $(\mathbf{X}\mathbf{X}^\top)^{-1}\mathbf{X}$ is a non-negative matrix for any $\mathbf{X} \in \mathcal{S}_{01}$. □ □

The assumption in Theorem 11 of $\mathbf{X}$ being of full rank is not restrictive. Indeed, if $\mathbf{X}$ is rank deficient then it has at least one null row. By removing the null rows of $\mathbf{X}$ and by dropping the corresponding rows and columns of $\mathbf{R}$, we fall back to the case where we have full rank without affecting the quality of the approximation. Note also that property 1 holds, more in general, for any choice of $\mathcal{O}$, while it is not necessarily true for property 2.

4.4 A tight relaxation of the graph compression objective

The matrix factorization in (4.2) can be regarded as a particular quadratic factorization, where the factor matrix $\mathbf{X}$ is left-stochastic and binary, and $\mathbf{R}$ is unconstrained. If we relax the binary constraint to a non-negativity constraint, we obtain a quadratic matrix factorization that has been studied in [89] and for which update rules have been indirectly provided.[1] By using them, one could potentially find a solution to (4.3) with $\mathcal{S}_{01}$ relaxed to $\mathcal{S}$. However, by doing so, there is no guarantee that the solution to the relaxed problem (once projected) will be also solution to the original factorization problem, *i.e.* the relaxation is *not* tight. Moreover, the update rules in [89] are not interior-point methods in the sense that feasibility is not guaranteed at intermediate steps. In this section, we provide an alternative relaxation to the graph compression objective, which is not per se in the form of a matrix factorization problem, but it is *tight* as it allows to recover the solution to (4.3). Moreover, the update rule we propose in Section 4.5 for $\mathbf{Y}$ is invariant to $\mathcal{S}$, *i.e.* every iteration yields a feasible solution.

Instead of optimizing (4.3), which exhibits a non-convex feasible set, we optimize a relaxation of the same program, where we drop the integer constraint. In the specific, we replace the variable $\mathbf{X} \in \mathcal{S}_{01}$ with a left-stochastic real matrix $\mathbf{Y} \in \mathcal{S}$. This replacement is performed after some algebraic calculations which allow to reduce the degree of the polynomial being the objective function. This yields the following program:

$$\begin{aligned} \min \quad & f(\mathbf{Y}, \mathbf{R}) \\ \text{s.t.} \quad & \mathbf{Y} \in \mathcal{S}\,, \quad \mathbf{R} \in \mathbb{R}^{\mathsf{k}\times\mathsf{k}}\,, \end{aligned} \tag{4.4}$$

where

$$f(\mathbf{Y}, \mathbf{R}) = \sum_{(i,j)\in\mathcal{O}} \sum_{k,h\in[\mathsf{k}]} \mathbb{1}_{(k,i)\neq(h,j)} \mathbf{Y}_{ki}\mathbf{Y}_{hj}(\mathbf{G}_{ij} - \mathbf{R}_{kh})^2 + \sum_{(i,i)\in\mathcal{O}} \sum_{k\in[\mathsf{k}]} \mathbf{Y}_{ki}(\mathbf{G}_{ii} - \mathbf{R}_{kk})^2\,.$$

The newly introduced optimization program satisfies a strong property provided in Theorem 12, which suggests that (4.4) is a tight relaxation of (4.3) and that minimizing (4.4) is equivalent to minimizing (4.3).

[1] there is no explicit update rule in the thesis to find the specific *relaxed* factorization, but it could be obtained by using Theorem 1.

Theorem 12. *Let* $\mathbf{X} \in \mathcal{S}_{01}$, $\mathbf{Y} \in \mathcal{S}$ *and* $\mathbf{R} \in \mathbb{R}^{\mathsf{k}\times\mathsf{k}}$. *If* $(\mathbf{X}, \mathbf{R})$ *is a minimizer of* (4.3) *then* $(\mathbf{X}, \mathbf{R})$ *is a minimizer of* (4.4). *If* $(\mathbf{Y}, \mathbf{R})$ *is a minimizer of* (4.4) *then* $(\Pi(\mathbf{Y}), \mathbf{R})$ *is a minimizer of* (4.3), *for any projection* $\Pi : \mathcal{S} \to \mathcal{S}_{01}$ *satisfying* $(\mathbf{Y}_{ki} = 0) \Rightarrow (\Pi(\mathbf{Y})_{ki} = 0)$ *for all* $k \in [\mathsf{k}]$ *and* $i \in [\mathsf{n}]$.

Proof. We start showing that for any $\mathbf{R} \in \mathbb{R}^{\mathsf{k}\times\mathsf{k}}$ and any $\mathbf{X} \in \mathcal{S}_{01}$ it holds that $g(\mathbf{X}, \mathbf{R}) = f(\mathbf{X}, \mathbf{R})$. Let ψ be the mapping encoded by $\mathbf{X}$ as described in Section 4.3. Then $g(\mathbf{X}, \mathbf{R})$ can be written as

$$\begin{aligned} g(\mathbf{X}, \mathbf{R}) &= \sum_{(i,j)\in\mathcal{O}} (\mathbf{G}_{ij} - \mathbf{R}_{\psi(i)\psi(j)})^2 \\ &= \sum_{(i,j)\in\mathcal{O}} \sum_{k,h\in[\mathsf{k}]} \mathbb{1}_{\psi(i)=k}\mathbb{1}_{\psi(j)=h}(\mathbf{G}_{ij} - \mathbf{R}_{kh})^2 \\ &= \sum_{(i,j)\in\mathcal{O}} \sum_{k,h\in[\mathsf{k}]} \mathbf{X}_{ki}\mathbf{X}_{hj}(\mathbf{G}_{ij} - \mathbf{R}_{kh})^2 \,. \end{aligned}$$

By noting that $\mathbf{X}_{ki}^2 = \mathbf{X}_{ki}$ we decompose the summation as follows:

$$\begin{aligned} g(\mathbf{X}, \mathbf{R}) = \sum_{(i,j)\in\mathcal{O}} \sum_{k,h\in[\mathsf{k}]} \mathbb{1}_{(k,i)\neq(h,j)}\mathbf{X}_{ki}\mathbf{X}_{hj}(\mathbf{G}_{ij} - \mathbf{R}_{kh})^2 \\ + \sum_{(i,i)\in\mathcal{O}} \sum_{k\in[\mathsf{k}]} \mathbf{X}_{ki}(\mathbf{G}_{ii} - \mathbf{R}_{kk})^2 = f(\mathbf{X}, \mathbf{R}) \,. \end{aligned} \tag{4.5}$$

($\Rightarrow$) To prove the first implication of the theorem, assume $(\mathbf{X}, \mathbf{R})$ to be a minimizer of (4.3) and let $(\mathbf{Y}, \mathbf{T})$ be a minimizer of (4.4), where $\mathbf{Y} \in \mathcal{S}$. Moreover, let $\mathbb{X} \in \mathcal{S}_{01}$ be a random variable generating a left-stochastic binary matrix according to the probabilities defined in $\mathbf{Y}$. Specifically, each ith column of the outcome matrix can be regarded as an independent realization from a categorical distribution parametrized by the ith column of $\mathbf{Y}$. Then, by (4.5), by noting that the expectation $\mathbb{E}[\cdot]$ and the function $f(\cdot, \mathbf{R})$ do commute, and since $(\mathbf{X}, \mathbf{R})$ is a minimizer of (4.3), we have that

$$f(\mathbf{X}, \mathbf{R}) = g(\mathbf{X}, \mathbf{R}) \leq \mathbb{E}[g(\mathbb{X}, \mathbf{T})] = \mathbb{E}[f(\mathbb{X}, \mathbf{T})] = f(\mathbb{E}[\mathbb{X}], \mathbf{T}) = f(\mathbf{Y}, \mathbf{T}) \,. \tag{4.6}$$

Since $(\mathbf{Y}, \mathbf{T})$ was assumed to be a minimizer of (4.4), the previous relation implies that necessarily $f(\mathbf{X}, \mathbf{R}) = f(\mathbf{Y}, \mathbf{T})$ and therefore also $(\mathbf{X}, \mathbf{R})$ must be a minimizer of (4.4).

($\Leftarrow$) To prove the other implication of the theorem, assume $(\mathbf{Y}, \mathbf{R})$ to be a minimizer of (4.4) and let $(\mathbf{X}, \mathbf{T})$ be a minimizer of (4.3). Then, similarly to (4.6), we have that $f(\mathbf{Y}, \mathbf{R}) = \mathbb{E}[g(\mathbb{X}, \mathbf{R})] \geq g(\mathbf{X}, \mathbf{T}) = f(\mathbf{X}, \mathbf{T})$. This implies that $\mathbb{E}[g(\mathbb{X}, \mathbf{R})] = g(\mathbf{X}, \mathbf{T})$ since $(\mathbf{Y}, \mathbf{R})$ is assumed to be a minimizer of (4.4) and therefore $g(\mathbf{Z}, \mathbf{R}) = g(\mathbf{X}, \mathbf{T})$ for all $\mathbf{Z} \in \mathcal{S}_{01}$ such that $\mathbb{P}[\mathbb{X} = \mathbf{Z}] > 0$. Now, since $\mathbb{P}[\mathbb{X} = \Pi(\mathbf{Y})] > 0$ by construction, it follows that $(\Pi(\mathbf{Y}), \mathbf{R})$ is a minimizer of (4.3). □ □

According to Theorem 12, we can focus on the optimization of (4.4), which has a continuous objective and domain, in place of (4.3), for any solution $\mathbf{Y}^*$ of the former can be cast into a solution $\mathbf{X}^*$ of the latter by applying a projection as defined by the theorem, *i.e.* $\mathbf{X}^* = \Pi(\mathbf{Y}^*)$. As an example of such a projection, we could replace each column of $\mathbf{Y}^*$ with an indicator vector for the largest element in the column to obtain $\mathbf{X}^*$.

4.5 Graph compression algorithm

We propose an optimization approach for (4.4), which alternates updates of the variable $\mathbf{R}$ and updates of the variable $\mathbf{Y}$. Both updates are shown to deliver a decrease of the objective function until a stationary point of (4.4) is reached.

4.5.1 Update rule for R.

Since $\mathbf{R}$ is an unconstrained matrix variable, we derive an update rule for it by setting the first-order partial derivative of f with respect to its second argument $\mathbf{R}$ to zero (see, proof of Theorem 13 for details). This yields

$$U_{\mathbf{R}}(\mathbf{Y})_{kh} = (\mathbf{L}_{hk})^{+} \left[\sum_{(i,j)\in\mathcal{O}} \mathbb{1}_{(k,i)\neq(h,j)} \mathbf{Y}_{ki}\mathbf{Y}_{hj}\mathbf{G}_{ij} + \mathbb{1}_{k=h} \sum_{(i,i)\in\mathcal{O}} \mathbf{Y}_{ki}\mathbf{G}_{ii} \right], \tag{4.7}$$

where $\mathbf{L}$ is defined as

$$\mathbf{L}_{kh} = \sum_{(i,j)\in\mathcal{O}} \mathbb{1}_{(k,i)\neq(h,j)} \mathbf{Y}_{ki}\mathbf{Y}_{hj} + \mathbb{1}_{k=h} \sum_{(i,i)\in\mathcal{O}} \mathbf{Y}_{ki}\,. \tag{4.8}$$

This update rule has complexity $O(\mathsf{k}^2|\mathcal{O}|)$ and guarantees a monotonic decrease of the objective function f as stated by the following theorem.

Theorem 13. *Let* $\mathbf{T} \in \mathbb{R}^{\mathsf{k}\times\mathsf{k}}$ *and* $\mathbf{Y} \in \mathcal{S}$. *Then*

$$f(\mathbf{Y}, \mathbf{T}) \geq f\left(\mathbf{Y}, U_{\mathbf{R}}(\mathbf{Y})\right)$$

with equality if and only if $\mathbf{T}$ *is a minimizer of* $f(\mathbf{Y}, \cdot)$.

Proof. Since $f(\mathbf{Y}, \cdot)$ is a quadratic form and it is lower bounded by 0, we can prove the result by simply showing that

$$\frac{\partial f}{\partial \mathbf{R}_{kh}}(\mathbf{Y}, U_{\mathbf{R}}(\mathbf{Y})) = 0\,. \tag{4.9}$$

Indeed, if this is true, then $U_{\mathbf{R}}(\mathbf{Y})$ will be a global minimizer of $f(\mathbf{Y}, \cdot)$ and therefore $f(\mathbf{Y}, \mathbf{T}) = f(\mathbf{Y}, U_{\mathbf{R}}(\mathbf{Y}))$ holds true if and only if $\mathbf{T}$ is a global minimizer of $f(\mathbf{Y}, \cdot)$ as well.

The partial derivative of f with respect to its second argument $\mathbf{R}$ is given by

$$\begin{aligned}\frac{\partial f}{\partial \mathbf{R}_{kh}}(\mathbf{Y},\mathbf{R}) &= 2\sum_{(i,j)\in\mathcal{O}} \mathbb{1}_{(k,i)\neq(h,j)}\mathbf{Y}_{ki}\mathbf{Y}_{hj}(\mathbf{R}_{kh}-\mathbf{G}_{ij}) \\ &\quad + 2\sum_{(i,i)\in\mathcal{O}} \mathbb{1}_{k=h}\sum_{k\in[\mathsf{k}]}\mathbf{Y}_{ki}(\mathbf{R}_{kh}-\mathbf{G}_{ii}) \\ &= 2\mathbf{R}_{kh}\mathbf{L}_{kh} - 2\sum_{(i,j)\in\mathcal{O}} \mathbb{1}_{(k,i)\neq(h,j)}\mathbf{Y}_{ki}\mathbf{Y}_{hj}\mathbf{G}_{ij} \\ &\quad - 2\sum_{(i,i)\in\mathcal{O}} \mathbb{1}_{k=h}\sum_{k\in[\mathsf{k}]}\mathbf{Y}_{ki}\mathbf{G}_{ii}\,,\end{aligned}$$

where $\mathbf{L}$ is defined as in (4.8). By expanding the derivative in (4.9), by isolating the term with $U_{\mathbf{R}(\mathbf{Y})}$ on the left-hand-side of the equation, and after halving both sides, we get

$$[U_{\mathbf{R}}(\mathbf{Y})]_{kh}\mathbf{L}_{kh} = \sum_{(i,j)\in\mathcal{O}} \mathbb{1}_{(k,i)\neq(h,j)}\mathbf{Y}_{ki}\mathbf{Y}_{hj}\mathbf{G}_{ij} + \mathbb{1}_{k=h}\sum_{(i,i)\in\mathcal{O}}\mathbf{Y}_{ki}\mathbf{G}_{ii}\,. \tag{4.10}$$

Clearly, by definition of $U_{\mathbf{R}}(\mathbf{Y})$, the equality holds true if $\mathbf{L}_{kh} \neq 0$. In the other case, $\mathbf{L}_{kh} = 0$ only if $\mathbf{Y}_{ki}\mathbf{Y}_{hj} = 0$ for all $(i,j) \in \mathcal{O}$. This in turn implies that the right-hand-side of (4.8) is zero as well. Hence, if $\mathbf{L}_{kh} = 0$ then (4.10) is always satisfied and in particular it is satisfied by the update rule (4.7), which would deliver $U_{\mathbf{R}}(\mathbf{Y})_{kh} = 0$, for $(\mathbf{L}_{kh})^+ = 0^+ = 0$. □ □

4.5.2 Update rule for Y.

We derive here a multiplicative update rule for $\mathbf{Y}$, which resembles the one proposed in [48] for Non-negative Matrix Factorization (NMF) under ℓ_2 divergence and the one related to the Baum-Eagon inequality [5].

Let $\mathbf{R} \in \mathbb{R}^{\mathsf{k}\times\mathsf{k}}$ and $\mathbf{Z} \in \mathcal{S}$ and consider the following optimization problem:

$$\begin{aligned}\min \quad & f_{\mathbf{R}}(\mathbf{Y}) = f(\mathbf{Y},\mathbf{R}) \\ \text{s.t.} \quad & \mathbf{Y} \in \mathcal{Y}_{\mathbf{Z}}\end{aligned} \tag{4.11}$$

where $\mathcal{Y}_{\mathbf{Z}} = \{\mathbf{Y} \in \mathcal{S} : (\mathbf{Z}_{ki} = 0) \Rightarrow (\mathbf{Y}_{ki} = 0)\}$. We say that $\mathbf{Y}$ is a *Karush-Kuhn-Tucker (KKT-)point* for (4.11) if it satisfies the first-order necessary conditions for

local optimality, which are given by:

$$\frac{\partial f_{\mathbf{R}}}{\partial \mathbf{Y}_{ki}}(\mathbf{Y}) - \gamma_i - \mu_{ki} - \alpha_{ki}\mathbb{1}_{\mathbf{Z}_{ki}=0} = 0\,, \tag{4.12}$$

$$\mu_{ki}\mathbf{Y}_{ki} = 0\,, \tag{4.13}$$

$$\sum_{h\in[\mathsf{k}]} \mathbf{Y}_{hi} = 1\,, \tag{4.14}$$

$$\mathbf{Y}_{hi}\mathbb{1}_{\mathbf{Z}_{ki}=0} = 0\,, \tag{4.15}$$

for some values of the Lagrangian multipliers $\gamma_i, \alpha_{ki} \in \mathbb{R}$ and $\mu_{ki} \in \mathbb{R}_+$, for all $k \in [\mathsf{k}]$ and $i \in [\mathsf{n}]$.

Let $\sigma_i(\mathbf{Z})$ be the support of the ith column of $\mathbf{Z}$, *i.e.* $\sigma_i(\mathbf{Z}) = \{k \in [\mathsf{k}] \,:\, \mathbf{Z}_{ki} > 0\}$, and let $\mathcal{Z}_i(\mathbf{Z}, \mathbf{R})$, $i \in [\mathsf{n}]$, denote the set

$$\mathcal{Z}_i(\mathbf{Z}, \mathbf{R}) = \left\{ k \in [\mathsf{k}] \,:\, \frac{\partial f_{\mathbf{R}}}{\partial \mathbf{Y}_{ki}}(\mathbf{Z}) = 0 \right\} \cap \sigma_i(\mathbf{Z})\,.$$

The following update rule $U_{\mathbf{Y}}(\mathbf{Z}, \mathbf{R})$ for $\mathbf{Y}$ guarantees a strict decrease of the objective function in (4.11) unless $\mathbf{Z}$ is a KKT-point (see Theorem 14):

$$U_{\mathbf{Y}}(\mathbf{Z}, \mathbf{R})_{ki} = \begin{cases} 0 & \text{if } k \notin \sigma_i \\ \dfrac{\mathbf{Z}_{ki}}{\displaystyle\sum_{h\in\mathcal{Z}_i} \mathbf{Z}_{hi}} & \text{if } k \in \mathcal{Z}_i \\ \dfrac{\mathbf{Z}_{ki}}{\lambda_i \dfrac{\partial f_{\mathbf{R}}}{\partial \mathbf{Y}_{ki}}(\mathbf{Z})} & \text{if } \mathcal{Z}_i = \emptyset,\ k \in \sigma_i \end{cases} \tag{4.16}$$

where we wrote σ_i and $\mathcal{Z}_i$ for $\sigma_i(\mathbf{Z})$ and $\mathcal{Z}_i(\mathbf{Z}, \mathbf{R})$, and where

$$\lambda_i = \sum_{h\in\sigma_i} \mathbf{Z}_{hi} \frac{\partial f_{\mathbf{R}}}{\partial \mathbf{Y}_{hi}}(\mathbf{Z})^{-1}\,.$$

The complexity of this update rule is $O(k^2|\mathcal{O}|)$ (see (4.19) for the formula of the first-order partial derivative).

Theorem 14. *Let* $\mathbf{R} \in \mathbb{R}^{\mathsf{k}\times\mathsf{k}}$ *and* $\mathbf{Z} \in \mathcal{S}$. *Then*

$$f(\mathbf{Z}, \mathbf{R}) \geq f(U_{\mathbf{Y}}(\mathbf{Z}, \mathbf{R}), \mathbf{R})$$

with equality if and only if $\mathbf{Z}$ *is a KKT-point for* (4.11).

Before providing the proof of Theorem 14, we need the following lemma:

Lemma 6. *Let $\mathbf{Z} \in \mathcal{S}$ and consider the following function:*

$$\phi(\mathbf{Y}, \mathbf{Z}) = f_{\mathbf{R}}(\mathbf{Z}) + \sum_{i \in [\mathsf{n}]} \sum_{k \in [\mathsf{k}]} (\mathbf{Y}_{ki} - \mathbf{Z}_{ki}) \frac{\partial f_{\mathbf{R}}}{\partial \mathbf{Y}_{ki}}(\mathbf{Z}) + \frac{1}{2} \sum_{i \in [\mathsf{n}]} \sum_{k \in [\mathsf{k}]} (\mathbf{Y}_{ki} - \mathbf{Z}_{ki})^2 \frac{\partial f_{\mathbf{R}}}{\partial \mathbf{Y}_{ki}}(\mathbf{Z}) \mathbf{Z}^{+}_{ki} \,. \quad (4.17)$$

Then $\phi(\mathbf{Y}, \mathbf{Z}) \geq f_{\mathbf{R}}(\mathbf{Y})$ for all $\mathbf{Y} \in \mathcal{Y}_{\mathbf{Z}}$.

Proof. By replacing $f_{\mathbf{R}}$ with its Taylor expansion about $\mathbf{Z}$ we obtain after simple algebraic manipulations:

$$\phi(\mathbf{Y}, \mathbf{Z}) - f_{\mathbf{R}}(\mathbf{Y}) = \frac{1}{2} \sum_{i \in [\mathsf{n}]} \sum_{k \in [\mathsf{k}]} (\mathbf{Y}_{ki} - \mathbf{Z}_{ki})^2 \frac{\partial f_{\mathbf{R}}}{\partial \mathbf{Y}_{ki}}(\mathbf{Z}) \mathbf{Z}^{+}_{ki} - \frac{1}{2} \sum_{i,j \in [\mathsf{n}]} \sum_{k,h \in [\mathsf{k}]} (\mathbf{Y}_{ki} - \mathbf{Z}_{ki})(\mathbf{Y}_{hj} - \mathbf{Z}_{hj}) \frac{\partial^2 f_{\mathbf{R}}}{\partial \mathbf{Y}_{ki} \partial \mathbf{Y}_{hj}}(\mathbf{Z}) \,, \quad (4.18)$$

where the first-order and second-order partial derivatives are given by

$$\frac{\partial f_{\mathbf{R}}}{\partial \mathbf{Y}_{ki}}(\mathbf{Z}) = \mathbb{1}_{(i,i) \in \mathcal{O}} (\mathbf{G}_{ii} - \mathbf{R}_{kk})^2 + \sum_{j \in [\mathsf{n}]} \sum_{h \in [\mathsf{k}]} \mathbb{1}_{(i,j) \in \mathcal{O}} \mathbb{1}_{(k,i) \neq (h,j)} \mathbf{Z}_{hj} \left(\mathbf{G}_{ij} - \mathbf{R}_{kh}\right)^2 \geq \sum_{j \in [\mathsf{n}]} \sum_{h \in [\mathsf{k}]} \mathbb{1}_{(i,j) \in \mathcal{O}} \mathbb{1}_{(k,i) \neq (h,j)} \mathbf{Z}_{hj} \left(\mathbf{G}_{ij} - \mathbf{R}_{kh}\right)^2 \,, \quad (4.19)$$

and

$$\frac{\partial^2 f_{\mathbf{R}}}{\partial \mathbf{Y}_{ki} \partial \mathbf{Y}_{hj}}(\mathbf{Z}) = \mathbb{1}_{(i,j) \in \mathcal{O}} \mathbb{1}_{(k,i) \neq (h,j)} (\mathbf{G}_{ij} - \mathbf{R}_{kh})^2 \,, \quad (4.20)$$

respectively.

Let $\mathbf{Q}_{ki} = (\mathbf{Y}_{ki} - \mathbf{Z}_{ki})\mathbf{Z}^{+}_{ki}$ and note that $\mathbf{Q}_{ki}\mathbf{Z}_{ki} = (\mathbf{Y}_{ki} - \mathbf{Z}_{ki})$ for all $\mathbf{Y} \in \mathcal{Y}_{\mathbf{Z}}$, because $\mathbf{Z}_{ki} = 0$ implies $\mathbf{Y}_{ki} = 0$ by definition and $\mathbf{Z}_{ki}\mathbf{Z}^{+}_{ki} = \mathbb{1}_{\mathbf{Z}_{ki} > 0}$. Accordingly, we can rewrite (4.18) as

$$\phi(\mathbf{Y}, \mathbf{Z}) - f_{\mathbf{R}}(\mathbf{Y}) = \frac{1}{2} \sum_{i \in [\mathsf{n}]} \sum_{k \in [\mathsf{k}]} \mathbf{Q}^2_{ki} \mathbf{Z}_{ki} \frac{\partial f_{\mathbf{R}}}{\partial \mathbf{Y}_{ki}}(\mathbf{Z}) - \frac{1}{2} \sum_{i,j \in [\mathsf{n}]} \sum_{k,h \in [\mathsf{k}]} \mathbf{Q}_{ki} \mathbf{Q}_{hj} \mathbf{Z}_{ki} \mathbf{Z}_{hj} \frac{\partial^2 f_{\mathbf{R}}}{\partial \mathbf{Y}_{ki} \partial \mathbf{Y}_{hj}}(\mathbf{Z}) \,,$$

By substitution of (4.19) and (4.20) and by reorganizing the terms we obtain the following inequality

$$\phi(\mathbf{Y},\mathbf{Z}) - f_{\mathbf{R}}(\mathbf{Y}) \geq \frac{1}{2} \sum_{(i,j)\in\mathcal{O}} \sum_{k,h\in[\mathsf{k}]} \mathbf{Z}_{ki}\mathbf{Z}_{hj}\mathbb{1}_{(k,i)\neq(h,j)}(\mathbf{G}_{ij} - \mathbf{R}_{hk})^2 \left(\mathbf{Q}^2_{ki} - \mathbf{Q}_{ki}\mathbf{Q}_{hj}\right) .$$

By exploiting the symmetry of $\mathbf{G}$ and $\mathbf{R}$, we have

$$\phi(\mathbf{Y},\mathbf{Z}) - f_{\mathbf{R}}(\mathbf{Y}) \geq \sum_{(i,j)\in\mathcal{O}} \sum_{k,h\in[\mathsf{k}]} \mathbf{Z}_{ki}\mathbf{Z}_{hj}\mathbb{1}_{(k,i)\neq(h,j)} \cdot (\mathbf{G}_{ij} - \mathbf{R}_{hk})^2 \left(\frac{1}{2}\mathbf{Q}^2_{ki} - \mathbf{Q}_{ki}\mathbf{Q}_{hj} + \frac{1}{2}\mathbf{Q}^2_{hj}\right) \geq 0 . \quad (4.21)$$

where the last inequality follows from $(\frac{1}{2}\mathbf{Q}^2_{ki} - \mathbf{Q}_{ki}\mathbf{Q}_{hj} + \frac{1}{2}\mathbf{Q}^2_{hj}) = \frac{1}{2}(\mathbf{Q}_{ki} - \mathbf{Q}_{hj})^2 \geq 0$ and the non-negative of all other terms. □ □

Proof of Theorem 14. Let $\mathbf{Y}^* = U_{\mathbf{R}}(\mathbf{Z},\mathbf{R})$. We start proving that $\mathbf{Y}^*$ is a minimizer of

$$\min_{\mathbf{Y}\in\mathcal{Y}_{\mathbf{Z}}} \phi(\mathbf{Y},\mathbf{Z}) , \quad (4.22)$$

where ϕ is defined as in (4.17).

A minimizer $\mathbf{Y}$ of (4.22) must be a KKT-point for it. The conditions that should be satisfied are (4.13), (4.14), (4.15) and

$$\frac{\partial\phi}{\partial\mathbf{Y}_{ki}}(\mathbf{Y},\mathbf{Z}) - \gamma_i - \mu_{ki} - \alpha_{ki}\mathbb{1}_{\mathbf{Z}_{ki}=0} = 0 \quad (4.23)$$

for all $k \in [\mathsf{k}]$ and $i \in [\mathsf{n}]$, where $\gamma_i, \alpha_{ki} \in \mathbb{R}$ and $\mu_{ki} \in \mathbb{R}_+$ are the Lagrangian multipliers. It is immediate to see that (4.14) and (4.15) are always satisfied by $\mathbf{Y}^*$. If $\mathbf{Z}_{ki} = 0$ then (4.23) is trivially satisfied as α_{ki} can take any value, and, since by definition of $\mathbf{Y}^*$, $\mathbf{Z}_{ki} = 0$ implies $\mathbf{Y}^*_{ki} = 0$, also (4.13) is satisfied. On the other hand, if $\mathbf{Z}_{ki} > 0$ we have by definition of $\mathbf{Y}^*$ that $\mathbf{Y}^*_{ki} > 0$ and by taking $\mu_{ki} = 0$, we have that (4.13) is satisfied. Now, if $\mathcal{Z}_i = \emptyset$ then it is easy to verify, after substituting $\mathbf{Y}^*$ with its definition, that (4.23) is satisfied by setting $\gamma_i = \lambda_i^{-1}$, while if $\mathcal{Z}_i \neq \emptyset$ then (4.23) is satisfied by taking $\gamma_i = 0$. This proves that $\mathbf{Y}^*$ is a KKT-point for (4.22). Since (4.22) is a convex optimization problem, we have that any KKT-point is also a global minimizer and therefore $\mathbf{Y}^*$ is a minimizer of (4.22).

By this result and by Lemma 6 we have that

$$f(\mathbf{Z},\mathbf{R}) = f_{\mathbf{R}}(\mathbf{Z}) = \phi(\mathbf{Z},\mathbf{Z}) \geq \phi(\mathbf{Y}^*,\mathbf{Z}) \geq f_{\mathbf{R}}(\mathbf{Y}^*) = f(U_{\mathbf{R}}(\mathbf{Z},\mathbf{R}),\mathbf{R}) . \quad (4.24)$$

To conclude the proof, assume $\mathbf{Z}$ to be a KKT- point of (4.11). Trivially $\mathbf{Y}^*_{ki} = \mathbf{Z}_{ki}$ by definition of $\mathbf{Y}^*$, if $\mathbf{Z}_{ki} = 0$. Assume now $\mathbf{Z}_{ki} > 0$. Then by (4.12) we have that $\frac{\partial f_{\mathbf{R}}}{\partial \mathbf{Y}_{ki}}(\mathbf{Y}) = \gamma_i$. By exploring this fact in the definition of $\mathbf{Y}^*$ we can easily derive that also in this case $\mathbf{Y}^*_{ki} = \mathbf{Z}_{ki}$. Hence, $\mathbf{Y}^* = \mathbf{Z}$ and therefore (4.22) holds with equality. On the other hand, assume that (4.24) holds with equality. Then $f_{\mathbf{R}}(\mathbf{Z}) = \phi(\mathbf{Y}^*, \mathbf{Z})$. This implies that $\mathbf{Z}$ is a solution of (4.22) and a KKT-point for it. Hence, $\mathbf{Z}$ satisfies the conditions (4.23), (4.13), (4.14) and (4.15). Now, since $\frac{\partial \phi}{\mathbf{Y}_{ki}}(\mathbf{Z}, \mathbf{Z}) = \frac{\partial f_{\mathbf{R}}}{\partial \mathbf{Y}_{ki}}(\mathbf{Z})$, it follows that $\mathbf{Z}$ satisfies also (4.12) and therefore it is a KKT-point also for (4.11). □ □

4.5.3 Summary of the algorithm.

The proposed optimization procedure summarized in Algorithm 4.1 works as follows. We start with a random matrix $\mathbf{Y}$ lying in the interior of $\mathcal{S}$. We repeatedly alternate between updating $\mathbf{R}$ as $\mathbf{R} \leftarrow U_{\mathbf{R}}(\mathbf{Y})$ and $\mathbf{Y}$ as $\mathbf{Y} \leftarrow U_{\mathbf{Y}}(\mathbf{Y}, \mathbf{R})$. We iterate until some stopping criterion is met, *e.g.* distance between $\mathbf{Y}$s of two consecutive iterations is below a given threshold, maximum number of iterations reached, etc. According to Theorem 13 and 14, this procedure guarantees a strict decrease of the objective in (4.3) until a KKT-point is reached. This might not correspond to a local minima, but *e.g.* a saddle point. By inducing a small local perturbation and by restarting the procedure we can escape from a saddle point with high probability. In this way, we can ensure that a local solution will be reached. Finally, we project $\mathbf{Y}$ on $\mathcal{S}_{01}$ by means of a projection operator Π satisfying the properties stated in Theorem 12. As an example of Π, we can set to 1 in each column of $\mathbf{Y}$ the element having highest value (in the column) and put to 0 the rest. The complexity of the algorithm is $O(|\mathcal{O}|\mathsf{k}^2\beta)$ if we have partial information and $O(\mathsf{n}^2\mathsf{k}\beta)$ if we employ the full graph $\mathbf{G}$. Here, β denotes the expected number of iterations required to converge.

4.5.4 Graph reconstruction with incomplete observations

In Section 4.3 we envisaged the possibility of having only partial information about the graph, which translates into a restricted set $\mathcal{O}$ of observed edges. As we have shown, our graph compression algorithm can cope with this eventuality and yet deliver the reduced graph $\tilde{\mathbf{R}}$ and the corresponding mapping $\tilde{\mathbf{X}}$, begin solution of (4.3). The advantage of this feature is that we can provide a reconstructed version

```
input  : graph G
output: (X,R)
begin
    Y ← draw a random matrix from S
    while not stopping criterion met do
        R ← U_R(Y)              /* update R   */
        Y ← U_Y(Y,R)            /* update Y   */
    end
    X ← Π(Y)              /* project to S_01  */
end
```

(a)

Figure 4.1: Graph compression

$\tilde{\mathbf{G}}$ of the complete graph $\mathbf{G}$ by taking $\tilde{\mathbf{G}} = \tilde{\mathbf{X}}^\top \tilde{\mathbf{R}} \tilde{\mathbf{X}}$ and, as a by-product, we are able to predict the values of the non-observed entries. We show in the experimental section some results evaluating the quality of the reconstructed graph under partial observations.

4.6 Clustering as graph compression

Although clustering is not the main focus of this thesis, it is interesting to point out the relation that exists between our graph compression and clustering (see, [29] for a more comprehensive discussion specific to block-models). Under our model, clustering can be regarded as a graph compression if we interpret the vertices of the reduced graph as the clusters. From this perspective, the mapping $\mathbf{X}$ encodes the clustering result. Now, if we consider a cluster as Before coming to the experiment, it is worth mentioning that the notion of cluster under this setting differs from the commonly accepted one of a set of elements being mutually similar and dissimilar to elements belonging to other clusters, which is the commonly accepted definition, then the related compression should exhibit a reduced graph being loosely connected with strong weight on the self-loops, In other words, we expect $\mathbf{R} \approx \mathbf{I}$.

Interestingly, we found out that some clustering algorithms proposed in the literature can be regarded as constrained variants of our graph compression method. In [71] a consensus clustering approach has been proposed based on the optimization of a functional like (4.3) with the integer constraint dropped and under the condition $\mathbf{R} = \mathbf{I}$. A clustering model similar to this one has been proposed also for community detection in [58]. Finally, in [72] a pairwise clustering approach with probabilistic cluster assignments has been proposed, which enforces the milder condition $\mathbf{R} = \alpha\mathbf{I}$, α being a nonnegative variable to optimize.

In general, an optimal graph compression under our model does not necessarily conform to a clustering outcome. As an intuitive example of this fact, consider a complete bipartite graph $(\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \mathcal{A} \cup \mathcal{B}$ and $\mathcal{E} = \mathcal{A} \times \mathcal{B}$. In this case, a lossless compression can be obtained by taking $\mathsf{k} = 2$, $\mathbf{R} = \mathbf{E} - \mathbf{I}$ and $\mathbf{X}_{ki} = \mathbb{1}_{k=1}\mathbb{1}_{i\in\mathcal{A}} + \mathbb{1}_{k=2}\mathbb{1}_{i\in\mathcal{B}}$. If we try to interpret this as a clustering result, our two clusters are $\mathcal{A}$ and $\mathcal{B}$. These two sets, however, are both internally highly dissimilar and externally very similar, yielding a contradistinction with the standard definition of cluster. This suggests that Still, clustering results being conformal to the standard notion can be extracted, *e.g.* in case of block-structured nearly diagonal matrices, but the type of analysis that can be potentially conducted with our method has a spectrum that is broader than clustering.

4.7 Experiments

We conducted both synthetic and real-world experiments to assess the effectiveness of the proposed graph compression method. Additionally, we performed some experiments on clustering motivated by the discussion conducted in Section 4.6.

4.7.1 Synthetic experiments

For the first experiment, we created a data-set of weighted graphs that can be in principle compressed without loss of information. Each instance having order $\mathsf{n} = 400$ was generated by randomly sampling the matrix $\mathbf{X} \in \mathcal{S}_{01}$, the compressed graph $\mathbf{R} \in [0,1]^{\mathsf{k}\times\mathsf{k}}$ and by computing $\mathbf{G}$ as $\mathbf{X}^\top\mathbf{R}\mathbf{X}$. We considered compression rates varying between 2^{-7} and 2^{-2}. For each level of compression, we randomly generated 40 graph instances. We evaluated the solution found $(\tilde{\mathbf{X}}, \tilde{\mathbf{R}})$ by our algorithm in terms of the average entry-wise matrix reconstruction error, *i.e.* $\|\mathbf{G} - \tilde{\mathbf{X}}^\top\tilde{\mathbf{R}}\tilde{\mathbf{X}}\|_F/\mathsf{n}^2$. In Figure 4.2(a), we report for each level of compression (in $\log_2$-scale) the error, averaged over the 40 graph instances, and the corresponding standard deviation. The overall performance of the algorithm is good as we achieve a low entry-wise error in the reconstruction. Nevertheless, we do not always achieve the lossless compression due to the existence of local solutions, begin close to the optimal one. We also notice that the reconstruction error is lower at higher levels of compression.

In Figure 4.3, we report the evolution of the objective $f(\mathbf{Y}, \mathbf{R})$ of 6 executions of our algorithm. Each execution is performed on a random graph of order $\mathsf{n} = 400$ generated as described in the previous paragraph that can be compressed with the value of k adopted for the specific run. The different values of k reported in the legend correspond to compression rates ranging from 2^{-7} to 2^{-2}. We notice that the objective is monotonically decreasing as stated by our theorems and that the objective stabilizes in a number of steps that grows with the value of k.[2] All runs are characterized by an initial slow decay of the objective followed by a drastic decrease, which culminates in a stable limit. In our experience, depending on the initial random guess of $\mathbf{Y}$, this transition between slow and fast descent might eventually occur more than one time.

In a second experiment, reported in Figure 4.2(b), we generated graphs that

[2] this however should not be taken as a truth holding in general, for it depends also on the structure of the input graph.

are factorizable (as previously detailed) but at a fixed compression rate $\mathsf{k}/\mathsf{n} = 0.1$. We corrupted the factorizable graphs with different levels of Gaussian noise. By doing so, the existence of a lossless compression is not guaranteed anymore. We considered 40 graph instances per noise level and we reported the average entry-wise reconstruction error (with respect to the *noise-less* graph) and the standard deviation obtained with our algorithm when trying to compress the *noisy* graph. To have a clue of the incidence of noise, we report the reconstruction error of the ground-truth factorization with respect to the noisy graph $\tilde{\mathbf{G}}$, *i.e.* $\|\mathbf{X}^\top\mathbf{R}\mathbf{X} - \tilde{\mathbf{G}}\|_F/\mathsf{n}^2$. As we can see, at low noise levels our algorithm provides a compression yielding a good reconstruction, even though it is not the optimal one due to the existence of local solutions. The robustness of our algorithm becomes however clear at higher levels of noise, where our reconstruction error remains significantly below the error induced by the corruption, which means that we achieve a good signal-to-noise ratio.

In a third experiment, we deal with unweighted graphs and structural noise. We created a dataset of graphs, each having order 400 and being generated in the following way. We sampled the matrix $\mathbf{X} \in \mathcal{S}_{01}$ and we sampled a random matrix $\mathbf{R} \in [0,1]^{\mathsf{k}\times\mathsf{n}}$ with 50% of its components set to 0. We computed then $\mathbf{G} = \mathbf{X}^\top\mathbf{R}\mathbf{X}$ and considered each entry of the resulting matrix as the probability of observing an edge. Finally, we created the graph instance by sampling each edge according to the corresponding probability. By reiterating this procedure, we generated 40 graphs for each compression rate ranging between 2^{-7} and 2^{-2}. We try to pursue the ambitious goal of recovering the vertex partition induced by $\mathbf{X}$ by means of our graph compression algorithm. If $(\tilde{\mathbf{X}}, \tilde{\mathbf{R}})$ are the factorization parameter recovered by our algorithm, we evaluated the quality of the reconstruction by comparing the partitions induced by $\mathbf{X}$ and $\tilde{\mathbf{X}}$ in terms of classification accuracy. In Figure 4.2(c), we report the average accuracy with standard deviation that we obtained at varying levels of compression (in $\log_2$-scale). We achieved impressively high accuracies at all the compression levels that we considered. This success reveals that the proposed factorization allows to recover information about the random process adopted to generate the graphs and confirms the existence of strong links with the Szemerédi regularity lemma as mentioned in the introduction.

Finally, we performed a last synthetic experiment, which takes into account partial information. We generated 3 sets of 10 graph, each graph having 500 nodes. The first set consists of Erdös-Rényi-type *random* graphs with density 0.01, which lack any structure. The second and third set are synthetic *unstructured* and *structured* community graphs generated according to the following procedure reflecting

the generative model described in [54]. Each node is assigned to a cluster according to a Chinese restaurant process, where the probability of being assigned to an existing cluster is proportional to the size of the cluster, with a non-zero probability proportional to α of being assigned to a new cluster. In our experiments we set $\alpha = 5$ for either type of community graphs. Once the clusters have been determined, undirected edges between vertices are sampled independently according to a probability that depends on their cluster assignments. Within-custer link probability $\eta_{\ell\ell}$ of cluster ℓ is sampled from a Beta distribution with parameters $(a_{\text{in}}, b_{\text{in}})$, while the between-cluster link probability between clusters ℓ and m is sampled according to a Beta distribution with parameters $(a_{\text{out}}, b_{\text{out}})$. In the case of structured community graphs, the Beta distribution for the between-cluster links is constrained to the interval $[0, x_{\ell m}]$, where $x_{\ell m} = \min\{\gamma_\ell \eta_{\ell\ell}, \gamma_m \eta_{mm}\}$ and γ_ℓ denotes the cluster gap probability of cluster ℓ being sampled from a Beta distribution with parameters $(a_{\text{gap}}, b_{\text{gap}})$. The parametrization used to generate the unstructured community graphs is $a_{\text{out}} = b_{\text{out}} = a_{\text{in}} = 1$ and $b_{\text{in}} = 10$, while the one used to generate the structured community graphs is $a_{\text{out}} = a_{\text{in}} = a_{\text{gap}} = 1$ and $a_{\text{out}} = b_{\text{out}} = b_{\text{gap}} = 10$. Intuitively, the structured community graphs are created in a way to fulfill the property that communities have more internal than external links, as opposed to unstructured community graphs. We compare our algorithm against other approaches, which are specifically tuned for this graph generation model as opposed to our approach: Infinite Relational Model (IRM), Infinite Diagonal Model (IDM) and Bayesian Community Detection (BCD). We used the implementation of these approaches provided by the authors of BCD [54]. In Table 4.1 we report the ability of each approach to predict the missing edges in terms of average Area-Under-Curve (AUC). The results show that with a random setting the prediction of each algorithm is worthless, yielding AUC$\approx$0.5. Indeed, those graphs lack a regularity, which renders the link-state discrimination difficult to achieve. As the information becomes more structured, our approach yields competitive results (AUC$>$80%). In fact, both structured and unstructured community graphs present a marked block-wise arrangement. These graphs can be effectively compressed and good levels of state-link discrimination can thus be achieved. The structured community graphs are slightly more regular than the unstructured ones, due to the constrained number of between-cluster versus within-cluster links, which motivates the improved AUC value (+3%).

DataSet	Methods			
	IRM	IDM	BCD	Graph Compression
Random	0.50 ± 0.01	0.50 ± 0.02	0.50 ± 0.02	0.50 ± 0.02
Unstructured	0.83 ± 0.01	0.45 ± 0.01	0.78 ± 0.01	0.81 ± 0.01
Structured	0.77 ± 0.03	0.67 ± 0.01	0.83 ± 0.01	0.84 ± 0.01

Table 4.1: Experimental results on Erdós-Renyi random graphs for the task of predicting the state of non-observed edges. We consider three different scenarios: purely random graph, unstructured graph and structured graph. We report the average AUC value (and standard deviation) obtained with 10 graph samples per setting.

4.7.2 Real-world experiments

In real-world scenarios, the graphs to be compressed do not exhibit necessarily a lossless compression at a given compression rate. Therefore, as opposed to the synthetic case, we try to determine the quality of the factorization in a more pragmatic way. We analyses the performances of some algorithms working on graphs and kernels, when applied to the original matrix $\mathbf{G}$ and to the reconstructed matrix $\tilde{\mathbf{X}}^\top \tilde{\mathbf{R}} \tilde{\mathbf{X}}$ at different compression rates, $(\tilde{\mathbf{X}}, \tilde{\mathbf{R}})$ being the solution found by our algorithm. The rationale of this experiment is that a good compression, having a low loss, will not impact the performance of an algorithm significantly, if we run it on the reconstructed graph/kernel rather than the original one.

We performed a first set of experiments on *clustering* with a spectral algorithm (SC) working on graphs [76] and with the Kernel K-Means algorithm (KKM) working with a kernel [18]. Table 4.2 reports the classification accuracies obtained by the two algorithms on different machine learning data-sets, namely Iris, Wine, Haberman, Pima, BloodT and Imox. Each algorithm was run on a graph/kernel reconstructed after having been compressed with a rate varying between 2^{-5} and 1 (in the tables we reported the rate in $\log_2$-scale). The results obtained at rate 1 represent the performance of the algorithms on the original data, without compression. As we can see, the accuracies reported are not significantly affected due to the compression, even at low rates. In some cases, we notice an increase of the performance at some intermediate compression levels (see, *e.g.* Imox for KKM and Haberman for SC).

We performed also some qualitative experiments on *dimensionality reduction* with Kernel- Principal Component Analysis (KPCA) [75]. We report in Figure 4.4

(a) Kernel K-Means

Dataset	n	k	Compression rate					
			1/32	1/16	1/8	1/4	1/2	1
Iris	150	3	0.85	0.85	0.89	0.90	0.91	0.91
Wine	178	3	0.56	0.72	0.72	0.72	0.71	0.73
Haberman	306	2	0.65	0.65	0.65	0.65	0.65	0.65
Pima	768	2	0.78	0.89	0.81	0.89	0.89	0.90
BloodT	748	2	0.41	0.40	0.38	0.39	0.40	0.39
Imox	192	4	0.75	0.88	0.86	0.87	0.77	0.78

(b) Spectral Clustering (Normalized Cut)

Dataset	n	k	Compression rate					
			1/32	1/16	1/8	1/4	1/2	1
Iris	150	3	0.85	0.85	0.89	0.90	0.89	0.89
Wine	178	3	0.71	0.57	0.71	0.59	0.57	0.58
Haberman	306	2	0.50	0.73	0.73	0.71	0.71	0.64
Pima	768	2	0.78	0.89	0.87	0.89	0.89	0.91
BloodT	748	2	0.41	0.42	0.40	0.40	0.40	0.41
Imox	192	4	0.75	0.78	0.78	0.77	0.79	0.79

Table 4.2: Clustering results obtained by Kernel K-means and Spectral Clustering (Normalized Cut) on different data-sets, after having compressed the data (graph/kernel) at different rates.

the data points of two machine learning data-sets, namely Iris and eColi, reprojected on the 3 principal components found with K-PCA and colored according to the ground-truth classes. For each data-set, we performed the analysis using both the original kernel (compression rate 1) and the same kernel reconstructed after having been compressed with rate 0.0625. As we can see, the degree of separation of the classes between the analysis conducted on the original kernel and on the reconstructed one are comparably good.

Finally, we perform a set of experiments to test the effectiveness of our approach under partial information on several real-world data-sets: Friend, Ecoli, Yeast, MIPS, Friend Same Sex, Friend Same Race, Caltech, Macaque and Lesmis. For each dataset we consider 10 random splits consisting of 80% of the input matrix $\mathbf{G}$ for training and 20% for testing. The training entries of the input matrix are used to compute the factorization, while the test entries are used to assess the quality of the predictions obtained using the computed factorization. We compare against Non-Negative Matrix Factorization (NMF) [48], which is a well-known ma-

trix factorization approach that factorizes an interaction matrix to low-dimensional representations with non-negativity constraints and has also been employed for community detection (see, *e.g.* [69]), and against Graph-regularized NMF (GNMF) [11], which is a recent, regularized variant of NMF. In Table 4.3 we report the results obtained in terms of AUC under different values of k.[3] When we have high compression rates ($\mathsf{k} = 3$) NMF and GNMF yield better results on most of the data-sets. This is not surprising because the number of parameters to optimize with $\mathsf{k} = 3$ is larger in NMF/GNMF than our approach and can thus better capture the matrix structure than we do. However, this over-parametrization has a negative effect as we increase k from 3 to 5 and 7, for our approach achieves the best results on 5/9 and 6/9 data-sets, respectively. Also this experiment shows that our model can better compress the information encoded in the structural data without loosing the most significant signals. Finally, we think that it is due remarking that, as opposed to NMF and our approach, GNMF comprises a free parameter than has to be tuned, which balances the matrix approximation term and the regularization term. We have manually tuned it to deliver better results, but wrong choices of this parameter might lead to bad results. Nevertheless, our approach turned out to be competitive despite this handicap.

[3]for NMF and GNMF this coincides with the number of latent dimensions

DataSet	n	NMF			GNMF			Graph Compression		
		3	5	7	3	5	7	3	5	7
Friend	90	0.78 (0.00)	0.72 (0.02)	0.75 (0.01)	**0.84** (0.01)	0.79 (0.00)	**0.79** (0.00)	0.78 (0.01)	**0.80** (0.01)	**0.79** (0.01)
Ecoli	230	**0.82** (0.01)	0.82 (0.01)	0.82 (0.00)	0.80 (0.00)	0.82 (0.00)	0.82 (0.01)	0.75 (0.00)	**0.84** (0.02)	**0.88** (0.01)
Yeast	283	0.84 (0.01)	0.88 (0.01)	0.84 (0.00)	**0.86** (0.01)	**0.89** (0.01)	**0.89** (0.00)	0.74 (0.00)	0.80 (0.00)	0.83 (0.00)
MIPS	871	0.81 (0.01)	0.85 (0.00)	0.87 (0.00)	**0.91** (0.00)	0.93 (0.00)	0.94 (0.00)	**0.91** (0.00)	**0.94** (0.00)	**0.97** (0.00)
Fr S. S.	90	0.97 (0.00)	0.98 (0.00)	0.99 (0.00)	**1.00** (0.00)	**1.00** (0.00)	**1.00** (0.00)	0.99 (0.00)	0.99 (0.00)	**1.00** (0.00)
Fr S. R.	90	0.92 (0.00)	0.96 (0.00)	0.99 (0.00)	0.96 (0.00)	0.96 (0.00)	0.96 (0.00)	**0.99** (0.00)	**0.99** (0.00)	**1.00** (0.00)
Calthec	762	0.87 (0.00)	**0.90** (0.00)	**0.91** (0.00)	**0.88** (0.00)	**0.90** (0.00)	**0.91** (0.00)	0.78 (0.00)	0.80 (0.00)	0.81 (0.00)
Macaque	47	0.86 (0.01)	0.81 (0.02)	0.81 (0.01)	**0.89** (0.01)	**0.87** (0.01)	**0.85** (0.01)	0.77 (0.02)	0.81 (0.01)	0.83 (0.00)
Lesmis	77	**0.88** (0.00)	0.84 (0.01)	0.83 (0.01)	0.86 (0.01)	0.86 (0.01)	0.81 (0.02)	0.86 (0.00)	**0.90** (0.01)	**0.89** (0.02)

Table 4.3: Experimental results on real-world data-sets for the task of predicting the state of non-observed edges. We report the average AUC (and standard deviation) obtained with 10 random training/test set (80%/20%) splits.

4.7.3 Clustering

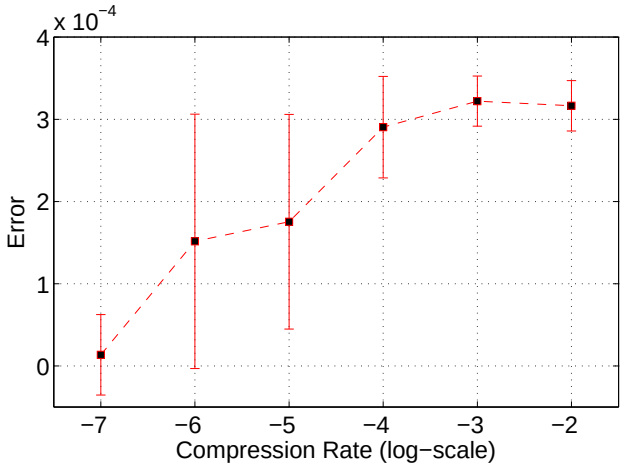
Motivated by the relation that exists between graph compression and clustering, as pointed out in Section 4.6, we tried to interpret the outcome of our graph compression algorithm as it were a clustering result. We performed experiments on synthetic and real-world machine learning data-sets, namely the S100 Block Stochastic (S-Block) dataset [86], the Iris dataset (Iris), the NIST handwritten digits dataset (Digit), a subset of the SCOP protein dataset (Scop) [39] and the Wine dataset. We performed a graph compression with k tuned to the optimal number of clusters for each dataset and we considered the factorization parameter $\tilde{\mathbf{X}}$ found by our graph compression algorithm as the clustering result. To escape local minima we run our algorithm 10 times and kept the solution exhibiting lowest reconstruction error. We report as evaluation measure the classification accuracy for each dataset of our solution against the ground truth solution. Due to the relation shown in Section 4.6

between our graph compression method and the Baum-Eagon (BE) clustering algorithm [72], we report for the sake of comparison the scores from the original paper for the data-sets where BE has been applied. As we can see from Figure 4.5, in all data-sets, excepting Digit the compression delivered by our method correlates well with the data-set's ground truth. Compared to BE, we perform slightly worse on Iris, S-Block and perform very bad on Digit. This is however an expected scenario. Indeed, according to the discussion done in Section 4.6, a good graph compression might not conform to a standard clustering result. The results obtained on Digit support this fact as we verified that the factorization found on digit, despite being very far from the ground-truth clustering, delivered a low reconstruction error, thus providing a good compression. The comparison with BE also reveals that by adding a constraint in our model enforcing the compression to fulfill the properties of a clustering, the results can clearly improve, one exception being Scope. We conclude remarking that the goal of this section is not to show that our method can be used as a clustering algorithm, but we considered interesting to emphasis the relation between clustering and graph compression also from an experimental perspective.

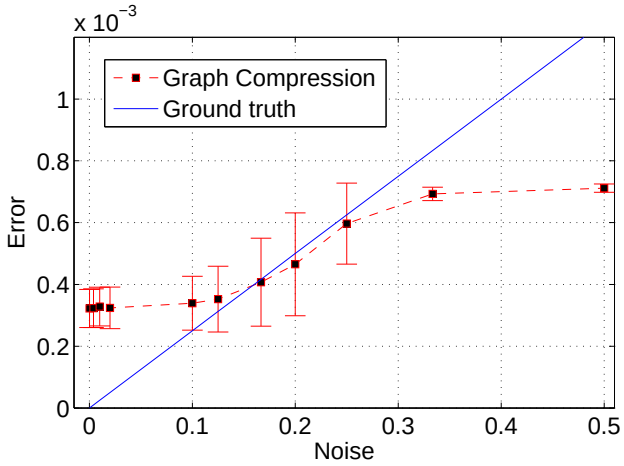
4.8 Discussion and future work

In this chapter, we have proposed a novel matrix factorization, which can be used to compress graphs and kernels, also in the presence of missing observations about the edges of the graph to be compressed. We have introduced an algorithm for finding such a factorization in an approximate way. Interestingly, we came up with an update rule which resembles the ones adopted for NMF and the one introduced by Baum and Eagon for maximizing polynomials with non-negative coefficients in probability domain. We have proven convergence properties of these update rules and we think that similar optimization dynamics can be employed for other types of matrix factorization in probability domain.

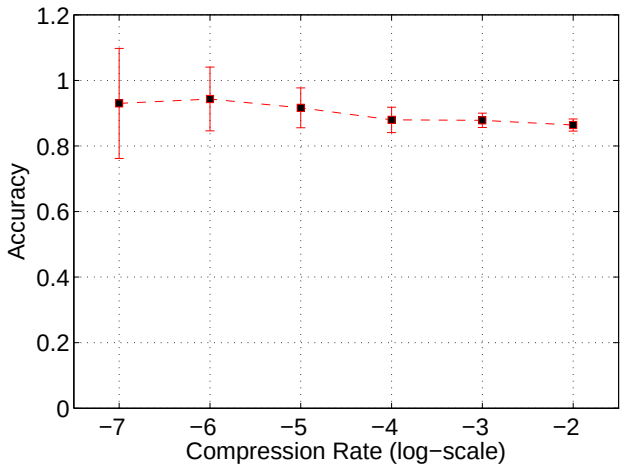
Besides data analysis, our graph compression can be potentially employed to speed-up existing graph-based and kernel-based algorithms. We did not address this topic in this chapter since our focus was to show the effectiveness of our graph compression formulation. However, the experiments conducted in Section 4.7.2 evidenced that the information preserved by the compression is sufficient to avoid drastic decay of the algorithm's performances. Actually, we experienced a constant performance even at low compression rates. The complexity of many algorithms can be reduced by replacing $\mathbf{G}$ with the factorization $\mathbf{X}^\top\mathbf{R}\mathbf{X}$ and by exploiting the sparsity of $\mathbf{X}$. As an example, the complexity of a matrix-vector multiplication can be reduced from n^2 to $(\mathsf{k}^2 + \mathsf{n})$.



(a)



(b)



(c)

Figure 4.2: Results obtained for different types of synthetic experiments. See Section 4.7.1 for details.

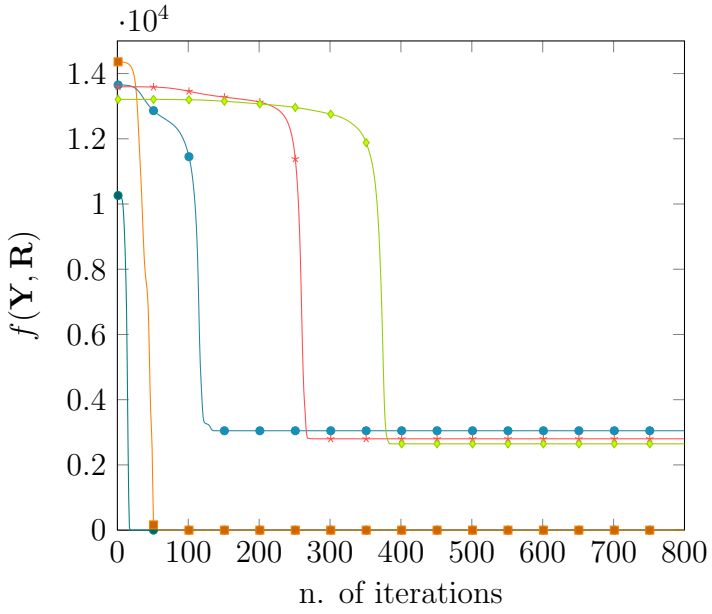
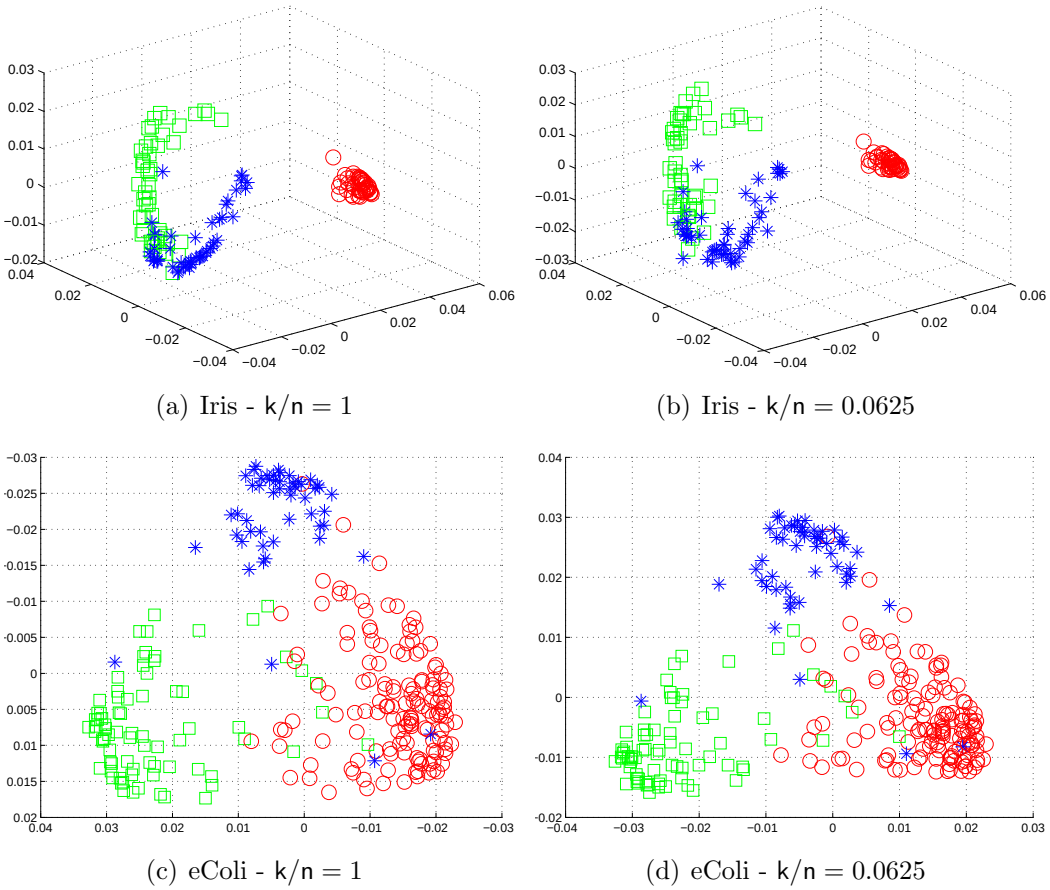


Figure 4.3: Evolution of the objective $f(\mathbf{Y}, \mathbf{R})$ during the execution of our graph compression algorithm. We report the evolution of 6 runs with different values of k on random graphs of order $\mathsf{n} = 400$ that can be potentially compressed to order k (see, Section 4.7.1 for details about the graph generation procedure). Markers have been sparsified for a better visualization.



(a) Iris - k/n = 1 (b) Iris - k/n = 0.0625

(c) eColi - k/n = 1 (d) eColi - k/n = 0.0625

Figure 4.4: Qualitative results of K-PCA for Iris and eColi dataset. See Section 4.7.2 for details.

Dataset	n	k
Iris	150	3
Scope	451	5
Digit	1000	10
S-Block	100	5

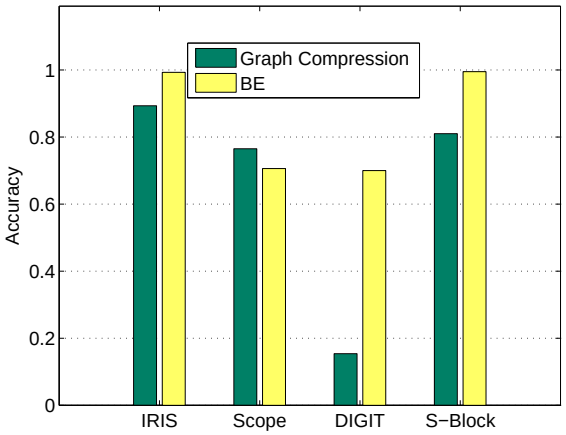


Figure 4.5: Clustering results obtained on different machine learning data-sets by our approach and BE [72]. For details see Section 4.7.3.

V

Appendix

A

Determining the Number of Dominant-Set Clusters and Community Detection

Desire: The Cause of All Suffering

Buddha

Online social networks facilitate connections between people sharing the same interests, values, or membership in particular groups. When we examine the structure of such networks, it is natural to consider groups as partially overlapping because people can show interest for multiple topics at the same time. For instance, most of the users have connections with several groups that embody different aspects of people's social life such friendship, family and colleague. Since a lot of information can be extracted by analyzing the topology of such connections, community extraction from social networks graphs represents a growing interest in computer science.

In the last decade, numerous classic graph clustering methods have been adapted for community detection which are namely: random walks, spectral clustering, modularity maximization, differential equations and statistical mechanics. However, most of these are limited in practice since they consider communities as dense disjoint regions. Therefore, multiple-community node clustering represents one of the most interesting research direction nowadays. Moreover, overlapping-community detection is one of the recent field in this domain.

Additional to the above motivation for the community detection, a community can be seen as a supper-node in a graph so by detecting communities in the graph, we can make a compression which is can be categorized as structural compression that is explained in introduction section. It has been reported that one of drawbacks

of the community detection in the literature is the number of community k should be specified in advance. Therefore we have proposed a method to detect the minimum number of communities or cluster in advance before detecting communities.

A.0.1 Dominant set based determination of number of clusters

Predicting Cluster Membership for Out-of-Sample

One of the main drawback of dominantset is handling a big data-sets such as document classification and database visualization. To resolve this problem pavan and pelillo proposed a new solution on their paper called efficient Out-of-Sample Extension of Dominant-set Clusters [63] which we are applying its main idea as part of our method.

In their paper, they have proposed a mechanism based on taking part of the data and performing clustering. Then for each new unseen instances calculates the similarity measure just between new instance and previous detected clusters. After that, they proposed a measure to decide to assign this new instances to detected clusters or insert a new cluster based on the sign of the highest similarity weight.

In another way as above, the sign of $W_{s\cup\{i\}}(i)$ provides an indication as to whether i is tightly or loosely coupled with the vertices in S (the condition $W_{s\cup\{i\}}(i) = 0$ corresponds to a non-generic boundary situation that does not arise in practice and will therefore be ignored). Accordingly, it is natural to propose the following rule for predicting cluster membership of unseen data:

if $W_{s\cup\{i\}}(i) > 0$, then assign vertex i to cluster S .

According to this rule, the same point can be assigned to more than one class, thereby yielding a soft partition of the input data. To get a hard partition one can use the cluster membership approximation measures. Moreover for some instance i that no cluster S satisfies the rule, in which case the point gets unclassified (or assigned to an "outlier" group). This should be interpreted as an indication that either the point is too noisy or that the cluster formation process was inaccurate. In our experience, however, this situation arises rarely [63].

Related work On Detecting Number of Clusters

A fundamental and the major problem in cluster analysis is how many clusters are appropriate for the description of a given system, which is a basic input for many clustering algorithm. A variety of methods have been proposed to estimate the number of clusters by different scholars on different time. Gordon [30] groups the approaches for determining the number of clusters into global and local methods. The former evaluate some measure over the entire data-set and optimize it as a function of the number of clusters. The later consider individual pairs of clusters and test whether they should be merged.

A drawback of most global methods is that there is no guidance for whether the data should be partitioned (best number of cluster is greater than 1) or not. However, it will not be a problem if users have good reasons to believe that there are clusters present in data. The local methods are intended to test the hypothesis that a pair of clusters should be merged or not. They are suitable for assessing only hierarchically-nested partitions. According to Gordon comments, the significance levels should not be interpreted strictly since multiple tests are involved in the procedure.

Global methods

Calinski and Harabasz's [12] in their work proposed a method for determining number of clusters based on an index called calinski harasbansz's (CH(g)) which is defined as follows:

$$CH\,(g)=\frac{B(g)\big/(g-1)}{W(g)\big/(n-g)}$$

Where B(g) and W(g) are the between- and within-cluster sum of squared errors,with g clusters. B(g) and W(g) defined mathematically as follows:
Suppose we have multivariate data containing n objects in p dimensions. Each object can be expressed as x_i=$(x_{i1}, x_{i2}, ..., x_{ip})^{'}$, i = 1,..., n. We define the dispersion matrix of each group by

$$W_m = \sum_{l=1}^{n_m} (x_{ml} - \bar{x_m})\,(x_{ml} - \bar{x_m})', m = 1, ..., g.$$

Then the pooled within-group dispersion matrix W is defined by

$$W = \sum_{m=1}^{g} \sum_{l=1}^{n_m} (x_{ml} - \bar{x_m}) (x_{ml} - \bar{x_m})'$$

The between-group dispersion matrix is defined by

$$B = \sum_{m=1}^{g} n_m (\bar{x}_m - \bar{x}) (\bar{x}_m - \bar{x})', \bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

Calinski and Harabasz's stated that, the value which maximizes CH(g) over g is the optimum number of the clusters.
According to the comparative study conducted by Milligan and Cooper [53], on 30 methods of determining the number of clusters in data, this method generally outperformed the others.

Hartigan's method

Hartigan [34] proposed the method which is based on the following index

$$Har(g) = \left[\frac{W(g)}{W(g+1)} - 1 \right] / (n - g = 1)$$

In their work, [34] proposed to calculate the value of $Har(g)$ starting from $g = 1$ and adding the cluster if the value of $Har(g)$ is significantly large. A simpler decision rule suggested by Hartigan is to add a cluster if $Har(g)$ is greater than 10. Hence, the cluster number is best estimated as the smallest g, $g = 1, 2, ...$, such that $H(g) \leq 10$. for more detail it is advisable to refer [34].

Silhouette statistic

In order to estimate the optimum number of clusters of a data-set Kaufman and Rousseeuw [73] proposed the silhouette index. The definition of the silhouette index is based on the silhouettes introduced by Rousseeuw [73], which are constructed to visualized graphically how well each object is classified in a given clustering output. To plot the silhouette of the mth cluster, for each object in C_m, calculate s(i) as

$$S(i) = \frac{a(i) - b(i)}{max\{a(i), b(i)\}}$$

where:

a(i)=average dissimilarity of object i to all other objects in the m^th cluster.

$$\text{b(i)} = \min_{C \neq C_m} d(i, C)$$

d(i,C)=average dissimilarity of object i to all other objects in cluster C; C $\neq C_m$

calculate the average of s(i)($\bar{S}$(g)) for all objects in the data which is also called the average silhouette width for the entire data set. This value reflects the within-cluster compactness and between-cluster separation of a clustering.
Compute $\bar{S}$(g) for $g = 1, 2, ...$(for all number of clusters which is assumed to be optimum) and select the one which maximizes $\bar{S}$(g). According to [73] the value of g which maximizes the average silhouette index($\bar{S}$(g)) is the optimum number of cluster of a data-set. [88]

$$\text{Optimum number of cluster } (\hat{G}) = \arg\max_g \bar{S}(g)$$

Gap method

Tibshirani et al. [83] proposed an approach for estimating the number of clusters(k) in a data set via the gap statistic. The main idea of the gap method is to compare the within-cluster dispersion in the observed data to the expected within-cluster dispersion assuming that the data came from an appropriate null reference distribution. The best value of k is estimated as the value $\hat{k}$ such that log(W($\hat{k}$)) falls the farthest below its expected curve. gap is defined as follows:

$$\text{Gap}_k\,(g) = E_n^* log(W(k)) - log(W(k))$$

where $E_n^* log(W(k))$ indicates the expected value of log(W(g)) under the null distribution. The value of k which maximizes $Gap_n(g)$ is the optimum number of clusters, $\hat{k}$. For detail explanation look [83]

Local methods

In this section we will see the two local methods used for estimating the number of cluster, which are among the top 5 performers algorithms according to the comparative study of milligan and cooper's [53]. The first one is proposed by Duda and Hart [23]. In their method, the null hypothesis that the mth cluster is homogeneous is tested against the alternative that it should be subdivided into two clusters. The test is based on comparing the within-cluster sum of squared errors of the mth cluster, $J_1^2(m)$, with the within-cluster sum of squared distances when the mth cluster is optimally divided into two, $J_2^2(m)$. If the mth cluster contains n_m objects in p dimensions, then the null hypothesis should be rejected if

$$J_1^2(m)\big/J_2^2(m) < 1 - 2\big/(\pi p) - z\big[2(1 - 8\big/(\pi^2 p))\big/(n_m p)\big]^{\frac{1}{2}}$$

where z is the cutoff value from a standard normal distribution specifying the significance level.

The second method proposed by Beale [6] tests the same hypothesis with a pseudo-F statistic, given by

$$F \equiv \left(\frac{J_1^2(m)-J_2^2(m)}{J_2^2(m)}\right) \Big/ \left(\left(\frac{n_m-1}{n_m-2}\right) 2^{\frac{2}{p}} - 1\right)$$

The homogeneous one cluster hypothesis is rejected if the value of the F statistic is greater than the critical value from an $F_p, (n_m-1)p$ distribution. In both tests, given the rejection of the null hypothesis, it follows that the subdivision of the mth cluster into two sub clusters is significantly better than treating it as a single homogeneous cluster [88].

A.1 Proposed Method

We have proposed a two steps method to detect the number of cluster, in the first step, we apply modified Dominant Set on Dissimilarity matrix to detect minimum number of clusters or stability number and ,in the second step we look for a saturated cluster. In the binary case, saturated clique is a maximal clique that all its node belongs to more than one clique. So out of sample strategy applied to handle the existence of undetected cluster.

The steps are explained in detail as follows:

A.1.1 Detect Number of Cliques

Step 1

Let $G = (V; E)$ be an undirected graph without self-loops, where $V = 1, 2, ..., n$ is the set of vertices and $E \subseteq V \times V$ the set of edges. We define the order of a graph G as the cardinality of V . Two vertices $u, v \in V$ are adjacent if $(u, v) \in E$. A subset C of vertices in G is called a clique if all its vertices are mutually adjacent. It is a maximal clique if it is not a subset of other cliques in G. It is a maximum clique if it has maximum cardinality. The cardinality of a maximum clique of G is also called clique number and denoted by $w(G)$, Moreover stability number is defined $\alpha(G) = \frac{1}{w(G)}$. It should be mentioned that number of maximal clique and stability

number are not equal always. More precisely, stability number is lower or equal to the number of maximal clique.

The adjacency matrix of G is the $n \times n$ symmetric matrix $A_G = (a_{ij})$, where $a_{ij} = 1$ if $(i, j) \in E$, $a_{ij} = 0$ and $\bar{A}$ is defined as dissimilarity matrix otherwise.

The adjacency matrix of an undirected graph can be regarded as the similarity matrix of a clustering problem and compliment of a graph G (dissimilarity) is defined $\bar{A} = 1 - A_G$ for unweighted case therefore our framework can be used to find the stability number.

Consider the following constrained quadratic program derived from weighted Motzkin-Strass formulation.

$$\begin{aligned} \tfrac{1}{w(G)} = \max \quad & x^T(\bar{A} + \alpha I)x \\ \text{s.t.} \quad & \mathbf{X} \in \triangle \subset \mathbb{R}^{\mathbf{n}}\,. \end{aligned} \tag{A.1}$$

With $$\triangle = (x \geq 0 \quad and \quad e^T = 1)$$

where n is the order of G, I the identity matrix, α is a real parameter and where $\triangle$ is the standard simplex of the n-dimensional Euclidean space.

In 1965, Motzkin and Straus [56] established a connection between the maximum clique problem with $\alpha = 0$. Specifically, they related the clique number of G to global solutions x^* of the program through the formula $w(G) = (1 - f_0(x^*))^{-1}$, and showed that a subset of vertices C is a maximum clique of G if and only if its characteristic vector $x^C \in \triangle$ is a global maximizer of f_0 on $\triangle$. Pelillo and Jagota [68], extended the Motzkin-Straus theorem by providing a characterization of maximal cliques in terms of local maximizers of f_0 in $\triangle$.

A drawback of the original Motzkin-Straus formulation is the existence of spurious solutions, *i.e.* , maximizers of f_0 over $\triangle$ that are not in the form of characteristic vectors. This was observed empirically by Pardalos and Phillips [61] and formalized later by Pelillo and Jagota [68]. In principle, spurious solutions represent a problem since, while providing information about the order of the maximum clique, do not allow us to easily extract its vertices. Fortunately, there is a straightforward solution to this problem which has been introduced by Bomze [10]. He, indeed, suggested to add a constant α on the diagonal of the adjacency matrix of the graph and basically proved that for $0 < \alpha < 1$ all local maximizer of A.1 are strict and in one-to-one correspondence with the characteristic vectors of the maximal cliques of G. In our case as reported in the paper of Pelillo and Jagota, α sets to a value close to 1.

In the weighted case, compliment Graph or weighted dissimilarity matrix is calculated by

$$\bar{A_{ij}} = exp(-\frac{||F(i) - F(j)||^2}{\sigma^2}) \tag{A.2}$$

Equation A.1 is designed for unweighted matrices by motzkin-strauss. We have extended their work to the weighted version. We have observed that in this way the minimum number of maximal cliques can be obtained.

Step 2

As mentioned before the first step detects the minimum number of clusters based on stability definition. To detect the existence of extra clusters which called as saturated clusters, Efficient Out-of-Sample Extension of Dominant-Set Clusters is applied to indirectly if there exist new (unseen) cluster representative. As explained before sign of similarity weight $W_S(i)$ uses as an indicator to decide to insert a new cluster or not.

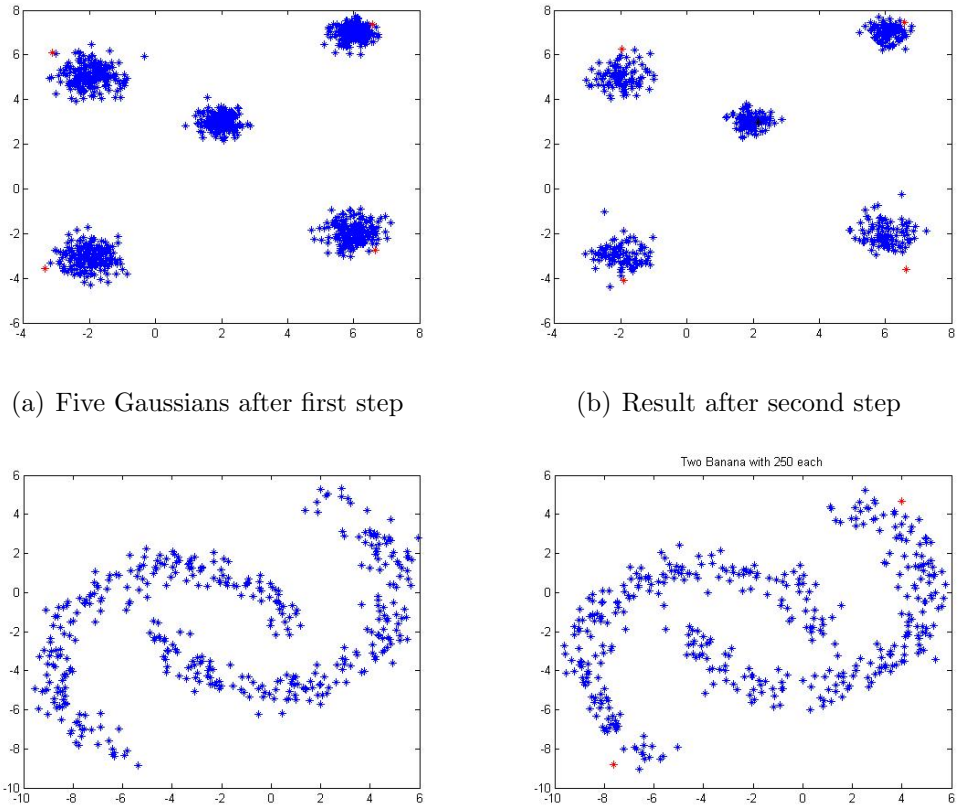
A.1.2 Experimental Results:

We conducted both synthetic and real-world experiments to assess the effectiveness of the proposed number of cluster detection method.

Synthetic Experiment:

For the first experiment, we created a data-set of weighted graphs with five different Gaussians which dont have overlapping with each other that can be in principle detected without difficulty. In the second experiment, we have made a performance on elongated data structure like two banana structure. We have used these toy data-sets to show different steps in our method visually. We have tried this experiment several times with different mean and sigma for each Gaussian and we are able to detect a reasonable result.

In the last synthetic experiment, we have explored the effect of noisy data in our proposed method. So we have chosen three separated Gaussians and we made them close to each other by changing their mean value respect to each other in each steps to evaluate the performance of the method respect to the noise. Moreover, we have changed the value of their sigma when they got very close. Figure A.3 shows two steps of explained procedure.



(a) Five Gaussians after first step

(b) Result after second step

(c) Two bananas after first step

(d) Result after second step

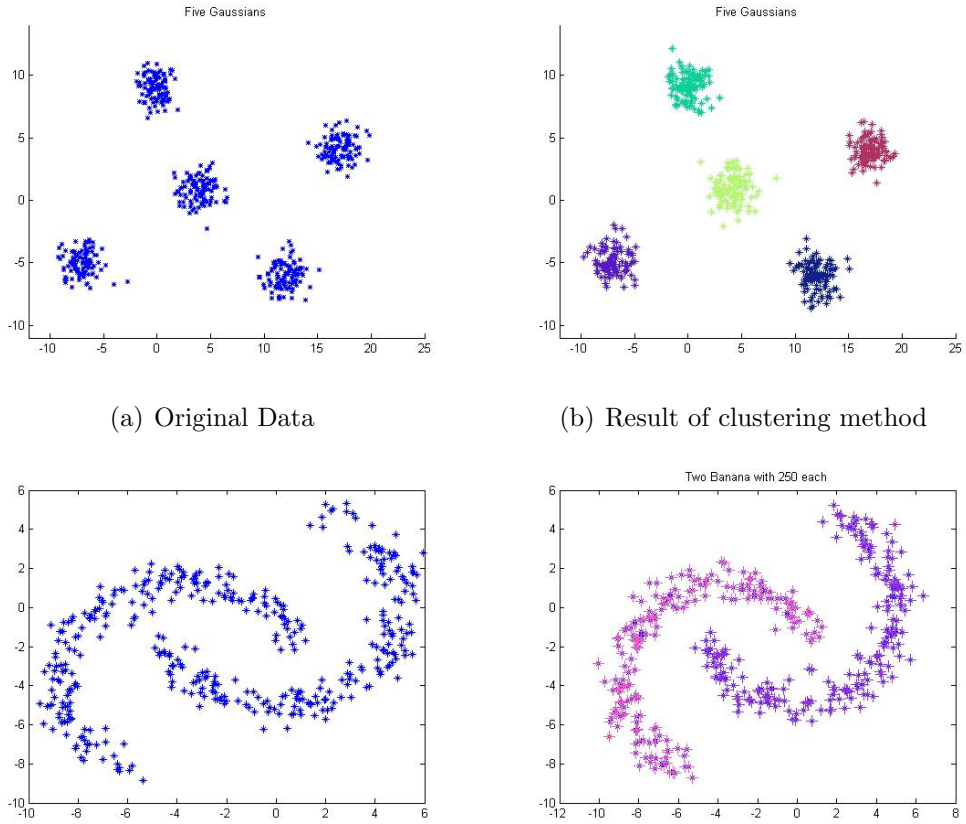
Figure A.1: Two steps of proposed method from left to right column.

Figure A.1 shows the different steps of the method on Gaussian and elongated data. As a result, it is clear that the first step of the method is able to detect outlier clusters and the overlapped one detected in the second step.

Figure A.2 gives the result of final step which is after applying clustering method based on detected number of clusters. For this experiment, we have used graph transduction for the clustering method. Figure A.3 shows the result of the method on noisy data. Three Gaussians are drawn to be close to each other with different σ.

Real World Experiment:

In real-world scenarios, we try to determine the quality of the number of cluster detection in a more pragmatic way. We analyze the performances of some algorithms



(a) Original Data

(b) Result of clustering method

(c) Original Data

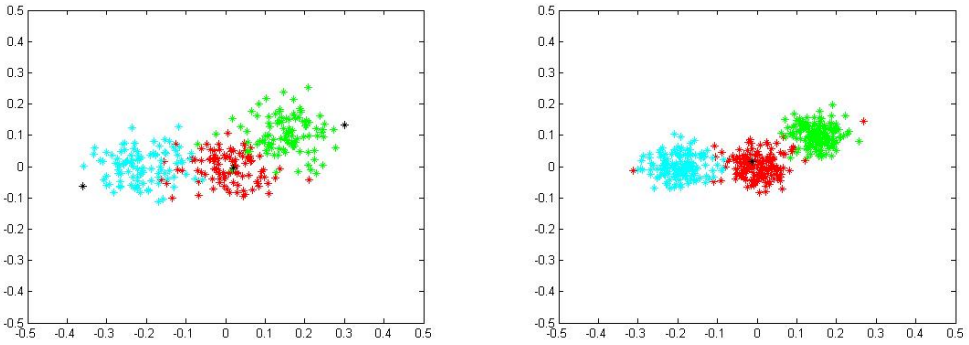
(d) Result of clustering method

Figure A.2: Graph transduction method is applied to show the final result

working on real world data-set like UCI and Social network data-sets. The rationale of this experiment is that impact the performance of an algorithm.

We performed a first set of experiments with a spectral algorithm (SC) and with the K-Means algorithm (KM) We have used graph transduction as a clustering method.

Table A.1 shows the result of proposed method on different data-sets and we have used graph transduction as a clustering method that is compared with SC and KM. The experiment in clustering step runs several times and average value is reported. It showed be mentioned that graph transduction might be sensitive to the initialization starting point so the final accuracy might be lower than SC in some cases.



(a) Three close Gaussians data with high σ value

(b) Three close dense Gaussians data with low σ value

Figure A.3: Shows the performance of proposed method on noisy data

A.2 Discussion and future work

We were able to propose an efficient method to detect the number of clusters automatically using the concept of Dominant set approach. We have proved the effectiveness of proposed method in qualitative and quantitative ways by making tests on different computer generated data sets, UCI repository data sets and social network data sets and we got an interesting and promising result.

For the future work, we have observed that, there is a growing interest in studying new algorithms to extract communities allowing a certain degree of overlap between nodes which can be combined with our work to detect communities without specifying number of clusters in advance.

		Data sets						
				number of classes		accuracy		
		Name	instances	original	detected	our method	K-means	Ncut
data source	computer generated(Toy)	FG	500	5	5	99.8	77	100
		Banana	500	2	2	94	75	75
		TGC	450	3	3	98	98	67
		SG	700	7	7	99.6	73	100
	UCI	wine	178	3	3	69	60	71
		Iris	150	3	3	86	86	90
		Ionosphere	351	2	2	70	71	68
		pima	768	2	2	67	66	61
		Ecoli	272	2	2	94	78	76
		Soybean	136	4	4	78	68	82
		liver	345	2	2	58	55	53
		haberman	306	2	2	75	52	51
		auto_mpeg	398	3	3	74	64	70
	Social network	karate	34	2	2	88		
		dolphins	62	2	3			
		food	45	7	6			
		collaboration	235		8			
		jazz muscian	198		7			

Table A.1: The result of proposed method on different data-sets with compression to SC and KM methods as clustering

Conclusions

In this thesis we presented some of the results that was obtained during my three years as a Ph.D. student.

C.1 Contributions

The aim of this thesis was to investigate different techniques to compress a graph. Graph compression is one of the fundamental methods to these days research as far as the amount of data is increasing rapidly. Graph compression can be utilized in many environments such as saving the memory space by compressing data. Therefore, the compression can be lossy or lossless depend on the technique.

Extremal Graph theory is a new branch in graph theory which is developed during early 20th century. One of the most practical theorem in Extremal Graph Theory is Szemerédi's Regularity Lemma which is used in celebrated proof of the Erdös-Turán Conjecture on arithmetic progressions in dense sets of integers. We have explored its power in the fields of Pairwise Clustering.

Roughly speaking, the Lemma guarantees the existence, for a sufficiently large graph, vertices of the graph can be partitioned to the small sub-graphs with equal size that satisfies particularly strict constraints on the edge density between each pair of subsets.

We have implemented a two steps strategy method based on Alon et al. [3] work who presented a first algorithmic method of the Szemerédis Regularity Lemma to make partition of vertices of an input graph in section III. This partitioning shows some interesting properties that can be useful for compression in a clustering context. This nature of partitioning can be applied to every graph to partition them. We have been able to use this method as pre-step clustering method and we have tested on different benchmark in UCI dataset. Moreover in practice, we had been able to justify that reduce graph is able to detect a significant sub-structure in orig-

inal graph as well as theory. This method is less sensitive to parameter variation and we were able to show that with changing the compression rate the performance does not vary much with using Dominant set, as the clustering method. Moreover, with applying SRL as a preprocessing for dominantset clustering, by increasing the size of original graph, the time efficiency of our method gets clearer.

We have proposed a novel matrix factorization method as a second technique in section IV for graph and kernel compression. We have tested our proposed method on different scenarios like link prediction in the presence of missing observations. We came up with an update rule which resembles the ones adopted for NMF and the one introduced by Baum and Eagon for maximizing polynomials with nonnegative coefficients in probability domain. We have proven convergence properties of these update rules. Our graph compression can be potentially employed to speed-up existing graph-based and kernel-based algorithms however it was not our focus to show the effectiveness of our graph compression formulation. However, the experiments conducted in Section 4.7.2 evidenced that the information preserved by the compression is sufficient to avoid drastic decay of the algorithm's performances. Actually, we experienced a constant performance even at low compression rates. The complexity of many algorithms can be reduced by replacing $\mathbf{G}$ with the factorization $\mathbf{X}^\top\mathbf{R}\mathbf{X}$ and by exploiting the sparsity of $\mathbf{X}$. As an example, the complexity of a matrix-vector multiplication can be reduced from n^2 to (k^2+n). However, for this to become a real benefit we have to reduce the complexity of the compression algorithm.

We have proposed a two steps method to estimate the number of cluster which is presented in appendix V. Based on the Motzkin-Straus theorem, they were able to show a connection between clique number ($\omega(G)$) and the global optimal value of a certain quadratic function over the standard simplex. We have extended their result to the weighted case by designing a two steps method to determine the number of clusters. In the first step, we use dissimilarity matrix as an input and by minimizing it with replicator, we are able to detect the minimum number of clusters based on our defined stability number. And then, we examine the existence of undetected cluster based on the idea of "Efficient-out-of-sample extension of dominant-set clusters" paper. We have proved the performance of our method on several synthetic and real world benchmarks in UCI and social network dataset.

Comparison of two Compression methods:

1. In the contest of clustering, Szemérdi Regularity method cannot be applied as a compression device and clustering method in the same time because it is impossible to think about a clustering procedure that forces all the clusters to have the same cardinality and another reason is this method cant produce a good grouping because of its nature that doesnt consider inter- similarity. In the other hand MFA method can be used as clustering method as we have shown in section. The main reason is based on the grouping strategy of its nature.

2. The SLR method can be applied to a large graph as its nature but MFA method it is not very efficient for the large graph. It is computationally expensive because iterative optimization procedure.

3. In SLR method focuses on inter-sets edges in its nature, while nothing has been said about intra-subset connections. In the case of MFA, it focuses in both inter and intra connection. In another words, in classical graph clustering, vertices that are well connected with each other are grouped together like MFA method, whereas in a regular decomposition, members of a group have stochastically similar relations to all other groups. In fact, the internal edge structure within a group is not considered at all in SRL.

4. Finally, one of cones about are work is we didn't apply our SRL method in different applications. Moreover, as far as both methods are lossy, we could investigate the ability of each method to how much compress and loose information for reconstructing back.

C.2 Impact and Future Works

To improve MFA approach to become a real benefit we have to reduce the complexity of the compression algorithm. To this end, a future direction of research is to exploit randomized algorithms for speeding-up the factorization and thus the compression step.

As future works we plan also to consider other types of Bregman divergences in the objective of (4.3), rather than simply ℓ_2 divergence. Another intriguing

direction of research consists in investigating generalizations or characterizations of the multiplicative update rule that we proposed in this thesis.

Finally, another interesting development of our method that we leave to future research consists in selecting automatically an optimal compression rate. This can be achieved by imposing a Occam's razor prior on the model complexity, which favours simple models (smaller values of k) as opposed to complex ones.

Bibliography

[1] Micah Adler and Michael Mitzenmacher. Towards compressing web graphs. In *In Proc. of the IEEE Data Compression Conference (DCC*, pages 203–212, 2000.

[2] Edoardo M. Airoldi, David M. Blei, Stephen E. Fienberg, and Eric P. Xing. Mixed Membership Stochastic Blockmodels. *J. Mach. Learn. Res.*, 9:1981–2014, 2008.

[3] N. Alon, R. A. Duke, H. Lefmann, V. Rödl, and R. Yuster. The algorithmic aspects of the regularity lemma. *Journal of Algorithms*, 16(1):80 – 109, 1994.

[4] Noga Alon and Raphael Yuster. Almost$H$-factors in dense graphs. *Graphs and Combinatorics*, 8(2):95–102, 1992.

[5] L. E. Baum and J. A. Eagon. An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology. *Bull. Amer. Math. Soc.*, 73:360–363, 1967.

[6] E.M.L. Beale. *Euclidean Cluster Analysis*. Scientific Control Systems Limited, 1969.

[7] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *J. Machine Learning Research*, 3:993–1022, 2003.

[8] Béla Bollobas. *Extremal Graph Theory*. Dover Publications, Incorporated, 2004.

[9] I. Bomze, M. Pelillo, and V. Stix. Approximating the maximum weight clique using replicator dynamics. *IEEE Trans. Neural Networks*, 11:1228–1241, 2000.

[10] Immanuel M. Bomze. Evolution towards the maximum clique. *J. of Global Optimization*, 10(2):143–164, March 1997.

[11] D. Cai, X. He, J. Han, and T. S. Huang. Graph regularized nonnegative matrix factorization for data representation. *IEEE Trans. Patt. Analysis Machine Intell.*, 33(8):1548–1560, 2011.

[12] T. Caliński and J. Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-Simulation and Computation*, 3(1):1–27, 1974.

[13] Y. Choi and W. Szpankowski. Compression of graphical structures: Fundamental limits, algorithms, and experiments. *IEEE Trans. Information Theory*, 58(2):620–638, 2012.

[14] Yongwook Choi and Wojciech Szpankowski. Compression of graphical structures. In *IEEE International Symposium on Information Theory, ISIT 2009, June 28 - July 3, 2009, Seoul, Korea, Proceedings*, pages 364–368, 2009.

[15] Thomas H. Cormen, Clifford Stein, Ronald L. Rivest, and Charles E. Leiserson. *Introduction to Algorithms*. McGraw-Hill Higher Education, 2nd edition, 2001.

[16] Andrzej Czygrinow and Vojtech Rdl. An algorithmic regularity lemma for hypergraphs. *SIAM J. Comput.*, 30(4):1041–1066, 2000.

[17] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *J. American Soc. for Information Science*, 41(6):391–407, 1990.

[18] I. S. Dhillon, Y. Guan, and B. Kulis. Kernel k-means: spectral clustering and normalized cuts. In *Int. Conf. on Knowledge Discovery and Data Mining*, volume 10, pages 551–556, 2004.

[19] Reinhard Diestel. *Graph Theory*, volume 173 of *Graduate Texts in Mathematics*. Springer-Verlag, Heidelberg, third edition, 2005.

[20] C. Ding, X. He, and H. D. Simon. On the equivalence of nonnegative matrix factorization and spectral clustering. In *SIAM Data Mining Conf.*, pages 606–610, 2005.

[21] C. Ding, T. Li, and M. I. Jordan. Nonnegative matrix factorization for combinatorial optimization: spectral clustering, graph matching and clique finding. In *IEEE Int. Conf. on Data Mining*, pages 183–192, 2008.

[22] C. Ding, T. Li, and W. Peng. On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing. *Comput. Stat. & Data Analysis*, 52(8):3913–3927, 2008.

[23] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. John Willey & Sons, New Yotk, 1973.

[24] Toms Feder and Rajeev Motwani. Clique partitions, graph compression, and speeding-up algorithms. In *STOC'91*, pages 123–133, 1991.

[25] Bernd Fischer and Joachim M. Buhmann. Path-based clustering for grouping of smooth curves and texture segmentation. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 25(4):513–518, 2003.

[26] A. Frieze. The regularity lemma and approximation schemes for dense problems. In *FOCS '96: Proceedings of the 37th Annual Symposium on Foundations of Computer Science*, Washington, DC, USA, 1996. IEEE Computer Society.

[27] Alan M. Frieze and Ravi Kannan. The regularity lemma and approximation schemes for dense problems. In *37th Annual Symposium on Foundations of Computer Science, FOCS '96, Burlington, Vermont, USA, 14-16 October, 1996*, pages 12–20, 1996.

[28] A. C. Gilbert and K. Levchenko. Compressing network graphs. In *Proceedings of the LinkKDD workshop at the 10th ACM Conference on KDD*, August 2004.

[29] Anna Goldenberg, Alice X. Zheng, Stephen E. Fienberg, and Edoardo M. Airoldi. A survey of statistical network models. *Found. Trends Mach. Learn.*, 2(2):129–233, 2010.

[30] A.D. Gordon. *Classification, 2nd Edition*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Taylor & Francis, 1999.

[31] W.T. Gowers. Hypergraph regularity and the multidimensional szemerèdi theorem. *Random Struct. Algorithms*, 2004.

[32] Peter D. Grünwald. *The Minimum Description Length Principle (Adaptive Computation and Machine Learning)*. The MIT Press, 2007.

[33] I. Norros H. Reittu, F. Bazsó. A graph compression algorithm inspired by szemerédis regularity lemma. *ArXiv preprint*, 2014.

[34] John A. Hartigan. *Clustering Algorithms*. John Wiley & Sons, Inc., New York, NY, USA, 99th edition, 1975.

[35] Penny E. Haxell, Brendan Nagle, and Vojtech Rödl. An algorithmic version of the hypergraph regularity method. In *23-25 October 2005, Pittsburgh, PA, USA, Proceedings*, pages 439–448, 2005.

[36] T. Hofmann. Learning the similarity of documents : an information-geometric approach to document retrieval and categorization. In *Advances in Neural Inform. Process. Syst.*, volume 12, pages 914–920, 2000.

[37] P. W. Holland, K. B. Laskey, and S. Leinhardt. Stochastic blockmodels: First steps. *Social Networks*, 5(2):109–137, 1983.

[38] R. A. Horn and C. R. Johnson. *Matrix analysis*. Cambridge University Press, 1985.

[39] Tim J. P. Hubbard, Alexey G. Murzin, Steven E. Brenner, and Cyrus Chothia. SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J. Molec. Biology*, 247:536–540, 1995.

[40] Wei Jin and Rohini K. Srihari. Graph-based text representation and knowledge discovery. In *Proceedings of the 2007 ACM Symposium on Applied Computing*, SAC '07, pages 807–811, New York, NY, USA, 2007. ACM.

[41] I. Jolliffe. *Principal Component Analysis*. Springer Verlag, 1987.

[42] Hamid Khalili, Amir Yahyavi, and Farhad Oroumchian. Web-graph pre-compression for similarity based algorithms. In *Third International Conference on Modeling, Simulation and Applied Optimization 2009, Sharjah, U.A.E, 2009*, pages 20–22, 2009.

[43] János Komlós and Miklós Simonovits. Szemerédi's regularity lemma and its applications in graph theory, 1996.

[44] Robert Krauthgamer and Tamar Zondiner. Preserving terminal distances using minors. *CoRR*, 2012.

[45] D. Kuang, H. Park, and C. Ding. Symmetric nonnegative matrix factorization for graph clustering. In *SIAM Int. Conf. Data Mining*, pages 106–117, 2012.

[46] B. Lakshminarayanan and R. Raich. Non-negative matrix factorization for parameter estimation in hidden Markov models. In *IEEE Int. Workshop on Machine Learning for Signal Processing*, pages 89–94, 2010.

[47] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999.

[48] D. D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural Inform. Process. Syst.*, pages 556–562, 2000.

[49] Ping Li, Jiajun Bu, Yi Yang, Rongrong Ji, Chun Chen, and Deng Cai. Discriminative orthogonal nonnegative matrix factorization with flexibility for data representation. *Expert Systems with Applications*, 41(4):1283–1293, 2014.

[50] Ping Li, Chun Chen, and Jiajun Bu. Clustering analysis using manifold kernel concept factorization. *Neural Computation*, 87:120–131, 2012.

[51] F. Lorrain and H. C. White. Structural equivalence of individuals in social networks. *J. of Math. Sociology*, 1:49–80, 1971.

[52] J. Maynard Smith. *Evolution and the Theory of Games*. Cambridge University Press, Cambridge, UK, 1982.

[53] Glenn Milligan and Martha Cooper. An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, 50(2):159–179, 1985.

[54] M. Mørup and M. Schmidt. Bayesian community detection. *Neural Computation*, 24(9):2434–2456, 2012.

[55] T. S. Motzkin and E. G. Straus. Maxima for graphs and a new proof of a theorem of Turán. *Canadian Journal of Mathematics*, 17:533–540, 1965.

[56] Theodore S Motzkin and Ernst G Straus. Maxima for graphs and a new proof of a theorem of turán. *Canad. J. Math*, 17(4):533–540, 1965.

[57] S. Navlakha, R. Rastogi, and N. Shrivastava. Graph summarization with bounded error. In *ACM SIGMOD Int. Conf. on Management of Data*, pages 419–432, 2008.

[58] Tamás Nepusz, Andrea Petróczi, László Négyessy, and Fülöp Bazsó. Fuzzy communities and the concept of bridgeness in complex networks. *Phys. Rev. E*, 77(1):016107, 2008.

[59] F. Nourbakhsh, S. Rota Bulò, and M. Pelillo. A matrix factorization approach to graph compression. In *Int. Conf. Patt. Recogn.*, 2014.

[60] P. Paatero and A. U. Tapper. Positive matrix factorization: a non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5:111126, 1994.

[61] Panos M Pardalos and AT Phillips. A global optimization approach for solving the maximum clique problem. *International Journal of Computer Mathematics*, 33(3-4):209–216, 1990.

[62] M. Pavan and M. Pelillo. Dominant sets and hierarhical clustering. In *Int. Conf. Comp. Vision*, volume 1, pages 362–369, 2003.

[63] M. Pavan and M. Pelillo. Efficient out-of-sample extension of dominant-set clusters. *Advances in Neural Inform. Process. Syst.*, 17:1057–1064, 2005.

[64] Massimiliano Pavan and Marcello Pelillo. Graph-theoretic approach to clustering and segmentation. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2003), 16-22 June 2003, Madison, WI, USA*, pages 145–152, 2003.

[65] Marcello Pelillo. A unifying framework for relational structure matching. In *Fourteenth International Conference on Pattern Recognition, ICPR 1998, Brisbane, Australia, 16-20 August, 1998*, pages 1316–1319, 1998.

[66] Marcello Pelillo. Replicator equations, maximal cliques, and graph isomorphism. *Neural Computation*, 11(8):1933–1955, 1999.

[67] Marcello Pelillo. Metrics for attributed graphs based on the maximal similarity common subgraph. *Int. J. Pattern Recognition Artif. Intell*, 2004:299–313, 2004.

[68] Marcello Pelillo and Arun Jagota. Feasible and infeasible maxima in a quadratic program for maximum clique. *J. Artif. Neural Networks*, 2:411–420, 1995.

[69] I. Psorakis, S. Roberts, M. Ebden, and B. Sheldon. Overlapping community detection using nonnegative matrix factorization. *Phys. Rev. E*, 83(6):066114, 2011.

[70] Vojtech Rödl and Jozef Skokan. Applications of the regularity lemma for uniform hypergraphs. *Random Struct. Algorithms*, 28(2):180–194, 2006.

[71] Samuel Rota Bulò, André Lourenço, Ana L. N. Fred, and Marcello Pelillo. Pairwise probabilistic clustering using evidence accumulation. In *Int. Work. on Struct. and Synt. Patt. Recogn.*, pages 395–404, 2010.

[72] Samuel Rota Bulò and Marcello Pelillo. Probabilistic clustering using the baum-eagon inequality. In *Int. Conf. Patt. Recogn.*, pages 1429–1432, 2010.

[73] Peter Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.*, 20(1):53–65, November 1987.

[74] L. Samuelson. Evolutionary games and equilibrium selection. *MIT Press, Cambridge, MA*, 1997.

[75] B. Schölkopf, A. Smola, and K. R. MÏller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

[76] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. Patt. Analysis Machine Intell.*, 22:888–905, 2000.

[77] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:888–905, 1997.

[78] A. Sperotto and M. Pelillo. Szemerédis regularity lemma and its applications to pairwise clustering and segmentation. In *Energy Minim. Methods in Computer Vision and Patt. Recogn.*, pages 13–27, 2007.

[79] Torsten Suel and Jun Yuan. Compressing the graph structure of the web. In *Data Compression Conference, DCC 2001, Snowbird, Utah, USA, March 27-29, 2001.*, pages 213–222, 2001.

[80] E. Szemerédi. Regular partitions of graphs. In *Problèmes combinatoires et thorie des graphes*, pages 399–401. CNRS, Paris, 1978.

[81] Endre Szemerédi. On sets of integers containing no k elements in arithmetic progression. *ICM: 2nd International Congress of Mathematicians*, 1975.

[82] Endre Szemerédi. Regular partitions of graphs. *Technical report, Stanford, CA, USA*, 1975.

[83] Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the number of clusters in a dataset via the gap statistic. 63:411–423, 2000.

[84] H. Toivonen, F. Zhou, A. Hartikainen, and A. Hinkka. Compression of weighted graphs. In *Int. Conf. on Knowledge Discovery and Data Mining*, pages 965–973, 2011.

[85] Hannu Toivonen, Fang Zhou, Aleksi Hartikainen, and Atte Hinkka. Compression of weighted graphs. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '11, pages 965–973, New York, NY, USA, 2011. ACM.

[86] D. Verma and M. Meila. Comparison of spectral clustering methods. Technical report, University of Washington, 2003.

[87] W. Xu and Y. Gong. Document clustering by concept factorization. In *Proc. of 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 202–209, 2004.

[88] Mingjin Yan. *Methods of Determining the Number of Clusters in a Data Set and a New Clustering Criterion*. PhD thesis, Virginia Polytechnic Institute and State University,Blacksburg, Virginia, 2005.

[89] Z. Yang and E. Oja. Quadratic nonnegative matrix factorization. *Pattern Recogn.*, 45(4):1500–1510, 2012.

[90] H. Zhang, Z. Yang, and E. Oja. Adaptive multiplicative updates for quadratic nonnegative matrix factorization. *Neural Computation*, 134:206–213, 2014.