Classification of signals and graphs by algebraic spectral approaches

Hadj Ahmed Bay-Ahmed

UNIVERSITE SCIENCES BRETAGNE DE LA MER LOIRE ET DU LITTORAL

UBO Université de Bretagne Occidentale

THESE DE DOCTORAT DE

L'UNIVERSITE DE BRETAGNE OCCIDENTALE
COMUE UNIVERSITE BRETAGNE LOIRE

ECOLE DOCTORALE N° 598
Sciences de la Mer et du littoral
Spécialité : *Acoustique sous-marine et traitement du signal*

Par
Hadj Ahmed BAY-AHMED

Classification des Signaux et des Graphes par Approches Spectrales Algébriques

Thèse présentée et soutenue à l'Ecole Navale, le 14 Décembre 2018
Unité de recherche : Institut de Recherche de l'Ecole Navale (IRENav) EA 3634

Rapporteurs avant soutenance :

Pierre BORGNAT	Directeur de Recherche CNRS,ENS Lyon, Laboratoire de Physique, UMR 5672
Cédric RICHARD	Professeur des Universités, Université de Nice Sophia-Antipolis, Observatoire de la Côte d'Azur, UMR 7293

Composition du Jury :

Président :

Pierre CHAINAIS	Professeur des Universités, Ecole Centrale de Lille, CRIStAL, UMR 9189

Examinateurs :

Pierre BORGNAT	Directeur de Recherche CNRS, ENS Lyon, Laboratoire de Physique, UMR 5672
Cédric RICHARD	Professeur des Universités, Université de Nice Sophia-Antipolis, Observatoire de la Côte d'Azur, UMR 7293
Abderrahim ELMOATAZ	Professeur des Universités, Université de Caen Basse Normandie, GREYC, UMR 6072
Rémi GRIBONVAL	Directeur de Recherche INRIA, INRIA Rennes-Bretagne Atlantique
Gilles BUREL	Professeur des Universités, Université de Brest Occidentale, Lab-STICC, UMR 6285
Delphine DARE-EMZIVAT	Maître de Conférences, Ecole Navale, IRENav, EA 3634

Dir. de thèse :

Abdel BOUDRAA	MCF (HdR),Ecole Navale, IRENav, EA 3634

Contents

List of Figures

List of Tables

Abbreviations

TV : Total Variation

TVG : Graph Total Variation

GE : Graph Energy

JET : Joint Energy and Total-variation

JSS : Joint Spectral Similarity

VG : Visibility Graph

VN : Von Neumann entropy

$\mathcal{N}$: Gaussian distribution

A : Adjacency matrix

L : Laplacian matrix

D : Degrees matrix

VPV : Von-Neumann Perturbation Velnurability

DGA : Dominent Graph Analysis

List of publications

International Journal Papers:

1. **H.A. Bay-Ahmed**, A.O. Boudraa, and D. Dare, "A joint spectral similarity measure for graphs classification", *Pattern Recognition Letters (Accepted, minor revisions)*, 2018.

Conferences Papers:

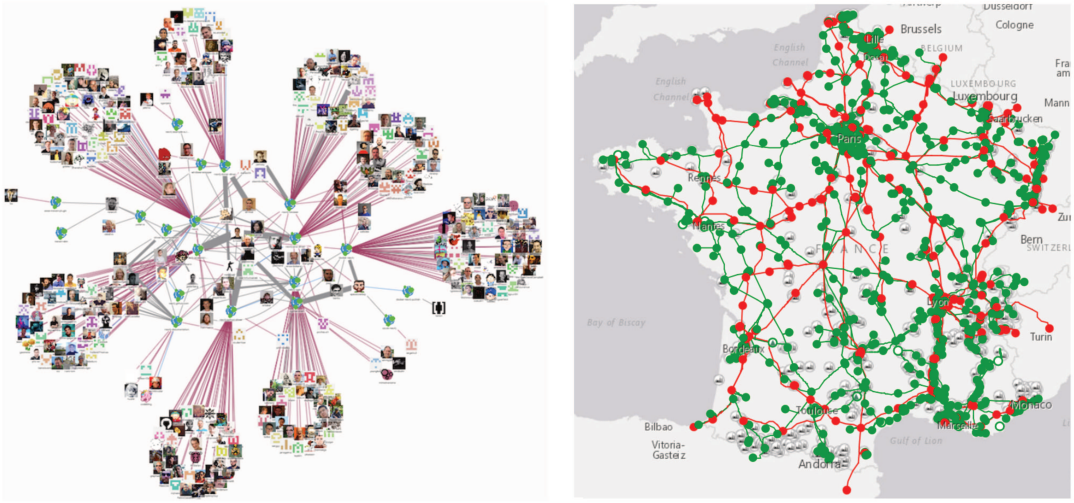
1. **H.A. Bay-Ahmed**, D. Dare-Emzivat and A.O. Boudraa, "Graph signals classification using total variation and graph energy informations", *In Proc. IEEE GlobalSIP*, pages 668-671, 2017.
2. **H.A. Bay-Ahmed**, D. Dare-Emzivat, A.O. Boudraa and Y. Préaux, "Classification des signaux sur graphes par mesures spectrales algébriques", *In Proc. GRETSI*, pages 1-4, 2017.

Preamble

1.1 Context

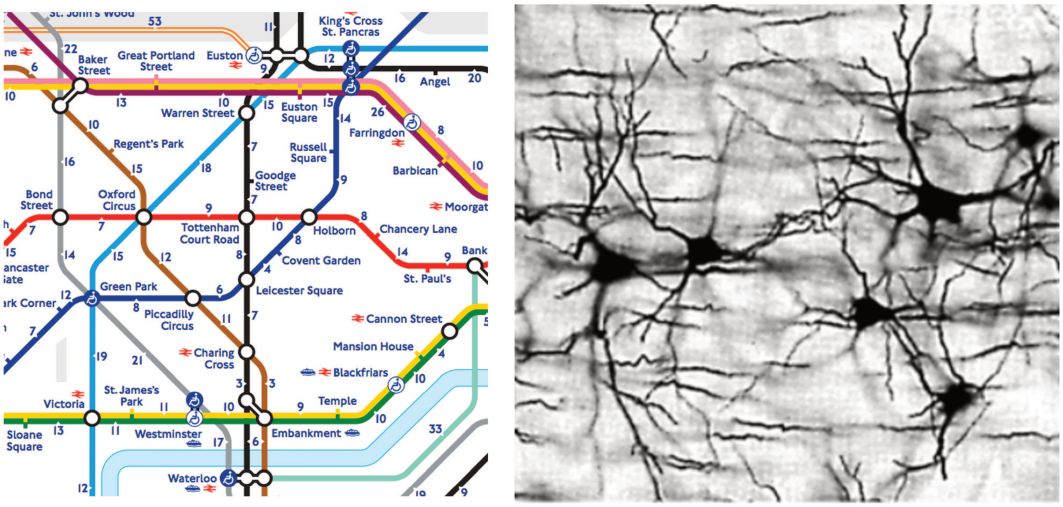
Nowadays, with the development of electronic instrumentation, computing and communications systems, world's technological capacity to store information has considerably increased, resulting in the accumulation of a huge amount of heterogeneous and structured data, often issued from sensor networks or natural phenomena showing up under irregular and complex forms. As long as sensors are installed everywhere, in urban centers to monitor pollution, noise, traffic, weather parameters, as well as in industrial, nuclear and high risk areas to track toxicity and radioactivity levels. Such data is particularly complex and irregular, depending both on the way sensors are deployed and on the adopted acquisition architecture of the system. Therefore, their analysis and processing require adapted and pertinent mathematical tools, such as graphs, which are a generic algebraic structures, useful for describing complex geometric structures and connections inhered in data. Formally, graphs are composed of nodes representing entities of interest, connected between them by a set of weighted edges. While connectivities are either dictated by the physics of the underlying problem or inferred from the data. If the edges have a physical meaning as in transport networks (roads, rails), communication and energy networks or in neural networks (i.e. Figure 0.2(b)), graphs are said to be natural. Unlike sensors, for social networks where edges refer to a virtual and logical relation, graphs are said to be conceptual. For instance, social networks (i.e. Figure 0.1(a)) are modeled by graphs in which edges represent a sort of social interaction between users. Whereas, in infrastructure networks (i.e. Figures 0.1(b), 0.2(a)), nodes represent important entities like metro stations, electric production plants, electric subways or logistic supply and storage spots.

The majority of real world graphs are labeled, that is numbers or symbols are associated to their nodes. Therefore, the emerging field of graph signal processing, D.I. Shuman et al., 2013, aims to develop suitable tools to explore and analyze such graphs in which nodes are indexed by real/complex numbers. These tools include Fourier transform A. Sandryhaila and J.M.F. Moura, 2014, filters S. Segarra et al., 2015, N. Tremblay and P. Borgnat, 2016, adaptive filters S. Chen et al., 2013, sampling algorithms S.



(a) (b)

Figure 0.1: (a) Software developers communities in Neo4j and GitHub platforms [https://neo4j.com/blog/meta-exploring-neo4j-graph-database/]. (b) The French electricity transmission network (high and very high voltage lines), as well as the planned structures like the lines, substations and power plants [https://www.rte-france.com/fr/la-carte-du-reseau].



(a) (b)

Figure 0.2: (a) A part of the London's "walk the Tube" map, which reveals the real distance between stations [https://www.theverge.com/2015/11/11/9712376/london-walk-tube-underground-map]. (b) Brain's neural network [http://www.pollen-multimedia.com/intelligence-artificielle].

Chen et al., 2015a, empirical modal decomposition transform N. Tremblay et al., 2014 and classifiers A. Sandryhaila and J.M.F. Moura, 2013. Even Time-Frequency analysis techniques are not excluded from this generalization effort, a spectrogram and a wavelets transform are indeed proposed for graphs D.I. Shuman et al., 2016, D.K. Hammond et al., 2011. These techniques share the ultimate objective of processing the structured data taking into consideration the underlying connectivity information between

their constituent entities. Regardless than the spectral analysis of graphs in the Fourier sense, algebraic spectral theory has also been widely developed for graphs analysis. Spectral graph theory is concerned with study, understanding and exploration of the graph's properties, such as connectivity, complexity and regularity, and quantifies its information content through the eigenvalues and eigenvectors of matrices naturally associated with this graph. The set of eigenvalues of the representation matrix of the graph is called the spectrum of the graph. The basic principle dominating spectral graph analysis is to relate important invariants of graph to its spectrum. Historically, the first relation between the spectrum and the structure of a graph was discovered in 1876 by Kirchhoff when he proved his famous matrix-tree theorem Kirchhoff, 1876. Given a graph, one may associate a variety of matrices with the graph. If we have a matrix that is naturally associated with a graph, the properties of this matrix, and therefore of the graph will be revealed, through an eigen-decomposition analysis, by its eigenvalues and eigenvectors Spielman, 2011. Depending on the used matrix, different informations such as the complexity or connectivity of the graph can be retrieved. Various parameters or measures such as graph energy, entropy and spectral moments of graph can be extracted or captured from eigenvalues of its representation matrix. A remarkable application of eigenvalues is in chemistry domain, where there is a close correspondence between the graph eigenvalues and the molecular orbital levels of π-electrons in conjugated hydrocarbons B. Zhou et al., 2007.

The effervescence that graph theory is experiencing does not spare the field of machine learning, which has undergone a real revolution recently, gaining a significant momentum in both academia and industry. In particular, statistical learning techniques have an enormous impact on several computer science and engineering fields, including objects recognition, speech recognition, natural language processing, robotics, autonomous cars and even drug design in bioinformatics. The real development of statistical learning came after 1986, when D.E. Rumelhart et al., 1986 proposed the nonlinear backpropagation algorithm which allowed the adaptation of all weights in a neural network for minimizing locally the error in a set of vectors belonging to a given pattern recognition problem. Since that, a variety of algorithms based on neural neutworks was proposed, like the Convolutional Neural Networks (CNNs) A. Dosovitskiy et al., 2014, Recurrent Neural Networks (RNNs) I. Sutskever et al., 2011, Deep Boltzmann Machines M.A. Côté and H. Larochelle, 2016 and Deep Reinforcement Learning H. Van Hasselt et al., 2016. Prior to 2006, searching the parameters space of the deep architecture was challenging and computationally costly. Meanwhile, other learning algorithms were gaining in popularity such as the Support Vector Machines (SVMs) Vapnik, 1998 and Conditional Random Field (CRF) J. Lafferty et al., 2001, because of their computational cost compared to neural networks and their success in many real world problems, like the handwritten pattern recognition problem and others. SVM technique is one of the most powerful learning techniques, based on the idea of mapping data points to a high dimensional

feature space where a separating hyperplane can be found by maximizing the distance (margin) between opposite classes. This mapping can be carried on by applying the so called kernel trick J. Shawe-Taylor and N. Cristianini, 2004, Vapnik, 2013, which implicitly transforms the input space into another high dimensional feature space.

Kernel trick permits the adaptation of the SVM algorithm into the structured data, until then applicable only on vectorial data. It replaces dot products present in the optimization problem by valid kernel functions, this principle is rather well known for potential function classifiers M.A. Aizerman et al., 1964, and it was revisited to construct SVMs as a generalization of large margin classifiers, B.E. Boser et al., 1992. So far, many kernel machines have been built to classify graph data, among them we evoke S.V.N. Vishwanathan et al., 2010, L. Bai et al., 2015a, N. M. Kriege et al., 2016. The bottleneck in the design of graph kernels is to determine a way to measure the similarity between graphs. Some of earliest work in the subject were undertaken by L. G. Shapiro and R. M. Haralick, 1985, who showed how string edit distance could be extended to graph structures. The idea is to measure the similarity of graphs by counting the number of graph edit operations required to transform a graph into another. However, the computational cost of such approach grows fast for larger graphs. More general approaches using concepts from information and probability theory were indeed proposed, such the work of W. J. Christmas et al., 1995, that shows how a relaxation labeling technique can be employed to match graphs by using pairwise attributes modeled by a Gaussian distribution. We should also mention the work of R. Myers et al., 2000, which uses maximum a posteriori estimation to perform purely structural graph comparison, but it needs some adequate probabilistic setup to optimize performance. In many cases, graphs could be labeled by a sort of strings. For instance B. Cao et al., 2013 use the depth-first search (DFS) algorithm as a graph labeling approach, and measure the similarity by the distance between the two DFS sequences, hence, by this way, the graph matching problem is turned to a string matching problem. More other structural comparison approaches were proposed, such as the aligned subtree kernel L. Bai et al., 2015b which incorporates explicit subtree correspondences between the compared graphs, assignment kernels N. M. Kriege et al., 2016, which decompose the graphs into smaller sub-graphs and try to find the optimal bijection between them, or even those based on random walks F. Fouss et al., 2007 and quantum walks L. Bai et al., 2015a.

In this context section, we have highlighted the scientific framework covered by our research work in this thesis. Our findings make the junction between the graph spectral theory and kernel based learning techniques for graph. In the following section, motivations and issues of our research work are presented.

1.2 Motivation and Issues

So far, in graph signal processing theory, signal values are considered to be the labels associated to nodes, while the connections and the structure of the graph are encoded in one of representation matrices (adjacency $\mathbf{A}$, laplacian $\mathbf{L}$ or others). Therefore, the graph signal is often handled in two distinct parts, the vector containing the labels of nodes and the representation matrix associated to its supporting structure. For instance, D.I. Shuman et al., 2013 define the graph Fourier basis as the eigenvectors of $\mathbf{L}$ matrix and the Fourier transform as the the projection of the vector containing the signal's values on that basis. The relationship between the signal and its supporting structure is not clear at least in this Fourier analysis framework. This prompts us to ask about **the way in which the signal interacts with the graph structure?** and **is the way how the signal oscillates between nodes through the structure could be a criteria for graph discrimination?** Moreover, A. Sandryhaila and J.M.F. Moura, 2014 consider the eigenvalues of the adjacency matrix as the graph frequencies, ordered via the total variation of their corresponding eigenvectors. Given that some graphs are not determined by their spectrum C.D. Godsil and B.D. McKay, 1982, and could share the same eigenvalues with other graphs, **how can we compare such graphs via their spectra?** and **is there any way to use both the eigenspectrum and the signal properties for their comparison?**

Recent works of the literature have emphasized the importance of matrix representations for graph characterization, pointing out the advantages and the drawbacks of some spectra associated to graphs E.R. Van Dam and W.H Haemers, 2003, I. Jovanović and Z. Stanić, 2014, including, those of adjacency ($\mathbf{A}$), Laplacian ($\mathbf{L}$), signless Laplacian $\mid \mathbf{L} \mid$ and distance ($\mathbf{D}^{\mathbf{G}}$) matrices. The spectrum of $\mathbf{L}$ matrix is indeed widely studied in spectral graph theory R.K. Fan Chung, 1996a, in reason of the symmetry and positive semi-definiteness of this matrix, which is useful for determining cuts and inherent graph components. Otherwise, the spectrum of $\mathbf{A}$ matrix is mainly used for the study of regularity J.H. Koolen and H. Yu, 2011, isomorphisms D. Conte et al., 2004 and bipartition Kunegis, 2015 of graphs. In spite of the simple linear relationship between them ($\mathbf{L} = \mathbf{D} - \mathbf{A}$) where $\mathbf{D}$ is the degrees matrix, these matrices seem to reveal informations about the graph in different ways, where it appears that some details are detected only by one of them, as in the case of cospectral graphs. The question of choosing either $\mathbf{A}$ or $\mathbf{L}$ matrix for graph representation is still a subject of debate. Spielman, 2004 argues that, even the adjacency matrix is the most natural matrix to associate with a graph, it is least useful. Eigenvalues and eigenvectors are most meaningful when used to understand a natural operator or a natural quadratic form, the adjacency matrix provides neither. The same observation was made by Lau, 2015 which points out that it is not clear that the eigenvalues of $\mathbf{A}$ should cary any information about the graph properties. For instance, in graph signal processing theory, D.I. Shuman et al., 2013 define the graph Fourier basis as the eigenbasis of $\mathbf{L}$ matrix, while A. Sandryhaila and J.M.F. Moura, 2014 prefer the eigenbasis of

A obtained via a Jordan decomposition. This difference can be justified in part by the nature itself of the decomposition basis, and also by the fact that not all graphs are determined by their spectra and there is a family of graphs that shares the same spectrum in respect to some matrix representation, commonly called cospectral graphs C.D. Godsil and B.D. McKay, 1982. Therefore, **we wonder about the best matrix to choose for graphs representation? A or L matrices ?** and **could we exploit them jointly for graphs discrimination? which one of them is more suitable for graphs classification ?**

Data are often structured and thus we need to take into account the structure behind the data A. Ortega et al., 2018. To understand a graph signal the structure of the associated graph must be considered. The signal is as interesting as the graph or the network itself. We quote as examples brain connectivity and the fMRI (functional Magnetic Resonance Imaging) signals or Gene regulatory and gene expression levels. Thus, this places strong emphasis on interaction between data (signal) and graph structure. This poses the challenging problem of the development of algorithms that fruitfully leverage from the interaction signal/structure. This raises the question, **how to extract valuable information from the data using innovative approaches that handle the structure of graph via, such as, the energy of the graph, the connectivity or the complexity of the graph?** A graph is an abstract construct which can model relationships or connections (edges) between entities such as sensors (nodes). For providing richer information we take advantage of physics of the graph or the network using their quantum state representation, where quantum information is the physical information that is held in the state of the quantum system. For example, tools from statistical mechanics can be used to characterize the degree distribution for different types of complex networks. Also, statistical mechanics and information theory have been used to understand more deeply variations in network structure J. Wang et al., 2017a. One of the successes has been to use quantum spin statistics to describe the geometries for complex networks. A pertinent attribute is the network entropy used to characterize the salient feature of the network systems Jianjia Wang et al., 2017. For example the Von Neumann entropy has been used as an effective characterization of network structure, starting from a quantum analogy in which the Laplacian matrix on graphs plays the role of density matrix. **A relevant question is how to exploit the Von Neumann entropy of graph to quantify its complexity? A key challenge in this regard, is how to combine tools from spectral graph theory and from quantum mechanics for deeper understanding and analysis complex networks.**

Nowadays the issue of vulnerability and protection of critical infrastructure is attracting a great deal of attention of scientific community. In general, a critical infrastructure system is represented as a graph in which nodes represent the main components of the network (power plants,...) and edges are the physical connections among them (electrical lines,...) P.C. Crucitti et al., 2005. The topology of the

network or the graph determines an influence structure among the nodes or the agents S. Segarra and A. Ribeiro, 2016. Following the graph-based approach, different strategies have been proposed with the purpose to measure the vulnerability of the graph or to find the best nodes to immunize (or equivalently, remove) to make the remaining nodes to be most robust to virus attack C. Chen et al., 2016. Failure and attacks can be simulated as the removal of a certain percentage either of nodes of the network. Nodes or edges immunization is essential to safeguard network systems against, for example, virus attacks and its propagation. This requires the quantification of importance of individual node (edge) or group of nodes (edges) in terms of their contribution towards vulnerability. A simple metric to judge the overall graph vulnerability is the one based on the largest eigenvalue λ of adjacency matrix of the graph K. Kanwar et al., 2017. The larger λ is, the more vulnerable the whole graph is. However, this global metric or score cannot be used for identifying or localizing a vulnerable edge or a group of edges that are vulnerable of the graph. **The challenge behind this problem is how to measure the vulnerability of each edge and to provide a vulnerability map of the graph, that helps to find the effective immunization strategy to be applied.**

1.3 Thesis outline

The outline of this thesis is as follows:

In the Chapter 1, we recall some basic notions about graphs and their spectral analysis. We present the most well-known representation matrices, as well as notions related to the structure of graphs, such as regularity, connectivity and bipartition. In addition of some notions related to the eigenspectrum of the adjacency and laplacian matrices, such as the Fiedler's value, the largest eigenvalue, the energy, the Kirchoff index. We included also a reminder about low-rank matrix approximation, and some primary results about its use in finding a backbone of the structure, which we called thereafter dominant graph.

In Chapter 2, we briefly discuss the problem of machine learning from data and especially supervised statistical learning. We detail the idea behind support vector machines and their mathematical modeling. Then, we show how this algorithm could work nonlinearly and how it could be adapted to structural data using kernel trick. We explain how to build valid kernel functions and how to combine them in the support vector machine. In the last sections, we present some important kernels proposed for graphs and suitable for structured data classification.

In Chapter 3, we discuss the problem of graph similarity. We try to answer the question about the way in which the signal interacts with the structure and we wonder to know if the manner how

the signal oscillates between nodes through the structure could discriminate graphs. For this purpose, we use the total variation of the graph signal as an indicator about oscillatory behaviour and as an attribute for graphs comparison. We have also been interested in the energies of graphs and how they can characterize them spectrally, we discuss their properties and their origin. Then we use the energy based on laplacian matrix to measure the similarity of the graphs and classify them. In addition, we proposed a joint similarity measure which combines the total variation and the laplacian graph energy informations in a single measure. The goal is to exploit both the signal information and the structure of the graph in the discrimination process.

In Chapter 4, we discuss the problem of graph similarity, but from a purely spectral perspective. We begin by treating the question of graphs representation via the adjacency ($\mathbf{A}$) and laplacian ($\mathbf{L}$) matrices. We exploit the framework of matrix perturbation theory and the Von Neumann entropy of quantum systems to reveal the role of each matrix when the edge weights of the graph are perturbed. Then we propose a similarity measure which combines the spectra of both matrices for comparing graphs permetting the evaluation of the ability of each matrix to discriminate graphs in a learning process. We clarify also how the new measure handles the cospectrality problem. In the last sections we present the classification results on some bioinformatics and time series data.

In Chapter 5, we treat the problem of vulnerability in networks, which is an important problem in strategic infrastructure networks, since they need to be resilient to failures and damages caused by eventual malicious attacks. We use the changes in the Von Neumann's entropy as an indicator about the sensitivity of a given edge in the graph to perturbations. Then we use it to associate to each edge a weight indicating its fragility and its importance in the structure viewed from an entropic perspective.

We conclude this dissertation with a general discussion about the outcomes of our work and the promising perspectives to investigate further in futur researchs.

1.4 Contributions

The main contributions of this thesis are summarized as follows:

⇒ We propose the following new graph similarity measures:

- TVG: is a measure based on total variation (TV) of the graph signal. It quantifies the oscillatory behaviour of the graph signal and its interaction with the supporting structure. We show that

it is an interesting informative and simple descriptor for graph signals comparison.

- GE: is a measure based on the laplacian graph energy which is calculated via the laplacian eigenspectrum of the graph. It is a pertinent information that characterizes well the graph, and measures the complexity degree of its structure, taking into account both connections distribution of the network and its density.
- JET: is a joint convex combination between the TVG and GE measures to take advantage from both. It allows to take into consideration the signal's properties and the complexity information about the supporting structure.
- JSS: is our second joint graphs similarity measure, which exploits both spectral informations from adjacency $\mathbf{A}$ and laplacian $\mathbf{L}$ matrices. The $\mathbf{A}$ matrix characterizes the topological graph complexity in terms of connections between nodes and underscores the local cohesiveness of nodes, while $\mathbf{L}$ matrix is well suited for recovering some information about clusters and communities in the graph, thus, capture its inherent structure. The JSS incorporates both advantages those of $\mathbf{A}$ and $\mathbf{L}$.

$\Rightarrow$ We integrate our similarity measures (TVG, GE, JET and JSS) in an exponential kernel, which we use in the SVM learning algorithm to classify graphs issued from bioinformatics and time series. Compared to the state-of-art methods, our measures are of low complexity and fast to run. We show that with simple pertinent global descriptors, we could do better than other complex methods. Via the (JET and JSS) measures we show that linear combination of multiple measures increases often the graphs discrimination power and enhances classification performance.

$\Rightarrow$ The JSS measure allow us to confirm our intuition that $\mathbf{A}$ and $\mathbf{L}$ matrices contribute unequally in graph characterization task, and to emphasize the fact that they represent differently the structural information about the underlying graph. In spite of the simple linear relationship between them ($\mathbf{L} = \mathbf{D} - \mathbf{A}$), these two matrices give rise to different inferences drawn from the graph.

$\Rightarrow$ We highlight the overlapping and the unequal contributions of ($\mathbf{A}$) and ($\mathbf{L}$) for graph representation, by comparing them in terms of the so called Von Neumann entropy, connectivity and complexity measures. The graph is viewed as a quantum system and thus, the calculated Von Neumann entropy of its perturbed density matrix emphasizes the overlapping in terms of information quantity.

$\Rightarrow$ We illustrate by classification findings on real and conceptual graphs the effectiveness of the JSS measure in terms of classification accuracies, and by which we highlight the varying information overlapping rates of $\mathbf{A}$ and $\mathbf{L}$ via a weighting parameter α, and we point out their different ways in recovering structural information of the graph.

⇒ We show that the JET and JSS measures handle the graph cospectrality issue, and they allow the distinction between graphs that share the same eigenvalues spectrum corresponding to **A** or **L** matrices.

⇒ We show that converting time series to graphs using VG algorithm (L. Lacasa et al., 2008) could enhance the classification accuracy of these series, and permits the application of graph kernels in the learning process.

⇒ We use the Von Neumann entropy to show that the edges of a given graph do not react to perturbations in the same way, and that their sensitivity to noise is not the same. We use the entropy distortion to score the vulnerability of each edge, and to formalize a graph weighting algorithm which we called **VPV**−*weighting*. For instance, our approach is useful for networks diagnostics and to study their resilience to malicious attacks and damages due to failures.

⇒ We use the Low-Rank matrix approximation to define the salient structure of the graph, which we refer to as Dominant Graph Component (**DGA**).

CHAPTER 1

Reminder about Spectral Graph Analysis

In this chapter, basics and principal tools of spectral graph analysis R.K. Fan Chung, 1996b related to problems tackled in this thesis are presented. Notions of graph theory, some basic definitions are given and sometimes illustrated with short examples. We recall the theorem of real symmetric matrices and the properties of the associated eigenvectors and eigenvalues. The most natural matrices of graphs representation namely the adjacency, the Laplacian and the normalized Laplacian matrices are presented and their relationships detailed. Proprieties of the eigen-spectrum of these matrices are analyzed. The spectral moments of the graphs are presented and particularly the concept energy of graph. The Rayleigh quotient is presented followed by the Courant-Fischer theorem which is a powerful tool for characterization of eigenvalues of matrices is detailed. Fiedler's theory of spectral graph partitioning is recalled and emphasis is placed on the Fiedler value and the Fielder vector with an illustration on graph clustering. The low rank approximation of matrices is presented. Using this compact represenation, a Reynolds like-decomposition of graphs is introduced. Furthremore, based on this approximation two nearest adjacency matrix retrieving strategies are proposed. First strategy is based on the minimization of the Frobenius norm and the second one is based on the maximization of the Kirchhoff index. This last is illustrated on communities graph detection. Also, using the low rank approximation a new graph analysis approach, called Dominant eigenGraph Analysis (DGA), that consists in reaveling the more prominent substruture of the graph or salient eingengraph is as well proposed. The proposed tools are illustrated on simulated graphs, and overall, the obtained preliminary results show their interest for graph analysis purposes.

1.1 Basic notions

A graph is an abstract construct which can model relationships (edges) between entities (vertices). In this thesis, the graphs under consideration are simple graphs, namely, finite graphs without loops or parallel edges.

Definitions 1

Formally, a weighted graph or network is denoted by the triplet $G = (V, \mathcal{E}, \mathbf{W})$ with a finite set of nodes or vertices $V = \{v_1, v_2, \ldots, v_n\}$ where $n =\mid V \mid$ and a set of edges defined as pairs (v_i, v_j) noted $\mathcal{E} = \{e_{ij}\} = \{(v_i, v_j) \mid i, j = 1, 2, \ldots n; i \neq j\} \subseteq V \times V$ with $m =\mid \mathcal{E} \mid$, where $\mid . \mid$ denotes the cardinality of a set. An edge e_{ij} connects vertices v_i and v_j if they are adjacent or neighbors. The in-neighbors of i is noted by $\mathcal{N}(i) = \{j \in V : (i, j) \in \mathcal{E}\}$. Matrix of weights $\mathbf{W}$ is constructed using the mapping, $\mathcal{E} \to \mathbb{R}$, from the set of edges to scalar w_{ij}, that represents the level of relationship (or strength of relationship) from i to j. This suppose that there is a weight function, w, mapping every edge to a real number. In many applications, the weight w_{ij} associated to edge e_{ij} is usually, a non-negative integer. An illustration of weighted graph is given in figure 1.1. A weighted graph is therefore a special type of labeled graph in which the labels are numbers. Prototypical examples of weighted graphs or networks can be found in the world-wide airport network and scientific collaboration network A. Barrat et al., 2004. In the aiport network, each given weight w_{ij} is the number of available seats on direct flights connections between the airports i and j. A graph G is *unweighed* when $w_{ij} \in \{0, 1\}$ for all $(i, j) \in \mathcal{E}$, and will be noted as

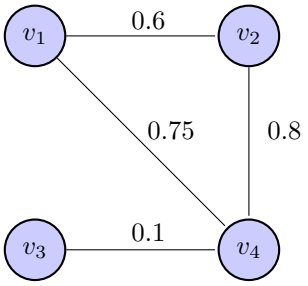


Figure 1.1: Example of a weighted graph.

$G = (V, \mathcal{E})$. A graph G is called *undirected*, if $(i, j) \in \mathcal{E}$ implies $(j, i) \in \mathcal{E}$, $w_{ij} = w_{ji}$ for all $(i, j) \in \mathcal{E}$.

1.1.1 Some particular graphs

Particular relationships between edges and nodes lead to constitute some specific graphs classes. We recall here, examples of basic graphs. If all the nodes of G have the same degree $d(v_i)$, G is a *regular graph*. Figure 1.2 illustrates the case of a regular graph of degree $'2'$, each of its nodes has degree 2. A

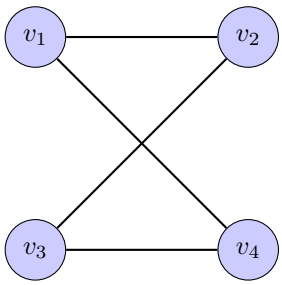


Figure 1.2: Example of a regular graph with $d(v_i) = 2$.

graph G is *complete* if there is one edge between every pair of vertices. The complete graph on $'n'$ nodees is denoted by K_nK_n has $n \times (n-1)/2$ edges and is a regular graph of degree $(n-1)$. As illustrated in figure 1.3, a K_4-graph has 6 edges and is a regular graph of degree 3.

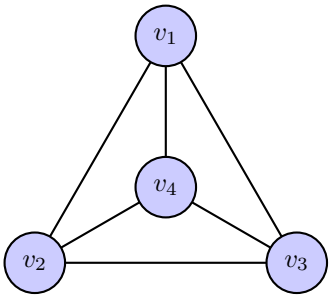


Figure 1.3: Example of a complete graph K_4.

A *Bipartite graph* (see figure 1.4) is a set of graph nodes such that the nodes that are in the same group have no edges between them (i.e $v_i \in V$ and $v'_i \in V'$). It is a graph in which the nodes can be put into two separate groups so that only the edges exist between those two groups (V and V'), and there are no edges between nodes within the same group.

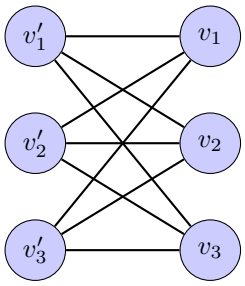


Figure 1.4: Example of a bipartite graph with $n = 6$ and $m = 9$.

An ordered sequence of connected nodes $(v_i, v_{i+1}, \ldots, v_j)$ that starts with node v_i and ends with node v_j forms a *path* between v_i and v_j. An example of path starting from v_1 and ending at v_3 is shown in figure 1.5. The graph is connected, if and only if there is a path from any node to any other vertex in the

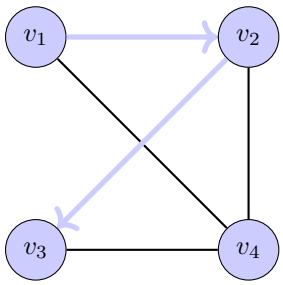


Figure 1.5: Example of a Path in a graph. Path (arrows) starting from node v_1 and ending at node v_3.

graph, as shown in figure 1.6. A connected graph may not be complete. A complete graph is a simple graph that contains exactly one edge between each pair of distinct nodes. The longest distance between

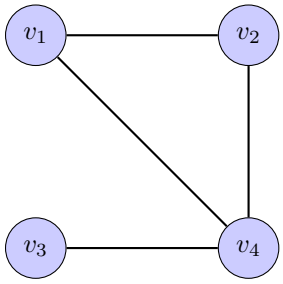


Figure 1.6: Example of a connected graph.

two nodes is called the *diameter* of the graph. The shortest path between two nodes is called *geodesic* and its length is called *distance* of the two nodes. A connected graph may not be (and often is not) complete. In this graph, there is a path between any given pair of vertices.

1.1.2 Graph signals

Graph signals are mapping $\mathbf{x} : V \to \mathbb{R}$ from the nodes of graph G into real (or complex) numbers. We consider the values of the signals on the set of the graph's nodes. Graph signals can be represented as vectors $\mathbf{x} \in \mathbb{R}^{n \times n}$. The signal $\mathbf{x}$ is a vector indexed by the graph's nodes

$$\mathbf{x} = (x(v_1), x(v_2), \ldots, x(v_n)) = (x_1, x_2, \ldots, x_n)$$

Notice that this assumes an indexing of the nodes, which coincides with the indexing used in the adjacency matrix.

1.1.3 Degree deviation in graphs

Using the degrees of a graph G, Nikiforov Nikiforov, 2006 introduced an irregularity measure referred to as *degree deviation* defined as

$$I_{\text{reg}}(G) = \sum_{i \in V} \left| d(v_i) - \frac{2m}{n} \right| \tag{1.1}$$

This statistic quantifies how much a graph deviates from being regular. Clearly, G is regular if and only if $I_{\text{reg}}(G) = 0$, si we say that G is close to regular if $I_{\text{reg}}(G) \to 0$.

1.2 Graph Spectra

In this section we present some known results, recalling basic facts about eigenvalues, eigenvectors, matrix diagonalization. We essentially focus on real symmetric matrices. Since most of standard matrices associated with graph are symmetric, in the following we review some of their important properties.

1.2.1 Spectral theorem of real symmetric matrices

Recall that a matrix $\mathbb{M}$ is symmetric if $\mathbb{M} = \mathbb{M}^T$. This implies that $\mathbb{M}$ is square, $\mathbb{M} \in \mathbb{R}^{n \times n}$.

Definition 2

An eigenvalue is a root of the characteristic polynomial associated with a matrix $\mathbb{M}$.

Definition 3

The set of all eigenvalues of $\mathbb{M}$ matrix is referred to as the *spectrum* of a graph represented by $\mathbb{M}$ matrix.

Definition 4

The trace of a matrix $\mathbb{M} = [m_{ij}]$ is the sum of its entries along the main diagonal. Trace of $\mathbb{M}$ is $Trace(\mathbb{M}) = \sum_{i=1}^{n} m_{ij}$.

Theorem 1.2.1. *An $n \times n$ symmetric matrix $\mathbb{M}$ has the following properties:*

1. *$\mathbb{M}$ has an eigendecomposition of the form*

$$\mathbb{M} = U \wedge U^T \tag{1.2}$$

where $\wedge \in \mathbb{R}^{n \times n}$ is a diagonal matrix and $U \in \mathbb{R}^{n \times n}$ is an orthogonal matrix. The diagonal entries of $\wedge$ are the eigenvalues of $\mathbb{M}$ and the columns of U are the corresponding eigenvectors:

$$\wedge = diag(\lambda_1, \lambda_2, \ldots, \lambda_n),\ U = [u_1 \mid u_2 \mid \ldots \mid u_n]$$

$$\mathbb{M}u_i = \lambda_i u_i$$

By definition an orthogonal matrix U satisfies $U^T = U^{-1}$, which means that the columns of U are orthogonal (any two of them are orthogonal and each has norm one).

The expression (1.2) of symmetric matrix in terms of eigenvalues and eigenvectors is referred to as spectral decomposition of $\mathbb{M}$. The set $(\lambda_1, \lambda_2, \ldots, \lambda_n)$ is called the spectrum of $\mathbb{M}$. Note that this set of eigenvalues includes multiplicities.

2. *Eigenvectors corresponding to distinct eigenvalues are necessarily orthogonal:*

$$\mathbb{M}u_1 = \lambda_1 u_1,\ \mathbb{M}u_2 = \lambda_2 u_2,\ \lambda_1 \neq \lambda_2 \implies u_1.u_2 = 0$$

Any two distinct eigenvectors from different eigenspaces are orthogonal.

3. *$\mathbb{M}$ is orthogonally diagonalizable.*

Remarks

- Note that if a matrix $\mathbb{M}$ is not symmetric, it might not have n eigenvectors. And even if it has n eigenvalues, their eigenvectors will not be orthogonal Spielman, 2004.
- The eigenvectors are not uniquely determined, although the eigenvalues are. Generally, the eigenvectors of a given eigenvalue are only determined up to an orthogonal transformation.

1.2.2 Energy of matrices

Let $\mathbb{M}$ be a $n \times n$ symmetric matrix with the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$. The energy of matrix $\mathbb{M}$ is given by Bravo et al., 2017:

$$E_{\mathbb{M}} = \sum_{i=1}^{n} \left| \lambda_i - \frac{Trace(\mathbb{M})}{n} \right| \tag{1.3}$$

As will be see later, if $\mathbb{M}$ is the adjacency matrix of graph, $E_{\mathbb{M}}$ is reduced to the energy of graph introduced by Gutman Gutman, 1978. As pointed out by Nikiforov, the energies of graphs are special cases of matrix norms (trace norms, or more generally Ky Fan or Schatten norms) Nikiforov, 2007.

1.2.3 Spectral moments

Let G be a graph without loops and multiple edges, with eigenvalues denoted by $\lambda_1, \lambda_2, \ldots, \lambda_n$ and are assumed to be labelled in a non-increasing way:

$$\lambda_1 \geq \lambda_2 \geq \ldots, \geq \lambda_n$$

For $k \in \mathbb{N}^*$, the k-th spectral moment of the graph G is defined as B. Zhou et al., 2007

$$M_k = M(G) = \sum_{i=1}^{n} |\lambda_i|^k \tag{1.4}$$

M_k is equal to the number of closed walk of of length k in G B. Zhou et al., 2007. Note the for $k = 1$, M_k is reduced to the energy of the graph G introduced by Gutman Gutman, 1978. We quote among the applications of the spectral moments, the analysis of complex networks V.M. Preciado and M.A. Rahimian, 2017,Q. Liu et al., 2017, V.M. Preciado and A. Jadbabaie, 2013.

1.3 Spectral graph representations

Recall that graphs are often represented via their adjacency or Laplacian matrices. In this section, these standard graph representations are presented, and both their properties and also their relationship with each other are reviewed.

1.3.1 Adjacency matrix

The most natural matrix to be associated with G is its adjacency matrix $\mathbf{A}(G)$, whose entries a_{ij} are given by

$$\mathbf{A} = [a_{ij}] = \begin{cases} 1 & \text{if } (v_i, v_j) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases}$$

Thus, for an unweighed graph, $\mathbf{A}$ is clearly a symmetric $(0,1)-$matrix. A possible notation for adjacency is $v_i \sim v_j$. The row sum of the adjacency matrix is the degrees of the vertices of G as we can see in figure 1.7. The adjacency matrix is real and symmetric.

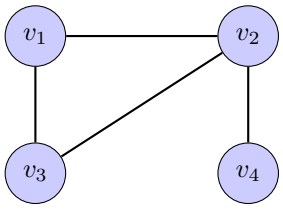


Figure 1.7: Example of a graph G with its adjacency matrix $\mathbf{A}$.

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

1.3.2 Incidence matrix

In the case of an undirected graph, the incidence matrix of a graph G is an $(n \times m)$ matrix $\mathbf{C} = \{c_{ij}\}$ where each row corresponds to a vertex v_i and each column corresponds to an edge such that if e_k is

an edge between v_i and v_j then all elements of the column $'k'$ are $'0'$ except for $c_{ik} = c_{jk} = 1$. The expression of $\mathbf{C}$ is given by :

$$\mathbf{C} = [c_{ij}] = \begin{cases} 1 & \text{if } (e_{ij}) \text{ exists} \\ 0 & \text{otherwise} \end{cases}$$

For such a matrix, each column representing an edge contains two non-zero entries, the rest being zero. The unit entries in a column identify the nodes of the edge between which it is connected. This is illustrated in figure 1.8. Most graphs have more edges than vertices. Thus the incidence matrix is usually bigger than the adjacency matrix requiring especially a larger storage space hence a less use of this incidence matrix.

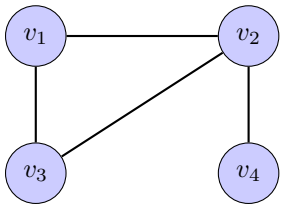


Figure 1.8: Example of a graph G and its incidence matrix $\mathbf{C}$.

$$\mathbf{C} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

1.3.3 Degree matrix

The number of neighbors of a node v_i is called the *degree* of v_i and is denoted by $d(v_i)$ where

$$d(v_i) = \sum_{j \in \rangle} a_{ij} \tag{1.5}$$

The diagonal matrix $\mathbf{D}$ contains information about the degree of each vertex, v_i, that is the number of edges, $d(v_i)$ attached to this vertex. Let G be an undirected graph. The degree matrix of G is given by :

$$\mathbf{D} = \begin{pmatrix} d(v_1) & 0 & \cdots & 0 \\ 0 & d(v_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d(v_n) \end{pmatrix} = \text{diag}(d(v_1), d(v_2), \ldots, d(v_n))$$

The degree matrix of graph G, reported in figure 1.7, is given by

$$\mathbf{D} = \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

In the case of a weighted graph, the matrix $\mathbf{A}$ is called *weighted adjacency matrix* and the entries of $\mathbf{D}$ are given now by :

$$d(v_i) = \sum_{j=1}^{n} w_{ij} \tag{1.6}$$

Thus, the weighted matrix, $\mathbf{W} = [w_{ij}] \in \mathbb{R}^{n \times n}$, can be written as

$$\mathbf{W} = \mathbf{D}^{-1}\mathbf{A} \tag{1.7}$$

1.3.4 Laplacian matrices and Signless matrix

1.3.4.1 Difference operator

The Laplacian acts as difference operator on graphs signals. Consider a graph signal $\mathbf{x}$ on G and define a new signal $\mathbf{y} = \mathbf{L}\mathbf{x}$ where each element y_i is computed as

$$y_i = [\mathbf{L}\mathbf{x}]_i = \sum_{j \in \mathcal{N}(i)} w_{ij}(x_i - x_j) \tag{1.8}$$

The output y_i measures the difference between the value of the signal $\mathbf{x}$ at note i and at its neighborhood.

1.3.4.2 Laplacian quadratic form

The most natural quadratic form associated with G is defined in terms of its Laplacian matrix $\mathbf{L}$.

Lemma 1.3.1. *Let* $\mathbf{L}$ *be the Laplacian matrix of dimensions* $n \times n$ *and let a vector* $\mathbf{x} \in \mathbb{R}^n$. *Then*

$$\mathbf{x}^T\mathbf{L}\mathbf{x} = \frac{1}{2} \sum_{(i,j) \in \mathcal{E}} w_{ij}(x(i) - x(j))^2 \geq 0, \forall \mathbf{x} \tag{1.9}$$

This form measures the smoothness of the function $\mathbf{x}$. *It will be small if the function* $\mathbf{x}$ *does not jump too much over any edge.* $\mathbf{x}^T\mathbf{L}\mathbf{x}$ *quantifies the local variation of signal* $\mathbf{x}$. *The matrix defining this form is the Laplacian matrix of the graph* G *defined as*

$$\mathbf{L} = \mathbf{D} - \mathbf{A} = [l_{ij}] = \begin{cases} -1 & \text{if } i \neq j \text{ and } v_i \sim v_j \\ d(v_i) & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

where $\mathbf{A}$ *is the adjacency matrix and* $\mathbf{D}$ *is the degree matrix.*

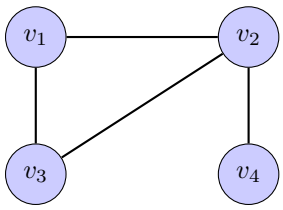


Figure 1.9: Example of a graph G and its laplacian matrix **L**.

$$\mathbf{L} = \begin{pmatrix} 2 & -1 & -1 & 0 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 2 & 0 \\ 0 & -1 & 0 & 1 \end{pmatrix}$$

Lemma 1.3.2. *Let* **L** *be the Laplacian matrix of* G*. Then the eigenvector coressponding to the eigenvalue zero is a ones vector, that is* $[1, 1, \ldots, 1]^T$*.*

Corollary 1.3.1. *The multiplicity of the zero* "0" *eigenvalue of the Laplacian matrix* **L** *corresponding to a graph* G *equals the number of its connected components.*

The Laplacian has at least one zero eigenvalue, and the number of such eigenvalues is equal to the number of disjoint parts in the graph.

When studying, for example, random walks on a graph G, it often useful to normalize the Laplacian **L** by its degrees. The normalized version of Laplacian matrix, $\mathbf{L}_N$, of G is defined by

$$\mathbf{L}_N = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$$

The relationships with the normalized Laplacian matrix show matrices **A**, **L**, **I** the identity matrix and $\mathbf{D}^{-1/2}$ a diagonal matrix with $\mathbf{D}^{-1/2}(i,i) = \dfrac{-1}{\sqrt{d(v_i)}}$.

$$\mathbf{L}_N = [l_{ij}]_N = \begin{cases} 1 & \text{if } i = j\,; d(v_i) \neq 0 \\ \dfrac{-1}{\sqrt{d(v_i).d(v_j)}} & if\ v_i \sim v_j \\ 0 & \text{otherwise} \end{cases}$$

As for the Laplacian of the graph, this matrix is positive semi-definite and so has all eigenvalues greater than or equal to zero. The normalisation factor means that the largest eigenvalue less than or equal to 2, with equality only when G is bipartite. Again, the matrix has at least one zero eigenvalue. Hence all the eigenvalues are in the range $0 \leq \lambda \leq 2$.

The normalized Laplacian is used in various graph analysis tasks, such as graph clustering and random walks on graphs. The eigenvalues of the normalized Laplacian are in "normalized" form and the spectra

of the normalized Laplacian relate well to other graph invariants for general graphs in a way that the other definitions fail to do Spielman, 2011.

1.3.4.3 Graph requencies

It has been shown that the eignvectors of the Laplacian matrix provides a harmonic analysis of graph signals which in turn provides a Fourier-Like interpretation R.K. Fan Chung, 1996b,Pitas, 2016. The Lapalcain eigenvalues can be interpreted as frequencies and the eigenvectors act as the natural vibration modes of the graph G, and the corresponding eignevalues as the associated graph-frequencies R.K. Fan Chung, 1996b. The frequncy interpretation of eigenvectors can be viewed in terms of number of zero-crossings (pair of connected nodes with different signs) of an eignevector of graph G. For any finite graph G, the eigenvectors with large eigenvalues havemore zero-crossingns (hence high frequency) than eigenvectors with small eigenvalues Pitas, 2016. Consider the case of the normalized laplacian matrix of G. Thus, $\lambda_i \in [0, 2]$. An eigenvector u_λ is either considered to be a "*low-pass*" eigenvector if $\lambda_i \in [0, 1]$ or "*high-pass*" eigenvector $\lambda_i \in]1, 2]$. The graph Fourier transform, denoted as $\hat{\mathbf{x}}$, is defined in D.K. Hammond et al., 2011 as the projection of $\mathbf{x}$ on the graph G onto the eigenvactors of G:

$$\mathbf{x} =< u_\lambda, \mathbf{x} >= \sum_{i=1}^{n} x(i) u_i \tag{1.10}$$

1. The eigenvectors associated with large eigenvalues (high frequency) vary rapidly
   ⇒ Dissimilar values on vertices connected by edges.

2. The eigenvectors associated with small eigenvalues (low frequency) vary slowly
   ⇒ Similar values on vertices connected by edges.

1.3.5 Grounded Laplacian matrix

A recent variant of the Laplacian matrix is the grounded Laplacian matrix, obtained by removing certain rows and columns from the Laplacian. This matrix forms the basis for the classical Matrix Tree Theorem (characterizing the number of spanning trees in the graph), and also plays a fundamental role in the study of continuous-time diffusion dynamics where the states of some of the nodes in the network are fixed at certain values P. Barooah and J.P. Hespanha, 2006. The eigenvalues of the grounded Laplacian characterize the variance in the equilibrium values for noisy instances of such dynamics, and determine the rate of convergence to steady state M.Pirani, 2014.

1.3.6 Signless matrix

Another variant of the Laplacian, is the *signless* matrix defined by

$$|\mathbf{L}| = \mathbf{D} + \mathbf{A}$$

This matrix seems to have good properties in the sense that it produces fewer cospectral (two graphs are said to be cospectral if they have the same eigenvalues with respect to the matrix representation being used) graphs than the Laplacian as we mention in Chapter 4.

Just like the adjacency matrix, the Laplacian matrix $\mathbf{L}$ has been widely studied in the spectral graph theory which attempts to rely the graph's structure and the eigenvalues of the matrix. That's why another properties about this matrix will be given in the following.

1.3.7 Properties of the eigen-spectrum of the Laplacian matrix

The Laplacian eigenvalues obey the well-known relations Merris, 1994:

$$\begin{aligned} \sum_{i=1}^{n} \lambda_i &= 2m \\ \sum_{i=1}^{n} \lambda_i^2 &= 2m + \sum_{i=1}^{n} d_i^2 \end{aligned} \tag{1.11}$$

1.3.8 Laplacian energy of graph

The Laplacian is a positive semi-definite matrix with a trivial eigenvalue 0 and the associated eigenvector of all ones $\mathbf{1}$. We denote the eigenvalues of matrix $\mathbf{L}$ by $\lambda_1, \lambda_2, \ldots, \lambda_n$. It is easy to check that the trace of $\mathbf{L}$ is $2m$:

$$\sum_{i=1}^{n} \lambda_i = 2m \tag{1.12}$$

For $m > 0$ at least one eigenvalue has value greater than the average degree $2m/n$. The Laplacian energy of graph G, $LE(G)$, is defined by I. Gutman and B. Zhou, 2006:

$$LE(G) = E_{\mathbf{L}}(G) = \sum_{i=1}^{n} \left| \lambda_i - \frac{2m}{n} \right| \tag{1.13}$$

The $LE(G)$ value is a broad measure of complexity. Relation (1.15) raises the question which graphs on n nodes maximizes $LE(G)$ C. Helmberg and V. Trevisan, 2017.

An additional Laplacian-spectrum-based graph invariant was put forward by Liu and Liu J. Liu and B. Liu, 2008:

$$LEL(G) = \sum_{i=1}^{n-1} \sqrt{\lambda_i} \tag{1.14}$$

and was named *Laplacian-energy-like invariant*. The motivation for introducing $LEL(G)$ was in its analogy to the earlier studied graph energy X. Li et al., 2012 and $LE(G)$ I. Gutman and B. Zhou, 2006. The $LEL(G)$ has many analogous properties as graph energy.

1.3.9 Signless Laplacian energy of graph

Let $\mu_1, \mu_2, \ldots, \mu_n$ be the eigenvalues of the signless Laplacian. The signless Laplacian energy of the graph G is defined as

$$LE^{+}(G) = \sum_{i=1}^{n} \left| \mu_i - \frac{2m}{n} \right| \tag{1.15}$$

1.3.10 Electrical network and Kirchhoff index

We can regard a connected graph G as an electrical network, where each edge can be viewed as a unit resistor (1 Ohm). The distance between two nodes v_i and v_j is defined as the length (=number of edges) of a shortest path connects v_i and v_j. Klein and Randic D.J. Klein and M. Randic, 1993 conceived the *resistance distance* defined in terms of electrical resistance in a network corresponding to the considered graph G, in which the resistance between any two adjacent nodes is 1 Ohm. An important invariant of the electrical network is its resistance distance. Given any two vertices v_i and v_j, the effective resistance between i and j is the voltage of a battery which, when connected to the two vertices, causes a current of 1 Ampere to flow. For a pair of vertices v_i and v_j, the resistance distance is noted $R(i, j)$. The sum of resistance distance between all pair of vertices of G, is called its *Kirchhoff index*, denoted by $KF(G)$, namely

$$KF(G) = \sum_{1=i \leq j \leq n} R(i, j) \tag{1.16}$$

It has been shown that the Kirchhoff index of G can be determined in terms of eigenvalues of L I. Gutman and B. Mohar, 1996.

Theorem 1.3.1. *For a connected graph G with order $n \geq 2$, the Kirchhoff index of G is given by*

$$KF(G) = n \sum_{i=2}^{n} \frac{1}{\lambda_i}, \tag{1.17}$$

where $0 = \lambda_1 \leq \lambda_2, \ldots, \lambda_n$ are eigenvalues of **L**.

The resistance distance and Kirchhoff index have received considerable attention in the literature, since they have useful connection with various fields such as the random walks A. Ghosh et al., 2008.

1.3.11 Distance matrix

Another matrix to mention is the *distance matrix* $\mathbf{D}^G$ corresponding to a square matrix $\mathbf{D}^G = [d_{(v_i,v_j)}]_{i,j}$, where the entry in the (i^{th} row, j^{th} column) is the distance or *the length of a shortest path* between the v_i and v_j vertex.

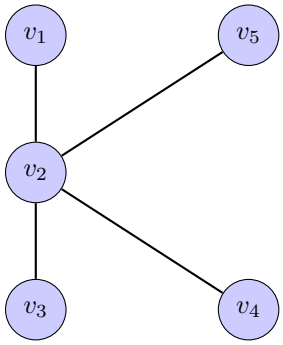


Figure 1.10: Example of a distance matrix $\mathbf{D}^G$.

$$\mathbf{D^G} = \begin{pmatrix} 0 & 1 & 2 & 2 & 2 \\ 1 & 0 & 1 & 1 & 1 \\ 2 & 1 & 0 & 2 & 2 \\ 2 & 1 & 2 & 0 & 2 \\ 2 & 1 & 2 & 2 & 0 \end{pmatrix}$$

1.3.12 Transition matrix

Recall that a random walk is a process that begins at some vertex then moves to random neighbor of that vertex. The transition matrix of the random walk on graph G is defined as the $n \times n$ matrix $\mathbf{P} = [p_{ij}]$ in which

$$p_{ij} = \frac{a_{ij}}{d(v_i)} \tag{1.18}$$

So

$$\mathbf{P} = \mathbf{D}^{-1}\mathbf{A} \tag{1.19}$$

Matrix $\mathbf{P}$ is also called *walk* matrix because it encodes the dynamics of a random walk on G. It is used to study the evolution of probability distribution of random walk. This matrix is often asymmetric, but one can define the normalized adjacency matrix of G as

$$\mathbf{Q} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2} \tag{1.20}$$

which is similar to

$$\mathbf{P} = \mathbf{D}^{-1/2}\mathbf{Q}\mathbf{D}^{-1/2} \tag{1.21}$$

and thus has the same eigenvalues as $\mathbf{P}$.

1.3.13 Properties of the eigen-spectrum of the adjacency matrix

We have previously introduced the adjacency matrix $\mathbf{A}(G)$ of a graph $G = (V, \mathcal{E})$ with n vertices and m edges. By the spectral theorem, it has an orthonormal basis of eigenvectors with real eigenvalues, $\lambda_1 \geq \lambda_2 \geq \ldots \lambda_n$. The eigenvalue λ_1 is called the *spectral radius* of G. Moreover, this radius is at least the average vertex degree in G that is

$$\lambda_1 \geq \frac{2m}{n} \tag{1.22}$$

with equality holding if and only if G is isomorphic to a regular graph K.CH.Das and S.A.Mojallal, 2016. Some well known properties of graph eigenvalues are

$$\sum_{i=1}^{n} \lambda_i = 0; \quad \sum_{i=1}^{n} \lambda_i^2 = 2m; \quad \det\mathbf{A} = \prod_{i=1}^{n} \lambda_i \tag{1.23}$$

The graph G is said to be *singular* if at least one of its eigenvalues is equal to zero. For singular graphs, evidently, $\det\mathbf{A} = 0$. A graph is *non-singular* if $\lambda_i \neq 0, \ \forall i \in \{1, 2, \ldots, n\}$. Then, $\det\mathbf{A} \neq 0$.

1.3.13.1 The largest eigenvalue, λ_{max}

Let d_{max} be the maximum degree of a vertex in G, and let d_{ave} be the average degree of a vertex in G. The largest eigenvalue λ_{max} of $\mathbf{A}$ verifies:

$$d_{ave} \leq \lambda_1 \leq \lambda_{max} \tag{1.24}$$

We can strengthen the lower bound by proving that λ_1 is at least the average degree of G.

1.3.13.2 The eigenvalue gap

The gap between the first and the second eigenvalues is an important parameter for graph characterization. Perron-Frobenuis theorem states that if G is a connected graph then λ_{max} of $\mathbf{A}$ has multiplicity 1. We can expect that the gap between λ_{max} and the nearest eigenvalue is related to some kind of connectivity measure of the graph G.

1.3.13.3 Energy of graph

A useful graph-invariant of G, which we will study later in the Chapter 3, is the sum of the absolute values of the eigenvalues of $\mathbf{A}$:

$$E_{\mathbf{A}}(G) = \sum_{i=1}^{n} \mid \lambda_i \mid \tag{1.25}$$

This quantity, $E_{\mathbf{A}}(G)$, called *energy* of G was first defined in 1978 by Gutman Gutman, 1978. Nikiforov first recognized that the energy of graph is equal to the sum of the singular values of its adjacency matrix Nikiforov, 2007. This spectrum-based graph invariant has been much studied in both chemical and mathematical literature K.CH.Das and I.Gutman, 2016. What nowadays is referred to as graph energy, given by relation (1.25), is closely related to the total π-electron energy calculated within the Ḧuckel molecular orbital approximation B. J. McClelland, 1971.

1.3.13.4 Real-values signal on graphs

When the adjacency operator is applied to a signal $\mathbf{x}$, the resulting value at a vertex v is the sum of the values of the signal $\mathbf{x}$ over all neighbors of v:

$$\mathbf{y} = \mathbf{A}\mathbf{x}; \quad y(i) = \sum_{i \sim j} x(i) \tag{1.26}$$

The quadratic form associated to $\mathbf{A}$ is given by

$$\mathbf{x}^T \mathbf{A}\mathbf{x} = \sum_{e_{ij}} x(i)x(j) \tag{1.27}$$

Remarks

There is debate, in the literature, as to whether the eigenvalues of the adjacency matrix provide information about the graph properties. For example, Spielman argues that, even the adjacency matrix is the most natural matrix to associate with graph, it is least useful Spielman, 2004. Eigenvalues and eigenvectors are most meaningful when used to understand a natural operator or a natural quadratic form. The adjacency matrix provides neither. The same observation was made by Lau which points out that it is not clear that the eigenvalues should any information about the graph properties Lau, 2015. But they do, and interesting information, using for example $E(G)$, can be obtained from them H.A. Bay-Ahmed et al., 2018.

1.4 Eigenvalues and optimization

One the reason that the eigenvalues of matrices have meaning is that they arise as the solution to natural optimization problems Spielman, 2004. More precisely, these eigenvalues are useful because they constitute the optimal solution of a very basic quadratic optimization problem. The main tool in relating eigenvalues and eigenvectors to optimization problem is the *Rayleigh quotient*.

Definition 4

The Rayleigh quotient of vector $\mathbf{x}$ with respect to a symmetric matrix $\mathbb{M} = [m_{ij}]$ is the ratio

$$\frac{\mathbf{x}^T\mathbb{M}\mathbf{x}}{\mathbf{x}^T\mathbf{x}} = \frac{\sum_{i,j} m_{ij}x_ix_j}{\sum_i x_i^2} \tag{1.28}$$

Note that if ψ is an eigenvector of $\mathbb{M}$ of eigenvalue λ, then the Rayleigh quotient is reduced to

$$\frac{\psi^T\mathbb{M}\psi}{\psi^T\psi} = \frac{\psi^T\lambda\psi}{\psi^T\psi} = \frac{\lambda\psi^T\psi}{\psi^T\psi} = \lambda \tag{1.29}$$

Theorem 1.4.1. *Let $\mathbb{M}$ be a $n \times n$ symmetric matrix with eigenvalues $\lambda_1 \geq \lambda_2 \geq, \ldots, \geq \lambda_n$ and let $\mathbf{x}$ be a non-zero vector that maximizes the Rayleigh quotient with respect to $\mathbb{M}$:*

$$\frac{\mathbf{x}^T\mathbb{M}\mathbf{x}}{\mathbf{x}^T\mathbf{x}} \tag{1.30}$$

Then, $\mathbf{x}$ is an eigenvector equal to the Rayleigh quotient. Moreover, this eigenvalue is the largest eigenvalue of $\mathbb{M}$:

$$\lambda_1 = \max_{\mathbf{x}} \frac{\mathbf{x}^T\mathbb{M}\mathbf{x}}{\mathbf{x}^T\mathbf{x}} \tag{1.31}$$

Theorem 1.4.2. *Let $\mathbb{M}$ be a $n \times n$ symmetric matrix with eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ and with corresponding eigenvectors $\psi_1, \psi_2, \ldots, \psi_n$. Then the sets $\{\psi_i\}_{i=1}^n$ and $\{\lambda_i\}_{i=1}^n$ are retrieved using the Rayleigh quotient as follows:*

$$\begin{aligned} \lambda_i &= \min_{\mathbf{x}\perp\psi_1,\ldots,\psi_{i-1}} \frac{\mathbf{x}^T\mathbb{M}\mathbf{x}}{\mathbf{x}^T\mathbf{x}} \\ \psi_i &= \arg\min_{\mathbf{x}\perp\psi_1,\ldots,\psi_{i-1}} \frac{\mathbf{x}^T\mathbb{M}\mathbf{x}}{\mathbf{x}^T\mathbf{x}} \end{aligned} \tag{1.32}$$

1.4.1 Courant-Fischer Theorem

The Courant-Fischer theorem provides a more powerful characterization of eigenvalues as solutions to optimization problems. This theorem is useful for proving upper bounds on the largest eigenvalue of matrix Spielman, 2004. The Courant-Fischer characterization of the eigenvalues of a symmetric matrix $\mathbb{M}$ in terms of the maximizers and minimizers of the Rayleigh quotient plays an important role in spectral graph theory Spielman, 2011.

Theorem 1.4.3. *Let $\mathbb{M}$ be a $n \times n$ symmetric matrix with eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$. Then*

$$\lambda_k = \min_{\substack{S\subseteq\mathbb{R}^n \\ dim(S)=k}} \max_{\substack{\mathbf{x}\in S \\ \mathbf{x}\neq 0}} \frac{\mathbf{x}^T\mathbb{M}\mathbf{x}}{\mathbf{x}^T\mathbf{x}} = \max_{\substack{T\subseteq\mathbb{R}^n \\ dim(T)=n-k+1}} \min_{\substack{\mathbf{x}\in T \\ \mathbf{x}\neq 0}} \frac{\mathbf{x}^T\mathbb{M}\mathbf{x}}{\mathbf{x}^T\mathbf{x}} \tag{1.33}$$

For example, the Courant-Fischer theorem tells us that

$$\lambda_1 = \max_{\mathbf{x}\in\mathbb{R}^n} \frac{\mathbf{x}^T\mathbb{M}\mathbf{x}}{\mathbf{x}^T\mathbf{x}} \text{ and } \lambda_n = \min_{\mathbf{x}\in\mathbb{R}^n} \frac{\mathbf{x}^T\mathbb{M}\mathbf{x}}{\mathbf{x}^T\mathbf{x}} \tag{1.34}$$

The minimum in the first expression of relation (1.33) is taken over all subspaces of dimension k, and the maximum in the second expression is taken over all subspaces of dimension (n-k+1). For example, consider the case $k = 1$. Thus, S is the span of ψ_1 and T is all of $\mathbb{R}^n$. For $k \neq 1$, the optima will be achieved when S is the span of $\psi_1, \psi_2, \ldots, \psi_k$ and when T is the span of $\psi_k, \psi_{k+1}, \ldots, \psi_n$ Spielman, 2011,R.A. Horn and C.R. Johnson, 1985.

Theorem 1.4.4 (Courant-Fischer formula)**.** *Let $\mathbb{M}$ be a $n \times n$ symmetric matrix with eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ and with corresponding eigenvectors $\psi_1, \psi_2, \ldots, \psi_n$.*

$$\lambda_1 = \min_{||\mathbf{x}||=1} \mathbf{x}^T \mathbb{M} \mathbf{x} = \min_{\mathbf{x} \neq 0} \frac{\mathbf{x}^T \mathbb{M} \mathbf{x}}{\mathbf{x}^T \mathbf{x}} \tag{1.35}$$

$$\lambda_2 = \min_{\substack{||\mathbf{x}||=1 \\ \mathbf{x} \perp \psi_1}} \frac{\mathbf{x}^T \mathbb{M} \mathbf{x}}{\mathbf{x}^T \mathbf{x}} = \min_{\substack{\mathbf{x} \neq 0 \\ \mathbf{x} \perp \psi_1}} \frac{\mathbf{x}^T \mathbb{M} \mathbf{x}}{\mathbf{x}^T \mathbf{x}} \tag{1.36}$$

$$\lambda_n = \max_{||\mathbf{x}||=1} \mathbf{x}^T \mathbb{M} \mathbf{x} = \max_{\mathbf{x} \neq 0} \frac{\mathbf{x}^T \mathbb{M} \mathbf{x}}{\mathbf{x}^T \mathbf{x}} \tag{1.37}$$

Corollary 1.4.1. *Let $G = (V, \mathcal{E})$ be a graph with laplacian matrix* $\mathbf{L}$

$$\lambda_1 = 0, \psi_1 = [\mathbf{1}, \mathbf{1}, .., \mathbf{1}]^{\mathbf{T}} \tag{1.38}$$

$$\lambda_2 = \min_{\substack{\mathbf{x} \neq 0 \\ \mathbf{x} \perp \psi_1}} \frac{\mathbf{x}^T L \mathbf{x}}{\mathbf{x}^T \mathbf{x}} = \min_{\substack{\mathbf{x} \neq 0 \\ \sum_i x_i = 0}} \frac{\sum_{(i,j) \in \mathcal{E}} (x_i - x_j)^2}{\sum_{i \in V} x_i^2} \tag{1.39}$$

$$\lambda_{max} = \max_{\mathbf{x} \neq 0} \mathbf{x}^T L \mathbf{x} = \max_{\mathbf{x} \neq 0} \frac{\sum_{(i,j) \in \mathcal{E}} (x_i - x_j)^2}{\sum_{i \in V} x_i^2} \tag{1.40}$$

This corollary states that the Rayleigh quotient is useful for bounding graph spectra. In order to get an upper bound on λ_2 we need only produce a vector $\mathbf{x}$ with small Rayleigh quotient. To get a lower bound on λ_{max} we need only to find a vector $\mathbf{x}$ with large Rayleigh quotient.

1.5 Fiedler's theory of spectral graph partitioning

With the advent of larger instances in applications such as social networks, or road networks, graph partitioning becomes highly important, multifaceted, and challenging. There are varying methods of accomplishing this. One method, proposed by Fiedler, is called spectral graph partitioning Fiedler, 1973. Given a connected graph G, spectral graph partitioning is a method of partitioning G into two subgraphs in such a way that the subgraphs have a nearly equal number of vertices (as close to equal as is possible) while also minimizing the number of edges between the two subgraphs.

1.5.1 Fiedler value

The multiplicity of zero in the spectrum of the Laplacian of graph is equal to the number of connected components in the graph. If we let the eigenvalues of the Laplacian be

$$0 = \lambda_1 \leq \lambda_2 \leq, \ldots, \leq \lambda_n$$

then $\lambda_2 = 0$ if and only if the graph is disconnected. Fiedler proved that the further λ_2 is from zero, the better connected the graph is Fiedler, 1973. This second smallest eigenvalue of the Laplacian is called the *Fiedler value*. This value can be used to determine whether an undirected graph or network is connected or not. Therefore, λ_2 is called the *algebraic connectivity* of the graph or the network Nikiforov, 2013. However, the multiplicity of the Fiedler eigenvalue depends on the graph's structure and it is difficult to analyse.

If a graph G is disconnected, the we partition it into two graphs G_1 and G_2 with no edges between them, and then write

$$\mathbf{L} = \begin{pmatrix} \mathbf{L}_1 & 0 \\ 0 & \mathbf{L}_2 \end{pmatrix}$$

As the eigenvalues of $\mathbf{L}$ are the union, with multiplicity, of the eigenvalues of $\mathbf{L}_1$ and $\mathbf{L}_2$, we see that $\mathbf{L}$ inherits a zero eigenvalue from each.

1.5.2 Fiedler vector

The Fiedler vector of a connected undirected graph is the eigenvector corresponding to the second smallest eigenvalue of the Laplacian matrix $\mathbf{L}$ of the graph. After introducing the algebraic connectivity, Fiedler noticed that the eigenvector associated to λ_2 indices partitions of the vertices of graph G that are natural connected clusters Fiedler, 1975,M. Fiedler and V. Nikiforov, 2009, D.A. Spielman and S.H. Teng, 2007. Note that the multiplicity of λ_2 may be greater than one, in which case there is more than a single Fiedler vector.

1.5.3 Connectivity and Spectral Clustering

In spectral graph theory, $\mathbf{L}$ matrix is often used for structural properties study of graphs. This is essentially motivated by its algebraic characteristics as the non-negativeness of its eigenvalue spectrum, or its interesting quadratic form written as $\mathbf{y^T L y} = \sum_{u \sim v} (\mathbf{y}(u) - \mathbf{y}(v))^2$, with $\mathbf{y}$ is a function that assigns to each vertex v of the graph a real value $\mathbf{y}(v)$, and $\sum_{u \sim v}$ denotes the sum over all unordered pairs (u, v) for which u and v are adjacent. This quadratic form is useful for getting a variational characterizations of the eigenvalues in terms of *Rayleigh quotient* R.K. Fan Chung, 1996a. The eigenpairs (λ_k, ψ_k) of $\mathbf{L}$

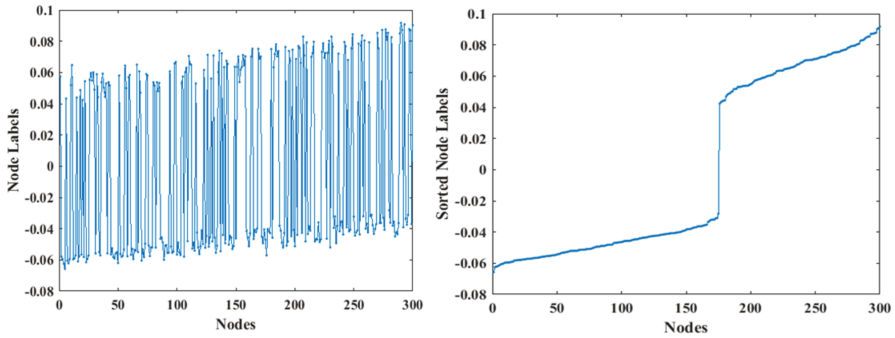


Figure 1.11: Laplacian's second eigenvector (Fiedler vector) of a connected random graph with 300 nodes. In the left: labels associated to nodes before sorting. In the right: After sorting.

can be formulated as convex optimization problems:

$$\lambda_k = \min_{\substack{\mathbf{y}\perp\mathbf{f_1},\ldots,\mathbf{f_{k-1}} \\ \mathbf{y}\neq\mathbf{0}}} \frac{\mathbf{y^T L y}}{\mathbf{y^T y}} = \min_{\substack{\mathbf{y}\perp\mathbf{f_1},\ldots,\mathbf{f_{k-1}} \\ \mathbf{y}\neq\mathbf{0}}} \frac{\sum\limits_{u\sim v}(\mathbf{y}(u)-\mathbf{y}(v))^2}{\sum\limits_{v}\mathbf{y}(v)^2}, \tag{1.41}$$

$$\psi_k = \arg\min_{\substack{\mathbf{y}\perp\mathbf{f_1},\ldots,\mathbf{f_{k-1}} \\ \mathbf{y}\neq\mathbf{0}}} \frac{\mathbf{y^T L y}}{\mathbf{y^T y}} = \arg\min_{\substack{\mathbf{y}\perp\mathbf{f_1},\ldots,\mathbf{f_{k-1}} \\ \mathbf{y}\neq\mathbf{0}}} \frac{\sum\limits_{u\sim v}(\mathbf{y}(u)-\mathbf{y}(v))^2}{\sum\limits_{v}\mathbf{y}(v)^2}. \tag{1.42}$$

It can be easily deduced from equation (1.41) that 0 is an eigenvalue of $\mathbf{L}$ corresponding to the constant eigenvector $\psi_1 = 1$. The second eigenvector must be orthonormal to the constant eigenvector ($\langle\psi_2, \mathbf{1}\rangle = \sum\limits_{v}\psi_2(v) = 0$). Therefore, a null second eigenvalue implies that the graph is composed of two disconnected components. This leads to the fundamental result of graph algebra which states that the multiplicity of zero eigenvalue corresponds to the number of the disconnected clusters in the graph. In connected graphs, the second eigenvalue is non-zero and corresponds to the minimal cost of the connections which can be lost in case of segmentation of the graph in two connected sub-graphs. This quantity has been defined by Fiedler as the algebraic connectivity of the graph, the corresponding second eigenvector is called Fiedler vector Fiedler, 1973. In the following example, we consider an undirected and unweighed random graph that contains implicitly two communities. We observe that the $\mathbf{A}$ matrix hides almost all indications about the existence of such communities or at least they remain difficult to detect (Fig. 1.12(a)). Fig. 1.11(b) shows that the Fiedler vector of $\mathbf{L}$ matrix associates to the nodes values of opposite signs in order to cluster them into subgroups. Using those labels, we reordered the raws and columns of $\mathbf{A}$ matrix in a way that reveals more structural information about the graphs (Fig. 1.12(b)). This example illustrates the interest of $\mathbf{L}$ matrix for recovering homogeneous clusters of nodes.

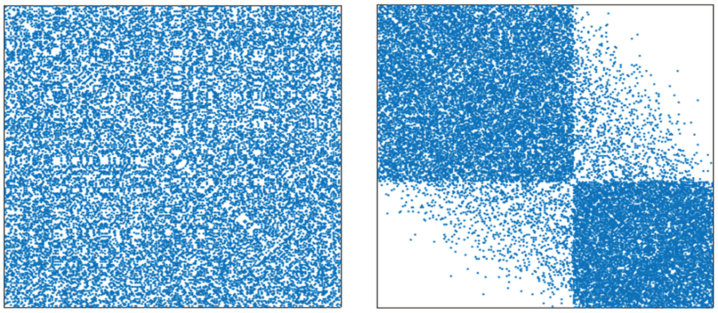


Figure 1.12: Adjacency matrix of the connected random graph with 300 nodes. In the left: the original **A** matrix. In the right: the reordered **A** matrix using the Fiedler labels order (From **L** matrix).

1.6 Low rank approximation and eigenvectors

The problem of low-rank approximation of a given matrix, is to approximate this matrix by a matrix of low rank so that, for example, the Frobenius norm of the error is minimized. The goal of this approximation is to obtain more compact representations of the data with limited loss of information N.K. Kumar and J. Schneider, 2016. One explanation for the utility of the eigenvectors of extreme eigenvalues of matrices is that they provide low-rank approximations of a matrix.

Consider a $n \times n$ symmetric matrix representation $\mathbb{M}$ of graph G with eigenvalues $\lambda_1 \leq \lambda_2 \leq, \ldots, \leq \lambda_n$. The eigenvalue decomposition of $\mathbb{M}$ is given by

$$\mathbb{M} = U\Lambda U^{-1} \tag{1.43}$$

where Λ is a diagonal matrix whose i-th diagonal element is the i-th eigenvalue of $\mathbb{M}$. Defining $U = [u_1, u_2, \ldots, u_n]$ and $(U^{-1})^T = [\tilde{u}_1, \tilde{u}_2, \ldots, \tilde{u}_n]$, where u_i and $\tilde{u}_i$ are $n \times 1$ column vectors of U and $(U^{-1})^T$, respectively, one can show that

$$\begin{aligned} \mathbb{M} = U\lambda_i U^{-1} &= \sum_{i=1}^{n} \lambda_i u_i u_i^T \\ &= \sum_{i=1}^{n} \lambda_i B_i \end{aligned} \tag{1.44}$$

The rank one matrix $B_i = u_i u_i^T$ is called the i-th *eigengraph* and u_i the i-th frequency component of $\mathbb{M}$. Note that the eigengraph B_i is a special graph such that $\mathbb{M}B_i = \lambda_i B_i$. If none of the elements of u_i and $\tilde{u}_i$ are zero, the corresponding eigengraph is a complete graph, meaning that all vertices are connected to each other. By defining the B_i as the i-th frequency component of the graph G, the value λ_i in $\mathbb{M} = \sum_{i=1}^{n} \lambda_i B_i$ can be interpreted as the significance of the corresponding component A. Gavili and X.P. Zhang, 2017.

We can measure how well matrix $\mathbf{J}$ approximates a matrix $\mathbb{M}$ by either the operator norm $\|\mathbb{M}-\mathbf{J}\|$, the Frobenius norm $\|\mathbb{M}-\mathbf{J}\|_{\mathrm{F}}$ or Nuclear norm $\|\mathbb{M}-\mathbf{J}\|_*$, where we recall

$$\begin{aligned}\|\mathbf{Z}\| &\stackrel{\text{def}}{=} \max_{\mathbf{x}} \frac{\|\mathbf{Z}\mathbf{x}\|}{\|\mathbf{x}\|}\\ \|\mathbf{Z}\|_{\mathrm{F}} &\stackrel{\text{def}}{=} \sqrt{\sum_{i,j} z_{ij}}\\ \|\mathbf{Z}\|_* &\stackrel{\text{def}}{=} \mathrm{Trace}(\mathbf{Z})\end{aligned}$$

Using the Courant-Fisher theorem, one can show that every k, the best approximation of $\mathbb{M}$ by a rank-k matrix is given by summing the terms $\lambda_i B_i$ over the k values of i

$$\tilde{\mathbb{M}}_k = \sum_{i=k}^{n} \lambda_i B_i \tag{1.45}$$

The measure of the quality of approximation depends on the distortion measure used (Frobenius norm,...). When $\|\mathbb{M}-\mathbf{J}\|$ is small, it explains why the eigenvectors of the largest k eigenvalues of $\mathbb{M}$ should provide a lot of information about $\mathbb{M}$ Spielman, 2011.

1.6.0.1 Reynolds decomposition for graphs

Le $\mathbb{M}$ be a $n \times n$ symmetric matrix representation of graph G with eigenvalues $\lambda_1 \leq \lambda_2 \leq, \ldots, \leq \lambda_n$. The Reynolds decomposition of $\mathbb{M}$ in terms of rank one matrix B_i is given by

$$\mathbb{M} = \underbrace{\sum_{i=1}^{k_s-1} \lambda_i B_i}_{\bar{u}=\text{Low Frequency}} + \underbrace{\sum_{i=k_s}^{n} \lambda_i B_i}_{\tilde{u}=\text{High Frequency}} \tag{1.46}$$

where $\bar{u}$ and $\tilde{u}$ can be viewed respectively as the mean component and the fluctuating component. The index k_s corresponds to the pseudo "cut-off frequency" that must be determined using an appropriate criteria. However, if $\mathbb{M} = \mathbf{L}_N$, then $\lambda_i \in [0,2]$ and

$$k_s = \underset{\{1\leq i\leq n |\ 0\leq\lambda_i\leq 1\}}{\arg\max} [\lambda_i]$$

Therefore, $\mathbb{M}$ takes the form

$$\mathbb{M} = \underbrace{\sum_{\lambda_i\leq 1} \lambda_i B_i}_{\bar{u}=\text{Low Frequency}} + \underbrace{\sum_{1<\lambda_i\leq 2} \lambda_i B_i}_{\tilde{u}=\text{High Frequency}} \tag{1.47}$$

1.6.0.2 Nearest Adjacency matrix

Instead of the original adjacency matrix, the idea is to represent the graph G with a matrix of low rank, called *nearest adjacency matrix*, $\hat{\mathbf{A}}$ with low lost of information. The goal is to approximate the original graph by a sparse graph and this makes operations on the graph such as clustering or community

detection easier. The nearest adjacency matrix can be, for example, retrieved using Frobenius norm or by maximization of the Kirchhoff index as follows:

$$k_d = \underset{1\leq k\leq n}{\arg\min} \|\mathbf{A} - \sum_{i=k}^{n} \lambda_i \mathbf{B}_i\|_{\mathrm{F}} \text{ where } \mathbf{A}(i,k) \in \{0,1\} \tag{1.48}$$

with t

$$\hat{H} = \sum_{i=k_d}^{n} \lambda_i B_i \tag{1.49}$$

The nearest adjacency matrix is obtained by thresholding the $\hat{H}$ matrix as follows:

$$\begin{aligned} \hat{\mathbf{A}}(i,j) &= 0 \text{ if } \hat{H}(i,j) < \frac{1}{2} \\ \hat{\mathbf{A}}(i,j) &= 1 \text{ if } \hat{H}(i,j) > \frac{1}{2} \end{aligned} \tag{1.50}$$

Using the Kirchhoff index strategy, we first start by determining all low rank approximations of $\mathbf{A}$

$$\hat{F}_k = \sum_{i=k}^{n} \lambda_i B_i \; k \in \{1, 2, \ldots, n\} \tag{1.51}$$

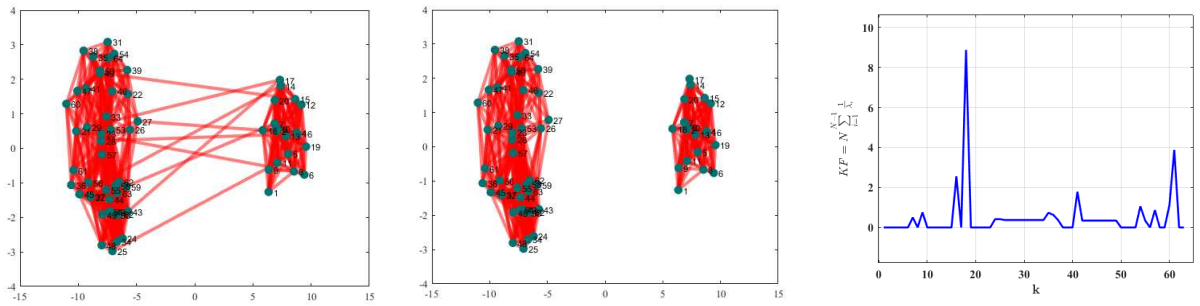
Then, these low rank approximated matrices, $\hat{F}_k$, are thresholded as follows:

$$\begin{aligned} \hat{\mathbf{A}}_k(i,j) &= 0 \text{ if } \hat{F}_k(i,j) < \frac{1}{2} \\ \hat{\mathbf{A}}_k(i,j) &= 1 \text{ if } \hat{F}_k(i,j) > \frac{1}{2} \end{aligned}$$

We retrieve from these thresholded matrices, $\hat{\mathbf{A}}_k$, the nearest one to $\mathbf{A}$ which shows the highest conductivity score according to Kirchhoff index (Eq. 1.17). We illustrate the second strategy on detection of communities of a graph (Figure 1.13(a)). Results are reported in Figure 1.13. The original graph is unweighted, and with no a priori about the strength of relationships between nodes. According to Kirchhoff index strategy, a set low rank approximation is calculated (1.50). Their corresponding Kirchhoff index is shown in Figure 1.13(c), where the plot exhibits a prominent peak. Since the Kirchhoff index is interpreted as a conductivity measure of the network, the nearest adjacency matrix $\hat{\mathbf{A}}_k$, associated to this peak corresponds to best approximation of $\mathbf{A}$ matrix in terms of conductivity (Figure 1.13(c)). The graph of this nearest matrix is given in Figure 1.13(b) and results in good partition of the graph, in two well separated communities (clusters).

1.6.1 Dominant eigen-Graph Analysis (DGA)

The principle of the Dominant eigenGraph Analysis (DGA) is inspired from that of the Dominant Component Analysis (DCA) used in image processing J.P. Havlicek et al., 2000. The idea of the DCA is to estimate at each instant t (or pixel for an image) in multicomponent signal (image), the values of the modulating functions of the component that dominates the local signal spectrum at that instant. DCA models the signal's nonstationary behavior at each instant by exclusively taking into account the



(a) (b) (c)

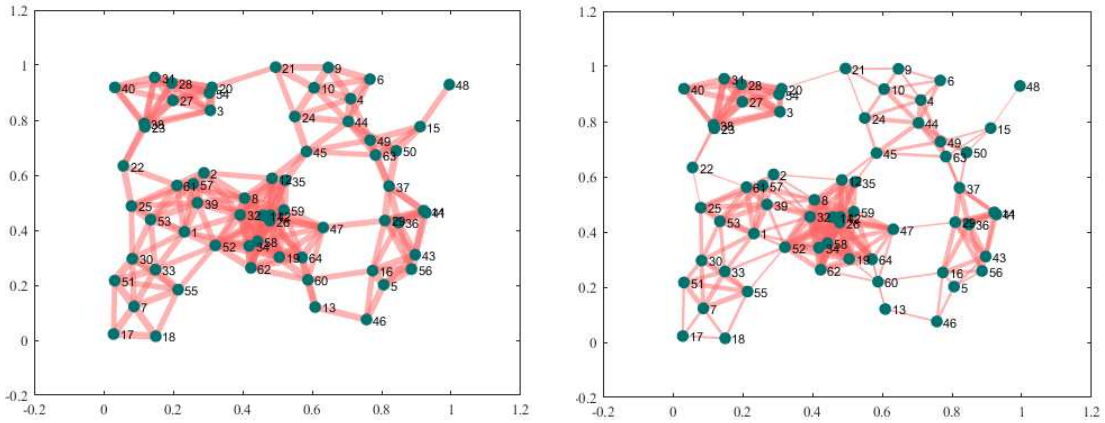
Figure 1.13: An Example of the nearest adjacency matrix in the sens of Kirchhoff's index obtained for the two communities graph, (a) Is the original graph, (b) The graph associated to the nearest adjacency matrix ($\hat{\mathbf{A}}_k$), (c) The Kirchhoff index corresponding to all row rank approximations.

component with the strongest response. Following the same principle, we introduce the idea of DGA, where a narrow band image is represented by an eigengraph: or *salient eigengraph*. The choice of the dominant eigengraph, DomG, is estimated, in the sense of a given criteria $\Gamma(.)$, as follows:

$$\mathrm{DomG}(k,l) = \max_{1 \leq i \leq n} [\lambda_i \Gamma(B_i(k,l))] \tag{1.52}$$

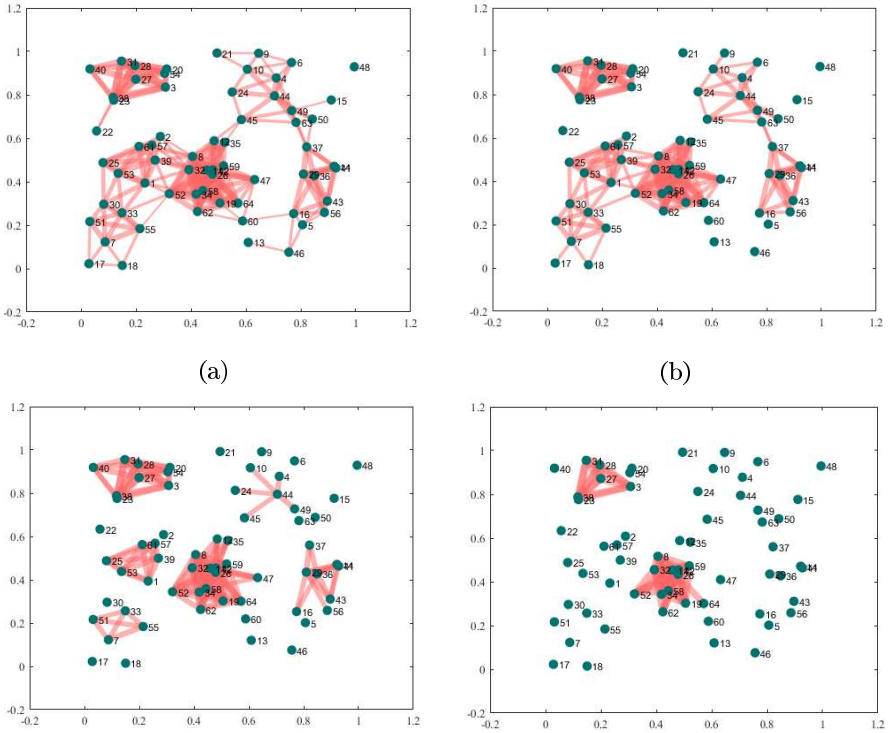
$\Gamma(.)$ can be the amplitude function.

The DGA is applied on sensors network (Figure 1.14(a)). The original graph is unweighted and no information is available about the salient structure of the graph. The DGA is applied using relation (1.52) with $\Gamma(.)$ as the amplitude (entries) function along the eigengraphs B_i of the original graph. The dominent graph is reported in Figure 1.14(b) and highlights dense subgraphs representing the backbone of the graph. Using different thresholds, multiscale representation of the dominant eigen-Graph (Figure 1.15).



(a) (b)

Figure 1.14: DGA approach applied to sensors graph. (a) The Original unweighted sensors graph, (b) Its corresponding Dominant eigen-Graph (DomG).



(a) (b)

(c) (d)

Figure 1.15: Multiscale representation of the dominant eigen-Graph corresponding to the sensors graph. (a), (b), (c) and (d) are the different backbone scales obtained respectively by the thresholds :0.2, 0.3, 0.5 and 0.7 of the maximal weight.

CHAPTER 2

Kernel Techniques for Graphs Classification

2.1 Introduction

Pattern recognition field is on full growth this last decade, driven by the big advances achieved on data management and processing techniques. Nevertheless, the efficient representation of data objects remains a central problem in learning processes, particularly for classification. To remedy this, two major ways are adopted by the community, vectorial and structural approaches. In the statistical one, Vapnik, 2013, objects are represented by feature vectors, that are a set of measured attributes. This offers some useful properties, in particular the mathematical wealth of tools adapted for vector spaces, such as the computing of a sum, a dot product, a mean or the distance between two instances. However, the use of feature vectors exhibits some drawbacks. First, as a vector always represents a predefined set of features, all vectors in a given problem have to preserve the same length regardless of the complexity or the size of the corresponding objects. Second, there is no direct possibility to describe binary relationships that might exist among different parts of an object. The structural approach, K. Riesen and H. Bunke, 2010, is based on symbolic data structures, such as strings, trees, or graphs. In fact, from an algorithmic viewpoint both strings and trees are a particular cases of graphs. The mentioned drawbacks of feature vectors can be overcome by graph-based representation N.B. Aoun et al., 2014. That is graphs are able to model inherent relationships of an object by means of edges, representing various connection natures (spatial, temporal, conceptual,...etc). Furthermore, graphs are not constrained in terms of size, where the number of nodes and edges can be customized to the target object size and complexity.

A large number of learning algorithms are designed exlusively for vectorial data, such as Support Vector Machines (SVM), Vapnik, 1998, Haasdonk, 2005 and K-Nearest Neighbors (KNN), J.M. Keller et al., 1985, and others. Nevertheless, these algorithms can be adapted to the structural data using kernel trick which replaces dot products by valid kernel functions, J. Shawe-Taylor and N. Cristianini, 2004.

The principle of use of kernels as dot products is rather well known for potential function classifiers, M.A. Aizerman et al., 1964, and it was revisited to construct support vector machines as a generalization of large margin classifiers, B.E. Boser et al., 1992. Futhermore, kernel based methods allow the extention of basic linear algorithms to complex non-linear ones in efficient way. So far, many kernel machines have been built to classify graph data, see: S.V.N. Vishwanathan et al., 2010, L. Bai et al., 2015a, N. M. Kriege et al., 2016, their common idea is to determine a manner for measuring similarity between graphs without embedding them into a vector space.

In this chapter, we recall some basic principles of statistical learning, focusing on the algorithmic design of the Support Vector Machines. And then, we explain the idea behind the kernel trick and how we can combine a kernel function into the SVM algorithm for classification. And finaly, we present some state-of-art kernels proposed for graphs. The ultimate goal of this chapter is to summarize the basic background needed to understant how to build a graph kernel and how to use it with an SVM machine for classifying graph data.

2.2 Learning Problem

Thinking, recognizing objects, learning from experience are the most powerful abilities of human beings. According to some previous knowledge, he is able to classify and distinguish trends, as well as infer new rules and new models to apply on completely unprecedented situations. Formulating these tasks in an algorithmic way using appropriate mathematical concepts provides us with the possibility to delegate their execution to machines. They are meant to learn autonomously the characteristics of a class, identify behavioral profiles and discriminate them. Typically, a machine is fed with training samples, coming from a certain real world process, whereon it tries to uncover hidden trends that fits the samples, and model the rules that solve a given recognition task. Through these training samples and particularly the inferred rules, the machine acquires generalization power via learning, making her able to predict and take decisions about the new unseen samples. Learning approaches are split to three families reffered to as supervised, unsupervised and semi-supervised learning. All these approaches share in common the need to a raw data and the fact that they express the behaviour of the process generating the data into an inclusive mathematical model.

Supervised learning: In supervised learning approach, the machine is fed in data labeled by another intelligence, often human. The labels designate classes of membership of the elements composing the data. The role of the machine is to determine a mathematical model of class assignment by learning patterns about classes, relying on the elements provided for training. We note that the test phase on new unseen data is necessary and it is only performed if the training phase is completed.

Unsupervised learning: Unlike in the supervised approach, raw data are not labelled in advance by any kind of intelligence, which justifies the unsupervised aspect of the approach. The machine performs blind learning looking for groups, affinities between data elements. In the end of the process, the data is segregated into homogeneous groups called clusters, hence, it is suitable for exploring inter-relationships among individual patterns. The concept of a cluster is close to that of a class, the difference is that the class of an object is intrinsic and can never be changed, while belonging to a cluster depends on several parameters that can be variable.

Semi-Supervised learning: Semi-supervised learning is a mix between supervised and unsupervised learning. The machine is fed on both labeled and unlabeled data. The aim is to propagate the labels to cover all the data using some similarity measure and a class attribution criteria. The initial distribution of labels plays a central role on this approach, and of course in many cases these methods perfom better than supervised methods. We notice that the approaches presented in this report are part of supervised learning methods for classification purposes only.

Formally, in the supervised learning approach, the data reside in a representatif space called pattern space and denoted $\mathcal{X}$, wherein each element $\mathbf{x} \in \mathcal{X}$ is a candidate for classification. The space $\mathcal{X}$ can contain vectorial data or any other kind of symbolic data like graphs. The other space $\mathcal{Y}$ contains all possible decisions that the classifier can make, therefore all possible labels. $\mathcal{Y}$ can be either the binary space $\{-1, +1\}$ for binary classification, or the space $\{\theta_1, ..., \theta_M\}$ of symbolic labels for multi-class case. As well as the labelled data samples used for learning are grouped in the set $\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_N, y_N)\}$. Considering that the correct classes are attributed according to an unknown target function $f : \mathcal{X} \to \mathcal{Y}$, that maps any instance $\mathbf{x} \in \mathcal{X}$ into the decision space $\mathcal{Y}$, then the learning algorithm exploits training data $\mathcal{D}$ to choose a function $g : \mathcal{X} \to \mathcal{Y}$ that approximates as possible the target function f. The function g could be any appropriate function, however, for practical consideration and to facilitate the searching processes, the algorithm picks g from a set of candidate functions, called hypothesis set $\mathcal{H}$. For instance, $\mathcal{H}$ could be the set of linear hyperplanes, polynomial, radial basis functions or even neural networks, see Figure 2.1 for an overview about the learning problem, Y.S. Abu-Mostafa et al., 2012.

The performance of a classifier can not be measured only in the training samples since it could classify them all correctly, while there is not any guarantee that it would do the same on the unseen data where it might perform poorly. In this case, the classifier is considered as in *overfitting*, because it is adapted too much to the training samples only. Otherwise, *underfitting* happens when the classifier fails

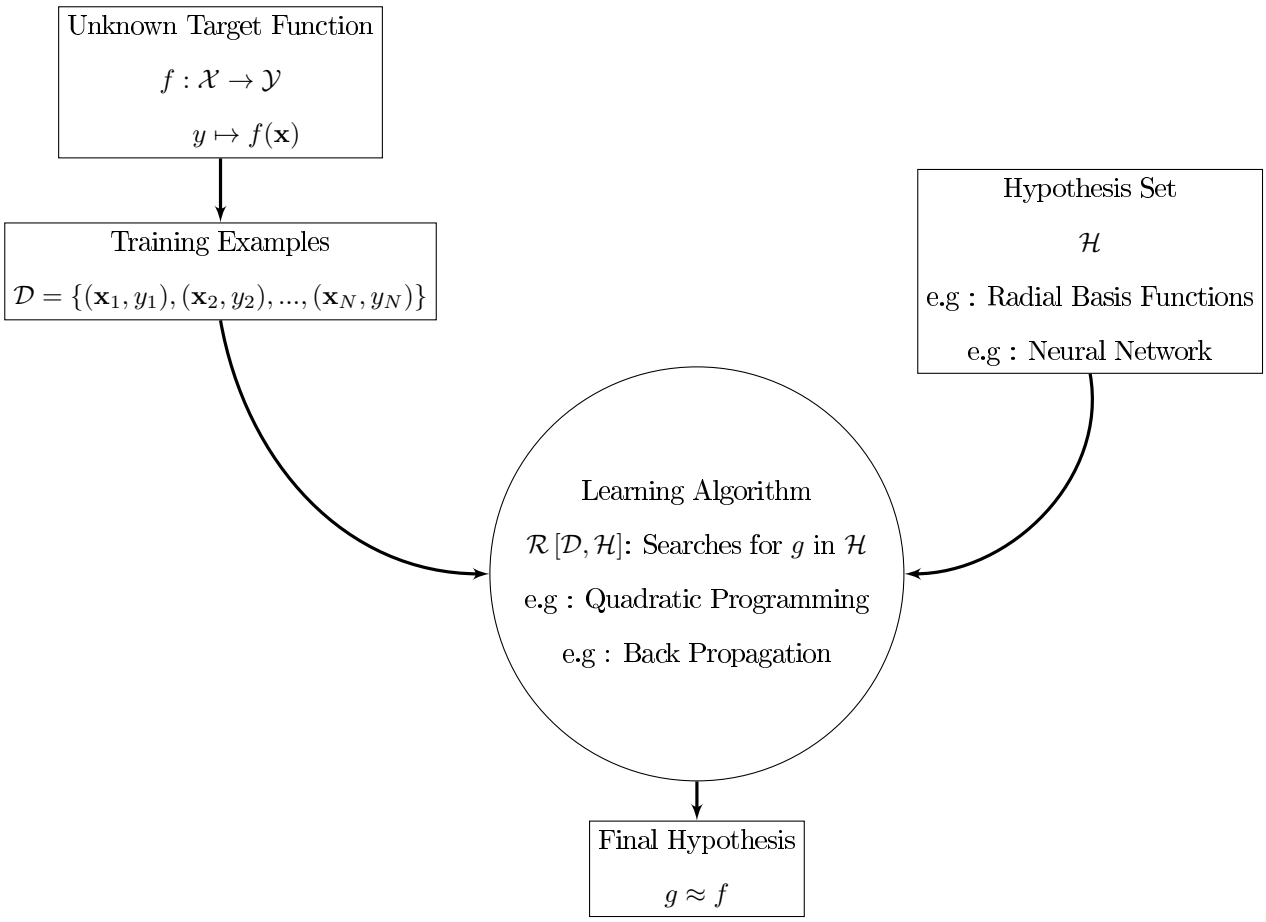


Figure 2.1: Basic Diagram of Learning Problem.

to approximate the target function f and to determine appropriate boundaries between classes. Hence, the machine should establish a compromise between *overfitting* and *underfitting* constraints during the learning (training) process. Consequently, the ultimate goal is to train a classifier using training data, that could classifiy adequately the bigger part of unseen patterns, or in other words, it shows a good generalization power to all possible unknown instances.

2.3 Linear Separable Data

Let $\{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_N, y_N)\} \subset \mathbb{R}^d \times \{+1, -1\}$ be a training set of a binary classification problem. B is the subset with labels $+1$ and C is the subset with labels -1. B and C are called linear separable in $\mathbb{R}^d$ if there is an hyper-plane $\langle \mathbf{w}, \mathbf{x} \rangle + b = 0$ and $\delta > 0$ such that $\langle \mathbf{w}, \mathbf{x} \rangle + b > \delta$ for $\mathbf{x} \in B$ and $\langle \mathbf{w}, \mathbf{x} \rangle + b < -\delta$ for $\mathbf{x} \in C$. It means that the distance $d(B, C)$ between the two subsets is positive when they are linearly separable. Suppose $\mathcal{X} \subset \mathbb{R}^d$ is a compact subset of surface of unit ball, and suppose an infinite dimensional Hilbert space $H^G(\mathcal{X})$, we denote by $\phi_d : \mathcal{X} \to H^{G_d}(\mathcal{X})$ the mapping function of $\mathcal{X}$ into the finite dimensional feature space $H^{G_d}(\mathcal{X})$:

Theorem 2.3.1 (On Linear Separability). *(D. Chen et al., 2007) Suppose $\mathcal{X} \subset \mathbb{R}^d$ is a compact subset of the surface of the unit ball and $\mathcal{X} = B \cup C$, $B \cap C = \emptyset$. Then $\phi(B)$ and $\phi(C)$ are linear separable in $H^G(\mathcal{X})$ if and only if the crowed point sets of B and C have empty overlap, i.e., the boundary point set of B and C is empty, (Figure 2.2).*

A binary classification problem can be considered as totally solved in linear way, if there is an hyperplane which can separate the classes in the training sample and also correctly classify every possible unseen data.

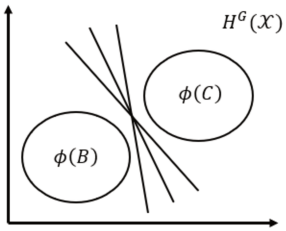


Figure 2.2: If $\mathcal{X}$ is compact and $\mathcal{X} = B \cup C$ s.t $B \cap C = \emptyset$, then $\phi(B)$ and $\phi(C)$ are linear separable in feature space $H^G(\mathcal{X})$.

The distance between the two convex hulls of two different classes is equal to zero if they are not linearly separable. Rather than using a nonlinear learning algorithm, that problem can be solved by increasing the dimensionality of the input space $\mathcal{X}$ using a non-linear mapping of the data to a higher feature space $\mathcal{F}$. The following theorem formalizes the intuition that increasing the dimensionality of representation space increases the number of possible linear separators that could separate correctly the classes, B. Schölkopf et al., 2002.

Cover's Theorem (Cover, 1965)**.** *Given a d-dimensional pattern space $\mathcal{X}$ and N points $\{\mathbf{x}_1, ..., \mathbf{x}_N\} \subseteq \mathcal{X}$ in general position. If $N \leq d + 1$, all 2^N separations are possible in $\mathcal{X}$. If $N > d + 1$, the number of linear separations amounts to:*

$$2 \sum_{i=0}^{d} \begin{pmatrix} N-1 \\ i \end{pmatrix}$$

This theorem provides a way to improve the performance of linear learning algorithms such as Support Vector Machines.

2.4 Linear Support Vector Machines

Pioneered by Vapnik, 1998, Vapnik, 2013, Support Vector Machine is one of the well known supervised learning algorithms, based on geometric linear algebra. It acts on data embedded in high dimensional

vector spaces. Considering a binary classification problem having a set of N labeled instances $\{\mathbf{x_i}, y_i/i = 1, ..., N\}$, with $\mathbf{x_i} \in \mathbb{R}^d$ be a d-dimensional real valued vector, labeled by y_i from $Y = \{-1, +1\}$. The learning task is to determine a function $f : \mathbb{R}^d \to Y$ that predicts the labels of unclassified data when satisfying $f(\mathbf{x_i}) = y_i$.

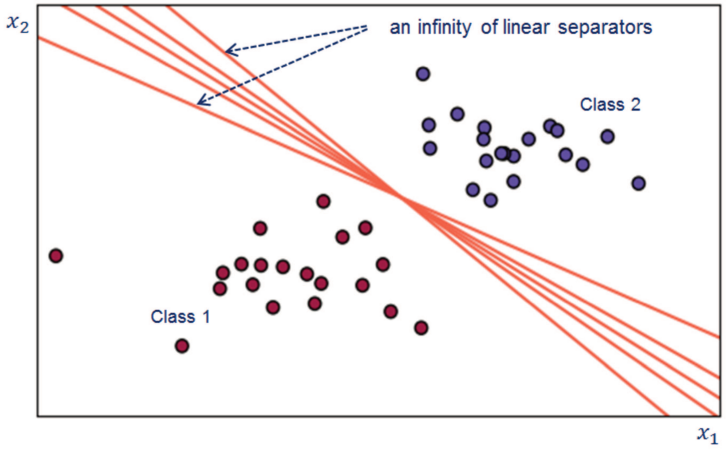


Figure 2.3: Linear Learning strategies seek to find the best separation hyperplane.

The SVM algorithm seeks an optimal linear separator maximizing the distance (margin) to the nearest instances from both classes (Figure 2.4). It takes the form of an hyperplane $\langle \mathbf{w}, \mathbf{x} \rangle + b = 0$, where $\mathbf{w} \in \mathbb{R}^d$ is the slope vector and $b \in \mathbb{R}$ is the bias. The margin between the nearest data samples and the optimal hyperplane is inversely proportional to the norm $\|\mathbf{w}\|$, as well as the search for this plan is reduced to the solving of the following optimization problem:

$$\min_{\mathbf{w} \in \mathbb{R}^d} \frac{1}{2} \|\mathbf{w}\|^2$$

$$\text{subject to: } y_i(\langle \mathbf{w}, \mathbf{x_i} \rangle + b) \geq 1, \forall i \in \{1, ..., N\}. \tag{2.1}$$

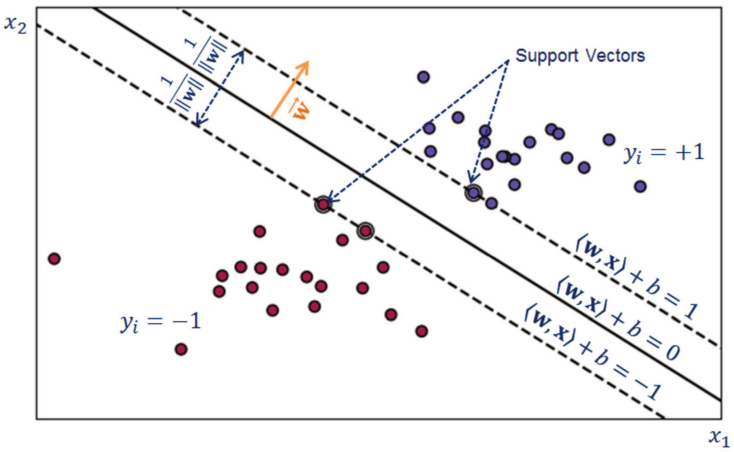


Figure 2.4: The SVM algorithm picks the hyperplane that maximizes the margin between classes.

This model supposes that data is linearly separable, while such constraint is hard to satisfy in most

classification problems. C. Cortes and V. Vapnik, 1995 introduced a new relaxed optimization problem using a set of slack variables $\xi_i \in \mathbb{R}^+$. These variables introduce some tolerance to misclassification during the learning phase. Hence, the problem (2.1) becomes:

$$\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i$$
$$\text{Subject to: } y_i(\langle \mathbf{w}, \mathbf{x_i}\rangle + b) \geq 1 - \xi_i, \text{ and } \quad \xi_i \geq 0, \ \forall i \in \{1, ..., N\}, \tag{2.2}$$

where $C \in \mathbb{R}$ is a regularization parameter allowing the weighting of misclassification error (tolerence).

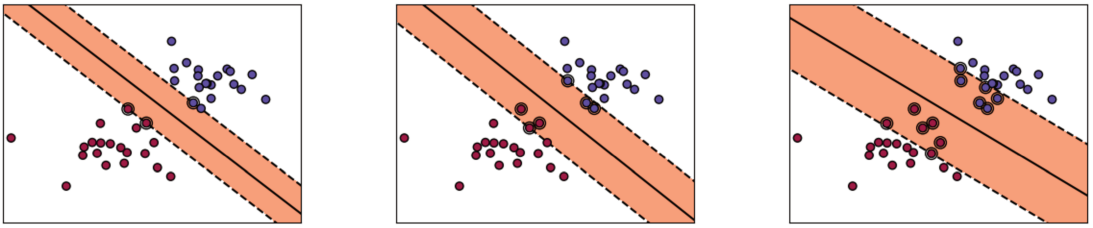


Figure 2.5: The SVM margin becomes larger when the regulation parameter C decreases. From left to right: $C = 1$, $C = 0.15$, $C = 0.04$.

The solution of the primal problem (2.2) can not be found easily without the use of the powerful quadratic programming resolvers, hence, we need to put it in a dual form. For this purpose, *Lagrange* function from equation (2.2) is introduced using *Lagrange* multipliers, each constraint is multiplied by a positive real number and subtracted from initial objective function. The *Lagrangian* of (2.2) takes then the form:

$$\mathcal{L}(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\alpha_i(y_i(\langle \mathbf{w}, \mathbf{x_i}\rangle + b) - 1 + \xi_i) - \sum_{i=1}^{N}\beta_i\xi_i, \tag{2.3}$$

where $\boldsymbol{\alpha}$ is the *Lagrange* multiplier concerning the separating hyperplane, and $\boldsymbol{\beta}$ concerns the positivity constraint of slack variables.

The constrained minimization problem in equation (2.2) can be solved by finding the saddle point of its *Lagrange* function (2.3). That point corresponds to the minimum of $\mathcal{L}$ according to the primal problem variables $(\mathbf{w}, b, \boldsymbol{\xi})$ and to the maximum of $\mathcal{L}$ according to the *Lagrange* multipliers $(\boldsymbol{\alpha}, \boldsymbol{\beta})$. For a fixed pair of multipliers $(\boldsymbol{\alpha}, \boldsymbol{\beta})$, the minimum of $\mathcal{L}$ according to $(\mathbf{w}, b, \boldsymbol{\xi})$ is obtained by canceling the following partial derivatives:

$$\frac{\mathcal{L}(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta})}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{N} \alpha_i y_i \mathbf{x_i} = 0, \tag{2.4}$$

$$\frac{\mathcal{L}(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta})}{\partial \boldsymbol{\xi}} = C\mathbf{e} - \boldsymbol{\alpha} - \boldsymbol{\beta} = 0, \tag{2.5}$$

$$\frac{\mathcal{L}(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta})}{\partial b} = -\boldsymbol{\alpha}^T \mathbf{y} = 0, \tag{2.6}$$

by inserting $\mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x_i}$ from (2.4) in equation (2.3), as well as equations (2.5) and (2.6), the minimum of $\mathcal{L}$ for any ($\boldsymbol{\alpha}$, $\boldsymbol{\beta}$) is:

$$\min_{(\mathbf{w}, b, \boldsymbol{\xi})} \mathcal{L}(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \mathbf{e}^T \boldsymbol{\alpha} - \frac{1}{2} \boldsymbol{\alpha}^T \mathbf{Q} \boldsymbol{\alpha}, \tag{2.7}$$

with $\mathbf{Q}_{i,j} = y_i y_j \langle \mathbf{x_i}, \mathbf{x_j} \rangle$.

The saddle point of $\mathcal{L}$ can then be obtained by maximizing the quantity (2.7) according to $\boldsymbol{\alpha}$:

$$\max_{\boldsymbol{\alpha}} \Big\{ \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i (y_i y_j \langle \mathbf{x_i}, \mathbf{x_j} \rangle) \alpha_j \Big\}$$
$$\text{Subject to: } \alpha_i \geq 0, \sum_{i=1}^{N} \alpha_i y_i = 0. \tag{2.8}$$

We notice that only $\boldsymbol{\alpha}$ appears in equation (2.7) and (2.8). Therefore, the maximization must be done while ensuring that the vector $\boldsymbol{\beta}$ satisfies the equation (2.5). Once the optimal coefficients $\boldsymbol{\alpha}$ are determined, the weights vector $\mathbf{w}$ is computed using the following equation (2.9):

$$\mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x_i} \tag{2.9}$$

while the scalar b is computed from $y_i(\mathbf{w}\mathbf{x_i} + b) = 1$ for any support vector. The class of a vector $\mathbf{x}$ outside the training samples is given by:

$$f(\mathbf{x}) = sign\Big(\sum_{i=1}^{N} (\alpha_i y_i \langle \mathbf{x}, \mathbf{x_j} \rangle) + b) \Big) \tag{2.10}$$

As we emphasized above, the SVM algorithm assumes that data is linearly separable, which is not the case of the majority real world learning problems. Moreover, the SVM takes only vectorial data as input, which is not practical for structural data such as graphs. These problem can be solved by using kernels instead of dot products in the SVM optimization problem. In the following sections, we give more details about kernels.

2.5 Kernel Functions

2.5.1 Kernel Trick

To understand the kernel trick, we need to make a reminder about dot products between a pair of vectors $\mathbf{x}, \mathbf{x}' \in \mathcal{H}$ in a Hilbert space.

Definition 2.5.1 (Dot Product). *A dot product in a vector space $\mathcal{H}$ is a function $\langle \cdot, \cdot \rangle : \mathcal{H} \times \mathcal{H} \to \mathbb{R}$ satisfying:*

- $\langle \mathbf{x}, \mathbf{x}' \rangle = \langle \mathbf{x}', \mathbf{x} \rangle$ *(Symmetry)*
- $\langle \alpha\mathbf{x} + \beta\mathbf{x}', \mathbf{x}'' \rangle = \alpha\langle \mathbf{x}, \mathbf{x}'' \rangle + \beta\langle \mathbf{x}', \mathbf{x}'' \rangle$ *(Linearity)*
- $\langle \mathbf{x}, \mathbf{x} \rangle = 0$ *for* $\mathbf{x} = \mathbf{0}$
- $\langle \mathbf{x}, \mathbf{x} \rangle > 0$ *for* $\mathbf{x} \neq \mathbf{0}$

for vectors $\mathbf{x}, \mathbf{x}', \mathbf{x}'' \in \mathcal{H}$ and scalars $\alpha, \beta \in \mathbb{R}$. If $\mathcal{H} = \mathbb{R}^d$, the standard dot product of two real vectors $\mathbf{x} = (x_1, ..., x_d), \mathbf{x}' = (x'_1, ..., x'_d) \in \mathbb{R}^d$ is given by $\langle \mathbf{x}, \mathbf{x}' \rangle = \sum_{i=1}^{d} x_i x'_i$.

In section 1.3, we explained the fact that using a non-linear mapping to map the data from input space to a higher dimensional feature space could solve learning problems when the data is not linearly separable. However, applying a non-linear mapping $\phi : \mathcal{X} \to \mathcal{F}$ to $\mathcal{X}$ is often costly, not practical and not feasible, especially, when the dimensionality of $\mathcal{F}$ is very high. Kernel trick offers a solution to this problem. It makes possible to carry out operations on the data embedded in the new space without knowing necessarily the mapping function ϕ or effectively access to $\mathcal{F}$. The simplest way to illustrate the kernel trick is to map data from $\mathbb{R}^2$ (input space $\mathcal{X}$) to $\mathbb{R}^3$ (feature space $\mathcal{F}$) using the direct mapping function: $\phi(\mathbf{x}) = (x_1^2, x_2^2, \sqrt{2}x_1x_2)$, see Figure 2.6.

Therefore, the dot product between two data instances in the new feature space $\mathcal{F} = \mathbb{R}^3$ is:

$$\begin{aligned}
\langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle &= \langle \phi(x_1, x_2), \phi(x'_1, x'_2) \rangle \\
&= x_1^2 x'^2_1 + x_2^2 x'^2_2 + 2x_1x_2x'_1x'_2 \\
&= (x_1x'_1 + x_2x'_2)^2 \\
&= \langle \mathbf{x}, \mathbf{x}' \rangle^2
\end{aligned}$$

It turns out that the dot product in the higher dimensional space $\mathcal{F}$ can be deducted by computing the squared dot product between instances in the input space $\mathcal{X}$, where the function $\mathcal{K}(\mathbf{x}, \mathbf{x}') = \langle \mathbf{x}, \mathbf{x}' \rangle^2$ is

called kernel. The euclidean distance $||\phi(\mathbf{x}) - \phi(\mathbf{x}')||$ between two mapped instances $\phi(\mathbf{x}), \phi(\mathbf{x}') \in \mathcal{F}$ can be deduced without computing the mapping function $\phi : \mathcal{X} \to \mathcal{F}$, as it can directly inferred from kernel function $\mathcal{K}$ in input space $\mathcal{X}$:

$$\begin{aligned} ||\phi(\mathbf{x}) - \phi(\mathbf{x}')|| &= \sqrt{\langle\phi(\mathbf{x}), \phi(\mathbf{x})\rangle + \langle\phi(\mathbf{x}'), \phi(\mathbf{x}')\rangle - 2\langle\phi(\mathbf{x}), \phi(\mathbf{x}')\rangle} \\ &= \sqrt{\langle\mathbf{x}, \mathbf{x}\rangle^2 + \langle\mathbf{x}', \mathbf{x}'\rangle^2 - 2\langle\mathbf{x}, \mathbf{x}'\rangle^2} \\ &= \sqrt{\mathcal{K}(\mathbf{x}, \mathbf{x}) + \mathcal{K}(\mathbf{x}', \mathbf{x}') - 2\mathcal{K}(\mathbf{x}, \mathbf{x}')} \end{aligned}$$

hence the kernel function $\mathcal{K}$ constitutes a shortcut for computing the dot product in $\mathcal{F} = \mathbb{R}^3$. According to the following theorem, each kernel $\mathcal{K}$ is actually a dot product in some implicit feature space $\mathcal{F}$. Hence, rather than mapping data from $\mathcal{X}$ to $\mathcal{F}$ and compute the dot product there, it is easier and faster to evaluate the value of the kernel function directly in $\mathcal{X}$:

Theorem 2.5.1 (J. Shawe-Taylor and N. Cristianini, 2004). *Let $\mathcal{K} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a valid kernel on a pattern space $\mathcal{X}$. There exists a possibly infinite-dimensional Hilbert space $\mathcal{F}$ and a mapping $\phi : \mathcal{X} \to \mathcal{F}$ such that*

$$\mathcal{K}(\mathbf{x}, \mathbf{x}') = \langle\phi(\mathbf{x}), \phi(\mathbf{x}')\rangle,$$

for all $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$ where $\langle ., . \rangle$ denotes the dot product in a Hilbert space $\mathcal{F}$.

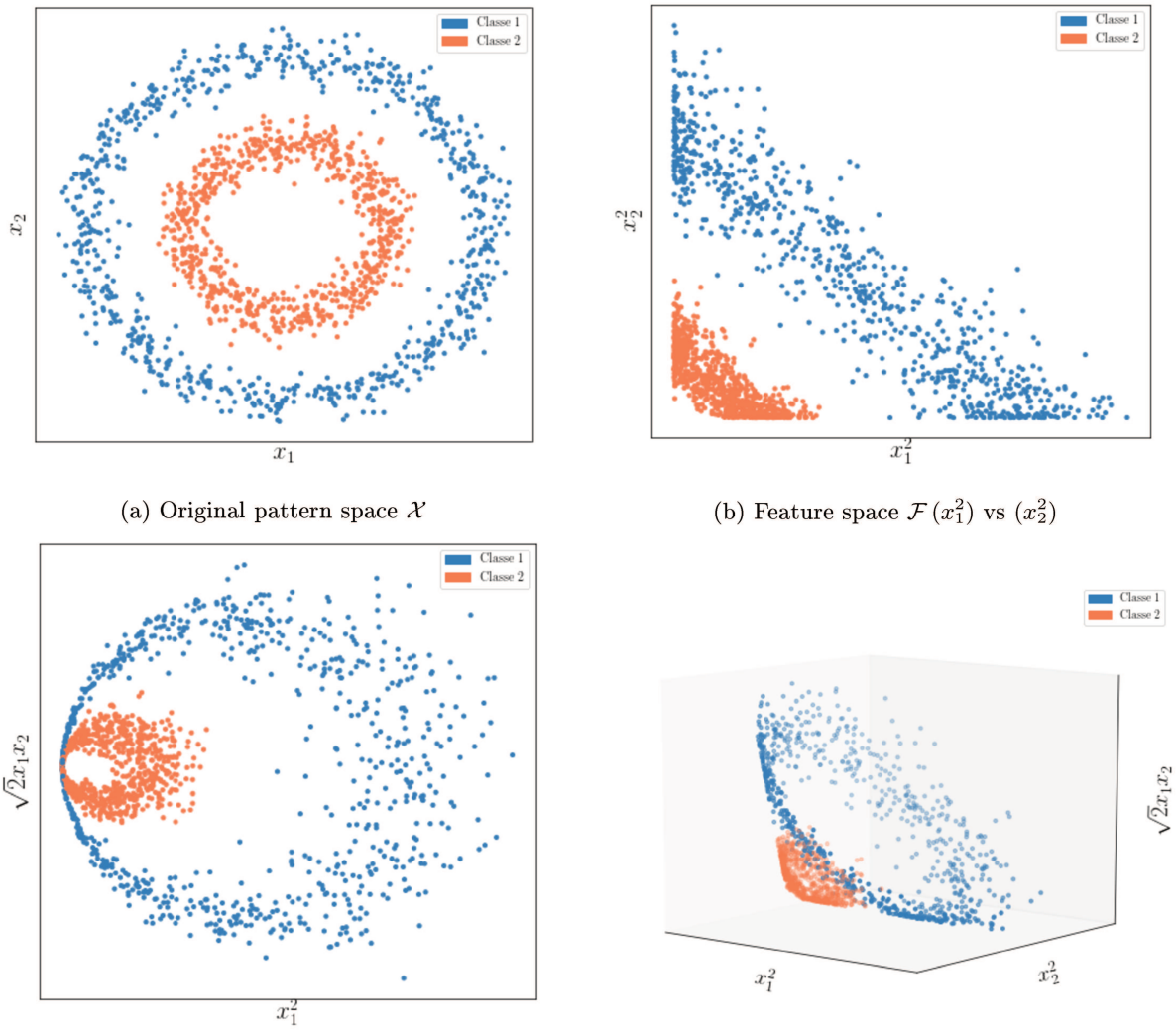
In Figure 2.6, we observe that the data is linearly separable in the new feature space $\mathcal{F}$, something that was not possible in the original space $\mathcal{X}$. Moreover, the use of kernel function $\mathcal{K}(\mathbf{x}, \mathbf{x}') = \langle\mathbf{x}, \mathbf{x}'\rangle^2$ is equivalent to the classification of data in feature space $\mathcal{F}$ without computing the mapping function $\phi(\mathbf{x})$.

2.5.2 Valid Kernels

Any measure of similarity defined in input space $\mathcal{X}$ can be considered as a kernel function. In addition of being symmetrical and real positive, it must correspond to a dot product in the new feature space $\mathcal{F}$. This can be checked out via the properties of the kernel matrix containing the mutual similarities between the data:

Definition 2.5.2 (Kernel Matrix). *Given a kernel $\mathcal{K}$ and a data set of N patterns $\{x_1, ..., x_N\} \subseteq \mathcal{X}$, we are able to form a $N \times N$ square matrix*

$$K = \begin{pmatrix} \mathcal{K}_{11} & \mathcal{K}_{12} & \cdots & \mathcal{K}_{1N} \\ \mathcal{K}_{21} & \mathcal{K}_{22} & \cdots & \mathcal{K}_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{K}_{N1} & \mathcal{K}_{N2} & \cdots & \mathcal{K}_{NN} \end{pmatrix} = \begin{pmatrix} \phi(\mathbf{x_1})^T\phi(\mathbf{x_1}) & \phi(\mathbf{x_1})^T\phi(\mathbf{x_2}) & \cdots & \phi(\mathbf{x_1})^T\phi(\mathbf{x_N}) \\ \phi(\mathbf{x_2})^T\phi(\mathbf{x_1}) & \phi(\mathbf{x_2})^T\phi(\mathbf{x_2}) & \cdots & \phi(\mathbf{x_2})^T\phi(\mathbf{x_N}) \\ \vdots & \vdots & \ddots & \vdots \\ \phi(\mathbf{x_N})^T\phi(\mathbf{x_1}) & \phi(\mathbf{x_N})^T\phi(\mathbf{x_2}) & \cdots & \phi(\mathbf{x_N})^T\phi(\mathbf{x_N}) \end{pmatrix}$$



(a) Original pattern space $\mathcal{X}$

(b) Feature space $\mathcal{F}\,(x_1^2)$ vs (x_2^2)

(c) Feature space $\mathcal{F}\,(x_1^2)$ vs $(\sqrt{2}x_1x_2)$

(d) Feature space $\mathcal{F}$

Figure 2.6: Example of data mapped from $\mathbb{R}^2$ (input space $\mathcal{X}$) to $\mathbb{R}^3$ (feature space $\mathcal{F}$), using the mapping function $\phi(\mathbf{x}) = (x_1^2, x_2^2, \sqrt{2}x_1x_2)$.

of real numbers $(\mathcal{K}_{i,j})_{i,j\in\{1,N\}}$ *commonly refferred to as kernel or Gram matrix. The matrix* $\mathbf{K} = (\mathcal{K}_{i,j})_{N\times N}$ *contains the kernel function values evaluated on all pairs of patterns in* $\{x_1, ..., x_N\}$.

A kernel function $\mathcal{K}$ is said to be valid and admissible, if its corresponding matrix $\mathbf{K}$ is positive semi definite. This fact is stated in the following theorem:

Mercer's Theorem. *(Mercer, 1909) Let* $\mathcal{X}$ *be a compact subset of* $\mathbb{R}^n$. *Suppose* $\mathcal{K}$ *is a continuous symmetric function such that the integral operator* $T_k : L_2(\mathcal{X}) \to L_2(\mathcal{X})$ *defined by*

$$(T_k f)(\,.\,) = \int_{\mathcal{X}} \mathcal{K}(\,.\,, \mathbf{x}) f(\mathbf{x}) d\mathbf{x}$$

is positive, which here means $\forall f \in L_2(\mathcal{X})$,

$$\int_{\mathcal{X}\times\mathcal{X}} \mathcal{K}(\mathbf{x},\mathbf{x}')f(\mathbf{x})f(\mathbf{x}')\,d\mathbf{x}\,d\mathbf{x}' \geq 0,$$

then we can expand $\mathcal{K}(\mathbf{x},\mathbf{x}')$ *in a uniformaly convergent series in terms of* T_k*'s eigenfunctions* $\psi_j \in L_2(\mathcal{X})$*, normalized so that* $||\psi||_2 = 1$*, and positive associated eigenvalues* $\lambda_j \geq 0$*,*

$$\mathcal{K}(\mathbf{x},\mathbf{x}') = \sum_{j=1}^{\infty} \lambda_j\, \psi_j(\mathbf{x})\, \psi_j(\mathbf{x}').$$

So we could define an infinite feature map in this way:

$$\phi(\mathbf{x}) = [\sqrt{\lambda_1}\psi_1(\mathbf{x}), \cdots, \sqrt{\lambda_j}\psi_j(\mathbf{x}), \cdots].$$

Involving that $\mathcal{K}(\mathbf{x},\mathbf{x}')$ is in fact a kernel function corresponding to the feature mapping ϕ. Consequently, it results from Mercer's theorem that a matrix is a Gram matrix, if and only if it is positive semi-definite, which means that it is a dot product matrix in some unknown feature space, N. Cristianini and J. Shawe-Taylor, 2000. Mercer's theorem checks out if a built kernel (similarity measure) is a dot product kernel.

Definition 2.5.3 (Positive Definite Matrix). *a matrix* $\mathbf{A}$ *is positive definite (PD) if all its eigenvalues are positive* $(\forall i\ \lambda_i(\mathbf{A}) > 0)$*; i.e, for all* $\mathbf{x} \in \mathcal{X}$*:*

$$\mathbf{x}^T\mathbf{A}\mathbf{x} > 0$$

from the Rayleigh quotient. We use $\mathbf{A} > 0$ *to denote that* $\mathbf{A}$ *is PD.*

Definition 2.5.4 (Positive Semi-Definite Matrix). *A matrix* $\mathbf{A}$ *is positive semi-definite (PSD) if all its eigenvalues are non-negative* $(\forall i\ \lambda_i(\mathbf{A}) \geq 0)$*; i.e, for all* $\mathbf{x} \in \mathcal{X}$*:*

$$\mathbf{x}^T\mathbf{A}\mathbf{x} \geq 0$$

We denote this by $\mathbf{A} \geqslant 0$*.*

Definition 2.5.5 (Positive Semi-Definite Kernel). *A kernel function* $\mathcal{K} : \mathcal{X}\times\mathcal{X} \to \mathbb{R}$ *is a symmetric function, i.e.* $\mathcal{K}(\mathbf{x}_i,\mathbf{x}_j) = \mathcal{K}(\mathbf{x}_j,\mathbf{x}_i)$*, mapping pairs of patterns* $\mathbf{x}_i,\mathbf{x}_j \in \mathcal{X}$ *to real numbers. A kernel function* $\mathcal{K}$ *is called positive semi definite, if, and only if, for all* $N \in \mathbb{N}$*,*

$$\sum_{i,j=1}^{N} \alpha_i\alpha_j\mathcal{K}(\mathbf{x}_i,\mathbf{x}_j) = \alpha^T\mathbf{K}\alpha \geq 0$$

for $\alpha \in \mathbb{R}^N$*, and any choice of* N *objects* $\{\mathbf{x}_1, ..., \mathbf{x}_N\} \subseteq \mathcal{X}$*.*

Kernel functions that are positive definite are often called *valid kernels*, *admissible kernels* or *Mercer kernels.*

2.5.3 Kernels Properties

Kernel functions measure similarity degree between the input objects, and therefore, it should be a way to combine different similarity measures to build more accurate new kernels. Those combinations are possible using the following closure properties.

Lemma 2.5.1 (Closure Properties). *Let $\mathcal{K}_1$ and $\mathcal{K}_2$ be valid kernels over $\mathcal{X} \times \mathcal{X}$, $\mathcal{K}_3$ a valid kernel over $\mathcal{H} \times \mathcal{H}$, $\phi : \mathcal{X} \to \mathcal{H}$, $f : \mathcal{X} \to \mathbb{R}$, and $a \in \mathbb{R}^+$. Then the kernel functions defined by*

1. $\mathcal{K}(\mathbf{x}, \mathbf{x}') = \mathcal{K}_1(\mathbf{x}, \mathbf{x}') + \mathcal{K}_2(\mathbf{x}, \mathbf{x}')$
2. $\mathcal{K}(\mathbf{x}, \mathbf{x}') = \mathcal{K}_1(\mathbf{x}, \mathbf{x}')\mathcal{K}_2(\mathbf{x}, \mathbf{x}')$
3. $\mathcal{K}(\mathbf{x}, \mathbf{x}') = a\mathcal{K}_1(\mathbf{x}, \mathbf{x}')$
4. $\mathcal{K}(\mathbf{x}, \mathbf{x}') = a + \mathcal{K}_1(\mathbf{x}, \mathbf{x}')$
5. $\mathcal{K}(\mathbf{x}, \mathbf{x}') = f(\mathbf{x})f(\mathbf{x}')$
6. $\mathcal{K}(\mathbf{x}, \mathbf{x}') = \mathcal{K}_3(\phi(\mathbf{x}), \phi(\mathbf{x}'))$

are also valid kernels.

2.5.4 Some Kernels for Vectorial Data

In fact, symmetric and positive definite kernel functions measure the similarity between patterns. Hence, the standard dot product in $\mathbb{R}^N$ can be interpreted as a kernel function, since it measures the geometric euclidean distance between data in $\mathbb{R}^N$. Some learning algorithms exploit the way how data is geometrically distributed in input space to separate the classes. Consequently, their mathematical formulation can be fully edited in terms of dot products.

Kernel Type	Definition	Parameter
Linear Kernel	$\mathcal{K}_{\langle .,. \rangle}(\mathbf{x}, \mathbf{x}') = \langle \mathbf{x}, \mathbf{x}' \rangle$	-
Polynomial Kernel	$\mathcal{K}_{poly}(\mathbf{x}, \mathbf{x}') = (\langle \mathbf{x}, \mathbf{x}' \rangle + c)^d$	$d \in \mathbb{N}$ and $c \geq 0$
Sigmoid Kernel	$\mathcal{K}_{sig}(\mathbf{x}, \mathbf{x}') = tanh(\alpha \langle \mathbf{x}, \mathbf{x}' \rangle + \beta)$	$\alpha > 0$ and $\beta < 0$
RBF Kernel	$\mathcal{K}_{rbf}(\mathbf{x}, \mathbf{x}') = exp\left(-\gamma \, \|\|\mathbf{x} - \mathbf{x}'\|\|^2\right)$	$\gamma > 0$

Table 2.1: Examples of some basic kernel functions applied on the vectors $\mathbf{x}, \mathbf{x}' \in \mathcal{H}$

Table 2.1 shows that other kernels can be derived from dot product kernel using closure properties such as polynomial and sigmoid kernels. The exponential function can be nearly approximated by poly-

nomials with positive coefficients. Therefore, the exponentiation $exp\,(\mathcal{K}(\mathbf{x},\mathbf{x}'))$ of a valid kernel $\mathcal{K}(\mathbf{x},\mathbf{x}')$ is also valid. The normalized version of that particular kernel is the radial basis function(RBF). We note that RBF, polynomial and dot product kernels are positive definite, while the sigmoid is not always valid. Further in our work, we will use the RBF kernel combined with the SVM to classify graphs. Therefore it is important to know that this particular kernel allows the classification of data in a space with infinite dimensionality, which increases the probability to linearly separate the classes. Some precisions are given below :

Radial Basis Function (RBF) is a popular kernel function used in various kernelized learning algorithms. It associates to a pair of vectors $\mathbf{x},\mathbf{x}'$ the similarity score:

$$\mathcal{K}(\mathbf{x},\mathbf{x}') = exp\Big(-\frac{||\mathbf{x}-\mathbf{x}'||^2}{2\sigma^2}\Big) \tag{2.11}$$

The quantity $||\mathbf{x}-\mathbf{x}'||^2$ refers to as the squared Euclidean distance between the two feature vectors $\mathbf{x}$ and $\mathbf{x}'$, where σ is a normalization parameter. The RBF kernel can also be expressed as $exp\,(-\gamma||\mathbf{x}-\mathbf{x}'||^2)$, putting $\gamma = \frac{1}{2\sigma^2}$. It projects data into an infinite dimensional feature space. Let $\mathbf{x}\in\mathbb{R}^n$ and $\gamma > 0$. In case of $n = 1$, we have:

$$\begin{aligned}
e^{-\gamma\,||x-x'||^2} &= e^{-\gamma\,x^2+2\gamma\,xx'-\gamma\,x'^2}\\
&= e^{-\gamma\,x^2-\gamma\,x'^2}\Big(1+\frac{2\gamma\,x\,x'}{1!}+\frac{(2\gamma\,x\,x')^2}{2!}+\frac{(2\gamma\,x\,x')^3}{3!}+\cdots\Big)\\
&= e^{-\gamma\,x^2-\gamma\,x'^2}\Big(1\cdot 1+\sqrt{\frac{2\gamma}{1!}}x\cdot\sqrt{\frac{2\gamma}{1!}}x'+\sqrt{\frac{(2\gamma)^2}{2!}}x^2\cdot\sqrt{\frac{(2\gamma)^2}{2!}}x'^2+\sqrt{\frac{(2\gamma)^3}{3!}}x^3\cdot\sqrt{\frac{(2\gamma)^3}{3!}}x'^3+\cdots\Big)\\
&= \phi(x)^T\phi(x')
\end{aligned}$$

where $\phi(x) = e^{-\gamma\,x^2}\Big[1,\sqrt{\frac{2\gamma}{1!}}x,\sqrt{\frac{(2\gamma)^2}{2!}}x^2,\sqrt{\frac{(2\gamma)^3}{3!}}x^3,\cdots\Big]^T$ is an infinite dimensional mapping.

2.6 Kernel-based Support Vector Machines

As shown in Table 1, the linear kernel corresponds to a simple dot product $\langle\mathbf{x},\mathbf{x}'\rangle$ between the vectors representing data in input space $\mathcal{X}$. This same dot product could be determined in the feature space $\mathcal{F}$ between the mapped versions of the vectors via a mapping function ϕ, yielding $\phi(\mathbf{x})^T\phi(\mathbf{x}')$. Whereas, the determination of the mapping function ϕ is not necessary, when the mutual similarities $\mathcal{K}(\mathbf{x},\mathbf{x}')$ between data samples are already known in input space $\mathcal{X}$. The kernel trick allows to transform some linear classification algorithms to a non-linear ones, when substituting dot products by appropriate kernel functions. Support Vector Machines take part of those algorithms, since the optimization problem in equation (2.8) includes dot products between the training samples, as well as in the decision function

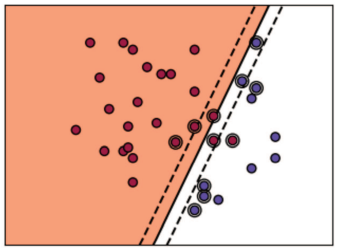
(2.10). Thus, a kernel function can be introduced replacing those dot products, leading to the following dual optimization problem:

$$\max_{\boldsymbol{\alpha}} \Big\{ \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i (y_i y_j \mathcal{K}(\mathbf{x_i}, \mathbf{x_j})) \alpha_j \Big\}$$
$$\text{Subject to: } \alpha_i \geq 0, \sum_{i=1}^{N} \alpha_i y_i = 0 \tag{2.12}$$

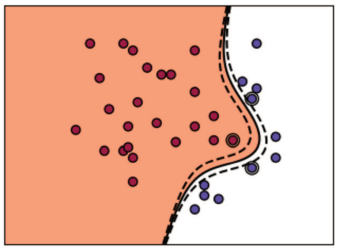
and the new decision function becomes:

$$f(\mathbf{x}) = sign\Big(\sum_{i=1}^{N} (\alpha_i y_i \mathcal{K}(\mathbf{x}, \mathbf{x_j})) + b) \Big) \tag{2.13}$$

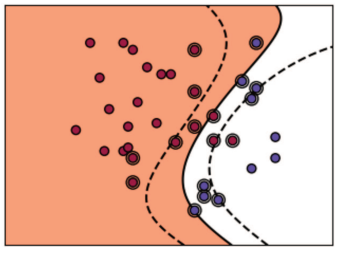
In Figure 2.7, we show the results from applying the kernelized SVM algorithm on some test data. We observe the performance of some kernel function when they try to separate the classes using different decision boundries. Indeed, kernels allow to separate data with a nonlinear boundary, which is useful for non linearly separable data.



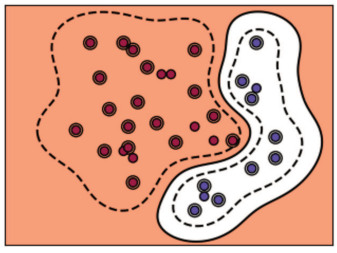
(a) Linear kernel



(b) Polynomial kernel



(c) Sigmoid kernel



(d) RBF kernel

Figure 2.7: The margin and the separation boundary obtained by integrating some kernels in the SVM algorithm.

2.7 Graph Kernels

In machine learning, the big part of kernel functions deals with data described by attribute vectors, while such representation is not adapted for structural data as the graphs. Their pertinence is limited, when applied to issues like biological sequences classification, R. Durbin et al., 1998, or toxicity prediction of chemical compounds, S.J Swamidass et al., 2005. Mathematically, standard kernels are applicable to data embedded in some space $\mathcal{X}$ endowed with a dot product only, which is not the case of graph represented data. As an alternative, many graph kernels have been developed to compare graphs and to generalize standard attribute based algorithms to structural data. We report in the following sections examples from the most important families of graph kernels, which we use further in our work as the state-of-art benchmark kernels.

2.7.1 Random Walk Kernel

A random walk on an undirected connected graph $G(V, \mathcal{E})$ of N nodes is a process that selects a sequence of k consecutive nodes $v^{(0)}, v^{(1)}, ..., v^{(t)}, ..., v^{(k)}$ randomly, such that $v \in V$ and $(v^{(t)}, v^{(t+1)}) \in \mathcal{E}$. The walker moves from the node $v_i^{(t)}$ to $v_j^{(t+1)}$ according to some transition probability:

$$P_{ij} = P(x^{(t+1)} = v_j \,|\, x^{(t)} = v_i), \tag{2.14}$$

The sum of transition probabilities from a node v_i to its neighbors ($\mathcal{N}(v_i) = \{v_j \,/ (v_i, v_j) \in \mathcal{E}\}$) is one, ($\sum_j P_{ij} = 1$). The row stochastic matrix $\mathbf{P}$ encodes all transition probabilities between the graph nodes. We note that in a random walk, the walker can visit a node more than once. Assuming that the walker goes from a node to the neighboring nodes according to an uniform probability, then the entries of transition matrix $\mathbf{P}$ are given by:

$$P_{ij} = \begin{cases} 1/d(v_i), & \text{if } (v_i, v_j) \in \mathcal{E} \\ 0, & \text{otherwise} \end{cases} \tag{2.15}$$

and also they can be written as: $P_{ij} = A_{ij}/d(v_i) = D_{ii}^{-1} A_{ij}$. Hence, $\mathbf{P}$ becomes nothing more than a normalized version of the adjacency matrix $\mathbf{A}$, where $D_{ij} = d_i \, \delta_{ij}$ is the diagonal degree matrix of G and $d(v_i)$ is the neighbors number of node v_i. Let $p_i^{(t)}$ be the probability that a walk is at node v_i at moment (t), then the probability that it will be at node v_j in the next moment $(t+1)$ depends on the transition probabilities within the selected path between them, and it is given by:

$$p_j^{(t+1)} = \sum_i P_{ij} p_i^{(t)} = \sum_i \frac{p_i^{(t)}}{d(v_j)} A_{ij}, \tag{2.16}$$

the probabilities $p_j^{(t+1)}$ are arranged in a vector determined by the matrices stated above:

$$p^{(t+1)} = p^{(t)}\mathbf{P} = p^{(t)}(\mathbf{D}^{-1}\mathbf{A}). \tag{2.17}$$

equation (2.17) shows that the initial probabilities are multiplied by the transition matrix after each step of the walk to get the new ones. Hence, after "k" steps, the probabilities vector becomes:

$$p^{(t+k)} = p^{(t)}\mathbf{P}^k = p^{(t)}(\mathbf{D}^{-1}\mathbf{A})^k. \tag{2.18}$$

The quantity $[(\mathbf{D}^{-1}\mathbf{A})^k]_{ij}$ is the probability of transition from node v_j to node v_i via a walk of length k. In case where $p^{(0)}$ is the initial probability distribution over nodes, the probability distribution $p^{(k)}$ predicting the position of the random walker after k steps is given by $p^{(k)} = (\mathbf{D}^{-1}\mathbf{A})^k p^{(0)}$. The j^{th} entry of $p^{(k)}$ is the probability of finishing a $k-$step walk at node v_j.

To measure the similarity between two graphs G and G', random walk kernel is based on the simple idea of counting the number of common walks between the concerned graphs. Noting that two walks are said to be common, if they have the same length, and that the neighborhood properties of the visited nodes are preserved in both graphs. Practically, common walks are determined using the adjacency matrix $\mathbf{A}_\times$ of the product graph $G_\times$ between G and G':

Definition 2.7.1 (Direct Product of Graphs). *Given two graphs $G(V, \mathcal{E})$ and $G'(V', \mathcal{E}')$, their direct product $G_\times$ is a graph with the following node and edge sets:*

$$V_\times = \{(v, v') \in V \times V' \,|v \in V, v' \in V'\}$$
$$\mathcal{E}_\times = \{((u, u'), (v, v')) \in V_\times \times V_\times \,|(u, v) \in \mathcal{E},\ (u', v') \in \mathcal{E}'\}$$

$G_\times$ is a graph over pairs of vertices from G and G', and two vertices in $G_\times$ are neighbors if and only if the corresponding vertices in G and G' are simultaneously neighbors. If $\mathbf{A}$ and $\mathbf{A}'$ are the respective adjacency matrices of G and G', then the adjacency matrix of $G_\times$ is $\mathbf{A}_\times = \mathbf{A} \otimes \mathbf{A}'$, where $(\otimes)$ is the Kronecker product.

T. Gärtner et al., 2003a, S.V.N. Vishwanathan et al., 2010, noticed that a random walk in a direct product graph is equivalent to a simultanieous random walk in the original graphs. Hence a similarity measure between G and G' can be defined using the normalized adjacency matrix of $G_\times$. For every $p \in \mathbb{N}$, the $p-$step random walk kernel between G and G' is:

$$\mathcal{K}_\times^p(G, G') = \sum_{i,j=1}^{|V_\times|} \Big[\sum_{l=0}^{p} \lambda_l (\mathbf{D}_\times^{-1}\mathbf{A}_\times)^l\Big]_{ij} \tag{2.19}$$

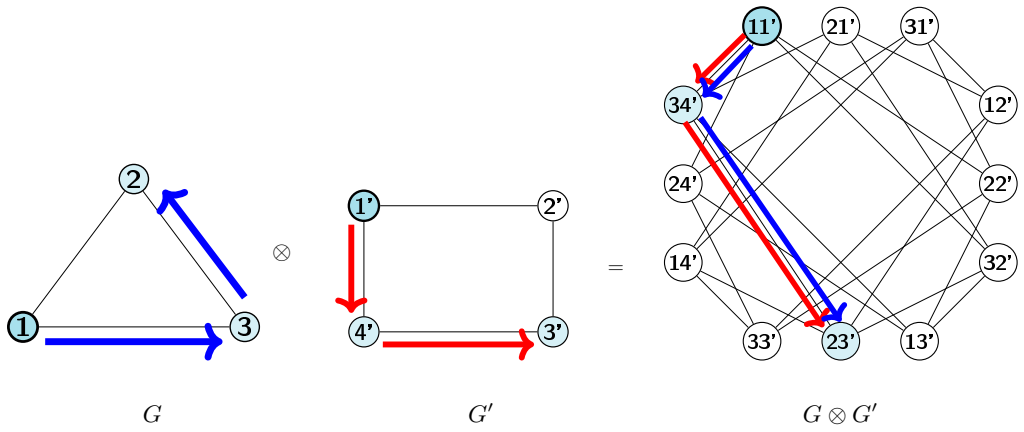


Figure 2.11: Direct product between the graphs G and G'. Every node of the direct product graph is labeled with a pair of nodes from G and G', an edge exists in the direct product if and only if the corresponding nodes are neighbors in both original graphs. A walk in a direct product graph is equivalent to a simultaneous walk with the same length in both graphs.

where $V_\times$ is the set of the $G_\times$ nodes and $\mathbf{D}_\times = \mathbf{D} \otimes \mathbf{D}'$ its degree matrix. The walks are weighted by a sequence of positive, real valued weights $\lambda_0, \lambda_1, \lambda_2, ..., \lambda_p$. Considering that $(\mathbf{D}_\times^{-1}\mathbf{A}_\times)^0 = \mathbf{I}$ being the identity matrix, its limit when p tends to ∞ is the so called *random walk kernel* ($\mathcal{K}_\times^\infty(G, G')$).

2.7.2 Geometric Random Walk kernel

Geometric random walk kernel is a special case of the standard random walk kernel proposed by T. Gärtner et al., 2003a. The difference resides in the weighting parameter λ_l, which takes the form of a geometric series in the sum, in which is replaced by λ^l. The advantage is to penalize the contribution of long walks to the kernel, as well as the limit of the random walk $\mathcal{K}_\times^p$ when $p \to \infty$ can be directly computed using the properties of geometric series, resulting in the following geometric random walk kernel :

$$\mathcal{K}_{Geo}(G, G') = \sum_{i,j=1}^{|V_\times|} \Big[\sum_{l=0}^{\infty} \lambda^l (\mathbf{D}_\times^{-1}\mathbf{A}_\times)^l \Big]_{ij} = \sum_{i,j=1}^{|V_\times|} \left[(\mathbf{I} - \lambda \tilde{\mathbf{A}}_\times)^{-1} \right]_{ij} \tag{2.20}$$

where I is the identity matrix and $\tilde{\mathbf{A}}_\times = \mathbf{D}_\times^{-1}\mathbf{A}_\times$ is the normalized adjacency matrix of the direct product graph $G_\times$.

Consider the equation $(\mathbf{I} - \lambda\tilde{\mathbf{A}}_\times)\,\mathbf{x} = \mathbf{0}$ for some vector $\mathbf{x}$, then $(\lambda\tilde{\mathbf{A}}_\times)^l\,\mathbf{x} = \mathbf{x}, \forall\, l \in \mathbb{N}$. So if $(\lambda\tilde{\mathbf{A}}_\times)^l$ converges to 0 when $l \to \infty$, then $(\mathbf{I} - \lambda\tilde{\mathbf{A}}_\times)$ is invertible since $\mathbf{x}$ becomes $\mathbf{0}$. Hence, from the formula $(\mathbf{I} - \lambda\tilde{\mathbf{A}}_\times)(\mathbf{I} + \lambda\tilde{\mathbf{A}}_\times + \lambda^2\tilde{\mathbf{A}}_\times^2 + ...)$, T. Gärtner et al., 2003b, it can be deduced that $\sum_{l=0}^{\infty} \lambda^l \tilde{\mathbf{A}}_\times^l = (\mathbf{I} - \lambda\tilde{\mathbf{A}}_\times)^{-1}$. Furthermore, Brualdi, 2011 shows that the geometric series of matrices, commonly called Neumann series $(I + \lambda\tilde{\mathbf{A}}_\times + (\lambda\tilde{\mathbf{A}}_\times)^2 + ... + (\lambda\tilde{\mathbf{A}}_\times)^\infty)$ converge only if the maximum eigenvalue of $\tilde{\mathbf{A}}_\times$, denoted by $\tilde{\mu}_{\times,Max}$,

is strictly smaller than $1 \setminus \lambda$. Hence, the geometric random walk kernel $\mathcal{K}_{Geo}$ is well-parametred only if $\lambda < 1 \setminus \tilde{\mu}_{\times,Max}$.

2.7.3 Graphlet Count kernel

Graphlets are small connected non-isomorphic subgraphs induced from larger networks. They were introduced for the first time by Pržulj, 2007 for designing a new highly sensitive method that measures local structural similarities of a graph. Formally, given a graph $G(V, \mathcal{E})$, we say that $G_H(V_H, \mathcal{E}_H)$ is a subgraph of G, denoted by $G_H \sqsubseteq G$, if and only if there is an injective mapping $\Omega_H : V_H \to V$ such that an edge $(v, w) \in \mathcal{E}_H$ exists only if $(\Omega_H(v), \Omega_H(w)) \in \mathcal{E}$ exists. It is important to precise that for each particular subgraph G_H, many injective mappings $\mathbf{\Omega} = \{\Omega_H | G_H \sqsubseteq G\}$ can be identified, representing different possible embeddings that G_H can have in G. The number of such mappings is denoted in the following by : $|\mathbf{\Omega}|$ or $\#\{\Omega_H | G_H \sqsubseteq G\}$. N. Shervashidze et al., 2009 propose a kernel function that compares graphs by counting graphlets of $p \in \{3, 4, 5\}$ nodes.

Consider $\mathcal{G}_p = \{G_i(V_i, \mathcal{E}_i) | i = 1, 2, ..., M\}$ be the set of size p graphlets and G be a graph of size N, proceeding to a matching between the $p-$graphlets and the graph G, a feature vector f_G of length M can be defined, whose i-th component corresponds to the occurence frequency of the graphlet G_i in G, which is equal to $\#\{\Omega_i | G_i \sqsubseteq G\}$. The feature vector f_G is called the $p-$spectrum of G, and is used to measure similarity between graphs. Given two graphs G and G' of size $N, N' \geq p$, the graphlet kernel $\mathcal{K}_g$ is defined as :

$$\mathcal{K}_g(G, G') = \langle f_G, f'_{G'} \rangle = f_G^T f'_{G'}. \tag{2.21}$$

In order to evaluate the kernel $\mathcal{K}_g$, a prior definition of all possible $p-$graphlets is needed, which is a hard task when p becomes big. Therefore, the kernel is basically studied using the $3, 4$, and $5-$spectra of the compared graphs. A graph that contains N nodes, has the quantity $\binom{N}{p}$ as number of $p-$graphelets, which is a hard task to enumerate. Hence, conceptors resorted to sampling in a way to reach a given confidence with a small probability of error. Unfortunately, the sampling algorithms proposed in the litterature: Pržulj, 2007, N. Kashtan et al., 2004, Wernicke, 2005 are ad-hoc and do not show restrictions about sampling rate, therefore N. Shervashidze et al., 2009 defined a bound involving some probability distributions.

Let $\mathcal{S} = \{1, 2, ..., a\}$ denotes a finite set of elements. Given a multiset $X := \{X_j\}_{j=1}^m$ of independent identically distributed random variables X_j drawn from some distribution η ($X_j \sim \eta$), the empirical

estimate of η via the m variables is given by :

$$\tilde{\eta}^m(i) = \frac{1}{m}\sum_{j=1}^{m}\delta(X_j = i), \tag{2.22}$$

where $i \in \mathcal{S}$.

Theorem 2.7.1. *Let η be a probability distribution on the finite set $\mathcal{S} = \{1, 2, ..., a\}$. Let $X := \{X_j\}_{j=1}^{m}$, with $(X_j \sim \eta)$. For a given $\xi > 0$ and $\tau > 0$,*

$$m = \frac{2(log\, 2 \cdot a + log\,(\frac{1}{\tau}))}{\xi^2} \tag{2.23}$$

samples suffice to ensure that $P\{\| \eta - \tilde{\eta}^m \|_1 \geq \xi\} \leq \tau$.

The theorem (2.7.1) can be applied in graphlets sampling problem, considering the set $\mathcal{S}$ as the set of all $p-$graphlets and suppose that their distribution follow some unknown probability law η, and m being the number of sampled graphlets for a given size of graphs. The quantity (2.23) bounds the minimum number of samples necessary for that $\tilde{\eta}^m$ gets closer to the real distribution η, with an error ξ and confidence $1-\tau$. As a particular case, the class of bounded degree graphs shows an opportunity to effectively identify all possible graphelets, hence, N. Shervashidze et al., 2009 present an algorithm for counting all connected graphlets that can be present in low degree graphs and established the following theorem :

Theorem 2.7.2. *(N. Shervashidze et al., 2009) Let G be a bounded degree graph, and let d_{max} denotes the maximum degree. Then all connected graphlets of G with size $p \in \{3, 4, 5\}$ can be enumerated in $O(N\,(d_{max})^{p-1})$ time.*

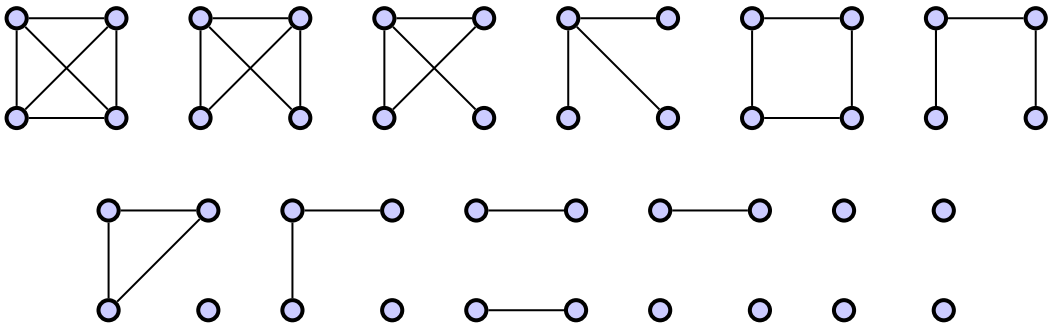


Figure 2.12: All possible graphlets with 4 nodes.

2.7.4 Ramon and Gärtner Subtree kernel

As stated in the previous section, the computation of kernels based on Random walks can be done in efficient manner by using direct product graphs and a form of power series of their adjacency matrices

($\mathbf{A}_\times$). However, some graphs can not be distinguished in feature space using only walk based maching kernels, thus, J. Ramon and T. Gärtner, 2003 proposed a kernel that counts the number of common *subtree patterns* between two graphs.

Consider a labeled graph $G(V, \mathcal{E}, \ell) \in \mathcal{G}$ with $\ell(v_i)$ the label of v_i. Therefore, any node $r \in V$ of G can be seen as a *subtree pattern* rooted at r and of height $h = 0$. Considering a set of subtree patterns $\{t_1, t_2, ..., t_n\}$ of G rooted at $\{r_1, r_2, ..., r_n | \, r_i \neq r_j \, \forall i, j \in \{1, 2, ..., n\}\}$ respectively, if these nodes are connected to an unique other node r such that : $(r, r_1), (r, r_2), ..., (r, r_n) \in \mathcal{E}$, then their combination in a new tree rooted at r is also a *subtree pattern* of G, where r is called the parent node of the root nodes r_i.

Each *subtree pattern* is characterized by a particular signature that distinguishes it from other subtrees, like in random walks which are described by the sequence of labels associated to the visited nodes during the walk. Otherwise, the numbers that refer to the occurrences of these signatures in the graph can be used as features that describe and distinguish the graph from other similar graphs.

Let $G(V, \mathcal{E}, \ell), G'(V', \mathcal{E}', \ell') \in \mathcal{G}$ be two labeled graphs. We denote by $\mathcal{K}_{r,s,h}$ the weighted count of subtrees pairs from both graphs that share the same signature, and that have a height not exceeding the value h, also, the subtrees are rooted respectively at $r \in V(G)$ and $s \in V(G')$. $\mathcal{K}_{r,s,h}$ value is given iteratively by :

$$\mathcal{K}_{r,s,h}(r, s) = \begin{cases} \delta(\ell(r), \ell(s)), & \text{if } h = 1 \\ \lambda_r \lambda_s \, \delta(\ell(r), \ell(s)) \sum_{R \in \mathcal{M}(r,s)} \prod_{(v,v') \in R} \mathcal{K}_{r,s,h-1}(v, v'), & \text{if } h > 1, \end{cases} \tag{2.24}$$

where :

$$\mathcal{M}(r, s) = \Big\{ R \subseteq \mathcal{N}(r) \times \mathcal{N}(s) | (\forall (u, u'), (v, v') \in R : u = v \Leftrightarrow u' = v') \cap (\forall (u, u') \in R : \ell(u) = \ell(u')) \Big\}, \tag{2.25}$$

and λ_r, λ_s are positive numbers less than one, used to ponderate and penalize the contribution of higher trees in the overall sum. The final subtree kernel of height h comparing all pairs of trees rooted at nodes from both $G(V, \mathcal{E}, \ell)$ and $G'(V', \mathcal{E}', \ell)$ is given by :

$$\mathcal{K}^h_{Tree}(G, G') = \sum_{r \in V} \sum_{s \in V'} \mathcal{K}_{r,s,h}(r, s), \tag{2.26}$$

For more precision, the set $\mathcal{M}(r, s)$ contains all the exact subsets matchings between the neighborhoods of r and s. Every subset R of $\mathcal{M}(r, s)$ contain pairs of nodes from the neighborhoods $\mathcal{N}(r)$ and

$\mathcal{N}(s)$, such that their labels are the same and that no node belongs to more than one pair. In other words, $\mathcal{K}_{r,s,h}$ takes into account all possible machings $\mathcal{M}(r,s)$ between the neighborhoods of two nodes r from G and s from G' that share the same label.

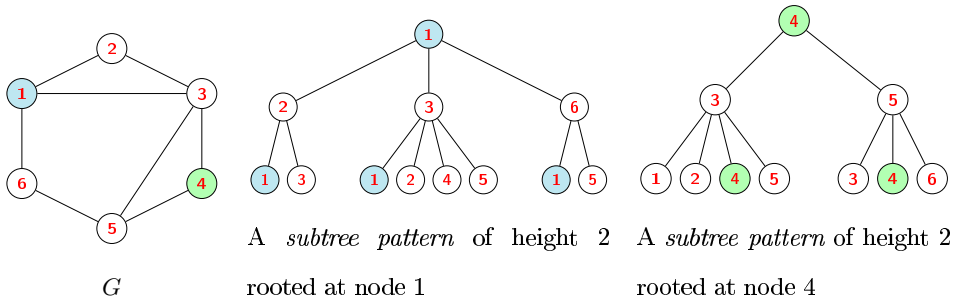


Figure 2.13: Examples of *subtree patterns* from the graph G.

2.7.5 Weisfeiler-Lehman Edge kernel

2.7.5.1 The Weisfeiler-Lehman Test of Isomorphism

The Weisfeiler-Lehman graph kernels are based on a concept developped by Weisfeiler and Lehman B. Weisfeiler and A.A. Lehman, 1968 to test the isomorphism of graphs using an iterative algorithm. In each iteration, the algorithm relables the graph nodes using the labels of their neighbors. We note that an isomorphism between two graphs G and G' is a bijective mapping between their respective node sets $g : V(G) \rightarrow V(G')$, such that any two nodes v and u are adjacent in G, if and only if $g(v)$ and $g(u)$ are adjacent in G'.

The key idea of the algorithm consists of adding to the original node label, the sorted neighboring labels as an extension, and then compressing it into a short one. These steps are repeated until that the label sets of G and G' mismatches, or the number of iterations becomes big. Label compression can be done by any function f satisfying injectivity condition and permitting a compact representation of the labels after extension, thus it ensures that the strings are distinguishable after compression. The sorting step is important, it permits the convergence of the labels sets to a unique set in case of isomorphic graphs.

The Weisfeiler-Lehman algorithm stops when the condition $(\{\ell_i(v)|v \in V\} \neq \{\ell_i(v')|v' \in V'\})$ is satisfied, meaning that the sets of the newly obtained labels of G and G' are different, and the graphs are then not isomorphic. Otherwise, after a large number of iterations, the graphs can be considered as isomorphic. However, J.Y. Cai et al., 1992 showed that the algorithm can miss in some cases the

Algorithm 1 One iteration of 1-D Weisfeiler-Lehman test of graph isomorphism

1. Neighboring-Labels Set determination

   · For $i = 0$, set $M_i(v) = \ell_0(v)$.

   · For $i > 0$, assign a set of neighboring labels $M_i(v)$ to each node v in G and G'

   which consists of the set $\{\ell_{i-1}(u) | u \in \mathcal{N}(v)\}$

2. Sorting the elements of each Neighboring-Labels Set

   · Sort elements in $M_i(v)$ in ascending order and concatenate them into a string $s_i(v)$.

   · Add $\ell_{i-1}(v)$ as a prefix to $s_i(v)$ ans call the resulting string $s_i(v)$.

3. Label compression

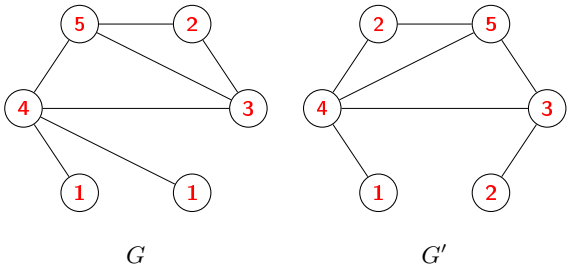
   · Sort all of the strings $s_i(v)$ for all v from G and G' in ascending order.

   · Map each string $s_i(v)$ to a new compressed label, using a function $f : \Sigma^* \to \Sigma$ such that $f(s_i(v)) = f(s_i(w))$ if and only if $s_i(v) = s_i(w)$ / Σ : is the set of all possible strings, $\Sigma^* \equiv \mathbb{N}$.

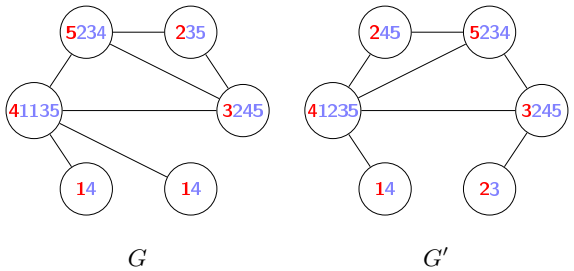
4. Relabeling

   · Set $\ell_i(v) = f(s_i(v))$ for all nodes in G and G'.

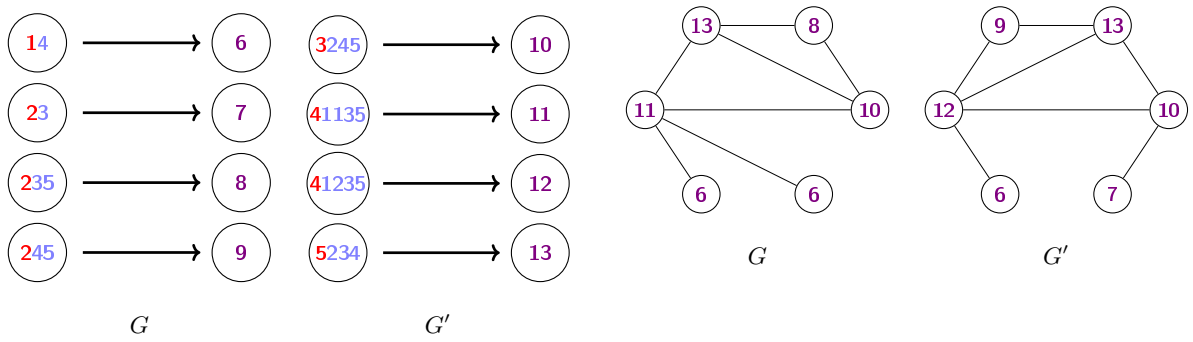
detection of the non isomorphism. Nevertheless, L. Babai and L. Kucera, 1979 have shown that the algorithm remains a valid test of isomorphism for the majority of graphs.



Two non-isomorphic labeled graphs G and G'



Steps 1 and 2 : Determination and sorting of nodes Neighboring-Labels



Step 3 : Example of Label compression and shortening

Step 4 : Relabeling

Figure 2.18: Example of Weisfeiler-Lehman graphs isomorphism test (One iteration).

2.7.5.2 The Weisfeiler-Lehman General Kernels

The Weisfeiler-Lehman kernels framework exploits the isomorphism testing algorithm presented above to measure the similarity between graphs. The relabeling procedure associates to the original graphs new labels $\ell_i(v)$ for all nodes V after each iteration, where these label-sets converge if the graphs are similar or identical. So, the idea is to associate to a given graph G a set of new graphs $\{G_i | i = 1, 2, ..., h\}$ that takes ℓ_i as labels and keeps the same structure of G. Every iteration of the Weisfeiler-Lehman relabeling process can be seen as a function π that generates a new graph from the previous one : $\pi((V, \mathcal{E}, \ell_i)) = (V, \mathcal{E}, \ell_{i+1})$, and behaves depending on the underlying graph properties. The following definitions formulate the WL-kernel idea based on the sequences of attribute graphs generated from the original compared graphs.

Definition 2.7.2 (N. Shervashidze et al., 2011). *Define the Weisfeiler-Lehman graph at height i of the graph $G = (V, \mathcal{E}, \ell_0)$ as the graph $G_i = (V, \mathcal{E}, \ell_i)$. We call the sequence of Weisfeiler-Lehman graphs of G up to height h the set :*

$$\{G_0, G_1, ..., G_h\} = \{(V, \mathcal{E}, \ell_0), (V, \mathcal{E}, \ell_1), ..., (V, \mathcal{E}, \ell_h)\},$$

where $G_0 = G$, $G_1 = \pi(G_0)$ and $\ell_0 = \ell$. G_1 is the graph resulting from the first relabling iteration.

Definition 2.7.3 (N. Shervashidze et al., 2011). *Let $\mathcal{K}$ be any valid positive semidefinite kernel for graphs, that is called the base kernel. Then the Weisfeiler-Lehman kernel up to h iterations based on $\mathcal{K}$ is given by :*

$$\mathcal{K}_{WL}^{(h)}(G, G') = \mathcal{K}(G_0, G'_0) + \mathcal{K}(G_1, G'_1) + \cdots + \mathcal{K}(G_h, G'_h), \tag{2.27}$$

where h is the number of Weisfeiler-Lehman iterations and $\{G_0, ..., G_h\}$, $\{G'_0, ..., G'_h\}$ are the Weisfeiler-Lehman sequences of G and G' respectively.

Theorem 2.7.3 (N. Shervashidze et al., 2011). *Let the base kernel $\mathcal{K}$ be any positive semidefinite kernel*

on graphs. Then the corresponding Weisfeiler-Lehman kernel $\mathcal{K}_{WL}^{(h)}$ is valid and positive semidefinite.

$\mathcal{K}_{WL}^{(h)}$ kernel generalizes basic kernel functions defined for discrete node-labeled graphs to different levels of node-labeling.

2.7.5.3 The Weisfeiler-Lehman Edge Kernel

The Weisfeiler-Lehman edge kernel is a particular instance of the Weisfeiler-Lehman kernels family defined by N. Shervashidze et al., 2011. It uses a base kernel $\mathcal{K}_{Edge}$ that counts the number of edges that have the same labels in endpoints between two graphs, and it is defined mathematically as :

$$\mathcal{K}_{Edge} = \langle \Phi_{Edge}(G), \Phi_{Edge}(G') \rangle, \tag{2.28}$$

where $\Phi_{Edge}(G)$ is an attributes vector containing the occurrences of ordered label pairs $(\ell(v_i), \ell(v_j))/(v_i, v_j) \in \mathcal{E}$, whith $\mathcal{E}$ is the edges set of G. Denoting $(\ell(v_i), \ell(v_j))$ and $(\ell^{'}(v_i'), \ell^{'}(v_j'))$ the ordered labels of endpoints concerning edges (v_i, v_j) and (v_i', v_j') from G and G' respectively, the base kernel $\mathcal{K}_{Edge}$ can be expressed as :

$$\sum_{(v_i,v_j)\in\mathcal{E}} \sum_{(v_i',v_j')\in\mathcal{E}'} \delta(\ell(v_i), \ell^{'}(v_i'))\ \delta(\ell(v_j), \ell^{'}(v_j')). \tag{2.29}$$

In the case of weighted edges via a weighting function $\mathcal{W}$, the base kernel $\mathcal{K}_{Edge}$ take then the form :

$$\sum_{(v_i,v_j)\in\mathcal{E}} \sum_{(v_i',v_j')\in\mathcal{E}'} \delta(\ell(v_i), \ell^{'}(v_i'))\ \delta(\ell(v_j), \ell^{'}(v_j'))\ \ \mathcal{K}_{\mathcal{W}}(\mathcal{W}(v_i, v_j), \mathcal{W}(v_i', v_j')), \tag{2.30}$$

where $\mathcal{K}_{\mathcal{W}}$ is a similarity function that compares edge weights. By introducing (2.29) or (2.30) in equation (2.27), we obtain the following new Weisfeiler-Lehman Edge kernel :

$$\mathcal{K}_{WL\,edge}^{(h)}(G, G') = \mathcal{K}_{Edge}(G_0, G'_0) + \mathcal{K}_{Edge}(G_1, G'_1) + \cdots + \mathcal{K}_{Edge}(G_h, G'_h). \tag{2.31}$$

2.8 Conclusion

In this chapter, we reviewed the essential of mathematical concepts that make support vector machines classify data in efficient manner in a large range of linear and non-linear learning problems. Via kernel functions, support vector machines are able to separate non-linearly separable data in a higher dimensional space. Their computational attractiveness is due to the fact that they can be applied in high dimensional feature spaces without suffering from the high cost of explicitly computing the feature map.

One advantage of kernel techniques among others is that they allow to run a large range of learning algorithms on structured data, so far restricted only for attribute-indexed data. As state of art, we presented some of these kernels developed particularly for graphs comparison and classification. Integrated in support vector machines, they show good performance on many applications as in bioinformatics, chemoinformatics and other real-world learning problems. Other than their expressitivity power, they hold several weaknesses. As many of them are NP-hard or at least complex to compute for large graphs, as well as being local and not include any kind of spectral information about the compared graphs. Therefore, we address thereafter the problem of graph similarity measure for classification, while ensuring that the measures be applicable over a wide range of graphs, less complex and easy to implement and adequately captures the topology of the underlying graphs.

CHAPTER 3

Energy and Total Variation for Graphs Classification

3.1 Introduction

Due to their ability to modelize complex relations in high-dimensional structured data, graphs constitute a potential tool for the analysis and the recovery of information from such data. One major interest in graph theory is to explore the structural differences between graphs, that is in the sense of graph isomorphism. However, from computational complexity perspective, the subgraph isomorphism problem is hard to solve, like many combinatorial problems in graph theory. Therefore, methods that gives quick and accurate estimate of the differences between two graphs are suitable for many applications, O. Macindoe and W. Richards, 2010, such as, the study of social media networks, Y. Shi and M. Macy, 2016, the detection of cyber attacks and anomalies in computer networks G.S. Bopche and B.M. Mehtre, 2017, the study of the brains connexions A. Mheich et al., 2015, and for pattern recognition tasks Y. Zhou et al., 2009, A. Sandryhaila and J.M.F. Moura, 2013. Some of earliest work in the subject were undertaken by L. G. Shapiro and R. M. Haralick, 1985, who showed how string edit distance could be extended to graph structures. The idea is to measure the similarity of graphs by counting the number of graph edit operations (i. e. node, edge insertions and deletions) required to transform a graph into another. However, the computational cost of such approach grows fast for larger graphs. More general approaches using concepts from information and probability theory was indeed proposed, such the work of W. J. Christmas et al., 1995, that show how a relaxation labeling technique can be employed to match graphs by using pairwise attributes modeled by a Gaussian distribution. Or the one of R. Myers et al., 2000, which uses maximum a prosteriori estimation to perform purely structural graph comparison, but it needs some adequate probabilistic setup to optimize performance. In many cases, graphs could be labeled by a sort of strings, for instance, B. Cao et al., 2013 use the depth-first search (DFS) algorithm as a graph labeling approach, and they measure the similarity by the distance between the two DFS

sequences, hence, by this way, the graph matching problem is turned to a string matching problem. More other structural compaison approaches was proposed, such as the aligned subtree kernel L. Bai et al., 2015b which incorporates explicit subtree correspondences between the compared graphs. Or assignment kernels N. M. Kriege et al., 2016, which decomposes the graphs into smaller sub-graphs and try to find the optimal bijection between them, or even those based on random walks F. Fouss et al., 2007 and quantum walks L. Bai et al., 2015a. So far, the majority of these methods, aims to sort the graphs using some structural similarity criteria without explicitly resorting to their spectral properties. While the spectrum of graphs has been widely used in graph theory to characterise their structural attributes, and it has also been employed for graph based pattern matching because of its invariance to labels changing. Therefore, spectral similarity measures remain attractive and hold many strengths for graphs discrimination, I. Jovanović and Z. Stanić, 2014, and many approaches have been proposed in this framework, we cite for instance, those based on spectral eccentricity, Dirac distance and Wasserstein distance, J. Gu et al., 2015.

In this chapter, we focus on the problem of graph-signals classification, using new similarity measures based on two discriminants, the total variation (TV) and laplacian graph energy ($E_{\mathbf{L}}$). The first one, is a structural attribute, which measures how the signal values changes (oscillate) upon the underlying graph structure. While the second one, is a spectral attribute, which measures the complexity degree of the graph and its connectivity, H.A. Bay-Ahmed et al., 2017b. We start by describing the advantages of the two concepts for graph characterization, then we propose three new similarity measures, which are of low complexity and easy to implement. The first one is based on TV, the second on $E_{\mathbf{L}}$ and the third is a weighted combination of the first ones. The measurements are then integrated into an exponential kernel adapted for an $C-SVM$ machine, in order to classify graph data from five different bioinformatics problems.

3.2 Total Variation Information

The characteraziation of continuous and discrete functions using the Total Variation (TV) concept was estabilished firstly by Jordan, 1881, in order to prove the convergence of Fourier series describing discontinuous periodic functions with bounded variation. In mathematics, the total variation refers to several closely related concepts, addressing local and global structural behaviour of a function f in his respective codomain. In the case of real valued continuous function f that takes the interval $[a,b] \subset \mathbb{R}$ as a definition domain, its corresponding total variation on $[a,b]$ measures the one-dimensional arclength of the curve described by $f\colon\ x \mapsto f(x)$, for $x \in [a,b]$. Other than this geometric interpretation, in signal processing, the total variation is used to characterize the oscillatory behaviour of a signal according to

one variable or more. Based on differential operators, the TV quantifies high variations, transitions and local fluctuations of a given signal. For instance, in image restoration problem, the $TV-$based approach use the ℓ_1-norm of the magnitude corresponding to the image's intensity gradient to restor corners and outlines, when minimizing the geometric oscillations of features in the reconstructed image L. I. Rudin and S. Osher, 1994. Total variation has been indeed used for image filtering for recovering *piecewise-constant* signals and for noise reduction while preserving prominent contours and edges in the underlying signal L.I. Rudin et al., 1992, Chambolle, 2004, I. W. Selesnick and P. Y. Chen, 2013. Furthermore, a wider range of $TV-$based approaches were proposed for sparse signal processing applications, such as deconvolution J. Oliveira et al., 2009, image reconstruction Y. Wang et al., 2008, compressed sensing W. Yin et al., 2008, interpolation and others.

3.2.1 Total Variation of 1D Signals

The total variation computed on 1D-signals, measures the total amplitude of oscillations and the roughness degree of their envelope. It applies a first, second or higher order derivatives to the signal in order to quantify fast variations and express them in a unique value that gives an indication about how smooth/rough the signal is. Let f be a one-dimensional differentiable continuous function, then its total variation is given by:

$$TV(f) = \int_{-\infty}^{+\infty} \mid f'(t) \mid dt \tag{3.1}$$

where f' refers to the first order derivative of f. For a given set of M local extrema and minima of f indexed by the abscissa $\{t_p : p = 1, ..., M\}$, in which $f'(t_p) = 0$, then its total variation can as well be defined as:

$$TV(f) = \sum_{p=1}^{M-1} \mid f(t_{p+1}) - f(t_p) \mid \tag{3.2}$$

Compared via their TV value, the gaussian-like function $f(t) = exp(-t^2)$ seems to be smoother than the cardinal sine function $f(t) = sin(\pi t)/(\pi t)$ which exhibits an infinite series of oscillations. The first one have a finite total variation ($TV(f) = 2$), while it is infinite for the second one ($TV(f) = +\infty$), since the cardinal sine has a local extrema at $t_p \in [p, p+1]$, and $\mid f(t_{p+1}) - f(t_p) \mid \sim \mid p \mid^{-1}$. In the case of non-differentiable functions, the total variation can not be calculated using equation (3.1). But even so, it can be defined alternatively in the sense of distributions, by approximating the derivative through a

finite difference on an interval h which tends towards zero:

$$TV(f) = \lim_{h \to 0} \int_{-\infty}^{+\infty} \frac{\mid f(t) - f(t-h) \mid}{\mid h \mid} dt. \tag{3.3}$$

The function f is said to have bounded variation if its total variation converges ($TV(f) < +\infty$). Although the total variation is well defined in the time domain, it remains difficult to evaluate in the spectral domain. Whether f' is the classical derivative of f or its derivative in the sense of distributions, its Fourier transform in general is given by: $\hat{f}'(\omega) = i\omega\hat{f}(\omega)$, hence a relation between the spectrum module and the total variation of f in time can be established:

$$\mid \omega \mid\mid \hat{f}(\omega) \mid \leqslant \int_{-\infty}^{+\infty} \mid f'(t) \mid dt = TV(f). \tag{3.4}$$

leading to:

$$\mid \hat{f}(\omega) \mid \leqslant \frac{TV(f)}{\mid \omega \mid}. \tag{3.5}$$

Although that $\mid \hat{f}(\omega) \mid = O(\mid TV(f) \mid\mid \omega \mid^{-1})$, it is not guaranteed that f has bounded variation. That observation is argumented by the case of cardinal sinus $f(t) = sin(\pi t)/(\pi t)$, which has a constant spectrum $\hat{f} = \mathbf{1}_{[-\pi,\pi]}$ in the interval $[-\pi, \pi]$ and inwhich, it satisfies the unequality $\mid \hat{f}(\omega) \mid \leqslant \pi \mid \omega \mid^{-1}$ despite the divergence of its corresponding total variation $\| f \|_V = +\infty$. In general, the total variation of f can not be deduced directly from $\mid \hat{f}(\omega) \mid$.

Furthermore, in the case of discrete signals, the discrete total variation is defined as the accumulation of the differences between consecutive signal samples. Let $f_M[n]$ be a discrete signal obtained from a continuous function f sampled uniformly at intervals of length $1/M$. The discrete TV is obtained by approximating the first order derivative by a finite difference over the sampling interval $h = 1/M$, and then approximate the integral in equation (3.3) by a Riemann sum, which gives:

$$TV(f_M) = \sum_n \mid f_M[n] - f_M[n-1] \mid . \tag{3.6}$$

The discrete signal is said to have a *bounded variation* if its $TV(f_M)$ is bounded by a constant value independent of the sampling rate. By putting the discrete signal in a column vector $f_M = [f_M(0), f_M(1), ..., f_M(M-1)] \in \mathbb{R}^M$, the total variation of f_M can also be written as:

$$TV(f_M) = \| \nabla f_M \|_1, \tag{3.7}$$

where $\| \cdot \|_1$ is the ℓ_1-norm and $\boldsymbol{\nabla}$ is the first-order derivative operator taking the form:

$$\boldsymbol{\nabla} = \begin{pmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \end{pmatrix}, \tag{3.8}$$

and having a size $(M-1) \times M$ corresponding to a signal of M-samples. The shape of the derivation operator $\boldsymbol{\nabla}$ plays an important role in total variation generalization approaches.

3.2.2 Total Variation of 2D Signals

Ostensibly, the total variation of 2D signals or images preserves the same interpretation that the one in 1D signals. Its strength depends on the magnitude of the variations as well as the length of the contours where the transition occurs. Let $f(x_1, x_2)$ be a two-dimentional continuous differentiable function, then its total variation can be written as:

$$TV(f) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \mid \overrightarrow{\nabla} f(x_1, x_2) \mid dx_1 dx_2, \tag{3.9}$$

where the operator $\overrightarrow{\nabla}(\cdot)$ is the 2D gradient vector, having a modulus of the form:

$$\mid \overrightarrow{\nabla} f(x_1, x_2) \mid = \left(\left| \frac{\partial f(x_1, x_2)}{\partial x_1} \right|^2 + \left| \frac{\partial f(x_1, x_2)}{\partial x_2} \right|^2 \right)^{1/2}. \tag{3.10}$$

Similar to the one-dimentional TV, the total variation (3.9) can be generalized to include the case of discontinuous 2D signals, by defining once again the derivatives in the general sense of distributions. Therefore, an equivalent expression is obtained by using finite differences over small intervals to approximate the partial derivatives:

$$\mid \Delta_h f(x_1, x_2) \mid = \left(\left| \frac{f(x_1, x_2) - f(x_1 - h, x_2)}{h} \right|^2 + \left| \frac{f(x_1, x_2) - f(x_1, x_2 - h)}{h} \right|^2 \right)^{1/2}, \tag{3.11}$$

where h is a narrow interval and $\mid \Delta_h f(x_1, x_2) \mid$ is an approximation of $\mid \overrightarrow{\nabla} f(x_1, x_2) \mid$. Moreover, the error of the approximated total variation is bounded by an upper and lower bounds Mallat, 1999:

$$TV(f) \leqslant \lim_{h \to 0} \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \mid \Delta_h f(x_1, x_2) \mid dx_1 dx_2 \leqslant \sqrt{2}\, TV(f). \tag{3.12}$$

The finite difference integral in (3.12) gets larger when the 2D signal exibits discontinuities along the diagonal of the plane $f(x_1, x_2)$. In the particular case of images, the sensor captures light intensity in discrete way, by averaging the intensity in a small region and by sampling the covered space uniformly, forming a grid of equally spaced samples. The sampled image is thus denoted by $f_N[n_1, n_2]$, resulting from a discretisation and averaging process of the original analog image $f(x_1, x_2)$, with N is the distance between two neighboring samples (resolution). The total variation is then defined by approximating the derivatives along the two axes as a finite differences over the distance (interval) N, and by expressing the integral of inequality (3.12) as a Riemann sum:

$$TV(f_N) = \frac{1}{N} \sum_{n1} \sum_{n2} \left(\left| f_N[n_1, n_2] - f_N[n_1 - 1, n_2] \right|^2 + \left| f_N[n_1, n_2] - f_N[n_1, n_2 - 1] \right|^2 \right)^{1/2}. \tag{3.13}$$

In digital image processing, total variation based denoising techniques, often expressed as regularization problems, are very effective in noise removal and image restoration. Initiated by L.I. Rudin et al., 1992, these approaches manage to smooth and remove noise in rough areas, while preserving important edges and features. The basic idea is that signals having high total variation values, contain often exorbitant details due to noise presence, which increases the modulus of the variation gradient. Therefore , $TV-$based denoising techniques aim to reduce the total variation of noisy signal subject to approach as possible the original signal, by removing irrelevant details and keeping the important ones as edges and contours.

3.2.3 Total Variation of Graph Signals

The central element in the calculation of the total variation of a signal is the derivation operator which quantifies local variations of the signal. As equations (3.6) and (3.13) show, a first order derivation of a discrete signal is approximated by a difference between neighbouring samples, that is, a difference between the original version of the signal and its shifted version. In the case of conventional 1D and 2D signals, the samples are distributed uniformly on a regular lattice and the shifting operation is well defined. Yet very often, graph signals are supported on an irregular and non-uniform structure, which opens the way to several forms of shifting operators. The basic non-trivial operator is called *graph shift* and is defined on a graph signal $G(V, \mathbf{A}, \mathbf{f})$ as a local operation that substitute the signal's value f_n at node v_n by a linear combination of the values of its neighbors ($v_m \in \mathcal{N}_n$) $\tilde{f}_n$:

$$\tilde{f}_n = \sum_{m \in \mathcal{N}_n} \mathbf{A}_{n,m} \, f_m, \tag{3.14}$$

Therefore, the resultant shifted signal is nothing else than the input signal multiplied by the adjacency matrix of the graph:

$$\tilde{\mathbf{f}} = [\tilde{f}_0, ..., \tilde{f}_{N-1}]^T = \mathbf{A}\,\mathbf{f}, \tag{3.15}$$

where N is the number of nodes, $\mathcal{N}_n$ is the set of v_n's neighbors and $\mathbf{A}$ is the adjacency matrix of the graph.

The total variation of equations (3.6) and (3.13) are adapted for time series and spacial signals. In (3.6), the total variation compares every adjacent samples and cumulates the difference between them over time, that is equivalent to compare the input signal $\mathbf{f}$ to its shifted version $\tilde{\mathbf{f}}$. Hence, the bigger is the difference between them, the higher is the total variation of $\mathbf{f}$. In the case of a graph signal, the difference is supposed to be between the signal value at every node and its neighbors, therefore, a generalization of the total variation (3.6) from the particular case of regular graphs to the general case is done by using the adjacency matrix as a shifting operator (3.15) and measuring the similarity between the graph signal and its shifted version:

Definition 3.2.1. *(S. Chen et al., 2015b): The total variation of a graph signal $G(V, \mathbf{A}, \mathbf{f})$ is defined as:*

$$TV_G(\mathbf{f}) = \| \mathbf{f} - \mathbf{A}\,\mathbf{f} \|_1, \tag{3.16}$$

where $\mathbf{A}$ is the adjacency matrix of G, $\mathbf{A}\,\mathbf{f} = \tilde{\mathbf{f}}$ is the shifted version of the signal $\mathbf{f}$ and $\| \cdot \|_1$ is the ℓ_1-norm.

Moreover, the 1D total variation defined in (3.6) is no more than a particular case of the one defined for graphs in (3.16). For intance, for a finite/periodic time series $\mathbf{f}$ of length N, i.e. Figure 3.1, the periodicity constraint $f_n = f_n \; mod\, N$ implies a modification of the total variation definition to take the form:

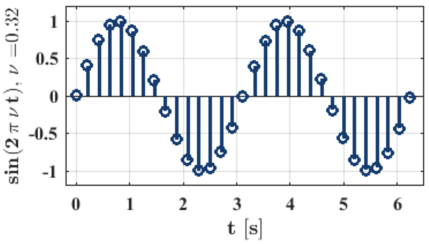
$$TV(\mathbf{f}) = \sum_{n=0}^{N-1} \mid f_n - f_{n-1} \; mod\, N \mid \tag{3.17}$$

Hence, such time series can be modeled by a directed cyclic graph as shown in Figure 3.1. The orientation of the edges indicates the time evolution from the past to the future, while the edge that links the first and the last node (v_{N-1}, v_0) captures the periodicity of the signal $(f_N = f_0)$. Consequently, the $N \times N$ adjacency matrix of such graph (Figure 3.1) is non-symmetric. More precisely, it is a cyclic permutation matrix having the form:

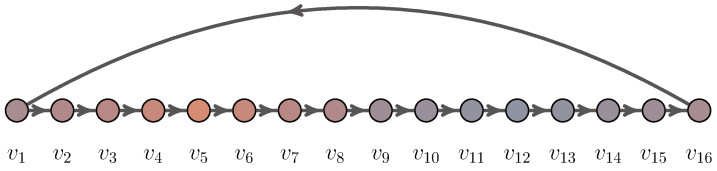
$$\mathbf{A} = \mathbf{C} = \begin{pmatrix} 0 & & & 1 \\ 1 & \ddots & & \\ & \ddots & \ddots & \\ & & 1 & 0 \end{pmatrix}, \tag{3.18}$$

Using the cyclic permutation matrix (3.18), the total variation (3.17) can be written as:

$$TV(\mathbf{f}) = \| \mathbf{f} - \mathbf{C}\,\mathbf{f} \|_1 . \tag{3.19}$$



$sin(2\,\pi\,\nu\,t)$, $\nu = 0.32$



Directed cyclic graph representing the sampled sine wave:
$G(V, \mathbf{A}, \mathbf{f}) : V \to \mathbb{R}$ such that: $v_n \mapsto f_n = sin(2\,\pi\,\nu\,[n])$

Figure 3.1: A discrete periodic series satisfying $f_n = f_n mod\, N$ ($N = 16$) can be modelized using a directed graph. The node values are the series samples, the edges represent the time flow and the edge (v_N, v_1) illustrates the periodicity of the series.

The intuition behind the graph total variation (3.16) comes from discrete mathematical models. While the discretized derivative of a signal $\mathbf{f}$ is defined as a simple local difference $\nabla_n(\mathbf{f}) = f_n - f_{n-1}$, the derivative (gradient) of a graph signal $G(V, \mathbf{A}, \mathbf{f})$ at a particular node v_n is given by:

$$\nabla_n(\mathbf{f}) = \frac{d\mathbf{f}}{dv_n} = f_n - \sum_{m \in \mathcal{N}_n} \mathbf{A}_{n,m}\, f_m. \tag{3.20}$$

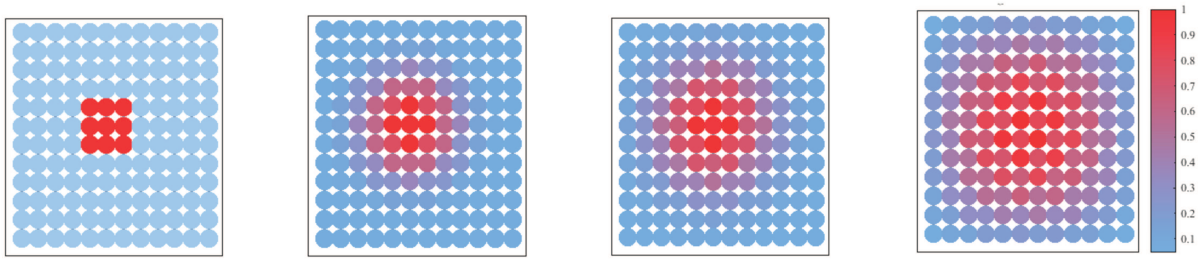
The magnitude of the gradient $\mid \nabla_n(\mathbf{f}) \mid$ at node v_n corresponds to the variation of the signal $\mathbf{f}$ in that node, and the total variation is the sum of all variations over the totality of nodes Mallat, 2008 D.I. Shuman et al., 2013. Furthermore, as a generalization, the discrete $p-$Dirichlet form of the total variation can be considered:

$$S_p(\mathbf{s}) = \frac{1}{p} \sum_{n=0}^{N-1} \mid \nabla_n(\mathbf{s}) \mid^p, \tag{3.21}$$

where, for the value $p = 1$, the equation (3.21) corresponds to the initial one (3.16):

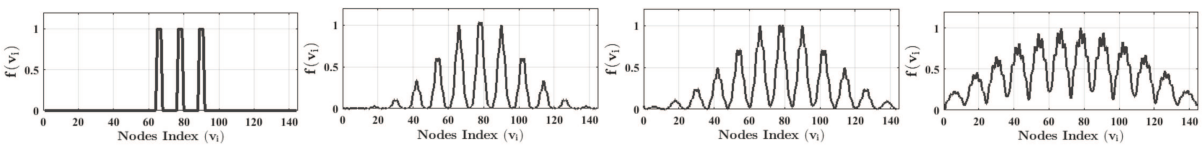
$$S_1(\mathbf{f}) = \sum_{n=0}^{N-1} \mid \nabla_n(\mathbf{f}) \mid \tag{3.22}$$
$$= \sum_{n=0}^{N-1} \left| f_n - \sum_{m \in \mathcal{N}_n} \mathbf{A}_{n,m}\, f_m \right| = \| \mathbf{f} - \mathbf{A}\,\mathbf{f} \|_1 \,.$$

In Discrete Graph Signal Processing $\mathrm{DSP_G}$ theory, the expression of the total variation (3.22) could be generalized to all graph signal by using any shifting operator that is valid for such signals, hence, more shifting operators are defined in the litterature (see B. Girault et al., 2015). Figures 3.2 and 3.3 illustrate the sensitivity of the total variation (TV_G) to oscillations and the fluctuations of the signal upon the structure. We choosed the heat propagation model to generate a graph signal that behaves in an ondulatory way accross the structure, thus making it possible to evaluate the response of the total variation to this change in behaviour. We notice that the form of the 1D-wave showed in Figure 3.3 depends on the order initially chosen for nodes. In this case of grid graph, we ordered nodes from left to right beginning from above to below.



$t = 0\,s,\ TV_G = 27$ $t = 0.5\,s,\ TV_G = 68.95$ $t = 0.9\,s,\ TV_G = 99.58$ $t = 7\,s,\ TV_G = 179.14$

Figure 3.2: A graph signal $\mathbf{f}$ associated to a 12×12 grid structure, in which it diffuses according to the heat propagation model: $\mathbf{f}(t+1) = exp\,(-\tau\, t\, \mathbf{L})\, \mathbf{f}(t)$, $\mathbf{L}$ being its laplacian matrix. During diffusion, the signal's values oscillate more and more, leading a higher total variation values.



$t = 0\,s$ $t = 0.5\,s$ $t = 0.9\,s$ $t = 7\,s$

Figure 3.3: Diffused graph signals indexed by nodes, corresponding to diffusion states showed in Figure 3.2. From left to right, we observe that the more heat propagates wider across the network, the more oscillations occurs.

3.2.4 $TV-$based Signals Denoising

Among the various applications of total variation, we invoke the one concerning the denoising of signals, whether they are uni- or bi-dimensional. The $TV-$based noise reduction approach has the ability of restoring prominent and sharp edges characterizing the original signal L.I. Rudin et al., 1992. The denoising process consists of looking for a solution to an optimization problem, wherein total variation is introduced as a regularization term. Therefore, the $TV-$filter is obtained after minimizing a certain objective function which includes a measure of distortion between the input signal and its estimated version. In the paper L.I. Rudin et al., 1992, total variation was introduced for the first time to regularize a filtering algorithm dedicated mainly for piecewise smooth images denoising. Its performance is due probably to the good tradeoff that $TV-$regularizer can establish between the description level of that particular class of piecewise smooth signals and the computational complexity of the algorithm. The goal of $TV-$regularization is to bound as possible the signal's amplitude variations, without penalizing extensively discontinuities, moreover, the introduction of the $TV-$regularization term in the objective function does not affect its convexity J. M. Bioucas-Dias et al., 2006.

Mathematically the $TV-$denoising algorithm is formulated as an optimization problem defined in a continuous domain L.I. Rudin et al., 1992:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \Big\{ \int_{\Omega} (\mathbf{y}(t) - \mathbf{x}(t))^2 dt + \lambda\, TV(\mathbf{x}) \Big\} \tag{3.23}$$

$TV(\mathbf{x})$ is the regularization term corresponding to the continuous 1D total variation of $\mathbf{x}$, defined by formula (3.1). The minimization operation spans the set Ω of all square integrable functions. Otherwise, the discrete version of the approach assumes that the signal $y(n)$ is distorted by an additive white Gaussian noise $w(n)$ such that $y(n) = x(n) + w(n)$ and N is the number of samples. The goal is to approximate $y(n)$ with a piecewise constant signal $x(n)$ by solving the following discrete domain optimization problem:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \Big\{ \frac{1}{2} \sum_{n=0}^{N-1} \mid y(n) - x(n) \mid^2 dt + \lambda \sum_{n=1}^{N-1} \mid x(n) - x(n-1) \mid \Big\}. \tag{3.24}$$

The regularization λ parameter controls the degree of smoothing to be performed on $x(n)$ to further minimize its quadratic error via compared to $y(n)$. High values of λ give more importance to the total variation of $x(n)$, therefore, to the details level of the output signal. According to the equation (3.7), the first order derivative of an $N-$sample signal $\mathbf{x}$ is given by $\boldsymbol{\nabla}\mathbf{x}$, while its total variation is $\parallel \boldsymbol{\nabla}\mathbf{x} \parallel_1$. Using this notation, the $TV-$based denoising problem can be expressed in a compact form as:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \Big\{ \frac{1}{2} \parallel \mathbf{y} - \mathbf{x} \parallel_2^2 + \lambda \parallel \boldsymbol{\nabla}\mathbf{x} \parallel_1 \Big\}. \tag{3.25}$$

To solve the optimization problem (3.25), an efficient algorithm was used by M. A. T. Figueiredo et al., 2006, called the Majorization-Minimization (MM) algorithm, which consists on the solving of simpler optimization problems instead the initial problem difficult to solve. Instead of minimizing a complex cost function $F(\mathbf{x})$, the MM algorithm minimizes a sequence of easier problems described by the objective functions $F'_k(\mathbf{x})\,/k \in \mathbb{N}$. The MM algorithm yields a sequence of solutions $\mathbf{x}_k$, each being issued from the minimization of the function $G_{k-1}(\mathbf{x})$. The functions $G_k(\mathbf{x})$ are chosen so that they are easy to minimize and they approximate at best $F(\mathbf{x})$. Formally, $G_k(\mathbf{x})$ must upper-bound $F(\mathbf{x})$ such that $G_k(\mathbf{x}) > F(\mathbf{x})/\ \forall \mathbf{x}$, and that it meets $F(\mathbf{x})$ at point $\mathbf{x} = \mathbf{x}_k$. Moreover, the functions $G_k(\mathbf{x})$ should be convex, giving the set of solutions $\mathbf{x}_{k+1}$ determined by MM algorithm. Therefore, when $F(\mathbf{x})$ is convex too, the obtained sequence of solutions $\mathbf{x}_k$ converges to the global solution (minimizer) of $F(\mathbf{x})$, (see M. A. T. Figueiredo et al., 2006). One way to solve the $TV-$based denoising problem (3.25) by the MM algorithm is to use a quadratic function to majorize the total variation $\parallel \boldsymbol{\nabla}\mathbf{x} \parallel_1$, so as to also majorize the total objective function $F(\mathbf{x})$, which can be in turn to be minimized by solving a system of linear equations. According to this approach, the sequence of consecutive solutions $\mathbf{x}_k$ are then linked by the following update function:

$$\mathbf{x}_{k+1} = \mathbf{y} - \boldsymbol{\nabla}^T \Big(\frac{1}{\lambda}\, diag(\mid \boldsymbol{\nabla}\mathbf{x}_k \mid) + \boldsymbol{\nabla}\boldsymbol{\nabla}^T \Big)^{-1} \boldsymbol{\nabla}\mathbf{y}. \tag{3.26}$$

where $\boldsymbol{\nabla}$ is the first order derivation operator given by the matrix (3.8). See Selesnick, 2012 for more details about this result.

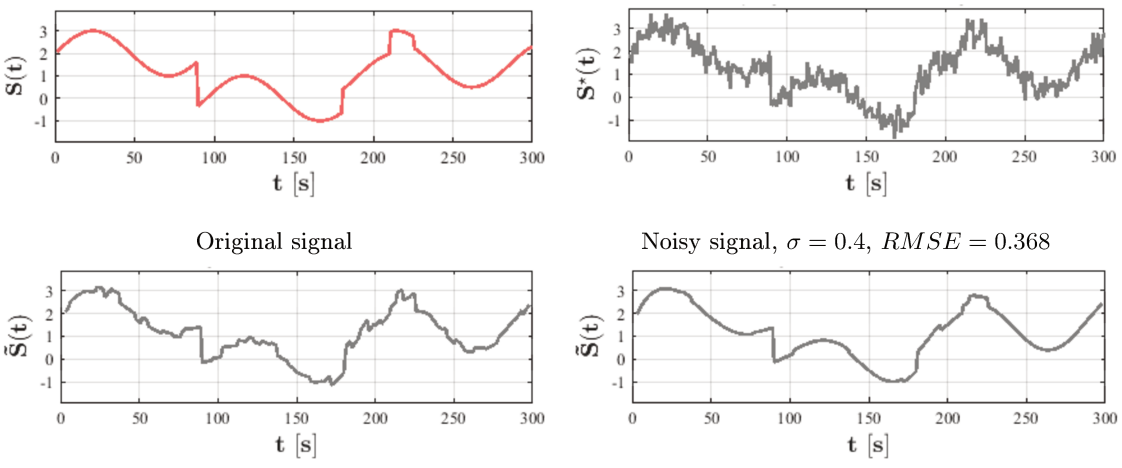


Figure 3.4: Denoising example of a 1D-signal based on TV and on the (MM) algorithm (Selesnick, 2012). The signal is destorted by an additive Gaussian White Noise $\sim \mathcal{N}(0, \sigma)$. λ si the regularization parameter of equation (3.26). The denoising performance is measured by the Root Mean Square Error ($RMSE$).

The $TV-$based denoising example of Figure 3.4 shows that the algorithm is able to recover important

edges and features of the signal, as well that the total variation via the parameter λ controls pretty well the degree of details allowed in the output signal. However, it is clear that the algorithm is adapted for piecewise constant smooth signals, because it does not tolerate fast variations of the envelope.

3.2.5 $TV-$based Graph Frequencies Ordering

In classical discrete signal processing theory, the concept of low and high frequencies is well defined and has an intuitive interpretation, since these signals including time series and images are analysed in Fourier's sense by decomposing them into a certain weighted sum of sinusoidal functions oscillating in different ways Mallat, 2008. The oscillation rate of each component gives an indication about its energy, therefore a physical interpretation about the concept of "high" and "low" frequencies. The faster the oscillation is, the higher its corresponding frequency is. However, this observation is valid only for signals uniformally sampled, that oscillate in a regular lattice, unlike graph signals that are supported on irregular and complex structures.

A graph signal is commonly expressed in terms of graph Fourier functions, corresponding to the Jordan eigen-decomposition of its adjacency matrix $\mathbf{A}$. Where, distinct eigenvalues of $\mathbf{A}$ are interpreted as the *graph frequencies* forming the *spectrum*, and the $m-$th Jordan eigenvector $\mathbf{v}_m$ is interpreted as a *frequency component* associated to the $m-$th frequency λ_m of the graph. Since the Jordan decomposition yields a generalized eigenvectors, a particular *graph frequency* could be associated to several *frequency components*. Hence, the ordering of graph frequencies is not trivial, it needs an objective criteria equivalent to the oscillation rate in classical DSP, to rank them appropriately. One possible criteria is total variation, proposed by A. Sandryhaila and J.M.F. Moura, 2014. They order the graph frequencies according to the oscillatory rate of the corresponding spectral components upon the underlying graph structure. They use the graph total variation function defined in (3.16) to measure the changes of signal values from individual nodes towards their neighbors, leading to a definition of "low" and "high" frequencies on graphs. Furthermore, they show that the resultant order is unique for graphs having real valued spectra (undirected graphs).

As mentioned previously, the Jordan eigen-decomposition of adjacency matrix $\mathbf{A}$ yields a Fourier basis for the corresponding graph signal $G(V, \mathbf{A}, \mathbf{f})$. Let λ be an eigenvalue of $\mathbf{A}$, and let $\mathbf{v} = \mathbf{v}_0, \mathbf{v}_1, ..., \mathbf{v}_R - 1$ be a Jordan chain of generalized eigenvectors associated to λ. Then, the following indicator function

$$i_r = \begin{cases} 0, & \text{if } r = 0 \\ 1, & \text{if } 1 \leq r \leq R, \end{cases} \tag{3.27}$$

points out whether $\mathbf{v}_r$ is a generalized or a specific eigenvector of $\mathbf{A}$. Thus, the generalized eigen-problem associated to the matrix $\mathbf{A}$ is then formulated as:

$$\mathbf{A}\mathbf{v}_r = \lambda \mathbf{v}_r + i_r \mathbf{v}_{r-1}. \tag{3.28}$$

The Fourier analysis of the graph signal $G(V, \mathbf{A}, \mathbf{f})$ consists on the projection of $\mathbf{f}$ over the Jordan eigenbasis. Hence, each frequency component $\mathbf{v}_r$ is seen as a signal supported on the same structure than that of $\mathbf{f}$, and thus, the total variation of the generalized eigenvector $\mathbf{v}_r$ on the structure of G can be calculated using equation (3.16):

$$\begin{aligned} TV_G(\mathbf{v}_r) &= \| \mathbf{v}_r - \mathbf{A}^{norm}\mathbf{v}_r \|_1 \\ &= \left\| \mathbf{v}_r - \frac{1}{\mid \lambda_{max} \mid} \mathbf{A}\mathbf{v}_r \right\|_1 \end{aligned} \tag{3.29}$$

where $\mathbf{A}^{norm}$ is a normalized version of the adjacency matrix $\mathbf{A}$, scaled up by the highest eigenvalue to guarantee the non scaling of the shifted resultant signal ($\mathbf{A}/ \mid \lambda_{max} \mid$). By substituting equation (3.28) in (3.29), the total variation of $\mathbf{v}_r$ becomes:

$$TV_G(\mathbf{v}_r) = \left\| \mathbf{v}_r - \frac{\lambda}{\mid \lambda_{max} \mid} \mathbf{v}_r - \frac{i_r}{\mid \lambda_{max} \mid} \mathbf{v}_{r-1} \right\|_1. \tag{3.30}$$

If $\mathbf{v}_r$ is a specific eigenvector of $\mathbf{A}$, with $r = 0$, $\mathbf{v} = \mathbf{v}_r = \mathbf{v}_0$ and $i_0 = 0$, then the total variation (3.30) corresponding to the eigenvector $\mathbf{v}$ is written as:

$$TV_G(\mathbf{v}) = \left| 1 - \frac{\lambda}{\mid \lambda_{max} \mid} \right| \| \mathbf{v} \|_1 \, . \tag{3.31}$$

The expression (3.31) indicates that the total variation of a frequency component indexed by a specific eigenvector of $\mathbf{A}$, is determied straightforwardly by its corresponding eigenvalue λ. This leads to the fact that all specific eigenvectors associated to the same frequency exhibit identical total variation. When the eigenvectors are generalized ones, then equation (3.29) is used to measure their variation towards other frequency components. After normalization, the norm of $\mathbf{v}$ becomes unity ($\| \mathbf{v} \|_1 = 1$) and its total

variation has an upper bound of two as stated by the following inequality:

$$TV_G(\mathbf{v}) = \left|1 - \frac{\lambda}{|\ \lambda_{max}\ |}\right| \leq 1 + \left|\frac{\lambda}{|\ \lambda_{max}\ |}\right| \leq 2. \tag{3.32}$$

Therefore, the total variation of a normalized specific eigenvector does not exceed 2 and obviously does not go under 0. As result, the total variation defined by equations (3.29) and (3.31), and calculated for every frequency component of G, is an appropriate tool to order frequencies in ascending way. Thus, "low" frequencies are those that have a slow variation, while fast variation corresponds to "high" frequencies. Moreover, the total variation (3.31) is valid uniquely for graphs that have diagonalizable adjacency matrix and only have specific eigenvectors, otherwise, equation (3.29) can be used for graphs having non-diagonalizable adjacency matrices with generalized eigenvectors.

Theorem 3.2.1. *(A. Sandryhaila and J.M.F. Moura, 2014) Consider two distinct real eigenvalues $\lambda_m, \lambda_n \in \mathbb{R}$ of the adjacency matrix $\mathbf{A}$ with corresponding eigenvectors $\mathbf{v}_m$ and $\mathbf{v}_n$. If the eigenvalues are ordered as $\lambda_m < \lambda_n$, then the total variation of their eigenvectors satisfy*

$$TV_G(\mathbf{v}_m) > TV_G(\mathbf{v}_n),$$

it follows from the theorem 3.2.1 that if a graph signal possess a spectrum of eigenvalues ordered in this way: $\lambda_0 > \lambda_1 > \cdots > \lambda_{M-1}$, then the eigenvalue λ_0 is the lowest possible oscillation of the signal in the given graph and $\lambda_M - 1$ is the highest one. In addition, that ordering is unique for each individual graph. Otherwise, for graph that have complex eigenvalues, the ordering is given by the following theorem:

Theorem 3.2.2. *(A. Sandryhaila and J.M.F. Moura, 2014) Consider two distinct complex eigenvalues $\lambda_m, \lambda_n \in \mathbb{C}$ associated to the adjacency matrix $\mathbf{A}$. Let $\mathbf{v}_m$ and $\mathbf{v}_n$ be thier corresponding eigenvectors. Then the total variation of these eigenvectors satisfy*

$$TV_G(\mathbf{v}_m) < TV_G(\mathbf{v}_n),$$

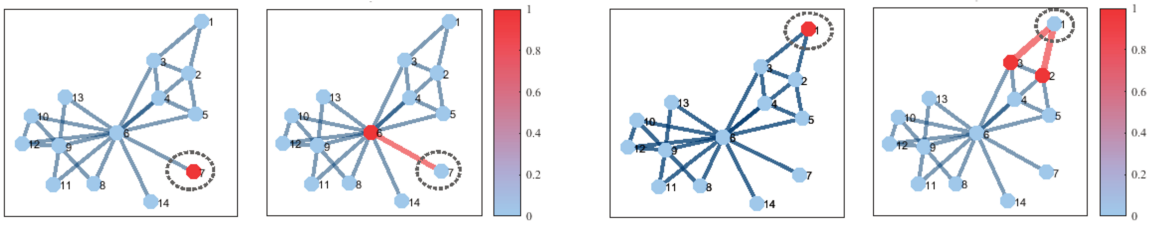
if the eigenvalue λ_m is located closer to the value $|\ \lambda_{max}\ |$ on the complex plane then the eigenvalue λ_n.

The theorem 3.2.2 states that for complex frequencies, the order is done according to their relative distance from the point $|\ \lambda_{max}\ |$. Unlike the order of real frequencies, this order is not unique, because the distinction between frequency components associated to distinct frequencies is not guaranteed, since they may have the same total variation. Concretely, all complex frequencies lying on a circle of radius σ centered at the point $|\ \lambda_{max}\ |$ on the complex plane share the same total variation $\sigma / |\ \lambda_{max}\ |$.

3.2.6 Total Variation as Similarity Measure

Graph total variation TV_G (3.16) characterizes the oscillations of the signal values upon the structure of the graph, where high variations illustrate the presence of high frequency components in the graph A. Sandryhaila and J.M.F. Moura, 2014. These frequencies depend directly on both signal values on each node and their corresponding degrees. A high local oscillation corresponds to abrupt variation of the signal value with respect to its neighbours. The more the neighbourhood is large, the more important the oscillation is. This observation is illustrated in Figure 3.6, where the signal associated to the structure of the graph is a one single value positionned in a particular node every time and equals zero in other nodes.The total variation of this signal is proportional to the difference between their original values and shifted ones. Hence, the TV_G increases when the neighborhood of the initial node gets wider. The last subfigure of Figure 3.6 shows that an oscillation in the graph its a local property and depends on the neighborhood where it occurs. We mean by δ_{ij}, the kronecker function associating one to the node having index $i = j$ and a zero to other nodes:

$$\delta_{ij} = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases}$$

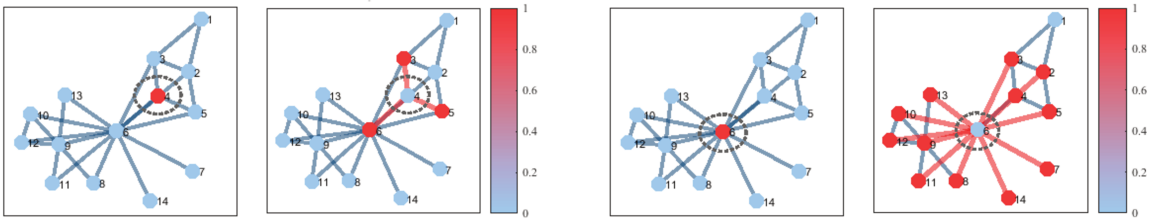


(a) $\mathbf{f} = \delta_{i7}$ (b) $\tilde{\mathbf{f}} = \delta_{i6}$, $TV_G = 2$ (a) $\mathbf{f} = \delta_{i1}$ (b) $\tilde{\mathbf{f}} = \delta_{i2}+\delta_{i3}$, $TV_G = 3$

Given the structure of the graph and the signal values on his nodes, the TV_G can be considered as a good and informative descriptor for graph discrimination. More precisely, TV_G is well suited for graph signals comparison. Consider a set of two graph-signals $\{G_i(V_i, \mathbf{A}_i, \mathbf{f}_i), G_j(V_j, \mathbf{A}_j, \mathbf{f}_j)\} \subseteq \mathcal{G}$ where $\mathbf{A}_i$ and $\mathbf{f}_i$ are respectively the adjacency matrix and the signal supported on the graph's nodes, with $\mathcal{G}$ being the domain of graph signals. We use the relation (3.16) to write the mean total variation of the graph signal indexed by $k \in \{1, 2\}$

$$\widetilde{\mathrm{TV}}_G(\mathbf{f}_k) = \frac{\mathrm{TV}_G(\mathbf{f}_k)}{n_k} = \frac{|| \mathbf{f}_k - \mathbf{A}_k \mathbf{f}_k||_1}{n_k}, \tag{3.33}$$

where n_k is the number of the nodes in the graph G_k. We use the formula (3.33) to measure the similarity between the graph signals, we suppose that two graphs go closer to each other, probably when their total variation tends to be the same. In other words, a two graph signals oscillate the same manner often when



(a) $\mathbf{f} = \delta_{i4}$ (b) $\tilde{\mathbf{f}} = \delta_{i3} + \delta_{i5} + \delta_{i6}$ $TV_G = 4$ (a) $\mathbf{f} = \delta_{i6}$ (b) $\tilde{\mathbf{f}} = \delta_{i2} + \delta_{i3} + \delta_{i4} + \delta_{i5} + \delta_{i7} + \delta_{i8} + \delta_{i9} + \delta_{i10} + \delta_{i11} + \delta_{i12} + \delta_{i13} + \delta_{i14},\ TV_G = 13$

Figure 3.6: The total variation increases when the oscillation from a node towards its neighbors is high in amplitude and range. The left hand figures "(a)" show initial state of the graph signal $\mathbf{f}$ composed of a single peak located at one of the nodes $\{v_7, v_1, v_4, v_6\}$. The right hand figures "(b)" show the shifted version of the signal according to equation (3.15). The TV_G measures the variation between $\mathbf{f}$ and $\tilde{\mathbf{f}}$, therefore, it increases when the neighborhood is large.

they share the same structural properties. Hence, as a distance between $G_i(\mathbf{f}_i, \mathbf{A}_i)$ and $G_j(\mathbf{f}_j, \mathbf{A}_j)$, we consider the simple difference between their respective mean total variations defined in (3.33), which we denote by

$$\mathrm{TVG}(G_i, G_j) = \mid \widetilde{\mathrm{TV}}_G(\mathbf{f}_i) - \widetilde{\mathrm{TV}}_G(\mathbf{f}_j) \mid . \tag{3.34}$$

Thus, small TVG value means high similarity. More, among the advantages of this distance (3.34), it compares graph signals while involving both, the signal's values on nodes and the structural information of the graph included in the adjacency matrix $\mathbf{A}$. Thereafter, this distance is used to compare molecular data for classification.

3.3 Graph Energy Information

3.3.1 Energy of Signals

Signals may represent a broad variety of phenomena. In many applications, signals are directly related to physical quantities capturing energy and power in a physical system. The concept of signal energy is of primary importance in the design of continuous and discrete domain systems. In the real world, we always transmit signals with finite energy $0 < E_x < +\infty$ where E_x is the total energy signal that represents the amount of energy contained in signal $x(t)$. Engineers refer to such signals as having finite energy, although E_x is not necessarily the *physical* energy of the signal $x(t)$. The measure E_x, analogue

to the squared length of multidimensional vectors, is proportional to energy (physical quantity) when $x(t)$ is a velocity, current, voltage or pressure. The total energy of the signal should be independent of the method used to calculate it. Alternatively, in frequency domain, the energy spectral density is given by $\mid X(f) \mid^2$, where $X(f)$ is the Fourier transform of $x(t)$. The quantity $\mid X(f) \mid^2$ expresses the energy contribution of a given frequency to E_x. The estimation of E_x is an important part of physics and signal processing and can be equivalently computed in either the time or frequency domain. For a continuous real time signal $x(t)$, the energy is expressed as:

$$E_x = \langle x(t), x^\star(t) \rangle = \int_{-\infty}^{+\infty} \mid x(t) \mid^2 dt = \int_{-\infty}^{+\infty} \mid X(f) \mid^2 df, \tag{3.35}$$

The Parserval's or Rayleigh's theorem, which connects the total energy in time and frequency domains (conservation of energy), states that the total energy contained in the signal $x(t)$ across all time is equal to the total energy of the signal's Fourier transform $X(f)$ over all of its frequency components. The quantity E_x can also be calculated from wavelet coefficients of $x(t)$ or its time-frequency representation and particularly for the study of nonstationary signals Boashash, 2015. This estimation of E_x is applied across different domains of signal processing such as communications, speech processing, Radar, Sonar or biomedical engineering. However, it is not easy to see from the identity (3.35) how signal frequencies affect the energy measures. More precisely, neither the time domain description of $x(t)$ and its energy density $\mid x(t) \mid^2$, nor the frequency domain description, $X(f)$, and the spectral energy density $\mid X(f) \mid^2$, reveal explicitly the frequency spectrum at a particular time or the time at which a particular frequency component occurs. Furthermore, the estimate E_x does not ever refer to what the signal $x(t)$ physically represents and, generally, it is unclear how this estimate relates to the physical energy in the system or the process that produced $x(t)$ J.F. Kaiser, 1990,Fang and L.E. Atlas, 1995,A.O. Boudraa and Salzenstein, 2015,Cohen, 1994.

As pointed out in Cohen, 1994, signal analysis has been extended to many diverse types data including economical and sociological nature. Thus it is certainly not obvious that in those cases we can meaningfully talk about the instantaneous energy per unit time and take $\mid x(t) \mid^2$ to be its value. For example, the total energy of the source system modeled as a mass suspended by a spring of a constant stiffness required to produce a simple undamped harmonic oscillation is calculated by the sum of the kinetic energy of the mass and the potential energy in the spring. By studying the second order differential associated to this harmonic oscillator, it is easy to show that a simple sinusoidal varies as a function of both amplitude and oscillation frequency of the signal, which is quite different from simple squaring of the signal magnitude Moshinsky and Y.F. Smirnov, 1996. It is this source modeling that is used for

characterizing signals by amplitude and frequency. In their work on non-linear speech modelling, Herbert and Shushan Teager pointed out the dominance of modulation as a process in the speech production Teager and Teager, 1983,Teager and Teager, 1990. Based on the Teager's work, Kaiser proposed an energy measure that includes both the amplitude and the frequency of the signal J.F. Kaiser, 1990. This measure is often referred to as the Teager-Kaiser (TK) energy operator. Using the conventional view of the energy, it is easy to see that two tones at 10Hz and 1000Hz of unit-amplitude have the same energy. However, the energy required to produce the signal of 1000Hz is much greater than that for the 10Hz signal J.F. Kaiser, 1990. Using TK definition of energy, the two tones show different energy. This definition highlights the concept of signal energy from the point of view of the generation of the signal and emphasizes the importance of analyzing signals from the energy aspect of the system needed to produce them. In its continuous form, TK energy operator, noted Ψ_C, when operating on continuous-time signal $x(t)$ is given by:

$$\begin{aligned}\Psi_C[x(t)] &= \left(\frac{dx(t)}{dt}\right)^2 - x(t)\left(\frac{d^2x(t)}{dt^2}\right) \\ &= \dot{x}^2(t) - x(t)\,\ddot{x}(t),\end{aligned} \tag{3.36}$$

where $\dot{x}(t)$ and $\ddot{x}(t)$ are the first and the second derivative of $x(t)$ with respect to time t respectively. When Ψ_C is applied to signals generated by a simple harmonic oscillator (mass-spring oscillator of constant stiffness), it can track the oscillator's energy (per half unit mass) which is proportional to the squared product of the oscillation amplitude and frequency. For narrowband signal $x(t)$ and under realistic conditions, $\Psi_C[x(t)]$ approximately estimates the energy of the source producing the oscillation $x(t)$. The matrix framework of the operator has been introduced in [13] by interpreting it as the determinant of a Toeplitz matrix containing the signal and its derivatives:

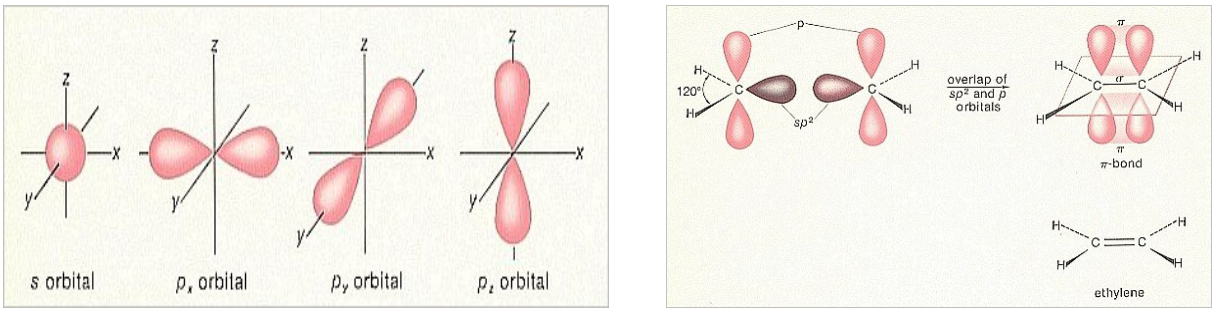
$$\Psi_C[x(t)] = \begin{bmatrix} \dot{x}(t) & x(t) \\ \ddot{x}(t) & \dot{x}(t) \end{bmatrix}. \tag{3.37}$$

The determinant is time-invariant for a signal with constant frequency [13]. Using this matrix framework, the output of the TK operator is interpreted as the measured energy corresponding to the square of the eigenvalue of its underlying energy matrix, a notion analogous to that seen in quantum mechanics [15].

3.3.2 On the Energy of Graphs

In quantum mechanics, the Schrödinger equation describes the changes over time of a physical system in which quantum effects occur. The time-independent version of the equation, written as $H\boldsymbol{\Psi} = E\boldsymbol{\Psi}$,

predicts the energy of stationary states (called "orbitals", as in atomic orbitals or molecular orbitals) when the Hamiltonian operator H of the system is time-invariant. The stationary state is represented by the wave function $\mathbf{\Psi}$, having the energy E. The solution of Schrödinger's equation is nothing else then the eigenpair $(E, \mathbf{\Psi})$. On the other hand, Hückel Molecular Orbital (HMO) theory, I. Gutman and O.E. Polansky, 2012, is a field of theoretical chemistry where approximations of the π-electron energies are established for single conjugated hydrocarbon molecules, inwhich the Hamiltonian operator takes the form $H = \alpha I + \beta \mathbf{A}$, where $\mathbf{A}$ is the adjacency matrix of the graph associated to the molecule structure and α, β are real constants. Hence, it results that the solution to the eigenvalue problem associated to H can be reduced the one of $\mathbf{A}$. The eigenvectors of $H\,/\,(\mathbf{A})$ describe the orbitals of the molecule, in which $0, 1,$ or 2 π-electrons could exist. Often, $2p$ orbitals contain a pair of π-electrons with a positive energy $E > 0$, and no π-electrons otherwise $(E < 0)$. In chemistry, a π-electron occupies an orbital which forms a π-bond in the molecule. Such bond is created between two atoms by overlapping orbitals having a secondary quantum number superior or equal to one (orbitals p and d). The overlap is lateral, and the two lobes of the concerned two orbitals are parallel. Unlike in the case of σ-bonds where the lobes of the two atomic orbitals point towards each other. Each of these atomic orbitals has zero electron density in the connection axis. Figure 3.7 illustrates the form of some atomic orbitals and their overlapping when forming a π-bond/σ-bond in ethylene molecule.



(a) (b)

Figure 3.7: In an atom, electrons orbit the nucleus, while occupying a well defined orbits with quantified levels of energy. In (a), the s and p type orbitals are illustrated. While (b) shows the overlap of atomic orbitals to form the ethylene molecule. The lateral overlap of parallel p orbitales forms a π-bound (connexion), while a frontal overlap forms a σ-bound.

Given that the energy E corresponds to the eigenvalues of the eigenproblem $H\mathbf{\Psi} = E\mathbf{\Psi}$, where $H = \alpha \mathbf{I} + \beta \mathbf{A}$, and that the trace of adjacency matrix is zero $(Trace(\mathbf{A}) = 0)$ (eigenvalues cancel each other), then the total energy of π-electron could be quantified by:

$$E_{\mathbf{A}}(G) = \sum_{i=1}^{n} \mid \mu_i \mid, \tag{3.38}$$

where n is the number of carbon atoms in the molecule (graph nodes), $\mu_i \approx E_i$ is the i^{th} eigenvalue of adjacency matrix $\mathbf{A}$ of the graph G associated to the carbon structure of the molecule. Given the energy $E_{\mathbf{A}}(G)$ of G, the π-electron energy is defined by $E_\pi = \alpha n_e + \beta E_{\mathbf{A}}(G)$, where α and β are the HMO parameters, n_e is the number of π-electrons. Moreover, Gutman argues that by means of E_π, it is possible to calculate accurately the values of thermodynamic functions of conjugated hydrocarbons such as enthalpy of combustion. The wide adoption of E_π in physico-chemical community is explained by the fact that L. J. Schaad and B. A. Hess, 1972 showed that also the σ-electron energy is proportional to $E_{\mathbf{A}}(G)$ and not only π-electron energy. The quantity (3.38) is then generalized to any arbitrary connected graph as the sum of its eigenvalues in absolute value Gutman, 2001, I. Gutman et al., 2007. In spectral graph theory, each graph G having m edges and n nodes fullfils the following relations:

$$\sum_{i=1}^{n} \mu_i = 0, \qquad \sum_{i=1}^{n} {\mu_i}^2 = 2m, \qquad \sum_{i=1}^{n} \lambda_i = 2m, \qquad \sum_{i=1}^{n} {\lambda_i}^2 = 2m + \sum_{v_i \in V} (d(v_i))^2, \tag{3.39}$$

where $\mu_1 \geqslant \mu_2 \geqslant \cdots \geqslant \mu_n$ are the eigenvalues of adjacency matrix ($\mathbf{A}$), $\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_n$ are those of laplacian matrix ($\mathbf{L}$) and $d(v_i)$ is the degree of the node v_i. Moreover, if the graph is segmented to k components ($k \geqslant 1$), then $\lambda_{n-i} = 0$ for $i \in \{0, 1, ..., k-1\}$ and $\lambda_{n-k} > 0$. Using these relation, the following properties about $E_{\mathbf{A}}$ hold:

- $E_{\mathbf{A}}(G) \geqslant 0$, equality is obtained if and only if the graph is empty, such that $m = 0$.
- If G is the union of the disconnected components G_1 and G_2, then $E_{\mathbf{A}}(G) = E_{\mathbf{A}}(G_1) + E_{\mathbf{A}}(G_2)$.
- If G contains a non-empty component G_1 and all others are isolated nodes, then $E_{\mathbf{A}}(G) = E_{\mathbf{A}}(G1)$.

This spectrum-based graph invariant has been largely studied in both chemical and mathematical literature. Some other interesting algebraic properties are investigated so far, defined on $E_{\mathbf{A}}$-like quantities. Among them, there is the one associated to the laplacian matrix ($\mathbf{L}$). Instead of formula (3.38), a $E_{\mathbf{A}}$-like quantity defined in terms of laplacian matrix eigenvalues and preserving the main features of $E_{\mathbf{A}}$ has been proposed by I. Gutman and B. Zhou, 2006:

$$E_{\mathbf{L}}(G) = \sum_{i=1}^{n} \left| \lambda_i - \frac{2m}{n} \right|, \tag{3.40}$$

where λ_i are the eigenvalues of the laplacian matrix of G and $2m/n$ is its average node degree. Since the sum of the nodes degrees in a graph is $2m$ and equals to the trace of $\mathbf{L}$, the quantity $2m/n$ is nothing more than the mean value of the $\mathbf{L}$'s eigenvalues. Otherwise, by introducing the auxiliary eigenvalues $\gamma_i = \lambda_i - \frac{2m}{n}$ for $i \in \{1, 2, ..., n\}$ and by analogy with relations (3.39), the following expressions can be

deduced:

$$\sum_{i=1}^{n}\gamma_i = 0, \qquad \sum_{i=1}^{n}\gamma_i^2 = 2M \ \text{ where } \ M = m + \frac{1}{2}\sum_{i=1}^{n}\Big(d(v_i) - \frac{2m}{n}\Big)^2. \tag{3.41}$$

Based on alternative graph representation matrices, other graph energies are proposed in the litterature. Instead of laplacian matrix $\mathbf{L}$, a signless laplacian matrix is defined by $\mid \mathbf{L} \mid = \mathbf{D} + \mathbf{A}$, with $\nu_1 \geqslant \nu_2 \geqslant \cdots \geqslant \nu_n$ its eigenvalues. Then similiraly to (3.40), the signless laplacian energy of G is given by:

$$E_{|\mathbf{L}|}(G) = \sum_{i=1}^{n}\left|\nu_i - \frac{2m}{n}\right|. \tag{3.42}$$

The distance matrix associated to a graph is the symmetric matrix $\mathbf{D^G}$, that have as entry $\mathbf{D^G}_{ij}$ the length of the shortest path between the nodes v_i and v_j. In case where the node v_k is isolated, then the entries $\mathbf{D^G}_{kj}$ for all j are set to be zero instead of infinity (∞). Considering a simple graph (without loops), then the diagonal entries of $\mathbf{D^G}$ are all zero, and the mean of eigenvalues is also zero. Therefore, the energy of the distance matrix is similar to the expression of $E_{\mathbf{A}}$ in (3.38):

$$E_{\mathbf{D^G}} = \sum_{i=1}^{n} \mid \eta_i \mid, \tag{3.43}$$

with $\eta_1 \geqslant \eta_2 \geqslant \cdots \geqslant \eta_n$ are the eigenvalues of $\mathbf{D^G}$.

Overall, the graph energies presented above are often considered as special cases of matrix norms (e.g. Ky Fan or Schatten norms), defined over the space of $n \times n$ complex matrices $M_n(\mathbb{C})$. For more details, see Nikiforov, 2016. More precisely, the trace norm of $\mathbf{B} \in M_n(\mathbb{C})$ is defined as $\parallel \mathbf{B} \parallel_\star = \sum_{i=1}^{n}\sigma_i$, where $\sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_n \geqslant 0$ are the singular values of $\mathbf{B}$. Therefore, if $\mathbf{B} \in M_n(\mathbb{C})$ is an Hermitian matrix with $Trace(\mathbf{B})$ and eigenvalues $\xi_1, \ldots, \xi_n$, then the matrix $\left(\mathbf{B} - \frac{Trace(\mathbf{B})}{n}\mathbf{I}_n\right)$ is also Hermitian with singular values $\left|\xi_1 - \frac{Trace(\mathbf{B})}{n}\right|, \ldots, \left|\xi_n - \frac{Trace(\mathbf{B})}{n}\right|$. As result, the norm of the new matrix is written as:

$$\left|\left|\mathbf{B} - \frac{Trace(\mathbf{B})}{n}\mathbf{I}_n\right|\right|_\star = \sum_{i=1}^{n}\left|\xi_i - \frac{Trace(\mathbf{B})}{n}\right| \tag{3.44}$$

Hence, each of the graph energies presented above can be seen as the trace norm of $\mathbf{B} - \frac{Trace(\mathbf{B})}{n}\mathbf{I}_n$ corresponding to a particular Hermitian matrix $\mathbf{B}$. For exemple, in case where $\mathbf{B}$ is the adjacency matrix $\mathbf{A}$ of a graph G, then:

$$E_{\mathbf{B}} = \sum_{i=1}^{n} \mid \xi_i \mid = \parallel \mathbf{A} \parallel_\star, \tag{3.45}$$

with $Trace(\mathbf{A}) = 0$. Similarly, if $\mathbf{B}$ is the laplacian matrix $\mathbf{L}$ of G with trace equals $2m$ ($tr(\mathbf{L}) = 2m$), where m is the number of edges, then:

$$E_{\mathbf{B}} = \sum_{i=1}^{n} \left| \xi_i - \frac{2m}{n} \right| = \left\| \mathbf{L} - \frac{Trace(\mathbf{L})}{n} \mathbf{I}_n \right\|_{\star}. \tag{3.46}$$

Likewise holds for the signless laplacian energy, the normalized laplacian energy and the distance energy. The right side expression in (3.44) is an alternative approach to extend the concept of energy to the set of all $n \times n$ matrices $M_n(\mathbb{C})$. It worth to note that this expression was conceived by V. Consonni and R. Todeschini, 2008, where $\mathbf{B}$ is a molecular matrix.

3.3.3 Bounds for Adjacency Energy Invariant

Among the many goals behind the study of graph energy is that of formulating some interesting bounds corresponding to some extremal examples of graphs. The first main step towards the understanding $E_{\mathbf{A}}$'s dependency on the structure of molecular graphs was established by B. J. McClelland, 1971, defining an upper and a lower bounds of $E_{\mathbf{A}}$ in terms of simple graph invariants:

$$\sqrt{2m + n(n-1) \mid det(\mathbf{A}) \mid^{2/n}} \leqslant E_{\mathbf{A}} \leqslant \sqrt{2mn}, \tag{3.47}$$

where n and m are the number of nodes and edges of a molecular graph, and $\mathbf{A}$ is its associated adjacency matrix. Precisely, in the case of molecular graphs representing conjugated hydrocarbons, n represents the number of carbon atoms and m is the number of carbon-carbon bonds (σ and π bonds). It should be noted that this bound (3.47) is not defined exclusively for molecular graphs, and it can be checked on any other connected graph. The superior bound $E_{\mathbf{A}} = \sqrt{2mn}$, often referred to as the McClelland upper bound, plays a crucial role in the theory of total π-electron energy and in the characterisation of π-bonds in conjugated hydrocarbons. This bound is attained exactly in the cases where G is either an empty graph or a 1-regular graph.

In the meantime, several other bounds for $E_{\mathbf{A}}$ were proposed in the litterature. The following upper and lower limits have been established for the purpose of relying only on the number of edges or nodes of the molecular graph, G. Caporossi et al., 1999:

$$2\sqrt{m} \leqslant E_{\mathbf{A}} \leqslant 2m, \tag{3.48}$$

$$E_{\mathbf{A}} \geqslant 2\sqrt{n-1}, \tag{3.49}$$

where m and n are respectively the number of edges and nodes of G. The equality $E_{\mathbf{A}} = 2\sqrt{m}$ holds if G has no isolated nodes, in addition, if and only if G is a complete bipartite graph. While the equality $E_{\mathbf{A}} = 2m$ holds if and only if G is regular of degree 1. Furthermore, the lower bound (3.49) applies only to graphs with no isolated nodes. Unlike the bounds (3.48) and (3.49), the following theorem gives an upper bound of $E_{\mathbf{A}}$ which have a physical interpretation related to the energy of the π-electron:

Theorem 3.3.1. *(J. H. Koolen and V. Moulton, 2001). Let G be a graph with n nodes and m edges. If n and m satisfy $2m \geqslant n$, then the inequality*

$$E_{\mathbf{A}} \leqslant \frac{2m}{n} + \sqrt{(n-1)\left[2m - \left(\frac{2m}{n}\right)^2\right]} \tag{3.50}$$

holds. With equality, if and only if G is either $\frac{n}{2}K_2, K_n$, or a non-complete connected strongly regular graph with two non-trivial eigenvalues, both with absolute value $\sqrt{(2m - (\frac{2m}{n})^2)/(n-1)}$.

Recall that K_n is a complete graph on n nodes, and $\frac{n}{2}K_2$ is a multi-components graph built by the union of $\frac{n}{2}$ complete graphs K_2 having 2 nodes. The inequality (3.50) represents an important upper bound of the total π-electron energy, and it holds for all conjugated hydrocarbons.

Otherwise, the following theorem defines an upper bound of $E_{\mathbf{A}}$ concerning bipartite graphs, especially those which arise from chemistry studying alternant hydrocarbons:

Theorem 3.3.2. *(J. H. Koolen and V. Moulton, 2003). Let G be a bipartite graph on $n > 2$ nodes, then*

$$E_{\mathbf{A}} \leqslant \frac{n(\sqrt{n} + \sqrt{2})}{\sqrt{8}} \tag{3.51}$$

holds, with equality holding if and only if $n = 2\nu$ and G is the incidence graph of a 2-$\left(\nu, \frac{\nu+\sqrt{\nu}}{2}, \frac{\nu+2\sqrt{\nu}}{4}\right)$-design.

The bound (3.50) is sharp, and it provides an infinite family of maximal energy bipartite graphs. It is very useful in the study of hyper-energetic graphs. See X. Yang et al., 2016 for more details on incidence graphs constructed from t-designs. Moreover, in the following theorem, Gutman, 2005 stated that the upper bound (3.52) of $E_{\mathbf{A}}$ is the best known bound in terms of the number of nodes, even better than the bound (3.49), except for graph with $n = 64, 256, 1024, 4096, ...$:

Theorem 3.3.3. *(J. H. Koolen and V. Moulton, 2001). Let G be a graph on n nodes, then*

$$E_{\mathbf{A}} \leqslant \frac{n}{2}(1 + \sqrt{n}) \tag{3.52}$$

holds, with equality if and only if G is a strongly regular graph with parameters $(n, (n+\sqrt{n})/2, (n+2\sqrt{n})/4, (n+2\sqrt{n})/4)$.

We recall that a k-regular graph G on n nodes is called *strongly regular* with parameters (n, k, λ, μ) if the following conditions hold. Every pair of adjacent nodes has the same number $\lambda \geqslant 0$ of common neighbors, and every pair of non-adjacent nodes has the same number of $\mu \geqslant 0$ of common neighbors. If $\mu = 0$, then G is non-complete, then the eigenvalues of G are k (the trivial eigenvalue) and r, s are the roots of the quadratic equation $x^2 + (\mu - \lambda)x + (\mu - k) = 0$.

The question of how the energy of a bipartite graph can be small has been addressed by I. Gutman et al., 2012 via a particular family of graphs. Let G be a bipartite graph with n nodes and m edges, satisfying $n \leqslant m \leqslant 2n - 4$, then

$$E_{\mathbf{A}} \geqslant 2\sqrt{m + 2\sqrt{(m - n + 2)(2n - m - 4)}}, \tag{3.53}$$

holds. The equality is achieved by the bipartite graph G having two node sets V_1 and V_2 with cardinalities $\mid V_1 \mid = 2$, $\mid V_2 \mid = n - 2$. The graph G is built by joining one node from V_1 to all nodes in V_2, and then the construction is completed by connecting the remaining node of V_1 to the $(m - n - 2)$ nodes of V_2.

3.3.4 Bounds for Laplacian Energy Invariant

Unlike adjacency based graph energy, laplacian energy does not have a clear connection to chemical problems. Nevertheless, it attracted much attention of mathematicians for its interesting algebraic properties. $E_{\mathbf{A}}$ and $E_{\mathbf{L}}$ energies have many similar properties, but also some differences. I. Gutman and B. Zhou, 2006 pointed out these aspects and established the following upper and lower bounds of $E_{\mathbf{L}}$ associated to particular graph families:

$$E_{\mathbf{L}} \leqslant \sqrt{2Mn}, \quad M = m + \frac{1}{2}\sum_{v_i \in V}\left(d(v_i) - \frac{2m}{n}\right)^2, \tag{3.54}$$

$$E_{\mathbf{L}} \leqslant \frac{2m}{n} + \sqrt{(n-1)\left[2M - \left(\frac{2m}{n}\right)^2\right]}, \tag{3.55}$$

$$2\sqrt{M} \leqslant E_{\mathbf{L}} \leqslant 2M. \tag{3.56}$$

Theorem 3.3.4. *(I. Gutman and B. Zhou, 2006). Inequality* (3.54) *holds for any* (n, m)*-graph* G*. Equality is atteined if and only if* G *is either regular of degree* 0 *or consists of* α *copies of complete graphs of order* k *and* $\alpha(k-2)$ *isolated nodes,* $\alpha \geqslant 1, k \geqslant 2$*. (Recall that in the case* $k = 2$*,* G *is regular of degree 1).*

Theorem 3.3.5. *(I. Gutman and B. Zhou, 2006). Let* G *be an* (n, m)*-graph and* p *the number of its*

components $(p \geqslant 1)$*, then*

$$E_{\mathbf{L}} \leqslant \frac{2m}{n}p + \sqrt{(n-p)\left[2M - p\left(\frac{2m}{n}\right)^2\right]}, \tag{3.57}$$

holds. For $p = 1$*, equality is attained if and only if* G *is either a regular graph of degree* $0, 1$*, or* $n-1$*, or a non-complete connected strongly regular graph with two non-trivial eigenvalues both having absolute value* $\sqrt{[2m - (2m/n)^2]/(n-1)}$*. For* $p = n$*,* G *consists of isolated nodes, thus* $E_{\mathbf{L}} = 0$*. For any* p*, equality holds for graphs consisting of* α *copies of complete graphs of order* k *and* $\alpha(k-2)$ *isolated nodes,* $\alpha \geqslant 1, k \geqslant 2$*, provided* $\alpha(k-2) = p$*. (Recall that in the case* $p = n/2$*, if* $k = 2$*, then* G *is regular of degree 1).*

Theorem 3.3.6. *(I. Gutman and B. Zhou, 2006). The left-hand side inequality* (3.56) *holds for any* (n, m)*-graph. Equality* $E_{\mathbf{L}} = 2\sqrt{M}$ *is attained if and only if* G *is the complete bipartite graph* $K_{n/2,n/2}$*. The right-hand side inequality* (3.56) *holds for graphs without isolated nodes. For such graphs, the inequality* $E_{\mathbf{L}} = 2M$ *is attained if and only if* G *is regular of degree1.*

3.3.5 Relation between $E_{\mathbf{A}}$ and $E_{\mathbf{L}}$

Let $\mathbf{B}$ be a real and symmetric square matrix of size n. Let $\sigma_i(\mathbf{B}), i \in \{1, 2, ..., n\}$, be its singular values and $\rho_i(\mathbf{B}), i \in \{1, 2, ..., n\}$ its eigenvalues. Then, the relation between them is given by: $\sigma_i(\mathbf{B}) = \mid \rho_i(\mathbf{B}) \mid$ for $i \in \{1, 2, ..., n\}$. This relation is important in the theory of graph energy, since Nikiforov, 2016 defined the energy of a graph as the sum of the singular values corresponding to its adjacency matrix $\mathbf{A}$. Another important tool is the following Ky Fan's theorem, which establishes the relationship between the singular values of a sum of two matrices and those of the resulting matrix, which is indeed widely used to define some bounds for graph energies:

Theorem 3.3.7. *(Fan, 1951). Let* $\mathbf{B_1}$*,* $\mathbf{B_2}$ *and* $\mathbf{B}$ *be square matrices of size* n $(\in M_n(\mathbb{R}))$*, such that* $\mathbf{B_1} + \mathbf{B_2} = \mathbf{B}$*, then*

$$\sum_{i=1}^{n} \sigma_i(\mathbf{B_1}) + \sum_{i=1}^{n} \sigma_i(\mathbf{B_2}) \geqslant \sum_{i=1}^{n} \sigma_i(\mathbf{B}), \tag{3.58}$$

holds. Equality occurs if and only if there exists an orthogonal matrix $\mathbf{P}$*, such that the matrices* $\mathbf{PB_1}$ *and* $\mathbf{PB_2}$ *are both positive semi-definite.*

As a first consequence of the theorem (3.3.7), one can conclude that for a set of graphs G_1, G_2 and G, whose adjacency matrices satisfy the condition: $\mathbf{A_1} + \mathbf{A_2} = \mathbf{A}$, their energies verify the inequality: $E_{\mathbf{A_1}} + E_{\mathbf{A_2}} \geqslant E_{\mathbf{A}}$. Moreover, some special cases of this inequality are given by the following corollaries (3.3.1) and (3.3.2):

Corollary 3.3.1. *(W. So et al., 2010). Let G be a graph on n nodes and let $\bar{G}$ denotes its complement, then*

$$E_{\mathbf{A}} + E_{\bar{\mathbf{A}}} \geqslant 2(n-1), \tag{3.59}$$

holds, with equality if and only if either $G \cong K_n$ or $G \cong \bar{K}_n$.

Corollary 3.3.2. *(W. So et al., 2010). Let G be a graph with n nodes and m edges, and let Δ be the highest degree among all node degrees, then*

$$E_{\mathbf{A}} \leqslant 2m - 2(\Delta - \sqrt{\Delta}), \tag{3.60}$$

holds, with equality if and only if G is a union of the star graph $S_{\Delta+1}$, with $m - \Delta$ isolated edges and $n - 2m + \Delta - 1$ isolated nodes.

Furthermore, W. So et al., 2010 have proved using the Ky Fan theorem (3.3.7) that the laplacian based graph energy $E_{\mathbf{L}}$ is upper bounded by the adjacency based energy $E_{\mathbf{A}}$ adjusted by a quantity dependent on the degrees of nodes, as stated by the following corollary (3.3.3):

Corollary 3.3.3. *(W. So et al., 2010). Let G be a non-empty graph with n nodes and m edges, with node degrees $d_1, d_2, ..., d_n$, then*

$$E_{\mathbf{L}} \leqslant E_{\mathbf{A}} + \sum_{i=1}^{n} \Big| d_i - \frac{2m}{n} \Big|, \tag{3.61}$$

holds, where $2m/n$ is the average node degree of G.

Proof (W. So et al., 2010).

Note that:

$$\mathbf{L} - \frac{2m}{n}\mathbf{I}_n = \mathbf{D} - \mathbf{A} - \frac{2m}{n}\mathbf{I}_n = -\mathbf{A} + \Big[\mathbf{D} - \frac{2m}{n}\mathbf{I}_n\Big]. \tag{3.62}$$

And by applying Ky Fan theorem (3.3.7) on equation (3.62), the following inequalities are deduced:

$$\begin{aligned}
E_{\mathbf{L}} &= \sum_{i=1}^{n} \sigma_i\Big(-\mathbf{A} + \Big[\mathbf{D} - \frac{2m}{n}\mathbf{I}_n\Big]\Big), \\
&\leqslant \sum_{i=1}^{n} \sigma_i(-\mathbf{A}) + \sum_{i=1}^{n} \sigma_i\Big(\Big[\mathbf{D} - \frac{2m}{n}\mathbf{I}_n\Big]\Big), \\
&\leqslant \sum_{i=1}^{n} \sigma_i(\mathbf{A}) + \sum_{i=1}^{n} \sigma_i\Big(\Big[\mathbf{D} - \frac{2m}{n}\mathbf{I}_n\Big]\Big), \\
&\leqslant E_{\mathbf{A}} + \sum_{i=1}^{n} \Big| d_i - \frac{2m}{n} \Big|,
\end{aligned}$$

knowing that $\Big(\mathbf{D} - \frac{2m}{n}\mathbf{I}_n\Big)$ is a diagonal matrix whose eigenvalues are $\Big(d_i - \frac{2m}{n}\Big), i \in \{1, 2, ..., n\}$.

Despite this result, I. Gutman et al., 2008 conjectured that the laplacian based energy is always greater than or equal to the one based on adjacency $E_{\mathbf{L}} \geqslant E_{\mathbf{A}}$. However, the validity of this conjecture was eventually disproved by means of counter-examples in J. Liu and B. Liu, 2009, D. Stevanović et al., 2009. Additionally, W. So et al., 2010 show by the following theorem that the conjecture at least remains valid for bipartite graphs:

Theorem 3.3.8. *(W. So et al., 2010). Let G be a bipartite graph with n nodes and m edges, with node degrees $d_1, d_2, ..., d_n$, then*

$$max\left\{E_{\mathbf{A}}, \sum_{i=1}^{n}\left|d_i - \frac{2m}{n}\right|\right\} \leqslant E_{\mathbf{L}} \leqslant E_{\mathbf{A}} + \left|d_i - \frac{2m}{n}\right|. \tag{3.63}$$

holds, with $2m/n$ is the average node degree of G.

3.3.6 $E_{\mathbf{L}}$ Measures Complexity

Due to its practical importance, graph complexity quantification has attracted significant attention in various domains such as pattern recognition, control theory or network analysis, F. Escolano et al., 2008a. The measure of this complexity is important for different applications including embedding, A. Robles-Kelly and E.R. Hancock, 2007, classification, A. Shokoufandeh et al., 1999 and filtering of image description hierarchies Y.Z. Song et al., 2010. Such a quantification not only allows the complexity of different graph structures to be compared, but also allows it to be measured versus the enhancement of fitting quality of data when a structure is being learned. The graph complexity can be measured in different ways. Among them there is the number of spanning trees and its connections with the laplacian spectrum, methods based on path-length chromatic decomposition and others, see F. Escolano et al., 2008a for more details. An attractive measure of complexity is the laplacian graph energy $E_{\mathbf{L}}$ (3.40), applied by Y.Z. Song et al., 2010 in image processing, allowing to measure images similarity by comparing their textures represented on graphs. They show that laplacian graph energy is a broad measure of graph complexity. They observe that regular structures which tend to be connected and monoton, such as polygons, exhibit lower laplacian graph energy than structures comprising randomly selected edges and in which some irregularities occur.

To reinforce their observations, we conducted a simulation to show that effectively, the complexity of a graph is correlated with its energy. This fact is shown by using Erdös-Rényi random graphs parametrized with the connectivity probability parameter p. As p increases from zero to one, the model becomes more and more likely to include graphs with more edges and less and less likely to include graphs with fewer edges. The figure 3.8 shows that the quantities $E_{\mathbf{L}}$ and $E_{\mathbf{A}}$ are both relevant for measuring the complexity of graphs, their respective curves take a smooth upward rate when the Erdös-Rényi probability increases, which indicates an augmentation in complexity of the generated graphs. Furthermore, the

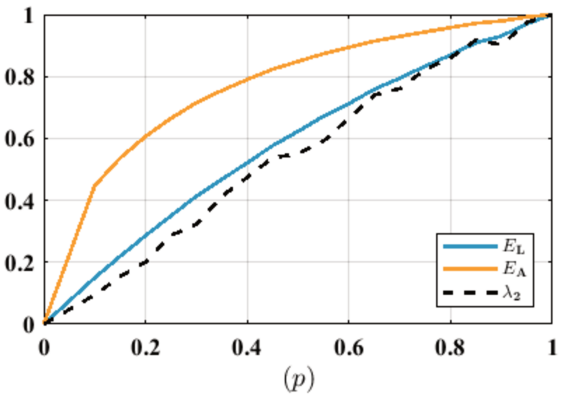


Figure 3.8: $E_{\mathbf{A}}$ and $E_{\mathbf{L}}$ evolution when the structural complexity of the graphs increases, illustrated in Erdös-Rényi random graphs, having 700 nodes. λ_2 is the algebraic connectivity of the graph, called the Fiedler eigenvalue.

growth tendency is not the same, revealing a difference in the sensitivity of these energies ($E_{\mathbf{L}}$ and $E_{\mathbf{A}}$) to structural changes in the graph. Overall, this may mean that the matrices $\mathbf{A}$ and $\mathbf{L}$ recover in different manner morphological characteristics of the graphs, thus their underlying complexity. We specify that all generated graphs are connected except for the first one corresponding to $p = 0$. We observe also the increase of the algebraic connectivity (λ_2) measured by the second eigenvalue of $\mathbf{L}$, as well as that the energy $E_{\mathbf{L}}$ estimate it in smoother and precise way. It is worth also mentioning that the connectivity and complexity concepts become close when it comes to connected graphs.

3.4 Laplacian Graph Energy as Similarity Measure

As explained above, the graphs energy is an interesting quantity that can characterize a wide range of structural data. Its link with the total π-electron energy, which is a physical invariant characterizing chemical molecules is of great advantage. In addition to its ability to measure the structural complexity of the graph in an affine way, we exploit the inherent information embedded in its eigenspectrum. Our idea is to compare the complexity of the graphs, by comparing their respective Laplacian energies, assuming that two graphs are probably to be the same or close when their energies go closer. Let G_i and G_j be two graphs having comparable sizes ($n_i \simeq n_j$). We define GE as the quantity that indicates their similarity degree, and given by:

$$\mathrm{GE}(G_i, G_j) = |E_{\mathbf{L}}(G_i) - E_{\mathbf{L}}(G_j)|. \tag{3.64}$$

GE is the 1D-euclidean distance between laplacian energies of the compared graphs. $E_{\mathbf{L}}(G_i)$ is computed using equation (3.40). The GE distance is a spectral measure of similarity, which considers the eigenspectrum of laplacian matrix as an invariant, assuming that is unique and specific to each graph. But,

this is not true for all graphs. There are some particular graphs that share the same laplacian spectrum despite their structural differences, called cospectral graphs, which are hard to separate based only on the GE measure. Otherwise, the GE is a global measure but does not include explicit local information about the structure of the compared graphs and even their node labels (Signal values). However, these weaknesses can be overcomed by combining in the same measure both the GE and TVG measures. For this purpose, we use a convex combination to define some trade-off between the two quantities, in which a better discrimination of the graphs is achievable. The new measure is called Joint Total variation Energy (JET), and it is denoted by:

$$\mathrm{JET}(G_i, G_j) = \alpha \times \mathrm{GE}(G_i, G_j) + (1 - \alpha) \times \mathrm{TVG}(G_i, G_j), \tag{3.65}$$

where $\alpha \in [0, 1]$ is a weighting parameter which controls the contributions of each measure, while taking into account both the global complexity of the graph and the interaction of the node values with its underlying structure. The effectiveness of these new measures is illustrated in classification tasks, by integrating them in an exponential function to have a valid kernel similar to the Radial Basis Function kernel (RBF)(B. Schölkopf and A.J. Smola, 2002). This kernel associates to each pair of graphs (G_i, G_j) the quantity:

$$\mathbf{K}_{i,j} = \sum_{n=0}^{\infty} \frac{1}{n!} (-\gamma\, \mathcal{S}(G_i, G_j))^n = \exp\left(-\gamma\, \mathcal{S}(G_i, G_j)\right), \tag{3.66}$$

where $\mathbf{K}$ is the square kernel matrix of size $M \times M$ representing the number of graphs to be compared, and $\mathcal{S}(G_i, G_j)$ is one of the similarity measures (TVG, GE, JET), defined by formulas (3.34), (3.64) and (3.65) respectively. γ is a smoothing factor, which controls the decreasing rate of the exponent and guarantees that high order terms of the sum vanishes gradually.

3.5 Graphs Classification using (TVG, GE, JET) Measures

3.5.1 Graph Datasets

To show the effectiveness of the proposed similarity measures, five data sets are used, all concerning chemical/biological compounds. We have thus:

- **MUTAG**: In genetics, a mutagen is a physical or chemical agent that changes the genetic material, usually DNA, of an organism. It interacts with the DNA resulting on the creation of a corrupted sequence by addition or deletion of specific sections. Therefore, mutagenic molecules have a high risk of toxicity, especially for humans. The Chemical Carcinogenicity Research Information

System (CCRIS) database contains scientifically tested data for 7,000 molecules almost. **MUTAG** dataset was prepared by A.K. Debnath et al., 1991, containing 188 mutagenic aromatic and heteroaromatic nitro compounds labeled according to whether or not they have mutagenic effect on the Gram-negative bacterium *Salmonella typhimurium*. K. Riesen and H. Bunke, 2008 converted them to a graph represented compounds.

- **NCI1**, **NCI109**: represent two balanced datasets of chemical compounds screened for activity against non-small cell lung cancer and ovarian cancer cell lines, N. Wale and G. Karypis, 2006, S. Kim et al., 2015. Discovery, design and development of new drugs is an expensive and challenging process. Any new drug should not only produce the desired response to the disease but should do so with minimal side effects. One of the key steps in the drug design process is the identification of the chemical compounds (hit compounds or just hits) that display the desired and reproducible behavior against the specific biomolecular target. This represents a significant hurdle in the early stages of drug discovery. Therefore, computational techniques that build models to correctly assign chemical compounds to various classes or retrieve compounds of desired class froma database have become popular in the pharmaceutical industry. The **NCI1**, **NCI109** datasets are derived from the PubChem website (S. Kim et al., 2015) that contains twelve datasets selected from the bioassay records for cancer cell lines. **NCI1** deals with the Non-Small Cell Lung human tumor, while **NCI109** deals with the Ovarian human tumor, both labeled either active or inactive via the cell line growth inhibition assay. Each compound is represented by a molecular graph, nodes correspond to the various atoms (e. g. carbon, nitrogen, oxygen, etc), and edges correspond to the bonds between the atoms (σ-bond, π-bond,...etc). K.M. Borgwardt et al., 2005 labeled the nodes by real numbers characterizing each type of atom.

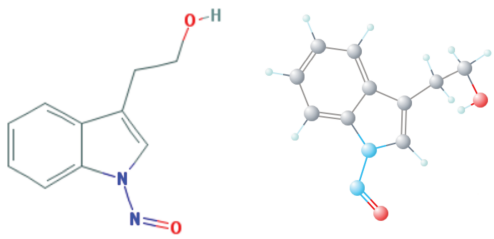
- **ENZYMES**: represent the largest and most diverse group of all proteins, catalysing all chemical reactions in the metabolism of all organisms. They play a key role in the regulation of metabolic steps within the cell. With the development and progress of projects of structural and functional genomics and metabolomics, the collection and processing of enzyme data becomes even more important in order to analyse and understand biological processes. BRENDA database (I. Schomburg et al., 2004) represents a comprehensive relational database containing all enzymes classified according to the EC system of the Enzyme Nomenclature Committee (IUBMB). This classification is based on the type of reaction (e. g. oxidation, reduction, hydrolysis, group transfert) catalysed by the enzyme. BRENDA holds information on 4200 EC numbers, which represent more than 83000 different enzyme molecules. The **ENZYMES** database consists on a labeled graphs representing

600 enzymes from BRENDA, built by K.M. Borgwardt et al., 2005 to classify them into one of the 6 top-level classes. The proteins were modeled by labeled an undirected graphs. Nodes in each graph represent $SSEs$ within the protein structure, (i. e. α-helix, β-sheet, β-turn,...etc), while the SSE means the Estimated Secondary Structure using a spectral modeling procedure based on the analysis of the infrared protein spectra. Edges connect nodes if those are direct neighbors along the secondary structure (Figure 3.13) or if they are neighbors spacially in the protein structure. The node structure gives a type label, stating whether they represent a helix, sheet or turn, and physical/chemical information about hydrophobicity, the van der Waals volume, the polarity and polarizability of the SSE represented by this node. One total normalized van der Waals value is determined for each node. Additionaly, each node is labeled with the total number of its residues with low, medium or high normalized van der Waals volume. The length of each SSE in secondary structure and the distance between the C_α atom of its first and last residue in Angströms constitute further node attributes. Every edge is labeled with its type, i. e. structural or sequential. Sequential edges are labeled with their length. The length of a structural edge between two $SEEs$ is calculated to be the distance between their centers, where the center of an SSE is the midpoint of the line between the C_α atom of its first and the C_α of its last residue. For precision, the alpha carbon (C_α) in organic molecules refers to the first carbon atom that attaches to a functional group.

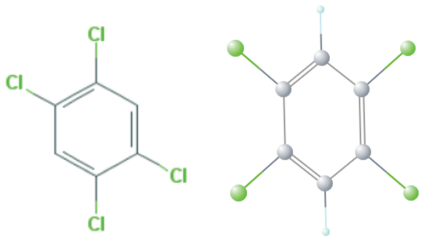
- **D&D**: The ability to predict protein function from structure is of increasing importance, as the number of structures resolved is growing more rapidly than our capacity to determine their function. The protein function can be predicted as enzymatic or not without resorting to alignments, which is at the basis of a big part of prediction methods. **D&D** is a graph dataset built by P.D. Dobson and A.J. Doig, 2003, that describes 1178 high-resolution proteins in a structurally non-redundant subset from the Protein Data Bank. The dataset is split into two functional classes, enzymes and non-enzymes. Each protein is represented by a graph, in which the nodes are amino acids (Figure 3.12) and two nodes are connected by an edge if they are separated by less than 6 Angström. The graphs were built from the primary structural level of proteins, (Figure 3.13).

3.5.2 Experimental Configuration

The kernel defined in equation (3.66) is integrated in a Support Vector Machine (C-SVM) as a kernel, then the classification is done following the 10-fold cross-validation outline, in which 9 folds are used for training and 1 for testing. All the datasets are randomly shuffled before partitioning and the whole experiment is repeated 10 times to avoid random effects of fold assignments, while the kernel parameter



1-Nitrosotryptophol, $C_{10}H_{10}N_2O_2$.



1,2,4,5-Tetrachlorobenzene, $C_6H_2Cl_4$.

Figure 3.11: Examples of chemical compounds get from the MUTAG dataset, the left compound is *mutagen*, while the right one is *non-mutagen*, (https://pubchem.ncbi.nlm.nih.gov).

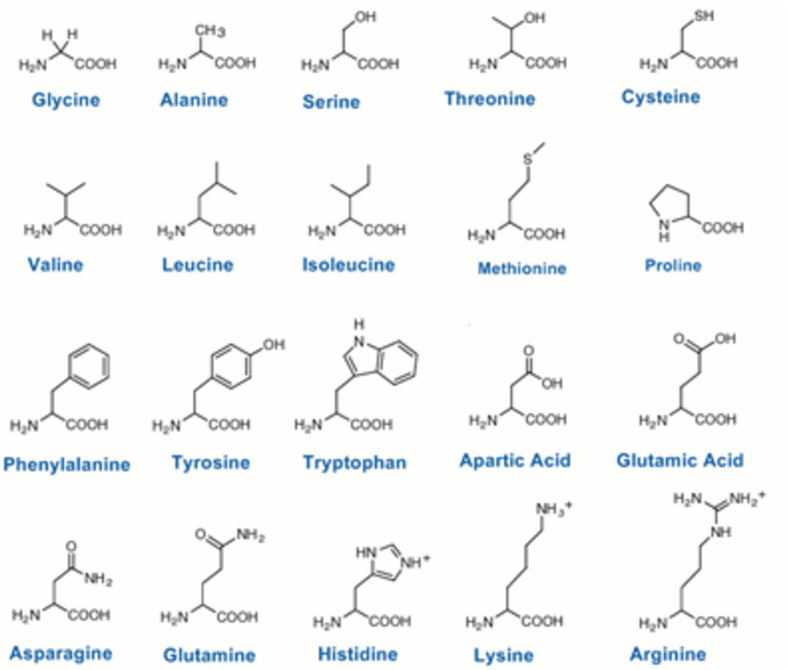


Figure 3.12: Basic Amino Acids that represent the alphabet to create proteins, (http://www.gustrength.com/amino-acids).

Table 3.1: Some statistics about the used bioinformatics datasets.

Methode/Dataset	MUTAG	NCI1	NCI109	ENZYMES	D&D
Number of graphs	188	4110	4127	600	1178
Maximum number of nodes	28	111	111	126	5748
Average number of nodes	17.93	29.87	29.68	32.63	284.32
Number of labels	7	37	38	3	82
Number of classes	2	2	2	6	2

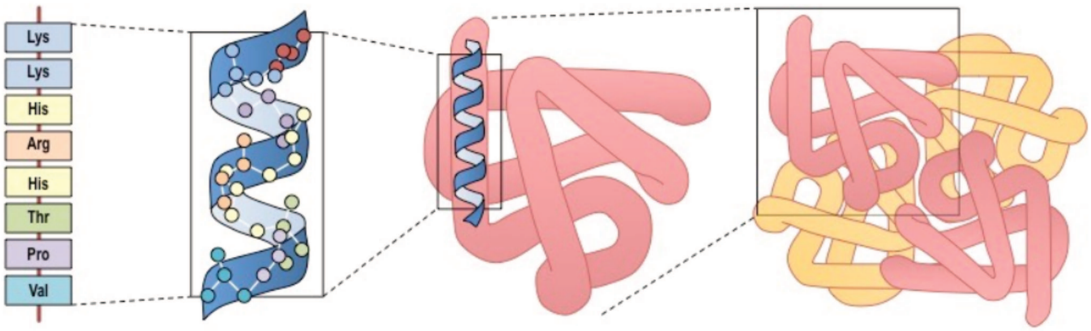


Figure 3.13: The four levels of the protein structure (Primary, Secondary, Tertiary and Quaternary structures), (http://ib.bioninja.com.au/higher-level/topic-7-nucleic-acids/73-translation/protein-structure.html).

γ is set to 1. The performance of the JET measure in terms of classification accuracy is optimized by tuning the parameter $\alpha \in [0, 1]$, and picking out the value that maximizes performance. However, the best value of α is not known a-priori and not universal, hence, its value depends on the dataset at hand. Average classification accuracies and their associated standard deviations are summarized in Table 3.2. The performances of the kernel based on TVG, GE and JET measures are compared to some kernels of the literature, in terms of prediction accuracy and computation runtime on graphs benchmark datasets. Some known graph kernels are tested: those based on walks, Weisfeiler-Lehman isomorphism and limited-size subgraphs. Thus, our similarity measures are compared to the fast geometric random walk kernel proposed by S.V.N. Vishwanathan et al., 2010, which counts the common labelled walks and also with p-random walk kernel that compares random walks up to length p in two graphs (a special case of random walk kernels: H. Kashima et al., 2003, Gärtner, 2003). In the case of limited-size subgraphs family, we compare with an extension of the graphlet kernel proposed by Shervashidze and Borgwardt, 2009 that counts common induced labelled connected subgraphs of size 3. From Weisfeiler-Lehman kernels, we chosed the Weisfeiler-Lehman edge kernel N. Shervashidze et al., 2011,Shervashidze and Borgwardt, 2009, which counts matching pairs of edges with identically labeled endpoints (incident nodes) in two graphs. Concerning computing set-ups, the accuracy and runtime values of the benchmark kernels are performed and reported by N. Shervashidze et al., 2011, while the runtime in minutes and seconds of our methods (Table 3.3) are measured in Anaconda2 4.1.1 Python 2.7.12 Lab, installed on a Windows machine endowed with a 3 GHz-Intel 8-Core processor and 16GB of RAM.

3.5.3 Results Evaluation

Overall, as shown in Table 3.2, on NCI1, NCI109, ENZYMES and D&D, Weisfeiler-Lehman edge kernels reach the highest accuracy but perform less than JET kernel on MUTAG. Indeed, as shown in Table 3.2, on MUTAG, NCI1, NCI109, and D&D, the TVG, GE, JET based kernels reach good accuracy and

are competitive with other kernels. On MUTAG, the GE and JET based kernels give the second best accuracy and perform better than random walk and Weisfeiler-Lehman edge kernels. On NCI1, NCI109, ENZYMES, and D&D, the JET kernel reaches the third best accuracy and performs better than random walk kernels. In terms of computation runtime, on all data sets the JET, TVG and GE kernels are more faster than all other kernels and particularly compared to Ramon and Gärtner kernel. As shown in Table 3.3, our similarity measures outperform state-of-the-art graph kernels in terms of computation runtime. The best computational time over all the data sets and all the considered methods is provided by the TVG kernel. This result highlights the low complexity of this kernel. Note that the JET kernel is faster ($\times 2000$) than Weisfeiler-Lehman edge kernel in D&D data set with almost the same accuracy (75% vs 78%). By combining TVG and GE similarity measures, we improve their individual performance about 7% for ENZYMES, 3% for NCI1/NCI109 and 1% for MUTAG. Regarding D&D, no improvement is obtained with the combination, the best accuracy is reached by TVG alone. Figure 4.4 shows the behaviour of prediction performance of JET kernel according to the weighting factor α. We note that the maximum accuracy is obtained between the limit values of α ($\alpha \in [0, 1]$), which confirms that with the contribution of both TVG and GE measures, higher accuracy rates can be reached.

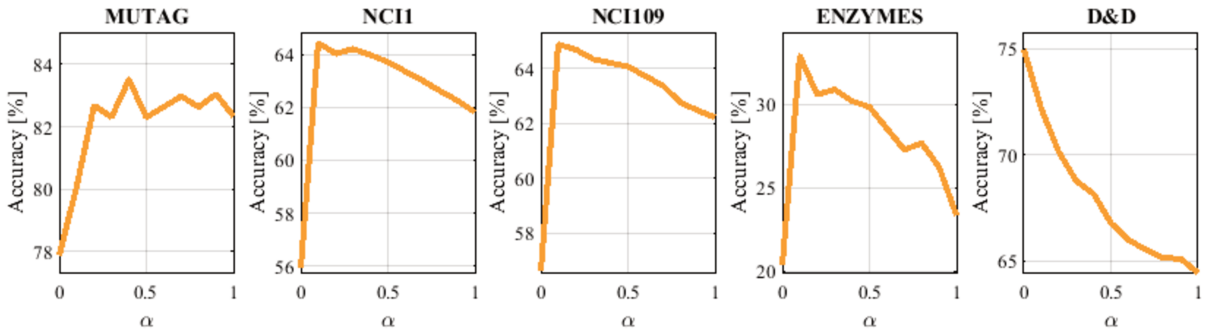


Figure 3.14: Variation of the accuracy achieved by the JET based kernel in function of the weighting parameter α.

3.6 Conclusion

In this chapter we have discussed the problem of similarity measurement of graphs, and their utility for learning applications, like the classification tasks. In addition, we reviewed the notions of the total variation (TV) of a signal and particularly of a signal on a graph, and the energy ($E_{\mathbf{L}}$) associated with its structure. By being calculated via one of the eigen-spectra associated with the graph, the energy is a pertinent information that characterizes well the graph, and measures the complexity degree of its structure, taking into account both connections distribution of the network and its density. While the total variation quantifies the oscillatory behaviour of the graph-signal and its interaction with the supporting

Table 3.2: Classification accuracy on some bioinformatics data (± standard deviation).

Methode/Dataset	MUTAG	NCI1	NCI109	ENZYMES	D&D
Graph Signal Total Variation TVG	77.88 (±1.43)	55.91 (±0.38)	56.62 (±0.10)	20.36 (±0.91)	75 (±0.10)
Graph Signal Energy GE	82.34 (±1.25)	61.81 (±0.40)	62.22 (±0.35)	23.33 (±1.44)	64.45 (±0.61)
Energy and Total Variation JET	83.51 (±1.14)	64.43 (±0.21)	64.88 (±0.16)	31 (±0.53)	75 (±0.04)
Ramon and Gärtner [T. Gärtner et al., 2003b]	85.72 (±0.49)	61.86 (±0.27)	61.67 (±0.21)	13.35 (±0.87)	57.27 (±0.07)
p-random walk [H. Kashima et al., 2003]	79.19 (±1.09)	58.66 (±0.28)	58.36 (±0.94)	27.67 (±0.95)	66.64 (±0.83)
Random walk [S.V.N. Vishwanathan et al., 2010]	80.72 (±0.38)	64.34 (±0.27)	63.51 (±0.18)	21.68 (±0.94)	71.70 (±0.47)
Graphlet count [Shervashidze and Borgwardt, 2009]	75.61 (±0.49)	66.00 (±0.07)	66.59 (±0.08)	32.70 (±1.20)	78.59 (±0.12)
Weisfeiler-Lehman edge [N. Shervashidze et al., 2011]	81.06 (±1.95)	84.37 (±0.30)	84.49 (±0.20)	53.17 (±2.04)	77.95 (±0.70)

Table 3.3: CPU runtime for kernel computation on some bioinformatics data.

Methode/Dataset	MUTAG	NCI1	NCI109	ENZYMES	D&D
Graph Signal Total Variation TVG	0.036″	17″	17.4″	0.4″	1.4″
Graph Signal Energy GE	0.12″	20.6″	21.8″	1″	2′19″
Energy and Total Variation JET	0.18″	51″	51.2″	1.6″	2′21″
Ramon and Gärtner [T. Gärtner et al., 2003b]	40′6″	81 days	81 days	38 days	103 days
p-random walk [H. Kashima et al., 2003]	4′42″	5 days	5 days	10′	4 days
Random walk [S.V.N. Vishwanathan et al., 2010]	12″	9 days	9 days	12′19″	48 days
Graphlet count [Shervashidze and Borgwardt, 2009]	3″	1′27″	1′27″	25″	30′21″
Weisfeiler-Lehman edge [N. Shervashidze et al., 2011]	3″	1′5″	58″	11″	3 days

structure. Given these properties, we proposed new graph-signals similarity measures based on the total variation and the laplacian graph energy, adapted for labeled weighted and unweighted graphs, which we called respectively (TVG) and (GE). These two measures integrated in an exponential kernel show competitive performance on binary and multiclass graph-signals classification. To take advantage from both measures, we combined them in a new joint measure called JET. Applied on some bioinformatics classification problems, our measures yield competitive accuracy levels on all considered data sets and outperform some state-of-the-art graph kernels in terms of computation runtime. The results of the JET measure show the benefits of hybrid approaches on discriminating graph signals without significant

increase of complexity. In spite of the new perspectives related to the optimization of the JET measure and its generalization to other types of data, we wondered about the choice of the graph energy that permits a better characterization and discrimination of its intrinsic structure? According to the different definitions of graph energy depending directly on the eigen-spectra, this leads us to ask the question about which matrix $(\mathbf{A}, \mathbf{L}, ..., \mid \mathbf{L} \mid)$ could represent the graph at best? This question is addressed in the next chapter.

CHAPTER 4

A Joint Spectral Similarity Measurement for Graphs Classification

4.1 Introduction

Graph spectral analysis is one of the hot topics in data processing community, motivated by the prominent need to develop new mathematical tools to process networked and structured data. These data are generated from various sources, as sensor, social, biological or transportation networks, where the information resides in complex and irregular structures. For this purpose, eigen-spectrum of matrices associated with graphs are often closely studied. Recent works of the literature have emphasized the importance of matrix representations for graph characterization, pointing out the advantages and the drawbacks of some spectra associated to graphs E.R. Van Dam and W.H Haemers, 2003, I. Jovanović and Z. Stanić, 2014, including, those of adjacency ($\mathbf{A}$), Laplacian ($\mathbf{L}$), signless Laplacian $\mid \mathbf{L} \mid$ and distance ($\mathbf{D}^G$) matrices. The spectrum of $\mathbf{L}$ matrix is indeed widely studied in spectral graph theory, R.K. Fan Chung, 1996a, in reason of the symmetry and positive semi-definiteness of the matrix, which is useful for determining cuts and inherent graph components. Otherwise, the spectrum of $\mathbf{A}$ matrix is mainly used for the study of regularity J.H. Koolen and H. Yu, 2011, isomorphisms D. Conte et al., 2004 and bipartition Kunegis, 2015 of graphs. The question of choosing either $\mathbf{A}$ or $\mathbf{L}$ matrix for graph representation is still a subject of debate. For instance, in graph signal processing theory, D.I. Shuman et al., 2013 define the graph Fourier basis as the eigenbasis of $\mathbf{L}$ matrix, while A. Sandryhaila and J.M.F. Moura, 2014 prefer the eigenbasis of $\mathbf{A}$ obtained via a Jordan decomposition. This difference can be justified in part by the nature itself of the decomposition basis, and also by the fact that not all graphs are determined by their spectra and there is a family of graphs that shares the same spectrum in respect to some matrix representation, commonly called

cospectral graphs, C.D. Godsil and B.D. McKay, 1982. The distinction between these cospectral graphs or between very similar graphs via their spectra is a tough task as stated by E.R. Van Dam and W.H Haemers, 2003. A big part of graph comparison algorithms aims to sort them using some structural similarity criteria without implicitly resorting to spectral analysis. In spite that, spectral invariants remain suitable for graphs discrimination, H.A. Bay-Ahmed et al., 2017a, we argued their strengths in chapter 3. We propose in this chapter to tackle the problem of graph classification using a spectral similarity measurement called Joint Spectral Similarity (JSS), which compares two graphs using jointly the spectra of $\mathbf{A}$ and $\mathbf{L}$. Such measure is interesting to distinguish graphs that have close spectral properties or even for sorting cospectral graphs which is a hard task when using only the spectrum of $\mathbf{A}$ or $\mathbf{L}$ at once. To understand the contribution of each matrix in graph information recovering, we investigate the framework of structural complexity measurements of graphs, more specifically, entropic measurements, K. Anand et al., 2011. Inspired from information theory and statistical mechanics, these measures capture similarities and differences between networks and quantify the organization level of the underlying graph structure. Among them, we focus on Von Neumann (VN) entropy, which can be interpreted in some cases as a measure of regularity in graphs, Neumann, 1955,L. Han et al., 2012a.

In this chapter, we show the effectiveness of the JSS similarity measure for graphs classification. We highlight the graph's representation disparity between $\mathbf{A}$ and $\mathbf{L}$ matrices, illustrated via VN entropy measure. Integrated in an exponential kernel, the JSS measure shows promising results in real world graph data, built from chemical components and from real time series. These results are compared to the state-of-art graph kernels in terms of classification accuracy and computing CPU time.

4.2 A-Spectrum or L-Spectrum?

There is debate, in the literature, as to whether the eigenvalues of the adjacency matrix provide information about the graph properties. For example, Spielman argues that, even the adjacency matrix is the most natural matrix to associate with graph, it is least useful Spielman, 2004. Eigenvalues and eigenvectors are most meaningful when used to understand a natural operator or a natural quadratic form. The adjacency matrix provides neither. The same observation was made by Lau which points out that it is not clear that the eigenvalues should any information about the graph properties Lau, 2015. But they do, and interesting information are obtained from them as shown in the following sections.

4.3 Graphs representation in quantum domain

Characterisation of graphs using the number of nodes or edges can express some properties of graphs, but they are not sufficient to reflect their complexity. Tools developed in statistical mechanics can be exploited to provide more meaningful measures of graphs complexity, by mapping them into quantum states R. Alberta and A.L. Barabasi, 2002. The graph is viewed as a physical system. The first way of mapping is to use the node and edge to represent quantum state and the interaction between quantum states respectively X.B. Chen and Y.X. Yang, 2015. The second way, introduced by S.L. Braunstein et al., 2006, is based on a faithful mapping between discrete Laplacian and quantum states. Ideas from quantum information theory are also useful in the understanding of the structure of a graph J. Wang et al., 2017b,G. Bianconi and A.L. Barabasi, 2001. Quantum information is the physical information that is held in the state of a quantum system. Thus, network entropy has been extensively used to characterize the salient features of the structure of network systems arising in various domains such biology or physics and the social sciences. An example of measure to quantify the quantum information is the Von Neumann entropy, introduced to describe the uncertainty of a quantum state, Neumann, 1955. This measure distinguishes between different graph structures. For example, it is maximal for random graphs, minimal for complete ones and takes on intermediate values for star graphs L. Han et al., 2012b. In the following, basics of the graph representation in quantum domain are presented. We introduce the density matrix used in quantum mechanics to describe a quantum state and its equivalent in graph domain, the normalized Laplacian matrix. The idea of Hamiltonian operator on graph and its link to normalized Laplacian with associated Von Neumann entropy are introduced.

4.3.1 Density operator

In quantum mechanics, objects modify their states according to the presence or not of an observer. They take only one state among other possible states, called pure states. But, it happens also that the observer measures a mixture of states, then the object is said to be in a mixed state or in a superposition. Such phenomenon is described by a density operator (or density matrix) $\boldsymbol{\rho}$, in state space $\mathcal{H}$. This space is a complex Hilbert space of dimension n. The density operator describes a system whose state is an ensemble of pure quantum states $|\Psi_i\rangle$, each with probability p_i, M.A. Nielsen and I.L. Chuang, 2000. $|\Psi_i\rangle$ is an eigenstate of $\boldsymbol{\rho}$ and $\{|\Psi_i\rangle\}_{i=1}^{n}$ forms an orthonormal basis of $\mathcal{H}$. The density operator is defined as :

$$\boldsymbol{\rho} = \sum_{i=1}^{n} p_i \, |\Psi_i\rangle\langle\Psi_i| \, . \tag{4.1}$$

Note that if p_i is associated to eigenvalue λ_i, such decomposition is the standard spectral decomposition

of $\boldsymbol{\rho}$.

Necessary and sufficient conditions to describe a statistical ensemble with density operator $\boldsymbol{\rho}$ are as follows:

1. Normalization: $Trace(\boldsymbol{\rho}) = 1$
2. Positivity: $\boldsymbol{\rho} \geqslant 0$
3. Hermitian operator: $\boldsymbol{\rho} = \boldsymbol{\rho}^{\dagger}$

These conditions are helpful for deriving density operator from graph. Denote by $\mathcal{D}(\mathcal{H})$ the set of all density operators. Thus, $\boldsymbol{\rho} \in \mathcal{D}(\mathcal{H}))$ if and only if $\boldsymbol{\rho}$ verifies the conditions (1)-(3). With proper construction, the faithful mapping between quantum state and graph is established J.Q. Li and Y.X. Yan, 2015. Using relation (4.1), Severini et al. S.L. Braunstein et al., 2006,F. Passerini and S. Severini, 2009 have extended this idea to the graph domain by scaling the normalized version of $\mathbf{L}$ matrix by the number of nodes n of the graph.

4.3.2 Hamiltonian operator of a graph

In quantum mechanics, the Hamiltonian operator contains the operations associated with kinetic and potential energies of all the particles in a given system. Each particle is represented by a wave-function $\Psi(x,t)$ such that $\Psi^*(x,t)\Psi(x,t)$ is the probability of finding this particle at that position x and that time t. Let denote by m' the mass of this particle and by (x_1, x_2, x_3) its coordinates. The Hamiltonian operator describes the particle propagation, according the Schrödinger equation, and is given by :

$$\hat{H} = \underbrace{-\frac{\hbar^2}{2m'}\nabla^2}_{\substack{\text{Operator associated} \\ \text{with kinetic energy}}} + \underbrace{V(x)}_{\text{Potential energy}}, \tag{4.2}$$

where

$$\nabla^2 = \frac{\partial^2}{\partial x_i \partial x_i} \tag{4.3}$$

is the Laplacian and $\hbar$ is the reduced Planck constant, as defined in S.L. Braunstein et al., 2006 et al.

We can write down the characteristic equation for the energy in this representation; it takes the form

$$\hat{H}\Psi = E\Psi, \tag{4.4}$$

where E is the eigenvalue energy of the system for $\hat{H}$ corresponding to the eigenstate Ψ. The Hamiltonian acts upon the wave-function Ψ to generate the evolution of the wave-function in time and space. This equation yields the allowed energies and corresponding amplitude (wave) functions. Rearranging equation (4.2) in the form

$$\nabla^2\Psi + \frac{\hbar^2}{2m'}(E - V(x))\Psi = 0, \tag{4.5}$$

we obtain the Schrödinger equation. The solution of this equation is a wave that describes the quantum aspects of a system. We recall that the Schrödinger equation plays the role of the Newton's laws and conservation energy in classical mechanics, that predicts the future behaviour of a dynamic system, Lawden, 1995. While Schröedinger equation (4.5) predicts both the allowed energies of a system as well as the probability of finding a particle in a given region of space.

If we consider a graph as a physical system, there are number of ways to define the Hamiltonian operator of this graph. If we specify the node potential energy as $V(x) = \mathbf{D}$, and replace the Laplacian by its combinational counterpart $\mathbf{L} = \mathbf{D} - \mathbf{A}$, J. Wang et al., 2017b,J. Wang et al., 2016, we obtain :

$$\mathbf{L} = -\frac{\hbar^2}{2m'}\nabla^2, \text{then } \hat{H} = -\mathbf{A}. \tag{4.6}$$

This operator is often used in the Hückel molecular orbital method, A. Streitwieser, 1961.

Another way is to consider the graph to be in contact with a heat reservoir, J. Wang et al., 2017b. In this case the eigenvalues of the $\mathbf{L}$ matrix can be viewed as the energy eigenstates, and these determine the Hamiltonian operator and hence the relevant Schröedinger equation which governs a system of particles, J. Wang et al., 2017b. The graph is considered as a thermodynamic system composed of N particles with energy states given by the network Hamiltonian and immersed in a heat bath at temperature T, J. Wang et al., 2016. More precisely, the particles occupy the energy states of the Hamiltonian subject to thermal agitation by the heat bath.

If the eigenvalues of graph with the $\mathbf{L}$ matrix can be viewed as the energy eigenstates, we then take the kinetic energy operator $-\frac{\hbar^2}{2m'}\nabla^2$ to be the negative of the normalized adjacency matrix, i.e. $-\mathbf{A}$, and the potential energy $V(x)$ to be identity matrix $\mathbf{I}$. The Hamiltonian operator is viewed as the normalized form of Laplacian matrix on graph :

$$\hat{H} = \mathbf{I} - \tilde{\mathbf{A}} = \tilde{\mathbf{L}} \tag{4.7}$$

In this case the energy states of the network are then the eigenvalues of the Hamiltonian

$$\hat{H}\,|\Psi_i\rangle = \tilde{\mathbf{L}}\,|\Psi_i\rangle = E_i\,|\Psi_i\rangle \tag{4.8}$$

Furthermore, the density matrix $\boldsymbol{\rho}$ commutes with the Hamiltonian, that is the associated Poisson bracket is zero, J. Wang et al., 2017b :

$$[\hat{H}, \boldsymbol{\rho}] = [\tilde{\mathbf{L}}, \frac{\tilde{\mathbf{L}}}{\mid V \mid}] = 0 \tag{4.9}$$

which means that the network is in equilibrium when there is no change in the density matrix $\boldsymbol{\rho}$ which describes the system.

4.3.3 Von Neumann entropy

The entropy is an effective way to measure the uncertainty associated with classical probability and is also a means of characterizing the structure of graphs or complex networks. There have been many attempts to compute the entropy of graph, K. Anand et al., 2011, S. Perseguers et al., 2009, D. Hu et al., 2017. When it comes to quantum world, the Von Neumann entropy corresponding to the entropy of quantum state is used, Neumann, 1955. This entropy was originally introduced by Von Neumann around 1927 for proving the irreversibility of a quantum measurement processes in mechanics, Neumann, 1955, D. Hu et al., 2017. The Von Neumann entropy is a quantitative measure of mixedness of the density matrix $\boldsymbol{\rho}$. The interpretation of the scaled normalized Laplacian matrix as a density operator of a physical system, opens up the possibility of characterizing a graph using the Von Neumann entropy from the quantum information theory, J. Wang et al., 2016. More precisely, it is natural to interpret the Von Neumann entropy of such density matrix as the Von Neumann entropy of the graph, with a view towards characterizing the information content of the graph, M. Dairyko et al., 2017,S.L. Braunstein et al., 2006.

The Von Neumann entropy of $\boldsymbol{\rho} \in \mathcal{D}(\mathcal{H}))$ is defined as:

$$S(\boldsymbol{\rho}) = -K_B Trace(\boldsymbol{\rho} \log \boldsymbol{\rho}), \tag{4.10}$$

where $Trace(B)$ is the trace of matrix B and K_B is the Boltzmann constant. The density matrix $\boldsymbol{\rho}$ of a graph G has a zero eigenvalue whose multiplicity is equal to the number of components of G. S.L. Braunstein et al., 2006 defined the density matrix of a graph G as:

$$\boldsymbol{\rho} = \frac{\mathbf{L}}{\sum\limits_{i} d(v_i)}, \tag{4.11}$$

Suppose that $\lambda_1 \geq \lambda_2 \geq, \ldots, \lambda_n = 0$ are the eigenvalues of the density operator $\boldsymbol{\rho}$ describing the graph

(or quantum-mechanical system), then :

$$S(\boldsymbol{\rho}) = -\sum_{i=1}^{n} \lambda_i \log_2 \lambda_i \tag{4.12}$$

is called the Von Neumann entropy of graph G. By convention $0 \log_2 0 = 0$. The Von Neumann entropy can be viewed as the Shannon entropy of the probability distribution represented by the eigenvalues of the density operator. It can be interpreted as a measure of regularity, S.L. Braunstein et al., 2006, F. Passerini and S. Severini, 2009, and can be used as a measure of graph complexity, L. Han et al., 2012b. A graph G has zero Von Neumann entropy if and only if one eigenvalue is 1 and the others are 0.

4.4 Relationship Between A and L via VN-Entropy

Since the Von Neumann entropy is interpreted as a measure of the information content in a graph G, we wonder about the contribution of each matrix (**A** and **L**) in the measured quantity of information. A part of the answer comes from quantum perturbation theory, where the sensitivity of eigenvalues to density matrix perturbations has been studied. Some of these studies have been extended by Chen to include the case of perturbations on VN-entropy, Chen, 2010. This opens the way to project the study on to the general case of graphs. The goal is to determine the cost of structural modifications of the graph in terms of information quantity, and to clearly understand the relationship between **A** and **L** in a purely entropic framework. To this end, **A** matrix is perturbed and the VN-entropy of the perturbed density matrix $\boldsymbol{\rho}$ is then computed.

Proposition 1. *Let G be a weighted graph whose the unperturbed density matrix is $\boldsymbol{\rho_0}$. If $\boldsymbol{\rho_0}$ undergoes a perturbation $\boldsymbol{\rho}^*$, the VN-entropy of the resulting perturbed matrix, $\boldsymbol{\rho} = \boldsymbol{\rho}_0 + \xi\ \boldsymbol{\rho}^*$, expanded up to the second order is given by*

$$S(\boldsymbol{\rho}) = S(\boldsymbol{\rho}_0) + \eta_{Diagonal}(\boldsymbol{\rho}^*) + \eta_{Off-Diagonal}(\boldsymbol{\rho}^*), \tag{4.13}$$

where

$$\begin{aligned} \eta_{Diagonal}(\boldsymbol{\rho}^*) &= -\xi \sum_n {\rho^*}_{nn} \log_2 \tilde{\lambda}_n - \frac{1}{2}\xi^2 \sum_n \frac{({\rho^*}_{nn})^2}{\tilde{\lambda}_n} + o(\xi^2) \\ \eta_{Off-Diagonal}(\boldsymbol{\rho}^*) &= -\xi^2 \sum_n \sum_{m \neq n} \frac{|{\rho^*}_{nm}|^2}{\tilde{\lambda}_n - \tilde{\lambda}_m} \log_2 \tilde{\lambda}_n + o(\xi^2), \end{aligned} \tag{4.14}$$

with ξ is a scalar parameter supposed to be small, and $\tilde{\lambda}_n \neq \tilde{\lambda}_m\ \forall\ n \neq m$ are the eigenvalues of $\boldsymbol{\rho}_0$.

Proof :

Since $\boldsymbol{\rho}$ is a density matrix, the perturbations are introduced in such way that

$$Tr(\boldsymbol{\rho}) = 1 \text{ and } \boldsymbol{\rho} = \boldsymbol{\rho}^T. \tag{4.15}$$

Restricting Taylor expansion to the second order, L. Han et al., 2012a, the corrections to the eigenvalues are written as :

$$\tilde{\lambda}_n^{(t)} = \tilde{\lambda}_n + \xi\tilde{\lambda}_n^{(1)} + \xi^2\tilde{\lambda}_n^{(2)} + o(\xi^2), \tag{4.16}$$

where the first and second order eigenvalue perturbations are given by

$$\tilde{\lambda}_n^{(1)} = {\rho^*}_{nn} \quad , \quad \tilde{\lambda}_n^{(2)} = \sum_{m\neq n} \frac{|{\rho^*}_{nm}|^2}{\tilde{\lambda}_n - \tilde{\lambda}_m}. \tag{4.17}$$

Therefore, the perturbed VN-entropy can be written as

$$S(\boldsymbol{\rho}) = -Tr(\boldsymbol{\rho}\log_2\boldsymbol{\rho}) \approx -\sum_n \tilde{\lambda}_n^{(t)}\log_2\tilde{\lambda}_n^{(t)}. \tag{4.18}$$

On the other hand, the Taylor expansion of $S(\boldsymbol{\rho})$ around zero up to second order is written as

$$S(\boldsymbol{\rho}) \approx S(\boldsymbol{\rho}_0) + \xi\frac{dS(\boldsymbol{\rho}_0)}{d\xi} + \frac{1}{2}\xi^2\frac{d^2S(\boldsymbol{\rho}_0)}{d\xi^2} + o(\xi^2). \tag{4.19}$$

Chen, 2010 gives the first and second order derivatives of $S(\boldsymbol{\rho}_0)$:

$$\frac{dS(\boldsymbol{\rho}_0)}{d\xi} = -\sum_n {\rho^*}_{nn}\log_2\tilde{\lambda}_n, \tag{4.20}$$

$$\frac{d^2S(\boldsymbol{\rho}_0)}{d\xi^2} = -\sum_n \frac{({\rho^*}_{nn})^2}{\tilde{\lambda}_n} - 2\sum_n \tilde{\lambda}_n^{(2)}\log_2\tilde{\lambda}_n. \tag{4.21}$$

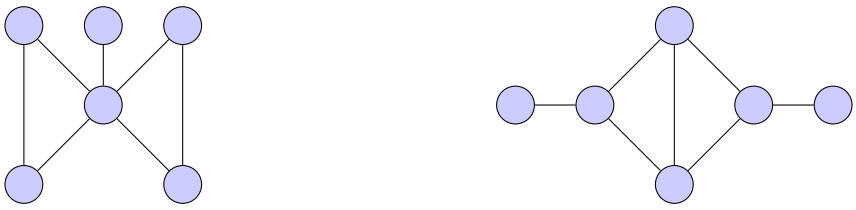
By substituting (5.12), (5.16) and (5.17) in equation (5.14), we get

$$\begin{aligned} S(\boldsymbol{\rho}) \approx S(\boldsymbol{\rho}_0) \quad &- \underbrace{\xi\sum_n {\rho^*}_{nn}\log_2\tilde{\lambda}_n - \frac{1}{2}\xi^2\sum_n \frac{({\rho^*}_{nn})^2}{\tilde{\lambda}_n}}_{\eta_{Diagonal}} \\ &+ \underbrace{\xi^2\sum_n\sum_{m\neq n}\frac{|{\rho^*}_{nm}|^2}{\tilde{\lambda}_n - \tilde{\lambda}_m}\log_2\tilde{\lambda}_n}_{\eta_{Off-Diagonal}} + o(\xi^2). \end{aligned} \tag{4.22}$$

A careful examination of equation (4.22) shows that the corrective terms $\eta_{Diagonal}$ and $\eta_{Off-Diagonal}$, include respectively both diagonal and off-diagonal perturbation elements. Since changes in edge weights of $\mathbf{A}$ lead to changes in node degrees, perturbations affect at first place off-diagonal elements of $\mathbf{A}$. Thus, it is expected that changes in weights introduce only off-diagonal elements in the entropy expression (4.22). But, diagonal elements related to node degrees changes appear also. Due to the linear relationship $\mathbf{L} = \mathbf{D} - \mathbf{A}$, these diagonal elements are introduced to $\mathbf{L}$ via $\mathbf{D}$ and thus they may explain why $\mathbf{L}$ matrix is more sensitive to structural changes of the graph than $\mathbf{A}$.

4.5 Graphs Cospectrality Issue

The spectrum of different matrix representations associated to graphs holds a variety of informations that differs from one matrix to another, A.E. Brouwer and W.H. Haemers, 2011. This is seen in the existence of cospectral graphs, or graphs that share the same spectrum for a particular matrix, whether **A**, **L** or others. They are called non-DS graphs referring to "Non Determined by the Spectrum". Among many graph families, Schwenk, 1973 stated that almost all trees are non-DS graphs in respect to **A**. Furthermore, C.D. Godsil and B.D. McKay, 1982 proposed a method based on edge switching to make two non-isomorphic graphs **A**-cospectral. Likewise, many families of **L**-cospectral graphs were defined in the literature, such those defined by Merris, 1997. W.H. Haemers and E. Spence, 2004 have invetigated the cospectrality of graphs up to size 11, extending a previous survey done by C.D. Godsil and B.D. McKay, 1982. Above 11 nodes, it becomes computationally costly to enumerate all possible cospectral graphs. In their study, they considered the adjacency matrix **A**, the laplacian **L** and the signless laplacian $|\mathbf{L}|$. A part of these results is summarized in Table 4.1. They show that the adjacency matrix appears to be the worst representation allowing a large number of cospectral graphs. The laplacian is superior and the signless laplacian even better. The signless laplacian, laplacian and adjacency matrices produce 3.8%, 9% and 21% respectively of cospectral graphs with 11 nodes.



(a) (b)

Figure 4.1: Two **A**-cospectral graphs: $\sigma_a(\mathbf{A}) = \sigma_b(\mathbf{A}) = [-1.9, -1, -1, 0.2, 1, 2.7]$, $\sigma_a(\mathbf{L}) = [0, 1, 1, 3, 3, 6]$, $\sigma_b(\mathbf{L}) = [0, 0.5, 1.2, 3.4, 4, 4.7]$, where $\sigma(\mathbf{B})$ means the eigen spectrum of the matrix **B**.

We show in Figure 4.2 an example of two graphs that share the same laplacian spectrum despite of their structural differences. As cospectrality appears to be a challenge to distinguish graphs via their spectra, it remains nevertheless surmountable by using a variety of spectral metrics simultaneously in the graph identification process. Figure 4.1 shows that it is possible to recognize **A**-cospectral graphs via their **L** spectrum, and inversely in Figure 4.2. This convinces us that combining both spectra of **A** and **L** in JSS measure for comparing graphs.

(a) (b)

Figure 4.2: Two **L**-cospectral graphs: $\sigma_a(\mathbf{L}) = \sigma_b(\mathbf{L}) = [0, 0.7, 2, 3, 3, 5.2]$, $\sigma_a(\mathbf{A}) = [-2.1, -1, -0.5, 0, 1.2, 2.5]$, $\sigma_b(\mathbf{A}) = [-2.5, -0.7, 0, 0, 0.7, 2.5]$, where $\sigma(\mathbf{B})$ means the eigen spectrum of the matrix **B**.

Table 4.1: Number of cospectral graphs when using different spectra in combination, knowing that the number of possible n-node undirected graphs is $2^{n(n-1)/2}$.

Graph Size (n)	**A** or **L**	**A** and **L**	**L** and | **L** |	| **L** | and **A**
6	112	**0**	0	0
7	853	**0**	16	0
8	11117	**0**	232	0
9	261080	**82**	4139	8
10	11716571	**13864**	107835	10716

4.6 Joint Spectral Similarity Measure

Let $\mathcal{G}$ be a set of undirected graphs, and consider two graphs $\mathrm{G}_1(\mathbf{A}_1, \mathbf{L}_1)$ and $\mathrm{G}_2(\mathbf{A}_2, \mathbf{L}_2) \in \mathcal{G}$ with $\mathbf{A}_1$, $\mathbf{A}_2$ their adjacency matrices and $\mathbf{L}_1$, $\mathbf{L}_2$ their laplacian matrices where (λ_{1i},λ_{2i}) and (μ_{1i},μ_{2i}) are the eigenspectra of their Laplacian and adjacency matrices ordered in descending order. Information about the degree distribution is encoded mainly in the eigenvalues of **L** as well as the number of components of the graph, while information about walks, paths and the bipartition of the graph are in the eigenvalues of **A**, E.R. Van Dam and W.H Haemers, 2003. Also to avoid the problem of cospectrality when comparing graphs via **A** and **L** spectra, we quantify the associated spectral similarity of the graphs as the trade-off between them. The introduced JSS measure aims to exploit both **A** and **L** spectral informations to better discriminate graphs. This measure, sum of two weighted spectral components, is given by the following convex linear combination :

$$\mathrm{JSS}(\mathrm{G}_1, \mathrm{G}_2) = \alpha \mathrm{JSS}_{\mathbf{A}}(\mathrm{G}_1, \mathrm{G}_2) + (1-\alpha)\mathrm{JSS}_{\mathbf{L}}(\mathrm{G}_1, \mathrm{G}_2) \tag{4.23}$$

with $\alpha \in [0, 1]$, a weighting factor and where the components are given by :

$$\mathrm{JSS}_{\mathbf{L}}(\mathrm{G}_1, \mathrm{G}_2) = \sum_{i=1}^{k} (\lambda_{1i} - \lambda_{2i})^2, \; \mathrm{JSS}_{\mathbf{A}}(\mathrm{G}_1, \mathrm{G}_2) = \sum_{i=1}^{k} (\mu_{1i} - \mu_{2i})^2 \tag{4.24}$$

where $k = \min(N_1, N_2)$ and N_1, N_2 are the numbers of eigenvalues corresponding to each graph. 'k' represents the important common eigenvalues between the graphs, however, this criterion is particularly suitable for graphs of similar and comparable sizes. The weighting factor α controls the significance of each distance and allows more importance to be given to the **A**-spectral distance or to the **L**-spectral distance.

4.7 Experimental Results

We illustrate the performance of the JSS measure on real world graphs from bioinformatics and on conceptual graphs obtained by mapping time series to the graph domain using the *Graph Visibility* (VG) algorithm proposed by L. Lacasa et al., 2008. There are altogether nine benchmark real data sets used in our experiment.

4.7.1 Bioinformatics Data

We use the same graph datasets presented in chapter 3, which are derived from chemical/biological molecular databases. As a reminder, **MUTAG** is a dataset of 188 mutagenic aromatic and heteroaromatic nitro compounds labeled according to whether or not they have mutagenic effect on the Gram-negative bacterium *Salmonella typhimurium*, A.K. Debnath et al., 1991. **NCI1** and **NCI109** represent two data sets of chemical compounds screened for activity against non-small cell lung cancer and ovarian cancer cell lines, N. Wale and G. Karypis, 2006. **ENZYMES** is a data set of protein tertiary structures obtained by K.M. Borgwardt et al., 2005, consisting of 600 enzymes from **BRENDA** enzyme database, I. Schomburg et al., 2004, where the task is to assign each enzyme to one of the 6 top-level classes. **D&D** is a data set of 1178 protein structures, P.D. Dobson and A.J. Doig, 2003, the classification task is to distinguish protein structures between enzymes and non-enzymes.

4.7.2 Time Series Data

Four problems are considered namely, A. Bagnall et al., 2002:

- **Computers**: This dataset was made from data recorded as part of government sponsored study called Powering the Nation. The idea was to collect behavioural data in order to know how consumers use electricity within the home to help reduce the UK's carbon footprint. The data contains readings from 251 households, sampled in two-minute intervals over a month. Each series is length 720 (24 hours of readings taken every 2 minutes). Here the purpose is to know if it is a Desktop or a Laptop computer. We specify that the data was prepared and donated by J. Lines and A. Bagnall, 2015.

- **ToeSegmentation1**: This data is derived from the CMU Graphics Lab Motion Capture Database(CMU). Motions in the database containing the keyword walk are classified by their motion descriptions into two categories. The first category is the normal walk, with only walk in the motion descriptions. The other is the abnormal walk, with the motion descriptions containing hobble walk, walk wounded leg, walk on toes bent forward, hurt leg walk, drag bad leg walk, or hurt stomach walk. In the abnormal walks, the actors are pretending to have difficulty walking normally. ToeSegmentation1 is the X-Axis. The purpose is to classify walks by their nature, whether they are normal or abnormal, L.Ye and E. Keogh, 2011.

- **SonyAIBORobotSurface1**: is a dataset donated by Manuela Veloso and Douglas Vail of Carnegie Mellon University and it is used by A. Mueen et al., 2011. AIBO (Artificial Intelligence Robot) is a series of robotic pets designed and manufactured by Sony. Sony announced the first prototype of AIBO in mid-1998, and in 2006, AIBO was added into the Carnegie Mellon University Robot Hall of Fame. The robot has roll/pitch/yaw accelerometers whose task is to detect the surface being walked on whether is cement or carpet, these time series representing the X-axis records.

- **Lightning2**: is a dataset containing signals captured by the FORTE satellite, that detects transient electromagnetic events associated with lightning using a suite of optical and radio-frequency (RF) instruments. Data is collected with a sample rate of 50 MHz for 800 microseconds. Spectrograms were calculated from the input data and then they were collapsed in frequency to produce a power density time series, with 3181 samples in each time series, these are smoothed to produce series of length 637. Here, the aim is to classify power densities into two different categories of lightning, D. Eads et al., 2002.

4.7.3 Convert Time Series to Graphs

In order to map time series into complex networks on the graph domain, we use *Visibility graph* (VG) algorithm, proposed by L. Lacasa et al., 2008. The advantage is that the obtained network inherits many interesting properties, and reveals nontivial information about the series itself. VG algorithm is becoming an emerging technique for the analysis of long-range dependency, fractality and dynamical properties of time series data. S. Supriya et al., 2016, used it to study EEG time series classification problem, and they showed that VG is efficient to distinguish different dynamical structures in the EEG recording of healthy and epileptic patients. In our experiment, we use the same weighting and graph building strategy defined by S. Supriya et al., 2016. VG algorithm determines the connections between nodes and the weights of the graph. Every node of the graph corresponds, in the same order, to a sample from the series data, and two nodes are connected, if visibility exists between the corresponding samples. More formally: two

arbitrary time series samples (t_a, y_a) and (t_b, y_b) will have visibility and become two connected nodes in the associated graph, if any other sample (t_c, y_c) placed between them fulfills:

$$y_c < y_b + (y_a - y_b)\frac{t_b - t_c}{t_b - t_a}. \tag{4.25}$$

S. Supriya et al., 2016 consider the edge weight between two nodes as the absolute value of the angle between the straight line that connects them and the horizontal axis, and is denoted by:

$$w_{ab} = \left| arctan\left(\frac{y_b - y_a}{t_b - t_a}\right)\right|. \tag{4.26}$$

The weights w_{ab} are the entries of the adjacency matrix ($\mathbf{A}$) corresponding to the constructed visibility graph.

An example of a visibility graph is given in figure 4.3.

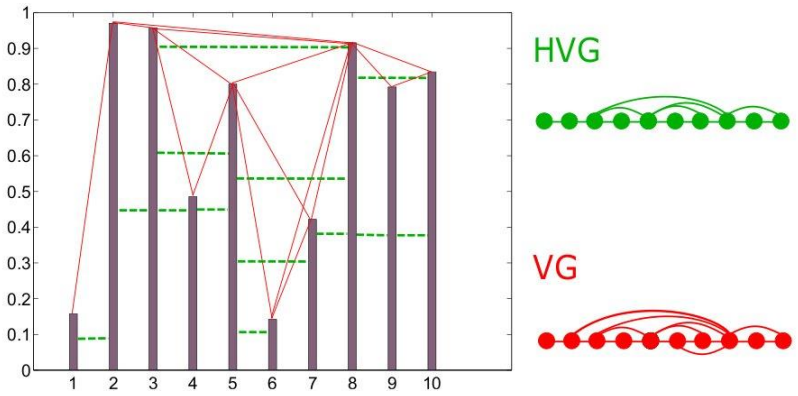


Figure 4.3: Example of a visibility graph, Lacasa, 2009. The green is built using Horizontal Visibility Graph (HVG) algorithm, B. Luque et al., 2009, and the red one using the general Visibility Graph (VG) algorithm, L. Lacasa et al., 2008.

4.7.4 Experimental Configuration

As the experiment in chapter 3, the Joint Spectral Similarity measure is integrated in an exponential function giving the following distance between the graphs G_i, G_j:

$$K_{i,j} = \exp(-\gamma\, s(G_i, G_j)), \tag{4.27}$$

where K is the kernel matrix and $s(G_i,G_j)$ can be here the JSS, JSS_A or JSS_L measure, and the parameter γ is a smoothing factor which we set to one. The kernel matrix is integrated in a support vector machine (SVM). Then, 10-fold cross-validation strategy is performed using 9 folds for training and 1 for testing. Datasets are randomly shuffled before partitioning and the whole experiment is repeated 10 times to avoid

random effects of fold assignments. Average classification accuracies and their corresponding standard deviations are summarized in Tables 4.2 and 4.4. For molecular data, the measures JSS, $JSS_{\mathbf{A}}$ and $JSS_{\mathbf{L}}$ are compared to some kernels of the literature in terms of prediction accuracy and computation runtime. For time series data, only prediction accuracies are reported. Otherwise, the well-known graph kernels are tested: those based on walks, sub-trees and Weisfeiler-Lehman isomorphism. Else, the measures are compared to the fast geometric random walk kernel S.V.N. Vishwanathan et al., 2010, that counts common labelled walks and to p-random walk kernel that compares random walks up to length p in two graphs, H. Kashima et al., 2003. From sub-tree kernels, we chose Ramon-Gärtner kernel, T. Gärtner et al., 2003b, which compares all pairs of nodes from two graphs by iteratively comparing their neighbourhoods. From Weisfeiler-Lehman kernels, we picked up the Weisfeiler-Lehman edge kernel, N. Shervashidze et al., 2011: it counts matching pairs of edges with identically labeled endpoints in two graphs. For molecular data, computing set-up, accuracy and runtime values of the benchmark kernels are performed by N. Shervashidze et al., 2011. We followed the same procedures to compute random walk and Weisfeiler-Lehman edge kernels for time series data, without Ramon-Gärtner and p-random walk kernels because of runtime constraint. The time series results are also compared to Linear and RBF kernels, applied directly on the samples of time series without converting them to graphs. As reported in Table 4.3, runtime in minutes and seconds of our method is measured using Anaconda2 4.1.1 Python 2.7.12 Lab installed on a Windows machine with 3 GHz Intel 8-Core processor and 16GB of RAM.

4.7.5 Results Analysis

As expected, the obtained results confirm the unequal contribution of both $\mathbf{A}$ and $\mathbf{L}$ matrices and their overlapping in terms of graphs representation. The reported results in Tables 4.2 and 4.4, and in figure 4.4 show that the achieved classification accuracies of JSS measure are promising with respect to some state-of-the-art methods. The JSS based kernel performs well in the majority of datasets. In molecular data, it reaches second best accuracy in all datasets (MUTAG, NCI1, NCI109, ENZYMES and D&D) compared to the benchmark kernels. But in terms of CPU time, the JSS based kernel is faster than other kernels in many cases, such as in MUTAG, ENZYMES and mostly in D&D, where JSS kernel performs almost as well as Weisfeiler-Lehman edge kernel (75.75 Vs 77.95), while execution time is 600 times faster. In Table 4.4, the JSS based kernel provides good results applied to time series classification problems. Especially in the Lightning2, ToeSegmentation1 and Computers databases, it performs better than other graph kernels, and even better than kernels applied directly to time series. The representation of time series as VG allowed a clear improvement of the classification accuracy, approaching 12% for ToeSegmentation1, 8% for Lightning2 and 6% for Computers. The reported results, on different data sets with varying complexity and heterogeneity, in terms of classification accuracy and computational cost, demonstrate

the effectiveness and the interest of the proposed JSS measure. This spectral similarity generalizes the spectral distance between graphs based on purely $\mathbf{A}$ ($\mathrm{JSS}_{\mathbf{A}}$) or $\mathbf{L}$ ($\mathrm{JSS}_{\mathbf{L}}$) matrix.

As reported in Tables 4.2 and 4.4, and figure 4.4, JSS is well sensitive to these graph properties and allows to effectively handle them. Figure 4.4 shows that the JSS-based method highly outperforms the method purely based on $\mathrm{JSS}_{\mathbf{A}}$ or $\mathrm{JSS}_{\mathbf{L}}$. Indeed a careful examination of figure 4.4 shows that for both $\alpha = 0$ and $\alpha = 1$, corresponding respectively to $\mathrm{JSS}_{\mathbf{L}}$ and $\mathrm{JSS}_{\mathbf{A}}$, neither of $\mathrm{JSS}_{\mathbf{A}}$ nor $\mathrm{JSS}_{\mathbf{L}}$ is able to perform better than JSS. However, overall, $\mathrm{JSS}_{\mathbf{A}}$ achieves better results than $\mathrm{JSS}_{\mathbf{L}}$ and this can be attributed, as expected, to the less sensitivity of $\mathbf{A}$ to structural changes, compared to $\mathbf{L}$ matrix, but more efficient for graph discrimination. According to these results, the weighting parameter lies in $]0; 1[$, showing that $\mathbf{L}$ and $\mathbf{A}$ are complementary and thus carry different information about the underlying graph. For all data sets, we have the parameter $\alpha \neq 0.5$, which indicates that $\mathrm{JSS}_{\mathbf{L}}$ and $\mathrm{JSS}_{\mathbf{A}}$ are unequally contributed and highlights that the $\mathbf{A}$ and $\mathbf{L}$ representation matrices recover different structures of the graph. Also, these results show that JSS effectively captures information conveyed by $\mathbf{A}$ and $\mathbf{L}$ matrices. The fact that the assigned weighting parameter α varies from one data set to another emphasizes that each graph has its own structure and also that both $\mathbf{A}$ and $\mathbf{L}$ matrices convey different information.

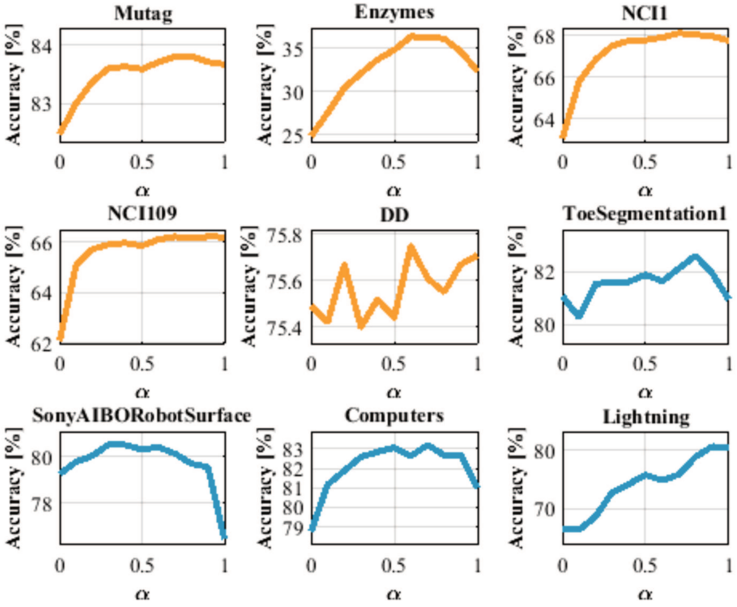


Figure 4.4: Accuracy variation of JSS based kernel as function of α.

4.8 Conclusion

The representation of graphs using matrices plays an important role in graph spectral theory and in many other applications dealing with graphs. In this chapter, a new joint spectral similarity (JSS) measure for graphs classification is introduced. We have shown that both adjacency and Laplacian matrices carry different structures information of the underlying graph. The adjacency matrix characterizes the topolog-

Table 4.2: Classification accuracy on some bioinformatics data (± standard deviation).

Method/Data	MUTAG	NCI1	NCI109	ENZYMES	D&D
JSS_A	83.67 ±0.22	67.74 ±0.14	66.21 ±0.28	32.35 ±1.15	75.71 ±0.14
JSS_L	82.48 ±0.36	63.07 ±0.25	62.07 ±0.26	24.86 ±0.93	75.49 ±0.24
JSS	83.81 ±0.48	68.10 ±0.16	66.22 ±0.13	36.35 ±0.90	75.75 ±0.19
Ramon and Gärtner [T. Gärtner et al., 2003b]	85.72 ±0.49	61.86 ±0.27	61.67 ±0.21	13.35 ±0.87	57.27 ±0.07
p-random walk [H. Kashima et al., 2003]	79.19 ±1.09	58.66 ±0.28	58.36 ±0.94	27.67 ±0.95	66.64 ±0.83
Random walk [S.V.N. Vishwanathan et al., 2010]	80.72 ±0.38	64.34 ±0.27	63.51 ±0.18	21.68 ±0.94	71.70 ±0.47
Weisfeiler-Lehman edge [N. Shervashidze et al., 2011]	81.06 ±1.95	84.37 ±0.30	84.49 ±0.20	53.17 ±2.04	77.95 ±0.70

Table 4.3: CPU runtime for kernel computation on some bioinformatics data.

Method/Data	MUTAG	NCI1	NCI109	ENZYMES	D&D
JSS_A	0.32″	2′29″	2′27″	3″	3′24″
JSS_L	0.88″	2′28″	2′30″	3.50″	3′34″
JSS	1.30″	4′14″	4′19″	5.88″	6′53″
Ramon and Gärtner [T. Gärtner et al., 2003b]	40′6″	81 days	81 days	38 days	103 days
p-random walk [H. Kashima et al., 2003]	4′42″	5 days	5 days	10′	4 days
Random walk [S.V.N. Vishwanathan et al., 2010]	12″	9 days	9 days	12′19″	48 days
Weisfeiler-Lehman edge [N. Shervashidze et al., 2011]	3″	1′5″	58″	11″	3 days

ical graph complexity in terms of connections between nodes and their intensities, and also underscores the local cohesiveness of nodes. These properties explain why the good classification accuracies achieved by JSS measure are more attributed to adjacency matrix ($\alpha > 0.5$). Through VN entropy, it is easy to see that Laplacian matrix brings out changes in node degrees information. Furthermore, this matrix is well suited to recover information about clusters of the graph and thus capture its inherent structure. The obtained results highlight the fact that JSS combines both advantages of Laplacian and adjacency matrices. Also, these findings confirm that these matrices contribute unequally and emphasize the fact

Table 4.4: Classification accuracy on some time series data (± standard deviation).

Method/Data	ToeSegmentation1	SonyAIBORobotSurface1	Computers	Lightning2
$\mathbf{JSS_A}$	80.94 ±0.82	76.50 ±0.41	80.97 ±0.67	80.52 ±1.76
$\mathbf{JSS_L}$	81.08 ±0.52	79.01 ±0.46	78.76 ±0.42	66.63 ±1.07
JSS	82.61 ±0.79	80.52 ±0.34	83.21 ±0.60	80.68 ±1.71
Random walk [S.V.N. Vishwanathan et al., 2010]	59.03 ±0.64	55.70 ±0.48	60.50 ±0.01	60.30 ±2.30
Weisfeiler-Lehman edge [N. Shervashidze et al., 2011]	61.08 ±1.68	56.46 ±1.01	76.22 ±1.02	49.15 ±2.65
Linear-SVM	54.10 ±2.05	97.43 ±0.23	53.42 ±1.67	59.54 ±3.21
RBF-SVM	71.11 ±1.64	99.08 ±0.07	56.28 ±0.65	72.99 ±1.06

that they represent differently information about structures of the underlying graph. Additionally, these results show the interest of the VG approach for classification of time series. As a result of this work, we hope to have increased the awereness about the importance of the properly choice of the representation matrix for graph spectral analysis purposes. Even the JSS measure handles cospectral graphs with respect to both $\mathbf{A}$ and $\mathbf{L}$, it can be extended to the case of graphs that are cospectral in regard to a large class of graph representation matrices. At last, the optimal value of α is in general not known and is determined only through experimentation. It is suitable to develop a strategy for finding automatically its optimal value.

CHAPTER 5

Graph Vulnerability in the Sense of Von Neumann's Entropy

5.1 Introduction

Real-world networks are critical infrastructure systems that function collaboratively and synergistically to produce essential services and facilitate human interaction C. Nan and I. Eusgeld, 2011,S. Wang et al., 2018. Examples of such systems include powers systems, water supply systems, natural gas supply systems, transportation systems and telecommunication systems S. Wang et al., 2018. The growth in generation and demand without networks expansion further increases of these systems and creates various security issues S. Gupta et al., 2018. In general, such systems are complex interconnected networks. This, is the case of the power networks that are particularly important because a lot of infrastructure systems are very dependent on the reliable supply of electricity to ensure their normal operations S. Wang et al., 2018,M. Ouyang, 2014. These networks are threatened by many factors which increase their vulnerabilities X. Yuan et al., 2017,J. Gao et al., 2016. Recent events, such as 2012 India blackout and 2003 North American blackout have highlighted the vulnerability of power networks and thus, the necessity for their assessment is of great importance. Nowadays the issue of vulnerability and protection of critical infrastructure is attracting a great deal of attention of scientific community. In general, a critical infrastructure system is represented as a graph in which nodes represent the main components of the network (power plants,...) and edges are the physical connections among them (electrical lines,...) P.C. Crucitti et al., 2005. The topology of the network or the graph determines an influence structure among the nodes or the agents S. Segarra and A. Ribeiro, 2016. During a vulnerability assessment of networks, graph theory techniques allow both representation and analysis; the theory being based on a set of measurements that evaluates networks and includes spectral measures J.A. Gutierrez-Perez et al.,

2013. Metrics derived from the spectrum of the network adjacency matrix quantify network invariants, thereby revealing pertinent information about the well-connectedness or not of the network in terms of connectivity intensity and failure tolerance J.A. Gutierrez-Perez et al., 2013. Following the graph-based approach, different strategies have been proposed with the purpose to measure the vulnerability of the graph or to find the best nodes to immunize (or equivalently, remove) to make the remaining nodes to be most robust to virus attack C. Chen et al., 2016. In the literature, for example, failure and attacks have been simulated as the removal of a certain percentage either of nodes R. Albert et al., 2000,P. Holme et al., 2002,P. Crucitti et al., 2003,R. Albert et al., 2004,P. Crucitti et al., 2004 or edges P. Holme et al., 2002,A.E. Motter et al., 2002 of the network. Nodes immunization is essential to safeguard network systems against, for example, virus attacks and its propagation. This requires the quantification of importance of individual node or group of nodes in terms of their contribution towards vulnerability. A simple metric to judge the overall graph vulnerability is the one based on the largest (first) eigenvalue λ of adjacency matrix of the graph C. Chen et al., 2016,K. Kanwar et al., 2017. The larger λ is, the more vulnerable the whole graph is. However, this global metric or score cannot be used for identifying or localizing a vulnerable edge or a group of edges that are vulnerable of the graph. The challenge behind this problem is to measure the vulnerability of each edge and to provide a vulnerability map of the graph that helps to find the effective immunization strategy to be applied. By using results from the theory of matrix perturbation combined with Von Neumann entropy, we propose a metric to quantify the vulnerability of each edge. The aim is also to guard high-risk edges. The vulnerability is measured by the Von Neumann entropy distortion produced by each perturbed edge.

5.2 Eigenvalue Sensitivies to Matrix Perturbations

Consider a square matrix $\mathbf{A} \in \mathbb{M}_n$, associated to a graph or a physical system with distinct eigenvalues λ_i, $(i = 1, 2, ..., n)$ arranged in a column vector. We would like to understand how eigenvalues of $\mathbf{A}$ change under perturbations of its elements. Thus it is useful to see how an eigenvalue λ_i is sensitive to changes of an individual element a_{kl} of $\mathbf{A}$, $(k, l = 1, 2, ..., n)$. This can be done by perturbing the matrix element a_{kl} by a quantity Δa_{kl}. We begin with equations relating the right eigenvalues λ_i to their corresponding right eigenvectors $\mathbf{u_i}$ of $\mathbf{A}$:

$$\mathbf{A}\mathbf{u_i} = \lambda_\mathbf{i}\mathbf{u_i}. \tag{5.1}$$

Similarly, the left eigenvalues λ'_j are related to the left eigenvectors $\mathbf{w_j}$ of $\mathbf{A}$ (Or to the right eigenvectors

of $\mathbf{A}^T$) by the equation :

$$\mathbf{A}^T\mathbf{w_j} = \lambda'_\mathbf{j}\mathbf{w_j}. \tag{5.2}$$

The eigenvectors $\mathbf{u_i}$ and $\mathbf{w_j}$ are orthogonal and can be scaled such that:

$$\mathbf{w_j}^T\mathbf{u_i} = \delta_{ij}, \tag{5.3}$$

where δ_{ij} is the Kronecker function. Note that the left and right eigenvalues/eigenvectors are the same for symmetric matrices. By differentiating both sides of equation (5.1) with respect to a parameter β of interest, we get:

$$\frac{\partial\mathbf{A}}{\partial\beta}\mathbf{u_i} + \mathbf{A}\frac{\partial\mathbf{u_i}}{\partial\beta} = \frac{\partial\lambda_i}{\partial\beta}\mathbf{u_i} + \lambda_i\frac{\partial\mathbf{u_i}}{\partial\beta}. \tag{5.4}$$

Then pre-multiplying equation (5.4) by $\mathbf{w_i}^T$ and using equation (5.3) and the transpose of equation (5.2), we obtain :

$$\mathbf{w_i}^T\frac{\partial\mathbf{A}}{\partial\beta}\mathbf{u_i} + \lambda_i\mathbf{w_i}^T\frac{\partial\mathbf{u_i}}{\partial\beta} = \mathbf{w_i}^T\frac{\partial\lambda_i}{\partial\beta}\mathbf{u_i} + \lambda_i\mathbf{w_i}^T\frac{\partial\mathbf{u_i}}{\partial\beta}, \tag{5.5}$$

and after simplification, we get :

$$\mathbf{w_i}^T\frac{\partial\mathbf{A}}{\partial\beta}\mathbf{u_i} = \frac{\partial\lambda_i}{\partial\beta}. \tag{5.6}$$

Equation (5.6) implies that the eigenvalue sensitivity to a parameter β is described by the changes of the matrix $\mathbf{A}$ according to this parameter.

One of the applications of equation (5.6) is to find the sensitivity with respect to a particular entry a_{kl} of $\mathbf{A}$. We know that the derivative of $\mathbf{A}$ with respect to a_{kl} equals one at its corresponding entry and zeros otherwise, which is written as :

$$\frac{\partial\mathbf{A}}{\partial a_{kl}} = [\alpha_{ij}] = [\delta_{ik}\delta_{jl}], \tag{5.7}$$

where $\alpha_{ij} = 1$ if the pair (i, j) equals (k, l), and $\alpha_{ij} = 0$ otherwise. By substituting equation (5.7) in equation (5.6) and by using equation (5.2) and equation (5.3), the eigenvalue sensitivity with respect to any entry a_{kl} of $\mathbf{A}$ is given by:

$$\frac{\partial\lambda_i}{\partial a_{kl}} = \mathbf{w_i}^T\frac{\partial\mathbf{A}}{\partial a_{kl}}\mathbf{u_i} = w_{i\mathbf{k}}u_{i\mathbf{l}} \quad /i, k, l \in \{1, 2, ..., n\}, \tag{5.8}$$

where $w_{i\mathbf{k}}$ is the k^{th} component of the left eigenvector $\mathbf{w_i}$ and $u_{i\mathbf{l}}$ the l^{th} component of the right eigenvector $\mathbf{u_i}$.The relation (5.6) shows that the eigensensitivity with respect to a given entry a_{kl} of $\mathbf{A}$ is quantified by the product of some compnents of its corresponding right and left eigenvectors. The right side of equation (5.6) corresponds to the entries of the sensitivity matrix $\mathbf{\Gamma}_i$:

$$\mathbf{\Gamma}_i = \left[\frac{\partial \lambda_i}{\partial a_{kl}}\right] = \mathbf{w_i u_i}^T \quad /i,k,l \in \{1,2,...,n\}. \tag{5.9}$$

The elements in $\mathbf{\Gamma}_i$ relate changes of the eigenvalue λ_i to the changes in the entries a_{kl} of $\mathbf{A}$. The sensitivity is the local slope of λ_i as a function of a_{kl}. If the elements a_{kl} are perturbed to $\tilde{a}_{kl} = a_{kl} + \Delta a_{kl}$, then the eigenvalue λ_i is perturbed to $\tilde{\lambda_i} = \lambda_i + \delta\lambda_i$ as follows :

$$\tilde{\lambda}_i = \lambda_i + \sum_{k,l=1}^{N} \frac{\partial \lambda_i}{\partial a_{kl}} \Delta a_{kl} = \lambda_i + \sum_{k,l=1}^{N} w_{i\mathbf{k}} u_{i\mathbf{l}} \Delta a_{kl} \tag{5.10}$$

5.3 Impact of Edge Perturbations on the Von Neumann Entropy

The perturbation of the weight associated to an edge e_{kl} in a graph G is equivalent to the perturbation of two symmetric elements (a_{kl} and a_{lk}) of its corresponding adjacency matrix $\mathbf{A}$. In order to quantify the effect of these perturbations on the graph's entropy, we introduce a small modifications in $\mathbf{A}$, by adding for example, to the elements a_{12} and a_{21} a quantity ξ. Therefore, we get the following new perturbed adjacency matrix:

$$\tilde{\mathbf{A}} = [\tilde{a}_{ij}] = \begin{pmatrix} 0 & a_{12}+\xi & \cdots & a_{1n} \\ a_{21}+\xi & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ a_{n1} & \cdots & \cdots & 0 \end{pmatrix}. \tag{5.11}$$

The changes introduced in $\mathbf{A}$ appear indeed in the degree matrix:

$$\tilde{\mathbf{D}} = [\tilde{d}_{ij}] = \begin{pmatrix} (\sum_{j=1}^{n} a_{1j}) + \xi & 0 & \cdots & 0 \\ 0 & (\sum_{j=1}^{n} a_{2j}) + \xi & & \vdots \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & \cdots & \sum_{j=1}^{n} a_{nj} \end{pmatrix}. \tag{5.12}$$

Likewise, using these perturbed adjacency $\tilde{\mathbf{A}}$ and degree $\tilde{\mathbf{D}}$ matrices, we obtain the perturbed laplacian

matrix ($\tilde{\mathbf{L}} = \tilde{\mathbf{D}} - \tilde{\mathbf{A}}$) :

$$\tilde{\mathbf{L}} = \begin{pmatrix} (L_{11}+\xi) & -(L_{12}+\xi) & \cdots & -L_{1n} \\ -(L_{21}+\xi) & (L_{22}+\xi) & & \vdots \\ \vdots & & \ddots & \vdots \\ -L_{n1} & \cdots & \cdots & L_{nn} \end{pmatrix}. \tag{5.13}$$

By modifying two elements of the adjacency matrix, we make appear two additional elements in the Laplacian matrix. Therefore, by normalizing it using its trace $Trace(\tilde{\mathbf{L}})$, we get its corresponding density matrix $\tilde{\boldsymbol{\rho}}$, having the eigenvalues $\tilde{\lambda} \in [0,1]$, where $\sum_{i=1}^{n} \tilde{\lambda}_i = 1$:

$$\tilde{\boldsymbol{\rho}} = \frac{\tilde{\mathbf{L}}}{Trace(\tilde{\mathbf{L}})}. \tag{5.14}$$

According to equation (5.10), the eigenvalues associated to the perturbed density matrix $\tilde{\boldsymbol{\rho}}$ are given by:

$$\tilde{\lambda}_i = \frac{1}{Trace(\tilde{\mathbf{L}})}\left[\lambda_i + \sum_{k,l=1}^{n} \frac{\partial \lambda_i}{\partial L_{kl}} \Delta L_{kl}\right] \tag{5.15}$$

$$= \frac{1}{Trace(\mathbf{L}) + \sum_{\mathbf{k=1}}^{\mathbf{n}} \boldsymbol{\Delta} L_{\mathbf{kk}}}\left[\lambda_i + \sum_{k,l=1}^{n} \frac{\partial \lambda_i}{\partial L_{kl}} \Delta L_{kl}\right], \tag{5.16}$$

where L_{kl} and λ_i are respectively the entries and the eigenvalues of original laplacian matrix $\mathbf{L}$. Furthermore, we observe in the perturbed laplacian matrix $\tilde{\mathbf{L}}$ (5.13), that only four entries are changed, therefore, the expression (5.16) can be reduced to:

$$\tilde{\lambda}_i = \frac{1}{Trace(\tilde{\mathbf{L}})}\left[\lambda_i + \frac{\partial \lambda_i}{\partial L_{11}}\Delta L_{11} + \frac{\partial \lambda_i}{\partial L_{22}}\Delta L_{22} + \frac{\partial \lambda_i}{\partial L_{12}}\Delta L_{12} + \frac{\partial \lambda_i}{\partial L_{21}}\Delta L_{21}\right]. \tag{5.17}$$

Knowing that : $\Delta L_{11} = \Delta L_{12} = \Delta L_{22} = \Delta L_{21} = \xi$, the equation (5.17) becomes:

$$\tilde{\lambda}_i = \frac{1}{Trace(\tilde{\mathbf{L}})}\left[\lambda_i + \left(\frac{\partial \lambda_i}{\partial L_{11}} + \frac{\partial \lambda_i}{\partial L_{12}} + \frac{\partial \lambda_i}{\partial L_{22}} + \frac{\partial \lambda_i}{\partial L_{21}}\right)\xi\right] \tag{5.18}$$

The Von Neumann entropy of the perturbed density matrix $\tilde{\boldsymbol{\rho}}$ is given by:

$$S(\tilde{\boldsymbol{\rho}}) = -\sum_{i=1}^{n} \tilde{\lambda}_i \log_2 \tilde{\lambda}_i, \tag{5.19}$$

thus, by substituting (5.18) in (5.19), we get:

$$S(\tilde{\boldsymbol{\rho}}) = -\sum_{i=1}^{n}\left[\frac{\lambda_i + (w_{i\mathbf{1}}u_{i\mathbf{2}} + w_{i\mathbf{2}}u_{i\mathbf{1}} + w_{i\mathbf{1}}u_{i\mathbf{1}} + w_{i\mathbf{2}}u_{i\mathbf{2}})\xi}{Trace(\mathbf{L}) + 2\xi}\right] \times \log_2\left[\frac{\lambda_i + (w_{i\mathbf{1}}u_{i\mathbf{2}} + w_{i\mathbf{2}}u_{i\mathbf{1}} + w_{i\mathbf{1}}u_{i\mathbf{1}} + w_{i\mathbf{2}}u_{i\mathbf{2}})\xi}{Trace(\mathbf{L}) + 2\xi}\right]. \tag{5.20}$$

(5.20) Illustrates the changes in the graph's entropy due to the perturbation of one edge e_{12} in its structure. At this stage, we can not confirm either the increase or decrease in the value of the entropy, it depends on the concerned components of the eigenvectors. In spite of the fact that the increase in entropy is often interpreted as due to the creation of information, we show later that this is not always the case, and that the deletion of an edge may cause an increase in entropy. In the next section, we exploit the changes in Von Neumann's entropy to measure the importance of edges in a given structure, and thus measure their vulnerability to changes and to attacks that distort the graph.

5.4 Graph Edges Vulnerability

In an unweighted graph, the evaluation of the importance that could be accorded to every edges in the structure remains a problematic issue, especially when it comes to graphs that model infrastructure or logistic networks for example. In the previous section, we showed that the weight perturbation of an edge in the graph affects directly its Von Neumann entropy. Nevertheless, the impact differs from one edge to another. The sensitivity of the entropy to changes varies according to the neighborhood and the local properties of the area where the edge affected by the perturbation is located. In order to quantify practically this sensitivity to perturbations, we measure the entropy distortion before and after the changes by the following difference:

$$\Delta_{i,j} = S(\boldsymbol{\rho}_0) - S(\tilde{\boldsymbol{\rho}}[\xi_{i,j}]), \tag{5.21}$$

where $\boldsymbol{\rho}_0$ is the density matrix of G before perturbation, and $\tilde{\boldsymbol{\rho}}[\xi_{i,j}]$ is the density matrix after perturbation of the entries $a_{i,j}, a_{j,i} \in \{0,1\}/e_{i,j} \in \mathcal{E}$ in the adjacency matrix by the quantity $\xi_{i,j} \in [0,1]$. We scale up the obtained entropic difference by an exponential function, thus, we get the following distortion in entropy relative to the edge $e_{i,j} \in \mathcal{E}$:

$$\eta_{i,j} = exp\,(\Delta_{i,j}). \tag{5.22}$$

In Figures 5.1, 5.3, 5.5, 5.7 and 5.10, we illustrate in some particular graphs, the sensitivity of the distance η (5.22) to different perturbation values introduced in the edges. Therefore, we notice three major observations: the first one is that the perturbation of edges does not affect in the same manner the entropy or the structural information contained in the graph. Some edges exhibit higher sensitivity compared to others, due to their role in the structure. The second one is that the perturbation or the deletion of an edge ($\xi = 1$) does not decrease always the entropy, contrariwise, in some cases it increases the entropy, yielding values of η smaller than 1 (red curves), which means that in the sense of Von Neumann entropy,

the deletion of some edges does not imperatively mean a destruction of information. The third one is that the behaviour of the distortion is monotonous as the value of perturbation increases. Our intuition proves right and argues the fact that the Von Neumann entropy is suitable for measuring in smooth manner the local changes that occur in the graph's structure. Therefore, the entropic distortion could represent an interesting tool to measure the edges vulnerability in the graph, especially for unweighted graphs, the idea being to associate the distortion value η to each edge as a weight, calculated after perturbation. We summarize this weighting approach in the following algorithm called **VPV**$-$*weighting*:

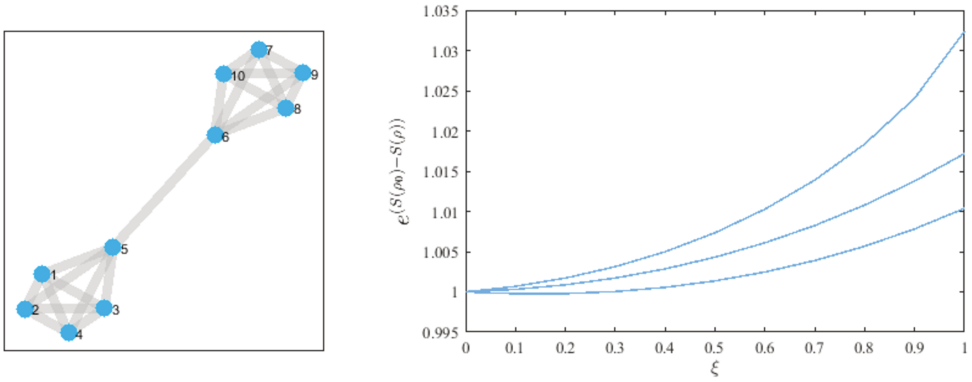
Algorithm: **VPV**$-$*weighting* (**V**on Neumann-**P**erturbation-**V**ulnerability)

Let $G = (\mathcal{V}, \mathcal{E})$ be an unweigthed graph, with $\mathcal{V}, \mathcal{E}$ are respectively the nodes and edges sets. Let $\mathbf{A}$, $\mathbf{D}$ be its adjacency and degrees matrices. Its new weighted adjacency matrix $\mathbf{W}$ can be computed following these steps:

1. $\mathbf{L} = \mathbf{D} - \mathbf{A}$
2. $\boldsymbol{\rho}_0 = \frac{\mathbf{L}}{Trace(\mathbf{L})}$
3. $S(\boldsymbol{\rho}_0) = -Trace(\boldsymbol{\rho}_0 \, Log_2 \, \boldsymbol{\rho}_0)$
4. Choose a perturbation value $\xi \in [0, 1]$
5. Choose an edge to perturbate, $e_{kl} = (v_k, v_l) \in \mathcal{E}$
6. $\tilde{\mathbf{A}} = [\tilde{a}_{ij}] = \begin{cases} a_{ij} - \xi, & \text{if } i, j \in \{k, l\} \\ 0 & \text{Otherwise} \end{cases}$
7. $\tilde{\mathbf{L}} = \tilde{\mathbf{D}} - \tilde{\mathbf{A}}$ with $\tilde{d}_i = \sum_{j=1}^{n} \tilde{a}_{ij}$
8. $\tilde{\boldsymbol{\rho}} = \frac{\tilde{\mathbf{L}}}{Trace(\tilde{\mathbf{L}})}$
9. $S(\tilde{\boldsymbol{\rho}}) = -Trace(\tilde{\boldsymbol{\rho}} \, Log_2 \, \tilde{\boldsymbol{\rho}})$
10. $\mathbf{W}_{kl} = \mathbf{W}_{lk} = exp(S(\boldsymbol{\rho}_0) - S(\tilde{\boldsymbol{\rho}}[\xi]))$
11. Iterate steps from 5 to 10 for all edges, $e_{kl} \in \mathcal{E}$

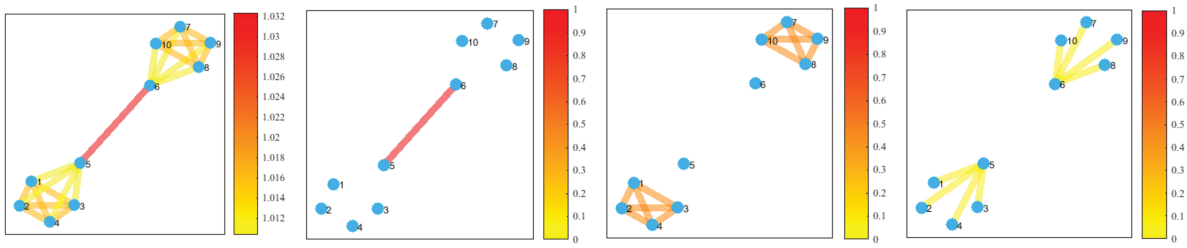
In Figure 5.1, we present the TowBalls graph and the entropic response of its edges to perturbations. The particularity of this graph is that it contains two highly connected regions (complete subgraphs of size 5) attached between them by a unique edge $e_{56} = (v_5, v_6)$, which guarantees the access to all nodes in the graph. We expect this edge to be the most sensitive to perturbations and the most vulnerable among other edges. As we observe in the entropy distortion curves (Figure 5.1(b)) where there are three types of edges corresponding to three levels of vulnerability.

We obtain in Figure 5.2 (a) the weighted version of TwoBalls graph using the **VPV**$-$*weighting*



(a) (b)

Figure 5.1: TowBalls graph, (a) is the original unweighted graph, (b) Curves representing changes of the entropy according to different perturbation values introduced to edges. Each curve corresponds to a particular edge, 21 in total.

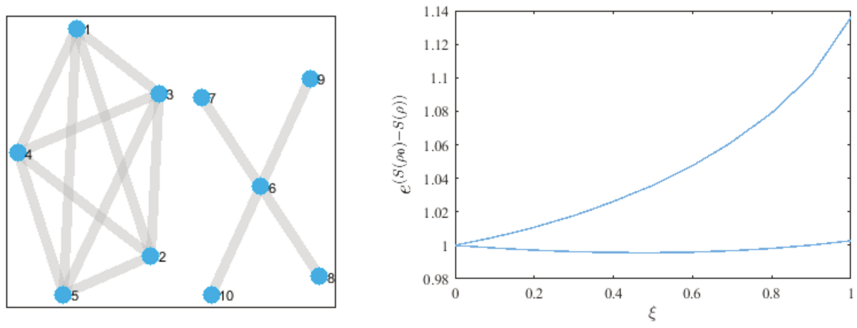


(a) (b) (c) (d)

Figure 5.2: (a) Weighted TwoBalls graph using the **VPV**$-$*weighting* algorithm. The edges are grouped into three levels (b), (c) and (d) according to their importance in the network.

algorithm. As expected the edge e_{56} is qualified as the most vulnerable in the structure, because it links two important communities in the graph and its deletion causes a disconnection of the network. We observe that the edges that have the same role in the graph, have the same level of vulnerability, as we see in Figures 5.2 (c) and (d).

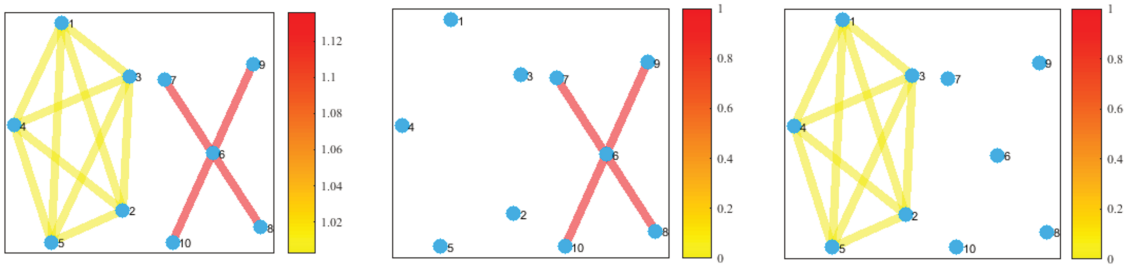
In Figures 5.3 and 5.4, we show that the **VPV**$-$*weighting* algorithm gives coherent results also on unconnected graphs. As well as the algorithm is very sensitive to discontinuities in the structure, it puts the Star part edges in red (Figure 5.4 (a)), because their disappearance isolates some nodes and makes the graph disconnected. On the other hand, we also show in the graph of the Figure 5.5 (a), that the algorithm is sensitive to several levels of discontinuity risk. We added the edge $e_{46} = (v_4, v_6)$ in the graph of Figure 5.3(a) to make it connected. The algorithm classified that edge as highly vulnerable node because it guarantees the access to all the nodes in the graph. However, we observe that the edges in the Star part (Figure 5.6 (b)) remain the most vulnerable, even more than the edge e_{46}. The reason is that deleting one of the edges e_{67}, e_{68}, e_{69} or e_{910} isolates some nodes, causing an abrupt rupture in the



(a) (b)

Figure 5.3: Complete-Star disconnected graph of size 10, (a) is the original unweighted graph, (b) Curves representing changes of the entropy according to different perturbation values introduced to edges. Each curve corresponds to a particular edge, 14 in total.

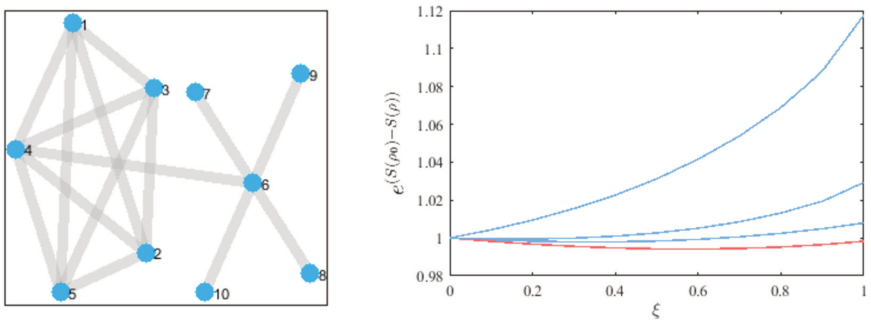
structure. While the deletion of the edge e_{46} disconnects the graph, but into two coherent subgraphs with comparable sizes, which we qualify as a smooth discontinuity. Hence, the **VPV**$-$*weighting* algorithm seem to make well the difference between these two levels of discontinuity risks (Figure 5.6 (b) vs Figure 5.6 (c)).



(a) (b) (c)

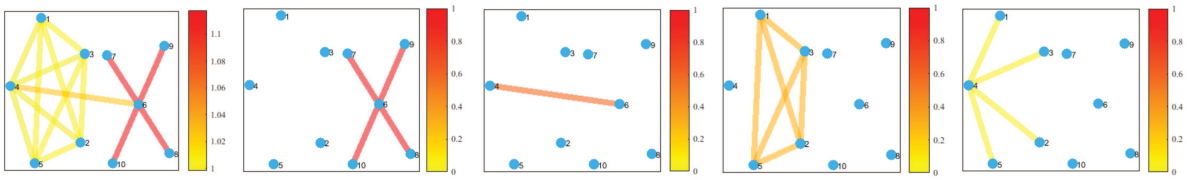
Figure 5.4: (a) Weighted Complete-Star disconnected graph using the **VPV**$-$*weighting* algorithm. The edges are grouped into two levels (b) and (c) according to their importance in the network.

Furthermore, we tested the **VPV**$-$*weighting* algorithm on some more complex graphs. In Figures 5.7 (a) and (b), we show the unweighted Sensors graph with its entropy distortion due to perturbations. While we show in Figure 5.8 (a) the resultant weighted graph. Overall, the attributed weightes seems to be coherent with the properties of the structure. By setting four levels of vulnerability, we obtained the subgraphs illustrated in Figures 5.9 (a), (b), (c) and (d). Therefore, the most interesting observation, is that the first level (Figure 5.9 (a)) contain only one edge ($e_{15,48}$) connecting the nodes v_{15} and v_{48}. This edge is the most vulnerable because its deletion disconnects the graph and isolates the node v_{48}. While edges in the second level (Figure 5.9 (b)) links the important regions in the structure, the deletion of one of them does not isolate individual nodes. We observe also that the lowest vulnerable edges are located in the most dense zone of the structure (Figure 5.9 (d)), inwhich the access to nodes in this zone



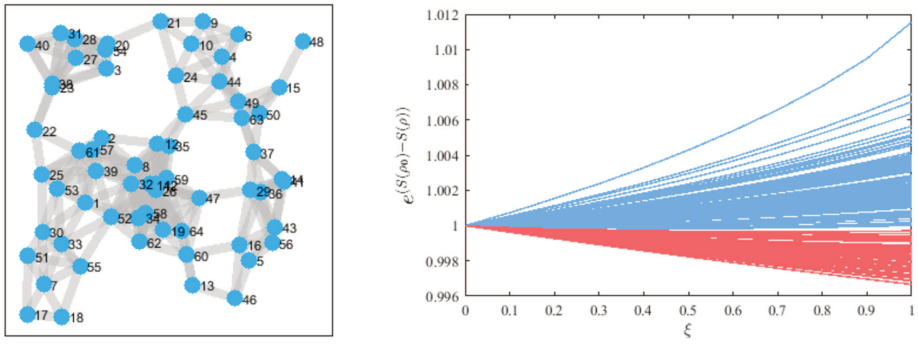
(a) (b)

Figure 5.5: Complete-Star connected graph of size 10, (a) is the original unweighted graph, (b) Curves representing changes of the entropy according to different perturbation values introduced to edges. Each curve corresponds to a particular edge, 15 in total. The red curves indicate a decrease of the entropy after the perturbation, compared to its initial value.



(a) (b) (c) (d) (e)

Figure 5.6: (a) Weighted Complete-Star connected graph using the **VPV**−*weighting* algorithm. The edges are grouped into four levels (b), (c), (d) and (e) according to their importance in the network.

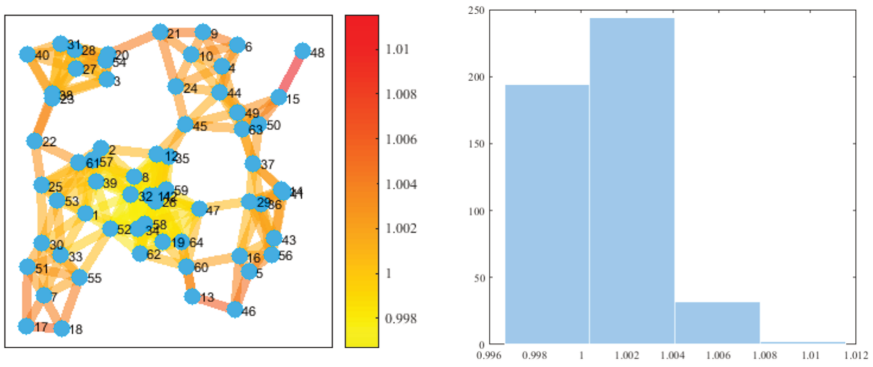


(a) (b)

Figure 5.7: Sensors graph of size 64, (a) is the original unweighted graph, (b) Curves representing changes of the entropy according to different perturbation values introduced to edges. Each curve corresponds to a particular edge, 236 in total. The red curves indicate a decrease of the entropy after the perturbation, compared to its initial value.

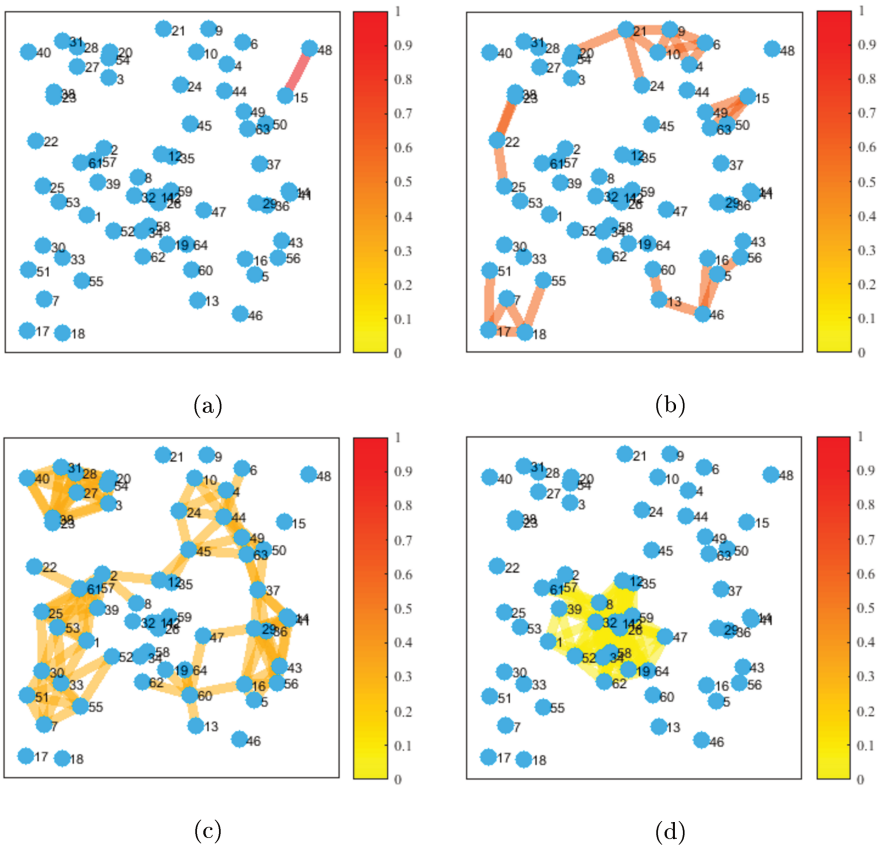
is redundent by many paths.

Like in Sensors graph, the **VPV**−*weighting* algorithm gives coherent results in Karate Club graph,



(a) (b)

Figure 5.8: Weighted Sensors graph using the **VPV**$-$*weighting* algorithm, (a) vulnerability map of edges, (b) four bins histogram of edge weights, allowing the segmentation of the graph's structure into four vulnerability levels.

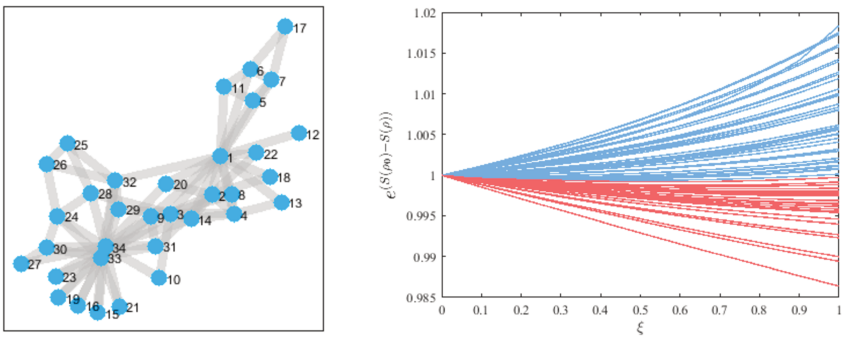


(a) (b)

(c) (d)

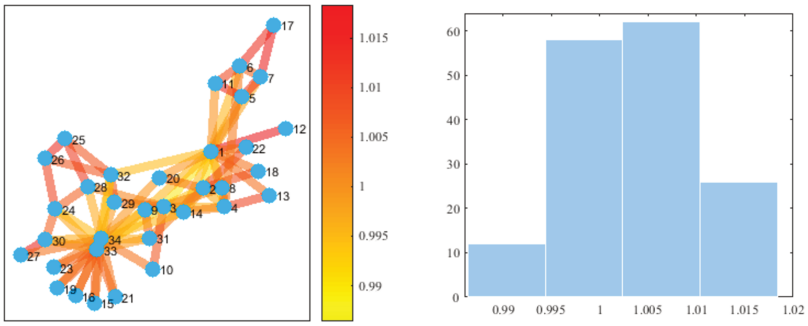
Figure 5.9: Using a four bins histogram, the Sensors graph is segmented into four vulnerability levels (a), (b), (c) and (d). In each level, edges share roughly the same importance in the graph's structure.

see Figures 5.10 (a) and 5.11 (a). We observe as a first vulnerability level (Figure 5.12 (a)), edges that risk to isolate some nodes once they are deleted, as in the case of the edges $e_{1,12}, e_{6,17}$ and $e_{7,17}$. Furthermore, we observe that the algorithm behaves in a particular manner with nodes of degree two, which we rank into two groupes: $\{v_{13}, v_{27}\}$ and $\{v_{15}, v_{16}, v_{19}, v_{21}, v_{23}\}$.



(a) (b)

Figure 5.10: Karate Club graph of size 34, (a) is the original unweighted graph, (b) Cuves representing changes of the entropy according to different perturbation values introduced to edges. Each curve correspond to a particular edge, 78 in total. The red curves indicate a decrease of the entropy after the perturbation, compared to its initial value.



(a) (b)

Figure 5.11: Weighted Karate Club graph using the **VPV**$-$*weighting* algorithm, (a) vulnerability map of edges, (b) four bins histogram of edge weights, allowing the segmentation of the graph's structure into four vulnerability levels.

In the first case, the two edges that permits the access to v_{13} and v_{27} do not have the same score of vulnerability, depending on the importance of the source nodes. For example, in node v_{27}, the edge $e_{27,30}$ is more vulnerable then the edge $e_{27,34}$, because the node v_{34} is more central and of high degree in the network, compared to the node v_{30}. Thus, the algorithm priorize the access to smaller degree nodes to avoid isolation risk. Likewise, concerning the node v_{13}, the edge $e_{4,13}$ seems to be more vulnerable than $e_{1,13}$, because the node v_4 is less important in the network than the node v_1.

In the second case, the nodes from the second group are connected to the network via two edges that share similar vulnerability level, because the source nodes have comparable centralities and degrees in the structure. For example, the node v_{21} is connected to the nodes v_{33} and v_{34} via $e_{21,33}$ and $e_{21,34}$, both nodes v_{33} and v_{34} are of high degree and have similar neighborhood, therefore, the edges $e_{21,33}$ and $e_{21,34}$ encounter the same risk level. We observe the same thing for nodes v_{15}, v_{16}, v_{19} and v_{23}.

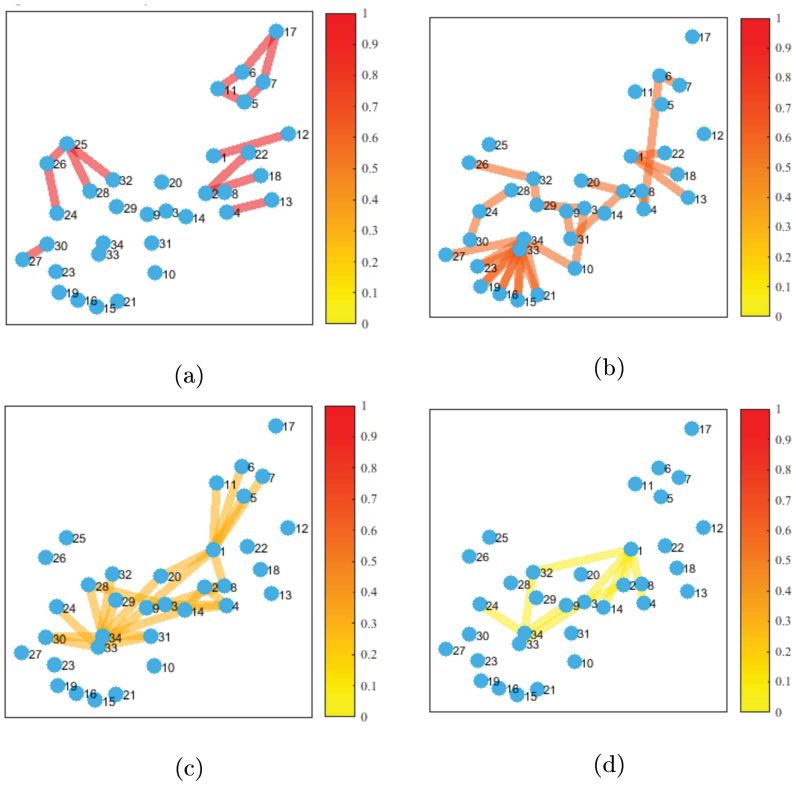


Figure 5.12: Using a four bins histogram, the Karate Club graph is segmented into four vulnerability levels (a), (b), (c) and (d). In each level, edges share roughly the same importance in the graph's structure.

5.5 Conclusion

In this chapter, we have proposed a new algorithm based on Von Neumann entropy to measure the vulnerability of connections in a network. The idea was to perturbe edges individualy and quantify its impact on the overall entropy of the graph. The algorithm has proved relevant in some simple and complex graphs, attributing weights in a coherent manner adapted to the local properties of the structure. The algorithm is sensitive to disconnection risks that could occur in the structure, by classifying as highly vulnerable the edges that connect fragile areas and easy to isolate. Our algorithm is useful especially for applications related to the fragility of infrastructure and logistics networks in terms of accessibility and resilience against attacks and failures. The **VPV**$-$*weighting* based vulnerability map will allow a better security of the network and guarantee a minimal service in case of partial damage, by ensuring permanent redundancy of access. Nevertheless, additional tests on a wider range of real graphs are required.

Conclusions and Perspectives

The main purpose of this thesis was to develop new spectral similarity measures for graphs comparison, and adapt them to be used with the SVM algorithm for learning purposes and classification of real world graphs. Moreover, we aimed to understand the nature of the relationship between the adjacency ($\mathbf{A}$) and the laplacian ($\mathbf{L}$) matrices beyond the simple linear relationship between them : $\mathbf{L} = \mathbf{D} - \mathbf{A}$.

We remind the main contributions of this dissertation:

$\Rightarrow$ We proposed the following new graph similarity measures:

- TVG: is a measure based on total variation (TV) of the graph signal, it quantifies the oscillatory behaviour of the graph signal and its interaction with the supporting structure. We showed that it is an interesting informative and simple descriptor for graph signals comparison.
- GE: is a measure based on the laplacian graph energy which is calculated via the laplacian eigenspectrum of the graph. It is a pertinent information that characterizes well the graph, and measures the complexity degree of its structure, taking into account both connections distribution of the network and its density.
- JET: is a joint convex combination between the TVG and GE measures to take advantage from both. It allows to take into consideration the signal's properties and the complexity information about the supporting structure.
- JSS: is our second joint graphs similarity measure, which exploits both spectral informations from adjacency $\mathbf{A}$ and laplacian $\mathbf{L}$ matrices. The $\mathbf{A}$ matrix characterizes the topological graph complexity in terms of connections between nodes and underscores the local cohesiveness of nodes, while $\mathbf{L}$ matrix is well suited for recovering some information about clusters and commu-

nities in the graph, thus, capture its inherent structure. The JSS incorporates both advantages those of **A** and **L**.

⇒ We integrated our similarity measures (TVG, GE, JET and JSS) in an exponential kernel, which we use in the SVM learning algorithm to classify graphs issued from bioinformatics and time series. Compared to the state-of-art methods, our measures are of low complexity and fast to run. We show that with simple pertinent global descriptors, we could do better than other complex methods. Via the (JET and JSS) measures we show that linear combinations of multiple measures increases often the graphs discrimination power and enhances classification performance.

⇒ The JSS measure allowed us to confirm our intuition that **A** and **L** matrices contribute unequally in graph characterization task, and to emphasize the fact that they represent differently the structural information about the underlying graph. In spite of the simple linear relationship between them ($\mathbf{L} = \mathbf{D} - \mathbf{A}$), these two matrices give rise to different inferences drawn from the graph.

⇒ We highlighted the overlapping and the unequal contributions of (**A**) and (**L**) for graph representation, by comparing them in terms of the so called Von Neumann entropy, connectivity and complexity measures. The graph is viewed as a quantum system and thus, the calculated Von Neumann entropy of its perturbed density matrix emphasizes the overlapping in terms of information quantity.

⇒ We illustrated by classification findings on real and conceptual graphs the effectiveness of the JSS measure in terms of classification accuracies, and by which we highlight the varying information overlapping rates of **A** and **L** via a weighting parameter α, and we point out their different ways in recovering structural information of the graph.

⇒ We showed that the JET and JSS measures handle the graph cospectrality issue, and they allow the distinction between graphs that share the same eigenvalues spectrum corresponding to **A** or **L** matrices.

⇒ We showed that converting time series to graphs using VG algorithm (L. Lacasa et al., 2008) could enhance the classification accuracy of these series, and permits the application of graph kernels in the learning process.

⇒ We used the Von Neumann entropy to show that the edges of a given graph does not react to perturbations the same way, and that their sensitivity to noise is not the same. We used the entropy distotion to score the vulnerability of each edge, and to formalize a graph weighting algorithm which we called **VPV**$-$*weighting*. For instance, our approach is useful for networks diagnostics and to study their resilience to malicious attaques and damages due to failures.

⇒ We used the Low-Rank matrix approximation to define the salient structure of the graph, which we refer to as Dominant Graph Component (**DGA**).

In Chapter 1, we recall some basic notions about graphs and their spectral analysis. We present the most well-known representation matrices, as well as notions related to the structure of graphs, such as regularity, connectivity and bipartition. In addition of some notions related to the eigenspectrum of the adjacency and laplacian matrices, such as the Fiedler's value, the largest eigenvalue, the energy, the Kirchoff index. We included also a reminder about low-rank matrix approximation, and some primary results about its use in finding a backbone of the structure, which we called thereafter dominant graph.

In Chapter 2, we briefly discussed the problem of machine learning from data and especially supervised statistical learning. We detailed the mathematical model describing the functioning of support vector machines (SVMs). They classify data efficiently in a wide range of linear and non-linear learning problems. Through kernel functions, they are able to separate non-linearly separable data in a higher dimensional space. Their computational attractiveness is due to the fact that they can be applied in high dimensional feature spaces without suffering from the high cost of explicitly computing the feature map. One advantage of kernel techniques among others is that they allow to run a large range of learning algorithms on structured data, so far restricted only for attribute-indexed data. This is why it is interesting to develop kernels adapted for graphs to take advantage of the framework of classical statistical learning algorithms, which is already mature and well developed.

In Chapter 3, we have discussed the problem of similarity measurement of graphs, and their utility for learning applications, like the classification tasks. In addition, we reviewed the notions of the total variation (TV) of a signal and particularly of a signal on a graph, and the energy ($E_{\mathbf{L}}$) associated with its structure. By being calculated via one of the eigen-spectra associated with the graph, the energy is a pertinent information that characterizes well the graph, and measures the complexity degree of its structure, taking into account both connections distribution of the network and its density. While the total variation quantifies the oscillatory behaviour of the graph-signal and its interaction with the supporting structure. Given these properties, we proposed new graph-signals similarity measures based on the total variation and the laplacian graph energy, adapted for labeled weighted and unweighted graphs, which we called respectively (TVG) and (GE). These two measures integrated in an exponential kernel show competitive performance on binary and multiclass graph-signals classification. To take advantage from both measures, we combined them in a new joint measure called JET. Applied on some bioinformatics classification problems, our measures yield competitive accuracy levels on all considered data sets and outperform some state-of-the-art graph kernels in terms of computation runtime. The results of the

JET measure show the benefits of hybrid approaches on discriminating graph signals without significant increase of complexity.

In Chapter 4, a new joint spectral similarity (JSS) measure for graphs classification was introduced. We have shown that both adjacency ($\mathbf{A}$) and laplacian ($\mathbf{L}$) matrices carry different structures information of the underlying graph. The adjacency matrix characterizes the topological graph complexity in terms of connections between nodes and underscores their local cohesiveness. These properties explain why the good classification accuracies achieved by JSS measure are more attributed to adjacency matrix ($\alpha > 0.5$). Through VN entropy, it is easy to see that laplacian matrix brings out changes in node degrees information. Furthermore, this matrix is well suited to recover information about clusters of the graph and thus capture its inherent structure. The obtained results highlight the fact that JSS combines both advantages of Laplacian and adjacency matrices. Also, these findings confirm that these matrices contribute unequally and emphasize the fact that they represent differently information about structures of the underlying graph. Additionally, these results show the interest of the VG approach for classification of time series. As a result of this work, we hope to have increased the awereness about the importance of the properly choice of the representation matrix for graph spectral analysis purposes. Even the JSS measure handles cospectral graphs with respect to both $\mathbf{A}$ and $\mathbf{L}$.

In Chapter 5, we have proposed a new algorithm based on Von Neumann entropy to measure the vulnerability of connections in a network. The idea was to perturbe edges individualy and quantify its impact on the overall entropy of the graph. The algorithm has proved relevant in some simple and complex graphs, attributing weights in a coherent manner adapted to the local properties of the structure. The algorithm is sensitive to disconnection risks that could occur in the structure, by classifying as highly vulnerable the edges that connect fragile areas and easy to isolate. Our algorithm is useful especially for applications related to the fragility of infrastructure and logistics networks in terms of accessibility and resilience against attacks and failures. The $\mathbf{VPV}-$*weighting* based vulnerability map will allow a better security of the network and guarantee a minimal service in case of partial damage, by ensuring permanent redundancy of access.

This research work and proposed approaches presented in this thesis open the following new promising research directions:

- We intend to expand the tests of our similarity measures (TVG, GE, JET and JSS) to other classification problems, different than bioinformatics problems. For example, use them for studying the profiles and behaviors in social networks, or detect behavioural patterns and habits of the population when they use transport networks.

- The JET measure seeks to establish a sort of trade-off between the total variation attribute and the graph's energy. This lets us envisage an optimization approach to determine the optimal value of the weighting factor α which maximizes the classification performance.

- As we highlighted in Chapter 3, the graph energy does not have an unique expression, depending on the mathematical formalism and on the representation matrix that we choose to associate to the graph. Hence, we wonder about the form of graph energy that could discriminate it at best. We plan to test other varieties of the JET measure including the adjacency graph energy ($E_{\mathbf{A}}$), signless laplacian graph energy ($E_{|\mathbf{L}|}$) and others.

- We used in our joint similarity measures JET and JSS a linear convex combination between two quantities. We are curious about combining more than two attributes in one measure. For instance, F. Escolano et al., 2008b used convex Birkhoff combinations to quantify the complexity of graphs. In our framework, this combination could take the form:

$$\sum_{\beta=1}^{\gamma}\{\prod_{i=1}^{\beta-1}(1-\alpha_i)\}\,\alpha_\beta\,\Delta_\beta$$

  where γ is the number of attributes to be combined, α_i are the weighting parameters and Δ_i are the combined attributes.

- We highlighted in Chapter 2 some facts about the relationship between the adjacency ($\mathbf{A}$) and laplacian ($\mathbf{L}$) matrices beyond the linear expression between them $\mathbf{L}=\mathbf{D}-\mathbf{A}$. We are interested by the so called generalized adjacency matrix of a graph defined as:

$$\mathbf{A}_\alpha = \alpha\,\mathbf{D} - (1-\alpha)\mathbf{A}.$$

  Such form allow us to tune the contribution of degrees and individual connections for representing the graph at best. We notice that the laplacian matrix is a particular case of this generalized adjacency matrix, for $\alpha=0.5$, the matrices contribute equally and we get $\mathbf{A}_{\frac{1}{2}} = \frac{1}{2}\mathbf{L}$.

- We explained in Chapter 4 that the JSS measure handles the problem of cospectral graphs, and allows to discriminate them even if they share the same eigenspectrum according to one of the representation matrices. We showed in Table 4.1 that the number of cospectral graph decreases significantly when combining the matrices $\mathbf{A}$ and $\mathbf{L}$. However, we see in the table that the $|\,\mathbf{L}\,|$ matrix does better than $\mathbf{L}$. Therefore, we plan to test for classification another version of the JSS measure which includes the signless laplacian matrix $|\,\mathbf{L}\,|$.

- In Chapter 5, we presented the $\mathbf{VPV}-$*weighting* algorithm, which associates to the edges of a graph appropriate vulnerability scores according to their importance in the structure. We plan to generalize

this approach to nodes. We consider that the vulnerability of a node is closely related to that of the edges connecting it to the neighbors. Therefore, as a first idea we can use the simple degree of the node as a vulnerability score, computed in the new weights matrix obtained via the **VPV**−*weighting* algorithm:

$$Vul(v_i) = \sum_{v_i \sim v_j} \eta_{i,j}$$

where $\eta_{i,j}$ is the entropic distortion corresponding to the edge $e_{i,j} \in \mathcal{E}$, defined by the formula (5.22). The second option is to compute the new weighted centrality score of the node:

$$Vul(v_i) = \sum_{v_i \sim v_j} \frac{\eta_{i,j}}{n-1},$$

where $n-1$ is the number of remaining nodes without v_i, that is the set $\{v_j / v_j \in V \text{ and } v_j \neq v_i\}$.

- We plan to use the dominant graph component obtained by the **DGA** approach in spectral clustering in graphs. Because it corresponds to the salient graph. We observe that they are often weakened, while they had the same weight as all the other edges. This is likely to improve the clustering performance and community detection.

Bibliography

[1] A. Ghosh, S. Boyd, and A. Saberi. Minimizing effective resistance of a graph. *SIAM Rev.*, 50(1):37–66, 2008.

[2] D.J. Klein and M. Randic. Resistance distance. *J. Math. Chem.*, 12:81–95, 1993.

[3] I. Gutman and B. Mohar. The quasi-wiener and the kirchhoff indices coincide. *J. Chem. Inf. Comput. Sc.*, 36:982–985, 1996.

[4] V.M. Preciado and M.A. Rahimian. Moment-based spectral analysis of random graphs with given expected degrees. *IEEE Trans. Net. Sc. Eng*, 4(4):215–228, 2017.

[5] V.M. Preciado and A. Jadbabaie. Moment-based spectral analysis of large-scale networks using local structural information. *IEEE/ACM Trans. Net.*, 21(2):373–382, 2013.

[6] X. Yuan et al. Eradicating catastrophic collapse in interdependent networks via reinforced nodes. *Proc. Natl. Acad. Sci.*, 114(13):3311–3315, 2017.

[7] A. Bagnall et al. Time series data for classification called: the ucr time series archive. 2002. URL: http://timeseriesclassification.com/dataset.php.

[8] A. Barrat, M. Barthélemy, and A. Vespignani. Weighted evolving networks: coupling topology and weights dynamics. *Phys. Rev. Lett.*, 92(22):1–4, 2004.

[9] A. Dosovitskiy et al. Discriminative unsupervised feature learning with convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 766–774, 2014.

[10] A. Gavili and X.P. Zhang. On the shift operator, graph frequency, and optimal filtering in graph signal processing. *IEEE Trans. Sig. Proc.*, 65(23):6303–6318, 2017.

[11] A. Mheich et al. A novel algorithm for measuring graph similarity: application to brain networks. In *Neural Engineering (NER), 2015 7th International IEEE/EMBS Conference on*, pages 1068–1071. IEEE, 2015.

[12] A. Mueen, E. Keogh, and N. Young. Logical-shapelets: an expressive primitive for time series classification. In *Proc. ACM SIGKDD*, pages 1154–1162, 2011.

[13] A. Ortega et al. Processing: overview, challenges, and applications. *Proc. IEEE*, 106(5):808–825, 2018.

[14] A. Robles-Kelly and E.R. Hancock. A riemannian approach to graph embedding. *Pattern Recognition*, 40(3):1042–1056, 2007.

[15] A. Sandryhaila and J.M.F. Moura. Classification via regularization on graphs. In *GlobalSIP*, pages 495–498, 2013.

[16] A. Sandryhaila and J.M.F. Moura. Discrete signal processing on graphs: frequency analysis. *IEEE Trans. Signal Processing*, 62(12):3042–3054, 2014.

[17] A. Shokoufandeh et al. Indexing using a spectral encoding of topological structure. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.* Volume 2, pages 491–497. IEEE, 1999.

[18] A. Streitwieser. Molecular orbital theory for organic chemists, 1961.

[19] A.E. Brouwer and W.H. Haemers. *Spectra of graphs*. Springer Science & Business Media, 2011.

[20] A.E. Motter, T. Nishikawa, and Y. Lai. Range-based attack on links in scale-free networks: are long-range links responsible for the small-world phenomenon? *Phy. Rev. E*, 66:1–4, 2002.

[21] A.K. Debnath et al. Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity. *Journal of medicinal chemistry*, 34(2):786–797, 1991.

[22] A.O. Boudraa and F. Salzenstein. *Teager-Kaiser energy operators in time-frequency analysis*. In B. Boashash, editor. 2 ed. Edition. Edited by B. Boashash, Elsevier Academic Press, 2 ed. Edition, 2015. Chapter In Time-Frequency Signal Analysis and Processing. A Comprehensive reference, pages 209–217.

[23] B. Cao, Y. Li, and J. Yin. Measuring similarity between graphs based on the levenshtein distance. *Appl. Math*, 7(1L):169–175, 2013.

[24] B. Girault, P. Gonçalves, and É. Fleury. Translation on graphs: an isometric shift operator. *IEEE Signal Processing Letters*, 22(12):2416–2420, 2015.

[25] B. J. McClelland. Properties of the latent roots of a matrix: the estimation of π-electron energies. *The Journal of Chemical Physics*, 54(2):640–643, 1971.

[26] B. Luque et al. Horizontal visibility graphs: exact results for random time series. *Physical Review E*, 80(4):046103, 2009.

[27] B. Schölkopf and A.J. Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[28] B. Schölkopf, A. Smola, and F. Bach. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[29] B. Weisfeiler and A.A. Lehman. A reduction of a graph to a canonical form and an algebra arising during this reduction. *Nauchno-Technicheskaya Informatsia*, 2(9):12–16, 1968.

[30] B. Zhou et al. On spectral moments and energy of graphs. *MATCH Commun. Math. Comput. Chem*, 57:183–191, 2007.

[31] B.E. Boser, I.M. Guyon, and V.N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992.

[32] B. Boashash. *Time-Frequency Signal Analysis and Processing. A Comprehensive reference.* Elsevier Academic Press, 2 ed. Edition, 2015.

[33] D. Bravo, F. Cubria, and J. Rada. Energy of matrices. *Appl. Math. Comput.*, 312:149–157, 2017.

[34] R. Brualdi. *The mutually beneficial relationship of graphs and matrices*, volume 115. American Mathematical Society, 2011.

[35] C. Chen et al. Node immunization on large graphs: theory and algorithms. *IEEE Trans. Knowledge and Data Eng.*, 28(1):113–126, 2016.

[36] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[37] C. Helmberg and V. Trevisan. Spectral threshold dominance, brower's conjecture and maximality of laplacian energy. *J. Math. Anal. Appl.*, 512:18–31, 2017.

[38] C. Nan and I. Eusgeld. Adopting hla standard for interdependency study. *Relib. Eng. Syst. Safety*, 96:149–159, 2011.

[39] C.D. Godsil and B.D. McKay. Constructing cospectral graphs. *Aequationes Mathematicae*, 25(1):257–268, 1982.

[40] A. Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical imaging and vision*, 20(1-2):89–97, 2004.

[41] X. Chen. Perturbation theory of von neumann entropy. *Chinese Physics B*, 19(4):1–7, 2010.

[42] L. Cohen. *Time-Frequency Analysis*. Prentice Hall, 1994.

[43] T. Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE transactions on electronic computers*, (3):326–334, 1965.

[44] D. Chen, Q. He, and X. Wang. On linear separability of data sets in feature space. *Neurocomputing*, 70(13-15):2441–2448, 2007.

[45] D. Conte et al. Thirty years of graph matching in pattern recognition. *Int. J. Patt. Recogn. Art. Intell.*, 18(3):265–298, 2004.

[46] D. Eads et al. Genetic algorithms and support vector machines for time series classification. In *Proc. SPIE*, volume 4787, pages 74–85, 2002.

[47] D. Hu et al. The von neumann entropy of random multipartite graphs. *Discrete Applied Mathematics*, 232:201–206, 2017.

[48] D. Stevanović, I. Stanković, and M. Milosević. More on the relation between energy and laplacian energy of graphs. *MATCH Commun. Math. Comput. Chem*, 61(2):395–401, 2009.

[49] D.A. Spielman and S.H. Teng. Spectral partitioning works: planar graphs and finite element meshes. *J. Math. Anal. Appl.*, 421:284–305, 2007.

[50] D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533, 1986.

[51] D.I. Shuman et al. The emerging field of signal processing on graphs: extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine*, 30(3):83–98, 2013.

[52] D.I. Shuman, B. Ricaud, and P. Vandergheynst. Vertex-frequency analysis on graphs. *Appl. Comput. Harmonic Anal.*, 40(2):260–291, 2016.

[53] D.K. Hammond, P. Vandergheynst, and R. Gribonval. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30(2):129–150, 2011.

[54] E.R. Van Dam and W.H Haemers. Which graphs are determined by their spectrum? *Linear Algebra and its applications*, 373:241–272, 2003.

[55] F. Escolano, E.R. Hancock, and M.A. Lozano. Polytopal graph complexity, matrix permanents, and embedding. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 237–246. Springer, 2008.

[56] F. Escolano, E.R. Hancock, and M.A. Lozano. Polytopal graph complexity, matrix permanents, and embedding. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 237–246. Springer, 2008.

[57] F. Fouss et al. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *IEEE Trans. Knowl. Data Eng.*, 19(3), 2007.

[58] F. Passerini and S. Severini. Quantifying complexity in networks: the Von Neumann entropy. *Int. J. Agent Technol. Syst.*, 1(4):58–67, 2009.

[59] K. Fan. Maximum properties and inequalities for the eigenvalues of completely continuous operators. *Proceedings of the National Academy of Sciences*, 37(11):760–766, 1951.

[60] J. Fang and L.E. Atlas. Quadratic detectors for energy estimation. *IEEE Trans. Sig. Proc.*, 43(11):2582–2594, 1995.

[61] M. Fiedler. A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory. *Czechoslovak Mathematical Journal*, 25(100):619–633, 1975.

[62] M. Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical J.*, 23(2):298–305, 1973.

[63] G. Bianconi and A.L. Barabasi. Bose-einstein condensation in complex networks. *Physical review letters*, 86(24):5632–5635, 2001.

[64] G. Caporossi et al. Variable neighborhood search for extremal graphs. 2. finding graphs with extremal energy. *Journal of Chemical Information and Computer Sciences*, 39(6):984–996, 1999.

[65] T. Gärtner. A survey of kernels for structured data. *ACM SIGKDD Explorations Newsletter*, 5(1):49–58, 2003.

[66] G.S. Bopche and B.M. Mehtre. Graph similarity metrics for assessing temporal changes in attack surface of dynamic networks. *Computers & Security*, 64:16–43, 2017.

[67] I. Gutman. The energy of a graph. *Ber. Math.-Statist. Sekt. Forsch. Graz*, (103):1–22, 1978.

[68] I. Gutman. The energy of a graph: old and new results. In *Algebraic combinatorics and applications*, pages 196–211. Springer, 2001.

[69] I. Gutman. Topology and stability of conjugated hydrocarbons: the dependence of total π-electron energy on molecular topology. *Journal of the Serbian Chemical Society*, 70(3):441–456, 2005.

[70] H. Kashima, K. Tsuda, and A. Inokuchi. Marginalized kernels between labeled graphs. In *ICML*, volume 3, pages 321–328, 2003.

[71] H. Van Hasselt, A. Guez, and D. Silver. Deep reinforcement learning with double q-learning. In *AAAI*, volume 2, page 5. Phoenix, AZ, 2016.

[72] H.A. Bay-Ahmed, A.O. Boudraa, and D. Dare. A joint spectral similarity measure for graphs classification. *Pattern Recognition Letters (Accepted, minor revisions)*, 2018.

[73] H.A. Bay-Ahmed et al. Classification des signaux sur graphes par mesures spectrales algébriques. In *Proc. GRETSI*, pages 1–4, 2017.

[74] H.A. Bay-Ahmed, D. Dare-Emzivat, and A.O. Boudraa. Graph signals classification using total variation and graph energy informations. In *Proc. IEEE GlobalSIP*, pages 668–671, 2017.

[75] B. Haasdonk. Feature space interpretation of svms with indefinite kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(4):482–492, 2005.

[76] I. Gutman and B. Zhou. Laplacian energy of a graph. *Linear Algebra and its applications*, 414(1):29–37, 2006.

[77] I. Gutman and O.E. Polansky. *Mathematical concepts in organic chemistry*. Springer Science & Business Media, 2012.

[78] I. Gutman, X. Li, and Y. Shi. *Graph Energy*. New York: Springer, 2012.

[79] I. Gutman et al. On the energy of regular graphs. *MATCH Commun. Math. Comput. Chem*, 57:435–442, 2007.

[80] I. Gutman et al. Relation between energy and laplacian energy. *MATCH Commun. Math. Comput. Chem*, 59:343–354, 2008.

[81] I. Jovanović and Z. Stanić. Spectral distances of graphs based on their different matrix representations. *Filomat*, 28(4):723–734, 2014.

[82] I. Schomburg et al. Brenda, the enzyme database: updates and major new developments. *Nucleic acids research*, 32(suppl 1):D431–D433, 2004.

[83] I. Sutskever, J. Martens, and G.E. Hinton. Generating text with recurrent neural networks. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 1017–1024, 2011.

[84] I. W. Selesnick and P. Y. Chen. Total variation denoising with overlapping group sparsity. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 5696–5700. IEEE, 2013.

[85] J. Gao, B. Barzel, and A.L. Barabasi. Universal resilience patterns in complex networks. *Nature*, 530:307–312, 2016.

[86] J. Gu, B. Hua, and S. Liu. Spectral distances on graphs. *Discrete Applied Mathematics*, 190:56–74, 2015.

[87] J. H. Koolen and V. Moulton. Maximal energy bipartite graphs. *Graphs and Combinatorics*, 19(1):131–135, 2003.

[88] J. H. Koolen and V. Moulton. Maximal energy graphs. *Advances in Applied Mathematics*, 26(1):47–52, 2001.

[89] J. Lafferty, A. McCallum, and C.N. F. Pereira. Conditional random fields: probabilistic models for segmenting and labeling sequence data, 2001.

[90] J. Lines and A. Bagnall. Time series classification with ensembles of elastic distance measures. *Data Min. Knowl. Disc.*, 29(3):565–592, 2015.

[91] J. Liu and B. Liu. A laplacian-energy-like invariant of a graph. *MATCH Commun. Math. Comput. Chem.*, 59:397–419, 2008.

[92] J. Liu and B. Liu. On relation between energy and laplacian energy. *MATCH Commun. Math. Comput. Chem*, 61(2):403–406, 2009.

[93] J. M. Bioucas-Dias, M. A. T. Figueiredo, and J. P. Oliveira. Total variation-based image deconvolution: a majorization-minimization approach. In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, volume 2, pages II–II. IEEE, 2006.

[94] J. Oliveira, J. Bioucas-Dias, and M. A. T. Figueiredo. Adaptive total variation image deblurring: a majorization–minimization approach. *Signal Processing*, 89(9):1683–1693, 2009.

[95] J. Ramon and T. Gärtner. Expressivity versus efficiency of graph kernels. In *First international workshop on mining graphs, trees and sequences*, pages 65–74. Citeseer, 2003.

[96] J. Shawe-Taylor and N. Cristianini. *Kernel methods for pattern analysis*. Cambridge university press, 2004.

[97] J. Wang, R.C. Wilson, and E. R. Hancock. Fmri activation network analysis using bose-einstein entropy. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 218–228. Springer, 2016.

[98] J. Wang, R.C. Wilson, and E.R. Hancock. Spin statistics, partition functions and network entropy. *Journal of Complex Networks*, 5(6):858–883, 2017.

[99] J. Wang, R.C. Wilson, and E.R. Hancock. Spin statistics, partition functions and network entropy. *Journal of Complex Networks*, 5(6):858–883, 2017.

[100] J.A. Gutierrez-Perez et al. Application of graph-spectral methods in the vulnerability assessment of water supply networks. *Mathematical and Computer Modeling*, 57:1853–1859, 2013.

[101] J.F. Kaiser. On a simple algorithm to calculate the energy of a signal. In *Proc. ICASSP*, pages 381–384, 1990.

[102] J.H. Koolen and H. Yu. The distance-regular graphs such that all of its second largest local eigenvalues are at most one. *arXiv preprint arXiv:1102.4292*, 2011.

[103] Jianjia Wang, R.C. Wilson, and E.R. Hancock. Network edge entropy from maxwell-boltzmann statistics. In *Proc. ICIAP*, pages 254–264, 2017.

[104] J.M. Keller, M.R. Gray, and J.A. Givens. A fuzzy k-nearest neighbor algorithm. *IEEE transactions on systems, man, and cybernetics*, (4):580–585, 1985.

[105] C. Jordan. Sur la série de fourier. *Comptes rendus hebdomadaires des séances de l'Académie des sciences*, 92:228–230, 1881.

[106] J.P. Havlicek, D.S. Harding, and A.C. Bovik. Multidimensional quasi-eigenfunction approximations and multicomponent AM-FM models. *IEEE Trans. Imag. Proc.*, 9, 2000.

[107] X. J.Q. Li and Y.X. Yan. Quantum state representation based on combinatorial laplacian matrix of star-relevant graph. *Quantum Information Processing*, 14(12):4691–4713, 2015.

[108] J.Y. Cai, M. Fürer, and N. Immerman. An optimal lower bound on the number of variables for graph identification. *Combinatorica*, 12(4):389–410, 1992.

[109] K. Anand, G. Bianconi, and S. Severini. Shannon and Von Neumann entropy of random networks with heterogeneous expected degree. *Phys. Rev. E*, 83(3):036109, 2011.

[110] K. Kanwar, H. Kumar, and S. Kaushal. A metric to compare vulnerability of the graphs of different sizes. *Electronic Notes in Discrete Mathematics*, 63:525–533, 2017.

[111] K. Riesen and H. Bunke. *Graph classification and clustering based on vector space embedding*, volume 77. World Scientific, 2010.

[112] K. Riesen and H. Bunke. Iam graph database repository for graph based pattern recognition and machine learning. *Structural, Syntactic, and Statistical Pattern Recognition*:287–297, 2008.

[113] K.CH.Das and I.Gutman. Bounds for the energy of graphs. *Journal of Mathematics and Statistics*, 45(3):695–703, 2016.

[114] K.CH.Das and S.A.Mojallal. On energy and laplacian energy of graphs. *Electronic Journal of Liner Algebra*, 31:167–186, 2016.

[115] G. Kirchhoff. *Vorlesungen über mathematische physik: mechanik*, volume 1. Teubner, 1876.

[116] K.M. Borgwardt et al. Protein function prediction via graph kernels. *Bioinformatics*, 21(suppl 1):i47–i56, 2005.

[117] J. Kunegis. Exploiting the structure of bipartite graphs for algebraic and spectral graph theory applications. *Internet Math.*, 11(3):201–321, 2015.

[118] L. Babai and L. Kucera. Canonical labelling of graphs in linear average time. In *Foundations of Computer Science, 1979., 20th Annual Symposium on*, pages 39–46. IEEE, 1979.

[119] L. Bai et al. A quantum jensen–shannon graph kernel for unattributed graphs. *Pattern Recognition*, 48(2):344–355, 2015.

[120] L. Bai et al. An aligned subtree kernel for weighted graphs. In *International Conference on Machine Learning*, pages 30–39, 2015.

[121] L. G. Shapiro and R. M. Haralick. A metric for comparing relational descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (1):90–94, 1985.

[122] L. Han et al. Graph characterizations from von neumann entropy. *Patt. Recogn. Lett.*, 33(15):1958–1967, 2012.

[123] L. Han et al. Graph characterizations from von neumann entropy. *Pattern Recognition Letters*, 33(15):1958–1967, 2012.

[124] L. I. Rudin and S. Osher. Total variation based image restoration with free local constraints. In *Image Processing, 1994. Proceedings. ICIP-94., IEEE International Conference*, volume 1, pages 31–35. IEEE, 1994.

[125] L. J. Schaad and B. A. Hess. Hueckel molecular orbital π-resonance energies question of the sigma structure. *Journal of the American Chemical Society*, 94(9):3068–3074, 1972.

[126] L. Lacasa et al. From time series to complex networks: the visibility graph. *PNAS*, 105(13):4972–4975, 2008.

[127] L. Lacasa. Software codes and figures about visibility graph. 2009. URL: http://www.maths.qmul.ac.uk/~lacasa/Software.html (visited on 05/11/2018).

[128] L. Lau. Algorithmic spectral graph theory: online lecture notes. 2015. URL: https://cs.uwaterloo.ca/~lapchi/cs798-2015/ (visited on 02/11/2018).

[129] D. Lawden. The mathematical principles of quantum mechanics, 1995.

[130] L.I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1-4):259–268, 1992.

[131] L.Ye and E. Keogh. Time series shapelets: a novel technique that allows accurate, interpretable and fast classification. *Data Min. Knowl. Disc.*, 22(1):149–182, 2011.

[132] M. A. T. Figueiredo et al. On total variation denoising: a new majorization-minimization algorithm and an experimental comparisonwith wavalet denoising. In *Image Processing, 2006 IEEE International Conference on*, pages 2633–2636. IEEE, 2006.

[133] M. Dairyko et al. Note on von neumann and rényi entropies of a graph. *Linear Algebra and its Applications*, 521:240–253, 2017.

[134] M. Fiedler and V. Nikiforov. Spectral radius and hamiltonicity of graphs. *arXiv preprint arXiv:0903.5353*, 2009.

[135] M. Ouyang. Review on modeling and simulation of interdependent critical infrastructure systems. *Reliab. Eng. Syst. Savety*, 121:43–60, 2014.

[136] M.A. Aizerman, E.M. Braverman, and L.I. Rozonoer. Theoretical foundations of potential function method in pattern recognition. *Automation and Remote Control*, 25(6):917–936, 1964.

[137] M.A. Côté and H. Larochelle. An infinite restricted boltzmann machine. *Neural computation*, 28(7):1265–1288, 2016.

[138] M.A. Nielsen and I.L. Chuang. Quantum computation and quantum information, 2000.

[139] S. Mallat. *A wavelet tour of signal processing*. Elsevier, 1999.

[140] S. Mallat. *A wavelet tour of signal processing: the sparse way*. Academic press, 2008.

[141] J. Mercer. Xvi. functions of positive and negative type, and their connection the theory of integral equations. *Phil. Trans. R. Soc. Lond. A*, 209(441-458):415–446, 1909.

[142] R. Merris. Laplacian matrices of graphs: a survey. *J. Math. Anal. Appl.*:143–176, 1994.

[143] R. Merris. Large families of laplacian isospectral graphs. *Linear and Multilinear Algebra*, 43(1-3):201–205, 1997.

[144] M. Moshinsky and Y.F. Smirnov. *The Harmonic Oscillator in Modern Physics*. Harwood academic publishers GmbH, 1996.

[145] M.Pirani. On spectral properties of the grounded laplacian matrix. *MSc. university of Waterloo*, 2014.

[146] N. Cristianini and J. Shawe-Taylor. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, New York, 2000.

[147] N. Kashtan et al. Efficient sampling algorithm for estimating subgraph concentrations and detecting network motifs. *Bioinformatics*, 20(11):1746–1758, 2004.

[148] N. M. Kriege, P. L. Giscard, and R. C. Wilson. On valid optimal assignment kernels and applications to graph classification. In *Advances in Neural Information Processing Systems*, pages 1623–1631, 2016.

[149] N. Shervashidze et al. Efficient graphlet kernels for large graph comparison. In *Artificial Intelligence and Statistics*, pages 488–495, 2009.

[150] N. Shervashidze et al. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research*, 12(Sep):2539–2561, 2011.

[151] N. Tremblay and P. Borgnat. Subgraph-based filterbanks for graph signals. *IEEE Transactions on Signal Processing*, 64(15):3827–3840, 2016.

[152] N. Tremblay, P. Borgnat, and P. Flandrin. Graph empirical mode decomposition. In *Signal Processing Conference (EUSIPCO), 2014 Proceedings of the 22nd European*, pages 2350–2354. IEEE, 2014.

[153] N. Wale and G. Karypis. Acyclic subgraph based descriptor spaces for chemical compound retrieval and classification. Technical report, DTIC Document, 2006.

[154] N.B. Aoun, M. Mejdoub, and C.B. Amar. Graph-based approach for human action recognition using spatio-temporal features. *Journal of Visual Communication and Image Representation*, 25(2):329–338, 2014.

[155] J. V. Neumann. *Mathematical foundations of quantum mechanics*, number 2. Princeton university press, 1955.

[156] V. Nikiforov. Beyond graph energy: norms of graphs and matrices. *Linear Algebra and its Applications*, 506:82–138, 2016.

[157] V. Nikiforov. Eigenvalues and degree deviation in graphs. *Linear Algebra Appl.*, 414:347–306, 2006.

[158] V. Nikiforov. The energy of graphs and matrices. *Journal of Mathematical Analysis and Applications*, 326(2):1472–1475, 2007.

[159] V. Nikiforov. The influence of miroslav fiedler on spectral graph theory. *J. Math. Anal. Appls*, 439:818–821, 2013.

[160] N.K. Kumar and J. Schneider. Literature survey on low rank approximation of matrices. *Linear and Multilinear Algebra*:1–33, 2016.

[161] O. Macindoe and W. Richards. Graph comparison using fine structure analysis. In *Social Computing (SocialCom), 2010 IEEE Second International Conference on*, pages 193–200. IEEE, 2010.

[162] P. Barooah and J.P. Hespanha. Graph effective resistance and distributed control: spectral properties and applications. *Proc. IEEE Conf. Decision and Control*:3479–3485, 2006.

[163] P. Crucitti et al. Efficiency of scale-free networks: error and attack tolerance. *Physica A*, 320:622–642, 2003.

[164] P. Crucitti, V. Latora, and M. Marchiori. Model for cascading failures in complex networks. *Phys. Rev. E*, 69:1–4, 2004.

[165] P. Holme et al. Attack vulnerability of complex networks. *Phys. Rev. E*, 65:1–15, 2002.

[166] P.C. Crucitti, V. Latora, and M. Marchiori. Locating critical lines in high-voltage electrical power grids. *Fluctuation and Noise Letters*, 5(2):201–208, 2005.

[167] P.D. Dobson and A.J. Doig. Distinguishing enzyme structures from non-enzymes without alignments. *Journal of molecular biology*, 330(4):771–783, 2003.

[168] I. Pitas. *Graph-Based Social Media Analysis*. CRC Press, 2016.

[169] N. Pržulj. Biological network comparison using graphlet degree distribution. *Bioinformatics*, 23(2):e177–e183, 2007.

[170] Q. Liu, Z. Dong, and E. Wang. Moment-based spectral analysis of large-scale generalized random graphs. *IEEE ACCESS*, 5:9453–9463, 2017.

[171] R. Albert, H. Jeong, and A.L. Barabasi. Attack and error tolerance in complex networks. *Nature*, 406(6794):387–482, 2000.

[172] R. Albert, I. Albert, and G.L. Nakarado. Structural vulnerability of the north american power grid. *Phys. Rev. E*, 69:1–5, 2004.

[173] R. Alberta and A.L. Barabasi. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1):47–97, 2002.

[174] R. Durbin et al. *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge university press, 1998.

[175] R. Myers, R. C. Wison, and E. R. Hancock. Bayesian graph edit distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(6):628–635, 2000.

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

[177] R.K. Fan Chung. *Spectral graph theory*. AMS, 1996.

[178] R.K. Fan Chung. *Spectral graph theory*. AMS, 1996.

[179] S. Chen et al. Adaptive graph filtering: multiresolution classification on graphs. In *Global Conference on Signal and Information Processing (GlobalSIP), 2013 IEEE*, pages 427–430. IEEE, 2013.

[180] S. Chen et al. Discrete signal processing on graphs: sampling theory. *arXiv preprint arXiv:1503.05432*, 2015.

[181] S. Chen et al. Signal recovery on graphs: variation minimization. *IEEE Transactions on Signal Processing*, 63(17):4609–4624, 2015.

[182] S. Gupta et al. Analysis and, analysis and prediction of vulnerability in smart power transmission system: a geometrical approach. *Electrical Power and Energy Systems*, 94:77–87, 2018.

[183] S. Kim et al. Pubchem substance and compound databases. 2015. URL: https://pubchem.ncbi.nlm.nih.gov/ (visited on 05/11/2018).

[184] S. Perseguers et al. Quantum complex networks. *Natural Physics*, 6:539–543, 2009.

[185] S. Segarra and A. Ribeiro. Stability and continuity of centrality measures in weighted graphs. *IEEE Trans. Sig. Proc.*, 64(3):543–555, 2016.

[186] S. Segarra et al. Interpolation of graph signals using shift-invariant graph filters. In *EUSIPCO*, pages 210–214, 2015.

[187] S. Supriya et al. Weighted visibility graph with complex network features in the detection of epilepsy. *IEEE Access*, 4:6554–6566, 2016.

[188] S. Wang, J. Zhang, and N. Duan. Multiple perspective vulnerability analysis of the power network. *Physica A*, 492:1581–1890, 2018.

[189] A. Schwenk. Almost all trees are cospectral. *New directions in the theory of graphs*:275–307, 1973.

[190] I. Selesnick. Total variation denoising (an mm algorithm). *NYU Polytechnic School of Engineering Lecture Notes*, 2012.

[191] N. Shervashidze and K. Borgwardt. Fast subtree kernels on graphs. In *NIPS*, pages 1660–1668, 2009.

[192] S.J Swamidass et al. Kernels for small molecules and the prediction of mutagenicity, toxicity and anti-cancer activity. *Bioinformatics*, 21(suppl_1):i359–i368, 2005.

[193] S.L. Braunstein, S. Ghosh, and S. Severini. The Laplacian of a graph as a density matrix: a basic combinatorial approach to separability of mixed states. *Ann. Combinatorics*, 10(3):291–317, 2006.

[194] D. Spielman. *Spectral graph theory*, volume 1. Chapman and Hall CRC Press, 2011. Chapter 16, pages 1–30.

[195] D. Spielman. Spectral graph theory and its applications: online lecture notes. 2004. URL: http://www.cs.yale.edu/homes/spielman/eigs/ (visited on 02/11/2018).

[196] S.V.N. Vishwanathan et al. Graph kernels. *Journal of Machine Learning Research*, 11(Apr):1201–1242, 2010.

[197] T. Gärtner, P. Flach, and S. Wrobel. On graph kernels: hardness results and efficient alternatives. In *Learning theory and kernel machines*, pages 129–143. Springer, 2003.

[198] T. Gärtner, P. Flach, and S. Wrobel. On graph kernels: hardness results and efficient alternatives. In *Learning Theory and Kernel Machines*, pages 129–143. Springer, 2003.

[199] H. Teager and S. Teager. A phenomenological model for vowel production in the vocal tract. In *Speech Sciences: Recent Advances*, pages 73–109. College Hill Press, 1983.

[200] H. Teager and S. Teager. Evidence for nonlinear production mechanisms in the vocal tract. In *NATO Advanced Study Inst. Speech Production Speech Modeling, Bonas, France*, pages 241–261, 1990.

[201] V. Consonni and R. Todeschini. New spectral indices for molecule description. *MATCH Commun. Math. Comput. Chem.*, 1:2, 2008.

[202] V. Vapnik. *Statistical learning theory*, volume 3. Wiley, New York, 1998.

[203] V. Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.

[204] W. J. Christmas, J. Kittler, and M. Petrou. Structural matching in computer vision using probabilistic relaxation. *IEEE Transactions on pattern analysis and machine intelligence*, 17(8):749–764, 1995.

[205] W. So et al. Applications of a theorem by ky fan in the theory of graph energy. *Linear algebra and its applications*, 432(9):2163–2169, 2010.

[206] W. Yin et al. Bregman iterative algorithms for l1-minimization with applications to compressed sensing. *SIAM Journal on Imaging Sciences*, 1(1):143–168, 2008.

[207] S. Wernicke. A faster algorithm for detecting network motifs. In *International Workshop on Algorithms in Bioinformatics*, pages 165–177. Springer, 2005.

[208] W.H. Haemers and E. Spence. Enumeration of cospectral graphs. *European Journal of Combinatorics*, 25(2):199–211, 2004.

[209] X. Li, Y. Shi, and I. Gutman. *Graph energy*. Springer, New York, 2012.

[210] X. Yang et al. Incidence graphs constructed from t-designs. *Appl. Anal. Discr. Math*, 10:457–478, 2016.

[211] J. X.B. Chen and Y.X. Yang. Quantum state representation based on combinatorial laplacian matrix of star-relevant graph. *Quantum Information Processing*, 14(12):4691–4713, 2015.

[212] Y. Shi and M. Macy. Measuring structural similarity in large online networks. *Social science research*, 59:97–106, 2016.

[213] Y. Wang et al. A new alternating minimization algorithm for total variation image reconstruction. *SIAM Journal on Imaging Sciences*, 1(3):248–272, 2008.

[214] Y. Zhou, H. Cheng, and J. X. Yu. Graph clustering based on structural/attribute similarities. *Proceedings of the VLDB Endowment*, 2(1):718–729, 2009.

[215] Y.S. Abu-Mostafa, M. Magdon-Ismail, and H.T. Lin. *Learning from data*, volume 4. AMLBook New York, NY, USA: 2012.

[216] Y.Z. Song et al. Finding semantic structures in image hierarchies using laplacian graph energy. *Computer vision–ECCV 2010*:694–707, 2010.

UNIVERSITE SCIENCES BRETAGNE DE LA MER LOIRE ET DU LITTORAL

UBO Université de Bretagne Occidentale

Titre : Classification des signaux et des graphes par approches spectrales algébriques.

Mots clés : Graphes, Analyse spectrale, Classification de graphes, Entropie de Von Neumann, Energie d'un graphe, Représentation quantique, Vulnérabilité des réseaux.

Résumé : De nos jours, le développement de l'instrumentation électronique, de l'informatique et des systèmes de communications conduit à une collecte de données réalisée à partir de réseaux de capteurs (réseau de bouées acoustiques en mer, capteurs de température des stations météorologiques, capteurs de surveillance des niveaux de pollution et de bruit ...). La complexité de ces réseaux de capteurs et leur interaction font que ces données sont portées par des structures complexes et irrégulières qui ne peuvent être traitées efficacement par les outils standards. Les graphes constituent un modèle mathématique pour la représentation de telles données en tenant compte de leur complexité. L'objectif principal de ce travail de thèse est d'étudier la question de la pertinence de cette représentation en se focalisant sur l'interaction données-structure d'une part, et d'autre part sur la modélisation matricielle de la structure du graphe portant ces données. Ces questions sont traitées dans le cadre de la classification des signaux et des graphes en utilisant des outils de la théorie spectrale des graphes. De nouvelles mesures de similarités spectrales entre graphes ont été proposées et testées sur des données synthétiques et réelles donnant de bons résultats en termes de temps de calculs et de taux de bonne classification par rapport à l'état de l'art.

Malgré la simple relation linéaire associant les matrices Laplacienne et d'adjacence, les résultats obtenus mettent en évidence le fait que ces matrices expriment différemment l'information structurelle du graphe. Cette différence de représentation a été analysée et illustrée via des mesures de complexité et de connectivité à partir du graphe associé, ainsi qu'en mesurant la déviation de l'entropie de Von Neumann du graphe, considéré alors comme un système quantique. Dans le cadre de la représentation et de l'analyse spectrale des graphes, nous nous sommes intéressés au cas particulier des graphes co-spectraux, graphes partageant le même spectre. De plus, en considérant la théorie de la décomposition en matrices de rang faible, l'analyse en graphe propre dominant, appelé DGA (pour Dominant eigenGraph Analysis), a été introduite et illustrée par la décomposition multi-échelle de la structure du graphe. En utilisant une reconstruction partielle de la matrice d'adjacence par ses graphes propres, une stratégie facilitant la détection de communautés au sein d'un graphe a été proposée. Concernant la représentation quantique du graphe, nous avons exploité l'entropie de Von Neumann pour mesurer la vulnérabilité du graphe aux perturbations structurelles. Un nouvel algorithme de pondération des connections a été ainsi proposé.

Title : Classification of signals and graphs by algebraic spectral approaches.

Keywords : Graphs, Spectral analysis, Classification of graphs, Von Neumann entropy, Energy of a graph, Quantum representation, Vulnerability of networks.

Abstract : Nowadays, the development of electronic instrumentation, data processing and communications systems leads to a collection of data carried out from networks of sensors (network of acoustic buoys at sea, temperature sensors of meteorological stations, sensors monitoring pollution and noise levels, etc.). The complexity of these sensor networks and their interaction mean that these data are carried by complex and irregular structures which cannot be processed efficiently by standard tools. The graphs constitute a mathematical model for the representation of such data taking into account their complexity. The main objective of this thesis is to study the question of the relevance of this representation by focusing on the data-structure interaction on the one hand, and on the other hand on the matrix modeling of the structure of the graph. carrying this data. These questions are addressed within the framework of the classification of signals and graphs using tools of spectral graph theory. New measurements of spectral similarities between graphs have been proposed and tested on synthetic and real data giving good results in terms of calculation time and good classification rate compared to the state of the art.

Despite the simple linear relationship between the Laplacian and adjacency matrices, the results obtained highlight the fact that these matrices express the structural information of the graph differently. This difference in representation was analysed and illustrated by measuring the complexity and connectivity of the associated graph, as well as by measuring the deviation of the Von Neumann entropy of the graph, which was then considered as a quantum system. In the context of the representation and spectral analysis of graphs, we are interested in the particular case of co-spectral graphs, graphs sharing the same spectrum. Moreover, considering the theory of low rank matrix decomposition, the dominant eigengraph analysis, called DGA (for Dominant eigenGraph Analysis), has been introduced and illustrated by the multi-scale decomposition of the graph structure. Using a partial reconstruction of the adjacency matrix by its eigengraphs, a strategy facilitating the detection of communities within a graph was proposed. Concerning the quantum representation of the graph, we exploited the Von Neumann entropy to measure the vulnerability of the graph to structural perturbations. A new algorithm of connection weighting has been proposed.