MAP 595 - Internship Report

Dario Shariatian

August 25, 2022

Enhancing Linkages - Spectral Graph Theory

Abstract. In this report we collect tools from spectral graph theory to devise new ways of analyzing graphs of linked companies in the stock market. We do not modify the initial graph and use it as the only source of information about relations between stocks. Firstly we will focus on spectral embedding, which consists in using linear algebra to assign a position in Euclidian space to the different companies, such that those tightly related get close together. This embedding holds many interesting properties, which we will discuss, and offers a convenient setting in order to subsequently apply any machine learning algorithm. Secondly we will explore the notion of entropy for graphs. This will be interesting to determine a classification of graphs, according to their inner structure, and detect regime change over time. The setting for such a study is directed weighted graphs, where weights represent the influence we believe company i has on company j. However most techniques described in the literature only focus on undirected graphs. Thus we will provide all the mathematics and justifications for generalization to the directed graph setting. Overall, the various notions, objects and algorithms will be introduced and explained in depth, so the reader can learn and differentiate them, know how and when they should be applied, and develop the necessary intuitions.

Keywords: Machine Learning, Graph Laplacian, Spectral Embedding, Graph Entropy, Directed Weighted Graphs

Declaration of Academic Integrity

I declare that the work that I hereby submit is my own work, and that contributions from other sources are fully acknowledged.

Confidentiality Note

This work has been produced using data and technologies provided by Squarepoint Capital and is to be used by the company. Confidentiality agreements require no proprietary information to be disclosed ; as such I will not provide details on how the data is collected, processed and interpreted.

Contents

Introduction

In almost every scientific fields using empirical data, may it be biology, physics or social sciences, one is interested in determining groups of elements displaying similar behaviour. In finance, we often encounter what we call 'linkage graphs', which represent influence companies have on each other. Typically, the rate of return of linked stocks tend to show significant correlation. Part I will summarize the context of our work. These graphs can be constructed in many different ways and from many different datasets. However, studying this plain graph object is rather difficult. Eventhough one can develop a systematic approach to assess the performance of different linkages, building a proper theoretical framework to understand their structure is important. It will let us visualize and compare linkage graphs much more easily, with insights in what makes their specific properties.

Spectral theory of graphs is our way into these endeavours. This theory studies mathematical objects we can associate to a graph (e.g adjacency matrix, degree matrix, Laplacian...) in the eyes of linear algebra to try to get results and insights on the inner structure of such objects. For instance, we can get estimations of the connectedness of a graph (Fiedler number, Cheeger's inequality, centrality measures...), the heterogeneity of node degrees (entropy), or natural embedding in the Euclidian space.

In our context, the interpretatiblity of the results makes this point of view interesting and valuable. Contrary to other ML techniques, it is possible to attribute clear meanings to the outputs, and determine what leads to observed behaviours.

Many different techniques exists. In this paper our main focus will be on :

- **Embedding** Study of the embeddings given by the graph Laplacian. Part II will introduce embeddings for undirected graphs, the different algorithms/Laplacian that exist and generalization to directed graphs. In Part III we will get to a similar embedding using a distance defined as "commute time" between companies. Much more intuitive, this will be a better approach to someone with background in mathematics for finance. We will even make use of PCA. Finally in Part IV we will introduce the tools available and the corresponding behaviours to look for in linkages. For instance we will see that clustering or "hub"-ness effects all materialize differently in these embeddings and in their corresponding graph spectrum.

- **Entropy** This will be of interest to determine the stability of linkage graph structure and detect regime change. We may also use it to classify graphs, or enable the possiblity to determine high similarity between linkage types. Part V will discuss entropy in details.

Initially, the stability of the embeddings was the main reason we were interested in such a study : if a relevant structure were discovered in the linkage graph embedding (in terms of market forecasting power), then we could expect that pattern to repeat in the future and rely on its predictions.

Part I
Context - Quant Research

1 Squarepoint Capital

Squarepoint Capital is a hedge fund specializing in quantitative finance and systematic trading. It invests money from its clients in financial markets, using different inventive products and strategies so as to make profit no matter market trends. The goal is to understand price fluctuations in high, mid or low frequency through systematic/algorithmic ways, anticipate them and place trades.

The company focuses on a collaborative approach, so as to develop shared infrastructures and knowledge, allowing more complex projects to be built without the pressure of generating higher returns immediately.

2 Equity - Mid Frequency

I worked in the Equity - Mid Frequency team. The goal is to analyze the behaviour of traditional stocks in a time scale ranging from a few hours to a few days.

The usual workflow is the following :

- **Data** Focus on a particular data set ; this could be coming from an external data provider or from inside the company. In the end it is any way to collect information to be used on a subset of stocks in the market
- **Alpha/Signal** Find a systematic strategy/set of rules with a clear and sound economic interpretation to forecast returns of such and such stocks. Usually, this mathematically translates to an "alpha". It basically means that we are able to predict the price evolution of an asset on the market. When we have a situation when we are very confident in our prediction and we want to act on it, we say we generate a "signal"
- **Backtest** Thoroughly test this strategy on past data, to solidly statistically determine if that signal is relevant or not.
- **Production** Trade on it to aim to generate profit.

3 Analyzing Market

We will describe through a few simple concepts how we are analyzing and looking at data. For the sake of simplicity, we will say that we are looking at the price of stocks at close, day after day. Let's write the value of the stock of a company i on day t

$$S_t^i.$$

Then the n-day future return from day t of this stock is naturally defined to be

$$r_{t,n}^i = \frac{S_{t+n}^i}{S_t^i}.$$

3.1 Hedged returns

As one may remark, stock markets are fluctuating along global trends, describing how much investors are globally pouring in or out of this financial market. We want to hedge ourselves against this behaviour. The simplest way to do that is to short an index future moving along the stock market (e.g S&P 500) along with the desired stock. Let's denote by *hedge* the value of the underlying this hedge instrument is tracking. If the value at time t of hedge is hedge(t), then shorting the hedge instrument for one day will account for the following returns:

$$-\frac{\text{hedge}(t+1)}{\text{hedge}(t)}.$$

Indeed we are "removing" the global market movement. If the market globally moves up, we lose its corresponding rate of return, and inversely. The n-day future hedged return then is:

$$h^i_{t,n} = \sum_{k=0}^{n-1} r_{t,k} \left(r_{t+k,1} - \frac{\text{hedge}(t+k+1)}{\text{hedge}(t+k)} \right),$$

which is emulating the pnl of buying a stock on date t, rehedging every day dollar for dollar with the hedge instrument, while never rebalancing the dollar value of held stock. We maintain constant number of shares invested. It is clear on the formula how we remove the global market movements: if the stock we are interested in beats the market **even when the market is plunging**, then we are profitable.

Looking at hedged returns is one of the standard ways to assess the performance of our strategies. In this report we won't precisely describe how this is done, but this is good to know to understand how data is interpreted.

See [BRR94] for a more precise description of simple models like CAPM (one risk factor, the expected return of the market) and APT [GK94] (more solid multi factor model, introducing notion of risk-return tradeoff and risk exposure profile).

3.2 GICS

The Global Industry Classification Standard (GICS) is a four-tiered, hierarchical industry classification system :

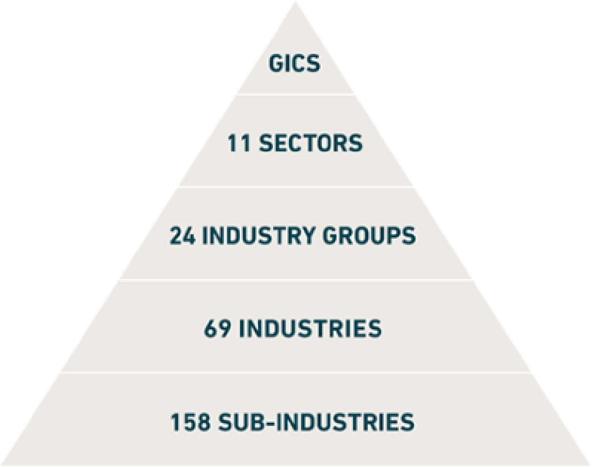


Figure 1: GICS hierarchy [MSC22]

Companies are classified quantitatively and qualitatively. Each company is assigned a single GICS classification at the Sub-Industry level according to its principal business activity. This classification is used extensively to study stocks, as same category companies often display similar behaviour. It is a first step into clustering companies. In our subsequent work on the *linkages*, we will be careful not to replicate the information contained in this classification. There would be no need to develop a complex theoretical framework if the final use cases are just equivalent to looking at intra-gics relations.

3.3 Linkage Graphs

It seems natural that companies of the same sector, competitors, suppliers etc. will behave relatively similarly on the stock market. These links could exist for long or short periods of time. For instance, say some big game console is being released, with CPU chips manufactured by Intel, and GPUs provided by AMD. Then we can expect these two companies to behave similarly for some period of time. We can express these bonds in a **linkage graph**, day after day. We will sometime simply refer to these graphs as **linkages**. There are many different way to construct these graphs. The GICS is a good example of such a linkage, where companies of each sub-industry could be interconnected in multiple distinct complete graphs. There is already quite an extensive literature on the subject, the curious reader can look at [TLM08] for a more general linkage discussion, and example of study leading to the construction of such a graph.

Usually, most of our linkages have the following structure :

- **Directed** They are directed graphs. We want to be able to model the situation where one company behaviour influences another one more or less later in time; sometimes it has more to do with causation than correlation.
- **Sparse** The graph is sparse, that is the set of edges is of relatively small cardinality. The linkage graph is describing more of clear connections between some companies than just being a dense similarity graph acting on the whole set of companies

Our work is going to focus on the study of these graphs. We try to answer the following type of questions:

- Is it possible to determine that some links/companies are more important than others?
- Is it possible to determine and discard "unrelevant" links (as in we can predict they won't really be of importance)?
- What kind of quantitative measure can we build to analyze and compare linkage graphs, between themselves or over time. Will this allow to identify regime change in a specific linkage?

Part II
Spectral Graph Theory for Embedding

4 Elements of Graph Theory

We will denote graphs we are working on by $G = (V, E)$, V being the set of vertices (representing companies) and $E \subset V \times V$ the set of edges (representing links). To each edge $e = (v_i, v_j)$ is associated a weight w_{ij} describing the strength of the linkage between company i and company j. Thus a useful representation of the graph is going to be its adjacency matrix W:

$$W = (w_{ij}).$$

If no edge exist between v_i and v_j, we have $w_{ij} = 0$. Remark that for an undirected graph, W is symmetric: for each edge from company i to company j of weight w_{ij} there must exist an edge entry in the opposite direction and of the same weight: $w_{ij} = w_{ji}$.

We define the outgoing degree of a vertex v_i to be:

$$d_i = \sum_j w_{ij},$$

which will be just called the degree of the vertex in the case of the undirected graph. We then define the degree matrix to be the diagonal matrix of the degrees:

$$D = \text{diag}(d_i).$$

Take $A \in V$ a subset of the nodes. As a shorthand we will define $i \in A$ and $i \in \{i \;/\; v_i \in A\}$ to be equivalent. We say that $G_A = (A, E \cap A \times A)$ is the subgraph of G induced by A. We denote by $\mathbf{1}_A = (\mathbf{1}_A(v_i))_i \in \mathbb{R}^n$ the indicator vector of the subset A. Moreover:

$$|A| = \#\{i/v_i \in A\} = \mathbf{1}_A^T \mathbf{1}_A = \|\mathbf{1}_A\|^2,$$

$$\text{vol}(A) = \sum_{i \in A} d_i = \mathbf{1}_A^T D \mathbf{1}_A = \|D^{1/2}\mathbf{1}_A\|^2.$$

So these are differently weighted measure of the subset "size". We say that A is connected if there exists a path in G_A between any two vertices. For two subsets $A, B \subset V$, we define the similarity between A, B to be:

$$W(A, B) = \sum_{i \in A, j \in B} w_{ij} = \mathbf{1}_B^T W \mathbf{1}_A.$$

Remark that we do not require A and B to be disjoint. This value can thus measure the intra-subset similarity of A with $W(A, A)$ or the isolated-ness of A with $W(A, \overline{A})$, where $\overline{A} = V \setminus A$.

5 Spectral Embedding - Undirected Graphs

For the whole section here, we will only focus on undirected graphs. They offer a very convenient symmetry which simplifies the whole analysis.

5.1 Spectral Graph Theory

We recommend looking at [Lux07] for an excellent description of spectral embedding for undirected graphs. While less thorough, [Bon19] is also a good resource. It explores the "energy" approach and in particular uses systems of n springs as an intuitive physicist point of view.

We will now describe how we get to the graph embedding, its properties and the useful interpretations we can get with this method.

5.1.1 Unnormalized Graph Laplacian

The unnormalized graph Lapacian matrix is defined as

$$L = D - W.$$

Remark that is is a real symmetric matrix. For the moment the Laplacian is introduced through the lens of 'energy'. The reader more familiar with finance in mathematics may prefer the alternative approach described in Part III.

Theorem 5.1. Laplacian as an energy operator

Take $x \in \mathbb{R}^n$. This vector represents values assigned to each company. Then:

$$2x^T Lx = \sum_{ij} w_{ij}(x_i - x_j)^2.$$

Proof. Remark that, by symmetry :

$$x^T Lx = \sum_i d_i x_i^2 - \sum_{ij} w_{ij} x_i x_j = \sum_j d_j x_j^2 - \sum_{ij} w_{ij} x_i x_j.$$

Adding these and remembering that $d_i = \sum_j w_{ij}$:

$$2x^T Lx = \sum_i d_i x_i^2 + \sum_j d_j x_j^2 - 2\sum_{ij} w_{ij} x_i x_j = 2\sum_{ij} w_{ij}(x_i^2 + x_j^2 - 2x_i x_j),$$

$$2x^T Lx = \sum_{ij} w_{ij}(x_i - x_j)^2.$$

□

There are a lot of ways to interpret this equation. They are all going to be explained throughout the study. The simplest one is the following : consider L to be an energy operator acting on the graph. We say this because it is a Hermitian linear operator, so it will admit a nice eigendecomposition adaped to this energy description. To any state of the system/positions of the companies $x \in \mathbb{R}^n$:

$$< x|L|x >= x^T Lx.$$

Remark how the total energy is the sum of multiple contributions of harmonic oscillators. This is modulated by the factors w_{ij}; as this factor increases, we need the values x_i, x_j to be all the closer for the same energy level. The minimization of this value would intuitively give a nice 1-d embedding where all the "related" companies get close together, and those loosely connected in the graph far apart in the embedding. These intuitions will be made more and more precise. For the moment let us understand this mathematical object more precisely.

Corollary 5.1.1. L is symmetric and positive semi-definite. Its eigenvalues are positives

Proof. All the weights w_{ij} being positive it is clear that $\forall x \in \mathbb{R}^n,\ x^T Lx \geq 0$ □

Theorem 5.2. Null space of L when G is connected
Suppose G is a connected graph. Then the null space of L is of dimension 1 and is spanned by the constant one vector $\mathbf{1}$.

Proof. Suppose $x \in \mathbb{R}^n$ is in the null space of L. Then $x^T Lx = \sum_{ij} w_{ij}(x_i - x_j)^2 = 0$ is equivalent to $x_i = x_j$ when $w_{ij} \neq 0$. Thus all the points in the same connected component of the graph have the same value. Since the graph is connected x is a multiple of $\mathbf{1}$. The reciprocal way is trivial. □

Null space of L, general case Suppose G is made of $G_1, ..., G_k$ disjoint connected components. Reorder the nodes so the adjacency matrix and the Laplacian are block diagonal:

$$L = \begin{pmatrix} L_1 & & & \\ & L_2 & & \\ & & \cdots & \\ & & & L_k \end{pmatrix}.$$

Then the eigendecomposition of L is the union of the egendecomposition of each of its subcomponent L_k. In particular,

Corollary 5.2.1. Spectrum of L and Connected Components
The dimension of the null space of L is equal to k the number of distinct connected components $G_{A_1}, ..., G_{A_k}$ in the graph. The nullspace is spanned by the indicator vectors $\mathbf{1}_{A_1}, \cdots, \mathbf{1}_{A_k}$.

We continue to understand why we are interested in this object for clustering; in the case of clusters that are perfectly isolated, the Laplacian is able to describe them perfectly.

5.1.2 Normalized Graph Laplacian

We mathematically introduce a close relative to the Laplacian, the normalized Laplacian. From now on we will differentiate both formulation. We will compare their usage and properties throughout the study. Mainly, they tackle similar but slightly different optimization functions on the graph, so that depending on the situation, one is more suited than the other. There are different variations of the normalized Laplacian in the literature. The two most cited ones [Lux07] are:

$$L_{sym} = D^{-1/2} L D^{-1/2},$$

$$L_{rw} = D^{-1} L.$$

The symmetric normalized matrix L_{sym} is usually what people refer to when talking about normalized Laplacian, and that is what we will do here. Remark that for $x \in \mathbb{R}^n$:

$$2x^T L_{sym} x = \sum_{ij} w_{ij} (\frac{x_i}{\sqrt{d_i}} - \frac{x_j}{\sqrt{d_j}})^2.$$

The points are weighted by the inverse of their relative importance in the graph, importance defined by their degree.

The random walk matrix L_{rw} (unsurprisingly) defines the transition matrix of a Markov process on the graph, where the probability of transition between company i and company j is proportional to w_{ij}:

$$L_{ij} = \frac{w_{ij}}{\sum_j w_{ij}}.$$

It may seem strange to define L_{rw} as given since it is not symmetric, while we emphasized it was one of the main property we required of our matrices. However it still admits an eigendecomposition similar to the other ones and all the different Laplacians are related by the following equations :

Theorem 5.3. Relations between Laplacians

1. $L \longleftrightarrow L_{sym}$.

   λ is an eigenvalue of L_{sym} with eigenvector u if and only if $w = D^{-1/2}u$ and λ are the solutions to the generalized eigenvalue problem

   $$Lx = \lambda Dx.$$

2. $L \longleftrightarrow L_{rw}$.

   λ is an eigenvalue of L_{rw} with eigenvector u if and only if they are the solutions to the generalized eigenvalue problem

   $$Lx = \lambda Dx.$$

3. $L_{sym} \longleftrightarrow L_{rw}$.

   λ is an eigenvalue of L_{sym} with eigenvector u if and only if $w = D^{-1/2}u$ and λ are an eigenpair of L_{rw}

Proof.

1. Take (λ, u) eigenpair of L_{sym}. Then

   $$L(D^{-1/2}u) = D^{1/2}\lambda u = \lambda D(D^{-1/2}u).$$

2. Same kind of argument

3. Same kind of argument

□

Moreover, we have the following properties in terms of distinct connected components:

Corollary 5.3.1. Spectrum of L_{sym}, L_{rw} and Connected Components The dimension of the null space of L_{sym}, L_{rw} is equal to k the number of distinct connected components $G_{A_1}, \cdots, G_{A_k}$ in the graph. Their nullspace is spanned by

- L_{sym} : the indicator vectors $(D^{1/2}\mathbf{1}_{A_1}, \cdots, D^{1/2}\mathbf{1}_{A_k})$.
- L_{rw} : the indicator vectors $(\mathbf{1}_{A_1}, \cdots, \mathbf{1}_{A_k})$, alike L.

5.2 Embedding

We will tackle the description of spectral embedding borrowing physics terminology.

5.2.1 Unnormalized

As stated earlier, L can be seen as an energy operator acting on the graph. L being symmetric, we can look at its eigendecomposition and deduce the different states (or eigenvectors) associated with this operator:

$$L = U\Lambda U^T, \quad \hat{x} = U^T x,$$

where U is the basis of eigenvectors, Λ the diagonal matrix of eigenvalues. The eigenvectors $(u_k)_k$ of L successively minimize the value

$$2x^T Lx = \sum_{ij} w_{ij}(x_i - x_j)^2,$$

when taken in increasing eigenvalue order. By successively we mean each new eigenvector is the minima of that function on the space orthogonal to the first eigenstates. This is of course a one of linear algebra main results, but let us emphasize it once again with its formal description. With $Vect(x_1, x_2, ...)$ the subspace spanned by the vectors $x_1, x_2, ...$, we have

$$\text{argmin}_{x\in\mathbb{R}^n, \|x\|=1,\ x\perp Vect(u_1,\cdots,u_k)}\ x^T Lx = u_{k+1}.$$

The (u_k) act as discrete functions such that they are going from the smoothest to the roughest variations between connected nodes. Since we are interested in geometrically concentrating related nodes, we want minimum distance variations between them; thus we look for the first eigenvectors of L.

Remark Keep in mind that the Laplacian helps us describe the quality of an embedding in terms of similarity between nodes. In this process, we want to minimize the measure of separatedness/energy it is encoding. This may seem counterintuitive because we are used to looking at the highest eigenvalues when studying matrices (for instance PCA, where we are trying to explain maximum variance), because that gives the best approximation in L^2 norm. We will see a connection with traditional PCA in Section 9.

Why work with L rather than W ? A first element of response would be to see how incorporating the degree information was necessary to get the perfect cluster description in the case of distinct connected graphs. Working with the Laplacian captures best the global structure of the graph. In the following sections, especially when describing the commute time construction in III, it will become clear how global notion of distances are described with this matrix, whereas this is not possible with the mere adjacency matrix. Some deeper work, as can be found in [RHK21] and [OLV14], specifically link the Laplacian to more general embedding methods via Kernel PCA.

Anyway let us recall the main property again : **important clusters and related nodes will group together**. Examining a bit further the eigendecomposition we see :

- The first $k \geq 1$ eigenvectors are indicators $\mathbf{1}_{A_i}$ of the distinct connected components in our graph. The associated energy is 0 : this is a "perfect" inter-cluster separation (BUT the intra-cluster points are all merged together in a single point)

- Remark that we need k dimensions to describe these: the points of cluster k will have non null value on the k-th components, 0 elsewhere.
- Say for simplicity we have a connected graph ($k = 1$). First eigenvector u_1 is constant. Now the second one u_2 gives us a non pathological description of the nodes on a 1-d line. Remember we must have (and that is true of all subsequent eigenvectors):
$$< u_1, u_2 >= \mathbf{1}^T u_2 = 0 \text{ and } \|u_2\| = 1.$$
So the positions it describes are centered and of unit variance. We obtained a discriminatory way to look at points where closer ones are more tightly connected on the graph, and inversely so.

Now we want to justify looking at k dimensions simultaneously. Everything works fine with Euclidian distance and $\mathbb{L}^2$ norm.

Theorem 5.4. ***k*-d embedding** ith X a matrix in $\mathcal{M}_{k,n}$ describing a R^k embedding of each node i, we have :
$$tr(XLX^\top) = \sum_{i,j} w_{ij} \|X_i - X_j\|^2.$$
Minimum reached for X the matrix of first k eigenvectors of L.

Thus we finally embbed our graph nodes in k-dimensions with:
$$\begin{pmatrix} u_{2,1} & u_{2,2} & \cdots & u_{2,n} \\ \cdot & & & \\ u_{k+1,1} & u_{k+1,2} & \cdots & u_{k+1,n} \end{pmatrix} = \begin{pmatrix} u_2^T \\ \cdot \\ u_{k+1}^T \end{pmatrix}.$$
We omit constant offset u_1^T (we will consider our graphs to be connected from now on).

5.2.2 Normalized

There are two ways to obtain an embedding with the normalized Laplacian.

L_{rw}, **Shi and Malik.** [SM00]
Instead of solving the eigenproblem for L
$$Lu = \lambda u.$$
We rather solve the generalized eigenproblem:
$$Lu = \lambda Du.$$
Remark that we obtain the eigenvectors of the normalized Laplacian L_{rw}. Another way to solve this eigenproblem is to find an eigenvector v of L_{sym} and set $u = D^{-1/2}v$.

L_{sym}, **Ng, Jordan, and Weiss** [NJW01]
See [Lux07] for additional details. It is known as the normalized spectral clustering according to Ng, Jordan, and Weiss ; its properties have not been explored in this study. It computes the k first eigenvectors of L_{sym}, call them $(u_k)_k$, and normalizes them as follows:
$$u_{ij} := \frac{u_{ij}}{\sqrt{\sum_k u_{ik}^2}}.$$
This methods empiricially performs poorly with our linkage graphs, with less stability in its embedding day after day. See [Lux07] or look at section 10 for clearer details on why that may be the case.

5.3 Connection to Network Theory

Initially, the algebraic study of graphs comes from relations between graph cut problems and the Laplacian matrix. Finding clusters in a graph in the eyes of a network theorist has more to do with finding specific partitions satisfying some lower or higher bound equations. This subsection will tie together these points of views, and constitutes in fact the historic introduction of spectral graph theory.

5.3.1 Relaxation of Graph Cut

Given a graph G with adjacency matrix W, the most direct way to construct a partition of the graph is to solve the mincut problem. Take k the number of clusters we wish to isolate. The mincut approach consists in finding a partition $A_1, ..., A_k$ of G such that

$$\text{cut}(A_1, .., A_k) = \frac{1}{2}\sum_{i=1}^{k} W(A_i, \overline{A}_i)$$

is minimized. We would like to find a partition such that edges between different groups have low weights. Here however high intra cluster connections is not optimized. We would like the subsets $A_1, ..., A_k$ to be reasonably large. Thus we work on a slightly different formulation of that problem which aim to balance cluster size:

$$\text{RatioCut}(A_1, ..., A_k) = \sum_i \frac{\text{cut}(A_i, \overline{A}_i)}{|A_i|}.$$

$$\text{NCut}(A_1, ..., A_k) = \sum_i \frac{\text{cut}(A_i, \overline{A}_i)}{\text{vol}(A_i)}.$$

Both notions of size will relate to our two different Laplacian; unnormalized corresponds to $|A_i| = \|\mathbf{1}_A\|^2$, normalized to $\text{vol}(A) = \|D^{1/2}\mathbf{1}_A\|^2$.

Case k = 2 Let's make these connections clear. Please refer to [Lux07] for full details.

RatioCut Write $r = \sqrt{\frac{|\overline{A}|}{|A|}} = \frac{\|\mathbf{1}_{\overline{A}}\|}{\|\mathbf{1}_A\|}$. Define f to be the following vector

$$f = r\mathbf{1}_A - \frac{1}{r}\mathbf{1}_{\overline{A}}.$$

Then

$$f^T L f = |V|\text{RatioCut}(A, \overline{A}).$$

Additionaly we have $f^T\mathbf{1} = 0$. So our minimization can be rewritten

$$\min_{A \subset V} f^T L f, \quad \text{subject to} \quad f^T\mathbf{1} = 0.$$

This is a discrete NP-hard optimization problem. However its most natural relaxation is

$$\min_{f \in \mathbb{R}^n} f^T L f, \quad \text{subject to} \quad f^T\mathbf{1} = 0,$$

which boils down to finding the second eigenvector of L. The partition A is then determined by $(\mathbf{1}_{f_i > 0})_i$.

NCut This time write $r = \sqrt{\frac{\mathrm{vol}(\overline{A})}{\mathrm{vol}(A)}} = \frac{\|D^{1/2}\mathbf{1}_{\overline{A}}\|}{\|D^{1/2}\mathbf{1}_A\|}$. Define f to be the following vector

$$f = rD^{1/2}\mathbf{1}_A - \frac{1}{r}D^{1/2}\mathbf{1}_{\overline{A}}.$$

Then

$$f^T L_{sym} f = |V| \cdot \mathrm{NCut}(A, \overline{A}).$$

Additionaly we have $f^T D^{1/2}\mathbf{1} = 0$. So our minimization can be rewritten

$$\min_{A \subset V} f^T L_{sym} f, \quad \text{subject to} \quad f^T D^{1/2}\mathbf{1} = 0.$$

Again, the relaxation of this problem boils down to finding the second eigenvector of L_{sym}. Remember its first eigenvector is indeed $D^{1/2}\mathbf{1}$.

5.3.2 Cheeger Inequality

In 1970, Jeff Cheeger proved an inequality between the 'Cheeger' isoperimetric constant, and the first non null eigenvalue of the Laplacian operator in the context of Riemannian manifold. This very influential idea inspired the analogous theory studied here. That is how algebraic study of graphs, in the way it is done for clustering, was born.

If we adapt the terminology of differential geometry by viewing the graph as a discretization of a Riemannian manifold, then $W(A, \overline{A})$ is a measure of the boundary of A, and $\mathrm{vol}(A)$ is naturally the volume of A. We can define

$$h_G(A) = \frac{W(A, \overline{A})}{\min(\mathrm{vol}(A), \mathrm{vol}(\overline{A}))},$$

and we call

$$h_G = \min_A h_G(A)$$

the Cheeger constant of the graph. The Cheeger inequality ([Chu97]) for graphs is the following

$$2h_G \geq \lambda_2 \geq \frac{h_G^2}{2},$$

where λ_2 is the smallest non null eigenvalue of the graph Laplacian. The Cheeger constant can be seen as a measure of maximum "bottleneck" of information diffusion in the graph. For instance, it will govern the convergence rate of the Markov Chain induced by the graph. It also quantitatively describes the quality of the best cut that can exist between two clusters in that graph.

See Chung [Chu97] for an insightful generalisation on weighted undirected graph. The generalization to directed graphs is also famously due to her in [Chu05]; she establishes the exact same inequality after adapting some notions to the directed graph setting.

5.4 Equivalent Physical Systems

5.4.1 Springs

$$\frac{\partial^2 x}{\partial t^2} = -Lx$$

Imagine spring connecting each nodes, stiffness being equal to linkage weight. Imagine the nodes being constrained to a single dimension (which changes nothing as we would project the PDE on each spatial dimension). Positions are then described by a single vector $x \in \mathbb{R}^n$. We are trying to find configurations of x minimizing the total system energy. We are thus naturally drawn to a study of eigenspaces.

Speaking of energy relates a lot to this interpretation. Energy here is the potential energy of an harmonic oscillator. Remark that for free moving oscillator energy=eigenvalue $\propto$ frequency

5.4.2 Heat Diffusion

$$\frac{\partial x}{\partial t} = -Lx$$

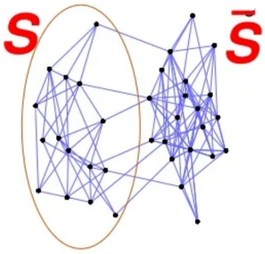
Imagine nodes being given a temperature (described by $x \in \mathbb{R}^n$), conductivity between nodes given by linkage weight. Then we are studying the process of temperature diffusion in this network. Faster the configuration tends to constant temperature, lowest its 'heat' bottleneck. Writing once again $(u_i)_i$ the eigenvectors of L, the dynamics of a diffusion process give us:

$$u_i(t) = \alpha_i(t)u_i, \quad \frac{\partial u_i}{\partial t}(t) = -Lu_i(t).$$

Thus $\alpha'(t)u_i = -L\alpha(t)u_i = -\lambda_i\alpha(t)u_i$ and

$$u_i(t) = e^{-\lambda_i t}u_i(0).$$

We could say that eigenvalues describe how well information is separated in the network. Imagine two practically separated clusters : we have a conductivity bottleneck, and then λ_2 value is low:



5.5 An example

Here the technique is applied to a linkage graph constructed from the highest correlation values of a selection of European stocks around 2011.

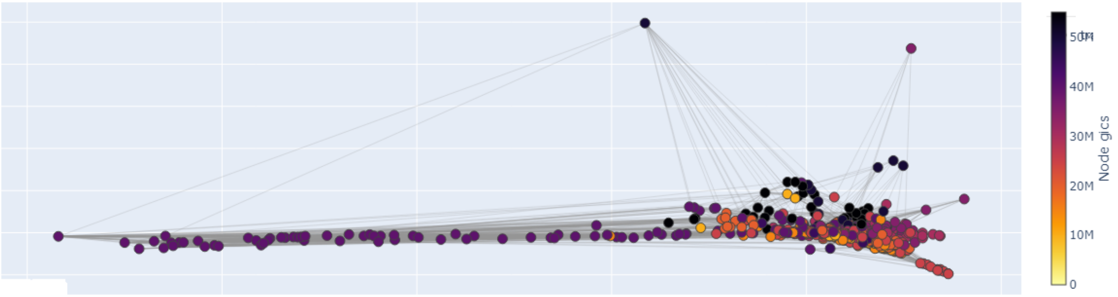


Figure 2: 2D embedding, (X, Y) plane

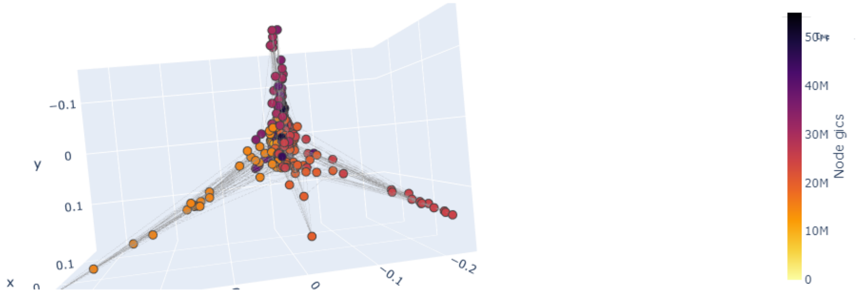


Figure 3: 3D embedding, (Y, Z) plane

The companies have been colored by GICS subindustry. Eventhough the first few dimensions seem to isolate same gics companies out of others (on the branches/cones), we visually see how different sectors are effectively mixed closer to the dense center. Looking a bit closer to this picture, one could see interesting added information over industry grouping. For instance the following are close together :

- Sanofi, Bayer (pharmaceutical) and Air Liquide (Industrial gasses)
- Tenaris (steel pipes manufacturer), Scannia AB (Swedish heavy vehicles manufacturer), Aixtron (semi conductor). More generally, dense mix of manufacturers and micro electronics (e.g STMicroelectronics) in some regions of space
- Lafarge, Holcim (merged in 2015), Saint Gobin, Nordea Bank
- Vestas Wind systems (wind turbine, energy-alternate), Petrofac (manufacturer, energy-oil&gas)

6 Spectral Embedding - Directed Graphs

Let us generalize our construction to directed graphs. Degree matrix and adjacency matrix will be modified to naturally fit the generalized setting of directed graphs. The full justification of such an object is contained in the work of Chung [Chu05]. Her work ties together this definition with the isoperimetric inequalities one can obtain in the undirected setting, the same inequalities that intially sparked our interest in the algebraic study of graphs. The Cheeger inequality is once again established, thus reinforcing our confidence in the method abilities in terms of clustering. This is different to other methods that just consider the graph to be undirected, for instance by simply adding the weights $w_{ij} + w_{ji}$ (e.g trying to use the directed incidence matrix of the graph). One cannot obtain the isoperimetric inequalities formulated in the directed setting this way.

The main difficulty is that the usual Laplacian, defined as $L = D - W$, is not symmetric anymore. The equations crumble and the spectral properties related to clustering are lost. The trick is to consider the linkage graph through the lens of Markov chains. Consider P the transition matrix associated with the graph. Then the probability to go from node i to node j is

$$P_{ij} = \frac{w_{ij}}{\sum_j w_{ij}}.$$

We can also write $P = D^{-1}W$.

$\Phi =$ diag(π): Define the new degree matrix Write $x_0 \in \mathbb{R}^n$ any initial probability distribution on nodes. Then :

$$x_{n+1}^T = x_n^T P.$$

For the moment, suppose the walk converges. The final distribution on each node gives a centrality score π:

$$\lim_n x_n = \pi \qquad \pi^T = \pi^T P.$$

π is the relative importance of each company in the graph : companies that are mentioned a lot get a high score and thus their relative importance in the graph as a "degree" as we would have said in the undirected setting is high. Remark that π also satisfy the following equation, equivalent to $\pi^T = \pi^T P$:

$$\pi_i = \sum_j w_{ij} \pi_j.$$

The term "centrality" comes from this relation. Now lets define

$$\Phi = \text{diag}(\pi).$$

This is our new degree matrix. In the undirected setting ($w_{ij} = w_{ji}$) we fall back to $\Phi \propto D$.

(Pagerank) Slight modification of P We supposed the random walk as defined previously always converges, and that a unique stationary measure existed on that directed graph. Actually we must demand for our graph to be irreducible (there exist a path between each and every node). Markov Chain theory then guarantees the existence and uniqueness of such a measure.

A slight modification of P (Pagerank) is introduced, with a sound interpretation in the context of financial markets. From each node :

- With high probability α, jump to the nodes it is connected to (usually $\alpha \approx 0.99$).
- With low probability $1 - \alpha$, randomly jump to any other node, with uniform distribution.

It is a model of random influence that can exist between companies inside this linkage. This slight modification makes the graph irreducible and guarantees existence and uniqueness of π.

$\hat{A}$: define the new adjacency matrix Simply define

$$\hat{A} = \frac{\Phi W + W^T \Phi}{2}, \qquad (\hat{A})_{ij} = \frac{\pi_i w_{ij} + \pi_j w_{ji}}{2}.$$

We introduced a natural symmetrisation of the linkage graph. Each linked pair is weighted by both companies relative importance in the whole graph

$\hat{L}$: define the new Laplacian matrix Our new Laplacian now is :

$$\hat{L} = \Phi - \hat{A}.$$

Remark the quadratic form it defines is very similar in structure to our initial Laplacian :

$$x^T \hat{L} x = \sum_{ij} \pi_i w_{ij} (x_i - x_j)^2.$$

The embedding is once again obtained from the eigendecomposition of $\hat{L}$. See [CT07] for thorough details.

Part III
Connection to Commute Time and Metric Multidimensional Scaling

As stated in an introductory paragraph, this embedding method resembles PCA a lot. As it happens a connection really exists; essentially, we are defining similarities between some points and try to get the best representation of these relations in an Euclidian space of low dimension. This is a well studied area of mathematics and these connections are going to be made clear and precise.

A natural notion of distance on the graph will be constructed: commute time. Metric multidimensional scaling will be our way into embedding and PCA will appear in this context. The final embedding will be very similar to the traditional spectral embedding, and during this whole section we will draw enlightening parallels between both methods. This alternative approach is yet another explanation for how things work and why.

There are deeper connection to other areas of mathematics related to embeddings method, especialy by the means of Kernel PCA. We will not dive into it, and we refer to two interesting papers for the curious reader: [RHK21], [OLV14].

7 Commute Time Distance Embedding

Refer to [SFYD04] for full proofs of the results here. Once again, interpret the graph as a Markov chain, with transition matrix defined as

$$P = D^{-1}W.$$

The idea is to group similar companies together. This time, by similarity we simply mean that it is possible to travel fast between two nodes. Equivalently, we mean there needs small average time to get from node i to j.

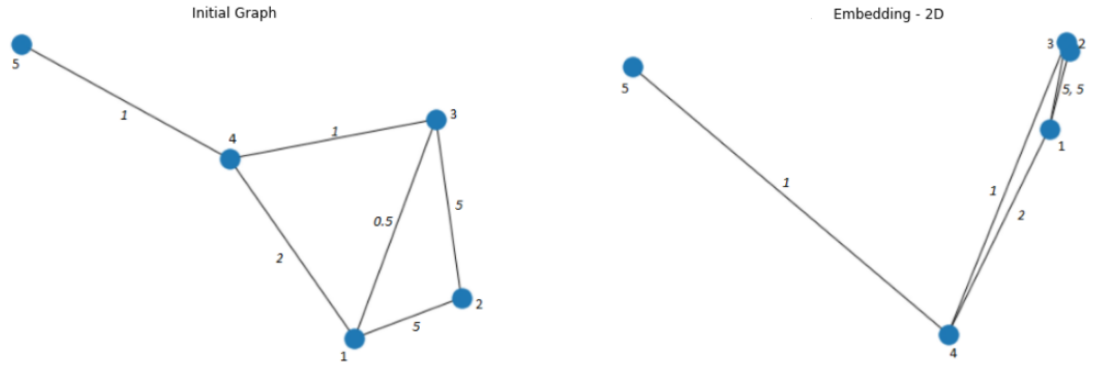


Figure 4: A graph and its desired embedding

While their linkage weight w_{13} is small, we would like 1 and 3 to be close together since they are tightly bound by the means of 2; there exists a fast path conecting 1 and 3.

Let us quantify this. Let T_{ij} be the time needed by a random walk to get from i to j. Then we define the average first passing time to be

$$m(i, j) = E(T_{ij}).$$

Remark that this is not always symmetric, even in the context of undirected graph:

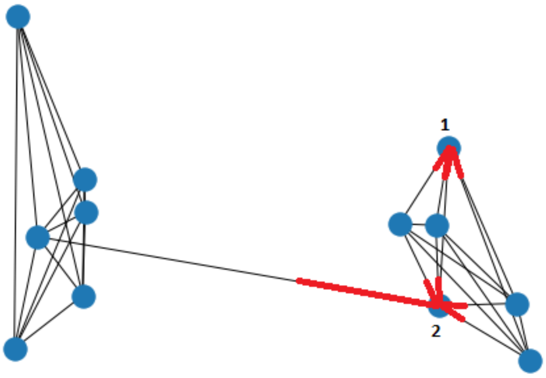


Figure 5: Assymetry of average first passing time

See how 2 is connected to the other cluster. This entails the inequality $m(2,1) > m(1,2)$. Thus we expand this to a more useful notion. Define the average commute time to be the average time needed to go from i to j and back again:

$$n(i,j) = m(i,j) + m(j,i).$$

Now this constitutes a natural symmetric distance between two companies. It can be additionnally proven that this verifies the triangle inequality, and so constitutes a metric distance. Looking at our first example, node 1 and 3 are now defined to be close together.

Connection to Laplacian Refer to [SFYD04] for full proof. The connection with the Laplacian is the following:

$$n(i,j) = l_{ii}^{+} + l_{jj}^{+} - 2l_{ij}^{+},$$

Where $L^{+} = (l_{ij}^{+})_{ij}$ is the pseudo inverse of L. This is not an obvious result. It essentially uses combinatorial properties of the graph. Thus, if we describe a company i by a point x_i in Euclidian space (we'll make it precise), we can see that this distance is derived from a scalar product defined between points as:

$$< x_i, x_j >= l_{ij}^{+}.$$

Once again it is clear how the Laplacian, rather than the adjacency matrix, holds the right information about our graph. This matrix of "similarity" between points does not merely reflect the local property of each node but their interaction with the whole graph. For instance one could have thought that the whole Laplacian embedding method was equivalent to naively set the similarity between points as, say, $< x_i, x_j >= \frac{1}{w_{ij}}$, and that is a question the energy interpretation cannot really tackle. Anyway, we are once again reassured.

8 Metric Multidimensional Scaling (MDS)

Now that distance has been defined between each companies, let us see how we get to the embedding. The theory behind MDS only requires distances to work; please refer to [Wil22] for a good explanation. Let us forget for a moment the connection with the Laplacian matrix as we want the MDS approach to be self-contained.

First, defining a metric distance (satisfying the triangle inequality, which is the case here) between n points $(x_i)_i$ is enough to uniquely determine their positions in Euclidan space, modulo rotations and translations. Additionally we require the points to be centered:

$$\|x_i - x_j\| = n(i,j), \quad \text{and} \sum_i x_i = 0,$$

which gives a unique description of position $X \in \mathcal{M}_{n,n}$ modulo rotations (line i represents embedding of company i). Let us prove this in detail.

Proof. Say a $\mathbb{R}^n$ centered embedding of the companies exists, where x_i is the position in space of company i. Let $X \in \mathcal{M}_{n,n}$ be the matrix of positions : line i represents embedding of company i. Write $K = XX^T = (< x_i, x_j >)_{ij}$ the Gram matrix of scalar products between each points. Then we must have:

$$n(i,j)^2 = \|x_i - x_j\|^2 = K_{ii} - 2K_{ij} + K_{jj}.$$

With $\Delta = (n(i,j)^2)_{ij}$ and $N = (K_{ii})_i = (\|x_i\|^2)_i$ the vector of norms;

$$\Delta = N\mathbf{1}^T - 2K + \mathbf{1}N^T.$$

But our data is centered. If we write $P_1 = I_n - \frac{\mathbf{1}\mathbf{1}^T}{\|\mathbf{1}\mathbf{1}^T\|^2}$ the projection on the orthogonal space of $\mathbf{1}$, we must have:

$$P_1\mathbf{1} = \mathbf{1}^T P_1 = 0,$$

$$P_1 \Delta P_1 = -2P_1 K P_1.$$

But $K\mathbf{1} = \mathbf{1}^T K = 0$ (data is centered), so $P_1 K = KP_1 = K$ and

$$K = -\frac{1}{2} P_1 \Delta P_1.$$

For the reciprocal direction, set $K = -\frac{1}{2}P_1\Delta P_1$. This gives a symmetric matrix which we can verify is double centered and positive semi-definite (this is because the distance verifies the triangle inequality). We write :

$$K = U\Gamma U^T = XX^T,$$

where

$$X = U\Gamma^{1/2}$$

describes the centered $\mathbb{R}^n$ embedding of each companies. We can see here how any rotation $V \in \mathcal{O}_n$ on $X := XV^T$ does not change the result. The Euclidian distance between each company i, j is indeed the commute time. □

Once again remember that $K = L^+$ (as L^+ also verifies $\mathbf{1}^T L^+ = L^+\mathbf{1} = 0$).

9 PCA

We set our embedding to be $X = U\Gamma^{1/2}$ such that:

$$L^+ = XX^T = U\Gamma U^T.$$

So the covariance matrix of the data is

$$C = X^T X = \Gamma = \begin{pmatrix} \gamma_1 & & \\ & \cdots & \\ & & \gamma_n \end{pmatrix}, \quad \text{with} \;\; \gamma_1 \geq \cdots \geq \gamma_n = 0.$$

It is diagonal; the embedding we obtained is already in its 'PCA optimized' format, and we can keep for each company positions the k first components. Now the relationship with traditional spectral embedding is clear. L and its pseudo inverse L^+ are closely related. Mainly:

$$L = U\Lambda U^T, \quad L^+ = U\Gamma U^T,$$
$$\Gamma \equiv \Lambda^{-1}.$$

In the sense that :

$$\Gamma = \text{diag}(\gamma_1, \cdots, \gamma_n), \quad \Lambda = \text{diag}(\lambda_1, \cdots, \lambda_n) \quad \text{and} \qquad \gamma_i = \begin{cases} \dfrac{1}{\lambda_i} & \text{if } \lambda_i \neq 0 \\ 0 & \text{otherwise} \end{cases}.$$

The eigenvectors described by U are the same.

So in the end we have the same embedding up to some scaling factor. In our project, we are using the traditional Laplacian embedding, that is setting $X = U$ rather than $X = U\Gamma^{1/2}$. This is the empiricially tested and robust way of looking at graphs. From this section however we understand why it makes sense to look at such an object and such a construction.

Part IV
Analysis and Usage of the Spectral Embedding

At this stage we would like to understand how to effectively make use of the spectral embedding. This mainly leads to ask the following questions: how many dimensions should be retained for the final embedding? Which type of Laplacian should I use for the embedding? How can I estimate the quality of the embedding? How can I compare two graphs?

10 Eigengaps and Spectral Clustering

As it happens, the spectrum of the graph, that is the eigenvalues of the Lapacian in increasing order, holds a lot of useful information. Some study has been done in this area, and knowing to what extent the spectrum (or eigendecomposition of L) is unique to a graph or a certain class of graph is a hard and broad question. In our context, there may be two interesting use cases. First would be to use the spectrum to compare two graphs. This eventually boils down to a signal analysis problem. Second is the study of *eigengaps*. An eigengap is a sudden jump in the increasing eigenvalues; we will see it is a good heuristic to determine the number k of dimensions to retain.

10.1 Examples and Graph spectrum

Let us take a look at some examples. These will build our intuition. In these examples, we are going to display

$$\gamma = \frac{1}{\lambda}$$

as the spectrum of the graph (looking at eigenvalues of L^+ in decreasing order). This corresponds to the commute time approach. It is easier to visually see eigengaps in this fashion. All the following remarks still stand when looking at the λ in increasing order.

10.1.1 Clustering

Imagine two practically separated clusters:

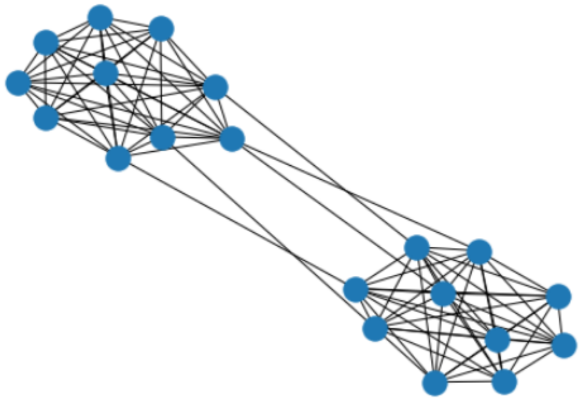


Figure 6: 2-cluster graph, all links of weight approx. equal to 1

Compute its 2D embedding:

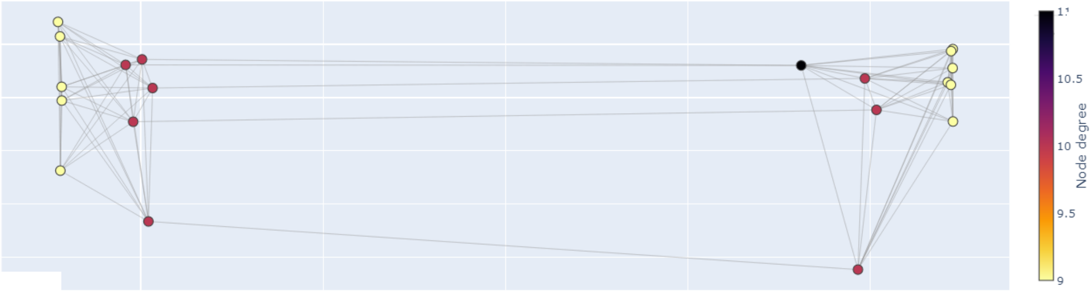


Figure 7: 2D embedding of 2-cluster graph, all links of weight approx. equal to 1

The method will compactly group both clusters around two distant points in the first dimension. However, we see that the second dimension separates nodes in a much less interesting way. From this point on, the exact replication of the distances between points is happening at a slow rate, and we have lost dissociative power. Let's see the spectrum of the graph (eigenvalues $(\gamma_i)_i$ in decreasing order):

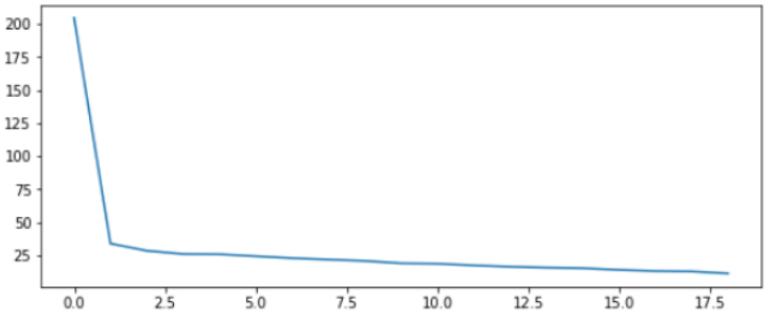


Figure 8: Spectrum of 2-cluster graph

γ_1 value is high, but $\gamma_2, \gamma_3, \cdots$ values are much lower. The eigengap at $i = 2$ indicates a clear 2-cluster structure. This phenomena naturally generalizes to k clusters. Lets take a quick look for $k = 4$:

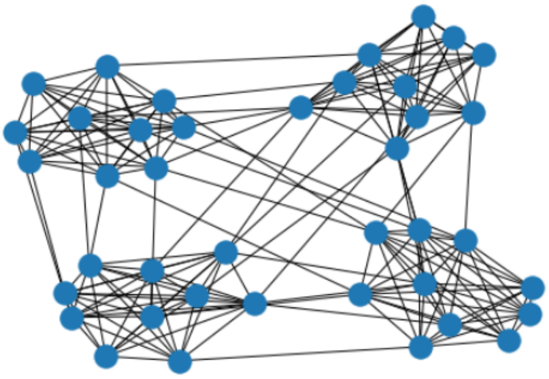


Figure 9: k-cluster graph

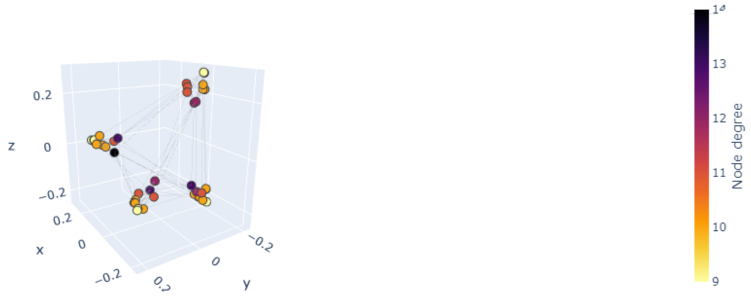


Figure 10: 3D embedding of k-cluster graph

Remark visually how for 4 clusters, three dimensions are enough to represent and separate them. The spectrum is as much revealing

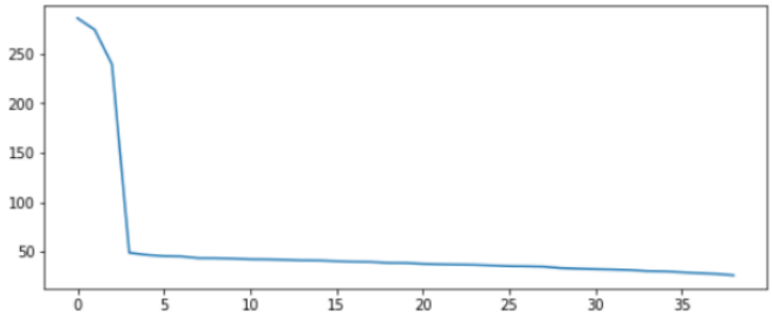


Figure 11: Spectrum of k-cluster graph

There is a sharp drop after the third eigenvalue γ_3.

10.1.2 Different types of network

When there is a clear custered center, the method is quite straght to the point. However, the linkages at hand do not always display such a nice structure. If cluster presence can be detected with a sharp drop/increase (either looking at γ/λ), something closer to a large scale network will show an opposite behaviour; a slow increase in the spectrum. We call this behaviour 'hub-ness' effect. Usually what happens is that a few companies aggregate the majority of links (the typical large scale network is such that the distribution of degrees follows a power law). The unnormalized Laplacian is the most sensitive to this effect. Without any penalization for low degree nodes, it can tend to isolate unique points out of the dense mesh of companies.

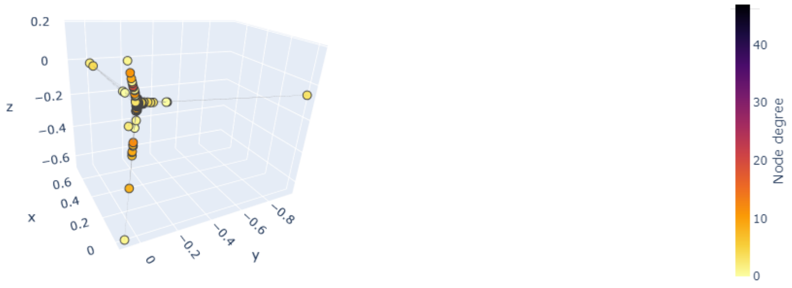


Figure 12: Embedding of a proprietary linkage using the unnormalized Laplacian

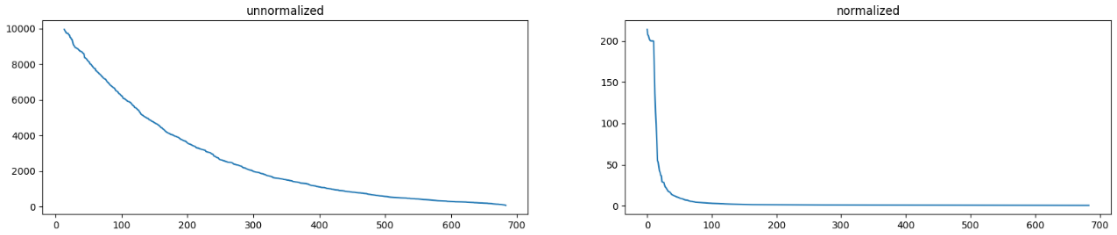


Figure 13: Spectrum of this linkage (unnormalized/normalized Laplacian)

The spectra displayed here show however how the normalized Laplacian can be a good parade against this behaviour. By acting differently on low/high degree nodes, it is able to separate companies effectively once again. See [Lux07]; an explanation is that the asymptotically unnormalized Laplacian can act as a Dirac function. These results demand to build much more complex tools, so we do not dive into it in this report. Again, for interesting insights look at [OLV14] and [RHK21], which connect the whole technique to Kernel PCA before looking at asymptotical operators.

10.2 Energy interpretation

Remember our primary equation (unnormalized Laplacian):

$$2x^T Lx = \sum_{ij} w_{ij}(x_i - x_j)^2.$$

For each eigenstate:

$$u_i^T Lu_i = \lambda_i.$$

An eigengap is then an energy jump. The associated eigenvector loses the ability to effectively aggregate connected parts of the graph, or equivalently to separate loosely connected ones.

The first eigengap can also indicate when all clusters have been indentified. Remember in the perfect scenario of k disconnected components the k first eigenvalues are 0 (0 energy). Then the eigenstate operate a dissociation in intra cluster points and we have an energy jump.

Additionally, we can also interpret an energy plateau like degenerescence; there are many dimensions to be filled by eigenvectors at the same energy level. Moreover we observe that when eigenvalues increase slowly, linkage has more of a hub like structure (like a large scale network).

10.3 Perturbation Theory

We are considering an embedding in k dimensions. Denote by L any Laplacian type of some graph (L_{sym}, L_{rw}, L), and call V_k the subspace spanned by its k first eigenvectors. Then for a small perturbation $\overline{L}$ of that Laplacian (adding, subtracting, changing edge weights) and the corresponding $\overline{V}_k$ we have:

$$d(V_k, \overline{V}_k) \leq \frac{\|L - \overline{L}\|}{|\lambda_{k+1} - \lambda_k|},$$

where d is a distance between those subspaces (see [Lux07] for more details). Say we have k perfect clusters in our graph. Then the eigenvectors are perfect indicators of the clusters, and we have a high eigengap (sudden energy jump), so that the embedding given by a small perturbation of that configuration will still be effective at locating the clusters.

Eventhough only the contraposition is true, it is somewhat reasonable to assume the reciprocal: the eigengap $|\lambda_{k+1} - \lambda_k|$ is a good measure of the effectiveness with which the embedding is able to "locate" some clusters if and when they exist, and the optimal number of clusters to uncover/dimensions to use is exactly k. Large eigengap means high stability of embedding in terms of perturbation. Remark that this is particularly the case for the unnormalized Laplacian L and the normalized random walk L_{rw}. Indeed, for the symmetric normalized Laplacian, consider a few distinct connected components. The eigenvectors of L_{sym} are

$$D^{1/2}\mathbf{1}_{A_i}.$$

This can introduce a lot of heterogeneity in the position values when the degrees are very different. Any small perturbation is liable to scramble the cluster information out, and the normalization step found in the algorithm of Ng, Jordan and Weiss does not correct this problem.

10.4 Commute Time

Remember we were looking at $\Gamma = \Lambda^{-1}$ during the commute time approach. Let us see how we get to the relevancy of the eigengap through this method. We had as a covariance matrix :

$$tr(C) = tr(X^T X) = tr(\Gamma) = \sum_{i=1}^{n} \gamma_i = \text{total variance.}$$

But remark that

$$n(i,j)^2 = \|x_i - x_j\|^2 = (e_i - e_j)^T X X^T (e_i - e_j) = (e_i - e_j)^T L^+ (e_i - e_j),$$

$$n(i,j)^2 = L^+_{ii} + L^+_{jj} - 2L^+_{ij}.$$

So when we are looking at the distance induced by the k dimensional embedding we have

$$\hat{n}(i,j)^2 = \|x_i^k - x_j^k\|^2 = (e_i - e_j)^T X_k X_k^T (e_i - e_j) = (e_i - e_j)^T \hat{L} (e_i - e_j),$$

$$\hat{n}(i,j)^2 = \hat{L}^+_{ii} + \hat{L}^+_{jj} - 2\hat{L}^+_{ij}.$$

Where we can write $L^+ = U\Gamma_k U^T$ where Γ_k is zero on its diagonal from the $k+1$-th component on. Then [SFYD04] :

$$\|n - \hat{n}\|^2 \leq \sum_{i=k+1}^{n} \gamma_i.$$

The key to spectral embedding is that the first few dimensions not only introduce the best explanation of variance, but also an excellent distortion of the graph in terms of distance approximation. We are looking at the best description of overall information flow, which gives best clusters in low dimensional representation. Sharp decrease in $\gamma = \frac{1}{\lambda}$ values indicate that we get excellent approximation in low dimensions. Once again we can say that eigenvalues describe how well "information" is separated in the network.

11 Choosing Normalized or Unnormalized Laplacian

There are two Laplacian matrices we can work with:

- Unnormalized : $L = D - W$. We have been working with this one from the beginning. Problem : when degrees are very heterogeneous, first eigenvectors tend to act as Dirac functions.

- Normalized : $\hat{L} = D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2}$. Remember the associated quadratic form:

  $$x^T \hat{L} x = \sum_{ij} w_{ij} \left(\frac{x_i}{\sqrt{d_i}} - \frac{x_j}{\sqrt{d_j}} \right)^2 .$$

  We penalize low degrees vertex from getting too far from other points. This should give a more balanced result. Moreover this operator is stable as the number of points grow larger (see [Lux07], [OLV14] and [RHK21]).

Another argument is the following. The indicator vectors when there is perfectly k clusters are $(\mathbf{1}_{A_i})$ instead of $(D^{1/2}\mathbf{1}_{A_i})$ for the unnormalized Laplacian. The perturbation approach tend to favor the unnormalized Laplacian which is more robust to noise.

Let us present a final argument from network theory. We defined

$$W(A, B) = \sum_{i \in A, j \in B} w_{ij}.$$

One could work on the following optimizations when embedding the graph:

1. Minimizing the between cluster similarity, that is dissociating points from different clusters.

   $$\text{Minimize } W(A, \overline{A}).$$

2. Maximizing within cluster similarity, that is embedding highly interconnected points close to each other:

   $$\text{Maximize } W(A, A) \text{ and } W(\overline{A}, \overline{A}).$$

It can be proven ([Lux07]) that the unnormalized Laplacian optimizes (1), while the normalized Laplacian optimizes simultaneously (1) and (2).

When trying to decide which Laplacian to use, one could look at their respective spectrum and decide which embedding seems the more satisfying (for instance by favoring steepest eigengap). Empirical results tend to favour the unnormalized version of the Laplacian when working with our linkages.

Part V
Entropy

We would like to find a way to distinguish different types of graph. As we have seen, the graph spectrum can be used in such a way. Here we will see an alternative method based on the study of the distribution of the node degrees. Studying degrees directly is somewhat inconclusive, and instead we will focus on a more natural way of looking at heterogeneity of degrees with entropy. We will first study characterizations of entropy in the network, then try to extend that to the directed graph setting, and finally determine a working technique.

12 Estrada Heteroegeneity

There are a few ways to compute unique quantities relating to entropy/heteroegeneity on the graph. As a first reasonable measure, Estrada [Est10] proposed the following:

$$\rho(G) = \sum_{u,v \in E} \left(\frac{1}{\sqrt{d_u}} - \frac{1}{\sqrt{d_v}}\right)^2.$$

Remark it can be rewritten with the normalized Laplacian:

$$\rho(G) = \mathbf{1}^T L_{sym} \mathbf{1}.$$

Writing $N = |V|$, we can bound this vaue between 0 and 1:

$$\overline{\rho}(G) = \frac{\rho(G)}{N - 2\sqrt{N-1}}.$$

The details are contained in [Est10]. A value of 0 is a regular graph (each node has same degree), a value of 1 is a star graph, the most unbalanced graph. It can be used to distinguish many types of graphs (small worlds, random graphs, scale free networks). However it does not generalize well to directed graph. We could write

$$\rho_{dir}(G) = \mathbf{1}^T \hat{L}_{sym} \mathbf{1},$$

where $\hat{L}_{sym} = \Phi^{-1/2} \hat{L} \Phi^{-1/2}$.

This one dimensional value is convenient for computations and it indeeds separates the different graph structures at hand. It does not really describe any changes of regime (for instance we could have seen a drop in heteroegeneity around march 2020; crisis, like covid, usually make the graph structure more organized and less diverse. This is not the case in our graphs). However, studying the equation and examining the terms in the sum $\mathbf{1}^T \hat{L}_{sym} \mathbf{1}$ does not provide the insights we get from the next method.

13 von Neumann Entropy

The key is to view the normalized Laplacian as a density matrix for the graph Hamiltonian ρ now defined as

$$\rho = \frac{L_{sym}}{N},$$

with $N = |V|$, so that $tr(\rho) = 1$. Then the graph entropy is the von Neumann entropy:

$$H = -tr(\rho \ln \rho),$$

$$H = -\sum_{i=1}^{N} \frac{\lambda_i^{sym}}{N} \ln \frac{\lambda_i^{sym}}{N}.$$

For which an approximation is

$$H = \sum_{i=1}^{N} \frac{\lambda_i^{sym}}{N}(1 - \frac{\lambda_i^{sym}}{N}),$$

$$H = \frac{1}{N} tr(L_{sym}) - \frac{1}{N^2} tr(L_{sym}^2).$$

Now from this formulation it is possible to get to a development for directed graphs. Simply use Chung formulation of the directed Laplacian that we normalize too:

$$\tilde{L} = \Phi^{-1/2} \hat{L} \Phi^{-1/2}.$$

See [Han16] for full details. After some computation we get:

$$H = 1 - \frac{1}{N} - \frac{1}{2N^2} \left(\sum_{u,v \in E} \frac{d_u^{in}}{d_u^{out}} \frac{1}{d_u^{out} d_v^{in}} - \sum_{u,v \in E_2} \frac{1}{d_u^{out} d_v^{out}} \right),$$

where a directed edge (u, v) connects nodes of ingoing/outgoing degrees $(d_u^{in}, d_u^{out}) \mapsto (d_v^{in}, d_v^{out})$, and edges are partitioned into $E = E_1 \sqcup E_2$ with E_1 unidirectional edges and E_2 bidirectional edges.

14 Index Histogram

Now the original idea is use the previous subsection to build a nice senseful histogram. To the best of my knowledge this is an idea from [Han16]. We are going to classify edges in bins defined by their nodes degree. Attribute to each edge a 4d vector:

$$e = (u, v) \mapsto (d_u^{in}, d_u^{out}, d_v^{in}, d_v^{out}),$$

and compute the entropy contribution of each of these bins. The normalized local entropic measure for each unidirectional edge is:

$$I_{uv} = \frac{d_u^{in}}{2|E||V| d_v^{in} (d_u^{out})^2},$$

to which we add the contribution of bidirectional edges:

$$I'_{uv} = \frac{1}{2|E_2||V| d_v^{out} d_u^{out}}.$$

Finally, since the 4D histogram becomes very sparse for big graphs, we are going to rank the degrees into m labels and attribute to the graph a 4D tensor M. Formally:

- For each node type u or v, compute the cumulative distribution function of the in/out degrees
$$F_u^{in}(x) = \sum_{0 \leq i \leq x} P(d_u^{in} = i).$$
We will call these functions F_{node}^{dir} where *node* is the node type and *dir* the in/out degree.
- Assign a label for each of the 4 components associated to an edge
$$q_{node}^{dir} = \min\{j \;/\; F_{node}^{dir}(d_{node}^{dir}) < \frac{j}{m}\}.$$
- Compute
$$M_{ijkl} = \frac{1}{2|E||V|} \sum_{(q_u^{din}, q_u^{dout}, q_v^{din}, q_v^{dout})=(i,j,k,l),(u,v)\in E} \left(I_{uv} + I'_{uv}\right).$$
- Remark it is sometimes possible to work on a 3D representation. If the graph is strongly directed (edges mostly in E_1), then describe each edge with
$$e = (u, v) \mapsto (d_u^{in}, d_u^{out}, d_v^{in}).$$
and compute
$$M_{ijkl} = \frac{1}{2|E||V|} \sum_{(q_u^{din}, q_u^{dout}, q_v^{din}, q_v^{dout})=(i,j,k,l),(u,v)\in E} \left(I_{uv}\right).$$

Concatenate the elements of M to obtain a vector of dimension m^4 (m^3 in the case of strongly directed graphs). This lets us compare the linkage graphs, for different linkage types and over some period of time. Perform for instance perform a 3D PCA with all the linkages considered month to month, over the last two years :

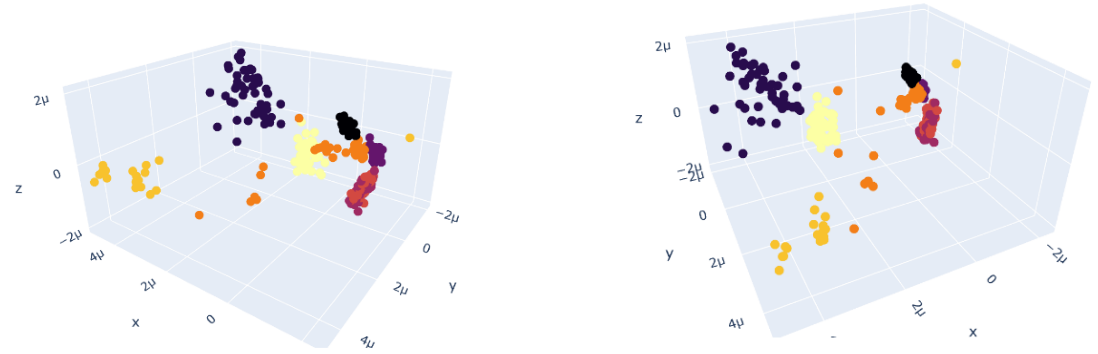


Figure 14: Index Histogram each month, 2020/2022. Each linkage type has its own color

As we can see, this method separates well the different linkage types (any clustering algorithm would classify them correctly). We are also able to see changes in regime over time for the same linkage type, when one of its vector gets located far from its usual position in space.

This is also reassuring in the context of embedding. Even if we work statically (embedding time t to time t without consideration for the previous states of the graph), we expect to keep some structure, stability and continuity in the results we obtain.

References

[Bon19] T. Bonald, *Spectral graph embedding.*

[BRR94] Burmeister, Roll, and Ross, *A practitioner's guide to arbitrage pricing theory*, The Research Foundation of The Institute of Chartered Financial Analysts (1994), 1–29.

[Chu97] Chung, *Laplacians of graphs and cheeger inequalities.*

[Chu05] ______, *Laplacians and the cheeger inequality for directed graphs.*

[CT07] Mo. Chen and Q. Tang, *Directed graph embeddings*, IJCAI (2007).

[Est10] E. Estrada, *Quantifying network heterogeneity.*

[GK94] Grinold and Kahn, *Multiple-factor models for portfolio risk*, The Research Foundation of The Institute of Chartered Financial Analysts (1994), 59–79.

[Han16] Edwin Hancock, *Network entropy*, 35–45.

[Lux07] U. Luxburg, *A tutorial on spectral clustering.*

[MSC22] MSCI, *The global industry classification standard (gics)*, 2022.

[NJW01] Andrew Ng, Michael Jordan, and Yair Weiss, *On spectral clustering: Analysis and an algorithm*, Advances in Neural Information Processing Systems (T. Dietterich, S. Becker, and Z. Ghahramani, eds.), vol. 14, MIT Press, 2001.

[OLV14] D. Orodnnez, J. Lee, and M. Verleysen, *Generalized kernel framework for unsupervised spectral methods of dimensionality reduction*, SSCI 2014 (2014).

[RHK21] J. Ryu, J. Huang, and Y. Kim, *On the role of eigendecomposition in kernel embedding.*

[SFYD04] M. Saerens, F. Fouss, L. Yen, and P. Dupont, *The principal components analysis of a graph, and its relationships to spectral clustering*, Machine Learning: ECML 2004 (2004).

[SM00] Jianbo Shi and Jitendra Malik, *Normalized cuts and image segmentation*, IEEE Trans. Pattern Anal. Mach. Intell. **22** (2000), no. 8, 888–905.

[TLM08] M. Tuminello, F. Lillo, and R. Mantegna, *Correlation, hierarchies, and networks in financial markets*, The Research Foundation of The Institute of Chartered Financial Analysts (2008), 1–29.

[Wil22] R. Wilkinson, *Multidimensional scaling (mds).*