Present_Polytechnique

August 25, 2022

1 Introduction

1.1 Enhancing Linkages - Spectral Analysis of Graph

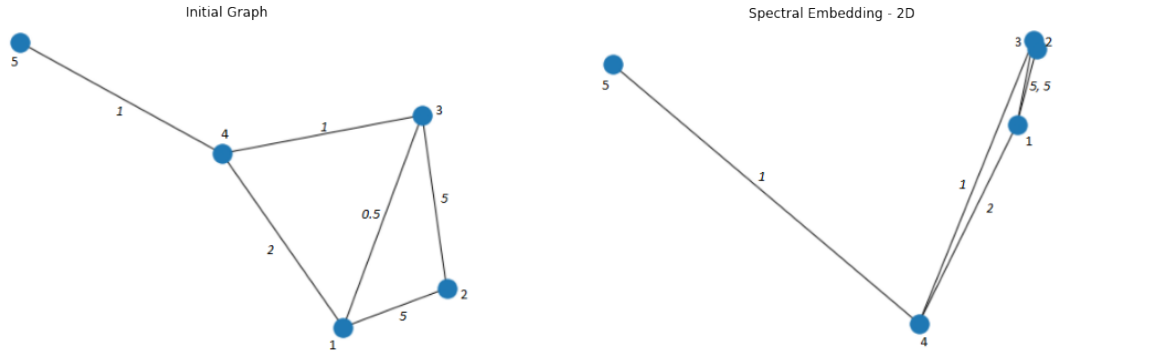
- Study graphs in the eyes of linear algebra (e.g adjacency matrix, outgoing degrees matrix, Laplacian...) to try to get results and insights on the inner structure of such objects.
- For instance, we can get estimations of the connectedness of a graph (Fiedler number, Cheeger's inequality, centrality measures...), or embeddings in the $\mathbb{R}^n$ space

The theory has many applications, but we will primarily focus on the **geometric study of embeddings**.

- Insights in the structure of the linkages, with meaning in the context of financial data.
- Ability to identify clustering of companies, high interconnectedness or "hub"-ness effects

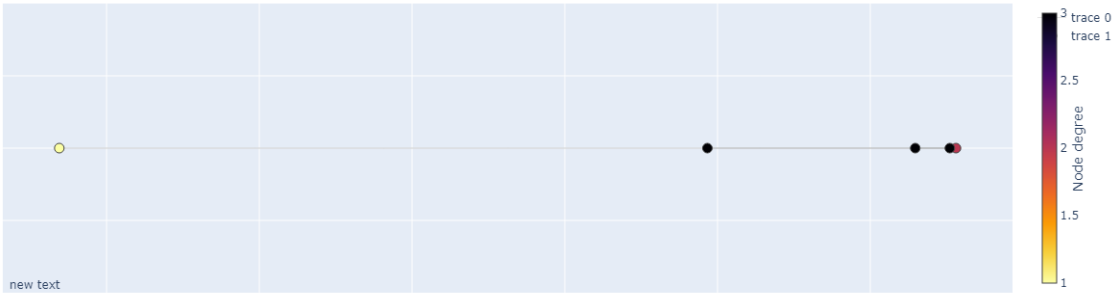
1.2 Quick example

[25]:



1D Embedding

[152]:
```
Scatter1dGraph(G)
```



- Initial graph : arbitrary positions for each node
- Spectral embedding : "closest" points are clustered together (see nodes 1 and 3)

2 Spectral Embedding - Undirected Graph

Describe linkage at date t by its graph $G = (V, E)$: * V is the set of vertices, or nodes. They will represent the companies * E is the set of edges : $E \subset V \times V$. Edges represent links.

We will be writing : * W the adjacency matrix of the graph : $(W)_{ij} = w_{ij}$ the weight of the edge between i and j * $D = \text{diag}(d_i)$ the diagonal matrix of degrees : $d_i = \sum_j w_{ij}$

2.1 Information commute time

In the end, we will relate this information commute time approach to traditional spectral embedding (using the graph laplacian). The former however has a much more natural interpretation.

- Interpret graph as transition matrix between companies :
  - Probability to jump from company i to company j is $\propto w_{ij}$
  - **Objective : group similar companies**
  - Similar companies = not too far apart in the graph of transition
  - **Similar companies = fast information flow/small time needed to get from company i to company j**

Let us quantify this :

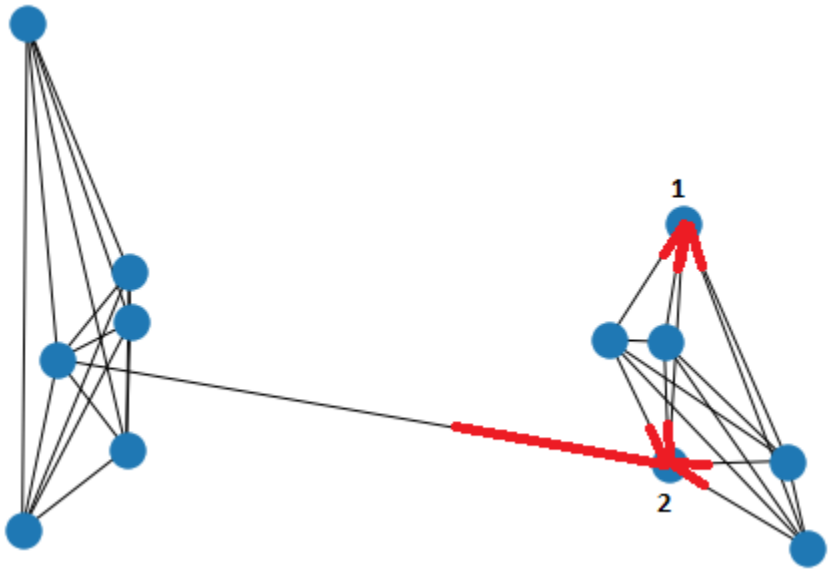
- **Average information first passing time** : let T_{ij} = time needed by random walk to get from i to j. Average first passing time then is

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

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

[7]:
```
from IPython import display
display.Image("./graph_not_sym.png", width=450)
```

[7]:



- Remark how 2 is connected to the other cluster : we have $m(2,1) > m(1,2)$

We expand this to a more useful notion

- **Average information commute time** : going from i to j and back again

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

  Natural symmetric distance between two companies

2.2 Embedding

By "embedding", we mean that we assign to each company i a n-dimensional vector $x_i \in \mathbb{R}^n$

2.2.1 Embedding in n-dimensions

Defining a distance between n points $(x_i)_i$ is enough to uniquely determine their positions in space, modulo rotations and translations.

(Metric multidimensional scaling theorem) Requiring that :

$$\|x_i - x_j\| = n(i,j)$$

$$\sum_i x_i = 0 \text{ (centering)}$$

Enables us to construct $X = \left(x_1^T \cdots x_n^T \in \mathcal{M}_{n,n}\right.$ a matrix of positions of companies. We required the positions to be centered in view of PCA.

- Outline of the construction :

  We are trying to find X **the matrix of position**.

  It is possible to deduce **the matrix K of scalar product** between our would-be embedded points from the information commute time distance n :

  $$K = (< x_i, x_j >)_{ij} = \text{some function of commute time n}$$

  So we see that in the end we must have :

  $$K = XX^T$$

  However K is symmetric, and thus admits an eigendecomposition :

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

  And we simply set

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

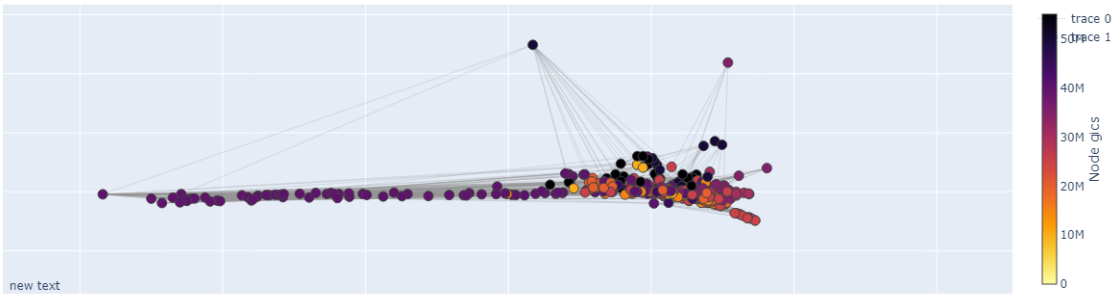
  We will see that this relates closely to traditional spectral embedding.

2.2.2 2D Example - eunation - correl_intraday

- Displaying first 2 columns of X ; each company i is described by (x_{i1}, x_{i2}) :

  $$X_{.2} = (x_{11}x_{12}x_{21}x_{22} \cdots \cdots x_{n1}x_{n2}$$

[22]:
```
Scatter2dGraph(G, "gics")
```



This plot just underlines the fact that we have obtained an embedding in euclidian space. No guarantee yet on distance approximation, variance or anything.

2.2.3 Projection in k-dimensions

- With PCA, we minimize distance between points and their linear projection in a lower dimensional space:

  $$\min \|X - \hat{X}\|^2$$

  This approximately preserves the original distance (commute time) :

  $$n(i,j) \approx \hat{n}(i,j)$$

- Remember we could write

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

  So the covariance matrix of the data is just :

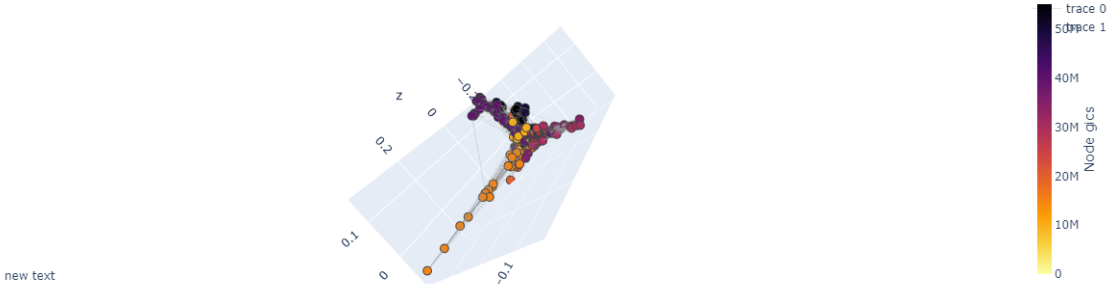
  $$C = X^T X = \Gamma$$

  The data is already available projected along its principal components !

- A k-dimensional projection thus simply consists in keeping the k first columns of X

2.2.4 3D Example - eunation - correl_intraday - mid 2011

```
[49]: Scatter3dGraph(G, "gics")
```



- Small distance between :
  - 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-alernate), Petrofac (manufacturer, energy-oil&gas)

We are going further than just grouping by gics.

2.3 Clustering Power

2.3.1 Eigengap interpretation

Remember the covariance matrix of the data :

$$C = X^T X = \Gamma = (\gamma_1 \cdots \gamma_n$$

$$\text{with} \quad \gamma_1 \geq \cdots \geq \gamma_n = 0$$

- We don't actually want to keep the initial distances, otherwise we just keep initial embedding. Remember :

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

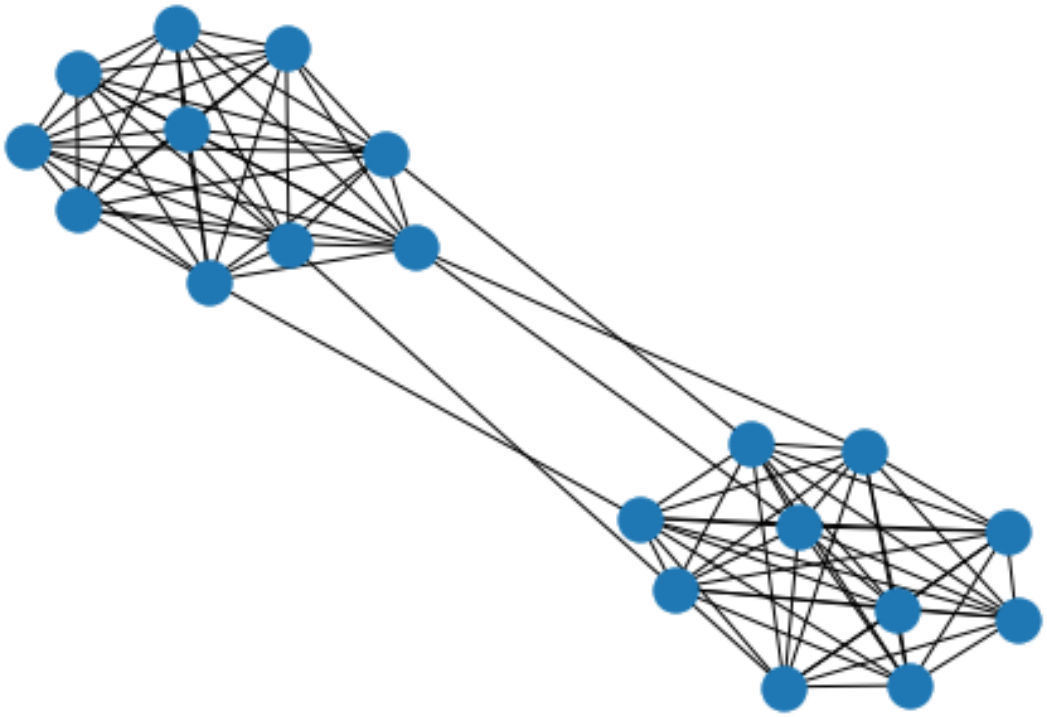
  Then for the dim k PCA embedding :

$$\|n(i,j) - \hat{n}(i,j)\|^2 \leq \sum_{i=k+1}^{n} \gamma_i$$

- 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 γ values** indicate that we get excellent approximation in low dimensions.
- We could also say that eigenvalues describe how well **"information" is separated in the network.**

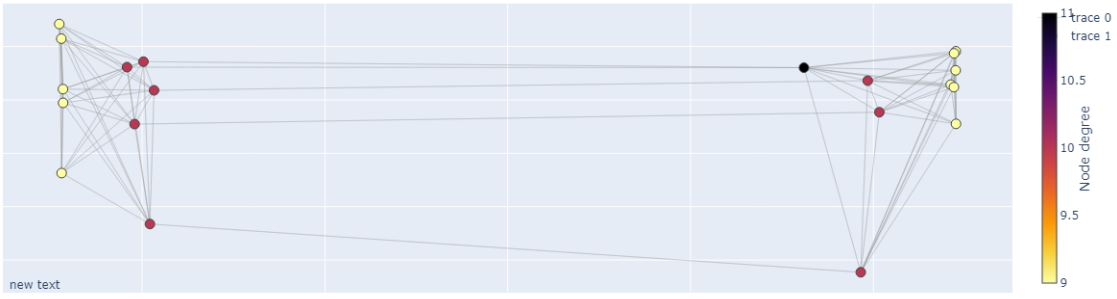
Imagine two practically **separated clusters** :

[17]:
```
G = nx.from_numpy_matrix(K)
nx.draw_spring(G)
```

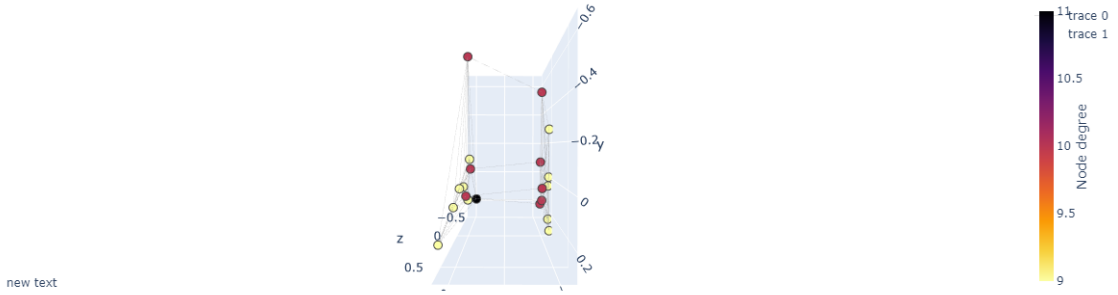


- The method will compactly group both clusters around two distant points in the first dimension
- However, we see that the second dimension separates nodes in a much less interesting way.
- From this point on, the exact replication of the distances between points is happening at a slow rate, and we have lost dissociative power.

[23]:
```
Scatter2dGraph(G_dir)
```



[22]:
```
Scatter3dGraph(G_dir)
```

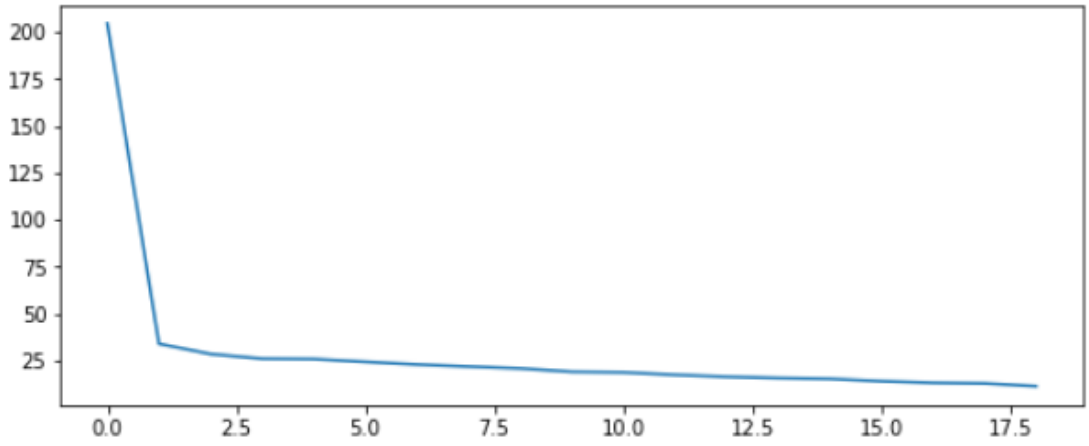


- γ_1 value is high, but $\gamma_2, \gamma_3, \cdots$ values are much lower. The eigengap at $i = 2$ indicates a clear 2-cluster structure.

  Let's see the spectrum of the graph (eigenvalues $(\gamma_i)_i$ in decreasing order) :

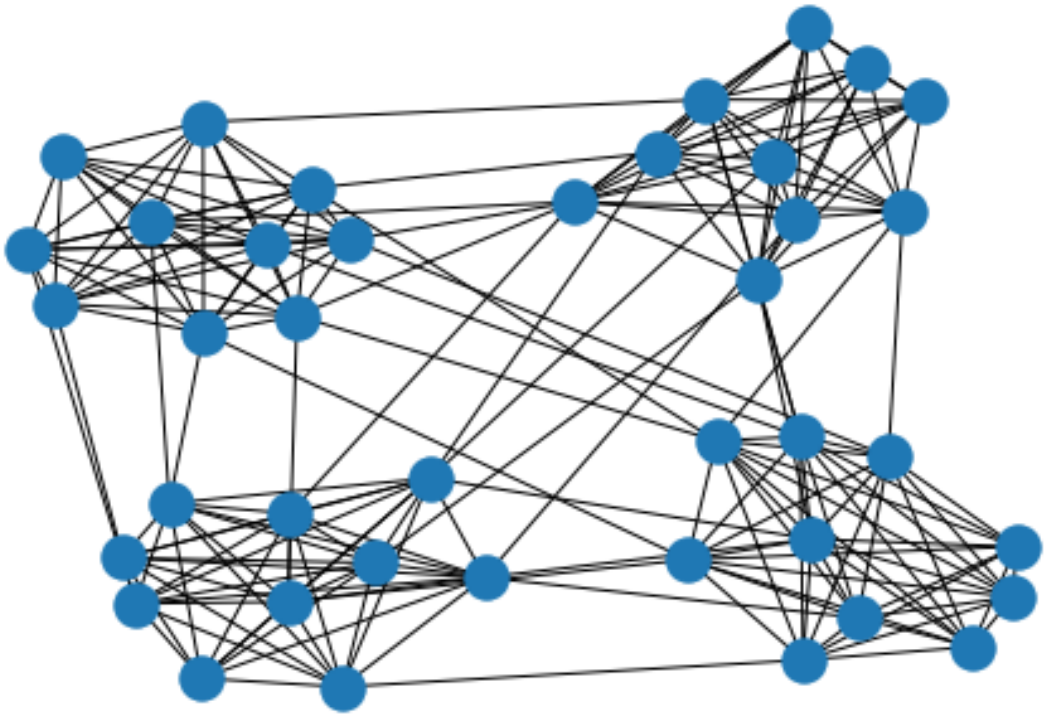
[7]:
```
from IPython import display
display.Image("./spectrum.png", width=600)
```

[7]:

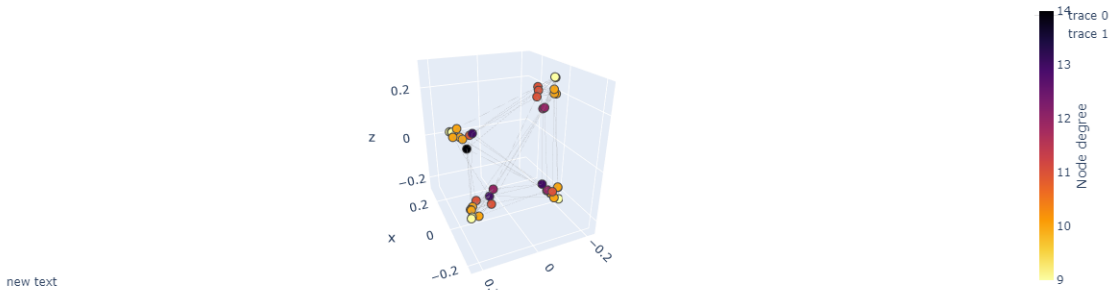


- Result generalizes to k clusters. Here with $k = 4$:

[6]:
```
G = nx.from_numpy_matrix(K)
nx.draw_spring(G)
```

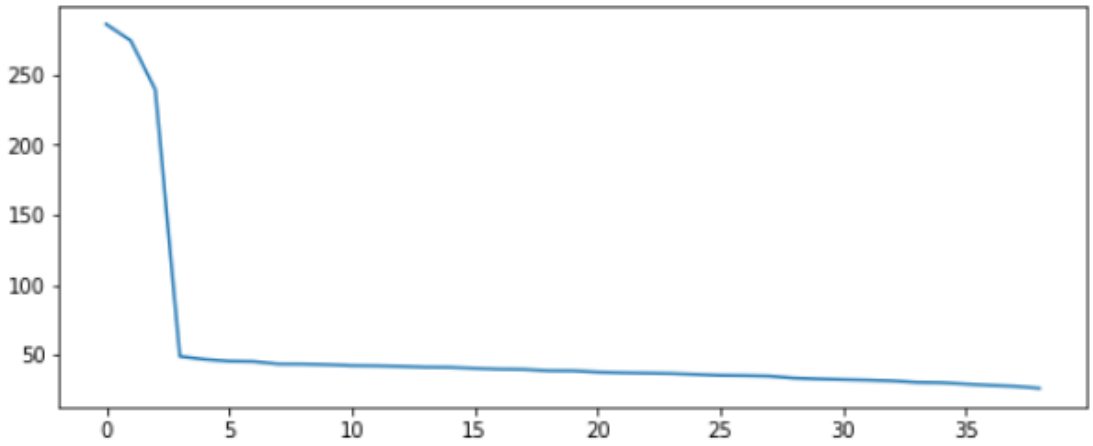


[93]:
```
Scatter3dGraph(G_dir)
```



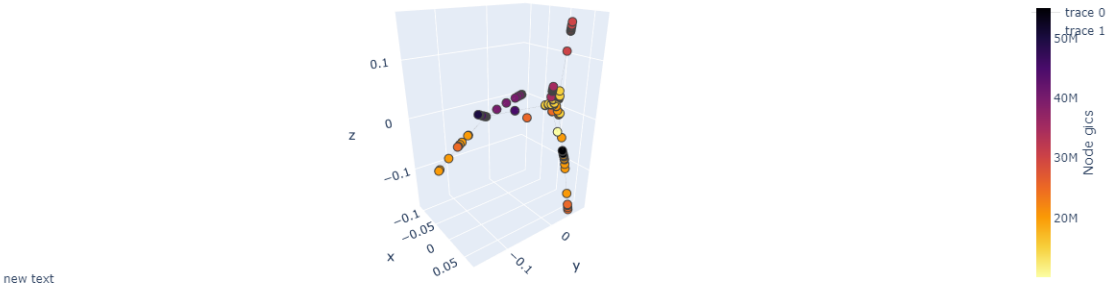
[8]:
```
from IPython import display
display.Image("./spectrum_k.png", width=600)
```

[8]:



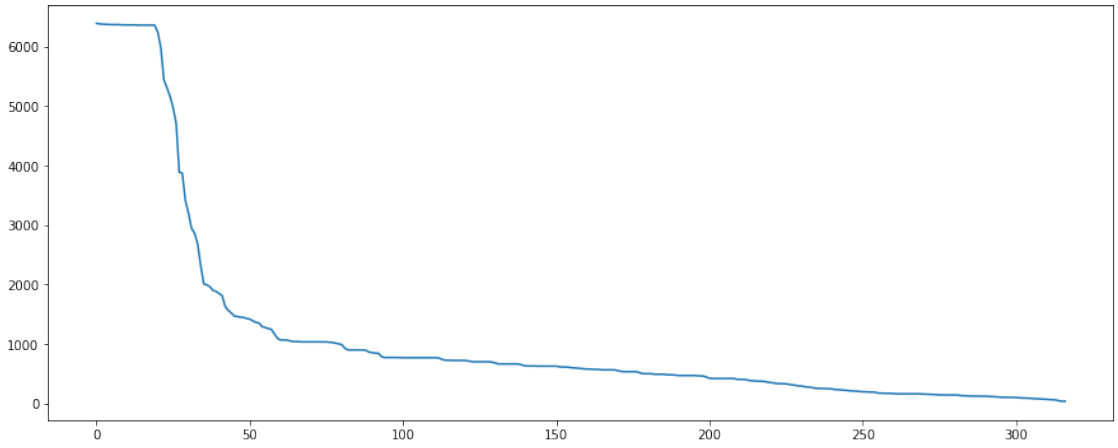
Linkage example

[131]:
```
Scatter3dGraph(G, 'gics')
```



- Intercluster nodes : Mothercare (retail), Qiagen (pharmaceutical tests)

[135]:
```
plt.figure(figsize=(15,6))
plt.plot(list(range(w_unnormalized.size)), w_unnormalized)
plt.show()
```



2.4 Relation with Traditional Spectral Embedding

- **(L) Traditional spectral embedding** we look at the symmetric laplacian of the graph

$$L = D - W$$

For which the eigendecomposition is

$$L = U\Lambda U^T$$

And we choose as an embedding

$$X = U$$

Keeping the first k columns of X when projecting in lower dimensional space.

- **(K) Commute time method** As it happens, the final embedding we obtained is closely related to traditional spectral embedding. Remember we wrote :

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

And we chose as an embedding

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

Keeping the first k columns of X when projecting in lower dimensional space.

Well, the matrices of eigenvectors U in the eigendecomposition of K and L are the same !

| Commute Time Distance | Traditional Spectral Embedding | |:----------:|:----------:| |

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

|

$$X = U$$

|

- ($K = L^{-1}$) **Relation** Indeed, it can be shown through combinatorial properties of the graph that commute time distance can be related to the **inverse** L^{-1} of L

  We have the equality :
  $$L^{-1} = K = U\Gamma U^T$$
  Remark, when $L = U\Lambda U^T$, that
  $$L^{-1} = U\Lambda^{-1}U^T$$
  This is why the eigenvectors are the same. The eigenvalues, however, are naturally reversed. So we can state as a final equality :
  $$\Gamma = \Lambda^{-1}$$

| Commute Time Distance | Traditional Spectral Embedding | |:----------:|:----------:| |
$$L^{-1} = U\Lambda^{-1}U^T$$
|
$$L = U\Lambda U^T$$
| |
$$X = U\Lambda^{-1/2}$$
|
$$X = U$$
|

- In this project, we are using the **traditional Laplacian embedding**.

  This is the empiricially tested and robust way of embedding graphs. From the previous study we understand why **it makes sense** to look at such an object and why it has **such interesting properties** (like clustering etc).

2.5 Energy: different interpretation of eigengaps

With $L = D - W$, remark that for $x \in \mathbb{R}^n$:

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

So we have the equivalent of an energy minimisation problem. Eigenvalues are energy level of the system, and eigenvectors are the description of the whole system at one exact energy level.

Clustering properties of the graph, as measured by "cuts" for instance, are approximated via equatoins of the type $x^T Lx$. That is why its study is important is important.

We have approximately the same equation for k-dim embedding. With $X \in \mathcal{M}_{k,n}$:

$$tr(X^T LX) = \sum_{ij} w_{ij}\|X_i - X_j\|^2$$

- An eigengap is then an **energy jump**
- High energy eigenvectors lose the ablity to effectively "separate" parts of the graph.

2.6 Perturbation Theory : yet another interpretation

Say we have k perfect clusters in our graph, denote L_k the corresponding laplacian. Then for a small perturbation $\overline{L}_k$, we have :

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

Where V_k is the subspace spanned by the first k eigenvectors, and d is a distance between those subspaces. * The eigengap $|\lambda_{k+1} - \lambda_k|$ is a good measure of the effectiveness with which the embedding "locates" the clusters

3 Spectral Embedding - Directed Graph

Problem Now, the laplacian defined as $L = D - W$ is not **symmetric**. The equations crumble. The spectral properties related to clustering etc are lost.

Generalisation

- $\Phi = \text{diag}(\pi)$: **define the new degree matrix**.

  The trick is to consider the linkage graph through the lens of **Markov chains**. Consider P **the transition matrix** associated with the graph. Probability to go from node i to node j :

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

  Write $x_0 \in \mathbb{R}^n$ any initial probability distribution on nodes. Then :

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

  Let the walk converge. 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. Write

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

  This is the equivalent of our degree matrix.

- **(Pagerank) Slight modification of** P, with a sound interpretation in the context of financial markets. From each node :
  - High probability α to jump to the nodes it is connected to
  - Low probability $1 - \alpha$ to randomly jump to any other node.

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

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

$$\hat{A} = \frac{\Phi W + W^T \Phi}{2}$$

Remark that $ ()_{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 graph

- $\hat{L}$ **: define the new Laplacian matrix**

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

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

- **We get the embedding from the eigendecomposition of** $\hat{L}$

Theoretical results guarantee similar properties in terms of clustering etc

4 Normalized vs Unnormalized

- **Unnormalized :** $L = D - W$.

Energy Formulation	Total energy
$\sum_{ij} w_{ij}(x_i - x_j)^2$	tr(L) = $\sum_i d_i$

We have been working with this one from the beginning. Problem : what happens when degrees are very heterogeneous ? First eigenvectors tend to act as Dirac functions

- **Normalized :** $\hat{L} = D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2}$.

Energy Formulation	Total energy
$\sum_{ij} w_{ij}(\frac{x_i}{\sqrt{d_i}} - \frac{x_j}{\sqrt{d_j}})^2$	tr() = n

We penalize low degrees vertex from getting far from other points. This should give a more balanced result.

Formally, Let's write

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

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 :

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

- Unnormalized Laplacian optimizes **1.**
- Unnormalized Laplacian optimizes simultaneously **1.** and **2.**

Perturbation Theory

Let's write :

$$L = D - W$$
$$L_{norm} = I - D^{-1/2} W D^{-1/2}$$
$$L_{rw} = I - D^{-1} W$$

For k perfect clusters (C_i), the first k eigenvectors are :

L	L_{rw}	L_{norm}
$U_i = \mathbb{1}_{C_i}$	$U_i = \mathbb{1}_{C_i}$	$U_i = D^{1/2}\mathbb{1}_{C_i}$

- Small perturbation of the normalized Laplacian may lead to an information loss about cluster position (some components can be very small if the degreesare very heterogeneous)