

Master Project title

Final master project

Cristina Gava

July 9, 2018

Contents

1. Introduction: the emerging field of graph signal processing	1
1.1 Mathematical description of the graph structure	3
1.2 the inverse covariance matrix	3
1.2.1 The graph Laplacian	3
1.3 Representing a graph: graph learning techniques and dictionary representation . .	5
1.4 The smoothness assumption	5
2. Brief state of the art	5
3. The algorithms we started from: Dorina's and Hermina's	5
4. Problem presentation	6
5. main improvements	6
5.1 first part: the dictionary learning under smoothness constraints	6
5.2 second part: the graph learning part	6
5.3 Merging the two approaches	6
6. further improvements	6
6.1 graph signal clustering	6
6.2 varargin	6
7. conclusions	6
8. acknowledgements	6

1. Introduction: the emerging field of graph signal processing

Nowadays huge amounts of data are collected every day, from engineering sciences to our personal data regarding health monitoring, to financial data or political influences. The analysis and the process of these huge datasets has become a non indifferent challenge, since, in fact, data tends to reside

on complex and irregular structures, which progressively required the introduction of innovative approaches to better handle and utilize them. [1] [2]

One of the central entities we will use in this work is the concept of *graph signal*. Graphs are generic data representation structures, which are useful in a great variety of fields and applications for everyday life and technology in general. These structures are composed by two main elements: the nodes and the edges; the former are a set of points, identifying an aspect of the graph structure itself, while the latter are connections between the nodes. Several structures we can encounter in natural entities and abstract constructions can be represented by a graph structure, and when we associate values to their set of nodes and edges we obtain a graph signal. To be specific, a graph signal is seen as a finite collection of samples located at each node in the structure and which are interconnected among themselves through the edges, to which we can associate numerical values representing the weights of the connections. The metric for these weights is not unique, as it depends on which relation between the nodes we are looking at: a typical metric for graph weights may be, for example, inversely proportional to the distance (physical or not) between two different nodes, but other options are possible.

As previously mentioned, graph structures appear to be significantly helpful when they are used to represent signals and their applications are the most varied: we could focus on transportation networks, where we could want to work with data describing human migration patterns or trade goods; we could also apply graph theory over social networks, in which users can be seen, for example, as our nodes while their connections to friends are our edges. [1] Brain imaging is another interesting application of graph signals: through it we could infer the inner structure of some regions of the brain, in a way that permits us to better understand physiological dynamics. [3] We could go on listing a considerable amount of valuable applications, from genetic and biochemical research to fundamental physical experiments; what these big amounts of data have in common is the fact that their complex inner structure makes it difficult to apply most of the well-known tools, like Principal Component Analysis (PCA), spectral analysis or Singular Value Decomposition (SVD), without redefining the signal structure. [2]

1.1 Mathematical description of the graph structure

In the field of Graph Signal Processing (GSP), spectral graph theory plays an important role, it focuses on analysing, constructing and manipulating graphs and makes uses of well know tools and concepts as frequency spectrum and the graph Fourier transform. In our work, the signals we are considering are defined on an undirected, connected, weighted graph $\mathcal{G} = \{\mathcal{V}, \varepsilon, W\}$ consisting on a finite set of vertices \mathcal{V} with $|\mathcal{V}| = N$, a set of edges ε and a weighted adjacency matrix W . The adjacency matrix contains the information of the connections between two nodes: if an edge e is present between two vertices i and j , then the entry $W_{i,j}$ represents the weight of that edge, while if there is no connection between two nodes, the value of the edge is null ($W_{i,j} = 0$). Moreover, if the graph was not connected and had K connected components, then we could decompose the graph signal over \mathcal{G} into M sections which can be processed each one independently of the other.

Over this structure, we can then lay (Cri says: cerca un verbo migliore) the signal we need to analyse: to be precise, a signal or function $f : \mathcal{V} \rightarrow \mathbb{R}$ defined on the vertices of our graph, can be described as a vector $f \in \mathbb{R}^N$ that has the signal value at the i^{th} vertex in \mathcal{V} as the i^{th} component of the vector f . Figure 1 shows an example of graph signal: the colours of the vertices represents the values the signal has on them.

1.2 the inverse covariance matrix

(Cri says: Vedere se aggiungere o meno questa sezione)

1.2.1 The graph Laplacian

To go from the graph vertex domain (which plays the same role of the time domain in the classical signal processing theory) to the spectral domain the *graph Laplacian operator*, or *Combinatorial graph Laplacian*, has been introduced, which is defined as $\mathcal{L} := D - W$, where D represents a diagonal matrix whose i^{th} diagonal element is equal to the sum of weights of all the edges incident to the vertex i . This entity is a difference operator, since it satisfies the condition:

$$((\mathcal{L}f)(i) = \sum_{j \in \mathcal{N}_i} W_{i,j}[f(i) - f(j)] \quad (1.1)$$

The element \mathcal{N}_i represents the set of vertices connected to vertex i by an edge.

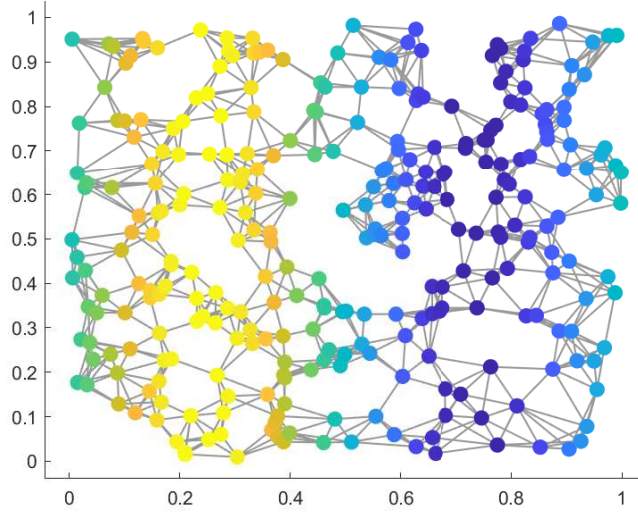


Figure 1: Random sensor graph

From its definition, the graph Laplacian \mathcal{L} is a real symmetric matrix, thus it has a complete set of orthonormal eigenvectors, here denoted by $\{u_l\}_{l=0,1,\dots,N-1}$, to which real and non-negative eigenvalues are associated ($\{\lambda_l\}_{l=0,1,\dots,N-1}$). Together with the eigenvectors, they satisfy the condition:

$$\mathcal{L}u_l = \lambda_l u_l, \text{ for } l = 0, 1, \dots, N - 1 \quad (1.2)$$

In the eigenvalues set, the one corresponding to 0 has a multiplicity equal to the number of connected components of the graph; from this, since we are considering only connected graphs, we can assume the Laplacian eigenvalues have the distribution: $0 = \lambda_0 < \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_{N-1} = \lambda_{max}$ and, of course, the set $\sigma(\mathcal{L}) = \{\lambda_0, \lambda_1, \dots, \lambda_{N-1}\}$ is the entire spectrum of our signal. (Cri says: trova un modo per usare la L polacca con questo font)

Eigenvalues and eigenvectors are then used to build up the graph version of a well known and useful transform, which is the *classical Fourier Transform* and that we recall here being defined as:

$$\hat{f}(\xi) = \langle f, e^{2\pi i \xi t} \rangle = \int_{\mathbb{R}} f(t) e^{-2\pi i \xi t} dt \quad (1.3)$$

and that we can see as the expansion of a function f in terms of the complex exponentials, elements

which represent the eigenfunctions of the one-dimensional Laplace operator:

$$-\Delta(e^{2\pi i \xi t}) = -\frac{\partial^2}{\partial t^2} e^{2\pi i \xi t} = (2\pi \xi)^2 e^{2\pi i \xi t} \quad (1.4)$$

From the observation of the classical Fourier Transform, we can analogously define the *Graph Fourier Transform* \hat{f} of any function $f \in \mathbb{R}^N$ on the vertices of \mathcal{G} as the expansion of f in terms of the eigenvectors of the graph Laplacian [3]:

$$\hat{f}(\lambda_l) = \langle f, u_l \rangle = \sum_{i=1}^N f(i) u_l(i) \quad (1.5)$$

while, at the same time, the *inverse Graph Fourier Transform* is given by:

$$f(i) = \sum_{l=0}^{N-1} \hat{f}(\lambda_l) u_l(i) \quad (1.6)$$

1.3 Representing a graph: graph learning techniques and dictionary representation

In the previous section we presented the main reason why signal processing field expressed the necessity of having different new structures in order to better represent the amount of information we can collect from a phenomena: what we did not focused on yet, is the fact that these amount of data usually are largely redundant, since they are a densely sampled version of the signal and they may represent multiple correlated versions of the same physical event. So normally the significative information regarding the underlying processes is largely reducible in dimensionality with respect of the collected dataset. [4] So we can obtain the data representations starting from the idea that our observations can be described by a sparse subset of elementary signals - so called *atoms* - taken from an overcomplete dictionary. When the dictionary forms a basis, then every signal can be univocally represented through the linear combination of the dictionary atoms. [5]

1.4 The smoothness assumption

2. Brief state of the art

3. The algorithms we started from: Dorina's and Hermina's

4. Problem presentation

5. main improvements

5.1 first part: the dictionary learning under smoothness constraints

5.2 second part: the graph learning part

5.3 Merging the two approaches

6. further improvements

6.1 graph signal clustering

6.2 varargin

7. conclusions

8. acknowledgements

References

- [1] A. Ortega, P. Frossard, J. Kovačević, J. M. F. Moura, and P. Vandergheynst, “Graph Signal Processing: Overview, Challenges and Applications,” pp. 1–18, 2017.
- [2] A. Sandryhaila, “Signal Processing,” no. September, pp. 80–90, 2014.
- [3] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, “The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains,” *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, 2013.
- [4] I. Tosić and P. Frossard, “Dictionary Learning, What is the right representation for my signal?,” *Signal Processing Magazine, IEEE*, vol. 28, no. 2, pp. 27–38, 2011.
- [5] B. R. Rubinstein, A. M. Bruckstein, and M. Elad, “Dictionaries for Sparse Representation Modeling.pdf,” vol. 98, no. 6, pp. 1045–1057, 2010.