Graph Learning from Data

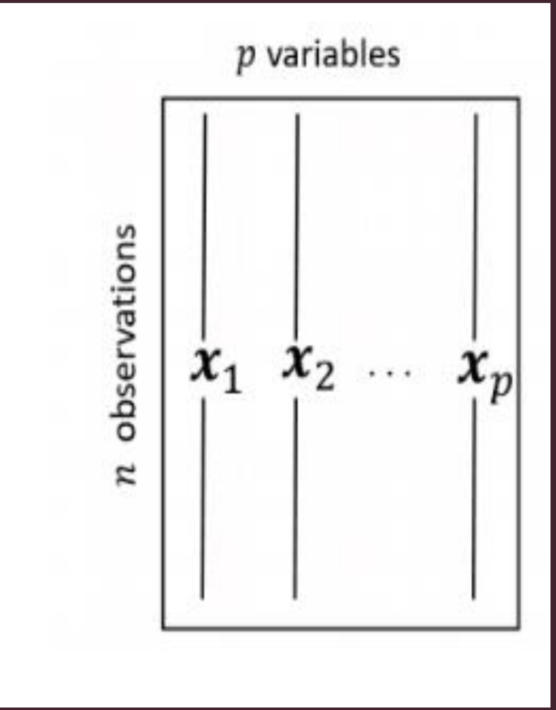
Outline

- Introduction
- Motivation
- Background for problem
- Content
- Future scope
- References

Introduction

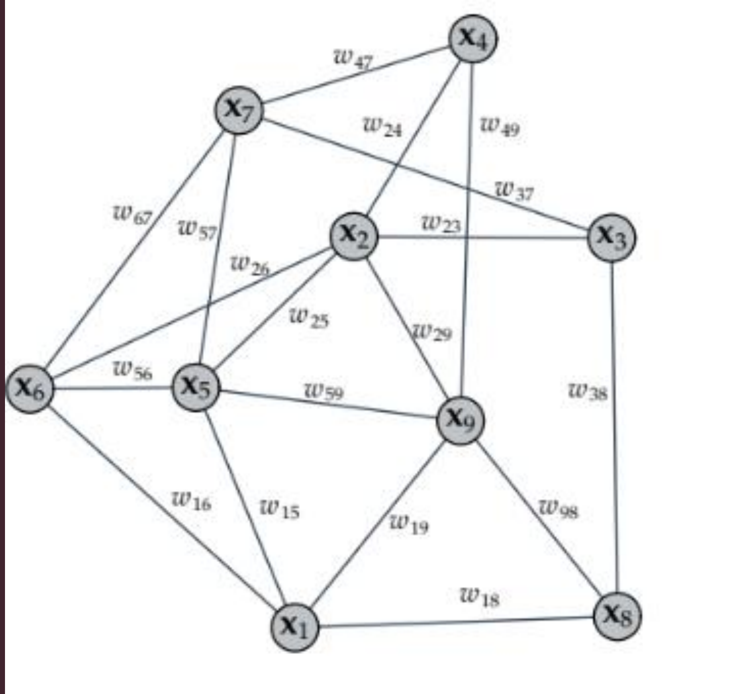
- Graphs are the fundamental mathematical structures used in various fields to represent data , signals and processes.
- Graphs are a general language for analyzing entities with relations/interactions.
- In the domain of signal processing and Machine learning ,the graphs are used for modelling the high dimensional datasets.
- The need for graph learning arises mainly because of the type of data we have in our nature.

Motivation

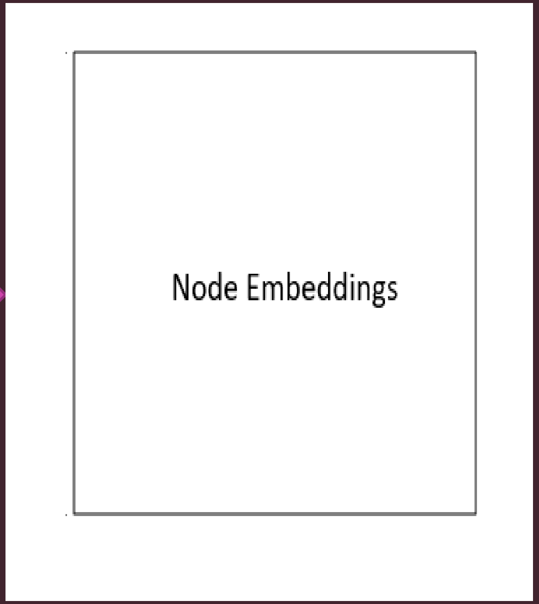


Graph learning

from Data



GCN



- Images 1st and 2nd take from Reference 3

Where Graph learning from Data is used ?

- Networked Data applications: signal processing ,learning and analysis on computer, social ,biological and many more type of networks.
- Signals/data – problems : Weighted Graphs can provide sparse representation for robust modelling of signals/data.
- More examples: Filtering , transformation and sampling of signals defined on graphs , semi supervised learning etc.

Background information about graphs

- Weighted Graph is denoted by :

G = (V,E,W,S)

Where , V = Set of vertices(or nodes)
E = Set of edges(or links)
W = Weight matrix (For edges)
S = Vertex weights Matrix (Self loops)

Let X be your Data matrix where $X \in R^{n \times N}$,Then |V| = n , i.e. each row of X matrix is a data point attached to one of the nodes of graph

Here the W is a non-negative matrix i.e. W = ($w_{ij} \geq 0$)

And S is a diagonal matrix defining the weights for the self loops(if any)

- We normally work with Graph Laplacian Matrix because of its certain properties.

Graph Laplacian Matrix

- Generalized Graph Laplacian Matrix is given by :

$$L = S + D - W$$

Here D is Degree Matrix defining the degree of each node.

We will talk about undirected graphs in this presentation :

- Laplacian matrix is symmetric, positive-semidefinite.
- The off diagonal elements are non-positive.

Graph learning from Data under Structural and Laplacian Constraints

- The Graph learning problems are posed as the estimation of Graph Laplacian matrices.
- In the remaining presentation we will focus on Estimation of GL matrices from observed Data under Given structural constraints like the level of sparsity, Graph connectivity etc.

Problem Formulation

- Here we represent our graph as :

$$G = (V\ , E\ , f_w, f_v)$$

This is a weighted graph with n vertices (let) . Here V and E have usual meanings.

f_w is a real valued weight function i.e. $f_w\left(\left(v_i, v_j\right)\right) \geq 0$ for all $i \neq j$

f_v is the real-valued vertex(self-loop) weight function

- We will restrict ourself to undirected weighted graphs with non-negative edge weights.
- We also introduce Connectivity Matrix A .

$$A_{i,j} = \begin{cases} 1\ , if\ \ W_{i,j} \neq 0 \\ 0\ , if\ W_{i,j} = 0 \end{cases}$$

Different types of GL matrices

- Here , we focus on the following set of GL matrices :

1) Generalized Graph Laplacian ($L_g = S + D - W$)
2) Combinatorial Graph Laplacian ($L_g = D - W$)

These two we already discussed .

3) Diagonally Dominant graph Laplacian : (L_D)

$$L_d = \{L \,|L \geq 0, (L)_{i,j} \leq 0 \; for \; i \neq j, \; L\mathbf{1} \; \geq 0\}$$

Note: if a Weighted graph has S $\geq O$,it's GGL matrix is diagonally-dominant.

Problem Formulation

- Goal: Given data matrix $X \in R^{kxn}$ (each column corresponds to a vertex) , Estimate the GL matrix from a data statistic R.
- The data statistic R is chosen according to application and underlying statistical assumptions, Ex . Sample covariance of X , Kernel matrix derived from data

  i.e. $\mathrm{R} = K(X, X)$ where K is pos. definite kernel function

Example: (RBF or polynomial)

Problem formulation

- The following objective function is minimized :

$$Tr(\theta R) - \log(\det(\theta)) + \|\theta \odot H\|_1$$

Here Target Matrix variable = θ ,nxn dimensional matrix

H = Symmetric regularization matrix.

- First two term together are Data Fidelity term.
- 3rd term is promoting sparsity(weighted L1 regularized term)
- Suitable H is selected for transforming above non-smooth function to a smooth function.

Transforming to Smooth function

- Because of non-negativity of edge weights , we can write :

$$\|\theta \odot H\|_1 = Tr(\theta H)$$

- By choosing $H = \alpha(2I - 11^T)$

$$Tr(\theta H) = \alpha\|\theta\|_1$$

- As trace is a linear operator ,we can rewrite the obj. function:

$$Tr(\theta K) - \log(\det(\theta)), where\ K = R + H$$

Optimization Problem

- Set of constraints on given set of graph Laplacians L and a connectivity matrix A:

$$L(A) = \left\{ \theta \in L \middle| \begin{array}{l} (\theta)_{ij} \leq 0 \; if \; (A)_{ij} = 1 \\ (\theta)_{ij} = 0 \; if \; (A)_{ij} = 0 \end{array} \right. \; for \; i \neq j$$

- GGL Problem :

$$\min_{\theta} \quad Tr(\theta K) - \log(\det(\theta))$$

subject to $\theta \in L_G(A)$

- Similarly other two can be defined.

Proposed Algorithm and Convergence

- The proposed optimization problems are solved using the Block coordinate Descent algorithm , espically because of the computation inefficiency of other algorithms like Vanilla Gradient descent or second order algorithms.
- In The proposed Algorithms, each block coordinate descent iteration has $O(T_p(n) + n^2)$.

Future Work

- The problem proposed in these methods can not guarantee the solution will have specific graph Topological properties (example: k-parite,Tree etc.) The problem of learning the graph with given topological constraints is in general a non-convex problem.
- The convergence analysis in the second paper for Graph learning under spectral constraints require certain approximation to assume which doesn’t hold always so other algorithms are proposed for better convergence results like using ADMM and MM in solving the optimization problem.