Screening for Graph Learning

Internship Defense 2024

Amrithya Balaji

Laboratoire Hubert Curien

Supervisor: Benjamin Girault

Overview

- Represent real-world datasets as graphs, which connect entities (nodes) via relationships (edges).
- Graph learning enables us model complex relationships and dependencies.

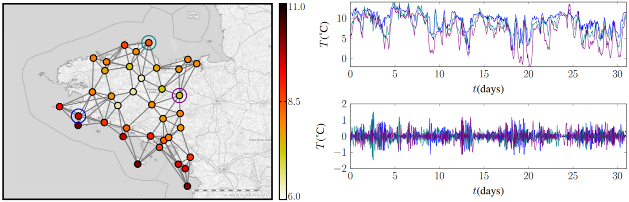


Figure. 1: Temperature variations in January 2014 in Brittany.Average temperature at every location(left).Three stations circled in (right) with respect to time variations.(left) [Gir15]

Objective

- To create a graph with the important connections between data points.
- The graph should be sparse.
- To **implement screening** in graph learning algorithm

Background on Graphs

Graphs

- versatile data representation.
- useful for modeling high-dimensional data.
- can be represented as a set of vertices $\mathcal{V}$, edge set $\mathcal{E}$, and an edge weight function w, as shown below.

$$\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$$

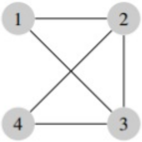
$$\text{with } w : \mathcal{E} \to \mathbb{R}, \quad e \mapsto w(e)$$

Background on Graphs

Laplacian of a Graph

- Represent the structure of the graph.
- The eigenvalues and eigenvectors of the Laplacian matrix are used to understand the properties of the graph, such as connectivity.

$$\mathbf{L} := \mathbf{D} - \mathbf{W}$$



$$\mathbf{D} = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix} \quad \mathbf{W} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

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

Figure. 2: Calculating the Laplacian of a graph

Background on Graphs

Laplacian of a Graph

Let the unweighted incidence matrix of a graph with m edges and n vertices be $B = [\mathbf{b}_1 \cdots \mathbf{b}_n]$ and $W = diag(\mathbf{w}_1 \cdots \mathbf{w}_n)$, the Laplacian also verifies,

$$L = \sum_e (w_e b_e b_e^T). \tag{1}$$

Background on Graphs

Sparse vs Well connected

The main goal of this paper is to model the statistical relations between vertices while remaining sparse.

- *Sparse graph* - Number of edges is significantly smaller compared to the maximum possible number of edges.
- *Well-connected graph* - higher edge density, larger proportion of possible edges.

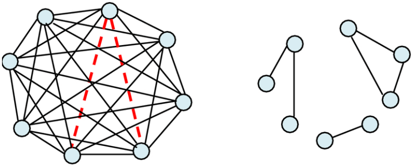


Figure. 3: A well connected graph (Left) and a sparse graph(right) [BSS19]

Coordinate Minimization

Coordinate Minimization Optimization algorithm used in machine learning

---

Coordinate Minimization

Input: objective function $f : \mathbb{R}^n \to \mathbb{R}$
while not converged **do**
  Choose coordinate x_i
  Compute $\min_{x_i} f(x)$
end while

Data model

We use the inverse covariance estimation approach.

Covariance Matrix of a Graph Signal [GPO23]:

Theorem 1 [GPO23] With the power spectrum $\gamma(\lambda) = (1+\lambda)^{-1}$ where γ is the the variance of spectral component of the signal, the covariance matrix of a graph signal can be written using the (L, Q)-GFT as:

$$\Sigma = [Q + L]^{-1}. \tag{2}$$

where $Q = diag(q)$ and q is the vertex importance vector

Multivariate Gaussian Distribution

For a random vector $x \in \mathbb{R}^N$ following a multivariate Gaussian distribution with mean vector μ and covariance matrix Σ, the probability density function (PDF) is given by:

$$p(x) = \frac{1}{(2\pi)^{N/2} \det(\Sigma)^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right)$$

When the mean vector $\mu = 0$, this simplifies to:

$$p(x) = \frac{1}{(2\pi)^{N/2} \det(\Sigma)^{1/2}} \exp\left(-\frac{1}{2}x^T \Sigma^{-1} x\right)$$

For our Data model, Maximum Likelihood

Cost function of our graph learning:

With the negative log-likelihood of our PDF and Theorem 1, we have the cost function as:

$$F(L) = -\log\det\left(L + \frac{1}{N}J\right) + \mathrm{tr}(LS) \tag{3}$$

where J is the all-one matrix.

$$F(L, Q) = -\log\det(Q + L) + \mathrm{tr}((Q + L)S) \tag{4}$$

where S is the empirical covariance matrix.

Trace term: weighted ℓ_1 norm

The trace term in the cost function (4) can be written as a sum $\mathrm{tr}(LS) + \mathrm{tr}(QS)$. Further more, the terms can be written as:

$$\text{tr}(LS) \text{ can be written as } \text{tr}(B^T w_e BS) = \sum_{i=1}^{n} (w_e B^T SB)$$

where $(B^T w_e BS) = h_e(\text{edge cost})$ and $w_e = \text{edge weight}$.

$$\text{tr}(QS) \text{ can be written as } \text{tr}(q_i \delta_i^t \delta_i S) = \sum_{i=1}^{m} (q_i \delta_i^T S \delta_i)$$

where $(\delta_i^T S \delta_i) = p_i(\text{vertex cost})$ and $q_i = \text{vertex importance}$

and with δ_i as a kronecker vector.

New form of the cost function

$$
\begin{aligned}
F(q,w) = &- \log\det\left(\sum_{i=1}^{n} \delta_i \delta_i^T q_i + \sum_{e=1}^{m} \mathbf{b}_e \mathbf{b}_e^T w_e\right) \\
&+ \operatorname{tr}\left(\left(\sum_{i=1}^{n} \delta_i \delta_i^T q_i + \sum_{e=1}^{m} \mathbf{b}_e \mathbf{b}_e^T w_e\right)\Sigma\right)
\end{aligned} \tag{5}
$$

Updates to our data model

Coordinate minimization iterates through all edges and vertices, updating their weights. Updates can be calculated with the introduced hyperparameter q_{min} where $q_{min} > 0$.

The optimal updates are :

$$\delta_e^{(t)} = \max\left(-w_e^{(t)}, \frac{1}{h_e} - \frac{1}{r_e}\right),$$

$$\delta_i^{(t)} = \max\left(q_{\min} - q_i^{(t)}, \frac{1}{p_i} - \frac{1}{u_i}\right).$$

where

$$r_e = b_e(Q+L)^{-1}b_e = \text{ effective resistance and}$$

$$u_i = (Q+L)^{-1}_{ii} = \Sigma_{ii} = \text{ vertex effective resistance.}$$

Why Screening

- To simplify a problem by identifying and removing redundant variables.
- In our problem, only a few are non-zero and have an impact on the final result.

Duality : Primal Problem

We have our cost function (4) in the form of

$$x \mapsto f(Ax) + g(x)$$

With reparametrization term $r = q - q_{\min}1$ and our primal problem $(\mathbf{r}^*, \mathbf{w}^*)$, we have:

$$\arg\min_{\mathbf{r},\mathbf{w}\geq 0} \; -\log\det\left(\sum_{i=1}^{n} r_i \delta_i \delta_i^T + \sum_{e=1}^{m} w_e \mathbf{b}_e \mathbf{b}_e^T + q_{min}\mathbf{I}\right) + \mathbf{p}^T\mathbf{r} + \mathbf{h}^T\mathbf{w}$$

Region-Free Screening

Duality : Fenchel conjugate

Since

$$f(P) = -\log\det(P). \tag{6}$$

(6) is assumed to be convex, closed, and proper. We compute the Fenchel conjugate f^* of (6).

$$f^*(U) = \sup_P(\mathrm{tr}(P^T U) + \log\det(P)) \tag{7}$$

With Maximization Condition $\frac{\partial}{\partial P}(\mathrm{tr}(P^T U) + \log\det(P)) = 0$:

$$U + P^{-1} = 0 \implies P = -U^{-1} \tag{8}$$

Fenchel Conjugate $f^*(U)$:

Substituting (8) in (7) [HED22]:

$$f^*(U) = -\mathrm{tr}(U^{-1}U) + \log\det(-U^{-1}) + \mathbb{I}(U \in \mathbb{S}_n^-)$$

Duality : Dual Problem

From the computed Fenchel conjugate, the dual problem on U:

$$d(U) = -q_{\min}\mathrm{tr}(U) + n + \log\det(U) + \mathbb{I}(U \in \mathbb{S}_n^-)$$

Dual Constraints:

Derived from the primal constraints, we have:

$$\delta_i^T U \delta_i \leq p_i, \quad \forall i \in \{1, \dots, n\}$$

$$b_e^T U b_e \leq h_e, \quad \forall e \in \{1, \dots, m\}$$

Dual Cost:

$$U^* = \arg\max_{U \in S_n} d(U)$$

$$d(U) := -q_{\min}\,\mathrm{tr}(U) + n + \log\det(U) \tag{9}$$

Region-Free Screening

Screening approach:

- Update the relaxed primal cost = cost_function + the primal cost update without the non-negative constraints.
- Primal cost update =

$$\text{Primal cost update} = \begin{cases} -\log(1+\delta_e r_e) - \delta_e h_e & \text{when } l = \text{edge}, \\ -\log(1+\delta_i u_i) - \delta_i p_i & \text{when } l = \text{vertex}. \end{cases}$$

- Compute a dual admissible solution

$$\bar{U} = \min\left\{ (\frac{p_i}{u_i})_i, (\frac{h_e}{r_e})_e \right\} \times \Sigma$$

- Compute the dual cost $d(\bar{U})$ according to (9)
- If dual cost $d(\bar{U}) >$ primal cost $p_l(r_l, w_l)$ set the variable to zero or update the w_e or q_i

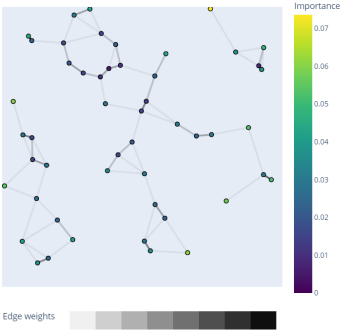
Results

Constraints		Without Q		With Q	
Range	Screening	Avg run time in sec	Sparsity	Avg run time in sec	Sparsity
1	✓	**10.3**	90.6%	**13.1**	90.9%
1	✗	11.1		13.7	
0.1	✓	117.6	9.8%	45.0	85.3%
0.1	✗	**104.4**		**39.0**	
0.01	✓	120.0	0%	3.4	79.5%
0.01	✗	**108.0**		**2.9**	

1: Comparison of Average Run Time in Seconds with and without Q

Results

(a) Initial graph



(b) Best graph

Figure. 4: Comparison of initial and best graphs

1. Introduction
2. Coordinate Minimization
3. Region-Free Screening
4. Results
5. **References**

[BSS19] Reham Barham, Ahmad Sharieh, and Azzam Sleit.
Multi-moth flame optimization for solving the link prediction problem in complex networks.
Evolutionary Intelligence, 12, 12 2019.

[Gir15] Benjamin Girault.
Stationary graph signals using an isometric graph translation.
In *2015 23rd European Signal Processing Conference (EUSIPCO)*, pages 1516–1520, 2015.

[GPO23] Benjamin Girault, Eduardo Pavez, and Antonio Ortega.
Joint graph and vertex importance learning, 2023.

[HED22] Cédric Herzet, Clément Elvira, and Hong-Phuong Dang.
Region-free Safe Screening Tests for l1-penalized Convex Problems.
In *Eusipco 2022 - 30th European Signal Processing Conference*, pages 1–5, Belgrade, Serbia, August 2022.

[PO20] Eduardo Pavez and Antonio Ortega.
An efficient algorithm for graph laplacian optimization based on effective resistances, 2020.

Thank you