Large Scale Graph Learning Using Smooth Signals

G.Yashwanth Naik EE18BTECH11017

Advisor:Prof ADITYA SIRIPURAM

November 2020

Abstract

Previously, the time complexity of state-of-the-art model to learn graph from smooth signals is $O(n^2)$ for n samples. In this paper the authors Vassilis Kalofolias, Nathanaël Perraudin does it with a time complexity of $O(n \log(n))$, with quality that approaches the exact graph learning model.

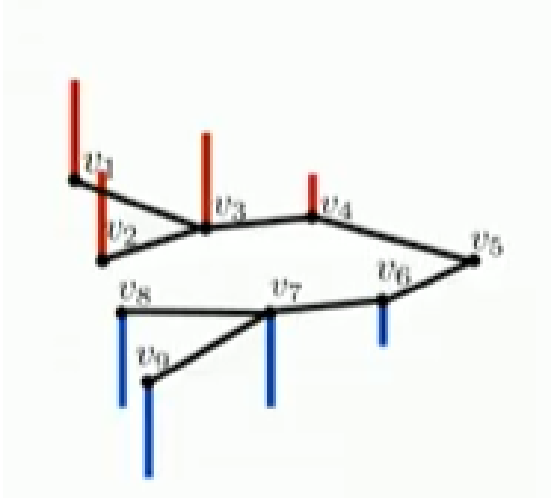
Smooth Signals

Let input be a set of vectors $x_1, \ldots, x_n \in \mathbb{R}^d$ on a given weighted undirected graph is usually quantified by the Dirichlet energy.

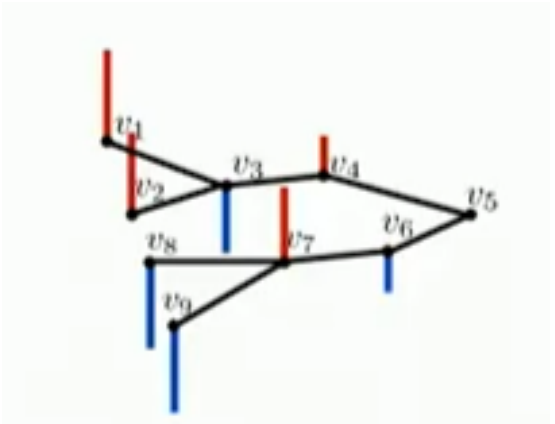
$$\frac{1}{2} \sum_{i,j} W_{ij} \left\| x_i - x_j \right\|^2 = \operatorname{tr} \left(X^\top L X \right) \tag{1}$$

where $W_{ij} \in \mathbb{R}_+$ denotes the weight of the edge between nodes i and j, $L = D - W$ is the graph Laplacian, and $D_{ii} = \sum_j W_{ij}$ is the diagonal weighted degree matrix.

Smooth Signals Examples



(a) Smooth Signal



(b) Not a smooth signal

For figure a the value of $\text{tr}\left(X^\top LX\right) = 1$ and for figure b $\text{tr}\left(X^\top LX\right) = 21$.So we can observe that $\text{tr}\left(X^\top LX\right)$ is measure of smoothness of a signal. So we need to minimize it.

.

Graph Learning Model

In 2016, Kalofolias proposed a unified model for learning a graph from smooth signals, that reads as follows:

$$\min_{W \in \mathcal{W}} \|W \circ Z\|_{1,1} + f(W)$$

Here, $Z_{ij} = \|x_i - x_j\|^2$, $\circ$ denotes the Hadamard product, and the first term is equal to tr $\left(X^\top L X\right)$ The optimization is over the set $\mathcal{W}$ of valid adjacency matrices (non-negative, symmetric, with zero diagonal).

The role of matrix function $f(W)$ is

- to prevent W from obtaining a trivial zero value($W = 0$)
- to control sparsity
- impose further structure using prior information

log *Model*, ℓ_2 model

Kalofolias obtained state-of-the-art results using

$$f(W) = -\alpha 1^\top \log(W1) + \frac{\beta}{2}\|W\|_F^2$$

where $1 = [1, \dots 1]^\top$. We will call this the log model. The previous state of the art was proposed by Hu et al. (2013) and by Dong et al. (2015), using

$$f(W) = \alpha\|W1\|^2 + \alpha\|W\|_F^2 + \mathbb{1}\{\|W\|_{1,1} = n\}$$

where $1\{$ condition $\} = 0$ if condition holds, ∞ otherwise. In the sequel we call this the ℓ_2 model. since $W1$ is the node degrees' vector, the log model prevents the formation of disconnected nodes due to the logarithmic barrier, while the ℓ_2 model controls sparsity by penalizing large degrees due to the first term.

Constrained Edge Pattern

- The important optimization will happen in this step. In traditional graph learning, all $\binom{N}{2}$ possible edges between n nodes are considered, that results in a cost of at least $\mathcal{O}\left(n^2\right)$ computations per iteration. Often, however, we need graphs with a roughly fixed number of edges per node, like in k -NN graphs.
- The log model can be solved efficiently when a constrained set $\mathcal{E}^{\text{allowed}} \subseteq \{(i,j) : i < j\}$ of allowed edges is known a priori.
- Accordingly, in $\tilde{z} = z\left(\mathcal{E}^{\text{allowed}}\right)$ we only keep the corresponding pairwise distances from matrix Z.
- We use approximate nearest neighbors (A-NN) algorithm which has an overall complexity of $O(n \log(n) d)$ for d-dimensional data.

Automatic Parameter Selection

Reduction to a Single Optimization Parameter:
Proposition 1. Let $W^*(Z, \alpha, \beta)$ denote the solution of model (3) for input distances Z and parameters $\alpha, \beta > 0$. Then the same solution can be obtained with fixed parameters $\alpha = 1$ and $\beta = 1$, by multiplying the input distances by $\theta = \frac{1}{\sqrt{\alpha\beta}}$ and the resulting edges by $\delta = \sqrt{\frac{\alpha}{\beta}}$:

$$W^*(Z, \alpha, \beta) = \sqrt{\frac{\alpha}{\beta}} W^* \left(\frac{1}{\sqrt{\alpha\beta}} Z, 1, 1 \right) = \delta W^*(\theta Z, 1, 1)$$

Proof. This follows from the property that

$$W^*(Z, \alpha, \beta) = \gamma W^*(Z, \frac{\alpha}{\gamma}, \beta\gamma)$$

with $\gamma = \sqrt{\frac{\alpha}{\beta}}$ and divide all operands by $\sqrt{\alpha\beta}$.

Setting the remaining regularization parameter:
We need find a relation between θ and the desired sparsity (the average number of neighbors per node).

Sparsity Analysis for One Node:

Keeping only one column w of matrix W, we arrive to the simpler optimization problem.

$$\min_{w \in \mathbb{R}^n_+} \theta w^\top z - \log\left(w^\top 1\right) + \frac{1}{2}\|w\|_2^2$$

Theorem . Suppose that the input vector z is sorted in ascending order. Then the solution of above problem has the form

$$w^* = \max(0, \lambda^* - \theta z) = [\lambda^* - \theta z_{\mathcal{I}}; 0]$$

with

$$\lambda^* = \frac{\theta b_k + \sqrt{\theta^2 b_k^2 + 4k}}{2k}$$

The set $\mathcal{I} = \{1, \ldots, k\}$ corresponds to the indices of the k smallest distances z_i and b_k is the cumulative sum of the smallest k distances in z, $b_k = \sum_{i=1}^{k} z_i$.

Proof: Introducing a slack variable l for the inequality constraint so that the KKT optimality conditions are,

$$\begin{aligned} \theta z - \tfrac{1}{w^\top 1} + w - l &= 0 \\ w &\geq 0 \\ l &\geq 0 \\ l_i w_i &= 0, \forall i \end{aligned}$$

the optimum of w can be revealed by introducing the term $\lambda^* = \frac{1}{w^{*\top} 1}$ and rewrite above as

$$w^* = \lambda^* - \theta z + l$$

Then, we split the elements of w in two sets, $\mathcal{A}$ and $\mathcal{I}$ according to the activity of the inequality constraint ($w \geq 0$) so that $w_\mathcal{I} > 0$ (inactive) and $w_\mathcal{A} = 0$ (active).

Lemma 1. An element w_i^* of the solution w^* of the above problem is in the active set $\mathcal{A}$ if and only if it corresponds to an element of z_i for which $\theta z_i \geq \lambda^*$

Proof. $(\Rightarrow)$: If w_i is in the active set we have $w_i = 0$ and $l_i \geq 0$, therefore, we have $\theta z_i - \lambda^* \geq 0$.$(\Leftarrow)$: Suppose that there exists $i \in \mathcal{I}$ for which $\theta z_i \geq \lambda^*$. The constraint being inactive means that $w_i^* > 0$. But we have that $l_i = 0$ and gives $w_i^* = \lambda^* - \theta z_i \leq 0, a$ contradiction.

Proof(Theorem,Cont..). As elements of θz are sorted in an ascending order, the elements of $\lambda^* - \theta z$ will be in a descending order. Furthermore, we know from Lemma 1 that all positive w_i^* will correspond to $\theta z_i < \lambda^*$. Then, supposing that $|\mathcal{I}| = k$ we have the following ordering:

$$-\theta z_1 \geq \cdots \geq -\theta z_k > -\lambda^* \geq -\theta z_{k+1} \geq \cdots \geq -\theta z_n \Rightarrow$$
$$\lambda^* - \theta z_1 \geq \cdots \geq \lambda^* - \theta z_k > 0 \geq \lambda^* - \theta z_{k+1} \geq \cdots \geq \lambda^* - \theta z_n$$

In words, the vector $\lambda^* - \theta z$ will have sorted elements so that the first k are positive and the rest are non-positive. Furthermore, we know that the elements of l in the optimal have to be 0 for all inactive variables w_I^*, therefore $w_I^* = \lambda^* - \theta z_{\mathcal{I}}$. The remaining elements of w will be 0 by definition of the active set:

$$w^* = [\lambda^* - \theta z_1, \cdots, \lambda^* - \theta z_k, 0, \cdots, 0]$$

What remains is to find an expression to compute λ^* for any given z. Keeping z ordered in ascending order, let the cumulative sum of z_i be $b_k = \sum_{i=1}^{k} z_i$. Then, from the definition of $\lambda^* = \frac{1}{w^{*T}1}$ and using the structure of w^* we have

$$w^{*\top}1\lambda^* = 1 \quad \Rightarrow \quad \left(k\lambda^* - \theta z_{\mathcal{I}}^{\top}1\right)\lambda^* = 1 \quad \Rightarrow \quad k\left(\lambda^*\right)^2 - \theta b_k \lambda^* - 1 = 0$$

which has only one positive solution,

$$\lambda^* = \frac{\theta b_k + \sqrt{\theta^2 b_k^2 + 4k}}{2k}$$

If we know the distances vector z and we want a solution w^* with exactly k non-zero elements, what should the parameter θ be?

Proof. From the proof of above Theorem we know that $\|w^*\|_0 = k$ if and only if $\lambda^* \in [\theta z_k, \theta z_{k+1})$. We can rewrite this condition as

$$\begin{aligned}
\theta z_k \leq \frac{\theta b_k + \sqrt{\theta^2 b_k^2 + 4k}}{2k} < \theta z_{k+1} \Leftrightarrow \\
2k\theta z_k - \theta b_k \leq \sqrt{\theta^2 b_k^2 + 4k} < 2k\theta z_{k+1} - \theta b_k \Leftrightarrow \\
4k^2\theta^2 z_k^2 - 4k\theta^2 b_k z_k \leq 4k < 4k^2\theta^2 z_{k+1}^2 - 4k\theta^2 b_k z_{k+1} \Leftrightarrow \\
\theta^2 \left(kz_k^2 - b_k z_k\right) \leq 1 < \theta^2 \left(kz_{k+1}^2 - b_k z_{k+1}\right)
\end{aligned}$$

So $\theta \in \left(\frac{1}{\sqrt{kz_{k+1}^2 - b_k z_{k+1}}}, \frac{1}{\sqrt{kz_k^2 - b_k z_k}} \right]$

While the above Theorem gives the form of the solution for a known k, the latter cannot be known a priori, as it is also a function of z. For this, we propose Algorithm 1 that solves this problem, simultaneously finding k and λ^* in $\mathcal{O}(k)$ iterations.

Algorithm 1 Solver of the one-node problem, (8).

Input: $z \in \mathbb{R}^n_{*+}$ in ascending order, $\theta \in \mathbb{R}_{*+}$
$b_0 \leftarrow 0$ {Initialize cumulative sum}
for $i = 1, \ldots, n$ **do**
  $b_i \leftarrow b_{i-1} + z_i$ {Cumulative sum of z}
  $\lambda_i \leftarrow \frac{\sqrt{\theta^2 b_i^2 + 4i} + \theta b_i}{2i}$
  if $\lambda_i > \theta z_i$ **then**
    $k \leftarrow i - 1$
    $\lambda^* \leftarrow \lambda_k$
    $w^* \leftarrow \max\{0, \lambda^* - \theta z\}$ {k-sparse output}
    break
  end if
end for

Figure 1: Algorithm 1

Overall Theoretical Time Complexity

- Step1:The time complexity to find $\tilde{z} = z\left(\mathcal{E}^{\text{allowed}}\right)$ using A-NN is $O(n\log(n)d)$ for d-dimensional data.
- Step2: We can also find θ for a given Sparsity level k with time complexity of $O(k)$.
- Step3: We can Learn edge weights using Primal-Dual Techniques for k desired number of neighbours and for r a small multiplicative factor with a time complexity of $O(nkrl)$ for l iterations.
- So for Large n, the dominating cost is asymptotically the one of computing the A-NN which is $O(nd\log(n))$

Results

Running Time:

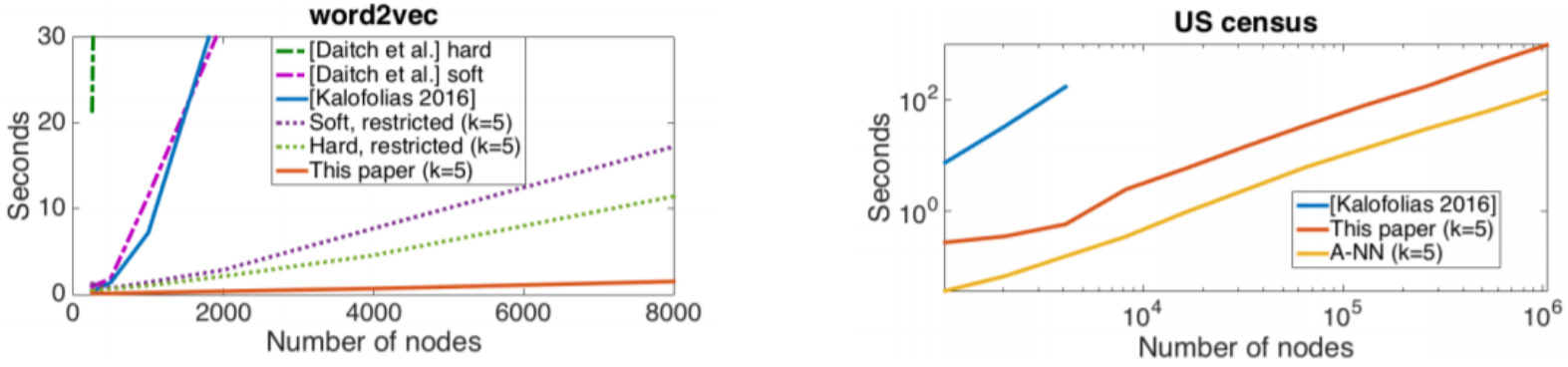


Figure 1: Time comparison of different ways to compute a graph. **Left:** Graph between 10,000 most frequent English words using a word2vec representation. **Right:** Graph between 1,000,000 nodes from 68 features (US Census 1990). Scalable algorithms benefit from a small average node degree k.

Results

Accuracy:

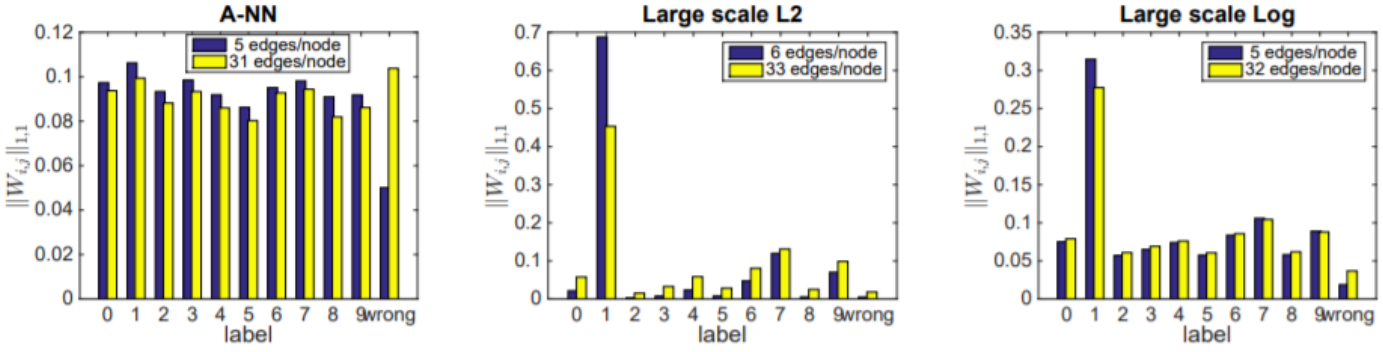


Figure 5: Connectivity across classes of MNIST. **Left**: A-NN graph. **Middle**: ℓ_2 model (4) neglects digits with larger distance. **Right**: log model (5) does not neglect to connect any cluster.

Conclusions

- This model costs roughly as much as A-NN, it achieves quality very close to state-of-the-art.
- Its ability to scale is based on reducing the variables used for learning, while out automatic parameter selection eliminates the need for grid-search in order to achieve a sought graph sparsity.
- Learning a graph of 1 million nodes only takes 16 minutes using simple Matlab implementation on a desktop computer.

References

Research papers [1, 2]

Vassilis Kalofolias.

How to learn a graph from smooth signals.

In *Artificial Intelligence and Statistics*, pages 920–929, 2016.

Vassilis Kalofolias and Nathanaël Perraudin.

Large scale graph learning from smooth signals.

arXiv preprint arXiv:1710.05654, 2017.

[3] Xiaowen Dong: Learning graphs from data: A signal processing perspective