Sparse Network Inference using the k-Support Norm

Position Paper

Aman Gupta[*]
Language Technologies Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh
amang@cs.cmu.edu

Haohan Wang[*]
Language Technologies Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh
haohanw@cs.cmu.edu

Rama Kumar Pasumarthi[*]
Language Technologies Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh
rpasumar@cs.cmu.edu

ABSTRACT

Network inference is an important problem in a variety of domains. In computational biology, gene interaction networks can be learned using the mRNA expression levels of genes. These networks capture how genes influence each other and can be used to identify potential malfunctions. In the context of social network analysis, network inference refers to the problem of inferring underlying network of influence, given time series data of when users performed certain actions (e.g., post, retweet, share). These networks capture the dynamics of influence and information diffusion.

In this position paper, we discuss various strategies to learn sparse networks with the help of the k-support norm, which corresponds to the tightest convex relaxation of sparsity combined with an l2 penalty. We also discuss specific applications of these strategies to the domains of computational biology and social network analysis.

Keywords

network inference, sparsity, K-support norm

1. INTRODUCTION

Inference of complex networks from data is a vital problem in a variety of domains like computational biology, social network analysis, operations research etc. With an explosion in the availability of data, the need of the hour is algorithms adept at modeling complex networks.

In the domain of biology, an important goal is understanding the regulation of various cellular processes and their responses to stimuli. Genes and proteins are at the core of all cellular processes. Genes produce messenger RNA during a process known as transcription, and mRNA in turn is processed into proteins. Proteins synthesized by genes may act as transcription factors for the synthesis of mRNA from other genes, or may be used in other important cellular activities. A set of genes can either promote or suppress the production of mRNA from a target gene. These interactions can be visualized as complex gene regulatory networks (GRNs). Inference of these networks then becomes crucial to understanding the underpinnings of cellular processes. This insight can be used to analyze altered gene expression, a result of conditions like cancer, for potential drug targets.

In social networks, information diffusion is often modelled as a stochastic process that occurs over an underlying network of influence. The diffusion of a single piece of *contagion* (a news snippet/meme/hashtag) occurs in a cascade like fashion. What is usually observed is the time series data of when users performed certain actions (also referred to as when a user is *infected*), for each contagion, but the underlying network is unobserved, or partially observed, where we have information such as who follows whom [11]. Inferring this underlying network of influence, a directed graph where the pairwise edges indicate strength of influence, is essential to understand the dynamics of information diffusion, and to solve problems such as influence maximization [6], design of viral marketing campaigns, or to stem the flow of malicious information.

Networks in both of the domains described above, although complex, involve a small number of interacting entities and can be represented using sparse models [5]. The lasso [14] technique has proved very popular in learning parsimonious models. However, the lasso, despite acting as a good surrogate for cardinality, is not very good at controlling the magnitude of parameters and is known at times to push too many variables down at zero. Thus, many problems require parameter re-estimation once lasso has been used to establish the support of the parameters. Also, in case of correlated variables, Lasso can sometimes arbitrarily select one out of a group of correlated variables. This can be problematic if the parameters are used to perform a predictive task like regression. In recent years, techniques like the elastic net [17] have been proposed to provide a balance between sparsity and the Euclidean norm of the parameters.

In this position paper, we discuss using the k-support norm for learning sparse networks. The k-support norm is tighter than the elastic net and provides a better convex relaxation for a sparse, low l-2 norm parameter vector. Since the k-support norm is not differentiable, we discuss fast proximal gradient descent algorithms for optimization. We also discuss various approaches for incorporating constraints on network learning during optimization.

[*] These authors contributed equally

DOI: 10.1145/1235

2. BACKGROUND AND RELATED WORK

In computational biology, the low connectivity property [13] of biological networks and sparsity of gene regulatory networks have introduced challenges in employing sparsity prior knowledge. This sparsity prior knowledge has been used as explicit constraints on the connectivity of network components [4]. Other works adopt L1-norm [14] regularization to build sparse networks. For example, [10] uses sparse regression with an L1-penalty induced for selecting the non-zero partial correlations and discover an undirected network encoding direct relations between genes. [3] infers gene regulatory network with the L1 norm method based on the auto-regressive model. [16, 15] propose a linear program that fits the data and satisfies the sparse structure with weighted L1 relaxation as the cost function, with additional linear constraints. Recently, Zavlanos et al. [16, 15] have shown that the inference performance is significantly improved by explicitly imposing the stability condition on the network.

In social network inference, survival theory has been used to model information propagation, where the instantaneous rate of infection of a user (*hazard rate*) depends on the infection times of previously infected users, as explanatory variables or covariates. The hazard rate has been modeled using parametric distributions such as Weibull and power law, or using non-parametric methods. The objective function for network inference has been shown to be convex in the space of pairwise influence parameters. Since these networks are in general sparse, Daneshmand *et al* [2] have used an L1 norm penalty to promote sparsity in the solution.

There is ample evidence for the importance of learning parsimonious models for both gene and social networks. In the following subsections, we discuss two popular models for inferring sparse gene and social networks.

2.1 Modeling gene networks as Linear Dynamical Systems

Gene regulatory networks of a set of p genes can be modeled as $p-$dimensional non-linear dynamical systems:

$$\frac{d\mathbf{x}}{dt} = f(\mathbf{x}, u) \tag{1}$$

Where **x** represents gene expression concentrations for genes and **u** represents perturbations applied to each gene. Both **x** and **u** are p-dimensional vectors. This is an effective model because the expression of a gene depends on the current gene expression of itself and its neighbours. **u** represents a systematic perturbation applied to a gene expression system at equilibrium.

Non-linear dynamical systems reach equilibrium when the rate of change of gene expression for all genes becomes zero, i.e. all genes attain steady-state values of mRNA concentrations.

Under small-perturbations on a steady-state system, these non-linear systems can be approximated to a first-order model [4]. The change in concentrations can be approximated using the following linear model:

$$\frac{d\mathbf{x'}}{dt} = \mathbf{Ax + u} \tag{2}$$

x' is the change in concentration of mRNA for all genes from previous equilibrium. The matrix $\mathbf{A}$ is $n \times n$ dimensional and encodes pairwise relationships between genes. Specifically, $\mathbf{A}_{ji}$ represents the influence of gene j on gene i. The vector **u** represents perturbations applied to all genes. The problem is then reduced to learning a sparse and consistent matrix A.

At steady-state, the system reaches equilibrium and the rate of change of gene expression is almost zero. Moreover, multiple perturbation experiments can be carried out to help make better inference. If there are m such experiments, then the steady-state matrix form of equation 2 is:

$$0 \approx \mathbf{AX + U} \tag{3}$$

X and **U** are $p \times m$ matrices. This model was introduced by [4]. They used the assumption that $m < n$, and imposed restrictions on the number of connections each gene can have to solve for an under-determined system by using multiple linear regressions.

A simple objective function for learning sparse **A** is:

$$\begin{gathered} \min t||\mathbf{A}||_1 + (1-t)\epsilon \\ \text{subject to} \\ ||\mathbf{AX} + \mathbf{U}||_1 <= \epsilon, \epsilon > 0 \end{gathered} \tag{4}$$

where t is a factor that controls the relative importance of sparsity and ϵ is the desired error threshold. This objective can be minimized using interior point methods.

2.2 Objective Formulation for Social Network Inference

We use the objective formulation in [2] for network inference. Let there be N nodes in the network. Let $\mathbf{A}$ be the matrix of influence parameters for each pair of nodes. Let $\mathbf{A_i} = \{\mathbf{A_{ji}}, 1 \leq j \leq N, j \neq i\}$. Note that $\mathbf{A}$ captures the underlying graph of influence, where $\mathbf{A_{ji}} = 0$ implies that a directed edge does not exist from j to i.
The observed data comprises a set of cascades $\mathbf{t} = \{\mathbf{t^1}, \mathbf{t^2}, .., \mathbf{t^n}\}$, where n cascades are observed from time 0 to T . Each cascade comprises of a time series of infection timestamps of nodes, with the time ∞ assigned to a node which is not observed in the cascade.

2.2.1 Data Generative Process

We use the continuous time diffusion model proposed in [11]. The process starts with a source node being infected at time 0 . Every node that is infected transmits the contagion via outgoing edges to other nodes. The infection timestamp for other nodes is drawn from a *transmission function* $f(t_i|t_j, \mathbf{A_{ji}})$, where the infection spreads from j to i. When multiple nodes propose a timestamp for an uninfected node, it chooses the earliest proposed timestamp.

2.2.2 Objective Formulation

In [2], it is shown that the objective function decomposes into a convex formulation per node, in the following form:

$$\begin{gathered} \min l_i(\mathbf{A_i}) + \lambda||\mathbf{A_i}||_1, \\ s.t. \\ \mathbf{A_{ji}} \geq 0, 1 \leq j \leq N, j \neq i \end{gathered}$$

where $l_i(\mathbf{A_i})$ corresponds to the negative log-likelihood of observing the infection times corresponding to node i.

The log-likelihood function comprises of terms involving the transmission function for pair of observed infection timestamps between nodes j and i : $f(t_i|t_j, \mathbf{A_{ji}})$. Whenever an infection of i is observed after j in a cascade, the log-likelihood has a positive term involving $f(t_i|t_j, \mathbf{A_{ji}})$, and whenever infection of i is not observed after j in a cascade, the log-likelihood has a negative term involving $f(t_i|t_j, \mathbf{A_{ji}})$.

3. METHODS

3.1 K - support norm

Networks in biology - gene, protein or of other kinds - are generally sparse in nature [5]. It is well-understood that only a handful of genes affect the expression of a particular gene. So is the case for social networks, where one person can affect only a handful of people directly. Learning parsimonious models thus becomes a necessity. A strategy to learn these networks is to use a sparsity regularizer such as the lasso [14]. In this work, we propose the use of the k-support norm[1] to ensure sparsity of the learned network.

The k-support norm has been previously used for classification [1]. Its superiority to Lasso has been shown in different settings [12],[8]. It has also been extended to apply to matrices and not just vectors [7]. To the best of our knowledge, this is the first time it is being used for gene regulatory network inference.

3.1.1 Definition

Consider a general vector w, which could be a regression vector. The k-support norm is the gauge function associated to the set $conv\{w \| \|w\|_0 \leq k, \|w\|_2 \leq 1\}$. It can be computed as

$$\|w\|_k^{sp} = \left(\sum_{i=1}^{k-r-1} (|w|_i^{\downarrow})^2 + \frac{1}{r+1} \Big(\sum_{i=k-r}^{d} |w|_i^{\downarrow} \Big)^2 \right)^{\frac{1}{2}}$$

with $|w|_i^{\downarrow}$ the i^{th} largest element of vector w and r is the unique integer in the set {1,...,k-1} such that

$$|w|_{k-r-1}^{\downarrow} > \frac{1}{r+1} \sum_{i=k-r}^{d} |w|_i^{\downarrow} \geq |w|_{k-r}^{\downarrow}$$

3.1.2 Relationship to lasso and ridge penalty

Also, in case of correlated variables, Lasso can sometimes arbitrarily select one out of a group of correlated variables. This can be problematic if the parameters are used to perform a predictive task like regression.

We observe that in the $k = 1$ case, the k-support norm is actually the same as the l_1 norm. When $k = d$ and $w \in R^d$, the k-support norm is equivalent to the l_2 norm. While the Lasso leads to sparse solutions, it doesn't capture group structure and randomly selects one variable among a group correlated variables. It also tends to push down a lot of variables to zero or values close to zero, requiring parameter re-estimation once the support has been found. By tuning the parameter k, we can choose the cardinality of the solution and therefore choose to keep groups of correlated variables. Hence, less sparse, but with more predictive power solutions can be chosen.

As observed by the authors of [12], for an objective $\min_w \lambda \|w\|_k^{sp} + f(w, X, y)$, when $k = d$, this minimization problem is equivalent to $\min_w \lambda \|w\|_2 + f(w, X, y)$. This problem differs from the traditional l_2 regularized one. However, by noting that this objective is the Lagrangian of the constrained minimization problem that minimizes f subject to $\|w\|_2 \leq B$ and that this constraint is equivalent to $\|w\|_2 \leq B^2$, we have, for any constant λ the existence of a constant $\tilde{\lambda}$ such that:

$argmin_w \lambda \|w\|_2 + f(w, X, y) = argmin_w \tilde{\lambda} \|w\|_2^2 + f(w, X, y)$ which is the usual l_2 regularizer.

3.1.3 Relationship to the elastic net

One common setting in microarray studies is to have high dimensional data with few examples. In this situation, the Lasso saturates as it can select a number of variables at most equal to the number of features. To address this limitation, [17] introduced the elastic net, which linearly combines the l_1 and the l_2 regularizations:

$$argmin_w \frac{1}{2} \|Xw - y\|_2 + \lambda_1 \|w\|_1 + \lambda_2 \|w\|_2^2 \quad (5)$$

Algorithm 1 Computation of the proximal operator

```
Input: $v \in R^d$
Output: $q = prox_{\frac{1}{2\beta}(\|.\|_k^{sp})^2}(v)$
Find $r \in \{0, ..., k-1\}$, $l \in \{k, ..., d\}$ such that
    $\frac{1}{\beta+1} z_{k-r-1} > \frac{T_{r,l}}{l-k+(\beta+1)r+\beta+1} \geq \frac{1}{\beta+1} z_{k-r}$    (6)
    $z_l > \frac{T_{r,l}}{l-k+(\beta+1)r+\beta+1)} \geq z_{l+1}$    (7)
where $z := |v|^{\downarrow}$, $z_0 := +\infty$, $z_{d+1} := -\infty$, $T_{r,l} := \sum_{i=k-r}^{l} z_i$
$q_i \leftarrow \begin{cases} \frac{\beta}{\beta+1} & \text{if } i = 1, ..., k-r-1 \\ z_i - \frac{T_{r,l}}{l-k+(\beta+1)r+\beta+1} & \text{if } i = k-r, ..., l \\ 0 & \text{if } i = l+1, ..., d \end{cases}$
reorder and change signs of q to conform with v
```

Like k-support, the elastic net interpolates between the l_1 and the l_2 norms. [1] show that the k-support norm is tighter than the elastic net by a factor of at most$\sqrt{2}$

3.2 Optimization

Optimization using the k-support norm can be performed using the negative log-likelihood and a regularization term. Since the k-support norm is not differentiable, we resort to proximal algorithms. The smooth part of the cost function is the negative log-likelihood and is convex and differentiate. The formulation of these two terms naturally suits proximal gradient descent, which we use to solve the problem [9]. Computation of the proximal operator is described in Algorithm 1, and the overall optimization algorithm is described in Algorithm 2.

The proximal operator can be computed in $O(d(k + \log k))$ steps which is given in the algorithm 1 of [1], indicated in Algorithm 1 although faster algorithms have been developed in the last two years.

Other constraints like maintaining positive-definiteness of the network can be handled using projection methods or eigenvalue thresholding.

Algorithm 2 Iterative soft-thresholding for parameter learning

```
Input: Data and initial parameter $w^0 \in R^d$
Iterate: For $t = 0, 1, 2...$ until convergence of $w$:
    1. Compute gradient of negative log-likelihood
    2. Solve for w : $w^t = Prox_{\lambda s_t}(w^{t-1} - s^t grad)$ where
       $Prox_{\lambda s_t}$ is the proximal operator described in Algorithm 1.
Output: $w$
```

4. ACKNOWLEDGMENTS

AG thanks Hakim Sidahmed for accommodating numerous discussions about the k-support norm.

References

[1] A. Argyriou, R. Foygel, and N. Srebro. Sparse prediction with the k-support norm. In *Advances in Neural Information Processing Systems*, pages 1457–1465, 2012.

[2] H. Daneshmand, M. Gomez-Rodriguez, L. Song, and B. Schölkopf. Estimating diffusion network structures: Recovery conditions, sample complexity & soft-thresholding algorithm. In *ICML '14: Proceedings of the 31th International Conference on Machine Learning*, 2014.

[3] A. Fujita, J. R. Sato, H. M. Garay-Malpartida, R. Yamaguchi, S. Miyano, M. C. Sogayar, and C. E. Ferreira. Modeling gene expression regulatory networks with the sparse vector autoregressive model. *BMC Systems Biology*, 1(1):39, 2007.

[4] T. S. Gardner, D. Di Bernardo, D. Lorenz, and J. J. Collins. Inferring genetic networks and identifying compound mode of action via expression profiling. *Science*, 301(5629):102–105, 2003.

[5] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, and C. D. Bloomfield. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, 1999.

[6] M. Gomez-Rodriguez, L. Song, N. Du, H. Zha, and B. Schoelkopf. Influence estimation and maximization in continuous-time diffusion networks. *ACM Transactions on Information Systems (TOIS)*, 2016.

[7] A. M. McDonald, M. Pontil, and D. Stamos. Spectral k-support norm regularization. In *Advances in Neural Information Processing Systems*, pages 3644–3652, 2014.

[8] M. Misyrlis, A. Konova, M. Blaschko, J. Honorio, N. Alia-Klein, R. Goldstein, and D. Samaras. Predicting cross-task behavioral variables from fmri data using the k-support norm. In *Sparsity Techniques in Medical Imaging (STMI)*, 2014.

[9] N. Parikh and S. P. Boyd. Proximal algorithms. *Foundations and Trends in optimization*, 1(3):127–239, 2014.

[10] J. Peng, P. Wang, N. Zhou, and J. Zhu. Partial correlation estimation by joint sparse regression models. *Journal of the American Statistical Association*, 2012.

[11] M. G. Rodriguez, D. Balduzzi, and B. Schölkopf. Uncovering the temporal dynamics of diffusion networks. *arXiv.org*, 2011.

[12] H. Sidahmed, E. Prokofyeva, and M. B. Blaschko. Discovering predictors of mental health service utilization with k-support regularized logistic regression. *Information Sciences*, 2015.

[13] D. Thieffry, A. M. Huerta, E. Pérez-Rueda, and J. Collado-Vides. From specific gene regulation to genomic networks: a global analysis of transcriptional regulation in escherichia coli. *Bioessays*, 20(5):433–440, 1998.

[14] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[15] M. M. Zavlanos, A. A. Julius, S. P. Boyd, and G. J. Pappas. Identification of stable genetic networks using convex programming. In *American Control Conference, 2008*, pages 2755–2760. IEEE, 2008.

[16] M. M. Zavlanos, A. A. Julius, S. P. Boyd, and G. J. Pappas. Inferring stable genetic networks from steady-state data. *Automatica*, 47(6):1113–1122, 2011.

[17] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.