Differential Parameter Learning

Adarsh Prasad

Abstract

Background. Graphical models have gained significant attention as a tool for discovering and visualizing dependencies among variables in multivariate data. Recently, attention has been drawn to situations where domain experts are interested in differences between the dependency structures of different populations. (e.g. differences between regulatory networks of different species, or differences between dependency networks of diseased versus healthy populations). The standard method for recovering these differences is to learn the structures independently and then compare them naïvely. However, this method has achieved limited success in practice as it is prone to high false discovery rates.

Aim. In this paper, we develop sample efficient estimators which learn the differential network with low false discovery rates.

Data. We evaluate our proposed estimators on real-data where, given samples Escherichia Coli regulatory network under different conditions, the goal is to recover the change in the network.

Methods. Our proposed estimators can be broadly classified into two categories; discriminative methods, which directly model the differential structure, and generative methods, which learn the structures independently, and control for false discovery rates by an additional step.

Results. We show empirically that our proposed methods achieve higher precision than existing techniques.

Conclusions. While in this paper we have focused on differential network analysis, the idea of differential learning can be generalized to learning semi-parametric models, where parameters of interest are plagued with nuisance parameters. Similar techniques could be used for differential feature selection to answer questions like "are there different bio-markers for cancer in men and women?", or for differential clustering to detect significant changes in cluster structures between populations.

Keywords: Graphical Models, ℓ_1-regularized learning, sparsity

DAP Committee Members:

Pradeep Ravikumar (MLD);
Sivaraman Balakrishnan (Statistics)

1 Introduction

Graphical Models or Markov Networks are a popular tool for inference, visualization, modeling and exploratory analysis of wide-ranging applications. Two popular classes of Markov Networks are the Gaussian Graphical Model, for continuous (Gaussian) variables, and the Ising / Potts model, for binary / categorical variables.

A more recent use of graphical models is to identify changes in dependencies for different populations or conditions. For example, traditionally, functional magnetic resonance imaging(fMRI) has been used to model the dependencies between different regions of the brain as it measures the activity level in different regions of the brain and is used to indicates regions which seem to be exchanging information. In such a setting, domain experts also want to understand how regions of the brain share information before and after a person acquires a particular skill. Answers to such questions help in identifying the regions of the brain that are most influential after a skill is learned, and hence, direct current stimulation can be applied to those regions to accelerate a person's learning process.

Similarly for oncology studies, a critical problem is to analyze how the dependency structure of plasma proteins changes between healthy patients and patients that have cancer. Identifying the changes helps with the goal of understanding the cancer biology and coming up with better diagnostics.

In such differential dependency network analysis problems, traditional methods based on learning the dependency network for each condition independently and then comparing them tend to produce a large number of spurious differences [17]. This hampers the analysis and prevents drawing any reliable conclusions, limiting the its usefulness significantly.

One way to control spurious differences is to learn the dependency networks for the different conditions jointly by imposing a bias that the learned networks be similar. The more heavily this bias is enforced, the fewer differences will be learned between networks. Algorithmically, such approaches are similar to transfer learning and multi-task learning where a number of algorithms for joint learning of dependency networks have been proposed and studied. One could use these algorithms for the task of providing a reliable differential analysis.

However, despite using the same algorithms, the fundamental goal of multi-task learning and differential analysis problems is different. Firstly, in multi-task learning, the goal is to recover the individual structures accurately, while in differential analysis the goal is to reliably identify differences between the dependency networks. In differential analysis, shared components of networks are essentially nuisance parameters, and one can sacrifice accuracy on them, to improve the recovery of differential parameters. Secondly, multi-task learning have poorer performance in cases when the networks are dissimilar, whereas, from differential analysis, network dissimilarity should make recovery easy.

This brings the following question to the front. *Is there a way to efficiently estimate the differential network parameters?*

Contributions.

2 Problem Statement

In this paper, we focus on the problem of estimating changes in dependency networks of two different populations: given samples from each. In particular, we focus on the case where the differential network has structure, such as sparsity or low/high degree. Note that individual networks may or may not have such structure, and the goal is to completely characterize the estimation of the differential network on such parameters. Our goal is to present results which answer questions like: can one design estimators where the number of samples required to estimate the differential network depends only on the degree of the differential network?

3 Background and Setup.

Exponential Family. The exponential family is a widely used family of distributions which are parametrized by a finite dimensional *canonical parameter*. The density function is given by:

$$\mathbb{P}_\theta(X) = h(X)\exp(\langle\theta, \phi(X)\rangle - A(\theta))$$

Here, θ^* is the vector of the true canonical parameters, $A(\theta)$ is the log-partition function and $\phi(X)$ is the sufficient statistic. Most of the standard discrete and continuous distributions used for structure modeling, such as the Ising Models, Gaussian MRFs *etc*. are special cases of the exponential family.

In this paper, we consider the problem of estimating changes in the canonical parameter, *i.e*., given two sets of samples $\mathcal{X}_1^{n_1} = \{x_i^1\}_{i=1}^{n_1}$ and $\mathcal{X}_2^{n_2} = \{x_i^2\}_{i=1}^{n_2}$, drawn from different populations(exponential families) with canonical parameters $\theta_1^* \in \mathbb{R}^p$ and $\theta_2^* \in \mathbb{R}^p$, the goal is to estimate $\delta\theta = (\theta_1^* - \theta_2^*)$. Additionally, either in a high-dimensional setting where $n_1, n_2 << p$, or using priors from domain knowledge, one may impose and exploit additional assumptions on the parameter $\delta\theta$ such as sparsity, block-sparsity. Note that the individual population parameters θ_1^* and θ_2^* may or may not have have any specific structure. In this paper, we consider Gaussian Graphical Models as our running example, although, as seen later, our results apply to all exponential families.

Gaussian Graphical Models. Let $X = (X_1, \ldots, X_p)$ denote a zero-mean gaussian random vector; it's density is fully-parametrized as by the inverse covariance or concentration matrix $\Theta = (\Sigma)^{-1} \succ 0$ and can be written as:

$$\mathbb{P}_\Theta(x) = \frac{1}{\sqrt{(2\pi)^p \det\left((\Theta)^{-1}\right)}} \exp\left(-\frac{1}{2}x^{\mathrm{T}}\Theta x\right) \tag{1}$$

Suppose that the variables $(X1, \ldots, X_p)$ are associated with the vertex set $V = \{1, 2, \ldots, p\}$ of an undirected graph $G = (V, E)$. We say that the concentration matrix Θ^* respects the edge structure of the graph if $\Theta_{ij}^* = 0$ for all $(i, j) \notin E$. The family of Gaussian distributions with this property is known as a Gauss-Markov random field with respect to the graph G.

4 Related Work.

In this section, we discuss the related work by dividing it into 3 broad categories.

Maximum likelihood estimation. Maximum likelihood estimation (MLE) is a commonly used parameter-estimation technique, where one learns(fits) by maximizing the likelihood(or probability) of seeing the given data. Maximum likelihood estimation (MLE) based estimators with ℓ_1-regularization has been widely used for estimating the precision matrix in the Gaussian case [13]. Let $\{X^i\}_{i=1}^n \in \mathbb{R}^p$ be drawn from a multivariate gaussian $X^i \sim \mathcal{N}(0, (\Theta^*)^{-1})$, then the MLE estimate($\widehat{\Omega}$) of Θ^* is given as:

$$\begin{aligned}\widehat{\Omega} &= \operatorname*{argmin}_{\Theta\succ 0} -\frac{1}{n}\sum_{i=1}^{n}\log(\mathbb{P}(X^{(i)}|\Theta)) + \lambda_n||\Theta||_{1,\text{off}} \\ &= \operatorname*{argmin}_{\Theta\succ 0} \text{tr}(\Theta, \widehat{\Sigma}) - \text{logdet}(\Theta)) + \lambda_n||\Theta||_{1,\text{off}}\end{aligned} \tag{2}$$

where $\widehat{\Sigma}$ is the sample covariance matrix. For sake of clarity, let $\ell_{\widehat{\Sigma}}(\Theta) = \text{tr}(\Theta, \widehat{\Sigma}) - \text{logdet}(\Theta))$.

4.1 Independent Learning.

The problem of recovering the structural differences between two populations has been of interest for several years, especially in bioinformatics. de la Fuente [4] provides a comprehensive survey of the techniques proposed in bioinformatics, in which the author claims that nearly all approaches fall in a *learn-then-compare* paradigm, where each graph $(\widehat{\Theta}_1, \widehat{\Theta}_2)$ is estimated individually disregarding any overlapping structure. *i.e.* $\widehat{\Theta}_1$ is just the MLE estimate for Θ_1^* obtained by solving the optimization problem in Equation 2. Similarly, $\widehat{\Theta}_2$ is the MLE estimate for Θ_2^*. Then, the differential structure is estimated as the the difference of the two MLE estimates. $\widehat{\Theta}_{\text{diff}} = \widehat{\Theta}_1 - \widehat{\Theta}_2$.

Optimization. Hsieh et al. [8] proposed a novel block coordinate descent algorithm for optimizing the MLE(Equation 2) by carefully exploiting the underlying structure of the problem. Specifically, the blocks to optimize are chosen via a clustering scheme to minimize repeated computations; and allowing for inexact computation of specific components.

Guarantees. The independent learning method involves estimating the individual structures completely. Hence, the number of samples required to estimate the differential structure is equivalent to the number of samples required to learn each individual network. Ravikumar et al. [13] showed that the number of samples required to learn a gaussian graphical model by solving the MLE scales as the $n = \Omega(d^2 \log p)$, where d is the degree of the original graph, and p is the number of vertices. Using permutation testing Zhang et al. [17] observed that such two-step procedures have high false discovery rate. High false discovery rates are observed because the two-step procedure suffers from the conceptual weakness of the structure change not being learnt directly.

4.2 Multi-task Learning.

One way to mitigate this indirect nature is to observe that the assumption of sparse differences is equivalent to having a large shared support between different graphs. Under this assumption, one can jointly learn the structures by explicitly encoding similarity of the graphs into the objective function and jointly learn the two networks. Such techniques have been well-studied in past under the umbrella of Multitask Learning[16, 3, 7]. Recently, Belilovsky et al. [2] used debiased versions of such multitask algorithms for linear regression to obtain confidence intervals on edge differences in GGMs.

4.2.1 ℓ_1/ℓ_∞ Regularization

We have n samples per population $Y = 1$ and $Y = 2$. Using these samples one can estimate each empirical covariance matrix, $\widehat{\Sigma}_1$ and $\widehat{\Sigma}_2$. Thus we have a collection of $K = 2$ graphical model learning problems. Under the assumption that $\Theta^*_{\text{diff}} = \Theta_1^* - \Theta_2^*$ is sparse, *i.e.* the two networks differ only over a few edges, then one can set-up the learning problem as joint-estimation problem with an added regularizer which encourages *sharing* of edges between the networks. This set-up suggests the use of ℓ_1/ℓ_∞ block-regularizer for the joint estimation of $\Theta = (\Theta_1, \Theta_2) \in \mathbb{R}^{p\times p\times 2}$. The block regularizer encourages an edge $(i, j) \in \mathcal{E}$ to be either present in all graphs or be absent in all. This leads to the following optimization problem:

$$\widehat{\Theta} = (\widehat{\Theta}_1, \widehat{\Theta}_2) = \underset{\Theta_1 \succ 0, \Theta_2 \succ 0}{\operatorname{argmin}} \ell_{\widehat{\Sigma}_1}(\Theta_1) + \ell_{\widehat{\Sigma}_1}(\Theta_1) + \lambda_n \sum_{\substack{i,j\in[p] \\ i\neq j}} || \left(\Theta_{1_{ij}}, \Theta_{2_{ij}}\right) ||_\infty \tag{3}$$

Note that we do not penalize the diagonals.

Optimization. Honorio and Samaras [7] proposed a block coordinate descent algorithm for optimizing Equation 3 by drawing connections between the multi-task structure learning problem and the continuous

quadratic knapsack problem, for which very efficient methods exist. They prove a $O(np^2 + Lp^3)$ runtime for L iterations of their algorithm.

Guarantees. Block-regularization based multi-task algorithms have been studied in the context of Linear Regression where the goal is to jointly learn the slopes for different populations[12]. It was shown that ℓ_1/ℓ_∞-regularization is more sample efficient than independent learning in learning the support of the regressors only when the overlap between supports was greater than $\frac{2}{3}$. We empirically show that a similar behavior is observed for (3)(See Appendix **??**.

Multi-task algorithms suffer from two issues: (1) These approaches do not work if each network is dense but only the change is sparse. (2) When the networks are dissimilar, such joint algorithms. Most existing network methods assume that the individual networks are sparse, whereas gene networks often contain hub nodes[1]. Hence, learning individual networks often require more samples than what would be needed for learning just the differences.

4.3 Direct Learning.

A complementary approach is to construct suitable loss functions which solely depend on the differential parameters. For example, Zhao et al. [18] proposed a loss function for estimating direct sparse changes in Gaussian graphical models (GGMs). However, their estimator is specific to GGMs and can not be applied to say Ising models. Liu et al. [9] observed that the ratio of distributions induced by GGMs is solely a function of the difference in network parameters. Utilizing this observation, they proposed *density-ratio* based estimators to **directly** estimate the differential parameters. They provided non-asymptotic error bounds for the estimator along with sample complexity results for the case of sparse changes, *i.e.* $\delta\theta^* \in \mathbb{R}^{p^2}$ is sparse. Recently, Fazayeli and Banerjee [6] extended the density-ratio based approach to other differential structures *e.g.* block sparse, node-perturbed sparse *etc.* as long as the structure can be characterized by a suitable (atomic) norm. They analyzed a norm-regularized estimator for directly estimating the change in structure for Ising models, and provided ℓ_2 error guarantees. Since density-ratio is also a direct change estimation technique, we describe and discuss it in detail. Note that the material discussed here is based on [6] and [9], and we refer the reader to them for more details.

4.3.1 Density Ratio Estimation.

Consider two exponential family distributions $\mathbb{P}_{\theta_1^*}(\cdot)$ and $\mathbb{P}_{\theta_2^*}$, then the ratio of the two densities can be written as:

$$r(X = x|\delta\theta) = \frac{\mathbb{P}_{\theta_1}(x)}{\mathbb{P}_{\theta_2}(x)} = \underbrace{\frac{\exp\left(\langle\theta_1, \phi(x)\rangle\right)}{\exp\left(\langle\theta_2, \phi(x)\rangle\right)}}_{r^*(x|\delta\theta)} \underbrace{\frac{Z(\theta_2}{Z(\theta_1)}}_{1/Z(\delta\theta)} = \frac{\exp\left(\langle\phi(x), \delta\theta\rangle\right)}{Z(\delta\theta)} \tag{4}$$

where $Z(\theta) = \log(A(\theta))$ is the partition function, and $\delta\theta = \theta_1 - \theta_2$ is the difference parameter. Also, it can be shown that $Z(\delta\theta) = \mathbb{E}_{X\sim\mathbb{P}_{\theta_2}}\left[\exp\left(\langle\phi(X), \delta\theta\rangle\right)\right]$ [6]. Hence, one can use samples from $\mathbb{P}_{\theta_2^*}$ to empirically estimate $Z(\delta\theta)$ as:

$$\widehat{Z}(\delta\theta) = \frac{1}{n_2}\sum_{i=1}^{n_2} \exp\left(\left\langle\phi(x_i^{(2)}, \delta\theta\right\rangle\right) \tag{5}$$

$\widehat{\delta\theta}$ is obtained by minimizing the KL divergence between $\widehat{r(X|\delta\theta)}.\mathbb{P}_{\theta_2^*}(X)$ and $\mathbb{P}_{\theta_1^*}(X)$. The empirical

loss function can then be written as:

$$\mathcal{L}(\delta\theta; \theta, \mathcal{X}_1^{n_1}, \mathcal{X}_2^{n_2}) = \frac{-1}{n_1} \sum_{i=1}^{n_1} \left\langle \phi(x_i^1), \delta\theta \right\rangle + \log \frac{1}{n_2} \sum_{i=1}^{n_2} \exp\left(\left\langle \phi(x_i^2), \delta\theta \right\rangle \right) \tag{6}$$

Finally, one estimates $\widehat{\delta\theta}$ by optimizing a regularized version of (6) with the regularizer encoding prior beliefs on the change such as sparsity, block sparsity etc.

$$\delta\widehat{\theta} \in \operatorname*{argmin}_{\delta\theta} \mathcal{L}(\delta\theta; \theta, \mathcal{X}_1^{n_1}, \mathcal{X}_2^{n_2}) + \lambda_{n_1,n_2} \mathcal{R}(\delta\theta) \tag{7}$$

Optimization. The empirical loss function doesn't decompose over samples and hence one cannot use stochastic gradient based algorithms. The optimization problem (7) has a smooth convex part corresponding to the loss function and a potentially non-smooth convex part corresponding to the regularizer. Fazayeli and Banerjee [6] gave a fast-iterative shrinkage-thresholding algorithm (FISTA) for (7) and showed an $O(1/t^2)$ convergence rate.

Guarantees. Fazayeli and Banerjee [6] provide non-asymptotic results $\|\Delta\|_2 = \left\|\delta\theta^* - \widehat{\delta\theta}\right\|_2$ for different regularizers. For example, when the change $\delta\theta^*$ is sparse with $s = \|\delta\theta^*\|_0$, Fazayeli and Banerjee [6] proved that when $n_2 > s \log p$, then $\|\Delta\|_2 = O\left(\sqrt{\frac{s \log p}{\min(n_1,n_2)}}\right)$. Previously, Liu et al. [10] had established a sample complexity of $n_1 = \Omega(s^2 \log p)$, and $n_2 = \Omega(n_1^2)$ for establishing support recovery and ℓ_∞ bounds.

5 Methods.

In this section, we propose two new methods. Our first estimator improves upon the naïve learn-then-compare estimator by performing an additional thresholding step on the difference of the individual estimates. Our second estimator falls in the direct learning regime, and proposes a discriminative approach to differential learning based on supervised classification.

5.1 Generative Approach: Learn-and-Threshold.

Under suitable assumptions such as irrepresentability and a sample scaling of $n > d^2 \log(p)$, solutions to the regularized MLE (2) achieve a $\sqrt{\frac{\log p}{n}}$ bound on the ℓ_∞ error, when λ_n is set to be $c.\sqrt{\frac{\log p}{n}}$.

$$\left\|\widehat{\Theta}_i - \Theta_i^*\right\|_\infty \precsim \sqrt{\frac{\log p}{n}}$$

This means that the independent learning based empirical change estimate, $\delta\widehat{\theta}_{\text{indep}} = \widehat{\Theta}_1 - \widehat{\Theta}_2$ has an ℓ_∞ error of $O\left(\sqrt{\frac{\log p}{n}}\right)$. To reduce the false discoveries in the difference, we perform an additional soft thresholding step which is a solution to the following optimization problem.

$$\delta\widehat{\theta} = \operatorname*{argmin}_{\delta\theta} \left\|\delta\theta - \delta\widehat{\theta}_{\text{indep}}\right\|_2^2 + \lambda_2 \|\delta\|_1$$

	Net1	Net2	Net3	Net4	Net5	Net6
Maximum Degree (d)	9	9	9	9	9	9
Number of Edges (s)	79	79	81	82	83	83
Number of Nodes (p)	99	99	99	99	99	99
Differential Degree (d_{diff})	0	8	10	14	16	16
Differential Sparsity(s_{diff})	0	40	84	113	126	138

Table 1: Characteristics of Networks.

Guarantees. We know that $\delta\widehat{\theta}_{\text{indep}} = \delta\theta^* + \epsilon$, where $\|\epsilon\|_\infty \precsim \sqrt{\frac{\log p}{n}}$, so setting $\lambda_2 \geq 4\|\epsilon\|_\infty$, will output a $\widehat{\delta\theta}$ with no false positives *i.e.* we'll recover the true differential support. Moreover, assuming the $\|\delta\theta^*\|_0 = s$, our output will have an ℓ_2-error scaling with $\sqrt{\frac{s\log p}{n}}$.

5.2 Discriminative Approach: Logistic Regression.

We treat the differential parameter learning problem as a classification problem. Each distribution represents a labeled class. *i.e.* $\mathbb{P}_{\theta_1^*}$ corresponds to say label $Y = 1$ and $\mathbb{P}_{\theta_2^*}$ corresponds to say label $Y = 0$. Now, suppose the two distributions, $\mathbb{P}_{\theta_1^*}$ and $\mathbb{P}_{\theta_2^*}$ are equally likely, then, using Bayes rule, we have that:

$$\begin{aligned}\mathbb{P}[Y=1|X] &= \frac{\mathbb{P}[X|Y=1]\mathbb{P}[Y=1]}{\mathbb{P}[X|Y=0]\mathbb{P}[Y=0]+\mathbb{P}[X|Y=1]\mathbb{P}[Y=1]} \\ &= \frac{1}{1+\exp\left(\langle\theta_1^*-\theta_2^*, -\phi(x)\rangle + c^*\right)} \end{aligned} \tag{8}$$

where $c^* = A(\theta_1^*) - A(\theta_2^*)$. Observe that Equation 8 is *just* a function $\delta\theta^* = (\Theta_1^* - \Theta_2^*)$. Now, we can estimate the difference by solving the task of learning the discriminative model given these samples via a regularized conditional MLE. The conditional MLE is simply logistic regression with the sufficient statistics as the features. For example, in case of GGMs and sparse differences, one would optimize ℓ_1-regularized logistic regression with quadratic features.

$$\begin{aligned}(\widehat{\delta\theta} = (\widehat{\Theta}_1 - \widehat{\Theta}_2), \hat{c}) = \underset{\theta,c}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^{n} Y^{(i)}\left(\left\langle\left\langle\theta, X^{(i)}(X^{(i)})^T\right\rangle\right\rangle + c\right) \\ + \log(1+\exp\left(\left\langle\left\langle\theta, X^{(i)}(X^{(i)})^T\right\rangle\right\rangle + c\right) + \lambda_n||\theta||_{1,\text{off}}\end{aligned} \tag{9}$$

6 Experiments

Setup. Using data from Roy et al. [14], we want to analyze the changes in the Escherichia coli regulatory network under different conditions. Roy et al. [14] used a (sub)network of 99 nodes from the Escherichia coli regulatory network[15] as the base network.(*Net 1*). The authors generated five other networks by flipping $10, 30, 50, 70$ and 100% of the edges of the base network. Table 6 summarizes the properties of these networks. 1000 samples were generated per network by perturbing all transcription factor nodes and measuring the steady state of all genes[11]. In this case we have access to ground truth networks as well.

Baseline. We use the density ratio method as our baseline[1]. Fazayeli and Banerjee [6] showed the efficacy of the density ratio over independent learning of networks, hence, we omit the independent We

[1]Code provided by [10]

used LIBLINEAR[5] to implement ℓ_1 regularized logistic regression with quadratic features (9).

Metric. At a fixed sample size n, for both density ratio and logistic regression, we vary the regularization penalty $\lambda_n = c.\sqrt{\frac{\log p}{n}}$ and generate an ROC curve. We calculate the area under the curve (AUC) and use that as our metric. For any sample size, we repeat the experiment for 10 trials and report the mean-AUC along with the error bars. The higher the AUC, the better it is.

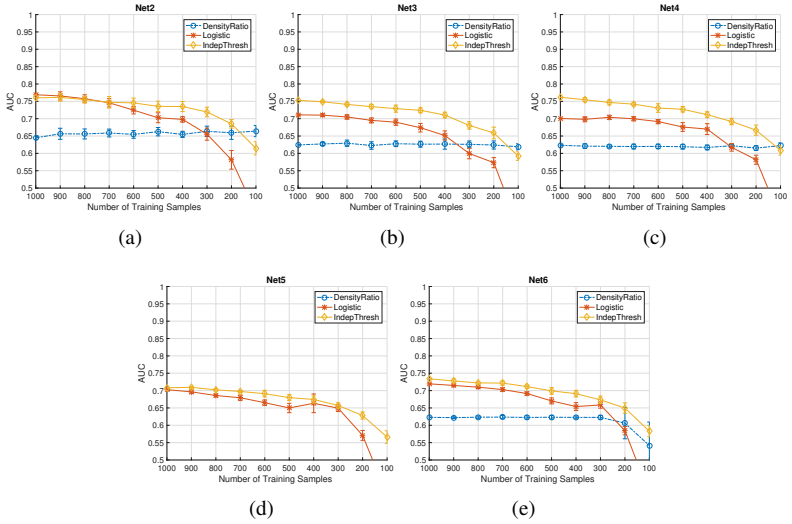

Figure 1: Performance comparison Logistic Regression and Density Ratio as a function of decreasing training data for six networks. Shown are mean-AUC score for inferred network structure from each partition size. Higher is better.

Results. Figure 6 shows that our proposed approach is competitive to Density Ratio. A surprising observation is that the so called "indirect method" of learning independently and then thresholding often works better than "direct methods". This can be attributed to the low degree of the original graphs which makes it easier to learn the individual graphs, but the results warrant a more in-depth analysis of the proposed methods.

For Logistic Regression, the performance gets worse as the number of differential edges increase, where as for Learn-and-Threshold, the performance depends on the learning of individual graphs.whilth

7 Conclusion.

In this work, we explored the problem of learning the difference between two networks. We proposed two new methods based on "direct" and "indirect learning". We evaluated our proposed estimators based on real-data, and found that our proposed methods perform better than the state-of-art performance.
In case of low-degree graphs, we propose a learn-and-threshold method, which after learning the graphs individually, performs a soft-thresholding to control the outliers. We find that this method performs better than the state of the art techniques.
In case of high-degree graphs, we propose a direct method, which uses ℓ_1 regularized logistic regression with quadratic features to compute the difference between the two networks. We compare it to density

ratio, and find that logistic regression is competitive. From a computational perspective, the objective for density ratio doesn't split over the samples, and hence requires special purpose solvers. On the other hand, optimizing ℓ_1 regularized logistic regression is a well studied problem and large scale instances can be easily optimized using stochastic gradient or parallel co-ordinate descent based techniques.

References

[1] Albert-Laszlo Barabasi and Zoltan N Oltvai. Network biology: understanding the cell's functional organization. *Nature reviews genetics*, 5(2):101–113, 2004.

[2] Eugene Belilovsky, Gaël Varoquaux, and Matthew B Blaschko. Testing for differences in gaussian graphical models: applications to brain connectivity. In *Advances in Neural Information Processing Systems*, pages 595–603, 2016.

[3] Patrick Danaher, Pei Wang, and Daniela M Witten. The joint graphical lasso for inverse covariance estimation across multiple classes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(2): 373–397, 2014.

[4] Alberto de la Fuente. From 'differential expression'to 'differential networking'–identification of dysfunctional regulatory networks in diseases. *Trends in genetics*, 26(7):326–333, 2010.

[5] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *Journal of machine learning research*, 9(Aug):1871–1874, 2008.

[6] Farideh Fazayeli and Arindam Banerjee. Generalized direct change estimation in ising model structure. *arXiv preprint arXiv:1606.05302*, 2016.

[7] Jean Honorio and Dimitris Samaras. Multi-task learning of gaussian graphical models. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 447–454, 2010.

[8] Cho-Jui Hsieh, Mátyás A Sustik, Inderjit S Dhillon, Pradeep K Ravikumar, and Russell Poldrack. Big & quic: Sparse inverse covariance estimation for a million variables. In *Advances in Neural Information Processing Systems*, pages 3165–3173, 2013.

[9] Song Liu, John A Quinn, Michael U Gutmann, Taiji Suzuki, and Masashi Sugiyama. Direct learning of sparse changes in markov networks by density ratio estimation. *Neural computation*, 26(6):1169–1197, 2014.

[10] Song Liu, Taiji Suzuki, and Masashi Sugiyama. Support consistency of direct sparse-change learning in markov networks. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pages 2785–2791. AAAI Press, 2015. ISBN 0-262-51129-0. URL `http://dl.acm.org/citation.cfm?id=2886521.2886709`.

[11] Pedro Mendes, Wei Sha, and Keying Ye. Artificial gene networks for objective comparison of analysis algorithms. *Bioinformatics*, 19(suppl 2):ii122–ii129, 2003.

[12] Sahand N Negahban and Martin J Wainwright. Simultaneous support recovery in high dimensions: Benefits and perils of block-regularization. *IEEE Transactions on Information Theory*, 57(6):3841–3863, 2011.

[13] Pradeep Ravikumar, Martin J Wainwright, Garvesh Raskutti, Bin Yu, et al. High-dimensional covariance estimation by minimizing ℓ_1-penalized log-determinant divergence. *Electronic Journal of Statistics*, 5:935–980, 2011.

[14] Sushmita Roy, Margaret Werner-Washburne, and Terran Lane. A multiple network learning approach to capture system-wide condition-specific responses. *Bioinformatics*, 27(13):1832–1838, 2011.

[15] Heladia Salgado, Socorro Gama-Castro, Martin Peralta-Gil, Edgar Díaz-Peredo, Fabiola Sánchez-Solano, Alberto Santos-Zavaleta, Irma Martínez-Flores, Verónica Jiménez-Jacinto, César Bonavides-Martínez, Juan Segura-Salazar, et al. Regulondb (version 5.0): Escherichia coli k-12 transcriptional regulatory network, operon organization, and growth conditions. *Nucleic acids research*, 34(suppl 1):D394–D397, 2006.

[16] Bai Zhang and Yue Wang. Learning structural changes of gaussian graphical models in controlled experiments. *arXiv preprint arXiv:1203.3532*, 2012.

[17] Bai Zhang, Huai Li, Rebecca B Riggins, Ming Zhan, Jianhua Xuan, Zhen Zhang, Eric P Hoffman, Robert Clarke, and Yue Wang. Differential dependency network analysis to identify condition-specific topological changes in biological networks. *Bioinformatics*, 25(4):526–532, 2009.

[18] Sihai Dave Zhao, T Tony Cai, and Hongzhe Li. Direct estimation of differential networks. *Biometrika*, page asu009, 2014.