Learning infinite mixture of networks

Yizhe Zhang *

Abstract

We propose a Bayesian method to discover entangled directed graphs from scratch data. This method can be applied to gene regulation study and other applications. We show that an EM approach can recover a fixed number of components. Using a Dirichlet process mixture model, it is also possible to discover infinite mixture of causality relationships.

1 Aim

Learning regulatory network structure is central to the holistic view of gene regulation. Typically, score-based search algorithms is commonly employed for discovering structure represented as directed acyclic graphs (DAGs) from gene expression data[3]. However, the experimental data may be collected from a variety of cells or conditions, thus may come from 'multiple' network. Inferring sub-network structures can help us understand the subtle differential regulations among cell population at higher resolution in an unsupervised manner. It should be noted that even if the underlying true network is shared by all cells or conditions, the appeared marginal network of several selected genes can still be different. Attempts have been made to learn finite mixture of DAGs[6]. However, the method have not been introduced to gene network discovery, and is limited by a predifined components size.

The aim of this study is to reconstruct network from heterogeneous data. In particular, we extend from finite mixture of networks to infinite mixture using Dirichlet process. A block gibbs sampler approach employed for inference. We also discussed the identifiablity of learning a mixture of networks.

2 Method

2.1 Learning structure from homogeneous data

We first describe the inference approach employed to find the posterior of structures from homogeneous data. In regulatory network discovery problem, a node in the graph represent a gene or transcription factor. The directed edge between nodes denote the causal relationship between the nodes. The causal relationship can be activation, suppression or non-linear regulation.

For a network G parameterized by Θ^G, the marginal distribution $P(D|G)$ is given by

$$P(D|G) = \int P(D|\Theta^G, G)P(\Theta^G|G)d\Theta^G$$

Affiliation: Computational Biology & Bioinformatics Program, Duke University

We use $\mathrm{Pa}(x_i)\,\mathrm{to\,denote}$ the parent node of x_i in G. In Gaussian case, a BGe prior [1] for the parameters θ let the marginal to be factorized.

$$\begin{aligned} P(D|G) &= \prod_{i=1}^{N} \mathrm{FamScore}(x_i|\,\mathrm{Pa}(x_i), D) \\ &= \prod_{i=1}^{N}\prod_{s=1}^{D} \int P(x_i = D^s_{x_i}|\,\mathrm{pa}(x_i) = D^s_{\mathrm{pa}(x_i)}, \theta_i) P(\theta_i|\alpha) d\theta \end{aligned}$$

Where N denotes the number of random variables, D is the total sample number. θ_i includes all relevant parameters in Θ^G for predicting x_i given $\mathrm{pa}(x_i)$. The conjugacy of BGe prior (Normal-Wishart) let the integral to have a close form, as the function of sufficient statistics from data. Alternatively, BIC can be employed for computational concerns.

If we further assume uniform prior knowledge in structure space, given by

$$\begin{aligned} P(G|D) &\propto P(G)P(D|G) \\ &\propto P(D|G) \end{aligned}$$

The posterior can be sampled by Metropolis-Hasting approach, where we define a move set (add, delete and reverse) and conduct MCMC sampling.

2.2 Learning mixture of networks

A finite mixture of DAG (mDAG) of K components is written as[6]

$$P(D|\,\mathrm{mDAG}) = \sum_k \pi_k P(D|G_k)$$

Where D is the generated data, π_k is the mixture weight ($\sum_k \pi_k = 1, \pi_k \geqslant 0$). A practical approach for mixture model is through EM algorithm. Starting with certain initial setting, we iteratively assign data samples to clusters and find the structure within each cluster. A hard EM approach will assign data sample to a single cluster, while a standard EM algorithm will evaluate the probability of assigning to each cluster, and use *expected sufficient statistics* (sufficient statistics weighted by cluster assignment probability) to update.

Algorithm 1 EM algorithm for mDAGs

```
Initiate the cluster assignment Z
Until convergence
for c = 1 : Cluster_size do
    Sample structures within cluster G_c
    Estimate cluster probability π_c
end for
for d = 1 : Sample_size do
    for c = 1 : Cluster_size do
        Calculate expected sufficient statistics for cluster c
        for g = 1 : Sampled_graph_size do
            Find assignment probability to this graph g
        end for
        Find assignment probability to this cluster c
    end for
end for
```

For hard EM, the assignment probability of data points D^s to cluster k given a graph G is given by,

$$\begin{aligned} P(D^s|D, G) &= \int P(D^s|\Theta^G, G) P(\Theta^G|D, G) d\Theta^G \\ &= \prod_{i=1}^{N} f(\mathrm{SS}_k) \end{aligned}$$

Where the posterior local distribution of Θ^G is still in the same family. SS_k are the expected sufficient statistics computed from data assigned to cluster k. The cluster assignment probability can be find via Bayesian rule.

$$P(C^s|D^s, D) \quad \propto \quad \pi_c \sum_{G \in \mathbb{G}_c} P(D^s|D^c, G)P(G)$$

Where $\mathbb{G}_c$ are the structure set of cluster c sampled from M-step. π_c is the proportion probability of cluster c. In the derivation for standard EM, the sufficient statistics should be replaced by expected sufficient statistics ESS_k.

2.3 Infinite mixture

We further took infinite limit of the number of mixture components, which we denoted as imDAG (infinite mixture of Directed Acyclic Graphs). The generative model of an imDAG is as follow

$$\begin{aligned} \pi &\sim \mathrm{stick}(\lambda) \\ z_i &\sim \pi \\ G, \Theta^G &\sim \mathrm{BGe}(\Gamma) \\ x_i &\sim S\left(G_{z_i}, \Theta^{G_{z_i}}\right) \end{aligned}$$

Where $\mathrm{BGe}(\alpha)$ denotes the BGe metric parameterized by α. S denote the generative model of network samples given the graph and corresponding parameters. We let $x_i^{(j)}$ be the value of jth variable in $i\,\mathrm{th}$ sample.

$$P(x_i|G, \Theta^G) \quad = \quad \prod_{j=1}^{N} P(x_i^{(j)}|\,\mathrm{pa}(x_i^{(j)}), \theta^{(j)})\,\mathrm{NW}(\theta^{(j)}|\Gamma^{(j)})$$

Where NW denote Normal-Wishart distribution parameterized by Γ. Following Dirichlet process mixture model[4], we marginalize over possible DAGs and parameters gives a collapsed Gibbs sampler for this model.

$$\begin{aligned} p(z_i = k|z_{-i}, x, \lambda, \mathbb{G}, \Theta^{\mathbb{G}}) &= p(z_i|z_{-i}, \lambda)p(x_i|x_{-i}, z_i = k, \mathbb{G}, \Theta^{\mathbb{G}}) \\ p(z_i|z_{-i}, \lambda) &= \begin{cases} \frac{N_{k,-i}}{\lambda+N-1}, \text{if } k \text{ appears before} \\ \frac{\lambda}{\lambda+N-1}, \text{if } k \text{ is new} \end{cases} \\ p(x_i|x_{-i,k}, \mathbb{G}_k, \Theta^{\mathbb{G}_k}) &= \sum_{G \in \mathbb{G}_k} P(x_i|x_{-i,k}, G)P(G) \\ P(x_i|x_{-i,k}, G) &= \int P(x_i|\Theta^G, G)\,\mathrm{NW}(\Theta^G|x_{-i,k}, G, \Gamma)d\Theta^G \end{aligned}$$

Where $\mathrm{NW}(\Theta^G|x_{-i,k}, G, \Gamma)$ is the posterior distribution of LPDs (local probability distribution) estimated from data in cluster k.

A more efficient sampling scheme for DP mixture model is by block sampling (Ishwaran & James, 2001). A Blocked Gibbs Sampler update $z_i \in \{1 \ldots N\}$ by multinomial sampling

$$P(z_i = k|-) = \frac{\pi_k L(D; G_k, \theta_k)}{\sum_l^K \pi_l L(D; G_l, \theta_l)}$$

Which is followed by updating stick-breaking weight and Θ

$$\begin{aligned} V_h \sim &\quad Be(1 + n_h, \alpha + \textstyle\sum_{l=h+1}^{N} n_l) \\ \pi_h \sim &\quad V_h \textstyle\prod_{l<h} V_l \end{aligned}$$

Where n_h denote the number of samples assigned to cluster h. This method requires a predefined number to upper bound the total cluster number. However, by choosing a relatively large number, the approximation error is typically small.

2.4 Identifiablity of mixture of DAGs

A mixture of DAGs is identifiable if the DAG components can be seperated. Unfortunately, a mixture of DAGs with discrete categorical variables is not identifiable. A trivial example is a degenerate DAG which have one single discrete categorical node. It can have arbitary numbers of possible decomposition. Generally, if a mixture of DAGs can be perceived as generated from one single DAG, it's unidentifiable.

Proposition 2.1 (Identifiablity) *A mixture of DAGs* $\{G_1, \Theta_{G_1}, \pi_1\}, \ldots, \{G_N, \Theta_{G_N}, \pi_N\}$ *is identifiable, if there exists no such DAG* $\{G, \Theta_G\}$*, where for each node* i*,* $\sum_{n=1}^{N} \pi_n P(v_i | Pa_{G_n}(v_i), \Theta_{G_n}) = P(v_i | Pa_G(v_i), \Theta_G)$

3 Simulation study

For synthetic data, we mixed three randomly generated DAGs, each of which have five variables. The LPD parameters are sampled from uniform distributions. For each DAG, 50 data cases were generated.

Starting from initial random guess, the both hard EM and standard EM can recover the right components within 20 iteractions. The average acception ratio of Metropolis Hasting is 48.

	Correct ratio	Negative log-likelihood
EM	0.8214	2612
hard EM	0.8067	2890

Table 1: Performance on simulation data

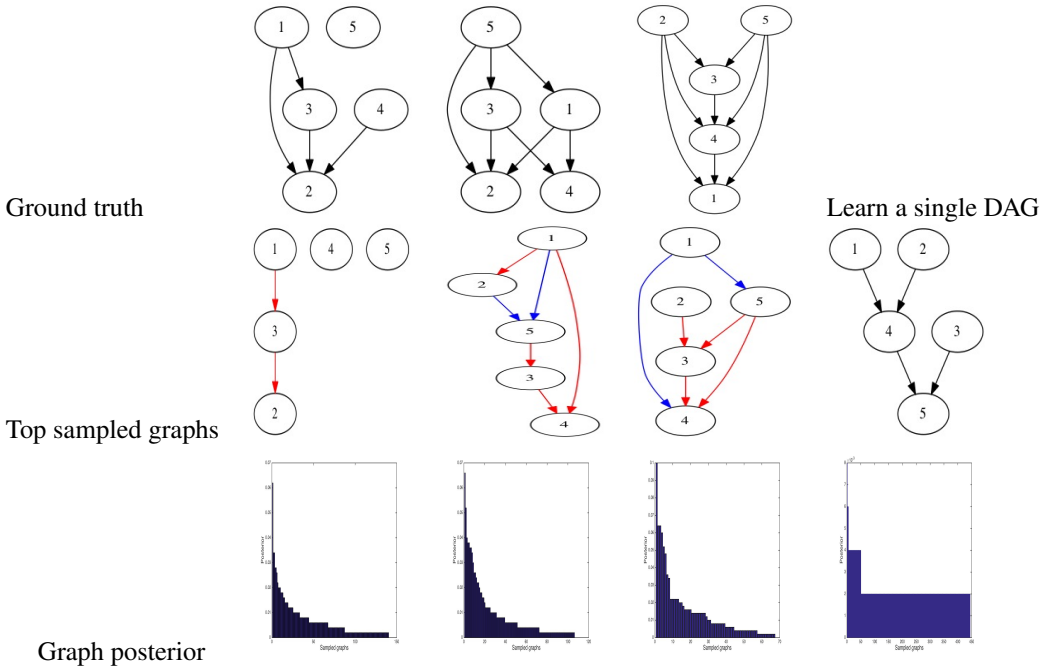

Figure 1: Top DAGs identified from each component. Correct edges are colored red, reversally correct edges are colored blue, indirectly correct edges are colored green

We use Gibbs sampling for find infinite components of DAGs from data. For computational concerns, the sampler was only run for 30 iterations. The sampler seems to be quickly converged to the right posterior, and the number of component is fluctuating around four.

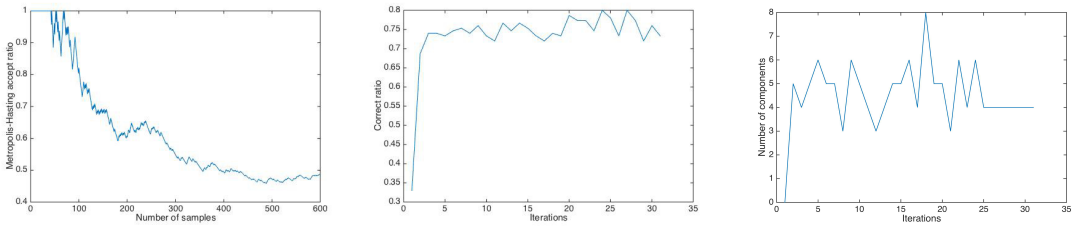

Figure 2: DP result. Left: MH accept ratio of structure sampling, middle: proportion of correct label, right: number of components.

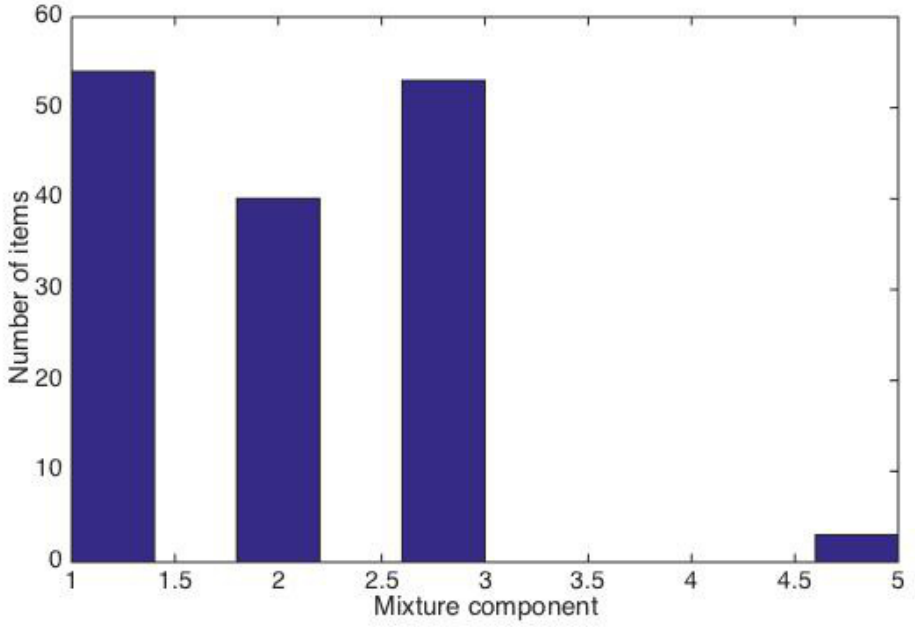

Figure 3: Number of items in each component after 30 iteractions

4 Real data analysis

We evaluated our methods on the DREAM5 challenge datasets. In particular, we examined a gene expression dataset with 4511 genes under 805 chip experiments. Among all those experiments, 45 have certain gene(s) knocked out. All indirect connections from DREAM reported graph were collapsed to be presented as direct link. Due to computational concerns, we only pick 10 genes for evaluation.

After 20 iterations, two major components are identified. The top component seems to correspond to the DREAM reported DAG, with several edges been falsely identified. The second component resembles the top one, except for the fact that several edges are missing. One of the missing edge $(3 \rightarrow 1)$ may reflect the knock-out of gene 3 in 23 experiments.

5 Discussion

Causal inference is challenging especially in regulatory network recovery where data can come from different conditions. In this paper we propose an approach to infer infinite number of network

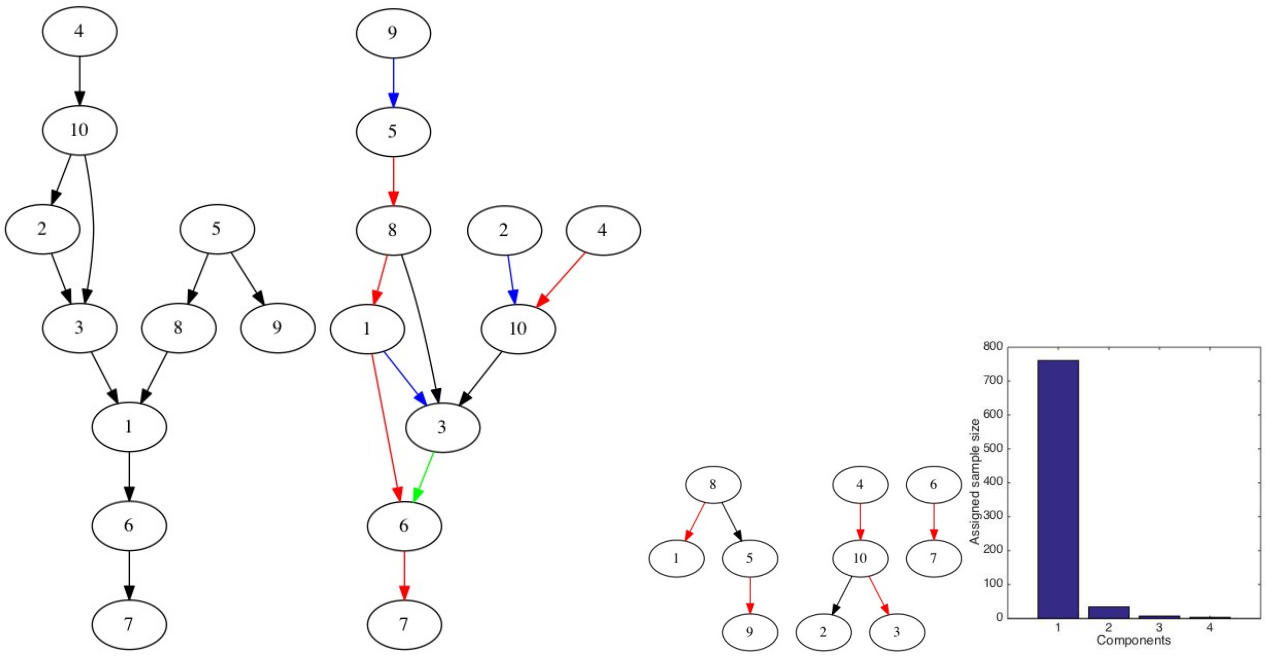

Figure 4: Identified DAGs from Real data. Left: DREAM reported graph, middle: top and second top components identified from data. right: the porportion for each component. Correct edges are colored red, reversally correct edges are colored blue, indirectly correct edges are colored green

mixtures from scratch observation data, with a full bayesian treatment. The unlimited nature of non-parameteric approach enables finding network structure in an automatic manner, but also come with a price. This nested MCMC approach involves MH sampling of graphs for collapsed sampling hidden assignment for each data experiment, which has a fairly heavy computational complexity. To alleviate this, one could either consider apply an approximate EM algorithm for DP mixture model, or conduct group sampling.

The current implementation of this method is based on Gaussian linear local probabilistic model. However, in real case the joint data distribution may often come from a model that is non-Gaussian or non-linear. The regulation of certain gene may saturate with additional regulator comes in, in which case it may be more appropriate to model the regulation effect as sigmoid function.

References

[1] D Geiger and D Heckerman. Learning gaussian networks. *Proceedings of the Tenth international conference on UAI*, , 1994.

[2] D. Koller and N Friedman. Probabilistic graphical models: principles and techniques. 2009.

[3] Florian Markowetz and Rainer Spang. Inferring cellular networks: a review. *BMC Bioinformatics*, 8(Suppl 6):0, 2007.

[4] C E Rasmussen. The infinite Gaussian mixture model. *NIPS*, , 1999.

[5] Le Song, Mladen Kolar and Eric P. Xing. Time-Varying Dynamic Bayesian Networks. , :1732–1740, 2009.

[6] B Thiesson, C Meek and D M Chickering. Learning mixtures of DAG models.*Proceedings of the ...* 1998.