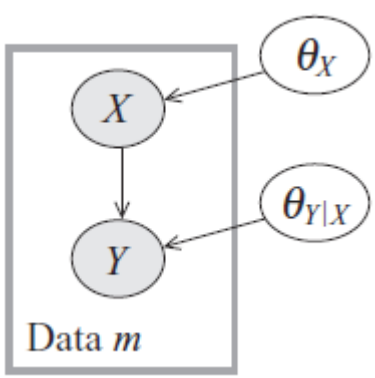
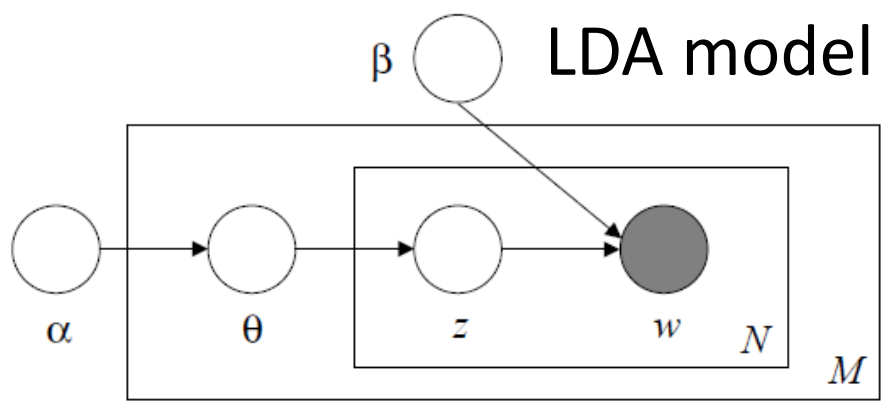
Reviews: Parameter Learning

- Some requirements:
  - Samples are *i.i.d.*: samples are independent if the parameters are given
    - $\mathrm{P}(\mathcal{D}|\theta) = \prod_{m=1}^{M} \mathrm{P}(x[m]|\theta)$
  - Parameters are decomposable (local learning)
    - No direct dependence between free parameters





Reviews: Parameter Learning

- Maximum likelihood estimation

$$\arg\max_{\theta} l(\theta : \mathcal{D})$$

- Bayesian estimation

$$\mathrm{P}(\theta|\mathcal{D}) = \frac{\mathrm{P}(\mathcal{D}|\theta)\,\mathrm{P}(\theta)}{\mathrm{P}(\mathcal{D})}$$

- Maximum a posterior estimation (MAP)

$$\arg\max_{\theta} \frac{\mathrm{P}(\mathcal{D}|\theta)\,\mathrm{P}(\theta)}{\mathrm{P}(\mathcal{D})} = \arg\max_{\theta}\{\log \mathrm{P}(\mathcal{D}|\theta) + \log \mathrm{P}(\theta)\}$$

Reviews: Parameter Learning

- For MLE & MAP
  - Point estimation & optimization
  - Stochastic gradient descent (convexity)
  - Undirected models (as log-linear)

- For Bayesian estimation
  - Probability inference
  - Conjugate priors
  - MCMC inference

Reviews: Learning with Incomplete Data

- General principles
  - Set initial parameters
  - **Infer** the hidden variables (or missing data) based on the observed variables (*POSTERIOR*!)
    - Fill with probability
    - Fill with a value (MCMC or MAP)
  - **Learn** (update) parameters
    - MLE (or MAP)
      - Expectation maximization (also hard-EM)
      - Stochastic gradient ascent
    - Bayesian Learning
      - MCMC sampling
      - Variational Bayesian learning
  - Repeat above steps until convergence

Jin Gu
Department of Automation, Tsinghua University
Email: jgu@tsinghua.edu.cn
Phone: (010) 62794294-866

Chapter 12 Structure Learning in Bayesian Networks

2021 Fall

Jin Gu (古槿)

Outlines: Structure Learning

- Constraint-Based Structure Learning
- Model Selection for Structure Learning
  - Maximum Likelihood Structure Score
  - Bayesian Structure Score
  - Bayesian Score for Bayesian Networks
  - Structure Search
- Bayesian Model Averaging
- Bayesian Networks in Practice
- Learning Relevance Networks

Chapter 12 Structure Learning in Bayesian Networks

Textbook1

Chapter 18.1-18.4 Structure Learning in Bayesian Networks

Chapter 20.7* Structure Learning in Undirected Models

Textbook2

Chapter 26 Graphical Model Structure Learning

* Advanced Readings

Other references

[1] Elidan G, Lotner N, Friedman N, Koller D. Discovering hidden variables: a structure-based approach. NeurIPS 2001.

[2] Elidan & Friedman. Learning hidden variable networks: the information bottleneck approach. JMLR 2005, 81-127.

Constraint-Based Structure Learning

- Goal: reconstruct a network structure that best captures the independencies in the data or find a minimal I-map for the data.
- The main question is how to test the independencies in the data. Assume we want to test the independence between X and Y

$$d_{\chi^2}(\mathcal{D}) = \sum_{x,y} \frac{\left(M[x,y] - M\hat{P}(x)\hat{P}(y)\right)^2}{M\hat{P}(x)\hat{P}(y)}$$

$$d_I(\mathcal{D}) = \frac{1}{M} \sum_{x,y} M[x,y] \log \frac{M[x,y]}{M[x]M[y]}$$

Constraint-Based Structure Learning

- $d_{\chi^2}(\mathcal{D}) = \sum_{x,y} \frac{\left(M[x,y] - M\hat{P}(x)\hat{P}(y)\right)^2}{M\hat{P}(x)\hat{P}(y)}$
- After calculating $d_{\chi^2}(\mathcal{D})$, we can calculate the tail distribution of χ^2 distribution
  - $pvalue = 1 - F_{\chi^2}\left(d_{\chi^2}(\mathcal{D})\right)$

In practice, the *adjusted* p-value should be < 0.01 or 0.05

  - The freedom of distribution is the number of different assignments of $[x, y] - 1$
    - Explain: according to CLT, when $M[x, y]$ is large, it should follows a normal distribution

Constraint-Based Structure Learning

- Above testing can be easily extended to more complex local structure $X \leftarrow Z \rightarrow Y$

$$d_{\chi^2}(\mathcal{D}) = \sum_{x,y,z} \frac{\left(M[x,y,z] - M\hat{P}(x|z)\hat{P}(y|z)\hat{P}(z)\right)^2}{M\hat{P}(x|z)\hat{P}(y|z)\hat{P}(z)}$$

- Use the same procedure of Build-PDAG in *Textbook Algorithm 3.5*, we can reconstruct the Bayesian network

Model Selection for Structure Learning

- Define a structure probability or scoring function, then select an optimal graphic structure from a pre-defined structure space (a set of graphic structures)

- The structure learning task is transformed as an optimization or model selection task

Likelihood Structure Score

- Now, the likelihood function have two super-parameters, one for structure space $\mathcal{G}$ and the other one for parameter space Θ

$$P\left(\mathcal{D}|\mathcal{G},\theta_{\mathcal{G}}\right)$$

Structure can also be treated as parameters!

- The maximum likelihood structure score

$$\text{score}_{ML}(\mathcal{G}:\mathcal{D}) = \max_{\theta} P\left(\mathcal{D}|\mathcal{G},\theta_{\mathcal{G}}\right) = P\left(\mathcal{D}|\mathcal{G},\hat{\theta}_{\mathcal{G}}\right)$$

- MLE for the structure

$$\mathcal{G}^* = \arg\max_{\mathcal{G}} P\left(\mathcal{D}|\mathcal{G},\hat{\theta}_{\mathcal{G}}\right)$$

PROBLEM: MLE will always favor the model with more complex structure!!!

Bayesian Score

- The Bayesian score is defined as:

$$\text{score}_B(\mathcal{G}:\mathcal{D}) = \log P(\mathcal{D}|\mathcal{G}) + \log P(\mathcal{G})$$

- The second item is the structure prior
- The first item is a *marginal distribution* over data by averaging the parameter space for a given structure

$$P(\mathcal{D}|\mathcal{G}) = \int P\left(\mathcal{D}|\theta_{\mathcal{G}},\mathcal{G}\right)P\left(\theta_{\mathcal{G}}|\mathcal{G}\right)d\theta_{\mathcal{G}}$$

First Term: Bayesian Structure Score

- The Bayesian score is defined as:

$$\text{score}_B(\mathcal{G}:\mathcal{D}) = \mathbf{log}\, \boldsymbol{P}(\boldsymbol{\mathcal{D}}|\boldsymbol{\mathcal{G}}) + \log P(\mathcal{G})$$

- The first term is called as *Bayesian structure score*: a *marginal distribution* over data with a given structure by averaging the parameter space

$$P(\mathcal{D}|\mathcal{G}) = \int P\left(\mathcal{D}|\theta_{\mathcal{G}}, \mathcal{G}\right) P\left(\theta_{\mathcal{G}}|\mathcal{G}\right) d\theta_{\mathcal{G}}$$

  – Compared with the ML score, $P\left(\mathcal{D}|\mathcal{G}, \hat{\theta}_{\mathcal{G}}\right)$ is a point estimation

Bayesian Structure Score: Parameter Averaging Controls Overfitting

- According to chain-rule, we have

$$P(\mathcal{D}|\mathcal{G}) = \prod_{m=1}^{M} P(x[m]|x[1], \cdots, x[m-1], \mathcal{G})$$

- $P^{[m]} = P(x[m]|x[1], \cdots, x[m-1], \mathcal{G})$ can be treated as the *prequential prediction* of m-th sample based on all the previous samples

$$P^{[m+1]} = \int P\left(x[m+1]|\theta^{[m]}, \mathcal{G}\right) P\left(\theta^{[m]}|x[1{\sim}m], \mathcal{G}\right) d\theta$$

Bayesian estimation is a kind of *prequential analysis*.
It can control the overfitting risk the learning algorithm.

Bayesian Structure Score: Simple Example

- Taking the coin example, we have the observation $\{1, 0, 0, 1, 1\}$
  - For MLE, we get $\theta^* = P(X = 1) = 0.6$. At that estimation, the likelihood $\boldsymbol{P}\left(\boldsymbol{\mathcal{D}}|\boldsymbol{\mathcal{G}}, \widehat{\boldsymbol{\theta}}_{\boldsymbol{\mathcal{G}}}\right)$ is equal to $0.6^3 \times 0.4^2 \approx 0.035$
  - For Bayesian estimation with Beta prior, we have the prediction $P(x[m+1] = 1) = \frac{m[1]+\alpha_1}{m+\alpha}$ ($m[1]$: the number of "1" in the previous m samples). So, we get $\boldsymbol{P}(\boldsymbol{\mathcal{D}}|\boldsymbol{\mathcal{G}}) = \frac{\alpha_1}{\alpha}\frac{\alpha_0}{1+\alpha}\frac{1+\alpha_0}{2+\alpha}\frac{1+\alpha_1}{3+\alpha}\frac{2+\alpha_1}{4+\alpha} \approx 0.017$ if $\alpha_i = 1$

Risk of overfitting: ML score has higher risk than *Bayesian Score*!

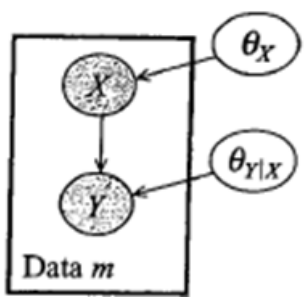
Bayesian Structure Score for BNs

- For the simple case with only two variables, if the structure has no edge $\mathcal{G}_\emptyset$

$$P(\mathcal{D}|\mathcal{G}_\emptyset) = \int P(\theta_X|\mathcal{G}_\emptyset) \prod_m P(x[m]|\theta_X, \mathcal{G}_\emptyset)\, d\theta_X \cdot \int P(\theta_Y|\mathcal{G}_\emptyset) \prod_m P(y[m]|\theta_Y, \mathcal{G}_\emptyset)\, d\theta_Y$$

- If the structure has the edge $X \to Y$

$$P(\mathcal{D}|\mathcal{G}_{X\to Y}) = \int P(\theta_X|\mathcal{G}_{X\to Y}) \prod_m P(x[m]|\theta_X, \mathcal{G}_{X\to Y})\, d\theta_X \cdot \int P\left(\theta_{Y|x^0}|\mathcal{G}_{X\to Y}\right) \prod_{m:x[m]=x^0} P\left(y[m]|\theta_{Y|x^0}, \mathcal{G}_{X\to Y}\right) d\theta_{Y|x^0} \cdot \int P\left(\theta_{Y|x^1}|\mathcal{G}_{X\to Y}\right) \prod_{m:x[m]=x^1} P\left(y[m]|\theta_{Y|x^1}, \mathcal{G}_{X\to Y}\right) d\theta_{Y|x^1}$$



$$\boxed{\text{score}_B(\mathcal{G}:\mathcal{D}) = \log P(\mathcal{D}|\mathcal{G}) + \log P(\mathcal{G})}$$

Bayesian Structure Score for BNs

- For general BNs, if the parameter satisfies the *global* and *local* independence

$$P(D|G)=\prod_{i}\prod_{u_i\in Val(Pa_i^G)}\int_{\Theta_{X_i|u_i}}\prod_{m:pa_i[m]=u_i}P(x_i[m]|u_i,\theta_{x_i|u_i},G)P(\theta_{x_i|u_i}|G)d\theta_{x_i|u_i}$$

- If we use *Dirichlet* prior for all parameters

$$P(D|G)=\prod_{i}\prod_{u_i\in Val(Pa_i^G)}\frac{\Gamma(\alpha^G_{X_i|u_i})}{\Gamma(\alpha^G_{X_i|u_i}+M[u_i])}\prod_{x_i^j\in Val(X_i)}\frac{\Gamma(\alpha^G_{x_i^j|u_i}+M[x_i^j,u_i])}{\Gamma(\alpha^G_{x_i^j|u_i})}$$

$$\alpha^G_{X_i|u_i}=\sum_{x_i^j\in Val(X_i)}\alpha^G_{x_i^j|u_i}$$

$$\boxed{\text{score}_B(\mathcal{G}:\mathcal{D})=\log P(\mathcal{D}|\mathcal{G})+\log P(\mathcal{G})}$$

Priors for Bayesian Structure Score

- Bayesian structure score has a good property
  - Decomposable if the parameter satisfies two requirements

$$\text{score}_{BS}(\mathcal{G}:\mathcal{D}) = \log P(\mathcal{D}|\mathcal{G}) = \sum_i \text{FamScore}\left(X_i|Pa_i^{\mathcal{G}}:\mathcal{D}\right)$$

- It means that the local score is independent with each other. When searching optimal structure $\mathcal{G}^*$, we can do local search for the optimal set of parents Pa_i for each variable X_i if we use Bayesian structure score

$$P(D|G) = \prod_i \prod_{u_i \in Val\left(Pa_i^G\right)} \int_{\Theta_{X_i|u_i}} \prod_{m:pa_i[m]=u_i} P\left(x_i[m]|u_i, \theta_{x_i|u_i}, G\right) P\left(\theta_{x_i|u_i}|G\right) d\theta_{x_i|u_i}$$

Bayesian Structure Score for BNs

- If we use *Dirichlet prior*, when M is large enough, Bayesian structure score is equal to:

$$\log P(D \mid G) = l\left(\hat{\theta}_G : D\right) - \frac{\log M}{2}\dim[G] + O(1)$$

$\dim[\mathcal{G}]$ is the number of independent parameters in $\mathcal{G}$

- Omit the last constant, we get the *Bayesian information criterion* (BIC) score

$$score_{BIC}(G : D) = l\left(\hat{\theta}_G : D\right) - \frac{\log M}{2}\dim[G]$$

BIC for BNs

- BIC score can be further derived like below:

$$score_{BIC}(G:D) = M\sum_{i=1}^{n} I_{\hat{P}}\left(X_i, Pa_{X_i}\right) - M\sum_{i=1}^{n} H_{\hat{P}}\left(X_i\right) - \frac{\log M}{2}\dim[G]$$

  - The first term is for the mutual information of local structures
  - The second term is for the entropy of variables
  - The last term is the smoothing factor across the parameter space (*Dirichlet* prior)

>>> The second term is not associated with BN structures
>>> Find an optimal structure which maximizing the mutual information but penalizing the model complexity

$$score_{BIC}(G:D) = M\sum_{i=1}^{n} I_{\hat{P}}(X_i, Pa_{X_i}) - M\sum_{i=1}^{n} H_{\hat{P}}(X_i) - \frac{\log M}{2}\dim[G]$$

- A scoring function is *consistent* if
  - The perfect map $\mathcal{G}^*$ will maximize the score
  - All $\mathcal{G}$ that are not I-equivalent to $\mathcal{G}^*$ will have strictly lower score

- **The BIC score is consistent**
  - See proof in ***Textbook Theorem 18.2***

In practice, BIC is commonly used rather than Bayesian score.

General Structure Priors

- Second term: structure priors
  - As the increasing of data, the importance of structure prior $\log P(\mathcal{G})$ is linearly decreasing
  - Uniform prior is commonly used: $P(\mathcal{G}) = const$
  - But when data are rare, we tend to penalize the number of edges in $\mathcal{G}$: $P(\mathcal{G}) \propto c^{|G|}, 0 < c < 1$

$$\boxed{\mathrm{score}_B(\mathcal{G}:\mathcal{D}) = \log P(\mathcal{D}|\mathcal{G}) + \boldsymbol{\log P(\mathcal{G})}}$$

Search for Optimal Structure

- Given below information, the optimal structure $\mathcal{G}^*$ can be found by searching the solution optimizing the structure score
- Given below inputs
  - Training set $\mathcal{D}$
  - Scoring function (Bayesian score, BIC score, etc.)
  - Searching space (a set of possible structures $\boldsymbol{\mathcal{G}}$)
- If the scoring function is *decomposable*, we can do local optimization search

Search for Optimal Structure

What is the remaining challenge?

- The structure is *discretized*, no gradient can be used for the scoring function optimization
- The problem is *NP-hard*: for a set of k variables, there are $2^{O(k^2)}$ possible structures

Learning Tree-Structured Networks

- If the search space is limited in tree structure, the model selection can be simplified

- *Tree structure: each variable has* ***at most one parent*** *in the graph*

Learning Tree-Structured Networks

- Let start from an empty structure $\mathcal{G}_\emptyset$:

$$\text{score}(\mathcal{G}_\emptyset; \mathcal{D}) = \sum_i \text{FamScore}(X_i : \mathcal{D})$$

- Add one edge $X_j \to X_i$ will increase the score:

$$\Delta_{j\to i} = \text{FamScore}\left(X_i | X_j : \mathcal{D}\right) - \text{FamScore}(X_i : \mathcal{D})$$

- In tree-structured networks, each variable has at most one parent ⇔ if the structure score is decomposable, adding an edge only causes a "local" increment $\Delta_{j\to i}$
- After adding k edges, we get new graph $\mathcal{G}$ with

$$\Delta(\mathcal{G}) = \sum_{(X_j \to X_i) \in \mathcal{G}} \Delta_{j\to i}, \mathcal{G}\ is\ a\ tree$$

Learning Tree-Structured Networks

In tree-structured networks, each variable has at most one parent ⇔ if the structure score is decomposable, adding an edge only causes a "local" increment $\Delta_{j\to i}$

After adding k edges, we get new graph $\mathcal{G}$ with

$$\Delta(\mathcal{G}) = \sum_{(X_j \to X_i) \in \mathcal{G}} \Delta_{j\to i}, \mathcal{G}\ is\ a\ tree$$

- We can transform the structure search problem to finding a maximum weight spanning forest

$$\mathcal{G}^* = \arg\max_{\mathcal{G}} \Delta(\mathcal{G}), \mathcal{G}\ is\ a\ tree$$

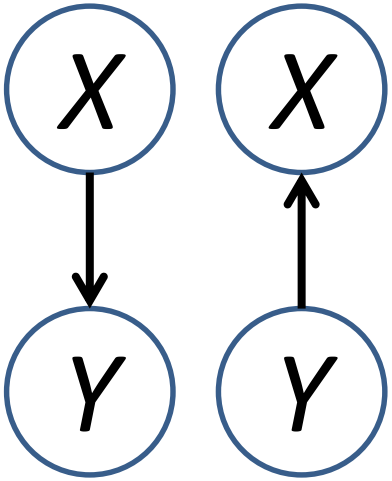
- The algorithm complexity: $O(n^2 M + n^2 \log n)$

General Structure Search

- Consider the structure space that each variable has at most d parents
- It is *NP-hard* if $d \geq 2$ for the below problem

$$\mathcal{G}^* = \arg\max_{\mathcal{G} \in \boldsymbol{\mathcal{G}}_d} \text{score}(\mathcal{G}: \mathcal{D})$$

70%? 15%?



MAP structure or sub-optimal structure is usually misleading due to the large search space. *A consensus probability of an edge* is more helpful than a simple structure!

General Structures with MCMC

- Using the three edge operations, any two graphs are reachable in finite steps
  - Edge addition, deletion, reversal

- We can use MCMC methods to get the posteriors of different structures given the data
  - State: a graph structure
  - Transition: edge addition, deletion, reversal

General Structures with MCMC

- It is hard to get the "exact" transition probability $P(\mathcal{G}'|\mathcal{G},\mathcal{D})$
- So, we can only use *Metropolis-Hastings* algorithm rather than Gibbs sampling

$$A(x \to x') = \min\left[1, \frac{\pi(x')\mathcal{T}^{Q}(x' \to x)}{\pi(x)\mathcal{T}^{Q}(x \to x')}\right]$$

  - Define $\mathcal{T}^{Q}(x \to x')$
  - In practice: 1) randomly select an operation using pre-defined probability $P(Action)$; 2) randomly select a connected graph with that operation $P(Transition)$

General Structures with MCMC

- Each step, we need to do calculate

$$\frac{\pi(x')\mathcal{T}^{Q}(x'\to x)}{\pi(x)\mathcal{T}^{Q}(x\to x')} = \frac{\pi(x')}{\pi(x)}\frac{\mathcal{T}^{Q}(x'\to x)}{\mathcal{T}^{Q}(x\to x')}$$

The transition probability term $\mathcal{T}^{Q}$ can be calculated based on the two operations defined in the previous slide.

$$\frac{\pi(x')}{\pi(x)} = \frac{P(G'|D)}{P(G|D)} = \frac{P(D|G')P(G')}{P(D|G)P(G)}$$

- If we use *BIC score* and *uniform prior*

$$\frac{\pi(x')}{\pi(x)} = \frac{P(D|G')P(G')}{P(D|G)P(G)} = \frac{\exp\{score_{BIC}(G':D)\}}{\exp\{score_{BIC}(G:D)\}}$$

$$score_{BIC}(G:D) = l(\hat{\theta}_G : D) - \frac{\log M}{2}\dim[G]$$

General Structures with MCMC

- The sampling process is **very time-consuming**
- Try to use collapsed particles
  - If we set the ordering of the variables, the learning process is relatively simple (polynomial complexity)
  - Please read Textbook 18.5.2
  - So we can use MCMC for modeling the transition between different orderings, and search for the optimal structure given the variable ordering

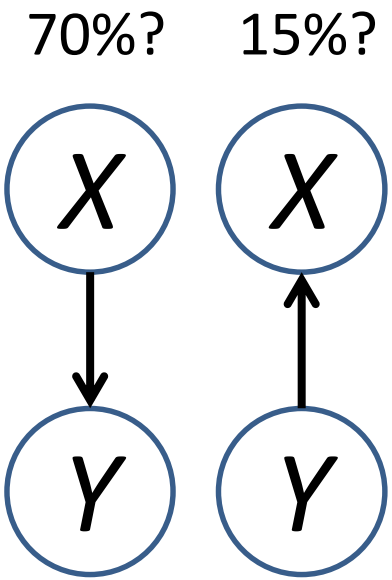
General Structures with MCMC

- After the chain is in steady state, you can estimate the probability of an observed edge (with direction) between any two variables based on generated particles

Toy example: record the percentages of the two possible edges

$$\mathrm{P}(X \rightarrow Y) \ \& \ \mathrm{P}(X \leftarrow Y)$$

based on the structures generated by MCMC



Bayesian Model Averaging

- Recall the Bayesian estimation for predicting
  - Bayesian prediction is calculating an expectation of probability over all possible parameters

$$P(X^{n+1}|X^1,\cdots,X^n) = \frac{1}{P(X^1,\cdots,X^n)}\int P(X^{n+1}|\theta)P(X^1,\cdots,X^n|\theta)P(\theta)d\theta$$

- We can do it in a similar way for unknown structures

$$P(X^{n+1}|X^1,\cdots,X^n) = \sum_{\mathcal{G}} P(X^{n+1}|X,\mathcal{G})P(\mathcal{G}|X)$$

Bayesian Model Averaging

- How to compute?
- MCMC again! In the previous section, MCMC is used to calculate the posteriors of structures
- Here, we used the collapsed MCMC to calculate the *estimated* target function
  - $P(X[M+1]|\mathcal{D}) = \sum_{\mathcal{G}} P(X[M+1]|\mathcal{G})P(\mathcal{G}|\mathcal{D})$
  - $P(X[M+1]|\mathcal{D}) \approx \frac{1}{K}\sum_{i=1}^{K} P(X[M+1]|\mathcal{G}_i)$
  - $\mathcal{G}_i$ is sampled from the posterior distribution $P(\mathcal{G}|\mathcal{D})$

Bayesian Networks in Practice

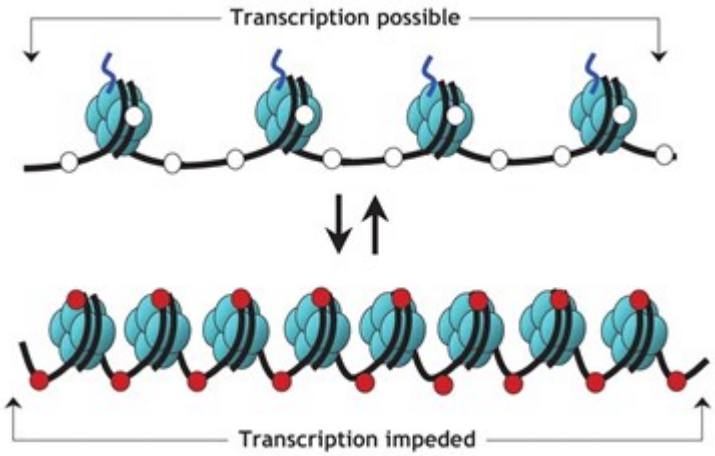
- Structure scores
  - BIC: Bayesian information criterion
    - $\text{score}_{BIC}(\mathcal{G}:\mathcal{D}) = l\left(\hat{\theta}_{\mathcal{G}}:\mathcal{D}\right) - \frac{\log n}{2}k$
  - AIC: Akaike information criterion
    - $\text{score}_{AIC}(\mathcal{G}:\mathcal{D}) = l\left(\hat{\theta}_{\mathcal{G}}:\mathcal{D}\right) - k - \frac{k(k+1)}{n-k-1}$
  - MDL: Minimum description length
- Structure priors
  - Uniform prior
  - Sparse prior: $P(\mathcal{G}) \propto c^{|G|}, 0 < c < 1$

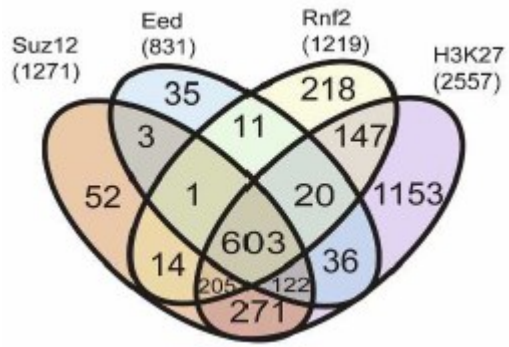
Bayesian Networks in Practice

- Structure learning frequently encounter local-optimal problem. The algorithm needs to re-run multiple times and find the most consistent structure.

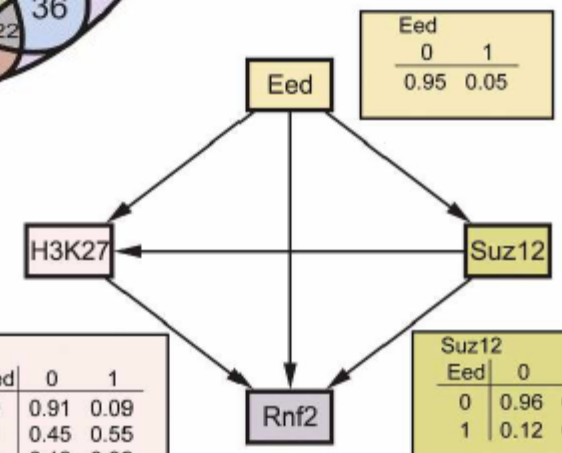
- Bootstrapping, a resampling method with sample replacement, is commonly used
  - **Sample bootstrapping** for evaluating classifier performance and Bayesian network structure stability
  - **Attribute bootstrapping** for evaluating cluster stability

Epigenetic Factor Network Inference





Hard to infer causal relation!!



Eed	
0	1
0.95	0.05

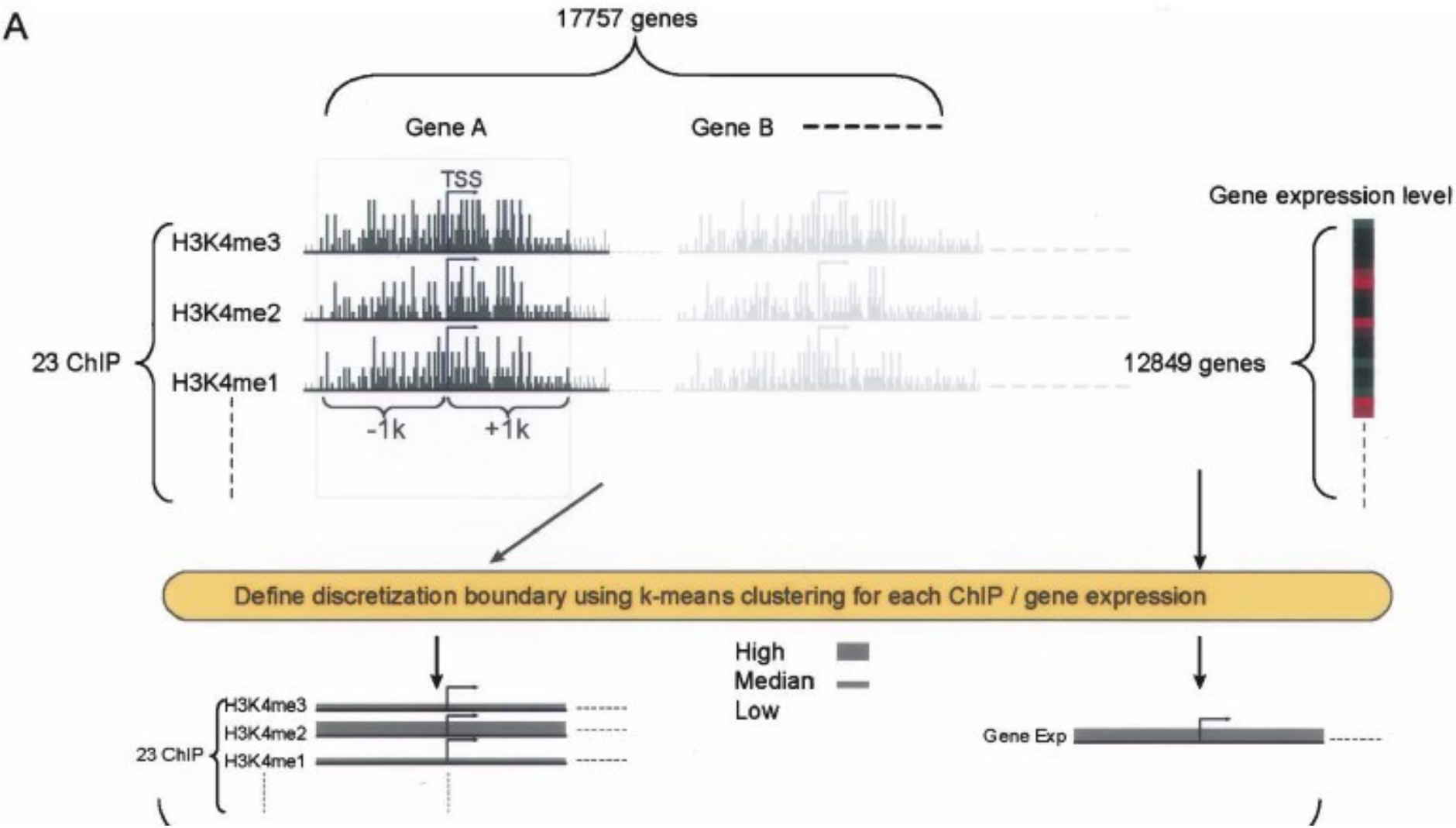
H3K27			
Suz12	Eed	0	1
0	0	0.91	0.09
0	1	0.45	0.55
1	0	0.12	0.88
1	1	0.01	0.99

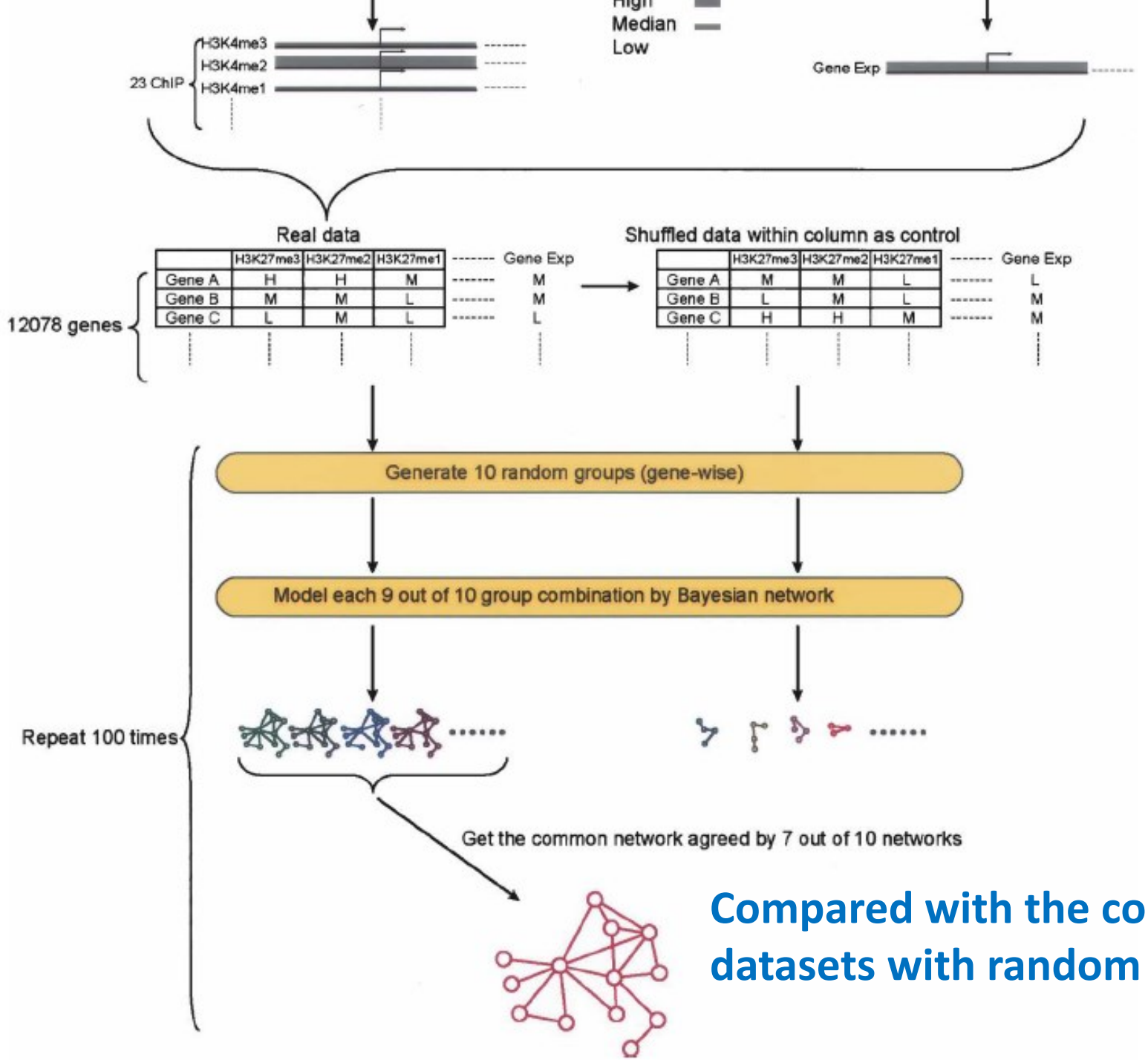
Suz12		
Eed	0	1
0	0.96	0.04
1	0.12	0.88

Rnf2				
H3K27	Suz12	Eed	0	1
0	0	0	0.98	0.02
0	0	1	0.75	0.25
0	1	0	0.78	0.22
0	1	1	0.67	0.33
1	0	0	0.89	0.11
1	0	1	0.64	0.36
1	1	0	0.57	0.43
1	1	1	0.17	0.83

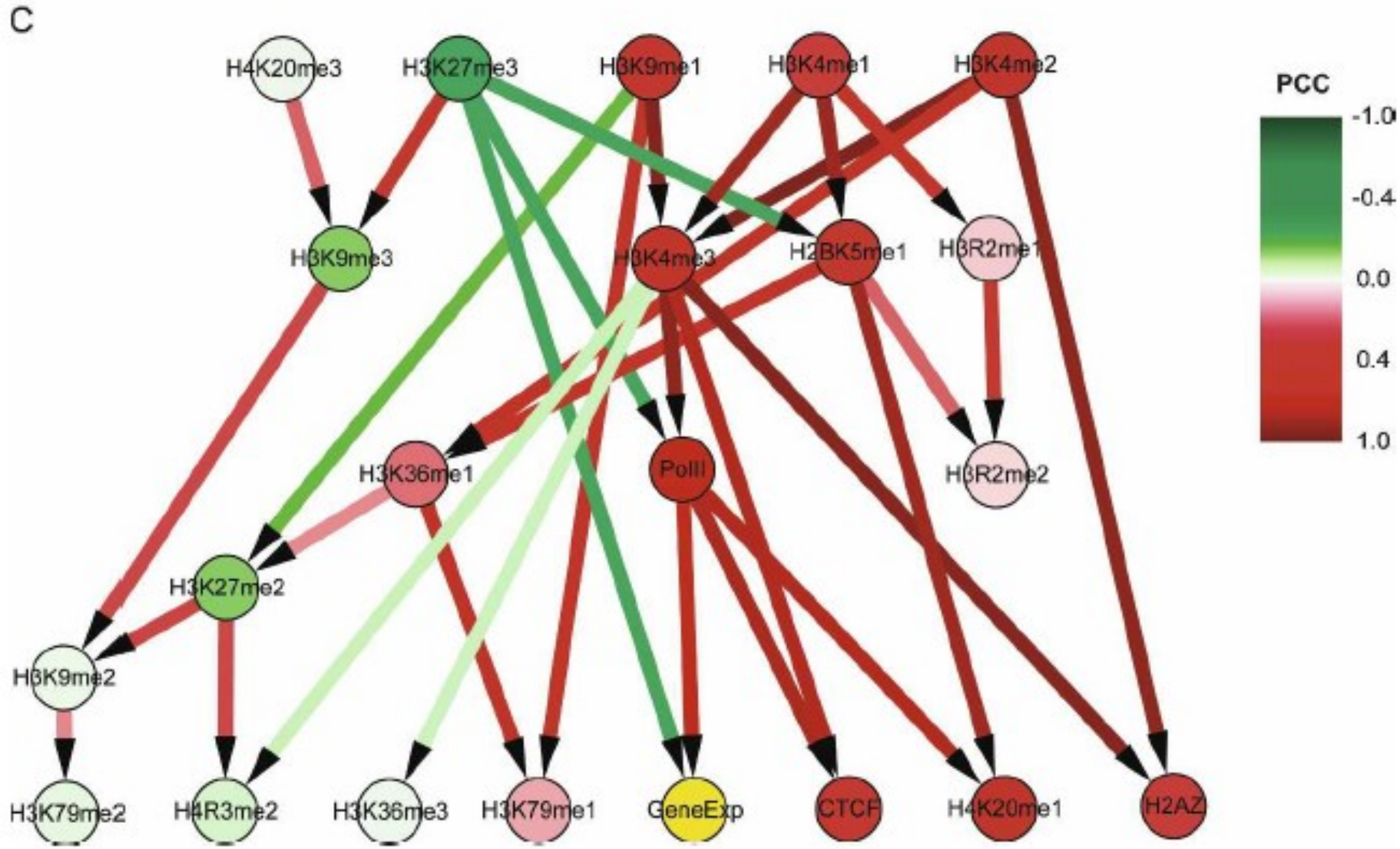
- Nodes (variables): the factors involved in epigenetic modifications
- First step: small-scale test on four factors

Hong et al. Genome Res 2008.


A
17757 genes
Gene A
Gene B
TSS
H3K4me3
H3K4me2
H3K4me1
23 ChIP
-1k
+1k
Gene expression level
12849 genes
Define discretization boundary using k-means clustering for each ChIP / gene expression
High
Median
Low
H3K4me3
H3K4me2
H3K4me1
23 ChIP
Gene Exp



Compared with the control datasets with random data


C
H4K20me3
H3K27me3
H3K9me1
H3K4me1
H3K4me2
PCC
-1.0
-0.4
0.0
0.4
1.0
H3K9me3
H3K4me3
H2BK5me1
H3R2me1
H3K36me1
PolII
H3R2me2
H3K27me2
H3K9me2
H3K79me2
H4R3me2
H3K36me3
H3K79me1
GeneExp
CTCF
H4K20me1
H2AZ

The Structures of Undirected Models?

- For undirected Gaussian & dicretized models, please see Chapter 26 of Textbook #2

- Parise S, Welling M. Structure learning in markov random fields. *NIPS* 2006.
- Chen Y, Welling M. Bayesian structure learning for markov random fields with a spike and slab prior. *AUAI* 2012.

Structure Score for Undirected Models

- The same as Bayesian networks, fully connected undirected models can “best” fit any observed data

- General solution
  - Structure constraint should be introduced to select “simpler” structure

Example: L1 Penalty for Gaussian Models

- Representation for Gaussian random fields: we can easily get the undirected models based on the *information matrix*

Gaussian Random Fields

- According to the distribution
  - $p(x) \propto exp\left(-\frac{1}{2}x^T Jx + (J\mu)^T x\right)$
- We can split it into two terms
  - $-\frac{1}{2}J_{ii}x_i x_i + h_i x_i$ ➔ single node
  - $-J_{ij}x_i x_j$ ➔ pairwise interaction
- Any multivariate Gaussian can be represented by a pairwise Markov network. This network is called a *Gaussian Markov Random Field*.
- But if the information matrix is not positive definite matrix, the GMRF may be illegal.

Example: L1 Penalty for Gaussian Models

- We can penalize the learned information matrix by L1 norm:

$$score_{L1}(J:D) = l(J:D) - \lambda \|J\|_1$$

$$J^* = \arg\max_J \left\{ \log\det J - tr(\Sigma J) - \lambda \|J\|_1 \right\}$$

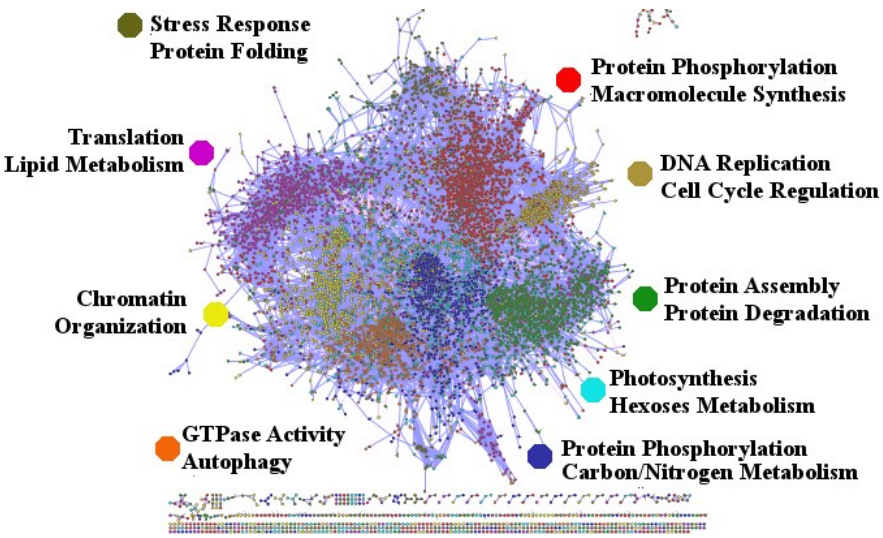
L1 norm penalizes **Sparsity**: simpler structures are preferred!

- In statistical inference, L1 norm based parameter learning is also called *Lasso* (least absolute shrinkage and selection operator)
- It can also be extended to L2 norm or a mixture L1 & L2 norm (refer to *Elastic Net*)

Relevance Networks

- Social networks

- Gene co-expression networks



Learning Relevance Networks

- From data matrix to relevance matrix
  - $\mathcal{D}_{N\times M}$: *N* features, *M* samples
    - For features: $A_{N\times N}$, a_{ij} indicates gene relevance
    - For samples: $B_{M\times M}$, b_{ij} indicates sample relevance

- Basic measurements (pairwise relevance)
  - Euclidean distance
  - Correlation (Pearson's & Spearman's correlation)
  - Mutual information

Learning Relevance Networks

- *Aim*: learn a similarity matrix *S*
- Input: data matrix $\mathcal{D}_{N\times M}$

- Kernel functions (used to measure similarities between observed samples)
  - $K(x[i], x[j])$ is an inner product (denoted as $K_{i,j}$)
  - Gaussian kernel: $\frac{1}{\sigma_{ij}\sqrt{2\pi}} \exp\left(-\frac{\|x[i]-x[j]\|_2^2}{2\sigma_{ij}^2}\right)$

Learning Relevance Networks

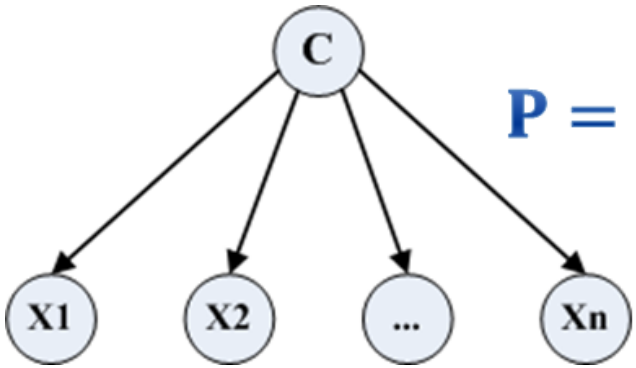
- Objective function
  - *Distance term*: $-\sum_{i,j} K_{i,j} S_{i,j}$
  - *Constraint* on S
    - F-norm: $\|S\|_F = \sqrt{\sum_i \sum_j s_{ij}^2} = \sqrt{\mathrm{tr}(S^* S)}$
    - Kernel norm: $\|S\|_* = \sum_i \lambda_i$ (rank regularization)
    - L1 norm: $\|S\|_1 = \sum_i \sum_j |s_{ij}|$ (sparsity regularization)
  - Additional constraints
- Minimization of objective function

$$\min_S -\sum_{i,j} K_{i,j} S_{i,j} + \beta \|S\|_F^2 + f(\lambda, S)$$

Need to define an objective function (based one the principle of the topology of the learned networks) to find the optimal S

How to Deal With Hidden Variables?

- As we see, hidden variables are crucial for solving complex problems
- Take naïve Bayes (binary) as the example
  - 2^n -1 parameters are needed if no class label
  - Only n+(n-1) parameters are needed for full model



$$\mathbf{P} = \prod_i \mathbf{P}(X_i|C)\mathbf{P}(C)$$

How to Deal With Hidden Variables?

- Can we infer the existence of hidden variables?
- Think about the ***AIM*** of introducing hidden variables: reduce the required parameters (reduce the dependences = reduce the number of edges in graph)

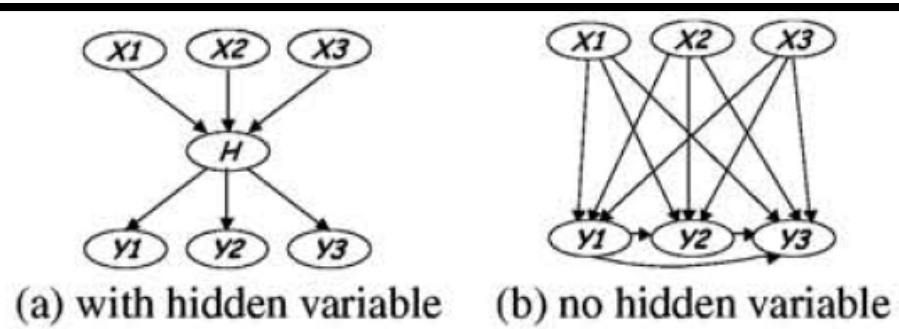


Figure 1: Hidden variable simplifies structure

Elidan G, Lotner N, Friedman N, Koller D. **Discovering hidden variables: a structure-based approach**. NIPS 2001.

Summary

- Constraint-Based Structure Learning
- Model Selection for Structure Learning
  - Bayesian Structure Score
  - Bayesian Score for Bayesian Networks
    - BIC score
  - Structure Search
- Bayesian Networks in Practice
- Learning Relevance Networks

The End of Chapter 12

BN structure learning is powerful for graph-based data representation