Part III. Systems Biology: 3. Learning the Structure of the Bayesian networks

Lecture 14 – Nov 18, 2015  
CSE 527 Computational Biology  
Instructor: Su-In Lee  
TA: Javad Hosseini  
TTh 12:00-1:20 @ SAV 130

Outline

- Bayesian network learning to infer gene regulatory networks
  - Parameter estimation
  - Structure learning
  - Example – learning module networks
- Evaluation of the method
  - Statistical evaluation
  - Biological interpretation
- Structure discovery in BNs
  - Model averaging
  - Bootstrapping

Inferring gene regulatory networks

- **Goal:** Reconstruct the gene regulatory network that controls gene expression
- **Method:** Let’s use probabilistic graphical models to represent the network

**Known structure, complete data**

- Network structure is specified
  - Learner needs to estimate parameters
  - Data does not contain missing values
Learning parameters

- Training data has the form:

\[
D = \begin{bmatrix}
\vdots & \vdots & \vdots & \vdots \\
\end{bmatrix}
\]

Likelihood function

- Assume i.i.d. samples
- Likelihood function is defined as:

\[
L(\Theta : D) = \prod_{m} P(E[m], B[m], A[m], C[m] : \Theta)
\]

- Joint distribution can be decomposed as:

\[
L(\Theta : D) = \prod_{m} \left( P(E[m] : \Theta) \times P(B[m] : \Theta) \times P(A[m] | B[m], E[m] : \Theta) \times P(C[m] | A[m] : \Theta) \right)
\]

- Reordering terms, we get

\[
L(\Theta : D) = \prod_{m} \left( \prod_{x} P(E[m] : \Theta_x) \times P(B[m] : \Theta_x) \times P(A[m] | B[m], E[m] : \Theta_{AB,E}) \times P(C[m] | A[m] : \Theta_{C,A}) \right)
\]

- Parameters can be estimated for each variable independently!
**General Bayesian networks**

- Generalization for any Bayesian network:
  \[
  L(\Theta : D) = \prod_m P(x_i[m],...,x_n[m] : \Theta) \\
  = \prod_m \prod_i P(x_i[m] : Pa_i[m] : \Theta) \\
  = \prod_i L_i(\Theta_i : D)
  \]

  - Parameters can be estimated for each variable independently!

**Score-based learning**

- Define scoring function that measures how well a certain structure fits the observed data.

  - Search for a structure that maximizes the score.

**Unknown structure, complete data**

- Network structure is not specified
  - Learner needs to estimate both structure and parameters
- Data does not contain missing values

**Structure score**

- Likelihood score (function of S):
  \[
  P(D|S, \hat{\Theta}_S)
  \]

  - Problem: Adding an edge will always increase the likelihood score.
    - E.g., The likelihood score of G2 is always higher than that of G1

  - Maximum likelihood parameters

---

Score(G₁) = 10  Score(G₂) = 1.5  Score(G₃) = 0.01
Structure score

- Likelihood score (function of S):
  \[ P(D|S, \hat{\theta}_S) \]
  - Maximum likelihood parameters

- Bayesian score (function of S)
  - Average over all possible parameter values
  \[ P(D|S, \theta) = \int P(D|S, \theta)P(\theta|S)d\theta \]
  - Marginal likelihood
  - Likelihood
  - Prior distribution over parameters

- Penalized likelihood score (function of S and \(\theta_S\))
  \[ \log P(D|S, \theta_S) - C \cdot \text{model complexity}(\theta_S, D) \]

Search for optimal network structure

- Start with a given network structure.
  - Empty network
  - Best simple structure (e.g. tree)
  - A random network

- At each iteration
  - Evaluate all possible changes
  - Apply change based on score

- Stop when no modification improves the score.

Decomposability of scores

- Likelihood score
  \[ L(\Theta : D) = \prod_i L_i(\Theta_i : D) \]

- Bayesian score
  \[
  P(D|S) = \int P(D|S, \theta)P(\theta|S)d\theta \\
  = \int_{\theta_1 \ldots \theta_i} \prod_i \left( \prod_m P(x_i[m]|Pa_i[m]:\Theta_i) \right) P(\Theta_i : S) d\Theta_i \\
  = \prod_i \int_{\theta_i} \left( \prod_m P(x_i[m]|Pa_i[m]:\Theta_i) \right) P(\Theta_i : S)d\Theta_i \\
  = \prod_i \text{BayesianScore}(\Theta_i : D)
  \]

Search for optimal network structure

- Typical operations:
Search for optimal network structure

- Typical operations:
  - Add or delete edges
  - Score decomposability:
    At each iteration only need to score the site that is being updated!

Challenges

- Too large search space
  - For a network with \( n \) genes, what is the number of possible structures?
    \( \sim 3^{n^2/2} \)
- Computationally costly
- Heuristic approaches may be trapped to local maxima.
- Biologically motivated constraints can alleviate the problems
  - Module-based approach
  - Only a certain set of genes can be parents of other variables

Learning module networks

- Iterative procedure
  - Cluster genes into modules (E-step)
  - Learn regulatory programs for modules (M-step)

How about the linear CPD?

<table>
<thead>
<tr>
<th>Activator X3</th>
<th>Repressor X4</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>false</td>
</tr>
<tr>
<td>true</td>
<td>true</td>
</tr>
</tbody>
</table>

Linear CPD
Module Networks*

- Learning quickly runs out of statistical power
- Poor regulator selection lower in the tree
- Many correct regulators not selected
- Arbitrary choice among correlated regulators
- Combinatorial search
- Multiple local optima

* Segal et al., Nature Genetics 2003

Regulation as Linear Regression

\[
\min \, w (w_1 x_1 + \ldots + w_N x_N - E_{\text{Module}})^2
\]

- But we often have very large \( N \)
- ... and linear regression gives them all nonzero weight!

Problem: This objective learns too many regulators

Lasso* (L\(_1\)) Regression

\[
\minimize_w \, (w_1 x_1 + \ldots + w_N x_N - E_{\text{Module}})^2 + \sum C |w_i|
\]

- Induces sparsity in the solution \( w \) (many \( w_i \)'s set to zero)
- Provably selects "right" features when many features are irrelevant
- Convex optimization problem
  - No combinatorial search
  - Unique global optimum
  - Efficient optimization

* Tibshirani, 1996

Learning Regulatory Network

- Cluster genes into modules
- Learn a regulatory program for each module

Lee et al., PLoS Genet 2009
Learning the regulatory network

Multiple regression tasks

\[
\begin{align*}
\text{minimize}_{w_1} & \left( \sum w_{1n} x_n - E_{\text{module 1}} \right)^2 + \sum C|w_{1n}| \\
\text{minimize}_{w_n} & \left( \sum w_{nn} x_n - E_{\text{module } M} \right)^2 + \sum C|w_{nn}|
\end{align*}
\]

Learning module networks

Iterative procedure

- Cluster genes into modules (E-step)
- Learn regulatory programs for modules (M-step)

Outline

- Bayesian network learning to infer gene regulatory networks
  - Parameter estimation
  - Structure learning
  - Example – learning module networks

- Evaluation of the method
  - Statistical evaluation
  - Biological interpretation

- Structure discovery in BNs
  - Model averaging
  - Bootstrapping

Statistical Evaluation

- Cross-validation test
  - Divide the data (experiments) into training and test data
  - Compute the likelihood function for the Test data

\[
\text{Maximum increase in the penalized likelihood score}
\]

\[
\begin{align*}
\text{Candidate regulators} \\
\text{Linear CPD}
\end{align*}
\]

\[
\begin{align*}
\text{Candidate regulators} \\
\text{Linear CPD}
\end{align*}
\]
Module Evaluation Criteria

- Are the module genes functionally coherent?
- Do the regulators have regulatory roles in the predicted conditions C (see slide 6)?
- Are the genes in the module known targets of the predicted regulators?
- Are the regulators consistent with the cis-regulatory motifs (TF binding sites) found in promoters of the module genes?

Functional Coherence

- How significant is the overlap?
  - Calculate \( P(\text{# overlap} \geq k \mid K, n, N; \text{two groups are independent}) \) based on the hypergeometric distribution

Module Functional Coherence

- 26 Modules >60% Coherent
- 41 Modules >40% Coherent

- **Metabolic:** AA, respiration, glycolysis, galactose
- **Stress:** Oxidative stress, osmotic stress
- **Cellular localization:** Nucleus, ER
- **Cellular processes:** Cell cycle, sporulation, mating
- **Molecular functions:** Protein folding, RNA & DNA processing, trafficking

Respiration Module

HAP4 known to up regulate Oxid. Phos.

HAP4, MSN4, XBP1 known to be regulators under predicted conditions

HAP4 Binding site found in 39/55 genes

Outline

- Bayesian network learning to infer gene regulatory networks
  - Parameter estimation
  - Structure learning
  - Example – learning module networks

- Evaluation of the method
  - Statistical evaluation
  - Biological interpretation

- Structure discovery in BNs
  - Model averaging
  - Bootstraping
**Structure discovery**

- **Task:** Discover structural properties
  - Is there a direction connection between X and Y?
  - Does X separate between two “subsystems”?
  - Does X causally affect Y?

- **Example:** scientific data mining
  - Disease properties and symptoms
  - Interactions between the expression of genes

**Model averaging**

- There may be many high-scoring models
- Answer should not be based on any single model
- Want to average over many models

Define a structural feature \( f(S) \) of a model \( S \).
- For example:
  \[
  f(S) = \begin{cases} 
  1 & \text{if a graph } S \text{ has } A \rightarrow C \\
  0 & \text{otherwise}
  \end{cases}
  \]

We are interested in computing
\[
E_{P(S|D)}[f(S)] = \sum_S f(S)P(S \mid D)
\]

**AN ALTERNATIVE METHOD...**
Bootstrapping

- Sampling with replacement

**Original data**

Bootstrapping data 1  data 2  ...  data N

Inferring sub-networks from perturbed expression profiles, Pe’er et al. Bioinformatics 2001

---

Model selection problem

- Which model do we think is the most likely?

**D** = Gene A Gene B Gene C

**RNA expression levels**

- Low expression level
- High expression level

**N instances**

**E_A** **E_B** **E_C**

Model I  Model II  Model III

**ET’S REVISIT THE MODEL SELECTION PROBLEM.**
Model selection problem

- Which model do we think is the most likely?
- Given data $D$, let’s solve $\arg\max_x P(\text{Model } x \text{ is true} | D)$

\[
P(\text{Model I is true} | D) \times P(D | \text{Model I is true}) \times P(\text{Model I is true})
\]
\[
P(\text{Model II is true} | D) \times P(D | \text{Model II is true}) \times P(\text{Model II is true})
\]
\[
P(\text{Model III is true} | D) \times P(D | \text{Model III is true}) \times P(\text{Model III is true})
\]

Which model do we think is the most likely?

Given data $D$, let’s solve $\arg\max_x P(\text{Model } x \text{ is true} | D)$

$P(D | \text{Model I is true}) P(\text{Model I is true})$

$P(D | \text{Model II is true}) P(\text{Model II is true})$

$P(D | \text{Model III is true}) P(\text{Model III is true})$

$D = A[l_1, l_2, l_3, l_4, l_5, ...]$

$B[l_1, l_2, l_3, l_4, l_5, ...]$

$C[l_1, l_2, l_3, l_4, l_5, ...]$