MLE in Bayesian networks

- Likelihood function

\[ L(\theta : D) = P(D|Data[1], ..., Data[M]) \theta| D| I| G| D| S| L \]

\[ = \prod_{m=1}^{M} P(D = d[m]; \theta, I = i[m]; G = g[m]; S = s[m]; L = l[m]) \theta_{d|m} \theta_{i|m} \theta_{g|m} \theta_{s|m} \theta_{l|m} \]

\[ \text{If } d[m] = d_1, \theta_{d|m} = \theta_{d_1} \]

\[ \text{If } s[m] = s_1 \text{ and } i[m] = i_0, \theta_{s|m} = \theta_{s_1|i_0} \]
Bayesian Network with table CPDs

The Thumbtack example

\[ X \text{ vs } \theta \hat{=} \theta_0, \theta_D, \theta_{G|D} \]

Joint distribution

\[ P(X) \]

Parameters

\[ 0 \]

Data

\[ D: \{H...x[m]...T\} \]

Likelihood function

\[ \theta^{m(1-\theta)^{(n-m)}} \]

MLE solution

\[ \hat{\theta} = \frac{M_1}{M_1 + M_i} \]

The Student example

\[ \text{Intelligence} \rightarrow \text{Difficulty} \rightarrow \text{Grade} \]

Data

\[ D: \{(i,d,g)(i[m],d[m],g[m])\} \]

Likelihood function

\[ L(\theta|D) = P(D|\theta) \]

MLE solution

\[ \hat{\theta} = \frac{M_{i|D,d,g}}{M_i} \]

Outline

- Bayesian network representation of regulatory networks
- Bayesian network learning to infer gene regulatory networks
  - Parameter estimation
  - Structure learning

Regulatory network

- Bayesian network representation
  - \( X_i \): expression level of gene i
  - \( \text{Val}(X_i) \): continuous

- Joint distribution
  - \( P(X) \)

- Interpretation
  - Conditional independence

Conditional probability distribution (CPD)?

<table>
<thead>
<tr>
<th>Parameters</th>
<th>X5=high</th>
<th>X5=low</th>
</tr>
</thead>
<tbody>
<tr>
<td>X3=high, X4=high</td>
<td>0.3</td>
<td>0.7</td>
</tr>
<tr>
<td>X3=high, X4=low</td>
<td>0.95</td>
<td>0.05</td>
</tr>
<tr>
<td>X3=low, X4=high</td>
<td>0.1</td>
<td>0.9</td>
</tr>
<tr>
<td>X3=low, X4=low</td>
<td>0.2</td>
<td>0.8</td>
</tr>
</tbody>
</table>

CPD for discrete expression level

- After discretizing the expression levels to "high" and "low"...
  - Parameters – probability values in every entry
Outline

- Bayesian network representation of regulatory networks
- Bayesian network learning to infer gene regulatory networks
  - Parameter estimation
  - Structure learning

Learning parameters

- Training data has the form:

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

Known structure, complete data

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

Likelihood function

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

\[ L(\Theta : D) = \prod_m P(E[m], B[m], A[m], C[m] : \Theta) \]
### Likelihood function

- Joint distribution can be decomposed as:

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

### General Bayesian networks

- Generalization for any Bayesian network:

\[
L(\Theta : D) = \prod_{m} P(x_i[m], \ldots, x_n[m] : \Theta)
\]

\[
= \prod_{m} \prod_{i} P(x_i[m] | Pa_i[m] : \Theta)
\]

\[
= \prod_{i} L_i(\Theta_i : D)
\]

- Parameters can be estimated for each variable independently!

### Likelihood function

- Reordering terms, we got

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

\[
= \prod_{m} \left( P(E[m] : \Theta_e) \times P(B[m] : \Theta_b) \times P(A[m]|B[m], E[m] : \Theta_{a,b,e}) \times P(C[m]|A[m] : \Theta_{c,a}) \right)
\]

- Parameters can be estimated for each variable independently!

### Unknown structure, complete data

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

Structure score

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

- 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.
Search for optimal network structure

- Typical operations:
  - Add $C \rightarrow D$
  - Delete $C \rightarrow D$
  - Score decomposability: At each iteration only need to score the site that is being updated!

$\Delta \text{score} = S(\{C,E\} \rightarrow D) - S(\{E\} \rightarrow D)$

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

Model selection problem

- Which model do we think is the most likely?

Let’s revisit the model selection problem.
Model selection problem

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

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

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

**Model selection problem**

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

\[
P(\text{Model I} \mid D) P(D \mid \text{Model I} \mid D) P(\text{Model I})
\]
\[
P(\text{Model II} \mid D) P(D \mid \text{Model II} \mid D) P(\text{Model II})
\]
\[
P(\text{Model III} \mid D) P(D \mid \text{Model III} \mid D) P(\text{Model III})
\]

**Model selection problem**

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

\[
P(E_A) P(E_B \mid E_A) P(E_C \mid E_A) P(E_A) P(E_B \mid E_A) P(E_C \mid E_A)
\]
\[
P(E_A) P(E_B \mid E_A) P(E_C \mid E_A) P(E_A) P(E_B \mid E_A) P(E_C \mid E_A)
\]
\[
P(E_A) P(E_C \mid E_A) P(E_B \mid E_A) P(E_A) P(E_C \mid E_A) P(E_B \mid E_A)
\]

\[
P(E_A) P(E_B \mid E_A) P(E_C \mid E_A) P(E_A) P(E_B \mid E_A) P(E_C \mid E_A)
\]
\[
P(E_A) P(E_B \mid E_A) P(E_C \mid E_A) P(E_A) P(E_B \mid E_A) P(E_C \mid E_A)
\]

\[
P(E_A) P(E_C \mid E_A) P(E_B \mid E_A) P(E_A) P(E_C \mid E_A) P(E_B \mid E_A)
\]

\[
P(E_A) P(E_B \mid E_A) P(E_C \mid E_A) P(E_A) P(E_B \mid E_A) P(E_C \mid E_A)
\]

\[
P(E_A) P(E_C \mid E_A) P(E_B \mid E_A) P(E_A) P(E_C \mid E_A) P(E_B \mid E_A)
\]
Model selection problem

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

\[
P(D | \text{Model I is true}) = \prod_i P(E_A = A[i]) P(E_B = B[i] | E_A = A[i]) P(E_C = C[i] | E_A = A[i])
\]

\[
P(D | \text{Model II is true}) = \prod_i P(E_A = A[i]) P(E_B = B[i] | E_A = A[i]) P(E_C = C[i] | E_A = A[i])
\]

\[
P(D | \text{Model III is true}) = \prod_i P(E_A = A[i]) P(E_B = B[i]) P(E_C = C[i] | E_A = A[i])
\]