Bayesian Networks Definition

A Bayes network represents the joint probability distribution over a collection of random variables.

A Bayes network is a directed acyclic graph and a set of conditional probability distributions (CPD’s):
- Each node denotes a random variable
- Edges denote dependencies
- For each node $X_i$ its CPD defines $P(X_i | Pa(X_i))$
- The joint distribution over all variables is defined to be

$$P(X_1 \ldots X_n) = \prod_i P(X_i | Pa(X_i))$$

Pa(X) = immediate parents of X in the graph
Bayesian Network

What can we say about conditional independencies in a Bayes Net?

One thing is this:
Each node is conditionally independent of its non-descendents, given only its immediate parents.

| Parents | $P(W|Pa)$ | $P(\neg W|Pa)$ |
|---------|-----------|---------------|
| $L, R$  | 0         | 1.0           |
| $L, \neg R$ | 0   | 1.0          |
| $\neg L, R$ | 0.2 | 0.8         |
| $\neg L, \neg R$ | 0.9 | 0.1         |

What You Should Know

- Bayes nets are convenient representation for encoding dependencies / conditional independence
- BN = Graph plus parameters of CPD’s
  - Defines joint distribution over variables
  - Can calculate everything else from that
  - Though inference may be intractable
- Reading conditional independence relations from the graph
  - Each node is cond indep of non-descendents, given only its parents

See Bayes Net applet: http://www.cs.cmu.edu/~javabayes/Home/applet.html
Inference in Bayes Nets

- In general, intractable (NP-complete)
- For certain cases, tractable
  - Assigning probability to fully observed set of variables
  - Or if just one variable unobserved
  - Or for singly connected graphs (i.e., no undirected loops)
    - Belief propagation
- For multiply connected graphs
  - Junction tree
- Sometimes use Monte Carlo methods
  - Generate many samples according to the Bayes Net distribution, then count up the results
- Variational methods for tractable approximate solutions

Example

- Bird flu and Allegies both cause Sinus problems
- Sinus problems cause Headaches and runny Nose
Prob. of joint assignment: easy

- Suppose we are interested in joint assignment \(<F=f,A=a,S=s,H=h,N=n>\)

What is \(P(f,a,s,h,n)\)?

Let's use \(p(a,b)\) as shorthand for \(p(A=a, B=b)\)

Prob. of marginals: not so easy

- How do we calculate \(P(N=n)\)?

Let's use \(p(a,b)\) as shorthand for \(p(A=a, B=b)\)
Generating a sample from joint distribution: easy

How can we generate random samples drawn according to \( P(F,A,S,H,N) \)?

Let’s use \( p(a,b) \) as shorthand for \( p(A=a, B=b) \).

Generating a sample from joint distribution: easy

Note we can estimate marginals like \( P(N=n) \) by generating many samples from joint distribution, then count the fraction of samples for which \( N=n \).

Similarly, for anything else we care about
\[ P(F=1|H=1, N=0) \]

\( \rightarrow \) weak but general method for estimating any probability term…
Prob. of marginals: not so easy
But sometimes the structure of the network allows us to be clever → avoid exponential work

eg., chain

\[ \text{A} \rightarrow \text{B} \rightarrow \text{C} \rightarrow \text{D} \rightarrow \text{E} \]

Inference in Bayes Nets

- In general, intractable (NP-complete)
- For certain cases, tractable
  - Assigning probability to fully observed set of variables
  - Or if just one variable unobserved
  - Or for singly connected graphs (ie., no undirected loops)
    - Variable elimination
    - Belief propagation
- For multiply connected graphs
  - Junction tree
- Sometimes use Monte Carlo methods
  - Generate many samples according to the Bayes Net distribution, then count up the results
- Variational methods for tractable approximate solutions
Learning of Bayes Nets

- Four categories of learning problems
  - Graph structure may be known/unknown
  - Variable values may be fully observed / partly unobserved

- Easy case: learn parameters for graph structure is known, and data is fully observed

- Interesting case: graph known, data partly known

- Gruesome case: graph structure unknown, data partly unobserved

Learning CPTs from Fully Observed Data

- Example: Consider learning the parameter
  \[ \theta_{s|ij} = P(S = 1|F = i, A = j) \]

- MLE (Max Likelihood Estimate) is
  \[ \hat{\theta}_{s|ij} = \frac{\sum_{k=1}^{K} \delta(f_k = i, a_k = j, s_k = 1)}{\sum_{k=1}^{K} \delta(f_k = i, a_k = j)} \]

- Remember why?
MLE estimate of $\theta_{j|i}$ from fully observed data

- Maximum likelihood estimate
  $\hat{\theta} = \arg \max_\theta \log P(\text{data}|\theta)$

- Our case:
  
  $P(\text{data}|\theta) = \prod_{k=1}^{K} P(f_k, a_k, s_k, h_k, n_k)$

  $P(\text{data}|\theta) = \prod_{k=1}^{K} P(f_k)P(a_k)P(s_k|f_k,a_k)P(h_k|s_k)P(n_k|s_k)$

  $\log P(\text{data}|\theta) = \sum_{k=1}^{K} \log P(f_k) + \log P(a_k) + \log P(s_k|f_k,a_k) + \log P(h_k|s_k) + \log P(n_k|s_k)$

  $\frac{\partial \log P(\text{data}|\theta)}{\partial \theta_{j|i}} = \sum_{k=1}^{K} \frac{\partial \log P(s_k|f_k,a_k)}{\partial \theta_{j|i}}$

  $\theta_{j|i} = \frac{\sum_{k=1}^{K} \delta(f_k = i, a_k = j, s_k = 1)}{\sum_{k=1}^{K} \delta(f_k = i, a_k = j)}$

---

Estimate $\theta$ from partly observed data

- What if FAHN observed, but not S?
- Can’t calculate MLE
  $\hat{\theta} = \arg \max_\theta \log \prod_k P(f_k, a_k, s_k, h_k, n_k|\theta)$

  - Let $X$ be all observed variable values (over all examples)
  - Let $Z$ be all unobserved variable values
  - Can’t calculate MLE:
    $\hat{\theta} = \arg \max_\theta \log P(X, Z|\theta)$

- WHAT TO DO?
Estimate $\theta$ from partly observed data

- What if FAHN observed, but not S?
- Can’t calculate MLE
  \[ \theta \leftarrow \arg \max_{\theta} \log \prod_k P(f_k, a_k, s_k, h_k, n_k|\theta) \]

- Let $X$ be all observed variable values (over all examples)
- Let $Z$ be all unobserved variable values
- Can’t calculate MLE:
  \[ \theta \leftarrow \arg \max_{\theta} \log P(X, Z|\theta) \]

- EM seeks* to estimate:
  \[ \theta \leftarrow \arg \max_{\theta} E_{Z|X,\theta}[\log P(X, Z|\theta)] \]
  * EM guaranteed to find local maximum

- EM seeks estimate:
  \[ \theta \leftarrow \arg \max_{\theta} E_{Z|X,\theta}[\log P(X, Z|\theta)] \]

- here, observed $X=\{F, A, H, N\}$, unobserved $Z=\{S\}$

\[
\log P(X, Z|\theta) = \sum_{k=1}^{K} \log P(f_k) + \log P(a_k) + \log P(s_k|f_k, a_k) + \log P(h_k|s_k) + \log P(n_k|s_k)
\]

\[
E_{P(Z|X,\theta)} \log P(X, Z|\theta) = \sum_{k=1}^{K} \sum_{i=0}^{1} P(s_k = i | f_k, a_k, h_k, n_k) \left[ \log P(f_k) + \log P(a_k) + \log P(s_k|f_k, a_k) + \log P(h_k|s_k) + \log P(n_k|s_k) \right]
\]
EM Algorithm

EM is a general procedure for learning from partly observed data.

Given observed variables X, unobserved Z (X={F,A,H,N}, Z={S}),

Define $Q(\theta' | \theta) = E_{P(Z|X,\theta)}[\log P(X, Z|\theta')]$

Iterate until convergence:

- **E Step**: Use X and current $\theta$ to calculate $P(Z|X,\theta)$
- **M Step**: Replace current $\theta$ by
  $$\theta \leftarrow \arg \max_{\theta'} Q(\theta' | \theta)$$

Guaranteed to find local maximum.

Each iteration increases $E_{P(Z|X,\theta)}[\log P(X, Z|\theta')]$

---

**E Step: Use X, $\theta$, to Calculate P(Z|X,$\theta$)**

observed X={F,A,H,N},
unobserved Z={S}


$$P(S_k = 1| f_k a_k h_k n_k, \theta) =$$

$$P(S_k = 1| f_k a_k h_k n_k, \theta) = \frac{P(S_k = 1, f_k a_k h_k n_k | \theta)}{P(S_k = 1, f_k a_k h_k n_k | \theta) + P(S_k = 0, f_k a_k h_k n_k | \theta)}$$
EM and estimating $\theta_s|\epsilon, j$

observed $X = \{F, A, H, N\}$, unobserved $Z = \{S\}$

E step: Calculate $P(Z_k|X_k; \theta)$ for each training example, $k$

$P(S_k = 1|f_k a_k h_k n_k; \theta) = \frac{P(S_k = 1, f_k a_k h_k n_k; \theta)}{\sum_{Z_k \in \{1, 0\}} P(S_k = 1, f_k a_k h_k n_k; \theta) + P(S_k = 0, f_k a_k h_k n_k; \theta)}$

M step: update all relevant parameters. For example:

$\theta_{sij} \leftarrow \frac{\sum_{k=1}^{K} \delta(f_k = i, a_k = j) E[s_k]}{\sum_{k=1}^{K} \delta(f_k = i, a_k = j)}$

Recall MLE was: $\theta_{sij} = \frac{\sum_{k=1}^{K} \delta(f_k = i, a_k = j, s_k = 1)}{\sum_{k=1}^{K} \delta(f_k = i, a_k = j)}$

EM and estimating $\theta$

More generally, Given observed set $X$, unobserved set $Z$ of boolean values

E step: Calculate for each training example, $k$

the expected value of each unobserved variable

M step:

Calculate estimates similar to MLE, but replacing each count by its expected count

$\delta(Y = 1) \rightarrow E_{Z|X, \theta}[Y]$  \hspace{1cm} $\delta(Y = 0) \rightarrow (1 - E_{Z|X, \theta}[Y])$
Using Unlabeled Data to Help Train Naïve Bayes Classifier

Learn $P(Y|X)$

<table>
<thead>
<tr>
<th></th>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>X4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>?</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>?</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

E step: Calculate for each training example, $k$ the expected value of each unobserved variable.
EM and estimating $\theta$

Given observed set $X$, unobserved set $Y$ of boolean values

E step: Calculate for each training example, $k$ the expected value of each unobserved variable $Y$

$$E_{P(Y|X_1,...,X_N)}[y(k)] = P(y(k) = 1|x_1(k),...,x_N(k); \theta) = \frac{P(y(k) = 1) \prod_i P(x_i(k)|y(k) = 1)}{\sum_{j=0}^{1} P(y(k) = j) \prod_i P(x_i(k)|y(k) = j)}$$

M step: Calculate estimates similar to MLE, but replacing each count by its expected count

Let’s use $y(k)$ to indicate value of $Y$ on $k$th example.

EM and estimating $\theta$

Given observed set $X$, unobserved set $Y$ of boolean values

E step: Calculate for each training example, $k$ the expected value of each unobserved variable $Y$

$$E_{P(Y|X_1,...,X_N)}[y(k)] = P(y(k) = 1|x_1(k),...,x_N(k); \theta) = \frac{P(y(k) = 1) \prod_i P(x_i(k)|y(k) = 1)}{\sum_{j=0}^{1} P(y(k) = j) \prod_i P(x_i(k)|y(k) = j)}$$

M step: Calculate estimates similar to MLE, but replacing each count by its expected count

$$\theta_{ijm} = \hat{P}(X_i = j|Y = m) = \frac{\sum_k P(y(k) = m|x_1(k),...,x_N(k)) \delta(x_i(k) = j)}{\sum_k P(y(k) = m|x_1(k),...,x_N(k))}$$

MLE would be: $\hat{P}(X_i = j|Y = m) = \frac{\sum_k \delta((y(k) = m) \land (x_i(k) = j))}{\sum_k \delta(y(k) = m)}$
**Inputs:** Collections $\mathcal{D}^l$ of labeled documents and $\mathcal{D}^u$ of unlabeled documents.

- Build an initial naive Bayes classifier, $\hat{\theta}$, from the labeled documents, $\mathcal{D}^l$, only. Use maximum a posteriori parameter estimation to find $\hat{\theta} = \arg\max_\theta P(\mathcal{D}|\theta)P(\theta)$ (see Equations 5 and 6).
- Loop while classifier parameters improve, as measured by the change in $L(\theta|\mathcal{D}^l; \pi)$ (the complete log probability of the labeled and unlabeled data).
  
  - **(E-step)** Use the current classifier, $\hat{\theta}$, to estimate component membership of each unlabeled document, i.e., the probability that each mixture component (and class) generated each document, $P(c_j|d; \hat{\theta})$ (see Equation 7).
  
  - **(M-step)** Re-estimate the classifier, $\hat{\theta}$, given the estimated component membership of each document. Use maximum a posteriori parameter estimation to find $\hat{\theta} = \arg\max_\theta P(\mathcal{D}|\theta)P(\theta)$ (see Equations 5 and 6).

**Output:** A classifier, $\hat{\theta}$, that takes an unlabeled document and predicts a class label.

---

**Experimental Evaluation**

- Newsgroup postings
  - 20 newsgroups, 1000/group

- Web page classification
  - student, faculty, course, project
  - 4199 web pages

- Reuters newswire articles
  - 12,902 articles
  - 90 topics categories

---

From [Nigam et al., 2000]
20 Newsgroups

Table 3. Lists of the words most predictive of the course class in the WebKB data set, as they change over iterations of EM for a specific trial. By the second iteration of EM, many common course-related words appear. The symbol \( D \) indicates an arbitrary digit.

<table>
<thead>
<tr>
<th>Iteration 0</th>
<th>Iteration 1</th>
<th>Iteration 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>intelligence</td>
<td>( DD )</td>
<td>( D )</td>
</tr>
<tr>
<td>artificial understanding</td>
<td>( D )</td>
<td>lecture</td>
</tr>
<tr>
<td>( DD )</td>
<td>cc</td>
<td>lecture</td>
</tr>
<tr>
<td>dist</td>
<td>( D^r )</td>
<td>cc</td>
</tr>
<tr>
<td>identical</td>
<td>( DD:DD )</td>
<td>( DD:DD )</td>
</tr>
<tr>
<td>rus</td>
<td>handout</td>
<td>due</td>
</tr>
<tr>
<td>arrange</td>
<td>due</td>
<td>homework</td>
</tr>
<tr>
<td>games</td>
<td>problem</td>
<td>assignment</td>
</tr>
<tr>
<td>dartmouth</td>
<td>set</td>
<td>handout</td>
</tr>
<tr>
<td>natural</td>
<td>tay</td>
<td>set</td>
</tr>
<tr>
<td>cognitive logic</td>
<td>( DD:Dam )</td>
<td>hw</td>
</tr>
<tr>
<td>proving</td>
<td>yurttas</td>
<td>exam</td>
</tr>
<tr>
<td>prolog</td>
<td>homework</td>
<td>problem</td>
</tr>
<tr>
<td>knowledge</td>
<td>kfoury</td>
<td>( DD:Dam )</td>
</tr>
<tr>
<td>human representation</td>
<td>sec</td>
<td>postscript</td>
</tr>
<tr>
<td>field</td>
<td>postscript</td>
<td>solution</td>
</tr>
</tbody>
</table>

Using one labeled example per class
20 Newsgroups

Unsupervised clustering

Just extreme case for EM with zero labeled examples…
Clustering

• Given set of data points, group them
• Unsupervised learning
• Which patients are similar? (or which earthquakes, customers, faces, web pages, …)

Mixture Distributions

Model joint $P(X_1 \ldots X_n)$ as mixture of multiple distributions. Use discrete-valued random var $Z$ to indicate which distribution is being used for each random draw.

So $P(X_1 \ldots X_n) = \sum_i P(Z = i) P(X_1 \ldots X_n | Z)$

Mixture of Gaussians:
• Assume each data point $X=<X_1, \ldots X_n>$ is generated by one of several Gaussians, as follows:
  1. randomly choose Gaussian $i$, according to $P(Z=i)$
  2. randomly generate a data point $<x_1,x_2,\ldots x_n>$ according to $N(\mu_i, \Sigma_i)$
EM for Mixture of Gaussian Clustering

Let’s simplify to make this easier:
1. Assume \( X = \{X_1, \ldots, X_n\} \), and the \( X_j \) are conditionally independent given \( Z \).
   \[
P(X|Z = j) = \prod_i N(X_i|\mu_{ji}, \sigma_{ji})
   \]
2. Assume only 2 clusters (values of \( Z \)), and \( \forall i, j, \sigma_{ji} = \sigma \)
   \[
P(X) = \sum_{j=1}^{2} P(Z = j|\pi) \prod_i N(x_i|\mu_{ji}, \sigma)
   \]
3. Assume \( \sigma \) known, \( \pi_1 \ldots \pi_K, \mu_{ji} \ldots \mu_{Kj} \) unknown

Observed: \( X = \{X_1, \ldots, X_n\} \)
Unobserved: \( Z \)

---

**EM**

Given observed variables \( X \), unobserved \( Z \)
Define \( Q(\theta'|\theta) = E_{Z|X,\theta}[\log P(X, Z|\theta')] \)
where \( \theta = (\pi, \mu_{ji}) \)

Iterate until convergence:
- E Step: Calculate \( P(Z(n)|X(n), \theta) \) for each example \( X(n) \).
  Use this to construct \( Q(\theta'|\theta) \)
- M Step: Replace current \( \theta \) by
  \[
  \theta \leftarrow \arg \max_{\theta'} Q(\theta'|\theta)
  \]
**EM – E Step**

Calculate $P(Z(n)|X(n), \theta)$ for each observed example $X(n)$. 

$X(n)=<x_1(n), x_2(n), \ldots, x_T(n)>$. 

\[
P(z(n) = k|x(n), \theta) = \frac{P(x(n)|z(n) = k, \theta) P(z(n) = k|\theta)}{\sum_{j=0}^{1} p(x(n)|z(n) = j, \theta) P(z(n) = j|\theta)}
\]

\[
P(z(n) = k|x(n), \theta) = \frac{[\prod_i P(x_i(n)|z(n) = k, \theta)] P(z(n) = k|\theta)}{\sum_{j=0}^{1} \prod_i P(x_i(n)|z(n) = j, \theta) P(z(n) = j|\theta)}
\]

\[
P(z(n) = k|x(n), \theta) = \frac{[\prod_i N(x_i(n)|\mu_{k,i}, \sigma)] (\pi^k(1-\pi)^{1-k})}{\sum_{j=0}^{1} [\prod_i N(x_i(n)|\mu_{j,i}, \sigma)] (\pi^j(1-\pi)^{1-j})}
\]

---

**EM – M Step**

First consider update for $\pi$ 

$Q(\theta'|\theta) = E_{Z|X,\theta}[\log P(X,Z|\theta')] = E[\log P(X,Z|\theta') + \log P(Z|\theta')]$ 

$\pi \leftarrow \operatorname{arg\,max}_{\pi'} E_{Z|X,\theta}[\log P(Z|\pi')]$ 

$E_{Z|X,\theta}[\log P(Z|\pi')] = E_{Z|X,\theta} \left[ \log \left( \pi' \sum_n z(n)(1-\pi') \sum_n (1-z(n)) \right) \right]$ 

\[
= E_{Z|X,\theta} \left[ \left( \sum_n z(n) \right) \log \pi' + \left( \sum_n (1-z(n)) \right) \log(1-\pi') \right]
\]

\[
= \left( \sum_n E_{Z|X,\theta}[z(n)] \right) \log \pi' + \left( \sum_n E_{Z|X,\theta}[1-z(n)] \right) \log(1-\pi')
\]

\[
\frac{\partial E_{Z|X,\theta}[\log P(Z|\pi')]}{\partial \pi'} = \left( \sum_n E_{Z|X,\theta}[z(n)] \right) \frac{1}{\pi'} + \left( \sum_n E_{Z|X,\theta}[1-z(n)] \right) \frac{(-1)}{1-\pi'}
\]

\[
\pi \leftarrow \frac{\sum_{n=1}^{N} E[z(n)]}{\left( \sum_{n=1}^{N} E[z(n)] \right) + \left( \sum_{n=1}^{N} (1-E[z(n)]) \right)} = \frac{1}{N} \sum_{n=1}^{N} E[z(n)]
\]
Now consider update for $\mu_{ji}$

\[
Q(\theta' | \theta) = E_{Z|X,\theta}[\log P(X,Z|\theta')] = E[\log P(X|Z,\theta') + \log P(Z|\theta')]
\]

$\mu_{ji}'$ has no influence

\[
\mu_{ji} \leftarrow \arg \max_{\mu_{ji}'} E_{Z|X,\theta}[\log P(X|Z,\theta')]
\]

\[
\mu_{ji} \leftarrow \frac{\sum_{n=1}^{N} P(z(n) = j|x(n), \theta) \cdot x_i(n)}{\sum_{n=1}^{N} P(z(n) = j|x(n), \theta)}
\]

Compare above to MLE if $Z$ were observable:

\[
\mu_{ji} \leftarrow \frac{\sum_{n=1}^{N} \delta(z(n) = j) \cdot x_i(n)}{\sum_{n=1}^{N} \delta(z(n) = j)}
\]

EM – putting it together

Given observed variables $X$, unobserved $Z$

Define $Q(\theta' | \theta) = E_{Z|X,\theta}[\log P(X,Z|\theta')]$

where $\theta = (\pi, \mu_{ji})$

Iterate until convergence:

- **E Step:** For each observed example $X(n)$, calculate $P(Z(n)|X(n), \theta)$

\[
P(z(n) = k | x(n), \theta) = \frac{\prod_i N(x_i(n)|\mu_{k,i}, \sigma)}{\sum_{i=0}^{K} \prod_i N(x_i(n)|\mu_{j,i}, \sigma)} \cdot (\pi^k(1-\pi)^{(1-k)})
\]

- **M Step:** Update $\theta \leftarrow \arg \max_{\theta'} Q(\theta' | \theta)$

\[
\pi \leftarrow \frac{1}{N} \sum_{n=1}^{N} E[z(n)]; \quad \mu_{ji} \leftarrow \frac{\sum_{n=1}^{N} P(z(n) = j|x(n), \theta) \cdot x_i(n)}{\sum_{n=1}^{N} P(z(n) = j|x(n), \theta)}
\]
Mixture of Gaussians applet

Go to: [http://www.socr.ucla.edu/htmls/SOCR_Charts.html](http://www.socr.ucla.edu/htmls/SOCR_Charts.html)
then go to Go to “Line Charts” → SOCR EM Mixture Chart
• try it with 2 Gaussian mixture components (“kernels”)
• try it with 4

What you should know about EM

• For learning from partly unobserved data
• MLE of $\theta = \arg \max_{\theta} \log P(data|\theta)$
• EM estimate: $\theta = \arg \max_{\theta} E_{Z|X,\theta}[\log P(X, Z|\theta)]$
  Where X is observed part of data, Z is unobserved

• EM for training Bayes networks
• Can also develop MAP version of EM
• Can also derive your own EM algorithm for your own problem
  – write out expression for $E_{Z|X,\theta}[\log P(X, Z|\theta)]$
  – E step: for each training example $X^k$, calculate $P(Z^k | X^k, \theta)$
  – M step: chose new $\theta$ to maximize $E_{Z|X,\theta}[\log P(X, Z|\theta)]$
Learning Bayes Net Structure

How can we learn Bayes Net graph structure?

In general case, open problem
• can require lots of data (else high risk of overfitting)
• can use Bayesian methods to constrain search

One key result:
• Chow-Liu algorithm: finds “best” tree-structured network
• What’s best?
  – suppose \( P(X) \) is true distribution, \( T(X) \) is our tree-structured network, where \( X = \langle X_1, \ldots, X_n \rangle \)
  – Chow-Liu minimizes Kullback-Leibler divergence:

\[
KL(P(X) \mid\mid T(X)) = \sum_k P(X = k) \log \frac{P(X = k)}{T(X = k)}
\]
Chow-Liu Algorithm

Key result: To minimize $KL(P \| T)$, it suffices to find the tree network $T$ that maximizes the sum of mutual informations over its edges.

Mutual information for an edge between variable $A$ and $B$:

$$I(A, B) = \sum_a \sum_b P(a, b) \log \frac{P(a, b)}{P(a)P(b)}$$

This works because for tree networks with nodes $X \equiv \langle X_1 \ldots X_n \rangle$

$$KL(P(X) \| T(X)) = \sum_k P(X = k) \log \frac{P(X = k)}{T(X = k)}$$

$$= - \sum_i I(X_i, Pa(X_i)) + \sum_i H(X_i) - H(X_1 \ldots X_n)$$

---

Chow-Liu Algorithm

1. for each pair of vars $A, B$, use data to estimate $P(A, B)$, $P(A)$, $P(B)$

2. for each pair of vars $A, B$ calculate mutual information

$$I(A, B) = \sum_a \sum_b P(a, b) \log \frac{P(a, b)}{P(a)P(b)}$$

3. calculate the maximum spanning tree over the set of variables, using edge weights $I(A, B)$
   (given $N$ vars, this costs only $O(N^2)$ time)

4. add arrows to edges to form a directed-acyclic graph

5. learn the CPD’s for this graph
Chow-Liu algorithm example
Greedy Algorithm to find Max-Spanning Tree

Bayes Nets – What You Should Know

• Representation
  – Bayes nets represent joint distribution as a DAG + Conditional Distributions
  – D-separation lets us decode conditional independence assumptions

• Inference
  – NP-hard in general
  – For some graphs, closed form inference is feasible
  – Approximate methods too, e.g., Monte Carlo methods, …

• Learning
  – Easy for known graph, fully observed data (MLE’s, MAP est.)
  – EM for partly observed data, known graph
  – Learning graph structure: Chow-Liu for tree-structured networks
  – Hardest when graph unknown, data incompletely observed