HMMs

Machine Learning – 10701/15781
Carlos Guestrin
Carnegie Mellon University

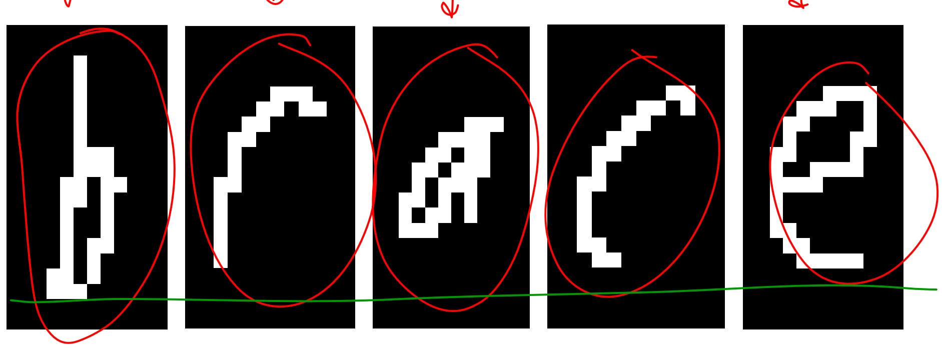
November 7th, 2007

Example of a hidden Markov model (HMM)

$P(X_3|X_2) \leftarrow P(X_3=a|X_2=r)$
prob an a follows an r in third position

$\{a,\dots,z\}$

$X_1 \rightarrow X_2 \rightarrow X_3 \rightarrow X_4 \rightarrow X_5$



Understanding the HMM Semantics

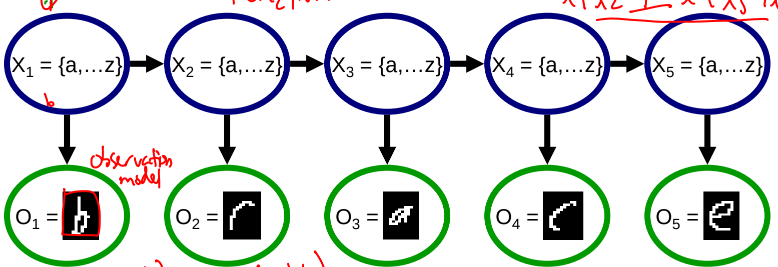
$P(X_1)$ prob. words start with letter x_1

← prob. second letter is x_2 given first is x_1

$P(X_2 | X_1)$

$X_1 X_2 \perp X_4 X_5 \mid X_3$

Markov model



if X_3 is the present time (now)

past indep. of future given present

$P(O_1 | X_1 = b)$

$P(O_2 | X_2)$

$O_2 \perp$ everybody $\mid X_2$

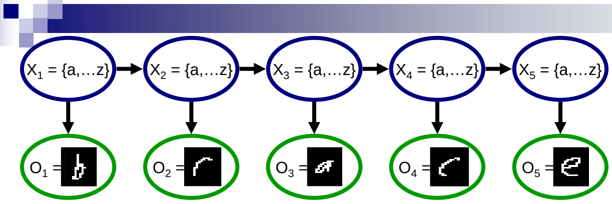
$X_3 \perp X_1 \mid X_2$

$X_1 \perp \{X_3, X_4, X_5 \ldots\} \mid X_2$

$\boxed{a}$ …

…

HMMs semantics: Details



Just 3 distributions:

$$P(X_1)$$ ← starting state dist

$$P(X_i \mid X_{i-1})$$ ← transition model

$$P(O_i \mid X_i)$$ ← observation model

usual assumption

stationarity:

obs:

$P(\boxed{b} \mid X_i = b)$

$= P(\boxed{b} \mid X_j = b) \quad i \neq j$

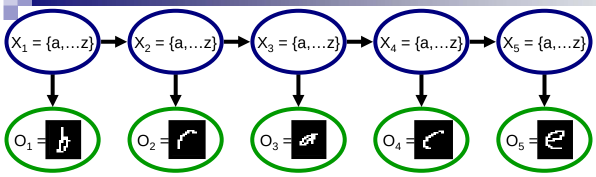
transition model:

$P(X_i = a \mid X_{i-1} = b)$

$= P(X_j = a \mid X_{j-1} = b)$

$i \neq j$

HMMs semantics: Joint distribution



$P(X_1)$

$P(X_i \mid X_{i-1})$

$P(O_i \mid X_i)$

$P(X_1, \ldots, X_n, O_1, \ldots O_n) = P(X_1) \cdot P(O_1|X_1) \cdot P(X_2|X_1) \cdot P(O_2|X_2) \ldots$

$= P(X_1) \cdot P(O_1|X_1) \prod_{t=2}^{n} P(X_t|X_{t-1}) \, P(O_t|X_t)$

Given $O = \{b, c, a, c, e\}$

$$\underline{P(X_1, \ldots, X_n} \mid \underline{o_1, \ldots, o_n)} = P(X_{1:n} \mid o_{1:n})$$

$$\propto P(X_1)P(\underline{o_1} \mid X_1) \prod_{i=2}^{n} P(X_i \mid X_{i-1})P(\underline{o_i} \mid X_i)$$

Learning HMMs from fully observable data is easy

$P(O_i = \boxed{b} \mid X_i = b)$

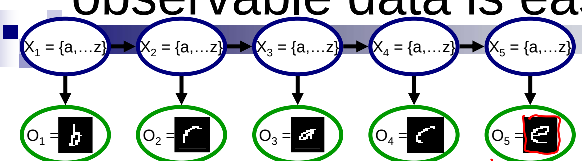
$= \prod_{j \in \text{Pixels}} P(O_{ij} = \ldots \mid X_i = b)$

NB

X_i

O_{i1} O_{i2} … O_i

e



Learn 3 distributions:

first letter of each word

$P(X_1 = b) = \dfrac{\text{Count}(X_1 = b)}{m}$

only see $\boxed{b}$ once…

need make some assumption about $P(O_i|X_i)$

$P(O_i = \boxed{b} \mid X_i = b) = \dfrac{\text{Count}(X_i = b, O_i = \boxed{b})}{\text{Count}(X_i = b)}$

# of words

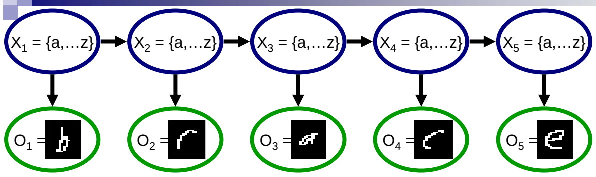
how often a b follows an a

$P(X_i = b \mid X_{i-1} = a) = \dfrac{\text{Count}(X_i = b, X_{i-1} = a)}{\text{Count}(X_{i-1} = a)}$

$\approx \dfrac{P(X_i = b, X_{i-1} = a)}{P(X_{i-1} = a)}$

how often I see an a, but not in last position in word

Possible inference tasks in an HMM



forwards-backwards

Marginal probability of a hidden variable:

$P(X_3 = b \mid O_1 = b, O_2 = r, a, c, e)$

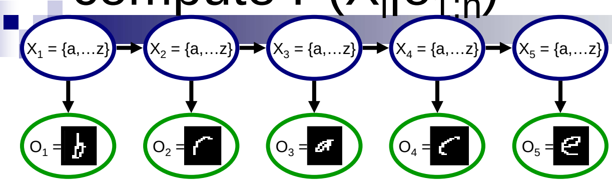
$P(X_3 \mid O_1, .. O_5)$

Viterbi decoding – most likely trajectory for hidden vars:

$\arg\max_{x_1, x_2, x_3, x_4, x_5} P(x_1, \ldots x_5 \mid o_1 = b, r, a, c, e)$

maximum likelihood for entire word

Using variable elimination to compute $P(X_i|o_{1:n})$



Compute:

$$\underline{P(X_i \mid o_{1..n})}$$

complexity

$O(n)$

Variable elimination order? $P(X_3 \mid O_{1:5})$

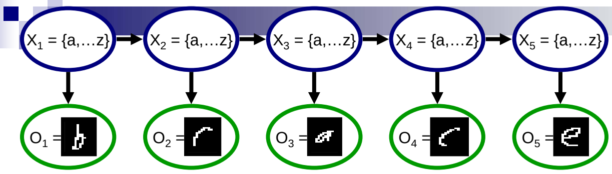
1, 2, 4, 5

Example:

$P(X_3, o_{1:5}) = \sum_{x_1 x_2 x_4 x_5} P(x_1) P(o_1|x_1) P(x_2|x_1) P(o_2|x_2) \ldots\ldots$

$= \sum_{x_2 x_4 x_5} P(o_2|x_2) P(x_3|x_2) P(o_3|x_3) \cdots \underbrace{\sum_{x_1} P(x_1) . P(o_1|x_1) . P(x_2|x_1)}_{P(x_2, o_1) = g_1(x_2)}$

What if I want to compute $P(X_i|o_{1:n})$ for each i?



Compute:

$$P(X_i \mid o_{1..n})$$

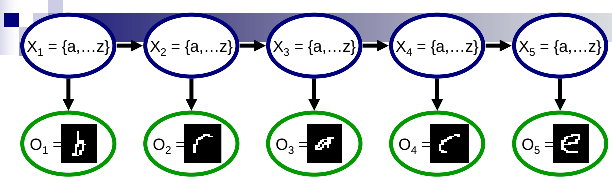
Variable elimination for ~~each~~ one i? $O(n)$

Variable elimination for each i, what's the complexity?

↳ naively ← $O(n^2)$

but if you are smart, you can do it in $O(n)$

Reusing computation



Compute:

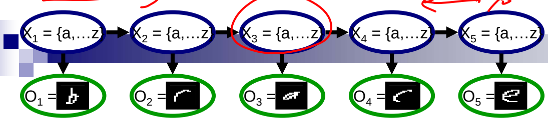
$$P(X_i \mid o_{1..n})$$

$$P(X_5, o_{1:5}) = \sum_{x_1 .. x_4} P(x_1)\,P(o_1|x_1)\,P(x_2|x_1)\,P(o_2|x_2)\dots$$

$$= \sum_{x_2 x_3 x_4} P(o_2|x_2)\dots \underbrace{\sum_{x_1} P(x_1)\,P(o_1|x_1)\,P(x_2|x_1)}_{g(x_2)}$$

$$P(X_4, o_{1:5}) = \sum_{x_2, x_3, x_5} P(o_2|x_2)\dots \overbrace{\sum_{x_1} P(x_1)\,P(o_1|x_1)\,P(x_2|x_1)}$$

The forwards-backwards algorithm



$$P(X_i \mid o_{1..n})$$

- Initialization: $\alpha_1(X_1) = P(X_1)P(o_1 \mid X_1)$ forwards
- For i = 2 to n
  - Generate a forwards factor α_i by eliminating X_{i-1}

$$\alpha_i(X_i) = \sum_{x_{i-1}} P(o_i \mid X_i)P(X_i \mid X_{i-1} = x_{i-1})\alpha_{i-1}(x_{i-1})$$

- Initialization: $\beta_n(X_n) = 1$
- For i = n-1 to 1
  - Generate a backwards factor β_i by eliminating X_{i+1}

$$\beta_i(X_i) = \sum_{x_{i+1}} P(o_{i+1} \mid x_{i+1})P(x_{i+1} \mid X_i)\beta_{i+1}(x_{i+1})$$

- ∀ i, probability is: $\boxed{P(X_i \mid o_{1..n}) \propto \alpha_i(X_i)\beta_i(X_i)}$

What you'll implement 1: multiplication

$$\alpha_i(X_i) = \sum_{x_{i-1}} P(o_i \mid X_i)P(X_i \mid X_{i-1} = x_{i-1})\alpha_{i-1}(x_{i-1})$$

$f(X_i = a, X_{i-1} = b) = P(o_i \mid X_i = a) \cdot P(X_i = a \mid X_{i-1} = b) \cdot \alpha_{i-1}(X_{i-1} = b)$

table for $f(X_i, X_{i-1})$

26

X_i \ X_{i-1}	a	b	c	…	z
a					
b					
⋮					
z					

26

What you'll implement 2: marginalization

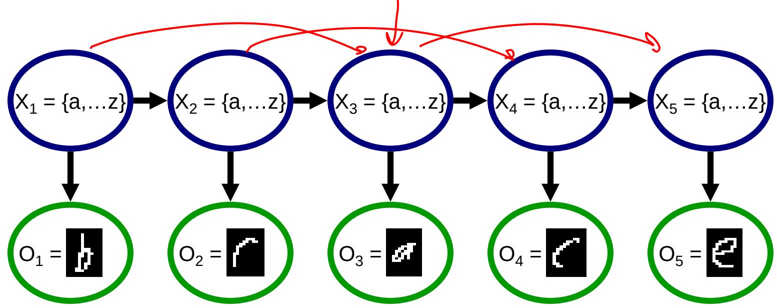
$$\alpha_i(X_i) = \sum_{x_{i-1}} \underbrace{P(o_i \mid X_i) P(X_i \mid X_{i-1} = x_{i-1})}_{f(X_i, X_{i-1})} \alpha_{i-1}(x_{i-1})$$

$$\alpha_i(X_i = a) = \sum_{x_{i-1}} f(X_i = a, X_{i-1} = x_{i-1})$$

Higher-order HMMs

more params to learn but could do better (potentially)

$P(X_3 \mid X_1, X_2)$



1st order HMM

Add dependencies further back in time !

better representation, harder to learn

What you need to know

- Hidden Markov models (HMMs)
  - Very useful, very powerful!
  - Speech, OCR,…
  - Parameter sharing, only learn 3 distributions
  - Trick reduces inference from $O(n^2)$ to $O(n)$
  - Special case of BN

Kalman Filter: HMM, with $\left.\begin{array}{l} P(X_i \mid X_{i-1}) \\ P(O_i \mid X_i) \end{array}\right\}$ conditional Gaussian

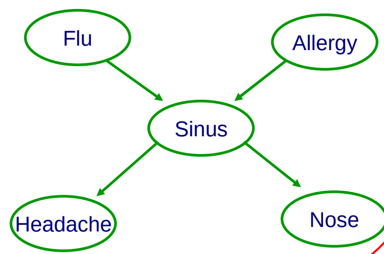
Bayesian Networks (Structure) Learning

Machine Learning – 10701/15781
Carlos Guestrin
Carnegie Mellon University

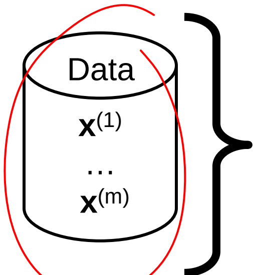
November 7^{th}, 2007

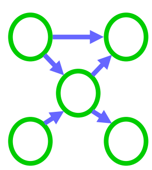
Review

- Bayesian Networks
  - Compact representation for probability distributions
  - Exponential reduction in number of parameters
- Fast probabilistic inference using variable elimination
  - Compute P(X|e)
  - Time exponential in tree-width, not number of variables
- Today
  - Learn BN structure



Learning Bayes nets





structure

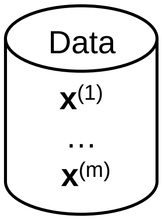
+

CPTs – $P(X_i|\, \mathbf{Pa}_{X_i})$

parameters

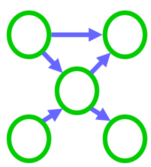
Learning the CPTs

$p(s \mid f, a)$



For each discrete variable X_i

$$P(X_i = x_i \mid Pa_{X_i} = u) \overset{MLE}{=} \frac{Count(X_i = x_i, Pa_{X_i} = u)}{Count(Pa_{X_i} = u)}$$



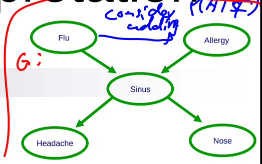
MLE: $P(X_i = x_i \mid X_j = x_j) = \dfrac{\text{Count}(X_i = x_i, X_j = x_j)}{\text{Count}(X_j = x_j)}$

Information-theoretic interpretation of maximum likelihood 1

$\log \prod = \sum \log$

$\log ab = \log a + \log b$

G:



consider coding $P(A \mid F)$

params Graph

iid

given G find θ

- Given structure, log likelihood of data:

$$\log P(\mathcal{D} \mid \theta_{\mathcal{G}}, \mathcal{G}) = \log \prod_{i=1}^{m} P(x^{(i)} \mid \theta_G, G) = \sum_i \log P(x^{(i)} \mid \theta_G, G)$$

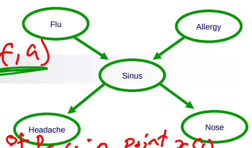
$$= \sum_i \log P(f^{(i)})\, P(a^{(i)})\, P(s^{(i)} \mid f^{(i)}, a^{(i)})\, P(h^{(i)} \mid s^{(i)})\, P(n^{(i)} \mid s^{(i)})$$

$$= \underbrace{\left[\sum_i \log P(f^{(i)})\right]}_{P(F)} + \underbrace{\left[\sum_i \log P(a^{(i)})\right]}_{P(A)} + \underbrace{\left[\sum_i \log P(s^{(i)} \mid f^{(i)}, a^{(i)})\right]}_{P(S \mid F, A)} + \underbrace{\left[\sum_i \log P(h^{(i)} \mid s^{(i)})\right]}_{P(H \mid S)} + \underbrace{\left[\sum_i \log P(n^{(i)} \mid s^{(i)})\right]}_{P(N \mid S)}$$

one per CPT $P(X_i \mid Pa_{X_i})$

only this term changes to something about $P(A \mid F)$

Information-theoretic interpretation of maximum likelihood 2



$p(s \mid f, a)$

- Given structure, log likelihood of data:

data points, nodes, point j, value of Pa_{X_i} in point $x^{(j)}$

$$\log P(\mathcal{D} \mid \theta_{\mathcal{G}}, \mathcal{G}) = \sum_{j=1}^{m} \sum_{i=1}^{n} \log P\left(X_i = x_i^{(j)} \mid \mathbf{Pa}_{X_i} = \mathbf{x}^{(j)}\left[\mathbf{Pa}_{X_i}\right]\right)$$

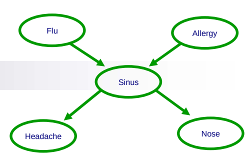
$$= \sum_{i=1}^{n} \sum_{j=1}^{m} \log P\left(X_i = x_i^{(j)} \mid Pa_{X_i} = x^{(j)}[Pa_{X_i}]\right)$$

for points j where $F=t, A=f, S=t$
$(Count(F=t, A=f, S=t))$
$\log P(S=t \mid F=t, A=f)$

$$= m \sum_{i=1}^{n} \sum_{x_i} \sum_{u} \underbrace{\frac{Count(X_i = x_i, Pa_{X_i} = u)}{m}}_{\hat{P}(X_i = x_i,\ Pa_{X_i} = u)} \log P(X_i = x_i \mid Pa_{X_i} = u)$$

MLE estimate

Information-theoretic interpretation of maximum likelihood 3



- Given structure, log likelihood of data:

hat because MLE

$$\log \hat{P}(\mathcal{D} \mid \theta, \mathcal{G}) = m \sum_i \sum_{x_i, \mathbf{Pa}_{x_i,\mathcal{G}} = u} \hat{P}(x_i, \mathbf{Pa}_{x_i,\mathcal{G}} = u) \log \hat{P}(x_i \mid \mathbf{Pa}_{x_i,\mathcal{G}} = u)$$

$$= -m \sum_i \hat{H}(X_i \mid Pa_{X_i})$$

MLE tries to pick parents s.t. $H(X_i \mid Pa_{X_i})$ small; little uncertainty about $X_i \mid Pa_{X_i}$

$$= m \sum_i \left(I(X_i, Pa_{X_i}) - H(X_i) \right)$$

$H(A \mid B)$
$= -\sum_{ab} P(a,b) \cdot \log P(a \mid b)$

$I(A,B)$
$= \sum_{ab} P(a,b) \log \frac{P(a,b)}{P(a) P(b)}$
$= H(A) - H(A \mid B)$

Decomposable score

- Log data likelihood

$$\log \hat{P}(\mathcal{D} \mid \theta, \mathcal{G}) = m \sum_i \hat{I}(x_i, \mathbf{Pa}_{x_i,\mathcal{G}}) - M \sum_i \hat{H}(X_i)$$

- Decomposable score:
  - Decomposes over families in BN (node and its parents)
  - Will lead to significant computational efficiency!!!
  - Score(*G : D*) = $\sum_i$ FamScore(X_i|$\mathbf{Pa}_{X_i}$: *D*)

How many trees are there?

Nonetheless – Efficient optimal algorithm finds best tree

Scoring a tree 1: equivalent trees

$$\log \widehat{P}(\mathcal{D} \mid \theta, \mathcal{G}) = M \sum_i \widehat{I}(x_i, \mathbf{Pa}_{x_i,\mathcal{G}}) - M \sum_i \widehat{H}(X_i)$$

Scoring a tree 2: similar trees

$$\log \widehat{P}(\mathcal{D} \mid \theta, \mathcal{G}) = M \sum_i \widehat{I}(x_i, \mathbf{Pa}_{x_i,\mathcal{G}}) - M \sum_i \widehat{H}(X_i)$$

Chow-Liu tree learning algorithm 1

- For each pair of variables X_i, X_j
  - Compute empirical distribution:

$$\widehat{P}(x_i, x_j) = \frac{\text{Count}(x_i, x_j)}{m}$$

  - Compute mutual information:

$$\widehat{I}(X_i, X_j) = \sum_{x_i, x_j} \widehat{P}(x_i, x_j) \log \frac{\widehat{P}(x_i, x_j)}{\widehat{P}(x_i)\widehat{P}(x_j)}$$

  - Nodes $X_1, \ldots, X_n$
  - Edge (i,j) gets weight

$$\widehat{I}(X_i, X_j)$$

Chow-Liu tree learning algorithm 2

$$\log \widehat{P}(\mathcal{D} \mid \theta, \mathcal{G}) = M \sum_i \widehat{I}(x_i, \mathbf{Pa}_{x_i, \mathcal{G}}) - M \sum_i \widehat{H}(X_i)$$

- Optimal tree BN
  - Compute maximum weight spanning tree
  - Directions in BN: pick any node as root, breadth-first-search defines directions

Can we extend Chow-Liu 1

- Tree augmented naïve Bayes (TAN) [Friedman et al. '97]
  - Naïve Bayes model overcounts, because correlation between features not considered
  - Same as Chow-Liu, but score edges with:

$$\hat{I}(X_i, X_j \mid C) = \sum_{c, x_i, x_j} \hat{P}(c, x_i, x_j) \log \frac{\hat{P}(x_i, x_j \mid c)}{\hat{P}(x_i \mid c)\hat{P}(x_j \mid c)}$$

Can we extend Chow-Liu 2

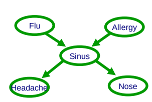
- (Approximately learning) models with tree-width up to *k*
  - [Chechetka & Guestrin '07]
  - But, O(n^{2k+6})…

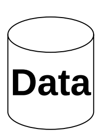
What you need to know about learning BN structures so far

- Decomposable scores
  - Maximum likelihood
  - Information theoretic interpretation
- Best tree (Chow-Liu)
- Best TAN
- Nearly best k-treewidth (in $O(N^{2k+6})$)

Scoring general graphical models – Model selection problem

What's the best structure?





$\langle x_1^{(1)},\ldots,x_n^{(1)}\rangle$

…

$\langle x_1^{(m)},\ldots,x_n^{(m)}\rangle$

The more edges, the fewer independence assumptions, the higher the likelihood of the data, but will overfit…

Maximum likelihood overfits!

$$\log \hat{P}(\mathcal{D} \mid \theta, \mathcal{G}) = M \sum_i \hat{I}(x_i, \mathbf{Pa}_{x_i,\mathcal{G}}) - M \sum_i \hat{H}(X_i)$$

- Information never hurts:
- Adding a parent always increases score!!!

Bayesian score avoids overfitting

- Given a structure, distribution over parameters

$$\log P(D \mid \mathcal{G}) = \log \int_{\theta_{\mathcal{G}}} P(D \mid \mathcal{G}, \theta_{\mathcal{G}}) P(\theta_{\mathcal{G}} | \mathcal{G}) d\theta_{\mathcal{G}}$$

- Difficult integral: use Bayes information criterion (BIC) approximation (equivalent as M! 1)

$$\log P(D \mid \mathcal{G}) \approx \log P(D \mid \mathcal{G}, \theta_{\mathcal{G}}) - \frac{\text{NumberParams}(\mathcal{G})}{2} \log M + \mathcal{O}(1)$$

- Note: regularize with MDL score
- Best BN under BIC still NP-hard

Structure learning for general graphs

- In a tree, a node only has one parent
- **Theorem**:
  - The problem of learning a BN structure with at most *d* parents is NP-hard for any (fixed) $d \geq 2$
- Most structure learning approaches use heuristics
  - Exploit score decomposition
  - (Quickly) Describe two heuristics that exploit decomposition in different ways

Learn BN structure using local search

Starting from Chow-Liu tree

Local search, possible moves:
- Add edge
- Delete edge
- Invert edge

Score using BIC

What you need to know about learning BNs

- Learning BNs
  - Maximum likelihood or MAP learns parameters
  - Decomposable score
  - Best tree (Chow-Liu)
  - Best TAN
  - Other BNs, usually local search with BIC score