Machine Learning

Learning in graphical models

Vinay Jethava

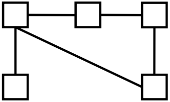
Chalmers University of Technology

TDA231 - Machine Learning
February 28, 2012

Recap

Factorization and graphs

- Factorization (of a function)
  $$f(x_1, x_2, x_3, x_4) = f_A(x_1) f_B(x_1, x_2) f_C(x_1, x_3, x_4)$$
- Graph
  - vertices (or nodes)
  - edges (connecting vertices)



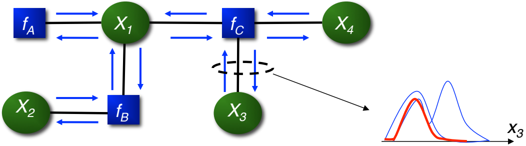
- Factor graph
  - represents factorization by a graph
  - normal style (not covered here) and conventional style

Recap

CHALMERS Chalmers University of Technology

Very high-level

- Factor graphs represent factorizations of functions
  $f(x_1, x_2, x_3, x_4) = f_A(x_1) f_B(x_1, x_2) f_C(x_1, x_3, x_4)$



- Sum-product algorithm (SPA) computes the marginals

$$g_{X_i}(x_i) = \int f(\mathbf{x}) d\mathbf{x}_{\bar{i}} \quad \text{with} \quad \mathbf{x} = [x_1 \; x_2 \; \ldots \; x_N]$$

- SPA is a message passing algorithm

This lecture

Representation

Parameter estimation

Structure Learning

Outline

Bayesian Network
Directed graphical model

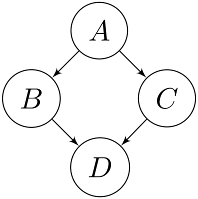


Figure: $P(A,B,C,D) = P(A)P(B|A)P(C|A)P(D|B,C)$

A node is independent of its non-descendants conditioned on its parents.

$B \perp\!\!\!\perp C$	
$B \perp\!\!\!\perp C \mid A$	
$B \perp\!\!\!\perp C \mid A, D$	

Bayesian Network
Directed graphical model

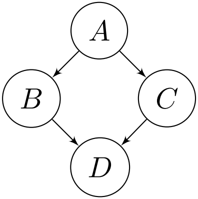


Figure: $P(A, B, C, D) = P(A)P(B|A)P(C|A)P(D|B, C)$

A node is independent of its non-descendants conditioned on its parents.

$B \perp\!\!\!\perp C$	False
$B \perp\!\!\!\perp C \vert A$	True
$B \perp\!\!\!\perp C \vert A, D$	False

Markov Network

Undirected graphical model

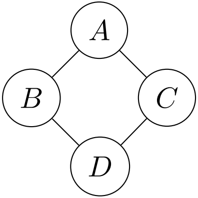


Figure: $P(A,B,C,D) = \frac{1}{Z}\phi(A,B)\phi(A,C)\phi(B,D)\phi(C,D)$

A node is independent of other nodes conditioned on its neighbours.

$B \perp\!\!\!\perp C$ $B \perp\!\!\!\perp C \mid A$ $B \perp\!\!\!\perp C \mid A, D$	

Markov Network
Undirected graphical model

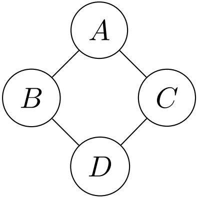


Figure: $P(A,B,C,D) = \frac{1}{Z}\phi(A,B)\phi(A,C)\phi(B,D)\phi(C,D)$

A node is independent of other nodes conditioned on its neighbours.

$B \perp\!\!\!\perp C$	False
$B \perp\!\!\!\perp C \mid A$	False
$B \perp\!\!\!\perp C \mid A, D$	True

Outline

Representation

Parameter estimation

Structure Learning

Parameter estimation: Overview

- Given data observations $x^{(1)}, \ldots, x^{(n)}$
- Given Bayesian network $P(x|\theta) = \prod_j P(x_j|Par(x_j), \theta)$

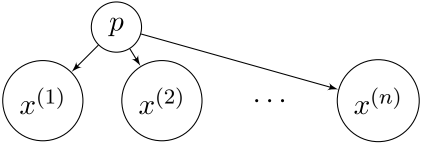
Goal

Find θ^* that best models observed data.

Maximum Likelihood Estimation (MLE)

- Likelihood $L(\theta) = \sum_{i=1}^{n} \log P(x^{(i)}|\theta)$
- $\theta^* = \arg\max L(\theta)$

Example: Loaded coin



- Coin (model parameter p)
  - *Heads* $P(x = 1|p) = p$,
  - *Tail* $P(x = 0|p) = (1 - p)$
- Observation (coin tosses): $x^{(1)} = 1,\ x^{(2)} = 0,\ \dots,\ x^{(n)} = 1$
- Model

$$P(x^{(1)}, \dots, x^{(n)}|p) = \prod_i P(x^{(i)}|p) = p^{n_1}(1-p)^{n-n_1}$$

where n_1 is number of heads.

Loaded coin

- Likelihood

$$L(p) = \sum_{i=1}^{n} \log P(x^{(n)}|p) = n_1 \log p + (n - n_1) \log(1 - p)$$

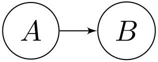
- Maximum Likelihood Estimate p^*

$$\begin{aligned} p^* &= \arg\max_p L(p) \\ &= \arg\max_p n_1 \log p + (n - n_1) \log(1 - p) \\ &= \frac{n_1}{n} \end{aligned}$$

Decomposability

$$
\begin{aligned}
\theta^* &= \arg\max_{\theta} L(\theta) \\
&= \arg\max_{\theta} \sum_{i=1}^{n} \sum_{j} \log P(X_j = x_j^{(i)} | Par(X_j) = Par(x_j), \theta_j) \\
&= \sum_{j} \arg\max_{\theta_j} L(\theta_j)
\end{aligned}
$$

Example (On board)



- $P(A, B) = P_a(X_a|\theta_a) P_{b|a}(X_b|X_a, \theta_{b|a})$
- $\theta_a^* = \arg\max_{\theta_a} \sum_{i=1}^{n} \log P(X_a^{(i)} = x_a^{(i)} | \theta_a)$
- $\theta_{b|a}^* = \arg\max_{\theta_{b|a}} \sum_{i=1}^{n} \log P_{b|a}(X_b^{(i)} = x_b^{(i)} | X_a^{(i)} = x_a^{(i)}, \theta_a)$

Learning from incomplete data
Expectation Maximization (EM)

- Y latent variable

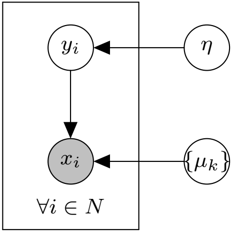
$$\begin{aligned} L(\theta) &= \log \sum_y P(x,y|\theta) = \log \sum_y q(y) \frac{P(x,y|\theta)}{q(y)} \\ &\geq \sum_y q(y) \log \frac{P(x,y|\theta)}{q(y)} \\ &= \sum_y q(y) \log p(x,y|\theta) - \sum_y q(y) \log q(y) = \mathcal{F}(q,\theta) \end{aligned}$$

- E-step: $q^{(t+1)} \leftarrow \arg\max_q \mathcal{F}(q, \theta^{(t)})$
- M-step: $\theta^{(t+1)} \leftarrow \arg\max_\theta \mathcal{F}(q^{(t+1)}, \theta)$

Example

(spherical) Gaussian mixture model

- Y_i Latent variable $P(Y_i = k) = \eta_k$
- $P(x_i | y_i = k, \mu_k) \sim \mathcal{N}(\mu_k, I)$



- If one knows Y_i:

$$\hat{\mu}_k = \langle x_i \rangle_{i:y_i=k}$$

- If one knows $\{\mu_k\}$:

$$\begin{aligned} \hat{y}_i &= \arg\max_k P(y_i = k | x_i, \mu_k) \\ &= \arg\max_k P(y_i = k) \mathcal{N}(x_i | \mu_k, I) \end{aligned}$$

Gaussian Mixture Model Example (contd.)

EM

- Iteration t, Estimates: $\hat{\mu}_k^{(t)}$
- E-step: Find $P(\hat{y}_i = k | x_i, \hat{\mu}_k)$ based on current $\hat{\mu}_k^{(t)}$
- M-step: Compute $\hat{\mu}_k^{(t+1)}$ based on $\hat{y}_i$

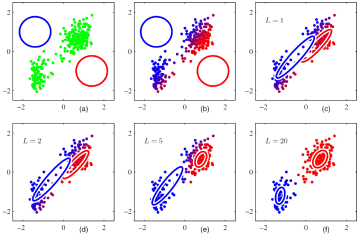


Figure: Bishop book

Parameter estimation in undirected graphical models

$$
\begin{aligned}
P(X|\theta) &= \frac{1}{Z(\theta)} \exp(\sum_c \theta_c \psi_c(X_c)) \\
Z(\theta) &= \sum_X \exp(\sum_c \theta_c \psi_c(X_c))
\end{aligned}
$$

- Given $X^{(1)}, \dots, X^{(n)}$

$$
L(\theta) = \sum_i \log p(X^{(i)}|\theta) = \sum_i \sum_c \theta_c \psi_c(X_c) - N \log Z(\theta)!!!
$$

- Finding $\theta^* = \arg\max_\theta L(\theta)$ using coordinate descent.
- Related: Iterative proportional fitting (IPF)

Gradient-based method

$$\begin{aligned}
\theta_c^{(t+1)} &= \theta_c^{(t)} + \epsilon g(\theta_c^{(t)}) \\
g(\theta_c^{(t)}) = \frac{1}{N}\frac{\partial}{\partial \theta_c} L(\theta) &= \frac{1}{N}\sum_{i=1}^{n} \psi_c(X_c) - \frac{\partial}{\partial \theta_c} \log Z(\theta) \\
&= \frac{1}{N}\sum_{i=1}^{n} \psi_c(X_c) - \sum_{X} \frac{1}{Z(\theta)} \psi_c(X_c) \\
&= \frac{1}{N}\sum_{i=1}^{n} \psi_c(X_c) - \mathbb{E}_{X_c|\theta}(\psi_c(X_c))
\end{aligned}$$

Example: Boltzmann Machine

Example: Boltzmann machine

- $P(X|W) = \frac{1}{Z(W)} \exp(-w_{12}x_1x_2 - w_{23}x_2x_3)$
- Data: $(x_1^{(i)}, x_2^{(i)}, x_3^{(i)})$ for $1 \leq i \leq N$

$$\frac{1}{N}\frac{\partial}{\partial w_{12}}L(w) \quad = \quad \frac{1}{N}\sum_{i=1}^{n} x_1^{(i)}x_2^{(i)} - \frac{\sum_{x_1,x_2} x_1x_2e^{-w_{12}x_1x_2}}{Z_{12}(w_{12})}$$

where $Z_{12}(w_{12}) = \sum_{x_1,x_2} e^{-w_{12}x_1x_2}$

Parameter Estimation: Summary

All variables observed	Maximum Likelihood (MLE)
Hidden variables	Expectation Maximization (EM)
Markov Network (undirected)	Gradient-based (IPF)

Table: Parameter estimation

Not discussed

- Conjugate priors/MAP Estimate
- MCMC for parameter estimation

Outline

Representation

Parameter estimation

Structure Learning

Overview: Structure Learning

Skeleton construction

- Conditional-Independence based
- Mutual-Information based
- Sparsity assumption (ℓ_1 prior)

Edge orientation (Bayesian networks)

Based on conditional independence.

Conditional Independence

- $X \perp\!\!\!\perp Y \Rightarrow (X, Y) \notin E$
- $X \perp\!\!\!\perp Y | Z \Rightarrow (X, Y) \notin E$

- Incrementally removes edges whenever conditional independence if observed.
- Method is not scalable.

Example

On board

Mutual-Information based

$I(X, Y)$ is the "mutual information" (positive number) between random variables X and Y

$$I(X, Y) = \sum_{x,y} P(x, y) * \log \frac{P(x, y)}{P_x(x) P_y(y)}$$

- $I(X, Y) \geq 0$, $I(X, Y) = 0$ if $X \perp\!\!\!\perp Y$

Example (On board)

- $X \sim Ber(0.5)$ i.e. $P(X = 0) = P(X = 1) = 0.5$
- $Y_1 \sim Ber(0.5)$ (independent of X); $Y_2 = X$ (dependent)
- $I(X, Y_1) = 0$
- $I(X, Y_2) = 1$

Optimal tree selection
Chow-Liu algorithm (1968)

For tree $T = (V, E)$, the Kullback-Liebler divergence measure is

$$D(\tilde{P}||P_T) = -\sum_{(i,j)\in E} I_{ij} + const.$$

and the optimal tree has minimum $D(\tilde{P}||P_T)$ i.e.

$$T^* = \arg\min_T D(\tilde{P}||P_T) = \arg\max_T \sum_{(i,j)\in E} I_{ij}$$

where I_{ij} is the "mutual information" (positive number) between dimension i and j

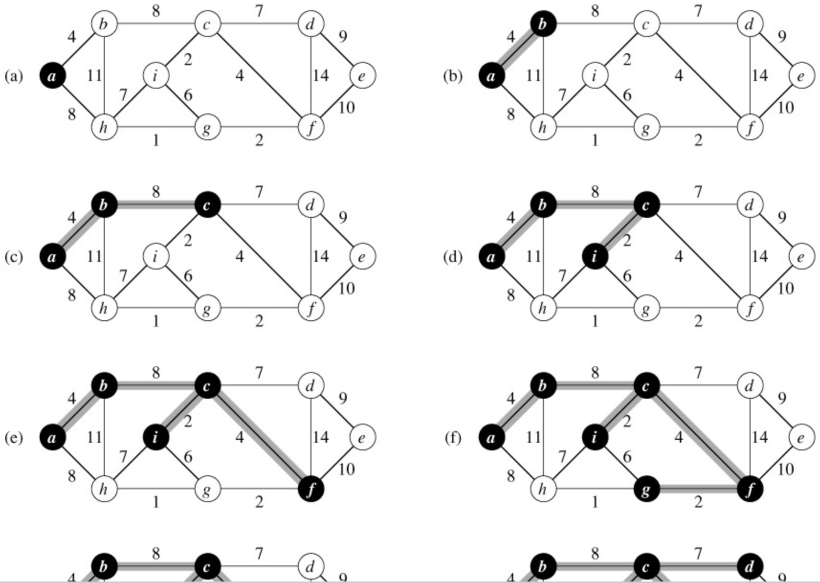
$$I_{ij} = \sum_{x_i,x_j} \tilde{P}_{ij}(x_i, x_j) * \log \frac{\tilde{P}_{ij}(x_i, x_j)}{\tilde{P}_i(x_i)\tilde{P}_j(x_j)}$$

Chow-Liu algorithm

Input: $\mathcal{D}$ {Matrix $(N \times K)$ containing facet labels for N queries}
  for i = 1 **to** K **do**
    for j= (i+1) **to** K **do**
      Find I_{ij} {Mutual information based on $\mathcal{D}$, $O(NK^2)$}
    end for
  end for
  Find MST over $\{I_{ij}\}$ {Prim/Kruskal's algorithm $O(K \log K)$}
Output: P_{T^*} {Optimal tree-structured distribution $O(NK^2)$}

Recap: Prim's algorithm

Greedy approach



Comparing P_T with independent dimensions (P_I)

$\mathbf{x_1 x_2 x_3}$	$\tilde{P}_{123}$	$P_I \sim \tilde{P}_1 \tilde{P}_2 \tilde{P}_3$	$P_T \sim \tilde{P}_{1\|2} \tilde{P}_{3\|2} \tilde{P}_2$
000	0.2	0.21	0.142857
100	0	0.21	0.0571429
010	0.1	0.09	0.1
110	0.2	0.09	0.2
001	0.3	0.14	0.357143
101	0.2	0.14	0.142857
011	0	0.06	0
111	0	0.06	0

$$D(\tilde{P}||P_I) = 0.4605$$

Kullback-Liebler divergence

$$D(\tilde{P}||\hat{P}) = \sum_{\mathbf{f}} \tilde{P}(\mathbf{f}) * \log \frac{\tilde{P}(\mathbf{f})}{\hat{P}(\mathbf{f})}$$

$$D(\tilde{P}||P_T) = 0.0823$$

Sparsity-based

- Finding structure is hard for general graphs.
- Assumption: Total number of edges is few (sparse model)

$$\begin{aligned}\theta^* &= \arg\max_\theta L(\theta) + ||\theta||_1 \\ &= \arg\max_\theta L(\theta) + \sum_c \theta_c\end{aligned}$$

- Leads to few large $\theta_c > 0$ and other $\theta_c = 0$.

Summary: Structure learning

Incremental	Conditional Independence	Does not scale
Chow-Liu	Mutual Information	Tree-structured dn.
Sparsity-based	Laplace prior (ℓ_1) on θ	Sparse graphs only

Table: Structure learning

- Learning structure for general graphical models is a hard (intractable) problem.
- Domain expertise helps in model selection.

Conclusion

All variables observed	Maximum Likelihood (MLE)
Hidden variables	Expectation Maximization (EM)
Markov Network (undirected)	Gradient-based (IPF)

Table: Parameter estimation

Incremental	Conditional Independence	Does not scale
Chow-Liu	Mutual Information	Tree-structured dn.
Sparsity-based	Laplace prior (ℓ_1) on θ	Sparse graphs only

Table: Structure learning