CS839: Probabilistic Graphical Models

Lecture 7: Learning Fully Observed BNs

Theo Rekatsinas
Exponential family: a basic building block

• For a numeric random variable X

\[ p(x|\eta) = h(x) \exp(\eta^T T(x) - A(\eta)) = \frac{1}{Z(\eta)} h(x) \exp(\eta^T T(x)) \]

is an exponential family distribution with natural (canonical) parameter \( \eta \)

• Function \( T(x) \) is a sufficient statistic.
• Function \( A(\eta) = \log Z(\eta) \) is the log normalizer
• Examples: Bernoulli, multinomial, Gaussian, Poisson, Gamma, Categorical
Why exponential family?

• Moment generating property

\[
\frac{dA}{d\eta} = \frac{d}{d\eta} \log Z(\eta) = \frac{1}{Z(\eta)} \frac{d}{d\eta} Z(\eta) \\
= \frac{1}{Z(\eta)} \frac{d}{d\eta} \int h(x) \exp\{\eta^T T(x)\} dx \\
= \int T(x) \frac{h(x) \exp\{\eta^T T(x)\}}{Z(\eta)} dx
\]

\[
\frac{d^2 A}{d\eta^2} = \int T^2(x) \frac{h(x) \exp\{\eta^T T(x)\}}{Z(\eta)} dx - \int T(x) \frac{h(x) \exp\{\eta^T T(x)\}}{Z(\eta)} dx \frac{1}{Z(\eta)} \frac{d}{d\eta} Z(\eta) \\
= E[T^2(x)] - E[T(x)]^2 \\
= Var[T(x)]
\]

We can easily compute moments of any exponential family distribution by taking the derivatives of the log normalizer \(A(\eta)\)
MLE for Exponential Family

• For iid data the log-likelihood is

\[
\ell(\eta; D) = \log \prod_{n} h(x_n) \exp\{\eta^T T(x_n) - A(\eta)\} \\
= \sum_{n} \log h(x_n) + \left(\eta^T \sum_{n} T(x_n) \right) - NA(\eta)
\]

\[
\frac{\partial \ell}{\partial \eta} = \sum_{n} T(x_n) - N \frac{\partial A(\eta)}{\partial \eta} = 0
\]

\[
\frac{\partial A(\eta)}{\partial \eta} = \frac{1}{N} \sum_{n} T(x_n)
\]

\[
\Rightarrow \quad \hat{\mu}_{\text{MLE}} = \frac{1}{N} \sum_{n} T(x_n)
\]

\[
\hat{\eta}_{\text{MLE}} = \psi(\hat{\mu}_{\text{MLE}})
\]
Generalized Linear Models

• The graphical model:
  • Linear regression
  • Discriminative linear classification

\[ E_p(T) = \mu = f(\theta^T X) \]

• Generalized Linear Model
  • The observed input $x$ is assumed to enter into the model via a linear combination of its elements $\xi = \theta^T x$.
  • The conditional mean $\mu$ is represented as a function $f(\xi)$ of $\xi$, where $f$ is known as the response function.
  • The observed output $y$ is assumed to be characterized by an exponential family distribution with conditional mean $\mu$. 
Learning in Graphical Models

- Goal: Given a set of independent samples (assignments to random variables), find the best Bayesian Network (both DAG and CPDs)

\[(B, E, A, C, R) = (T, F, F, T, F)\]
\[(B, E, A, C, R) = (T, F, T, T, F)\]
\[...\]
\[(B, E, A, C, R) = (F, T, T, T, F)\]
Learning in Graphical Models

- Goal: Given a set of independent samples (assignments to random variables), find the best Bayesian Network (both DAG and CPDs)

\[(B,E,A,C,R) = (T,F,F,T,F)\]
\[(B,E,A,C,R) = (T,F,T,T,F)\]
...
\[(B,E,A,C,R) = (F,T,T,T,F)\]
Learning in Graphical Models

- Goal: Given a set of independent samples (assignments to random variables), find the best Bayesian Network (both DAG and CPDs)

\[(B,E,A,C,R) = (T,F,F,T,F)\]
\[(B,E,A,C,R) = (T,F,T,T,F)\]
...
\[(B,E,A,C,R) = (F,T,T,T,F)\]

Structure learning

Parameter learning

|   | B | E | P(A | E,B) |
|---|---|---|---------|
| e | b | 0.9 | 0.1    |
| e | b | 0.2 | 0.8    |
| e | b | 0.9 | 0.1    |
| e | b | 0.01| 0.99   |
Learning in Graphical Models

• Goal: Given a set of independent samples (assignments to random variables), find the best Bayesian Network (both DAG and CPDs)

(B, E, A, C, R) = (T, F, F, T, F)
(B, E, A, C, R) = (T, F, T, T, F)
...
(B, E, A, C, R) = (F, T, T, T, F)

Structure learning
Later in the class

Parameter learning
Parameter Estimation for Fully Observed GMs

- The data: \( D = (x_1, x_2, x_3, \ldots, x_N) \)
- Assume the graph \( G \) is known and fixed
  - Expert design or structure learning
- Goal: estimate from a dataset of \( N \) independent, identically distributed (iid) training examples \( D \)
- Each training example corresponds to a vector of \( M \) values one per node random variable
  - Model should be completely observable: no missing values, no hidden variables

\[
\ell(\theta; D) = \log p(D | \theta) = \log \prod_n \left( \prod_i p(x_{n,i} | x_{n,\pi_i}, \theta_i) \right) = \sum_i \left( \sum_n \log p(x_{n,i} | x_{n,\pi_i}, \theta_i) \right)
\]
Density estimation

• A construction of an *estimate*, based on observed data, of an unobservable underlying probability density function

• Can be viewed as single-node graphical models

• Instances of exponential family distribution

• Building blocks of general GM

• MLE and Bayesian estimate
Discrete Distributions

- Bernoulli distribution: \( P(x) = p^x (1 - p)^{1-x} \)

- Multinomial distribution: Mult(1, \( \theta \))

\[
X = [X_1, X_2, X_3, X_4, X_5, X_6] \quad X_j = [0, 1], \quad \sum_{j \in [1, \ldots, 6]} X_j = 1
\]

\[
X_j = 1 \text{ with probability } \theta_j, \quad \sum_{j \in [1, \ldots, 6]} \theta_j = 1
\]

\[
P(X_j = 1) = \theta_j
\]
Discrete Distributions

- Multinomial distribution: Mult(n, θ)

\[ n = [n_1, n_2, \ldots, n_k] \text{ where } \sum_{j} n_j = N \]

\[ p(n) = \frac{N!}{n_1!n_2! \cdots n_k!} \theta_1^{n_1} \theta_2^{n_2} \cdots \theta_K^{n_K} \]
Example: multinomial model

- Data: We observed $N$ iid die rolls ($K$-sided): $D = \{5, 1, K, \ldots 3\}$

$$x_n = [x_{n,1}, x_{n,2}, \ldots, x_{n,K}] \text{ where } x_{n,k} = 0, 1 \sum_{k=1}^{K} x_{n,k} = 1$$

- Model: $X_{n,k} = 1$ with probability $\theta_k$ and $\sum_{k \in \{1, \ldots, K\}} \theta_k = 1$

- Likelihood of an observation: $P(x_i) = P(\{x_{n,k} = 1, \text{ where } k \text{ is the index of the n-th roll}\})$

$$= \theta_k = \theta_1^{x_{n,1}} \theta_2^{x_{n,2}} \ldots \theta_K^{x_{n,k}} = \prod_{k=1}^{K} \theta_k^{x_{n,k}}$$

- Likelihood of $D$: $P(x_1, x_2, \ldots, x_N | \theta) = \prod_{n=1}^{N} P(x_n | \theta) = \prod_{k} \theta_k^{n_k}$
MLE: constrained optimization

• Objective function: 
  \[ l(\theta; D) = \log P(D|\theta) = \log \prod_k \theta_{nk}^{n_k} = \sum_k n_k \log \theta_k \]

• We need to maximize this subject to the constraint: 
  \[ \sum_{k\in\{1,\ldots,K\}} \theta_k = 1 \]

• Lagrange multipliers: 
  \[ \bar{l}(\theta; D) = \sum_k n_k \log \theta_k + \lambda(1 - \sum_k \theta_k) \]

• Derivatives: 
  \[ \frac{\partial \bar{l}}{\partial \theta_k} = \frac{n_k}{\theta_k} - \lambda = 0 \]
  \[ n_k = \lambda \theta_k \Rightarrow \sum_k n_k = \lambda \sum_k \theta_k \Rightarrow N = \lambda \]
  \[ \hat{\theta}_{k,MLE} = \frac{1}{N} \sum_n x_{n,k} \]

• Sufficient statistics?
Bayesian estimation

- I need a prior over parameters $\theta$

- Dirichlet distribution

\[
P(\theta) = \frac{\Gamma(\sum_k \alpha_k)}{\prod_k \Gamma(\alpha_k)} \prod_k \theta_{\alpha_k-1} = C(\alpha) \prod_k \theta_{\alpha_k-1}
\]

- Posterior of $\theta$

\[
P(\theta| x_1, \ldots, x_N) = \frac{p(x_1, \ldots, x_N|\theta)p(\theta)}{p(x_1, \ldots, x_N)} \propto \prod_k \theta_{n_k} \prod_k \theta_{\alpha_k-1} = \prod_k \theta_{\alpha_k+n_k-1}
\]

- Isomorphism of the posterior with the prior (conjugate prior)

- Posterior mean estimation

\[
\theta_k = \int \theta_k p(\theta|D)d\theta = C \int \theta_k \prod_k \theta_{\alpha_k+n_k-1}d\theta = \frac{n_k + \alpha_k}{N + |\alpha|}
\]
Continuous Distributions

- Uniform
  \[ p(x) = \begin{cases} 
  1/(b-a) & \text{for } a \leq x \leq b \\
  0 & \text{elsewhere} 
  \end{cases} \]

- Gaussian
  \[ p(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-(x-\mu)^2/2\sigma^2} \]

- Multivariate Gaussian
  \[ p(X; \mu, \Sigma) = \frac{1}{\left(\sqrt{2\pi}\right)^{n/2} \left|\Sigma\right|^{1/2}} \exp \left\{ -\frac{1}{2} (X - \mu)^T \Sigma^{-1} (X - \mu) \right\} \]
MLE for a multivariate Gaussian

• You can show that the MLE for $\mu$ and $\Sigma$ is

$$\mu_{MLE} = \frac{1}{N} \sum_{n} (x_n)$$

$$\Sigma_{MLE} = \frac{1}{N} \sum_{n} (x_n - \mu_{ML})(x_n - \mu_{ML})^T$$

• What are the sufficient statistics?
MLE for a multivariate Gaussian

• You can show that the MLE for $\mu$ and $\Sigma$ is

\[
\mu_{MLE} = \frac{1}{N} \sum_{n} (x_n)
\]

\[
\Sigma_{MLE} = \frac{1}{N} \sum_{n} (x_n - \mu_{ML})(x_n - \mu_{ML})^T
\]

• What are the sufficient statistics?
  • Rewrite

\[
S = \sum_{n} (x_n - \mu_{ML})(x_n - \mu_{ML})^T = \left(\sum_{n} x_n x_n^T\right) - N\mu_{ML}\mu_{ML}^T
\]

• Sufficient statistics are:

\[
\sum_{n} (x_n) \quad \left(\sum_{n} x_n x_n^T\right)
\]

• Similar for Bayesian estimation: Normal prior
MLE for general BNs

• If we assume the parameters for each CPD are globally independent, and all nodes are fully observed, then the log-likelihood function decomposes into a sum of local terms, one per node

\[
\ell(\theta; D) = \log p(D | \theta) = \log \prod_{n} \left( \prod_{i} p(x_{n,i} | x_{n,\pi_i}, \theta_i) \right) = \sum_{n} \left( \sum_{i} \log p(x_{n,i} | x_{n,\pi_i}, \theta_i) \right)
\]

• MLE-based parameter estimation of GM reduces to local est. of each GLIM.
Decomposable likelihood of a BN

• Consider the GM:

\[ p(x \mid \theta) = p(x_1 \mid \theta_1)p(x_2 \mid x_1, \theta_2)p(x_3 \mid x_1, \theta_3)p(x_4 \mid x_2, x_3, \theta_4) \]

• This is the same as learning four separate smaller BNs each of which consists of a node an its parents.
MLE for BNs with tabular CPDs

• Each CPD is represented as a table (multinomial) with

\[ \theta_{ijk} \overset{\text{def}}{=} p(X_i = j \mid X_{\pi_i} = k) \]

• In case of multiple parents the CPD is a high-dimensional table
• The sufficient statistics are counts of variable configurations

\[ n_{ijk} = \sum_n x_{n,i}^j x_{n,\pi_i}^k \]

• The log-likelihood is

\[ \ell(\theta; D) = \log \prod_{i,j,k} \theta_{ijk}^{n_{ijk}} = \sum_{i,j,k} n_{ijk} \log \theta_{ijk} \]

• And using a Lagrange multiplier to enforce that conditionals sum up to 1 we have:

\[ \theta_{ijk}^{ML} = \frac{n_{ijk}}{\sum_{j'} n_{ij'k}} \]
What about parameter priors?

• In a BN we have a collection of local distributions

• How can we define priors over the whole BN?

• We could write $P(x_1, x_2, \ldots, x_N; G, \theta)P(\theta | \alpha)$
  • Symbolically the same as before but $\theta$ is defined over a vector of random variables that follow different distributions.
  • We need $\theta$ to decompose to use local rules. Otherwise we cannot decompose the likelihood any more.

• We need certain rules on $\theta$
  • Complete Model Equivalence
  • Global Parameter Independence
  • Local Parameter Independence
  • Likelihood and Prior Modularity
Global and Local Parameter Independence

• Global Parameter Independence
  • For every DAG model
    \[ p(\theta_m \mid G) = \prod_{i=1}^{M} p(\theta_i \mid G) \]

• Local Parameter Independence
  • For every node
    \[ p(\theta_i \mid G) = \prod_{j=1}^{q_i} p(\theta_{x_i^k \mid x_i^{j \pi_i}} \mid G) \]

\[ P(\theta_{\text{Call}} \mid \text{Alarm} = \text{YES}) \text{ independent of } P(\theta_{\text{Call}} \mid \text{Alarm} = \text{NO}) \]
Global and Local Parameter Independence

Global Parameter Independence

Local Parameter Independence

sample 1

sample 2
Which PDFs satisfy these assumptions?

- **Discrete DAG Models** \( x_i \mid \pi_{X_i}^j \sim \text{Multi}(\theta) \)
  
  **Dirichlet prior:**
  
  \[
  P(\theta) = \frac{\Gamma\left(\sum_k \alpha_k\right)}{\prod_k \Gamma(\alpha_k)} \prod_k \theta_k^{\alpha_k - 1} = C(\alpha) \prod_k \theta_k^{\alpha_k - 1}
  \]

- **Gaussian DAG Models** \( x_i \mid \pi_{X_i}^j \sim \text{Normal}(\mu, \Sigma) \)
  
  **Normal prior:**
  
  \[
  p(\mu \mid \nu, \Psi) = \frac{1}{(2\pi)^{n/2} | \Psi |^{1/2}} \exp\left\{ -\frac{1}{2} (\mu - \nu)' \Psi^{-1} (\mu - \nu) \right\}
  \]
Parameter sharing

- Transition probabilities $P(X_t|X_{t-1})$ can be different: What is the parameterization cost?
Parameter sharing

• Transition probabilities $P(X_t|X_{t-1})$ can be different: What is the parameterization cost?
  • I need a different local conditional distribution.
  • How can I learn the transition probabilities? I need iid data. So my training examples should correspond multiple rolls of the dice. I need to do alignment etc.
  • What do we do?
Parameter sharing

• We will make the assumption that for every transition we follow the same conditional

• Consider a time-invariant (stationary) 1\textsuperscript{st}-order Markov model
  • Initial state probability vector
  • State transition probability matrix \( \pi_k \stackrel{\text{def}}{=} p(X^k_1 = 1) \quad A_{ij} \stackrel{\text{def}}{=} p(X^j_t = 1 | X^i_{t-1} = 1) \)

• Now: \( p(X_{1:T} | \theta) = p(x_1 | \pi) \prod_{t=2}^{T} \prod_{t=2}^{T} p(X_t | X_{t-1}) \) optimize separately
  • \( \pi \) (multinomial)
  • What about A?
Learning a Markov chain transition matrix

- A is a stochastic matrix with \( \sum_{j} A_{i,j} = 1 \)
- Each row of A is a multinominal distribution
- MLE of \( A_{ij} \) is the fraction of transitions from i to j

\[
A_{ij}^{ML} = \frac{\#(i \rightarrow j)}{\#(i \rightarrow \bullet)} = \frac{\sum_{n} \sum_{t=2}^{T} x_{n,t-1}^i x_{n,t}^j}{\sum_{n} \sum_{t=2}^{T} x_{n,t-1}^i}
\]

- Sparse data problem:
  - If i to j did not occur in the data then \( A_{ij} \) is 0. Any future sequence with pair i to j will have zero probability
  - Backoff smoothing \( \tilde{A}_{i \rightarrow \bullet} = \lambda \eta_t + (1 - \lambda) A_{i \rightarrow \bullet}^{ML} \)
Example: HMM supervised ML estimation

• Given \( x = x_1, x_2, \ldots, x_N \) for which the true state path \( y_1, y_2, \ldots, y_N \) is known:

  • Define

  \[
  A_{ij} = \# \text{ times state transition } i \to j \text{ occurs in } y \\
  B_{ik} = \# \text{ times state } i \text{ in } y \text{ emits } k \text{ in } x
  \]

• The maximum likelihood parameters \( \theta \) are:

\[
a_{ij}^{ML} = \frac{\#(i \to j)}{\#(i \to \bullet)} = \frac{\sum_n \sum_{t=2}^{T} y_{n,t-1}^i y_{n,t}^j}{\sum_n \sum_{t=2}^{T} y_{n,t-1}^i} = \frac{A_{ij}}{\sum_{j'} A_{ij'}}
\]

\[
b_{ik}^{ML} = \frac{\#(i \to k)}{\#(i \to \bullet)} = \frac{\sum_n \sum_{t=1}^{T} y_{n,t}^i x_{n,t}^k}{\sum_n \sum_{t=1}^{T} y_{n,t}^i} = \frac{B_{ik}}{\sum_{k'} B_{ik'}}
\]

• If \( x \) is continuous we can apply learning rules for a Gaussian
Supervised ML estimation

- **Intuition:** when we know the underlying states, the best estimate of $\theta$ is the average frequency of transitions and emissions that occur in the training data.

- **Drawback:** Given little data, we may overfit (remember zero probabilities).

- **Example:**
  - Given 10 rolls we have
    
    \[
    x = 2, 1, 5, 6, 1, 2, 3, 6, 2, 3 \\
    \]
  
    \[
    a_{FF} = 1; \quad a_{FL} = 0 \\
    b_{F1} = b_{F3} = .2; \\
    b_{F2} = .3; \quad b_{F4} = 0; \quad b_{F5} = b_{F6} = .1
    \]
Pseudocounts

• Solution for small training sets:
  • Add pseudocounts

\[
A_{ij} = \text{# times state transition } i \to j \text{ occurs in } y + R_{ij} \\
B_{ik} = \text{# times state } i \text{ in } y \text{ emits } k \text{ in } x + S_{ik}
\]

• The pseudocounts represent our prior belief

• Large total pseudocounts => strong prior belief
• Small total pseudocounts => smoothing to avoid 0 probabilities
• Equivalent to Bayesian estimation under a uniform prior with parameter strength equal to the pseudocounts.
Summary

• For fully observed BN, the log-likelihood function decomposes into a sum of local terms, one per node; thus learning is also factored

• Learning single-node GM – density estimation: exp. Family
  • Typical discrete distribution
  • Typical continuous distribution
  • Conjugate priors

• Learning BN with more nodes:
  • Local operations