Parameter and Structure Learning

Dhruv Batra,
10-708 Recitation
10/02/2008

Overview

- Parameter Learning
  - Classical view, estimation task
  - Estimators, properties of estimators
  - MLE, why MLE?
  - MLE in BNs, decomposability

- Structure Learning
  - Structure score, decomposable scores
  - TAN, Chow-Liu
  - HW2 implementation steps

Note

- Plagiarism alert
  - Some slides taken from others
  - Credits/references at the end

Coin Toss

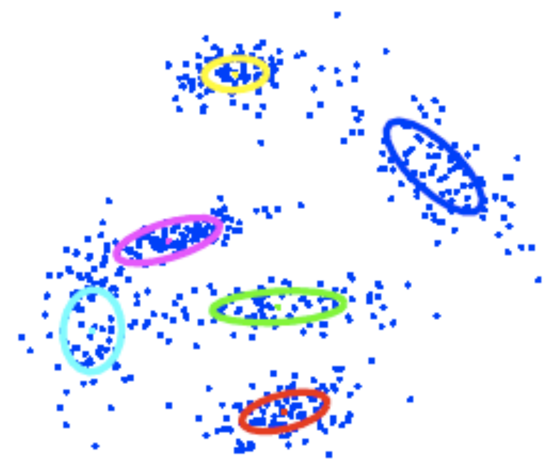
Data: $\mathcal{D} = (H\,T\,H\,H\,H\,T\,T\,\ldots)$

Parameters: $\theta \stackrel{\text{def}}{=}$ Probability of heads

$$P(H|\theta) = \theta$$
$$P(T|\theta) = 1 - \theta$$

Goal: To infer θ from the data and predict future outcomes $P(H|\mathcal{D})$.

Clustering with Gaussian Mixtures (Density Estimation)



Data: $\mathcal{D} = \{\mathbf{x}^{(n)}\}$ for $n = 1, \ldots, N$

$$\mathbf{x}^{(n)} \in \mathbb{R}^D$$

Parameters: $\boldsymbol{\theta} = \left((\mu^{(1)}, \Sigma^{(1)}) \ldots, (\mu^{(m)}, \Sigma^{(m)}), \boldsymbol{\pi}\right)$

Model:

$$\mathbf{x}^{(n)} \sim \sum_{i=1}^{m} \pi_i \, p_i(\mathbf{x}^{(n)})$$

where

$$p_i(\mathbf{x}^{(n)}) = \mathcal{N}(\mu^{(i)}, \Sigma^{(i)})$$

Goal: To infer $\boldsymbol{\theta}$ from the data and predict the density $p(\mathbf{x}|\mathcal{D}, m)$

Parameter Learning

- Classical statistics view / Point Estimation
  - Parameters unknown but not random
  - Point estimation = “find the right parameter”
  - Estimate parameters (or functions of parameters) of the model from data

- Estimators
  - Any statistic
  - Function of data alone

- Say you have a dataset $\mathcal{D} = \{\mathbf{x}^{(n)}\}$
  - Need to estimate mean
  - Is $\hat{\mu} =$ 5, an estimator?
  - What would you do?

Properties of estimator

- Since estimator gives rise an estimate that depends on sample points (x1,x2,,,xn) estimate is a function of sample points.

- Sample points are random variable therefore estimate is random variable and has probability distribution.

- We want that estimator to have several desirable properties like
  - Consistency
  - Unbiasedness
  - Minimum variance

- In general it is not possible for an estimator to have all these properties.

A General MLE strategy

Suppose $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_n)^T$ is a vector of parameters.

Task: Find MLE $\boldsymbol{\theta}$ assuming known form for p(Data| $\boldsymbol{\theta}$,stuff)

1. Write LL = log P(Data| $\boldsymbol{\theta}$,stuff)
2. Work out ∂LL/$\partial\boldsymbol{\theta}$ using high-school calculus
3. Solve the set of simultaneous equations

$$\frac{\partial LL}{\partial \theta_1} = 0$$
$$\frac{\partial LL}{\partial \theta_2} = 0$$
$$\vdots$$
$$\frac{\partial LL}{\partial \theta_n} = 0$$

4. Check that you're at a maximum

The MLE μ

$$\mu^{mle} = \arg\max_{\mu} p(x_1, x_2, \ldots x_R \mid \mu, \sigma^2)$$

$$= \arg\min_{\mu} \sum_{i=1}^{R} (x_i - \mu)^2$$

$$= \mu \text{ s.t. } \quad 0 = \frac{\partial \mathrm{LL}}{\partial \mu} = \frac{\partial}{\partial \mu} \sum_{i=1}^{R} (x_i - \mu)^2$$

$$- \sum_{i=1}^{R} 2(x_i - \mu)$$

$$\text{Thus} \quad \mu = \frac{1}{R} \sum_{i=1}^{R} x_i$$

Unbiased Estimators

- An estimator of a parameter is unbiased if the expected value of the estimate is the same as the true value of the parameters.
- *If* $x_1, x_2, \ldots x_R \sim$(i.i.d) $N(\mu,\sigma^2)$ then

$$E[\mu^{mle}] = E\left[\frac{1}{R}\sum_{i=1}^{R} x_i\right] = \mu$$

μ^{mle} is unbiased

Biased Estimators

- An estimator of a parameter is biased if the expected value of the estimate is different from the true value of the parameters.
- *If* $x_1, x_2, \ldots x_R \sim$(i.i.d) $N(\mu,\sigma^2)$ then

$$E\left[\sigma^2_{mle}\right] = E\left[\frac{1}{R}\sum_{i=1}^{R}(x_i - \mu^{mle})^2\right] = E\left[\frac{1}{R}\left(\sum_{i=1}^{R}x_i - \frac{1}{R}\sum_{j=1}^{R}x_j\right)^2\right] \neq \sigma^2$$

σ^2_{mle} is biased

So why MLE?

- MLE has some nice properties
  - MLEs are often simple and easy to compute.
  - MLEs have asymptotic optimality properties (consistency and efficiency).
  - MLEs are invariant under reparameterization.
  - and more..

Let's try

5 [10 pts] ML and MAP Estimation

Recall the probability mass function for a Poisson distribution:

$$p(x|\theta) = \frac{\theta^x e^\theta}{x!}$$

5.1 [2 pts]

Derive the maximum likelihood estimate of θ.

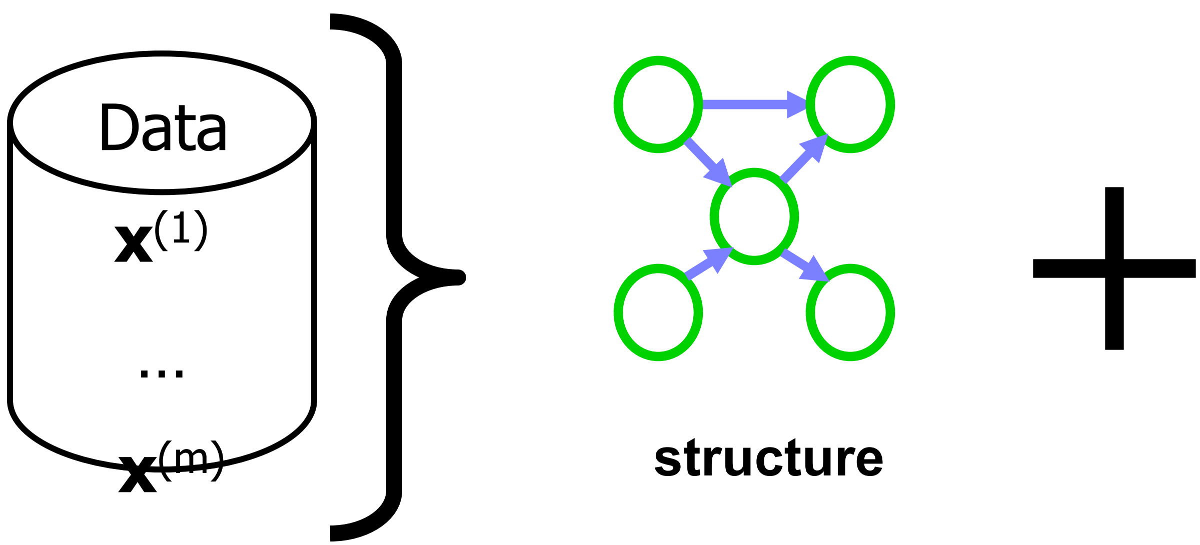
5.2 [4 pts]

Prove that the maximum likelihood estimate is invariant to any $1-1$ reparameterization of θ. That is, given an invertible function $f(\mu) = \theta$ which yields the reparametrized distribution $p(x|f(\mu)) = p(x|\theta)$, prove that the maximum likelihood estimates of μ and θ satisfy,

$$f(\hat{\mu}) = \hat{\theta}$$

Back to BNs

- MLE in BN
  - Data
  - Model DAG G
  - Parameters CPTs
  - Learn parameters from data

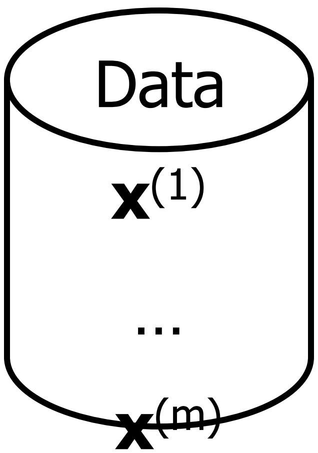


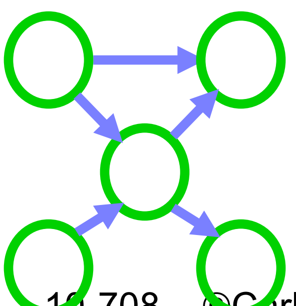
structure

CPTs – $P(X_i \mid \mathbf{Pa}_{X_i})$

parameters

Learning the CPTs





For each discrete variable X_i

$Pa_{X_i} = U$

$P(X_i \mid Pa_{X_i}) = P(X_i \mid U)$

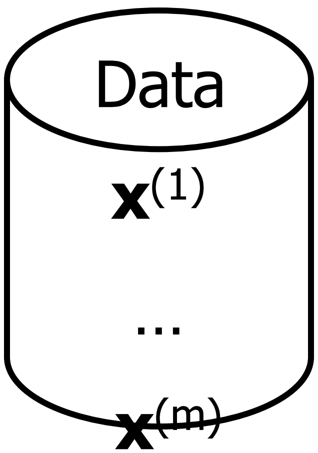
$$\hat{P}_{MLE}(X_i = x_i \mid U = u) = \frac{\text{Count}(X_i = x_i, U = u)}{\text{Count}(U = u)}$$

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

Example

- Learning MLE parameters

Learning the CPTs



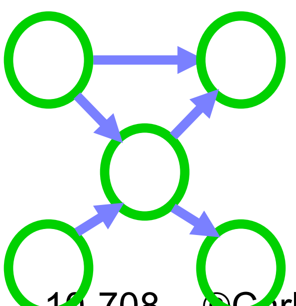
For each discrete variable X_i

$Pa_{X_i} = U$

$$P(X_i \mid Pa_{X_i}) = P(X_i \mid U)$$

$$\hat{P}_{MLE}(X_i = x_i \mid U = u) = \frac{\text{Count}(X_i = x_i, U = u)}{\text{Count}(U = u)}$$

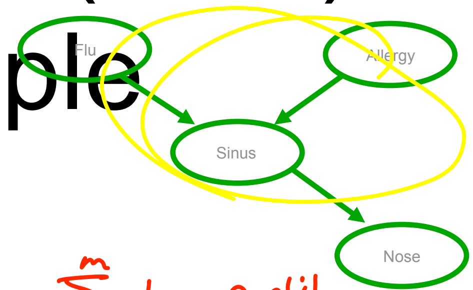
Why??



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

Maximum likelihood estimation (MLE) of BN parameters – example

$\log a \cdot b = \log a + \log b$



- Given structure, log likelihood of data:

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

for the example

$$\sum_{j=1}^{m} \log P(f^{(j)}, a^{(j)}, s^{(j)}, n^{(j)} \mid \theta_G, G) = \sum_{j=1}^{m} \log P(f^{(j)} \mid \theta_G, G) \cdot P(a^{(j)} \mid \theta_G, G)\, P(s^{(j)} \mid a^{(j)}, f^{(j)}, \theta_G, G)\, P(n^{(j)} \mid s^{(j)}, \theta_G, G)$$

$$= \sum_{j=1}^{m} \left[\log P(f^{(j)} \mid \theta_G, G) + \log P(a^{(j)} \mid \theta_G, G) + \log P(s^{(j)} \mid a^{(j)}, f^{(j)}, \theta_G, G) + \log P(n^{(j)} \mid s^{(j)}, \theta_G, G) \right]$$

$$= \underbrace{\sum_{j=1}^{m} \log P(f^{(j)} \mid \theta_F, G)}_{P(F)} + \underbrace{\sum_{j=1}^{m} \log P(a^{(j)} \mid \theta_A, G)}_{P(A)} + \underbrace{\sum_{j=1}^{m} \log P(s^{(j)} \mid a^{(j)}, f^{(j)}, \theta_{S|FA}, G)}_{P(S|FA)} + \underbrace{\sum_{j=1}^{m} \log P(n^{(j)} \mid s^{(j)}, \theta_{N|S}, G)}_{P(N|S)}$$

Broke up problem into independent subproblems : one for each CPT

Decomposability

- Likelihood Decomposition

$$
\begin{aligned}
L(\theta : \mathcal{D}) &= \prod_m P_{\mathcal{G}}(\xi[m] : \theta) \\
&= \prod_m \prod_i P(x_i[m] \mid \mathbf{pa}_i[m] : \theta) \\
&= \prod_i \left[\prod_m P(x_i[m] \mid \mathbf{pa}_i[m] : \theta) \right]
\end{aligned}
$$

$$L(\theta : \mathcal{D}) = \prod_i L_i(\theta_{X_i \mid \mathbf{Pa}_i} : \mathcal{D}),$$

What's the difference?

- Local likelihood function

$$L_i(\theta_{X_i \mid \mathbf{Pa}_i} : \mathcal{D}) = \prod_m P(x_i[m] \mid \mathbf{pa}_i[m] : \theta_{X_i \mid \mathbf{Pa}_i}).$$

Global parameter independence!

Taking derivatives of MLE of BN parameters – General case

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

$$P(X_i = x_i \mid Pa_{X_i} = u) = \theta_{X_i = x_i \mid Pa_{X_i} = u} = \theta_{x_i \mid u}$$

$$\frac{\partial}{\partial \theta_{x_i \mid u}} \log P(D \mid \theta_G, G) = \sum_{k=1}^{n} \sum_{j=1}^{m} \underbrace{\frac{\partial}{\partial \theta_{x_i \mid u}} \log P\left(X_k = x_k^{(j)} \mid Pa_{X_k} = x^{(j)}[Pa_{X_k}]\right)}_{k \neq i \quad \text{derivative} = 0}$$

$$= \sum_{j=1}^{m} \frac{\partial}{\partial \theta_{x_i \mid u}} \log P\left(X_i = x_i^{(j)} \mid Pa_{X_i} = x^{(j)}[Pa_{X_i}]\right)$$

Structure Learning

- Constraint Based
  - Check independences, learn PDAG
  - HW1

- Score Based
  - Give a score for all possible structures
  - Maximize score

Score Based

- What's a good score function?

- How about our old friend, log likelihood?

$$\begin{aligned}\max_{\mathcal{G},\boldsymbol{\theta}_{\mathcal{G}}} L(\langle \mathcal{G}, \boldsymbol{\theta}_{\mathcal{G}} \rangle : \mathcal{D}) &= \max_{\mathcal{G}}[\max_{\boldsymbol{\theta}_{\mathcal{G}}} L(\langle \mathcal{G}, \boldsymbol{\theta}_{\mathcal{G}} \rangle : \mathcal{D})] \\ &= \max_{\mathcal{G}}[L(\langle \mathcal{G}, \hat{\boldsymbol{\theta}}_{\mathcal{G}} \rangle : \mathcal{D})]\end{aligned}$$

- So here's our score function:

$$\mathrm{score}_L(\mathcal{G} : \mathcal{D}) = \ell(\langle \mathcal{G}, \hat{\boldsymbol{\theta}}_{\mathcal{G}} \rangle : \mathcal{D})$$

Score Based

- [Defn]: Decomposable scores

- Why do we care about decomposable scores?

- Log likelihood based score decomposes!

$$\text{score}_L(\mathcal{G} : \mathcal{D}) = M \sum_{i=1}^{n} \boldsymbol{I}_{\hat{P}}(X_i; \mathbf{Pa}_{X_i}^{\mathcal{G}}) - M \sum_{i=1}^{n} \boldsymbol{H}_{\hat{P}}(X_i)$$

Need regularization

Score Based

- Chow-Liu

1. Compute $I_{\hat{P}_D}(X_i; X_j)$ between each pair of variables, $i \neq j$, where

$$I_P(\mathbf{X}; \mathbf{Y}) = \sum_{\mathbf{x},\mathbf{y}} P(\mathbf{x}, \mathbf{y}) \log \frac{P(\mathbf{x}, \mathbf{y})}{P(\mathbf{x})P(\mathbf{y})}$$

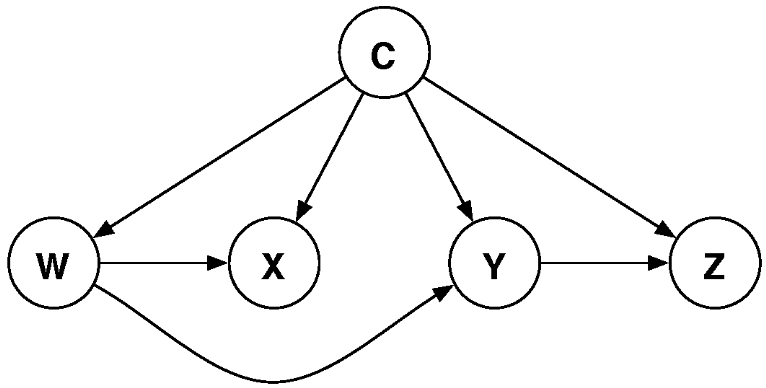
2. Build a complete undirected graph in which the vertices are the variables in $\mathbf{X}$. Annotate the weight of an edge connecting X_i to X_j by $I_{\hat{P}_D}(X_i; X_j)$.

3. Build a maximum weighted spanning tree.

4. Transform the resulting undirected tree to a directed one by choosing a root variable and setting the direction of all edges to be outward from it.

Score Based

- Chow-Liu modification for TAN (HW2)



1. Compute $I_{\hat{P}_D}(A_i; A_j \mid C)$ between each pair of attributes, $i \neq j$.

2. Build a complete undirected graph in which the vertices are the attributes $A_1, \ldots, A_n$. Annotate the weight of an edge connecting A_i to A_j by $I_{\hat{P}_D}(A_i; A_j \mid C)$.

3. Build a maximum weighted spanning tree.

4. Transform the resulting undirected tree to a directed one by choosing a root variable and setting the direction of all edges to be outward from it.

5. Construct a TAN model by adding a vertex labeled by C and adding an arc from C to each A_i.

Slide and other credits

- Zoubin Ghahramani, guest lectures in 10-702
- Andrew Moore tutorial
  – http://www.autonlab.org/tutorials/mle.html
- http://cnx.org/content/m11446/latest/
- Lecture slides by Carlos Guestrin