Georgia Institute of Technology

CSIP Center for Signal & Image Processing

ECE 8803

Parameter Learning in Graphical Models

Module 9: Part A

Maximum Likelihood Estimation

1. Known Structure &
2. Fully Observed Variables

Faramarz Fekri

Center for Signal and Information Processing

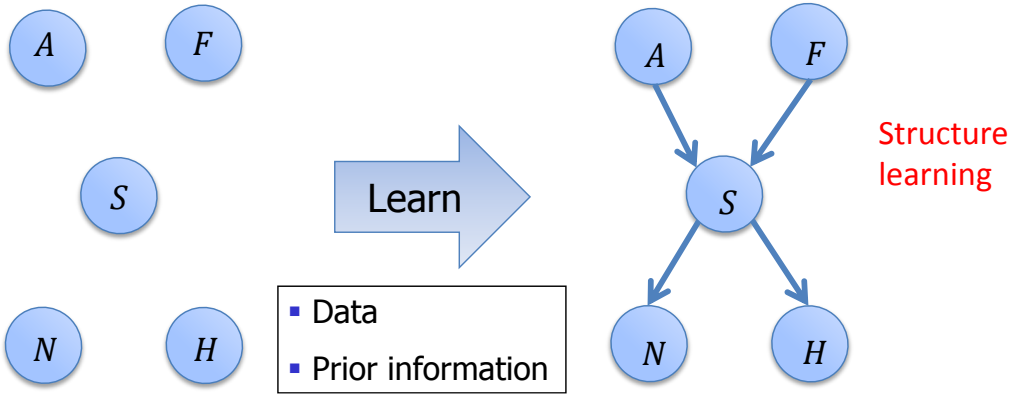
Overview

- Learning problems in graphical models
  - Parameter Estimation Problem
    - Maximum likelihood estimation (MLE)
      - Biased Coin Toss
      - Gaussian (single and multivariate)
    - Sufficient Statistics
      - MLE for Bayesian Networks
      - Limitations of MLE
    - Bayesian estimation (next Lecture)

Chapters 16 and 17 from textbook K&F

Learning Graphical Models

- Up to now, we assumed that the Graphical networks were given.
- Where do the networks come from?
  - Knowledge engineering with aid of experts
  - Learning: automated construction of networks (via instances)
- Our goal: given set of independent samples (assignments of random variables), find the best (i.e., the most likely) graphical model (both structure and the parameters).



(A,F,S,N,H) = (T,F,F,T,F)
(A,F,S,N,H) = (T,F,T,T,F)
…
(A,F,S,N,H) = (F,T,T,T,T)

S \ FA	TF	TF	FT	FF
t	0.9	0.7	0.8	0.2
f	0.1	0.3	0.2	0.8

parameter learning

Learning

- Measures of success
  - How close is the learned network to the original distribution?
    - Use distance measures between distributions
      - Often hard since we do not have the true underlying distribution
    - Instead, evaluate performance by how well the network predicts new unseen examples ("test data")
      - Classification accuracy
  - How close is the structure of the network to the true one?
    - Use distance metric between structures
    - Hard because we do not know the true structure
    - Instead, ask whether independencies learned hold in test data

Prior Knowledge

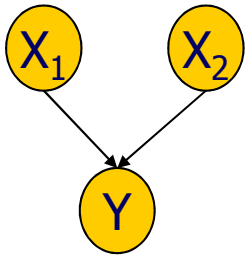
- Prespecified structure
  - Need to learn only CPDs
- Prespecified variables
  - Need to learn both network structure and CPDs
- Hidden variables
  - Need to learn hidden variables, structure, and CPDs

- Complete/incomplete data
  - Missing data
  - Unobserved variables

Learning Problems in GM (I)

- Four types of problems will be covered
  1. Fully observed (complete) data, and known structure:
     - Data does not contain missing values
     - Goal: Parameter (CPD) estimation

Initial network

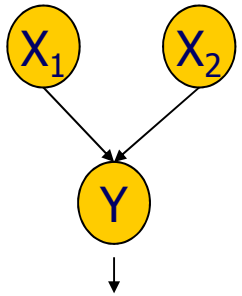


Input Data

X_1	X_2	Y
x_1^0	x_2^1	y^0
x_1^1	x_2^0	y^0
x_1^0	x_2^1	y^1
x_1^0	x_2^0	y^0
x_1^1	x_2^1	y^1
x_1^0	x_2^1	y^1
x_1^1	x_2^0	y^0

Learning

- Data
- Prior information
- Choice of parametric family for each CPD, e.g., $P(X_i|Pa(X_i))$



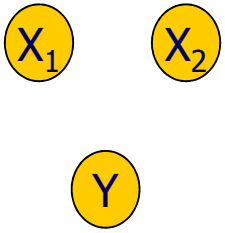
X_1	X_2	$P(Y/X_1,X_2)$ y^0	y^1
x_1^0	x_2^0	1	0
x_1^0	x_2^1	0.2	0.8
x_1^1	x_2^0	0.1	0.9
x_1^1	x_2^1	0.02	0.98

CPD

Learning Problems in GM (II)

2. Fully observed (complete) data, and unknown structure:
   - Data does not contain missing values
   - Goal: Structure learning & Parameter (CPD) estimation

Initial network

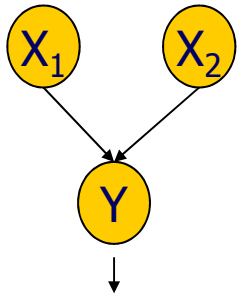


Input Data

X_1	X_2	Y
x_1^0	x_2^1	y^0
x_1^1	x_2^0	y^0
x_1^0	x_2^1	y^1
x_1^0	x_2^0	y^0
x_1^1	x_2^1	y^1
x_1^0	x_2^1	y^1
x_1^1	x_2^0	y^0

Learning

- Data
- Prior information
- Choice of parametric family for each CPD, e.g., $P(X_i|Pa(X_i))$



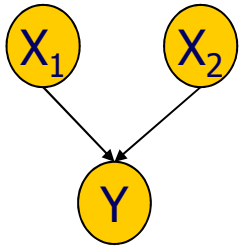
X_1	X_2	$P(Y/X_1,X_2)$ y^0	y^1
x_1^0	x_2^0	1	0
x_1^0	x_2^1	0.2	0.8
x_1^1	x_2^0	0.1	0.9
x_1^1	x_2^1	0.02	0.98

CPD

Learning Problems in GM (III)

3. Missing (incomplete) data, and known structure:
   - Data contains missing values
   - Goal: Parameter (CPD) estimation

Initial network

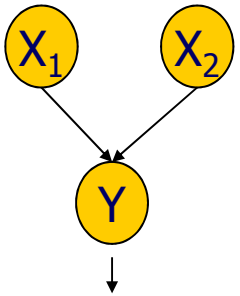


Input Data

X_1	X_2	Y
?	x_2^1	y^0
x_1^1	?	y^0
?	x_2^1	?
x_1^0	x_2^0	y^0
?	x_2^1	y^1
x_1^0	x_2^1	?
x_1^1	?	y^0

Learning

- Data
- Prior information
- Choice of parametric family for each CPD, e.g., $P(X_i|Pa(X_i))$



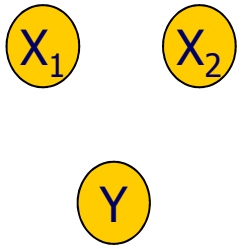
X_1	X_2	$P(Y/X_1,X_2)$ y^0	y^1
x_1^0	x_2^0	1	0
x_1^0	x_2^1	0.2	0.8
x_1^1	x_2^0	0.1	0.9
x_1^1	x_2^1	0.02	0.98

CPD

Learning Problems in GM (IV)

4. Missing (incomplete) data, and unknown structure:
   - Data contains missing values
   - Goal: Structure learning & Parameter (CPD) estimation

Initial network

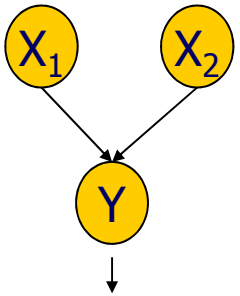


Input Data

X_1	X_2	Y
?	x_2^1	y^0
x_1^1	?	y^0
?	x_2^1	?
x_1^0	x_2^0	y^0
?	x_2^1	y^1
x_1^0	x_2^1	?
x_1^1	?	y^0

Learning

- Data
- Prior information
- Choice of parametric family for each CPD, e.g., $P(X_i|Pa(X_i))$



X_1	X_2	$P(Y/X_1,X_2)$ y^0	y^1
x_1^0	x_2^0	1	0
x_1^0	x_2^1	0.2	0.8
x_1^1	x_2^0	0.1	0.9
x_1^1	x_2^1	0.02	0.98

CPD

Learning Principle in GM

- Estimation principle:
  - Maximal Likelihood Estimation (MLE)
  - Bayesian Estimation (BE)

- Common Feature
  - Utilize distribution factorization
  - Utilize inference algorithms
  - Utilize regularization/prior

	Known Structure	Unknown Structure
Fully observable data	Relatively Easy	Hard
Missing data	Hard (EM)	Very hard

Parameter Estimation Problem

- Biased Coin Toss Example
  - Coin can land in two positions: Head or Tail

- Estimate the probability θ=P(X=h); landing in heads using a biased coin:
  - Given a sequence of m independently and identically distributed (iid) flips D = {x[1],..., x[m]}
  - Example: D = {x[1] ,…., x[m]} = {1,0,1,…,0}, $x_i \in$ {0,1}
  - Denote P(H) and P(T) to mean θ=P(X=h) and P(X=t)=1-θ.

- Assumption: i.i.d samples
- Tosses are controlled by an (unknown) parameter θ
- Tosses are sampled from the same distribution
- Tosses are independent of each other

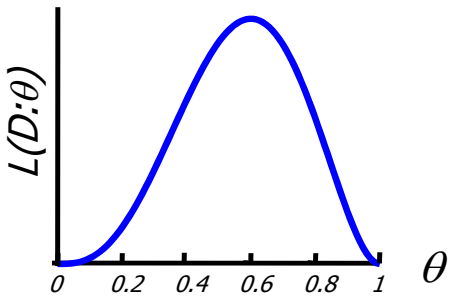
Parameter Estimation: Biased Coin Toss

- Model: $P(x|\theta) = \theta^x(1-\theta)^{1-x}$
  - $P(x|\theta) = \begin{cases} 1-\theta, for\ x=0 \\ \theta, \quad for\ x=1 \end{cases}$

- "Predict the data well" = likelihood of the data given θ
  - $L(D:\theta) = P(D|\theta) = \prod_{i=1}^{m} P(x[i]|x[1],\ldots,x[i-1],\theta) = \prod_{i=1}^{m} P(x[i]|\theta)$
  - $\prod_{i=1}^{m} \theta^{x_i}(1-\theta)^{1-x_i} = \theta^{\sum_i x_i}(1-\theta)^{\sum_i 1-x_i} = \theta^{\#head}(1-\theta)^{\#tail}$

- Example: probability of sequence H,T,T,H,H

$L(H,T,T,H,H:\theta) = P(H|\theta)\,P(T|\theta)\,P(T|\theta)\,P(H|\theta)\,P(H|\theta) = \theta^3(1-\theta)^2$



Maximum Likelihood Estimator (MLE)

- MLE is a very popular estimator, which is simple and has good statistical properties
- Find parameter θ that maximizes L(D:θ), (assume N observed data)
  - $\theta = argmax_\theta P(D|\theta) = argmax_\theta \prod_{i=1}^{N} P(x_i|\theta)$
- MLE for Biased Coin:
  - Objective function: log likelihood,

$$l(\theta;\ D) = \log P(D|\theta) = \log \theta^{n_h}(1-\theta)^{n_t} = n_h \log\theta + (N - n_h)\log(1-\theta)$$

  - We need to maximize this w.r.t. θ
  - Take derivatives w.r.t. θ

$$\frac{\partial l}{\partial \theta} = \frac{n_h}{\theta} - \frac{(N-n_h)}{1-\theta} = 0 \Rightarrow \hat{\theta}_{MLE} = \frac{n_h}{N} \text{ or } \hat{\theta}_{MLE} = \frac{1}{N}\sum_i x_i$$

Sufficient Statistics

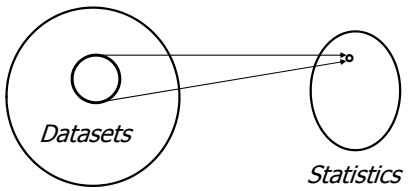
- For computing the parameter θ of the coin toss example, we only needed M_H and M_T since

$$L(\theta : D) = P(D : \theta) = \theta^{M_H}(1-\theta)^{M_T}$$

- M_H and M_T are sufficient statistics.
- A function s(D) is a sufficient statistic from instances to a vector in R^k if, for any two datasets D and D′ and any $\theta \in \Theta$, we have

$$\sum_{x[i]\in D} s(x[i]) = \sum_{x[i]\in D'} s(x[i]) \quad \Rightarrow \quad L(D:\theta) = L(D':\theta)$$

- We often refer to the tuple $\sum_{x[i]\in D} s(x[i])$ as the sufficient statistics of the Data set D.



Sufficient Statistics for Multinomial

- Y: multinomial, k values (e.g. result of a dice throw)
- A sufficient statistics for a dataset D over Y is the tuple of counts $\langle M_1,\ldots M_k\rangle$ such that M_i is the number of times that $Y=y_i$ is in D.
- Likelihood function:

$$L(D:\theta)=\prod_{i=1}^{k}\theta_i^{M_i} \quad \text{where} \quad \theta_i = P(Y=y^i)$$

- MLE Principle: Choose θ such that it maximizes L(D:θ).
- It can be shown that Multinomial MLE:

$$\theta_i = \frac{M_i}{\sum_{i=1}^{k} M_i}$$

Sufficient Statistic for Single Variable Gaussian

- Single variable Gaussian distribution: $X \sim N(\mu, \sigma^2)$
  - Probability density function (pdf):

$$p(X) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

  - Can be written as:

$$p(X) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-x^2 \frac{1}{2\sigma^2} + x\frac{\mu}{\sigma^2} - \frac{\mu^2}{\sigma^2} \right)$$

  - It can be shown that sufficient statistics for Gaussian:

$$\langle M, \Sigma_m x[m], \Sigma_m x[m]^2 \rangle$$

- MLE Principle: Choose θ such that it maximizes $L(D:\theta)$.
- It can be shown that Gaussian MLE:

$$\mu = \frac{1}{M}\sum_m^M x[m]$$

$$\sigma = \sqrt{\frac{1}{M}\sum_m^M (x[m]-\mu)^2}$$

Multivariate-Gaussian

- Likewise, we can show MLE for a multivariate-Gaussian:

$$x_n = \begin{pmatrix} x_{n,1} \\ x_{n,2} \\ \vdots \\ x_{n,K} \end{pmatrix}$$

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

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

$$X = \begin{pmatrix} --- x_1^T --- \\ --- x_2^T --- \\ \vdots \\ --- x_N^T --- \end{pmatrix}$$

where the scatter matrix is

$$S = \sum_n (x_n - \mu_{ML})(x_n - \mu_{ML})^T = \left(\sum_n x_n x_{n\,n}^T\right) - N\mu_{ML}\mu_{ML}^T$$

- The sufficient statistics are $\Sigma_n x_n$ together with $X^TX=\Sigma_n x_n x_n^T$.
- Remark: *X^TX may not be a full rank matrix. As such S would not be invertible.*

MLE for Vanilla Bayesian Networks (I)

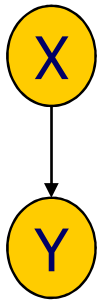
- Parameters
  - θ_{x0}, θ_{x1}
  - $\theta_{y0|x0}$, $\theta_{y1|x0}$, $\theta_{y0|x1}$, $\theta_{y1|x1}$
- Training data:
  - tuples <x[m],y[m]> m=1,…,M
- Likelihood function:

$$L(D:\theta) = \prod_{m=1}^{M} P(x[m], y[m]:\theta)$$

$$= \prod_{m=1}^{M} P(x[m]:\theta_X) P(y[m] \mid x[m]:\theta_{Y|X})$$

$$= \left(\prod_{m=1}^{M} P(x[m]:\theta_X) \right) \left(\prod_{m=1}^{M} P(y[m] \mid x[m]:\theta_{Y|X}) \right)$$

X	
x^0	x^1
0.7	0.3



	Y	
X	y^0	y^1
x^0	0.95	0.05
x^1	0.2	0.8

→ Likelihood decomposes into two separate terms, one for each variable ("decomposability of the likelihood function")

-Lee

MLE for Vanilla Bayesian Networks (II)

- Terms further decompose by CPDs:

$$\prod_{m=1}^{M} P(y[m] \mid x[m] : \theta) = \prod_{m:x[m]=x^0} P(y[m] \mid x[m] : \theta_{Y|X}) \prod_{m:x[m]=x^1} P(y[m] \mid x[m] : \theta_{Y|X})$$
$$= \prod_{m:x[m]=x^0} P(y[m] \mid x[m] : \theta_{Y|x^0}) \prod_{m:x[m]=x^1} P(y[m] \mid x[m] : \theta_{Y|x^1})$$

- By sufficient statistics

$$\prod_{m:x[m]=x^1} P(y[m] \mid x[m] : \theta_{Y|x^1}) = \theta_{y^0|x^1}^{M[x^1,y^0]} \cdot \theta_{y^1|x^1}^{M[x^1,y^1]}$$

where $M[x^1,y^1]$ is the number of data instances in which X takes the value x^1 and Y takes the value y^1

- MLE

$$\theta_{y^0|x^1} = \frac{M[x^1,y^0]}{M[x^1,y^0]+M[x^1,y^1]} = \frac{M[x^1,y^0]}{M[x^1]}$$

-Lee

MLE for General Bayesian Networks

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

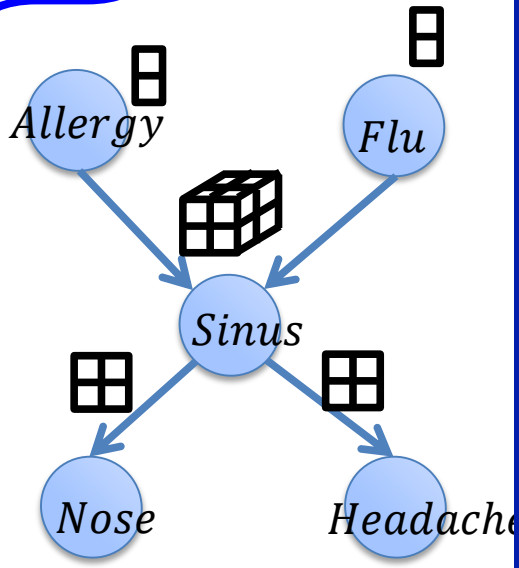
$$l(\theta;\ D) = \log P(D|\theta)$$

$$= \log \prod_i \prod_j P\left(x_j^i \left| pa_{X_j}^i, \theta_j\right.\right) = \sum_i \sum_j \log P\left(x_j^i \left| pa_{X_j}^i, \theta_j\right.\right)$$

$$= \underbrace{\sum_i \log P\left(a^i|\theta_a\right)}_{\text{example}} + \underbrace{\sum_i \log P\left(f^i|\theta_f\right)}_{\text{One term for each CPT; break up MLE problem into independent subproblems}} + \sum_i logP\left(s^i\left|a^i, f^i, \theta_s\right.\right) + \sum_i logP(h^i|s^i, \theta_h)$$

- Here we just need to estimate each CPT separately:
  - For each variable Xi:

$$P_{MLE}\left(X_i = x_i \left| Pa_{X_i} = u\right.\right) = \frac{\#(X_i{=}x\ , Pa_{X_i}{=}u)}{\#(Pa_{X_i}{=}u)}$$

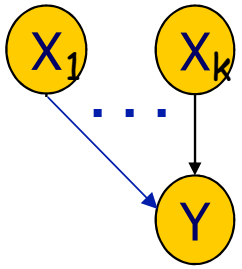


MLE for Table CPD in Bayesian Networks

- Multinomial CPD

$$L_Y(D:\theta_{Y|\mathbf{X}}) = \prod_m \theta_{y[m]|\mathbf{X}[m]}$$

$$= \prod_{\mathbf{x}\in Val(\mathbf{X})} \left[\prod_{y\in Val(Y)} \theta_{y|\mathbf{x}}^{M[\mathbf{x},y]} \right]$$



- For each value $\mathbf{x}\in\mathbf{X}$, we get an independent multinomial problem where the MLE is

$$\theta_{y^i|\boldsymbol{x}} = \frac{M[\boldsymbol{x}, y^i]}{M[\boldsymbol{x}]}$$

Limitations of MLE

- Coin A is tossed 10 times, and comes out 'head' 3 of the 10 tosses,
  - Then MLE will give probability of head = 0.3
- Coin B is tossed 1,000,000 times, and comes out 'head' 300,000 of the 1,000,000 tosses
  - MLE will give probability of head = 0.3
- Shall we place the same bet on the next Coin A toss as we would on the next Coin B toss?
- We need to incorporate prior knowledge
  - Prior knowledge should only be used as a guide

Frequentist vs Bayesian Parameter Estimation

- Frequentists think of a parameter as a fixed, unknown constant, not a random variable (in Bayesian)
- Hence different "objective" estimators, instead of Bayes' rule:
  - These estimators have different properties, such as being "unbiased", "minimum variance", etc.
  - MLE is one popular example in frequentists method.
- Bayesian treat the unknown parameters as a random variable, and incorporates prior knowledge into estimate.
- Bayesian estimation has been criticized for being "subjective"