Machine Learning, CS 6375

Vibhav Gogate
University of Texas, Dallas

MLE Parameter Learning for Bayesian networks

What we will cover

Type of Data sets:

- Fully observed
- Partially observed

Tasks:

- Parameter Learning
- Structure Learning

Approach:

- Maximum likelihood estimation approach
- Bayesian approach

8 combinations and we will study the 2 highlighted combinations.

PART 1

Fully Observed Data
Parameter Learning
MLE approach

Maximum Likelihood Estimation principles

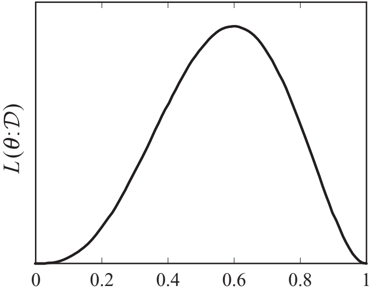
Single variable example: A biased coin

- Two outcomes: *head* and *tail*
- Data set: Tosses of the biased coin
- Task: Estimate the probability of heads/tails on the next flip
- Assumption: the process is controlled by a probability distribution $\Pr(x)$ where $x \in \{h, t\}$
- Value of $\Pr(x = h) = \theta$ if 60 out of 100 tosses yield heads.

MLE scoring for the coin example

Distribution: $\Pr(x = h) = \theta$ and $\Pr(x = t) = 1 - \theta$

- Evaluation metric: How well we can predict the data?
- Example data: H, H, T, H, T
- Likelihood of data $= \prod_i \Pr(x_i) = \theta.\theta.(1-\theta).\theta.(1-\theta)$



MLE scoring for the coin example: Analytical derivation

Distribution: $\Pr(x = h) = \theta$ and $\Pr(x = t) = 1 - \theta$.

- Log-Likelihood function

$$LogL(\theta) = \log(\theta^{\#heads}.(1-\theta)^{\#tails})$$

$$= \#heads.\log(\theta) + \#tails.\log(1-\theta)$$

- MLE Aim: Find θ^* such that $LogL(\theta^*)$ is maximum.
- Differentiate the likelihood function with respect to θ and set the derivative to zero. We get:

$$\theta^* = \frac{\#heads}{\#heads + \#tails}$$

Extending the MLE principle to a Bayesian network

Given a Bayesian network $\Pr(x) = \prod_{i=1}^{n} \theta_{x_i|pa(x_i)}$

- Decomposition of Likelihood function

$$
\begin{aligned}
L(\theta, \mathcal{D}) &= \prod_{j=1}^{m} \Pr(x^{(j)}) \\
&= \prod_{j=1}^{m} \prod_{i=1}^{n} \theta_{x_i^{(j)}|pa(x_i^{(j)})} \\
&= \prod_{i=1}^{n} \prod_{j=1}^{m} \theta_{x_i^{(j)}|pa(x_i^{(j)})}
\end{aligned}
$$

- Each term is a conditional likelihood of a variable given its parents

Extending the MLE principle to a Bayesian network

Given a Bayesian network $\Pr(x) = \prod_{i=1}^{n} \theta_{x_i|pa(x_i)}$

$$L(\theta, \mathcal{D}) = \prod_{i=1}^{n} \prod_{j=1}^{m} \theta_{x_i^{(j)}|pa(x_i^{(j)})}$$

- Let $\#(x_i, pa(x_i))$ be the number of times the tuple $(x_i, pa(x_i))$ appears in the data set. We can write Likelihood function as:

$$L(\theta, \mathcal{D}) = \prod_{i=1}^{n} \theta_{x_i|pa(x_i)}^{\#(x_i, pa(x_i))}$$

Extending the MLE principle to a Bayesian network

Given a Bayesian network $\Pr(x) = \prod_{i=1}^{n} \theta_{x_i|pa(x_i)}$

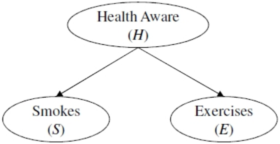
- Given (fully observed) data $\mathcal{X}$, MLE solution is:

$$\theta^*_{x_i|pa(x_i)} = \frac{\#(x_i, pa(x_i))}{\#(pa(x_i))}$$

  where $\#(x_i, pa(x_i))$ is the number of times the tuple $(x_i, pa(x_i))$ appears in $\mathcal{X}$. $\#(pa(x_i))$ is the number of times the tuple $pa(x_i)$ appears in $\mathcal{X}$.

- $\#(x_i, pa(x_i))$ is called the sufficient statistic.
- Any function of the data is called a statistic. A sufficient statistic is a statistic that contains all of the information in the data set that is needed for a particular estimation task.

MLE Learning example: Bayesian network



(a) network structure

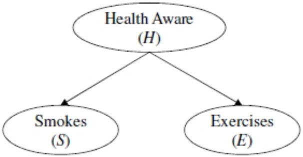
Case	H	S	E
1	T	F	T
2	T	F	T
3	F	T	F
4	F	F	T
5	T	F	F
6	T	F	T
7	F	F	F
8	T	F	T
9	T	F	T
10	F	F	T
11	T	F	T
12	T	T	T
13	T	F	T
14	T	T	T
15	T	F	T
16	T	F	T

(b) complete data

H	S	E	$\Pr_{\mathcal{D}}(.)$
T	T	T	$2/16$
T	T	F	$0/16$
T	F	T	$9/16$
T	F	F	$1/16$
F	T	T	$0/16$
F	T	F	$1/16$
F	F	T	$2/16$
F	F	F	$1/16$

(c) empirical distribution

MLE Learning example: Bayesian network



We have the following parameter estimates:

H	θ_H^{ml}
h	$3/4$
$\bar{h}$	$1/4$

H	S	$\theta_{S\|H}^{ml}$
h	s	$1/6$
h	$\bar{s}$	$5/6$
$\bar{h}$	s	$1/4$
$\bar{h}$	$\bar{s}$	$3/4$

H	E	$\theta_{E\|H}^{ml}$
h	e	$11/12$
h	$\bar{e}$	$1/12$
$\bar{h}$	e	$1/2$
$\bar{h}$	$\bar{e}$	$1/2$

MLE Learning: Bayesian network (fully observable case)

Impact of data set size

- ML estimate will have different values depending upon the size of the data set
- The variance of the estimate will decrease as the data set increases in size.

Theorem:

The distribution of the ML estimate is asymptotically normal and can be approximated by a Gaussian with mean $\Pr(x_i|pa(x_i))$ and variance:

$$\frac{\Pr(x_i|pa(x_i))\times(1-\Pr(x_i|pa(x_i)))}{N\times\Pr(pa(x_i))}$$

Issue: $\Pr(pa(x_i))$ should not be too small.

PART 2

Partially Observed Data
Parameter Learning
MLE approach

- Examples: missing data, hidden variables, some variables are just not observable
- Gradient Ascent (Not covered)
- Expectation maximization (The EM algorithm)

Partially Observed Data (POD)

- Missing data, hidden variables
- $H, T, H, ?, T, ?, \ldots$
- Why is the data missing?
  - Randomly missing
  - Deliberately missing

Why is parameter learning in presence of POD challenging?

Likelihood function for POD:

$$L(\theta, \mathcal{X}) = \prod_{j=1}^{m} \sum_{\mathbf{y} \notin \mathbf{x}^{(j)}} \Pr_{\theta}(\mathbf{x}^{(j)}, \mathbf{y})$$

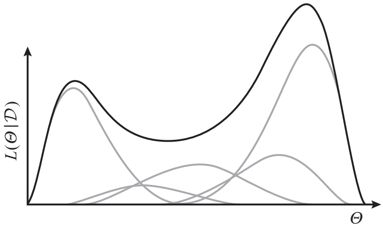
Compare with Likelihood function for FOD:

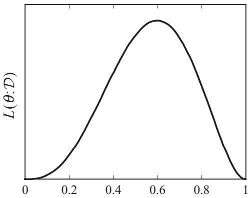
$$L(\theta, \mathcal{X}) = \prod_{j=1}^{m} \Pr_{\theta}(\mathbf{x}^{(j)})$$

Likelihood function for POD:

- is not unimodal.
- cannot be expressed in closed form
- is not decomposable

Why is parameter learning in presence of POD challenging?





POD case:
Each point in the sum yields a unimodal distribution. When combined, we get a multi-modal distribution.

FOD case:
Unimodal distribution

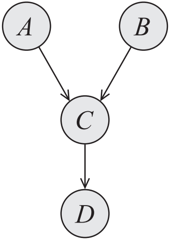
- The optimization problem, a.k.a. maximizing our objective, the likelihood of the data is hard. We need an iterative approach.

Approach 1: The Expectation Maximization (EM) Algorithm

- Start with random parameters
- Repeat until convergence
  1. Complete the incomplete data using current parameters.
  2. Update the parameters based on the completed data

STEP 1: computes expected sufficient statistics (E-step)
STEP 2: maximizes the likelihood (M-step)

The Expectation Maximization Algorithm: Example



$\theta_a = .3$
$\theta_b = .9$
$\theta_{c|\bar{a},\bar{b}} = .83$
$\theta_{c|\bar{a},b} = .09$
$\theta_{c|a,\bar{b}} = .6$
$\theta_{c|a,b} = .2$
$\theta_{d|\bar{c}} = .1$
$\theta_{d|c} = .8$

Data instance: $(a, ?, ?, \bar{d})$
How to complete this example?
For each possible completion

- STEP 1: Compute how likely the completion is.
- STEP 2: Data set is now weighted

The Expectation Maximization Algorithm: E-Step

- Data set is now bigger and weighted
- $(a, ?, ?, \bar{d})$ corresponds to four weighted examples
  - $(a, b, c, \bar{d})$, weight $= .0492$
  - $(a, b, \bar{c}, \bar{d})$, weight $= .8852$
  - $(a, \bar{b}, c, \bar{d})$, weight $= .0164$
  - $(a, \bar{b}, \bar{c}, \bar{d})$, weight $= .0492$
- Intuition is nice. But if a large number of values are missing, the amount of computation involved is huge!!! (exponential in the number of missing values).
- Fortunately, we only need to estimate the sufficient statistics which do not require access to the completed data.

The Expectation Maximization Algorithm: M-Step

Updating: $\theta_{d|\bar{c}}$

- Unweighted MLE estimate:

$$\theta_{d|\bar{c}} = \frac{\#(d, \bar{c})}{\#(\bar{c})}$$

- Weighted MLE estimate:

$$\theta_{d|\bar{c}} = \frac{TotalWeight(d, \bar{c})}{TotalWeight(\bar{c})} = \frac{\sum_{j=1}^{m} \Pr_{\theta}(d, \bar{c}|\mathbf{x}^{(j)})}{\sum_{j=1}^{m} \Pr_{\theta}(\bar{c}|\mathbf{x}^{(j)})}$$

$\Pr_{\theta}(d, \bar{c}|\mathbf{x}^{(j)})$ and $\Pr_{\theta}(\bar{c}|\mathbf{x}^{(j)})$ are the conditional marginal probabilities of the partial assignments $(d, \bar{c})$ and $\bar{c}$ given evidence $\mathbf{x}^{(j)}$ and the current setting of parameters θ. They can be computed using variable elimination or belief propagation.

EM: Properties

- EM may converge to different parameters, with different likelihoods, depending on the initial estimates $\theta^{(0)}$ that it starts with.
- Each iteration of the EM algorithm will have to perform inference on a Bayesian network.
- In each iteration, the algorithm computes the probability of each instantiation $(x, \mathbf{u})$ given each example as evidence.
- All of these computations correspond to posterior marginals over network families. Namely, the require inference. That is why inference is the key problem in Bayesian networks.

EM: Properties

- EM parameter estimates are the only estimates that maximize the expected log-likelihood function
- EM is indeed searching for estimates that maximize the expected log-likelihood function, which also explains its name.
- Parameters that maximize the expected log-likelihood function cannot decrease the log-likelihood function.
  - Each iteration of EM can only increase the likelihood and never decrease it.
  - It will always converge to a local maxima.