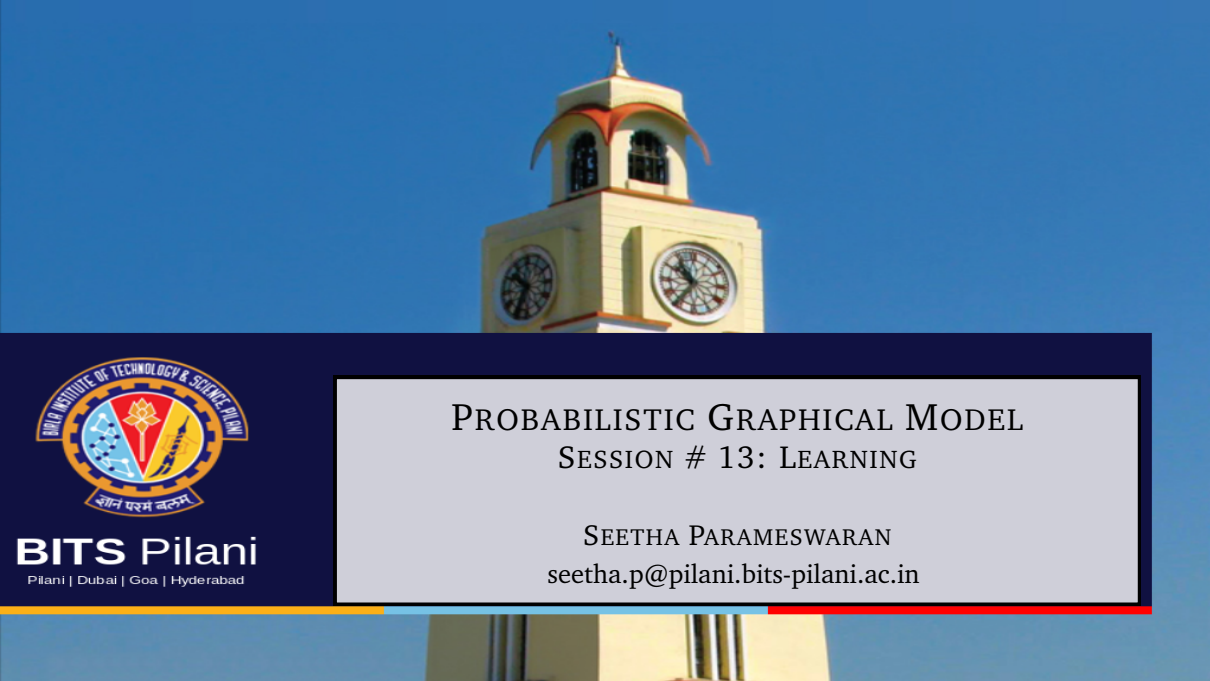


BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Probabilistic Graphical Model

Session # 13: Learning

Seetha Parameswaran
seetha.p@pilani.bits-pilani.ac.in

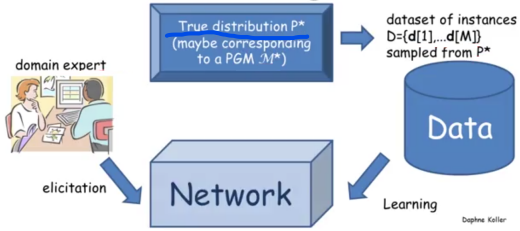
Table of Contents

Motivation

- For Inference or predictions, the starting point was the PGM. The structure and parameters were part of the input.
- How to acquire a model?
  - Construct the network by hand, with the help of an expert – “manual” network construction
  - Learn a model using a set of examples generated from the distribution we wish to model.
- Predictions of structured objects; sequences, graphs, trees
- Incorporate prior knowledge into model.
- Learning a single model for multiple tasks.
- Framework for knowledge discovery.

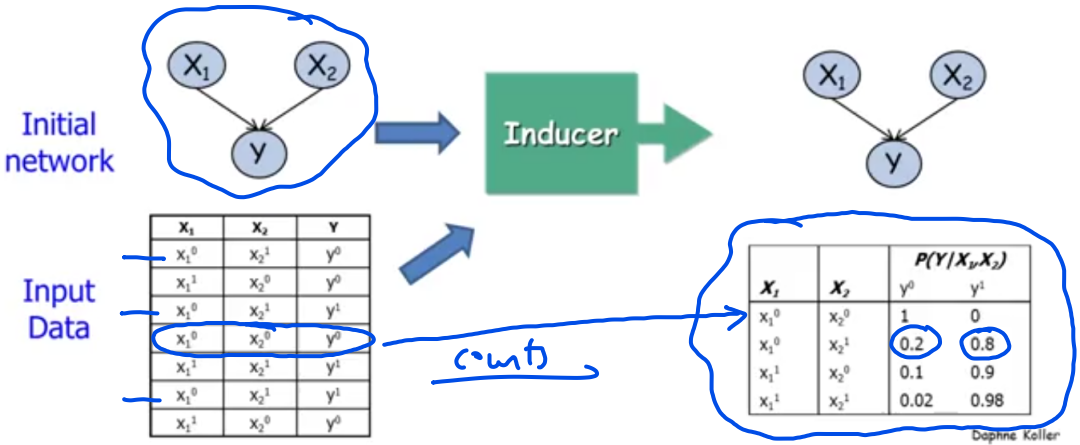
Model Learning

- The task of constructing a model from a set of instances is generally called **model learning**.
- Goal: Learn a model $\tilde{\mathcal{M}}$ from a family of models that defines a distribution $P_{\tilde{\mathcal{M}}}$.

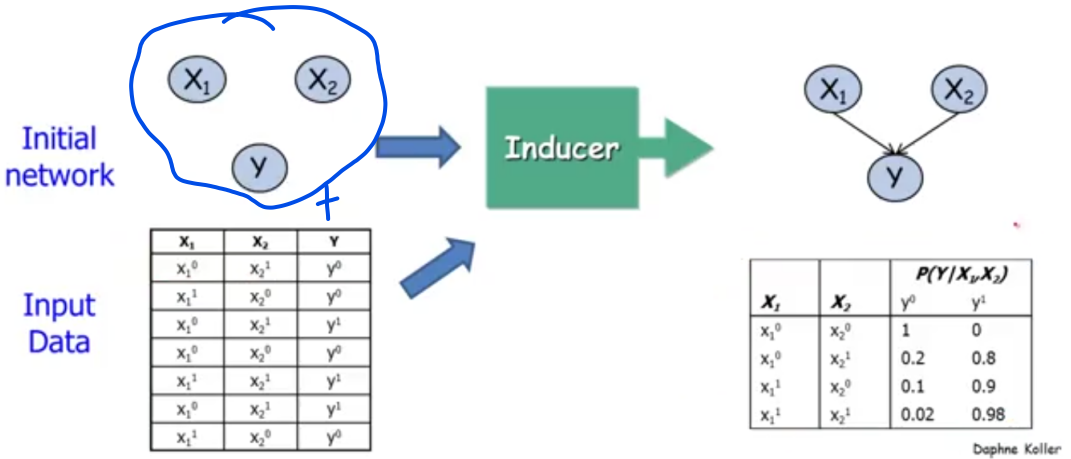


- Amount of data is insufficient, esp for high dimensional distributions. Select $\tilde{\mathcal{M}}$ so as to construct the "best" approximation to $\mathcal{M}^*$.

Known Structure Complete Data



Unknown Structure Complete Data

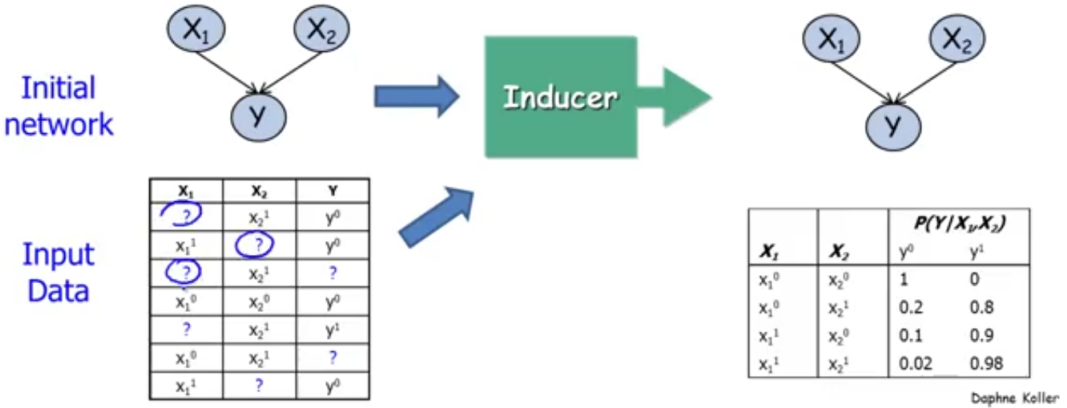


X_1	X_2	Y
x_1^0	x_2^1	y^0
x_1^1	x_2^0	y^0
x_1^0	x_2^1	y^1
x_1^0	x_2^0	y^0
x_1^1	x_2^1	y^1
x_1^0	x_2^1	y^1
x_1^1	x_2^0	y^0

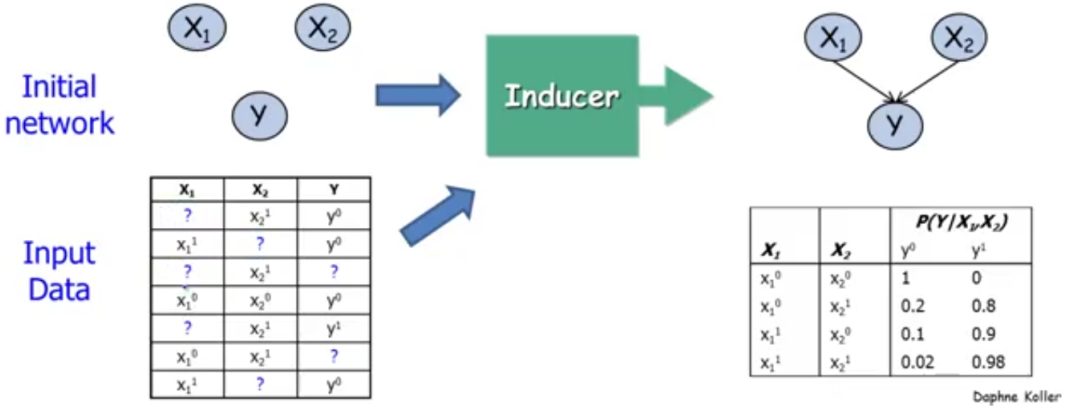
		$P(Y/X_1,X_2)$	
X_1	X_2	y^0	y^1
x_1^0	x_2^0	1	0
x_1^0	x_2^1	0.2	0.8
x_1^1	x_2^0	0.1	0.9
x_1^1	x_2^1	0.02	0.98

Daphne Koller

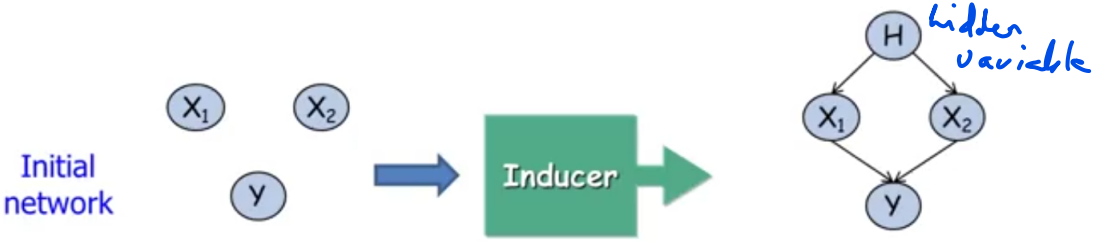
Known Structure Incomplete Data



Unknown Structure Incomplete Data



Latent (Hidden) Variables Incomplete Data

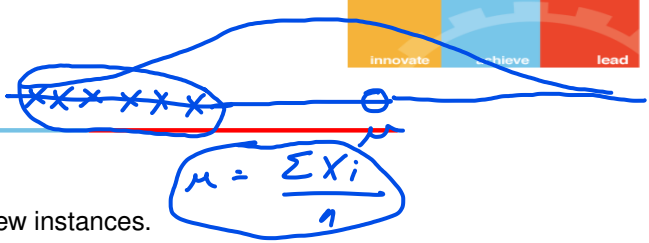


X_1	X_2	Y
?	x_2^1	y^0
x_1^1	?	y^0
?	x_2^1	?
x_1^0	x_2^0	y^0
?	x_2^1	y^1
x_1^0	x_2^1	?
x_1^1	?	y^0

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

Daphne Koller

Goals of Learning I



- Density estimation:
  - Answer general queries about new instances.
  - Metric: Training set likelihood

$$P(\mathcal{D} : \mathcal{M}) = \prod_m P(d[m] : \mathcal{M})$$

  - Care about new data: generalization performance – Evaluate on test set likelihood $P(\mathcal{D}' : \mathcal{M})$
  - Minimize the expected loss :

$$\mathbb{E}_{\mathcal{D}}[loss(d : \mathcal{M})] = \frac{1}{|\mathcal{D}|} \sum_{d \in \mathcal{D}} loss(d : \mathcal{M})$$

Goals of Learning II

- Prediction or Classification Task:
  - Specific prediction task on new instances.
  - Select a MAP assignment to predict a set of variables **Y**, given a set of observed variables **X**.
  - Eg: Text document classification, Image segmentation, speech recognition
  - Select model to optimize
    - ★ likelihood

$$\prod_m P(d[m] : \mathcal{M})$$

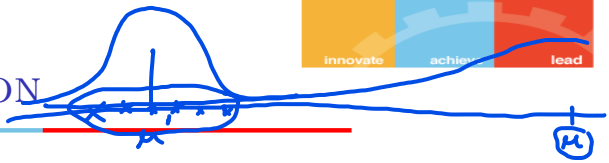
    - ★ conditional likelihood

$$\prod_m P(y[m] \mid x[m] : \mathcal{M})$$

Goals of Learning III

- Knowledge Discovery of $\mathcal{M}^*$:
  - Discover knowledge about P^*.
  - Distinguish direct and indirect dependencies.
  - Possibly directionality of edges.
  - Presence and location of hidden variables.
  - Reconstruct correct model $\mathcal{M}^*$.
  - Measure the success in terms of the model = differences between $\mathcal{M}^*$ and $\tilde{\mathcal{M}}$.

Learning as Optimization



- **Hypothesis space** – a set of candidate models.
- **Objective function** – a criterion for quantifying our preference for different models.
- **Learning Task** – find a high-scoring model within the hypothesis space.
- Use data $\mathcal{D}$ to define an **empirical distribution** $\hat{P}_{\mathcal{D}}$

indicator function

$$\hat{P}_{\mathcal{D}}(A) = \frac{1}{M} \sum_{m} \mathbf{1}\{d[m] \in A\}$$

count

- The probability of the event A is simply the fraction of training examples that satisfy A.
- Use of the empirical log-loss (or log-likelihood) as the objective.
- This type of objective tends to over-fit the learned model to the training data . Use regularization, cross-validation techniques.

Generative Training

- Perform a particular task such as predicting **Y** from **X**.
- Goal: get $\tilde{\mathcal{M}}$ close to overall joint distribution $P^*(\mathbf{Y}, \mathbf{X})$.
- Model trained to generate all the variables.
- Naive Bayes model
- Higher bias
- Encode independence assumption about feature variables **X**.
- Defines $\tilde{P}(\mathbf{Y}, \mathbf{X})$ and induces $\tilde{P}(\mathbf{Y} \mid \mathbf{X})$ and $\tilde{P}(\mathbf{X})$ using the same overall model for both.
- Works better when learning from limited amounts of data.

Discriminative Training

- Goal: get $\tilde{P}(\mathbf{Y} \mid \mathbf{X})$ to be close to $P^*(\mathbf{Y} \mid \mathbf{X})$
- Undirected model
- Train a conditional random field (CRF)
- Model directly encodes a conditional distribution $P(\mathbf{Y} \mid \mathbf{X})$.
- Encode independence assumptions about $\mathbf{Y}$ and their dependence on $\mathbf{X}$.
- Find a good fit only to $P^*(\mathbf{Y} \mid \mathbf{X})$ without containing the same model to provide a good fit to $P^*(\mathbf{X})$.

Learning Tasks

Input to learning tasks

- Prior knowledge about $\tilde{\mathcal{M}}$
- Set of $\mathcal{D}$ of data instances $\{d[1], \ldots d[M]\}$ with are sample IID from P^*.

Output of learning tasks

- Model $\tilde{\mathcal{M}}$ with structure and parameters.

3 Axes

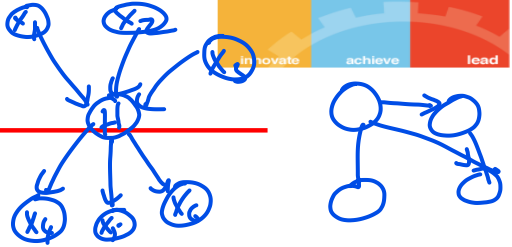
1. The type of graphical model we are trying to learn – a Bayesian network or a Markov network.
2. Hypothesis space
3. Data observability

Learning Tasks

Hypothesis space

- Given a graph structure, and learn only (some of) the parameters.
- We may not know the structure, and we have to learn both parameters and structure from the data.
- We may not even know the complete set of variables over which the distribution P^* is defined. We may only observe some subset of the variables in the domain and possibly be unaware of others.

Learning Tasks



Data Observability

- The data are **complete**, or **fully observed**, so that each of the training instances$d[m]$ is a full instantiation to all of the variables in $\mathcal{X}^*$.
- The data are **incomplete**, or **partially observed**, so that, in each training instance, some variables are not observed.
- The data contain **hidden variables** whose value is never observed in any training instance. The inclusion of a hidden variable in the network can greatly simplify the structure, reducing the complexity of the network that needs to be learned.

Table of Contents

Parameter Estimation in Bayesian Networks

- Network structure is fixed.
- Data-set $\mathcal{D}$ consists of fully observed data instances.

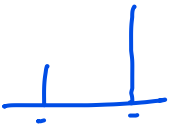
$$\mathcal{D} = \{d[1], \ldots, d[M]\}$$

- Two approaches
  1. Maximum likelihood estimation
  2. Bayesian estimation

Biased Coin Example

- Head and tails outcome are controlled by a parameter θ.
- $\theta =$ frequency of heads in the coin tosses.

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



- The distribution P is Bernoulli distribution.
- The data-set $\mathcal{D}$ is sampled IID from P.

$$\mathcal{D} = \{x[1], \ldots, x[M]\}$$

  - Tosses are independent of each other.
  - Tosses are sampled from the same distribution.
- Goal: Take $\mathcal{D}$ and reconstruct θ.

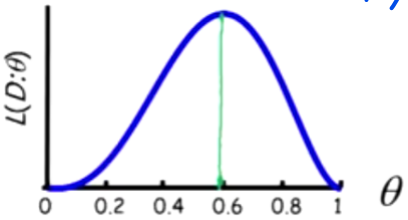
Biased Coin Example

$x[1]$ and $x[2]$ are independent given θ

$P(x[1], x[2]/\theta) = P(x[1]/\theta) P(x[2]/\theta)$

- Suppose we observe: H, T, T, H, H

$$P(X[1] = H) = \theta$$
$$P(X[2] = T) = (1 - \theta) \quad \text{IID samples}$$
$$P(< H, T, T, H, H >: \theta) = \theta(1-\theta)(1-\theta)\theta\theta$$
$$= \theta^3(1-\theta)^2$$



$\max P(\theta_1, \theta_2, \ldots \theta_N/\theta) = \max P(\theta_1/\theta) P(\theta_2/\theta) \ldots P(\theta_N/\theta)$

Biased Coin Example

- The probability of the data changes as a function of θ.
- Define the **likelihood function**:

$$\mathcal{L}(\theta :< H, T, T, H, H >) = P(< H, T, T, H, H >: \theta) = \theta^3(1-\theta)^2$$

- Use the likelihood function as the measure of quality for different parameter values.
- Select the parameter value that maximizes the likelihood; this value is called the **maximum likelihood estimator (MLE)**.
- Observations: M_H heads and M_T tails .
- M_H and M_T are sufficient statistics.

$$\hat{\theta} = \frac{M_H}{M_H + M_T} = \frac{3}{5} = 0.6$$

max log likelihood

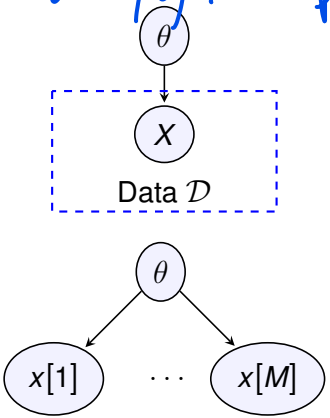
$\log(\theta^3(1-\theta)^2)$

$3\log\theta + 2\log(1-\theta)$

$\frac{3}{\theta} + \frac{2(-1)}{1-\theta} = 0$

$\theta = \frac{3}{5}$

Bernoulli Distribution as PGM



- Distribution

$$P(x[m] \mid \theta) = \begin{cases} \theta & x[m] = x^1 \\ 1 - \theta & x[m] = x^0 \end{cases}$$

- Hypothesis space

$$\Theta = [0, 1] : \sum_i \theta_i = 1$$

- Likelihood function

$$\mathcal{L}(\theta : \mathcal{D}) = P(\mathcal{D} : \theta) = \prod_{m=1}^{M} P(x[m] \mid \theta) = \theta^{M_1}(1-\theta)^{M_0}$$

Bernoulli Distribution as PGM

- Observations: M_1 and M_0
- The counts M_1 and M_0 give a compact distribution of the likelihood. These are called **sufficient statistics**.
- M_1 and M_0 are sufficient statistics.
- Find θ that maximizes likelihood:

$$\mathcal{L}(\theta : \mathcal{D}) = P(\mathcal{D} : \theta) = \theta^{M_1}(1-\theta)^{M_0}$$

- Equivalently maximize log-likelihood:

$$\ell(\theta : \mathcal{D}) = M_1 \log \theta + M_0 \log(1-\theta)$$

- Differentiate log-likelihood and solve for θ as

$$\hat{\theta} = \frac{M_1}{M_1 + M_0}$$

Sufficient Statistics

- Sufficient statistic is a function of the data that summarizes the relevant information for computing the likelihood.

Definition (Sufficient Statistics)

A function $\tau(\xi)$ from instances $\mathcal{X}$ to $\mathbb{R}^\ell$ is a sufficient statistic, if for any two data sets $\mathcal{D}$ and $\mathcal{D}'$ and any $\theta \in \Theta$

$$\text{If } \sum_{\xi[m] \in \mathcal{D}} \tau(\xi[m]) = \sum_{\xi'[m] \in \mathcal{D}'} \tau(\xi'[m])$$

$$\text{then } \mathcal{L}(\theta : \mathcal{D}) = \mathcal{L}(\theta : \mathcal{D}')$$

- The tuple $\sum_{\xi[m] \in \mathcal{D}} \tau(\xi[m])$ is referred to as sufficient statistics of the data-set $\mathcal{D}$.

Multinomial Distribution as PGM

- Multinomial variable X takes the values $x^1, \ldots, x^K$.
- Distribution :

$$P(X : \theta) = \theta_k \quad \text{if} \quad x = x^k$$

- Hypothesis space:

$$\Theta = \left\{ \theta \in [0,1]^K : \sum_i \theta_i = 1 \right\}$$

- Sufficient Statistics:

$$\tau(x^k) = \overbrace{0, \ldots, 0}^{k-1}, 1, \overbrace{0, \ldots, 0}^{n-k}$$

K dimensional vector.

Multinomial Distribution as PGM

- Likelihood function:

$$\mathcal{L}(\theta : \mathcal{D}) = \prod_k \theta_k^{M[k]}$$

- Maximum Likelihood Estimation:

$$\mathcal{L}(\theta : \mathcal{D}) = \max_{\theta \in \Theta} \mathcal{L}(\theta : \mathcal{D})$$

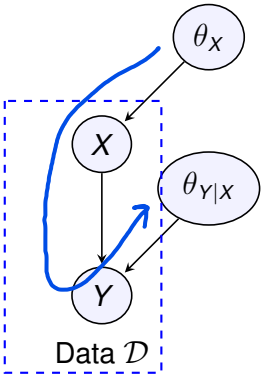
$$\hat{\theta} = \frac{M[k]}{M}$$

MLE Summary

- MLE is a simple principle for estimating or parameter selection given a data-set $\mathcal{D}$.
- Likelihood function uniquely determined by sufficient statistic that summarize $\mathcal{D}$.
- MLE has a closed form solution for many parametric distributions.

Table of Contents

MLE for Bayesian Networks



- A network consisting of two binary variables.
- Each assignment or training instance is given by $< x[m], y[m] >$.
- The network is parametrized by θ, which defines the set of parameters for all the CPDs in the network.
- Parameters for X : θ_{x^1} and θ_{x^0}
- Parameters for Y : $\theta_{Y|X} = \theta_{Y|x^1} \cup \theta_{Y|x^0}$

$$\theta_{Y|x^1} = \{\theta_{y^1|x^1}, \theta_{y^0|x^1}\}$$
$$\theta_{Y|x^0} = \{\theta_{y^1|x^0}, \theta_{y^0|x^0}\}$$

MLE for Bayesian Networks

- Goal: Maximize the likelihood function.

$$\begin{aligned}
\mathcal{L}(\theta : \mathcal{D}) &= \prod_{m=1}^{M} P(x[m], y[m] : \theta) \\
&= \prod_{m} P(x[m] : \theta) P(y[m] \mid x[m] : \theta) \\
&= \left[\prod_{m} P(x[m] : \theta_X) \right] \left[\prod_{m} P(y[m] \mid x[m] : \theta_{Y|X}) \right] \\
&= \left[\prod_{m} P(x[m] : \theta_X) \right] \left[\prod_{m:x[m]=x^0} P(y[m] \mid x[m] : \theta_{Y|x^0}) \right] \left[\prod_{m:x[m]=x^1} P(y[m] \mid x[m] : \theta_{Y|x^1}) \right]
\end{aligned}$$

- The likelihood function decomposes into **local likelihood** terms, each one depends only on the parameters for that variable's CPD.

MLE for Bayesian Networks

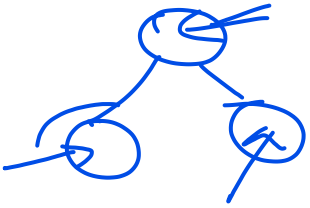
- Goal: Maximize the likelihood function.

$$\mathcal{L}(\theta : \mathcal{D}) = \prod_m P_{\mathcal{G}}(x[m], y[m] : \theta)$$

$$= \prod_m \prod_i P(x_i[m] \mid pa_{X_i}[m] : \theta)$$

$$= \prod_i \left[\prod_m P(x_i[m] \mid pa_{X_i}[m] : \theta) \right]$$

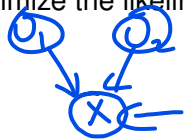
$$= \prod_i L_i(\theta_{X_i|Pa_{X_i}} : \mathcal{D})$$



- When the parameter sets $\theta_{X_i|Pa_{X_i}}$ are disjoint, then MLE can be computed by maximizing each local likelihood separately. The likelihood decomposes as a product of independent terms, one for each CPD in the network.

MLE for Bayesian Networks for Table CPDs

- Goal: Maximize the likelihood function.



$$\mathcal{L}_X(\theta_{X|U} : \mathcal{D}) = \prod_m \theta_{x[m]|u[m]}$$

- MLE parameters

$$\hat{\theta}_{x|u} = \frac{M[u,x]}{M[u]}$$

$M[u,x]$ = frequency of $x[m] = x$ and $u[m] = u$ in $\mathcal{D}$

$$M[u] = \sum_x M[u,x]$$

MLE for Bayesian Network Summary

- For Bayesian Network with disjoint sets of parameters in CPDs, likelihood decomposes as a product of local likelihood functions, one per variable.
- For table CPDs, local likelihood functions further decomposes as a product of likelihood for multinomials, one for each parent combination.

Table of Contents

Drawback of MLE

MLE does not distinguish between

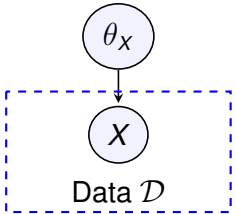
- ~~a biased coin and unbiased coin~~. thumbtack and coin
- 10 tosses and 1,000,000 tosses of the coin.

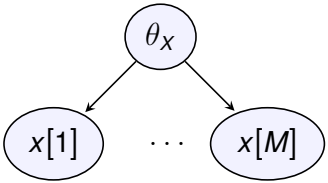
Another approach – Bayesian Estimation.

Joint Probabilistic Model

- If θ is unknown, tosses are not marginally independent.
- Each toss tells us something about θ and about the probability of next toss.
- Tosses are conditionally independent given θ.
- Treat θ as a random variable.

Parameter Estimation as PGM





Distribution $P(x[m] \mid \theta) = \begin{cases} \theta & x[m] = x^1 \\ 1-\theta & x[m] = x^0 \end{cases}$

Prior distribution $P(\theta) \in [0,1]$ Uniform Prior

Parameter Estimation as PGM

- Joint Distribution or Likelihood

$$P(x[1], \ldots, x[M], \theta) = P(x[1], \ldots, x[M])P(\theta)$$
$$= P(\theta)\prod_{m=1}^{M} P(x[m] \mid \theta)$$
$$= P(\theta)\theta^{M[1]}(1-\theta)^{M[0]}$$

- Posterior Distribution

$$P(\theta \mid x[1], \ldots, x[M]) = \frac{P(x[1], \ldots, x[M] \mid \theta)P(\theta)}{P(x[1], \ldots, x[M])}$$

Posterior $\propto$ likelihood $\times$ prior

Parameter Estimation as PGM

- To predict the next value $X[M+1]$; integrate the posterior over θ.

$$P(x[M+1] \mid x[1], \ldots, x[M]) = \int P(x[M+1] \mid \theta) P(\theta \mid x[1], \ldots, x[M]) d\theta$$

$$P(X[M+1] = x^1 \mid x[1], \ldots, x[M]) = \frac{1}{P(x[1], \ldots, x[M])} \int \theta \, \theta^{M[1]} (1-\theta)^{M[0]} d\theta$$

$$= \frac{M[1]+1}{M+2}$$

- As the number of samples grows, the Bayesian estimator and the MLE estimator converge to the same value.

More Detailed Derivation

$$P(x[M+1], D) = \int P(x[M+1], D, \theta)\, d\theta$$

where $D = x[1], x[2], \cdots x[M]$

$$P(x[M+1], D) = \int P(x[M+1]/D, \theta)\, P(D, \theta)\, d\theta$$

$$P(\theta)\, P(x[M+1]/D) = \int P(x[M+1]/D, \theta)\, P(\theta/D)\, P(D)\, d\theta$$

$$P(x[M+1]/D) = \int P(x[M+1]/\theta) P(\theta/D) d\theta$$

Since $P(x[M+1]/D,\theta) = P(x[M+1]/\theta)$

Priors and Posteriors

- Observe a training set $\mathcal{D} = x[1] \ldots x[M]$
- M IID samples of random variable $\mathcal{X}$ from an unknown distribution $P^*(\mathcal{X})$.
- Assume a parametric model $P(\xi \mid \theta)$ with a parameter space Θ.
- Treat θ as a random variable.
- Joint distribution

$$P(\mathcal{D}, \theta) = P(\mathcal{D} \mid \theta) P(\theta)$$

- The first term is the **likelihood function**. Compactly described by using sufficient statistics.
- The second term is the **prior distribution** over the possible values in Θ. Captures initial uncertainty about the parameters.

Priors and Posteriors

- Derive the **posterior probability** over parameters

$$P(\theta|\mathcal{D}) = \frac{P(\mathcal{D} \mid \theta)P(\theta)}{P(\mathcal{D})}$$

- **Marginal likelihood**

$$P(\mathcal{D}) = \int_{\Theta} P(\mathcal{D} \mid \theta)P(\theta)d\theta$$

Priors and Posteriors

- For a multinomial distribution, parameter space Θ is a space of

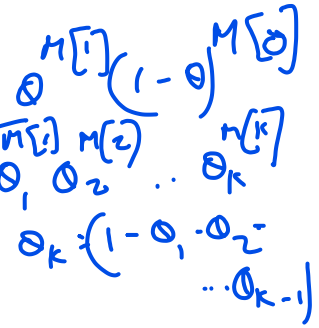
$$\theta = < \theta_1, \ldots, \theta_K >$$

$$\sum_k \theta_k = 1$$

- Likelihood function

$$\mathcal{L}(\theta : \mathcal{D}) = \prod_k \theta_k^{M[k]}$$

- Prior is assumed to follow Dirichlet distribution.



Dirichlet Distribution

Definition (Dirichlet Distribution)

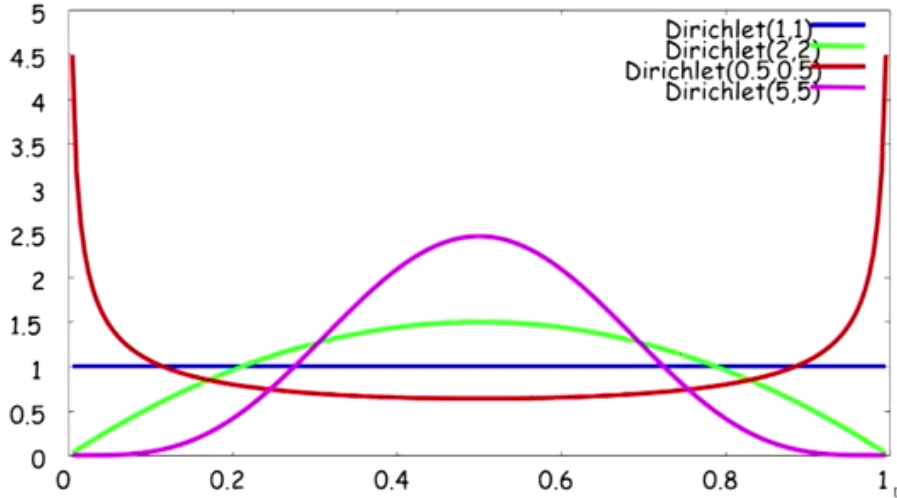
A Dirichlet distribution is specified by a set of hyper-parameters $< \alpha_1, \ldots, \alpha_K >$

$$\theta \sim \textit{Dirichlet}(\alpha_1, \ldots, \alpha_K) \qquad \text{if} \qquad P(\theta) \propto \prod_k \theta_k^{\alpha_k - 1}$$

- If $P(\theta)$ is Dirichlet distribution with hyper-parameters $< \alpha_1, \ldots, \alpha_K >$ and $\alpha = \sum_j \alpha_j$, then

$$\mathbb{E}[\theta_k] = \frac{\alpha_k}{\alpha}$$

Dirichlet Distribution



Dirichlet Priors and Posteriors

- Posterior distribution

$$P(\theta|\mathcal{D}) \propto P(\mathcal{D} \mid \theta) P(\theta)$$

- Likelihood function – multinomial

$$P(\mathcal{D} \mid \theta) = \mathcal{L}(\theta : \mathcal{D}) = \prod_k \theta_k^{M[k]}$$

- Prior distribution – Dirichlet

$$P(\theta) \propto \prod_k \theta_k^{\alpha_k - 1}$$

- Posterior distribution – Dirichlet

Dirichlet Priors and Posteriors

- If Prior $P(\theta)$ is Dirichlet and the Likelihood $P(\mathcal{D} \mid \theta)$ is multinomial, then the Posterior $P(\theta|\mathcal{D})$ is also Dirichlet.

$$\begin{aligned} \textit{Prior} &= \textit{Dirichlet}(\alpha_1, \ldots, \alpha_K) \\ \text{Data counts} &= M_1, \ldots M_K \\ \textit{Posterior} &= \textit{Dirichlet}(\alpha_1 + M_1 \ldots, \alpha_K + M_K) \end{aligned}$$

- Prior and Posterior have the same form of distribution.
- Dirichlet is a conjugate pair for multinomial.

Bayesian Prediction

- To predict for a new sample

$$P(x[M+1] \mid \mathcal{D}) = \int P(x[M+1] \mid \mathcal{D}, \theta) P(\theta \mid \mathcal{D}) d\theta$$

$$= \int P(x[M+1] \mid \theta) P(\theta \mid \mathcal{D}) d\theta$$

$$= \mathbb{E}_{P(\theta \mid \mathcal{D})}[P(x[M+1] \mid \theta)]$$

$$P(x[M+1] = x^k \mid \mathcal{D}) = \frac{M[k] + \alpha_k}{M + \alpha}$$

- Equivalent sample size $= \alpha = \alpha_1 + \ldots + \alpha_k$
- Larger α implies more confidence in our prior.

Example

- For a given binomial data with uniform distribution for parameter θ, it is observed that

$$(M[1], M[0]) = (5, 2)$$

uniform = Dirichlet(1,1)

Predict $P(X[8] = 1)$ using MLE and Bayesian prediction.

- MLE

$$P(X[8] = 1) = \frac{M_1}{M} = \frac{5}{7} = 0.71$$

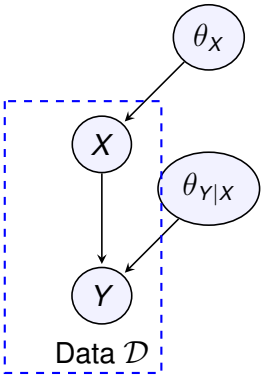
- Bayesian prediction

$$P(X[8] = 1) = \frac{\alpha_1 + M_1}{\alpha + M}$$
$$= \frac{1 + 5}{2 + 7} = \frac{6}{9} = 0.66$$

Bayesian Estimation Summary

- Bayesian Learning treats parameters as random variables.
- Dirichlet distribution as conjugate pair of multinomial distribution.
  - Posterior has the same form as prior.
  - Can be updated in closed form using sufficient statistic from data.
- Bayesian prediction combines sufficient statistic from imaginary Dirichlet sample and real data samples.
- Dirichlet hyper-parameters determine both the prior beliefs and their strengths.
- Bayesian Learning is robust in sparse data regime in terms of its generalization ability.

Bayesian Estimation on Bayesian Network



- A network consisting of two binary variables.
- Training data consists of M observations given by $< x[m], y[m] >$
- Unknown parameters θ_X and $\theta_{Y|X}$
- Instances are independent given the unknown parameters.
- $< x[m], y[m] >$ are d-separated from $< x[m'], y[m'] >$ once we know the parameter variables.
- Assume that the priors for the individual parameters variables are apriori independent. That is, we believe that knowing the value of one parameter tells us nothing about another.

Bayesian Estimation on Bayesian Network

- The Bayesian network have parameters

$$\theta = (\theta_{X_1|Pa_{X_1}}, \ldots, \theta_{X_n|Pa_{X_n}})$$

- Prior distribution satisfies **global parameter independence** if

$$P(\theta) = \prod_i P(\theta_{X_i|Pa_{X_i}})$$

- If the complete data $< x[m], y[m] >$ are observed for all m, then parameters θ_X and $\theta_{Y|X}$ are d-separated.

$$P(\theta_X, \theta_{Y|X} \mid \mathcal{D}) = P(\theta_X \mid \mathcal{D}) P(\theta_{Y|X} \mid \mathcal{D})$$

- Given the data set $\mathcal{D}$, determine the posterior over θ_X independently of the posterior over $\theta_{Y|X}$.

Bayesian Estimation on Bayesian Network

- Let X have parents U. Then the prior $P(\theta_{X|U})$ satisfies **local parameter independence** if

$$P(\theta_{X|U}) = \prod_{u} P(\theta_{X|u})$$

- If $P(\theta)$ satisfies global and local parameter independence then,

$$P(\theta \mid \mathcal{D}) = \prod_{i} \prod_{Pa_{X_i}} P(\theta_{X_i|Pa_{X_i}} \mid \mathcal{D})$$

Bayesian Estimation on Bayesian Network

- Multinomial Parameters $\theta_{X|u}$
- $P(\theta_{X|U})$
  - Dirichlet prior with hyper-parameters $\alpha_{x^1|u}, \ldots, \alpha_{x^K|u}$
- $P(\theta_{X|U} \mid \mathcal{D})$
  - Dirichlet posterior with hyper-parameters $\alpha_{x^1|u} + M[u, x^1], \ldots, \alpha_{x^K|u} + M[u, x^K]$

Summary

- In Bayesian networks, if parameters are independent apriori, then they are also independent in the posterior.
- For multinomial Bayesian networks, estimation uses sufficient statistic $M[x, u]$) (counts).

$$\text{MLE} \quad \hat{\theta}_{X|u} = \frac{M[X, u]}{M[u]}$$

$$\text{BL} \quad P(x \mid U, \mathcal{D}) = \frac{\alpha_{X,u} + M[x, u]}{\alpha_u + M[u]}$$

- Bayesian Learning requires a choice of prior.

Example

Given the following data and the structure that $X \rightarrow Y$, learn the parameters using MLE.

X	Y
0	1
0	1
1	0
1	1
1	0
1	1
1	0
0	1
0	1

$$\hat{\theta}_{X|u} = \frac{M[X,u]}{M[u]}$$

$$\theta_X = P(X) = \begin{bmatrix} 4/9 & 5/9 \end{bmatrix}$$

$$\theta_{Y|X} = P(Y \mid X) = \begin{bmatrix} 0/4 & 4/4 \\ 3/5 & 2/5 \end{bmatrix}$$