Computer Science

CSC480/580: Principles of Machine Learning

Probabilistic ML: Probabilistic Graphical Models

Chicheng Zhang

Outline

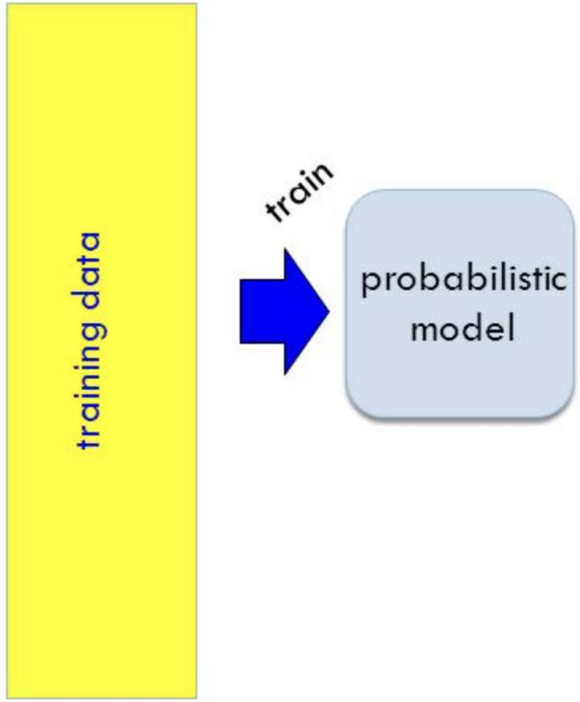
- Probabilistic Graphical Models
- Case study: Naïve Bayes

Outline

- Probabilistic Graphical Models
- Case study: Naïve Bayes

Probabilistic modeling: systematic approach for ML

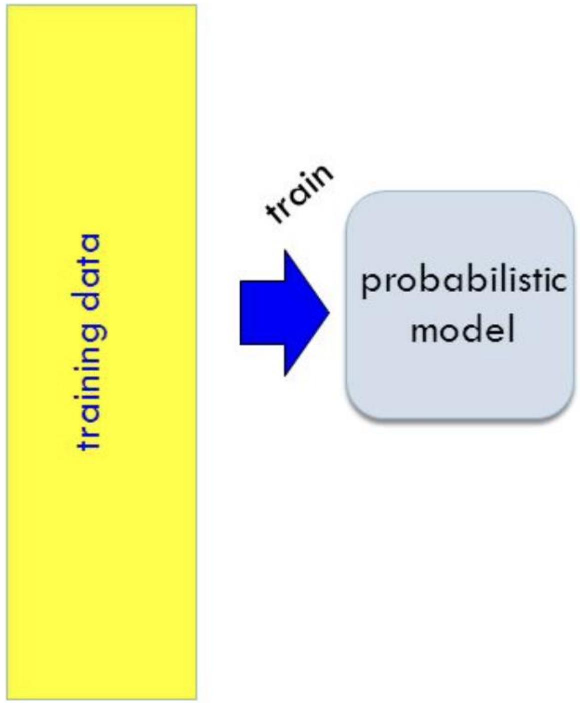
- The recipe:
  1. Model how the data is generated by probabilistic models, but with parameters unspecified (modeling assumption / generative story)
     - Each example $z \sim P(z; \theta)$ for some $\theta \in \Theta$
       - For $z = (x, y)$ => supervised learning
       - For $z = x$ => unsupervised learning
  2. (Training) Learn the model parameter $\hat{\theta}$
     - Important example: maximum likelihood estimation (MLE), $\text{maximize}_{\theta \in \Theta} \log P(z_1, \ldots, z_n; \theta)$
  3. (Test) Make prediction / decision based on the learned model $P(z; \hat{\theta})$
     - Important example: predict using the Bayes classifier of $P(z; \hat{\theta})$ (for supervised learning)



http://slideplayer.com/slide/4527958/

Probabilistic modeling: systematic approach for ML

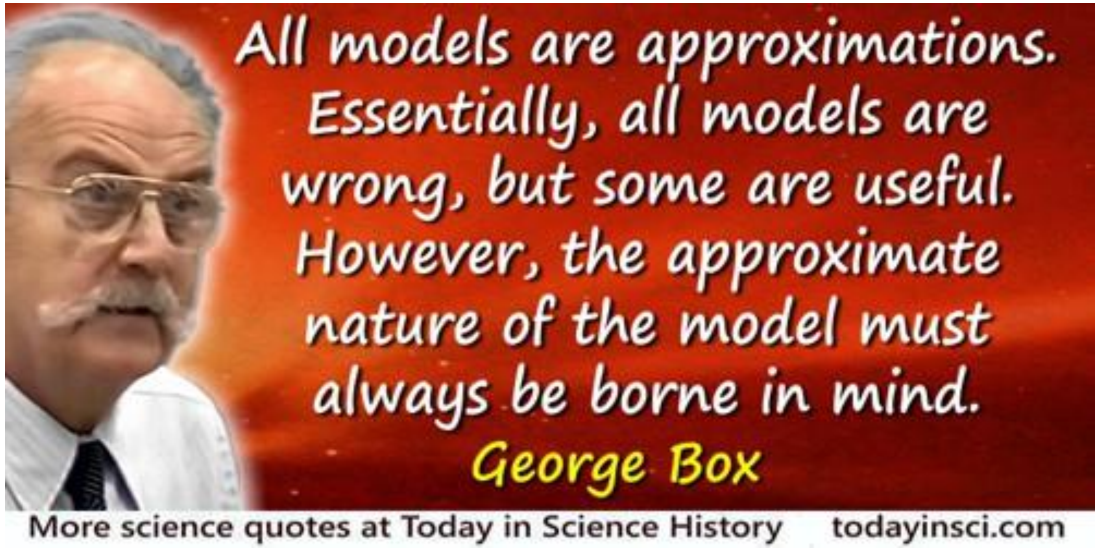
- The recipe:
  1. Model how the data is generated by probabilistic models, but with parameters unspecified (modeling assumption / generative story)
  2. (Training) Learn the model parameter $\hat{\theta}$
  3. (Test) Make prediction / decision based on the learned model $P(z; \hat{\theta})$



http://slideplayer.com/slide/4527958/

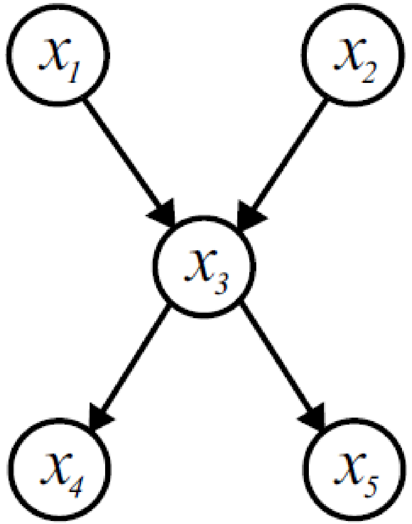
Probabilistic modeling (cont'd)

- Why probabilistic modeling?
  - Right thing to do if the model is correct
  - If not…
    - "All models are wrong, but some are useful" -- George Box
  - Interpretability
  - A view taken by classical statistics

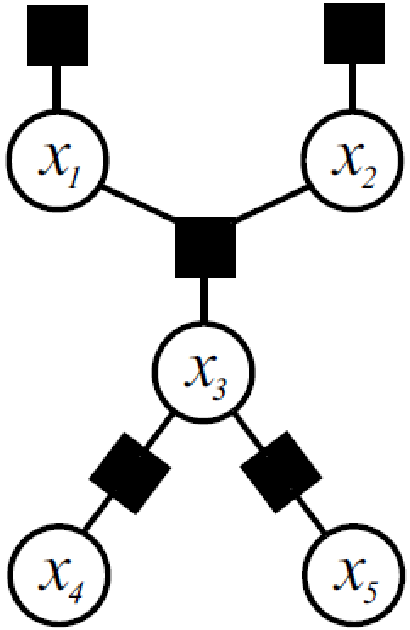


Graphical Models

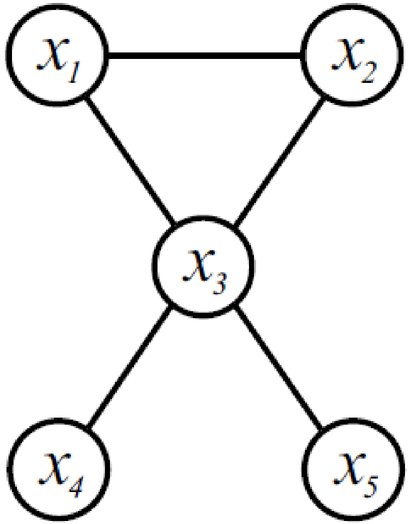
A variety of graphical models can represent the same probability distribution



Bayes Network



Factor Graph



Markov Random Field

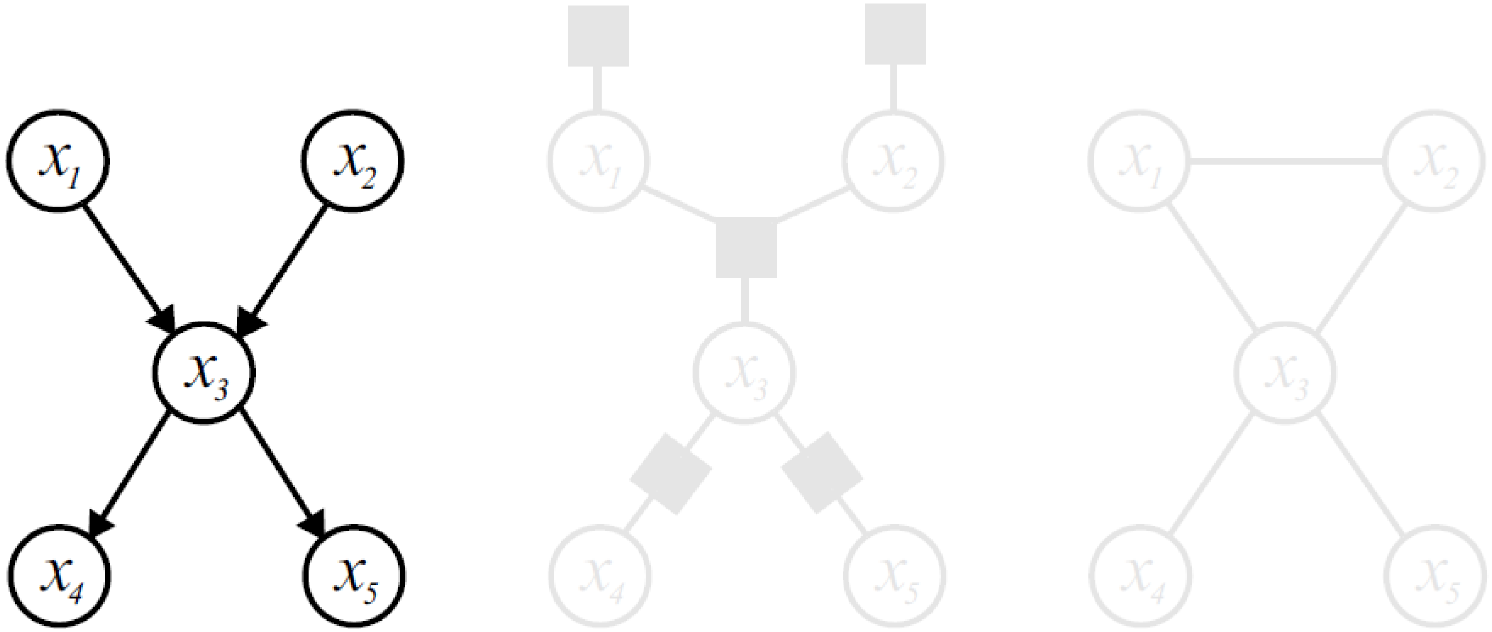
Directed Models

Undirected Models

[Source: Erik Sudderth, PhD Thesis]

Graphical Models

A variety of graphical models can represent the same probability distribution



Bayes Network | **Factor Graph** | **Markov Random Field**

Directed Models | **Undirected Models**

[Source: Erik Sudderth, PhD Thesis]

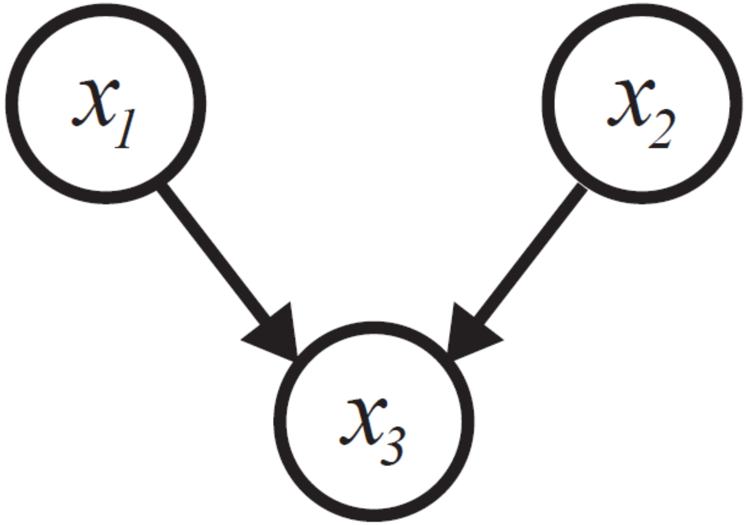
From Probabilities to Pictures

A probabilistic graphical model allows us to pictorially represent a probability distribution

Probability Model:

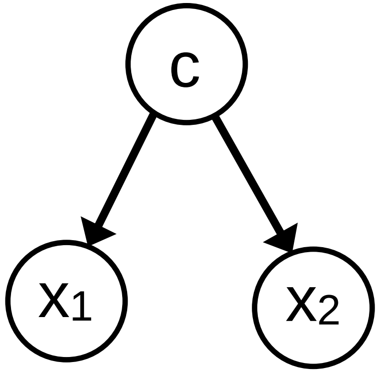
$$p(x_1, x_2, x_3) = p(x_1)p(x_2)p(x_3 \mid x_1, x_2)$$

Graphical Model:



- Conditional distribution of each RV is dependent on its parent nodes in the graph
- Intuition: arrows may encode causal relationship (e.g. x1=smoking, x2=exercise, x3=cancer)

Directed Graphical Models



Directed models are <u>generative models</u>…

…tells how data are generated (called **generative story; ancestral sampling**)

Step 1 Sample root node: $c \sim p(C)$

Step 2 Sample children, given sample of parent (likelihood):

$$x_1 \sim p(X_1 \mid C = c) \qquad x_2 \sim p(X_2 \mid C = c)$$

$$\Longrightarrow \; p(C, X_1, X_2) = p(C)p(X_1 \mid C)p(X_2 \mid C)$$

A graph induces an **<u>ordered factorization</u>** of the joint distribution

Probability Chain Rule

Recall the **probability chain rule** says that we can decompose any joint distribution as a product of conditionals….

$$p(x_1, x_2, x_3, x_4) = p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_1, x_2)p(x_4 \mid x_1, x_2, x_3)$$

Valid for *any ordering* of the random variables…

$$p(x_1, x_2, x_3, x_4) = p(x_3)p(x_1 \mid x_3)p(x_4 \mid x_1, x_3)p(x_2 \mid x_1, x_3, x_4)$$

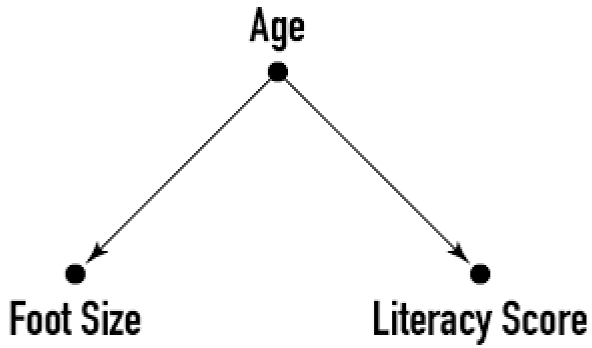
For a collection of N RVs and any permutation ρ :

$$p(x_1, \ldots, x_N) = p(x_{\rho(1)}) \prod_{i=2}^{N} p(x_{\rho(i)} \mid x_{\rho(i-1)}, \ldots, x_{\rho(1)})$$

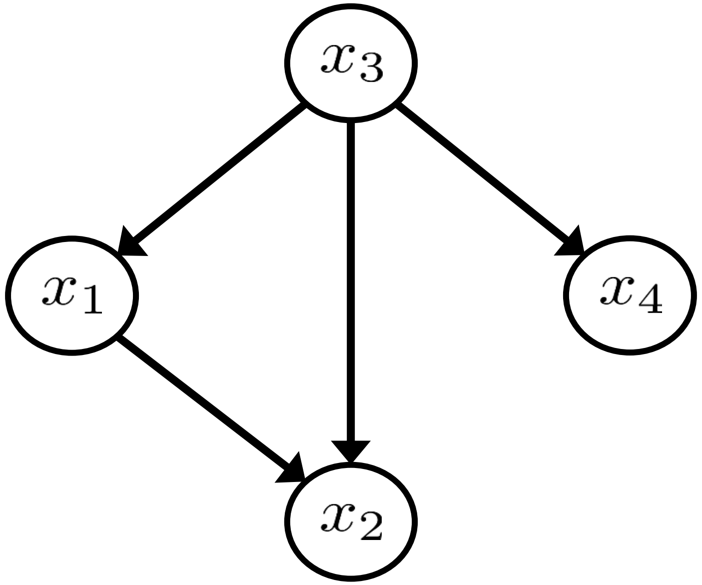
Conditional Independence

Recall two RVs X and Y are **conditionally independent** given Z (or $X \perp Y \mid Z$) iff:

$$p(X \mid Y, Z) = p(X \mid Z)$$



Idea Apply *chain rule* with ordering that <u>exploits conditional independencies</u> to simplify the terms



Ex. Suppose $x_4 \perp x_1 \mid x_3$ and $x_2 \perp x_4 \mid x_1, x_3$ then:

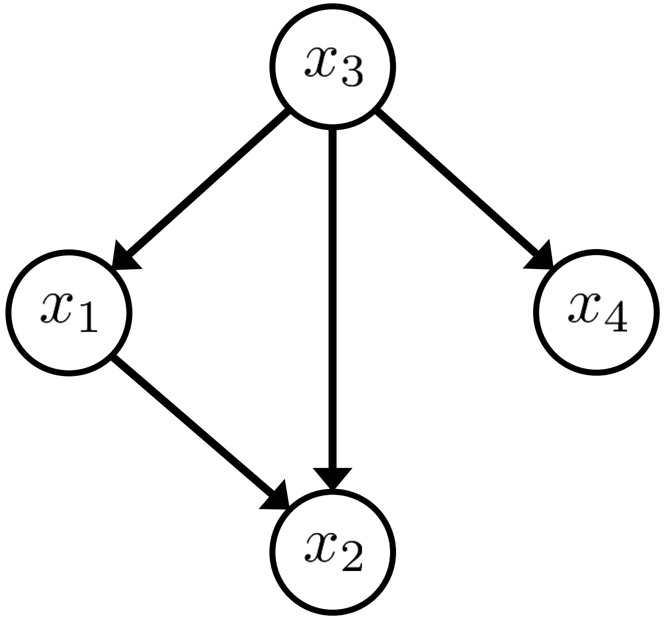
$$p(x) = p(x_3)p(x_1 \mid x_3)p(x_4 \mid x_1, x_3)p(x_2 \mid x_1, x_3, x_4)$$

$$= p(x_3)p(x_1 \mid x_3)p(x_4 \mid x_3)p(x_2 \mid x_1, x_3)$$

an <u>**ordered factorization**</u> of the joint distribution induces a directed acyclic graph (DAG)

General Directed Graphs

Def. A directed graph is a graph with edges $(s,t) \in \mathcal{E}$ (arcs) connecting parent vertex $s \in \mathcal{V}$ to a child vertex $t \in \mathcal{V}$



Def. Parents of vertex $t \in \mathcal{V}$ are given by the set of nodes with arcs pointing to t ,

$$\mathrm{Pa}(t) = \{s : (s,t) \in \mathcal{E}\}$$

Children of $t \in \mathcal{V}$ are given by the set,

$$\mathrm{Ch}(t) = \{t : (t,k) \in \mathcal{E}\}$$

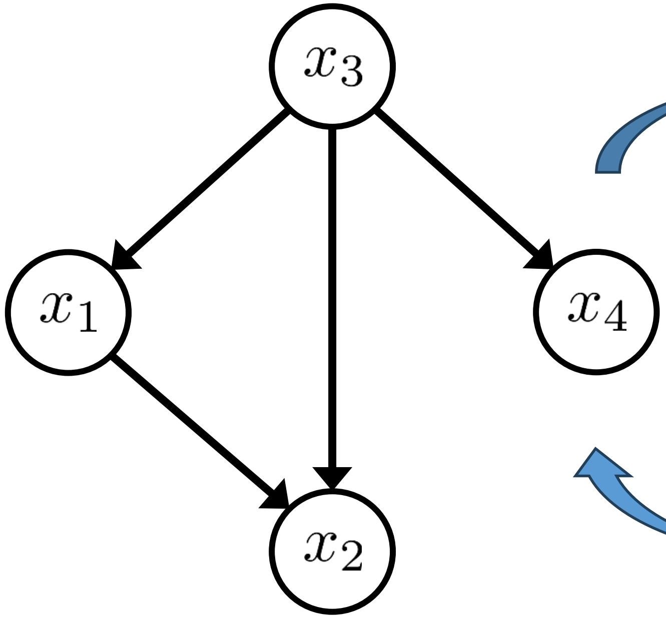
Ancestors are parents-of-parents.
Descendants are children-of-children.

Directed PGM = Bayes Network

Directed acyclic graph (DAG) $\Leftrightarrow$ factorized form of joint probability

$$p(x) = \prod_{s \in \mathcal{V}} p(x_s \mid x_{\mathrm{Pa}(s)})$$

Parents of node *s*



$$p(x) = p(x_3)p(x_1 \mid x_3)p(x_4 \mid x_3)p(x_2 \mid x_1, x_3)$$

Each factor has a node; conditioning relationship represented by edge

- Model factors are normalized conditional distributions
- *Locally normalized* factors yield *globally normalized* joint probability

Bayes network: A real-world example

- Joint distribution = graph structure + conditional probability table

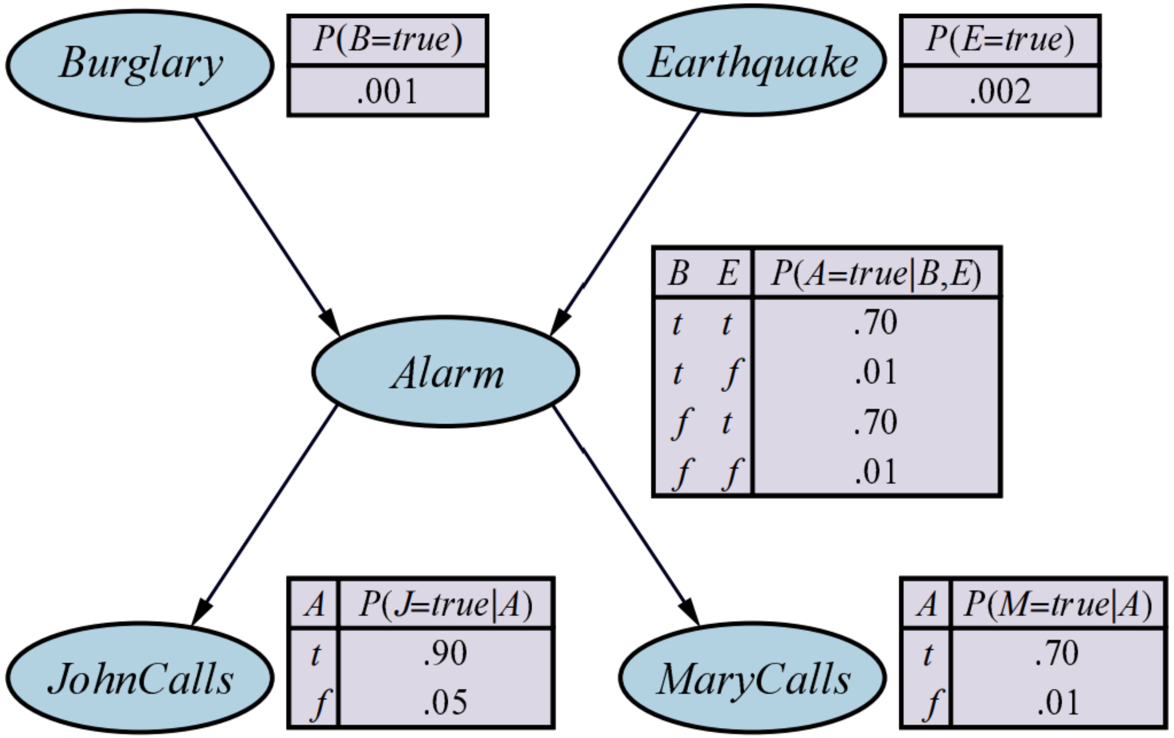
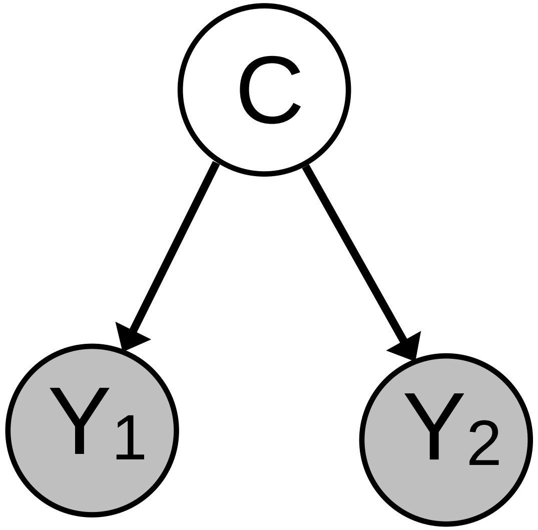


Figure 13.2 A typical Bayesian network, showing both the topology and the conditional probability tables (CPTs). In the CPTs, the letters B, E, A, J, and M stand for *Burglary*, *Earthquake*, *Alarm*, *JohnCalls*, and *MaryCalls*, respectively.

Inference



Denote observed data with shaded nodes,

$$Y_1 = y_1 \qquad Y_2 = y_2$$

e.g. C = flu, Y1 = fever, Y2 = cough, y1=y2=True

Infer *latent* variable C via Bayes' rule:

$$p(c \mid y_1, y_2) = \frac{p(c)p(y_1 \mid c)p(y_2 \mid c)}{p(y_1, y_2)}$$

- This is (obviously) a simple example
- Models and inference task can get really complicated
- But the fundamental concepts and approach are the same

Bayes' Rule

Posterior represents all uncertainty after observing data…

prior probability

likelihood function for the parameters

$$p(c \mid y) = \frac{p(c)p(y \mid c)}{p(y)}$$

posterior probability

marginal likelihood

or: evidence

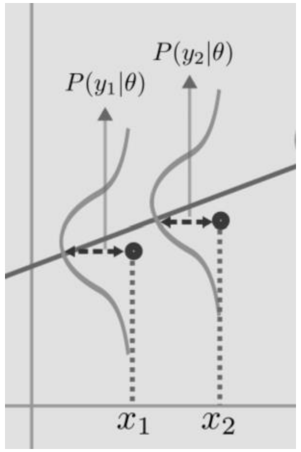
or: partition function

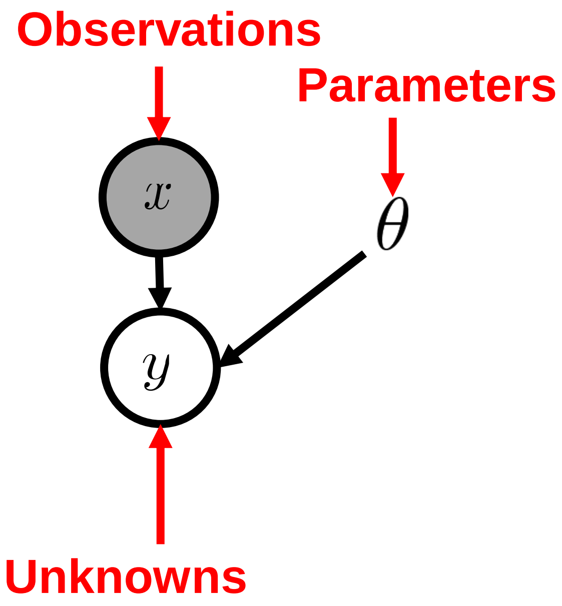
or: normalizer

Discriminative vs Generative modeling

Discriminative model:

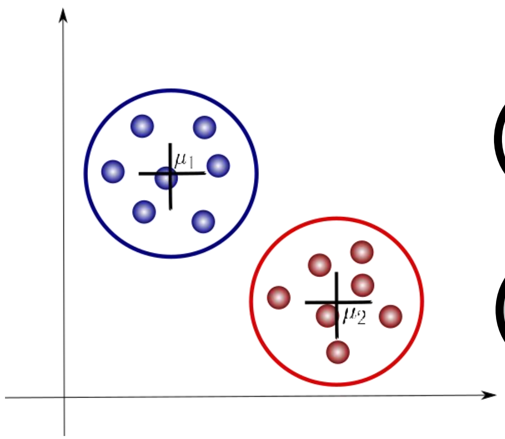
- Only models $P(\, y \mid x, \theta \,)$ -- i.e. *doesn't model data x*
- Recall linear regression: $y \mid x; \theta \sim N(x^\top \theta, \sigma^2)$
- Logistic regression: $y \mid x; \theta \sim \text{Bernoulli}(\sigma(x^\top \theta))$

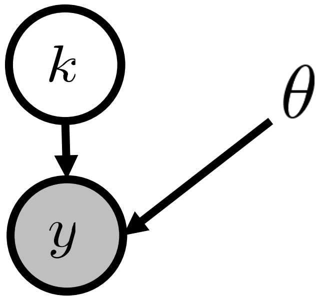




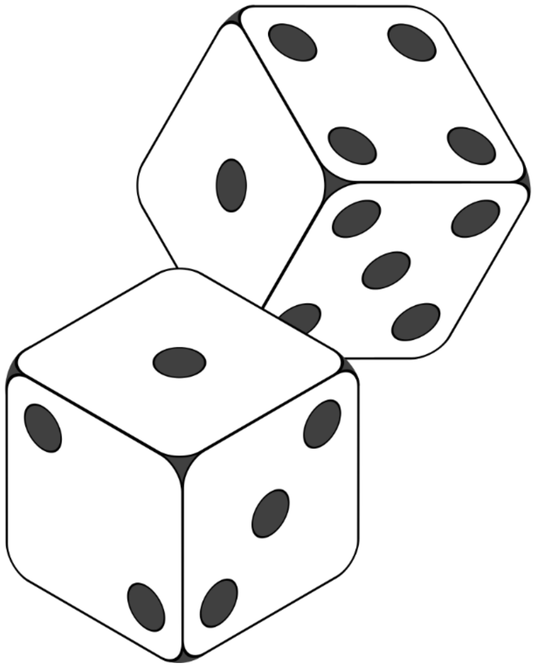
Generative model:

- Models everything including data: $P(k, y) = P(k)P(y \mid k, \theta)$
- e.g., Gaussian mixture model (GMM)
  - $\theta = (\pi_k, \mu_k, \Sigma_k)_{k=1}^{K}$
  - $k \sim \text{Categorical}(\pi)$ (*hidden*), i.e. $P(k = l) = \pi_l$
  - $y \mid k \sim N(\mu_k, \Sigma_k)$





(Aside) Categorical Distribution



Distribution on integer-valued RV $X \in \{1, \dots, K\}$

$$p(X) = \prod_{k=1}^{K} \pi_k^{\mathbf{I}(X=k)} \quad \text{or} \quad p(X) = \sum_{k=1}^{K} \mathbf{I}(X=k) \cdot \pi_k$$

with parameter $p(X=k) = \pi_k$ *and Kroenecker delta:*

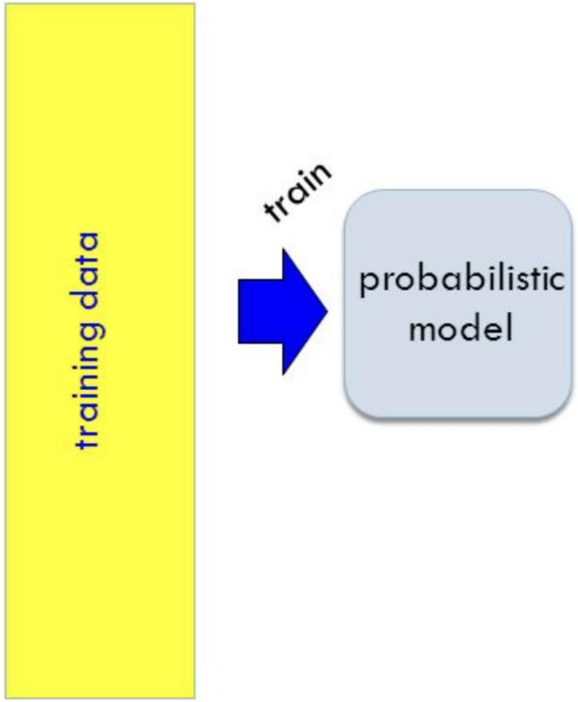
$$\mathbf{I}(X=k) = \begin{cases} 1, & \text{If } X = k \\ 0, & \text{Otherwise} \end{cases}$$

Can also represent X as *one-hot* binary vector,

$$X \in \{0,1\}^K \quad \text{where} \quad \sum_{k=1}^{K} X_k = 1 \quad \text{then} \quad p(X) = \prod_{k=1}^{K} \pi_k^{X_k}$$

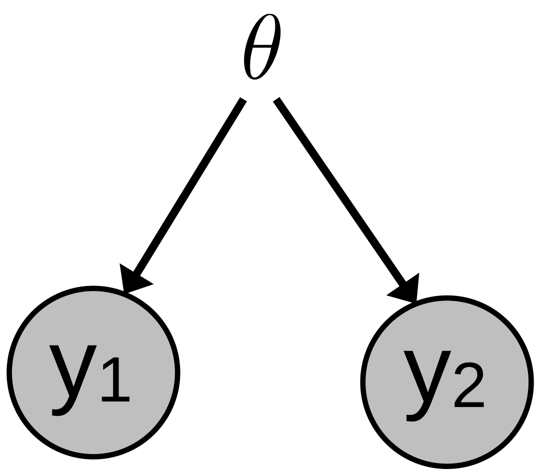
Probabilistic modeling: systematic approach for ML

- The recipe:
  1. Model how the data is generated by probabilistic models, but with parameters unspecified (modeling assumption / generative story)
  2. (Training) Learn the model parameter $\hat{\theta}$
  3. (Test) Make prediction / decision based on the learned model $P(z; \hat{\theta})$



http://slideplayer.com/slide/4527958/

Learning / Training



Model random data with hyperparameters θ :

$$y \sim p(y \mid \theta)$$

Sometimes we use:
$p(y; \theta)$

Given training data:

$$\{y_i\}_{i=1}^{n} \overset{\text{i.i.d.}}{\sim} p(y \mid \theta)$$

Learn parameters, e.g. via *maximum likelihood estimation*:

$$\hat{\theta}^{\text{MLE}} = \arg\max_{\theta} \log p(y_1, \ldots, y_n \mid \theta)$$

We will talk more about MLE in coming weeks

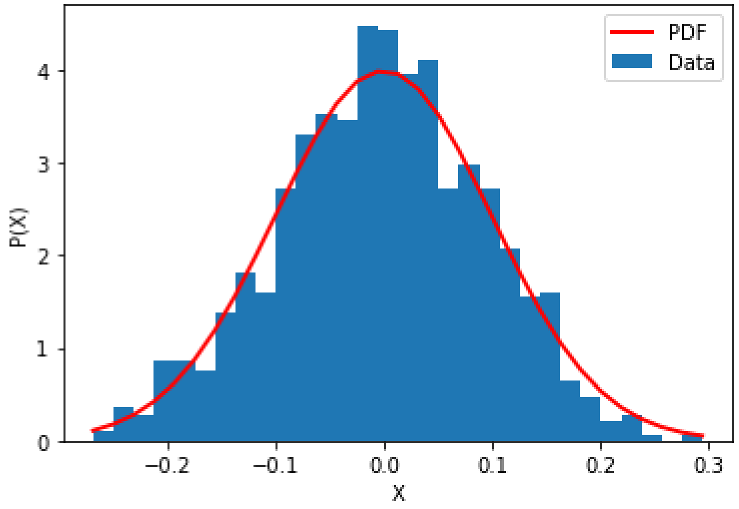
Other estimators are possible:

- *Maximum a posteriori* (MAP)
- *Minimum mean squared error (MMSE)*
- *Etc.*

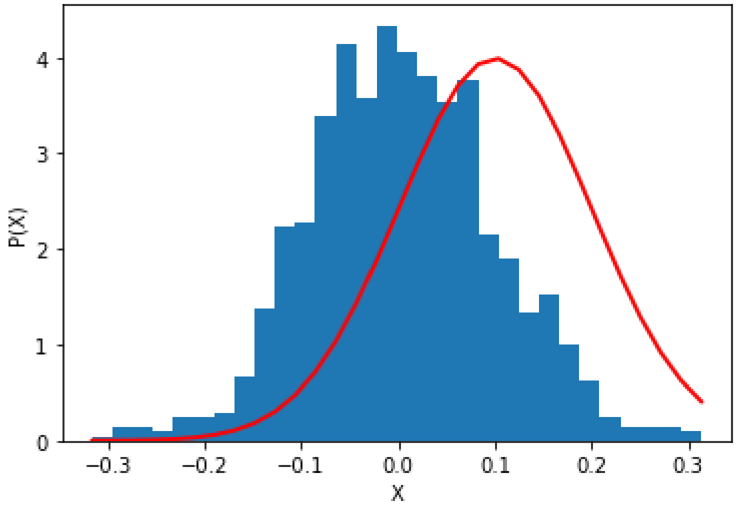
Likelihood (Intuitively)

Suppose we observe N data points from a Gaussian model and wish to estimate model parameters…

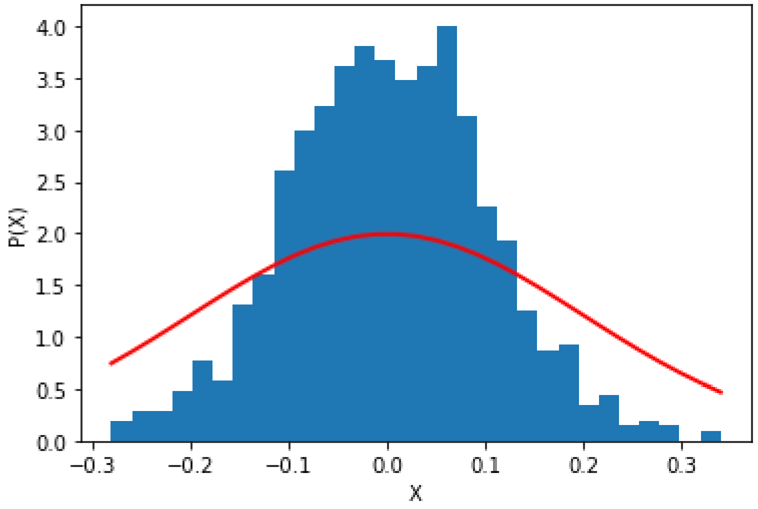
High Likelihood



Low Likelihood (mean)



Low Likelihood (variance)



Likelihood Principle *Given a statistical model, the likelihood function describes all evidence of a parameter that is contained in the data.*

Likelihood Function

Suppose $x_i \sim p(x;\theta)$, then what is the **joint probability** over N *independent identically distributed* (iid) observations $x_1, \ldots, x_N$?

$$p(x_1, \ldots, x_N; \theta) = \prod_{i=1}^{N} p(x_i; \theta)$$

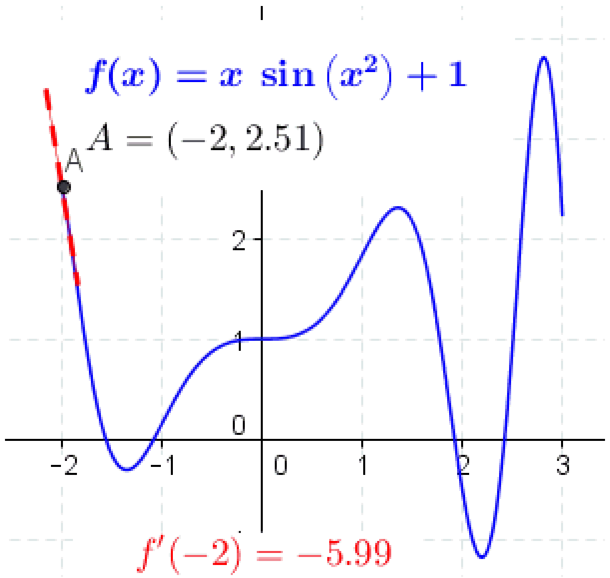
- We call this the **likelihood function**, often denoted $\mathcal{L}_N(\theta)$
- It is a function of the parameter θ, the data are fixed
- Measures how well parameter θ describes data (*goodness of fit*)

How could we use this to estimate a parameter θ*?*

Maximum Likelihood

Maximum Likelihood Estimator (MLE) as the name suggests, maximizes the likelihood function.

$$\hat{\theta}^{\text{MLE}} = \arg\max_{\theta} \mathcal{L}_N(\theta) = \prod_{i=1}^{N} p(x_i;\theta)$$



Question How do we find the MLE?

Answer Remember calculus… to maximize $f(\theta)$:

Is $f(\theta)$ concave?

Yes → Unique solution →

Approach

- Compute derivative $\frac{df}{d\theta}$
- Set to zero and solve

$$\frac{df}{d\theta} = 0 \Rightarrow \hat{\theta}^{\text{MLE}}$$

No → Gradient-based optimization → Still have to compute derivative…

Maximum Likelihood

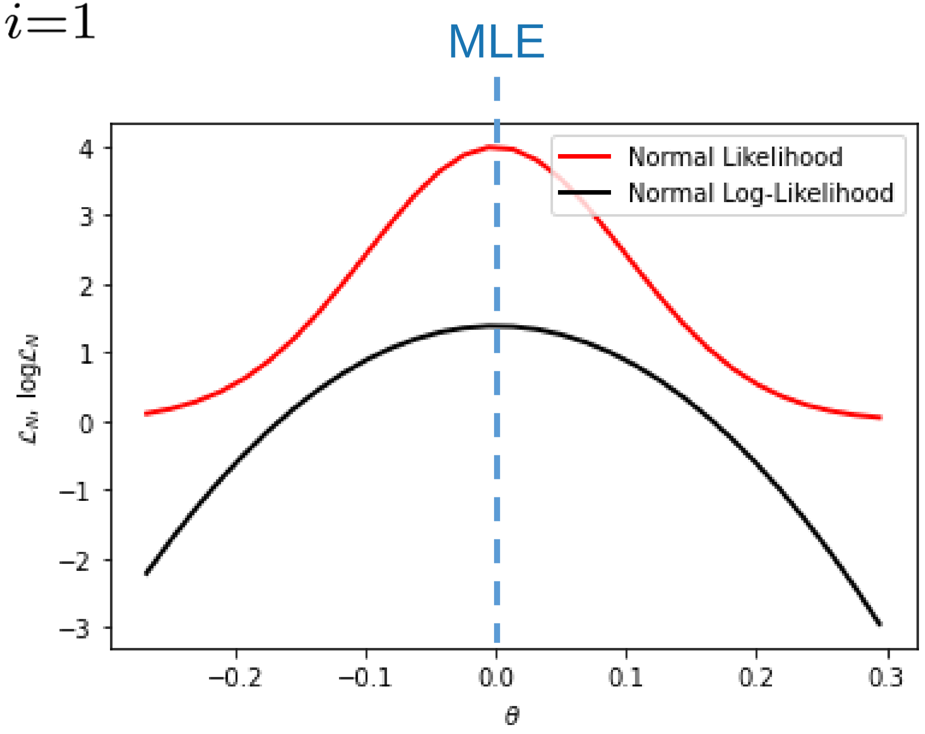
Maximizing log-likelihood makes the math easier (as we will see) and doesn't change the answer (logarithm is an increasing function)

$$\hat{\theta}^{\mathrm{MLE}} = \arg\max_{\theta} \; \log \mathcal{L}_N(\theta) = \sum_{i=1}^{N} \log p(x_i;\theta)$$

Derivative is a linear operator so,

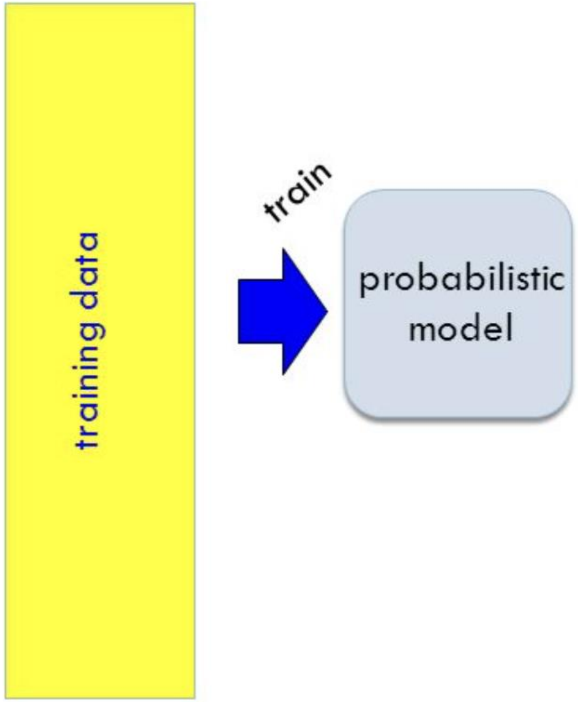
$$\frac{d}{d\theta}\log \mathcal{L}_N(\theta) = \underbrace{\sum_{i=1}^{N} \frac{d}{d\theta} \log p(x_i;\theta)}_{\text{One term per data point; Can be computed in parallel (big data)}}$$

One term per data point
Can be computed in parallel
(big data)



Probabilistic modeling: systematic approach for ML

- The recipe:
    1. Model how the data is generated by probabilistic models, but with parameters unspecified (modeling assumption / generative story)
    2. (Training) Learn the model parameter $\hat{\theta}$
    3. (Test) Make prediction / decision based on the learned model $P(z; \hat{\theta})$

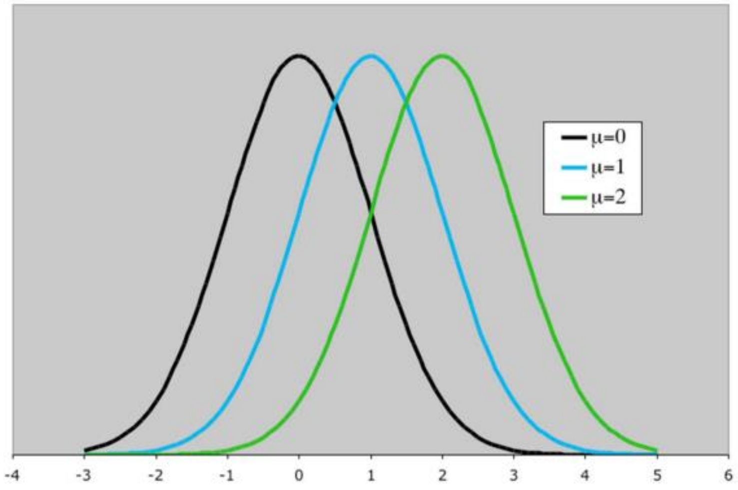


http://slideplayer.com/slide/4527958/

Example: Barbershop

Suppose you go to a barbershop at every last Friday of the month. You want to be able to predict the waiting time. You have collected 12 data points (i.e., how long it took to be served) from the last year: $S = \{x_1, \dots, x_{12}\}$

- 1. Modeling assumption: $x_i \sim$ Gaussian distribution $N(\mu, 1)$
  - $p(x; \mu) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2}\right)$
  - Observation: this distribution has mean μ

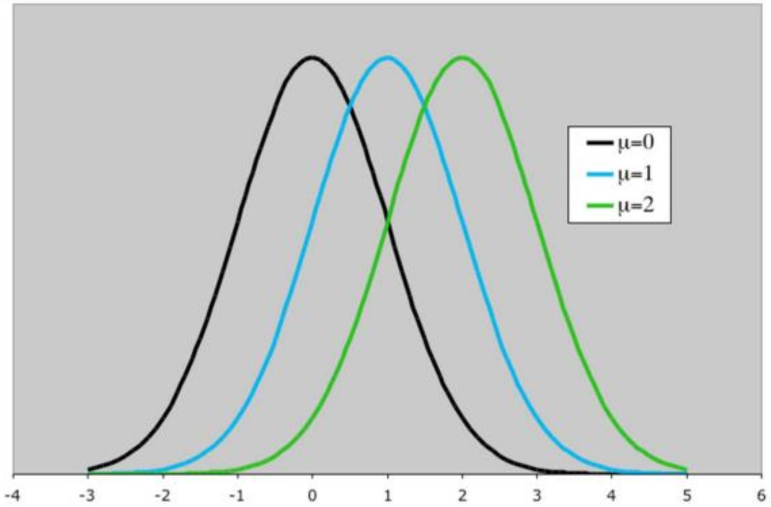


Is this a <u>generative</u> or <u>discriminative</u> model?

- 2. Training: find the MLE $\hat{\mu}$ from data S
  - (2.1) write down the neg. log likelihood of the sample

$$L_n(\mu) = -\ln P(x_1, \dots, x_n; \mu) = 12 \ln\sqrt{2\pi} + \frac{1}{2}\sum_{i=1}^{12}(x_i - \mu)^2$$

Generative model example: barbershop (cont'd)



2. Find the MLE $\hat{\mu}$ from data S

- (2.2) compute the first derivative, set it to 0, solve for λ (be sure to check convexity)

  $L_n'(\mu) = \sum_{i=1}^{12}(x_i - \mu) = 0 \Rightarrow \mu = \frac{x_1 + \cdots x_{12}}{12}$ Sample Mean

3. The learned model $N(\hat{\mu}, 1)$ is yours!

- Simple prediction: e.g., predict the next wait time by $\mathbb{E}_{X \sim N(\hat{\mu},1)}[X]$
- which is $\hat{\mu} = \frac{x_1 + \cdots x_{12}}{12}$

4. (Optional: Model Checking) Generate some data... Does it look realistic?

Basic Example II: balls from a bin

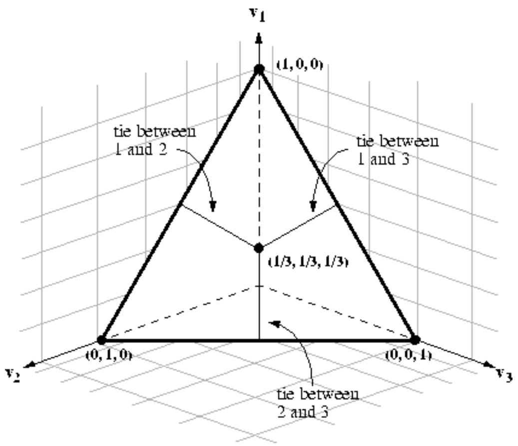
Data $S = \{y_i\}_{i=1}^{n}$, where $y_i \in \{1, \dots, C\}$

1. Generative Story

$y \sim \mathrm{Categorical}(\pi)$, where $\pi = (\pi_1, \dots, \pi_C) \in \Delta^{C-1}$ ($\pi_c \geq 0$ and $\pi_1 + \cdots + \pi_C = 1$)

e.g. y_i = the color of i-th ball drawn randomly from a bin (with replacement)

$$p(y; \pi) = \pi_y \left(= \prod_{c=1}^{C} \pi_c^{I(y=c)} \right)$$



2. Training

(2.1) $L_n(\pi) = -\ln P(y_1, \dots, y_n; \pi) = \sum_{i=1}^{n} -\ln \pi_{y_i} = -\sum_{c=1}^{C} n_c \ln \pi_c$,

where $n_c = \#\{i : y_i = c\} = \sum_{i=1}^{n} I(y_i = c)$

Basic Example II (Cont'd)

2. Training

(2.2) $\text{minimize}_{\pi\in\Delta^{C-1}} L_n(\pi) := -\sum_{c=1}^{C} n_c \ln \pi_c$

Constrained maximization problem; solve by Lagrange multipliers

$$\frac{\partial}{\partial\pi}\left(-\sum_{c=1}^{C} n_c \ln \pi_c - \lambda\left(\sum_{c=1}^{C}\pi_c - 1\right)\right) = -\frac{n_c}{\pi_c} - \lambda = 0 \Rightarrow \pi_c = -\frac{n_c}{\lambda}$$

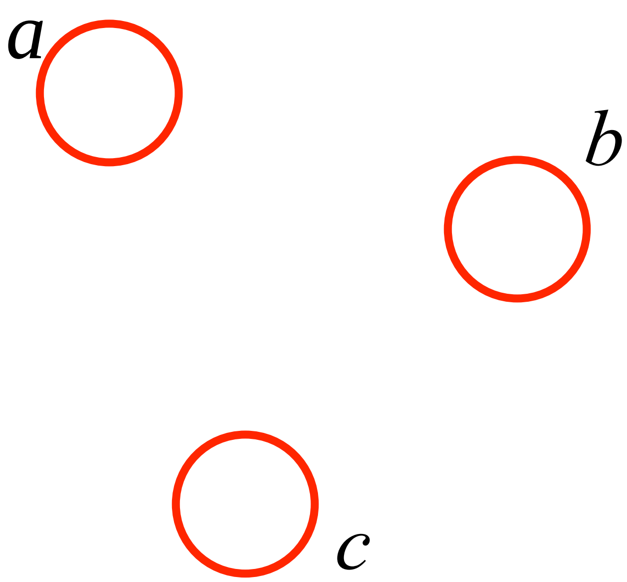
Combined with the constraint that $\pi_1 + \cdots + \pi_C = 1 \Rightarrow \hat{\pi}_c = \frac{n_c}{n}$, for all c

3. Test predict label $\text{argmax}_c P(y = c; \hat{\pi}) = \text{argmax}_c \hat{\pi}_c$

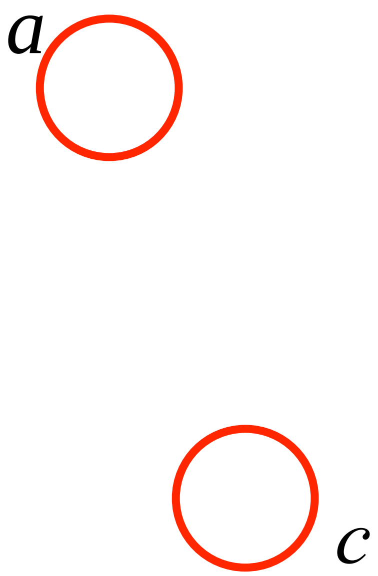
Outline

- Probabilistic Graphical Models
- Case study: Naïve Bayes

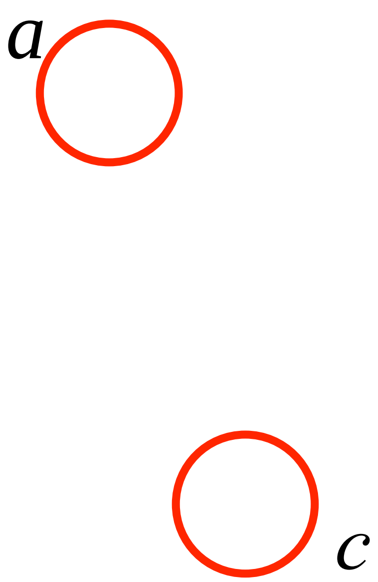
What is the joint factorization?



$$\mathbf{p(a,b,c) = p(a)p(b)p(c)}$$

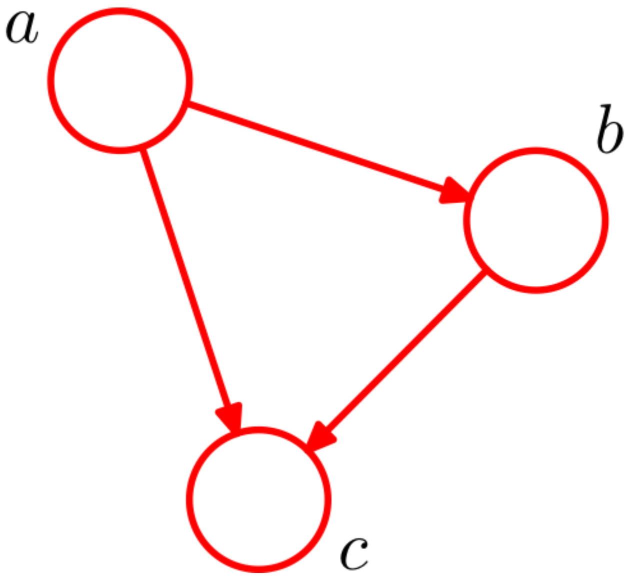


Are a and b independent ($a \perp b$)?



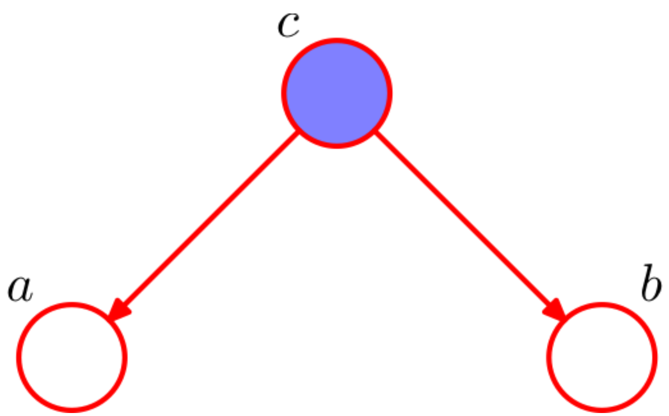
$$\mathbf{p(a,b,c) = p(a)p(b)p(c)}$$

$p(a,b,c) = p(a)p(b|a)p(c|a,b)$



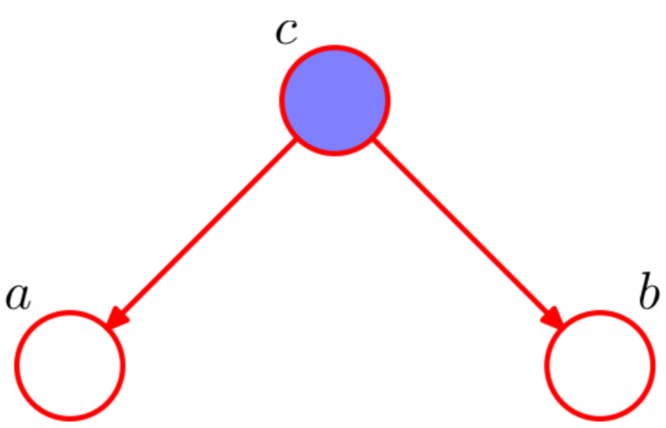
Note there are **no conditional independencies**

Case one where c is observed



Is $a \perp b \mid c$?

Case one where c is observed

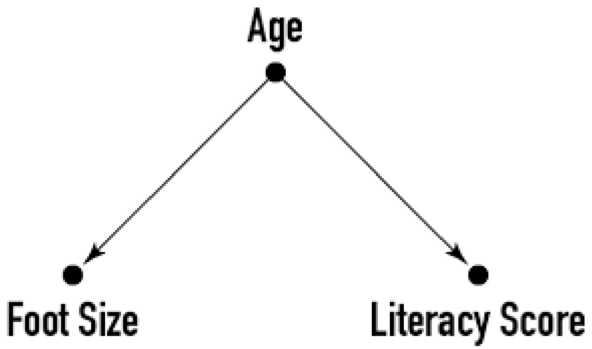


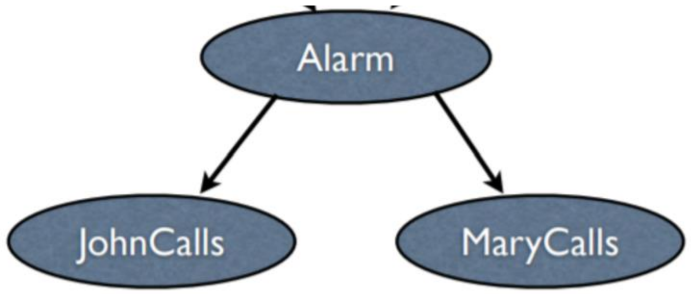
$$a \perp b \mid c$$

$$p(a,b,c) = p(c)p(a|c)p(b|c) \quad \text{(what the graph represents in general)}$$

$$p(a,b|c) = p(a|c)p(b|c) \quad \text{(with } c \text{ observed)}$$

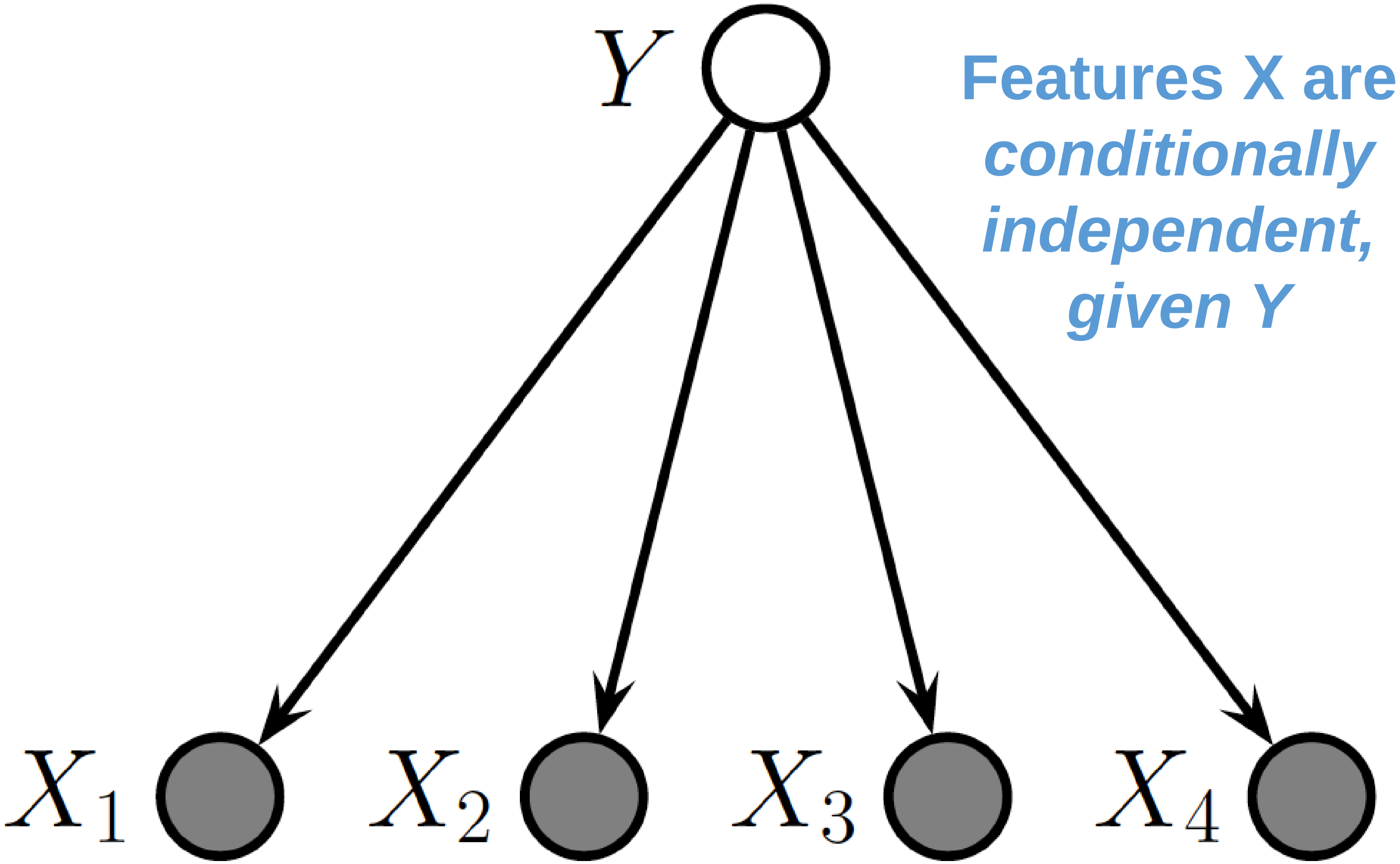
This is the definition of $a \perp b | c$





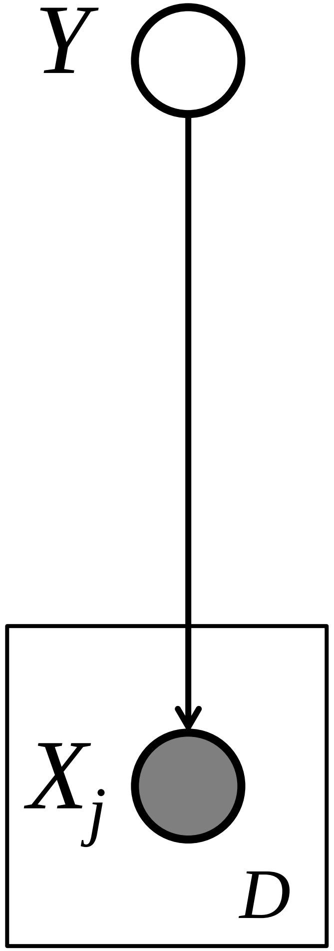
Shading & Plate Notation

Convention: *Shaded nodes are observed, open nodes are latent/hidden/unobserved*



Features X are *conditionally independent, given Y*

$$p(y, \mathbf{x}) = p(y) \prod_{j=1}^{D} p(x_j | y)$$



Plates *denote replication of random variables*

Naïve Bayes for supervised learning

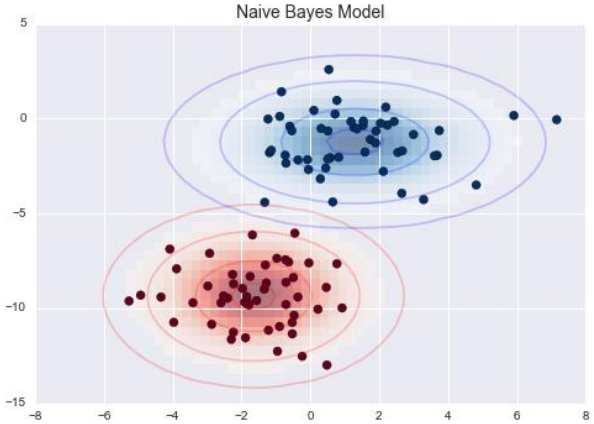
- Motivation: supervised learning for classification
- high-dimensional $x = (x(1), \dots, x(F))$, modeling $P(x \mid y)$ can be tricky
- In general, $P(\,x \mid y\,) = P(\,x(1) \mid y\,) \cdot P(\,x(2) \mid x(1), y\,) \cdot \dots \cdot P(x(F) \mid x(1), \dots, x(F-1), y)$

- A modeling assumption: $x(1), \dots, x(F)$ are conditionally independent given y

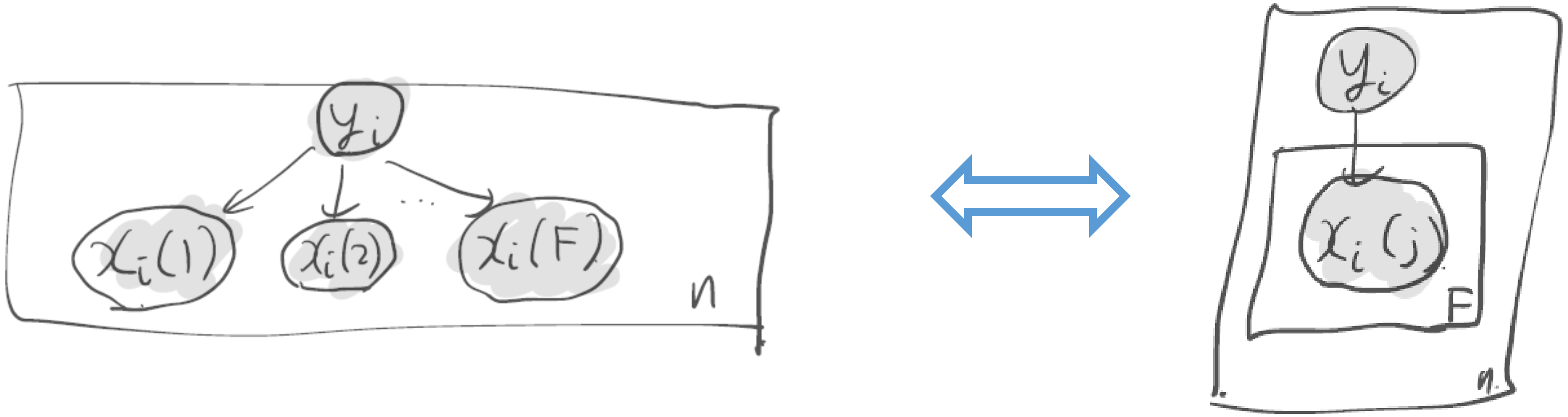
  i.e. for all i

$$x(i) \perp\!\!\!\perp \big(x(1), \dots, x(i-1), x(i+1), \dots, x(F)\big) \mid y$$

  (Conditional independence notation: $A \perp\!\!\!\perp B \mid C$)



- Equivalently $P(\,x \mid y\,) = P(\,x(1) \mid y\,) \cdot \dots P(x(F) \mid y)$



Example: Class Preference Prediction

Define the labeled training dataset $S = \{(x_i, y_i)\}_{i=1}^{m}$

To make this a <u>binary</u> classification we set
"Like" = {+2,+1,0}
"Not Like" = {-1,-2}

Features

Feature Values

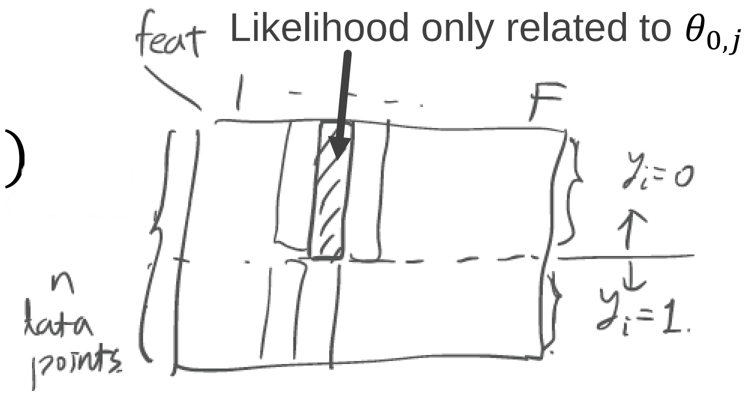
Labels

Data Point

Rating	Easy?	AI?	Sys?	Thy?	Morning?
+2	y	y	n	y	n
+2	y	y	n	y	n
+2	n	y	n	n	n
+2	n	n	n	y	n
+2	n	y	y	n	y
+1	y	y	n	n	n
+1	y	y	n	y	n
+1	n	y	n	y	n
0	n	n	n	n	y
0	y	n	n	y	y
0	n	y	n	y	n
0	y	y	y	y	y
-1	y	y	y	n	y
-1	n	n	y	y	n
-1	n	n	y	n	y
-1	y	n	y	n	y
-2	n	n	y	y	n
-2	n	y	y	n	y
-2	y	n	y	n	n
-2	y	n	y	n	y

Naïve Bayes: binary-valued features

Training Data $S = \{(x_i, y_i)\}_{i=1}^{n}$, $x_i \in \{0,1\}^F$ $y_i \in \{0,1\}$



Generative Story

$y \sim \text{Bernoulli}(\pi)$; for all $j \in [F]$, $x(j) \mid y = c \sim \text{Bernoulli}(\theta_{c,j})$

#parameters = $1 + 2F$

Training (denote by $\theta = \{\theta_{c,j}\}$)

$$\max_{\pi,\theta} \sum_{i=1}^{n} \ln P(x_i, y_i; \pi, \theta)$$

$$= \max_{\pi,\theta} \sum_{i=1}^{n} \ln P(y_i; \pi) + \sum_{i=1}^{n} \ln P(x_i \mid y_i; \theta)$$

$$= \max_{\pi,\theta} \underbrace{\sum_{i=1}^{n} \ln P(y_i; \pi)}_{\text{Only related to } \pi} + \underbrace{\sum_{i:y_i=0} \ln P(x_i \mid y_i; \theta)}_{\text{Only related to } \theta_{0j}\text{'s}} + \underbrace{\sum_{i:y_i=1} \ln P(x_i \mid y_i; \theta)}_{\text{Only related to } \theta_{1j}\text{'s}}$$

Only related to π **Only related to θ_{0j}'s** **Only related to θ_{1j}'s**

=> The maximizing $\pi, \{\theta_{0j}\}, \{\theta_{1j}\}$ can be obtained separately!

Naïve Bayes: binary-valued features

Training Data $S = \{(x_i, y_i)\}_{i=1}^{n}$, $x_i \in \{0,1\}^F$ $y_i \in \{0,1\}$

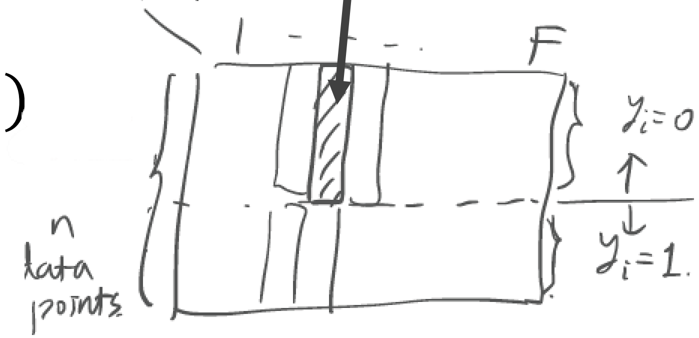
Generative Story

$y \sim \text{Bernoulli}(\pi)$; for all $j \in [F]$, $x(j) \mid y = c \sim \text{Bernoulli}(\theta_{c,j})$

#parameters = $1 + 2F$

Training

Likelihood only related to $\theta_{0,j}$



Optimal π: $\max_\pi \sum_{i=1}^{n} \ln P(y_i; \pi) = \max_\pi n_0 \ln(1-\pi) + n_1 \ln(\pi)$ => $\hat{\pi} = \frac{n_1}{n}$

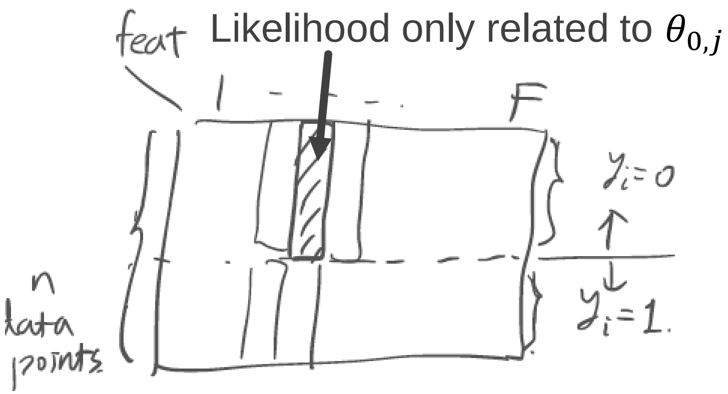
How about optimal $\{\theta_{0j}\}$, $\{\theta_{1j}\}$?

Naïve Bayes: binary-valued features (cont'd)

By the Naïve Bayes modeling assumption,

$$\max_{\{\theta_{0,j}\}} \sum_{i:y_i=0} \ln P(x_i \mid y_i; \theta) = \max_{\{\theta_{0,j}\}} \sum_{j=1}^{F} \boxed{\sum_{i:y_i=0} \ln P\big(x_i(j) \mid y_i; \theta_{0,j}\big)}$$

Only related to θ_{0j}



Again, can optimize each $\theta_{0,j}$ separately,

$$\underset{\theta_{0,j}}{\operatorname{argmax}} \sum_{i:y_i=0,\, x_i(j)=1} \ln \theta_{0,j} + \sum_{i:y_i=0,\, x_i(j)=0} \ln\,(1-\theta_{0,j})$$

- Solution: $\hat{\theta}_{0,j} = \frac{\#\{i:\, y_i=0,\, x_i(j)=1\}}{\#\{i: y_i=0\}}, j = 1, \dots, F$
- Similarly, $\hat{\theta}_{1,j} = \frac{\#\{i:\, y_i=1,\, x_i(j)=1\}}{\#\{i: y_i=1\}}, j = 1, \dots, F$

Naïve Bayes: binary-valued features (cont'd)

Test Given $\hat{\pi}$, $\{\hat{\theta}_{c,j}\}$, Bayes optimal classifier

$$\hat{f}_{BO}(x) = \text{argmax}_y\, P(x, y; \hat{\pi}, \{\hat{\theta}_{c,j}\}) = \text{argmax}_y \log P(x, y; \hat{\pi}, \{\hat{\theta}_{c,j}\})$$

- $\log P(x, y = 0; \pi, \{\theta_{c,j}\}) = \ln(1-\pi) + \sum_{j=1}^{F} \ln P\big(x(j) \mid y; \theta_{0,j}\big)$

$$= \ln(1-\pi) + \sum_{j=1}^{F} \ln\big(1-\theta_{0,j}\big) I(x(j) = 0) + \ln\big(\theta_{0,j}\big) I(x(j) = 1)$$

$$= \ln(1-\pi) + \sum_{j=1}^{F} \ln(1-\theta_{0,j}) + \sum_{j=1}^{F} x(j) \ln \frac{\theta_{0,j}}{1-\theta_{0,j}}$$

- Similarly, $\log P(x, y = 1; \pi, \{\theta_{c,j}\}) = \ln(\pi) + \sum_{j=1}^{F} \ln(1-\theta_{1,j}) + \sum_{j=1}^{F} x(j) \ln \frac{\theta_{1,j}}{1-\theta_{1,j}}$

Naïve Bayes: binary-valued features (cont'd)

Test Given $\hat{\pi}$, $\{\hat{\theta}_{c,j}\}$, Bayes optimal classifier

$$\hat{f}_{BO}(x) = \text{argmax}_y \, P(x, y; \hat{\pi}, \{\hat{\theta}_{c,j}\}) = \text{argmax}_y \log P(x, y; \hat{\pi}, \{\hat{\theta}_{c,j}\})$$

- Therefore, $\hat{f}_{BO}(x) = 1$

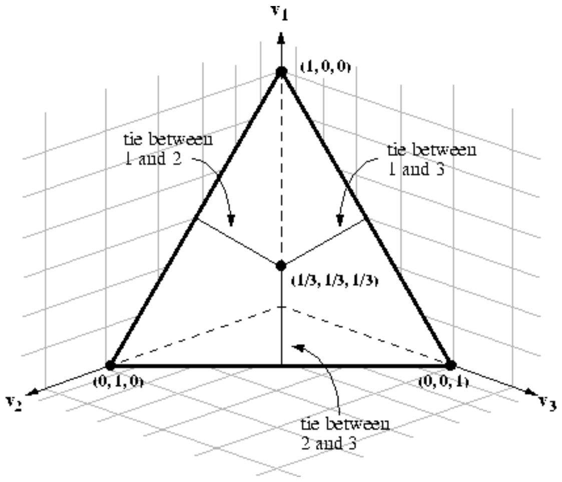
$$\Leftrightarrow \log P(x, y = 1; \hat{\pi}, \{\hat{\theta}_{c,j}\}) \geq \log P(x, y = 0; \hat{\pi}, \{\hat{\theta}_{c,j}\})$$

$$\Leftrightarrow \underbrace{\ln\left(\frac{\hat{\pi}}{1-\hat{\pi}}\right) + \sum_{j=1}^{F} \ln\left(\frac{1-\hat{\boldsymbol{\theta}}_{1,j}}{1-\hat{\theta}_{0,j}}\right)}_{b} + \sum_{j=1}^{F} x(j) \underbrace{\left(\ln\frac{\hat{\theta}_{1,j}}{1-\hat{\theta}_{1,j}} - \ln\frac{\hat{\theta}_{0,j}}{1-\hat{\theta}_{0,j}}\right)}_{w(j)} \geq 0$$

- Therefore, in this setting, Bayes classifier is *linear*

Naïve Bayes: Discrete (Categorical-valued) features

Data $S = \{(x_i, y_i)\}_{i=1}^n$, $x_i \in [W]^F$ $y_i \in \{0,1\}$



Generative story

$y \sim \mathrm{Bernoulli}(\pi)$; for all $j \in [F]$, $x(j) \mid y = c \sim \mathrm{Categorical}(\theta_c)$ $(\theta_c \in \Delta^{W-1})$

#parameters = $1 + 2W$

Note: in this example, θ_c *shared across all features*!

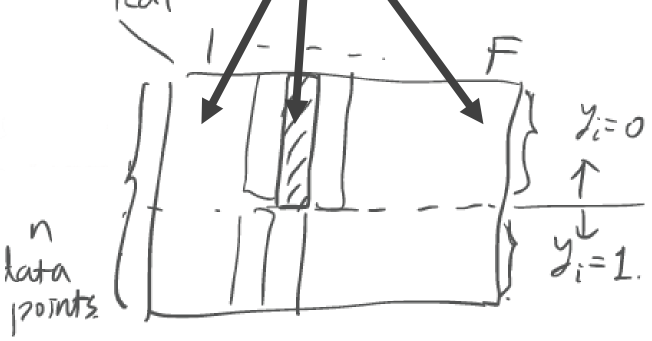
Training

Similar to previous example, optimal π, optimal θ_0, optimal θ_1 can be found separately,

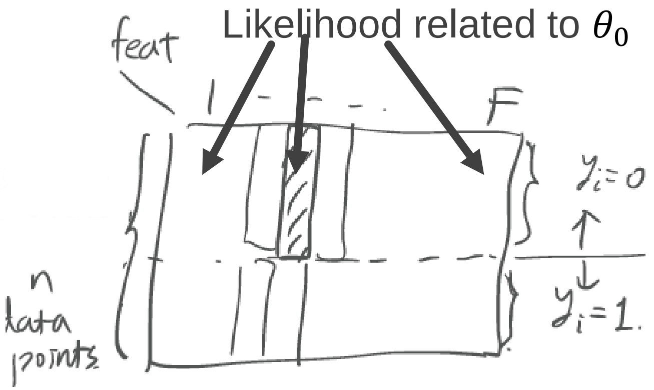
by maximizing the respective part of the likelihood function (exercise)

Optimal π same as previous example

Likelihood related to θ_0



Naïve Bayes: Discrete features (cont'd)



Training

Optimal θ_c:

$$\max_{\theta_0} \sum_{i:y_i=0} \ln P(x_i \mid y_i; \theta_0) = \max_{\theta_0} \sum_{j=1}^{F} \sum_{i:y_i=0} \ln P(x_i(j) \mid y_i; \theta_0)$$

$$= \max_{\theta_0} \sum_{w=1}^{W} \sum_{j=1}^{F} \sum_{i:y_i=0} I(x_i(j) = w) \ln \theta_{0,w}$$

$$= \max_{\theta_0} \sum_{w=1}^{W} \ln \theta_{0,w} \, \#\{(i,j): y_i = 0, x_i(j) = w\}$$

$$\Rightarrow \hat{\theta}_{c,w} = \frac{\#\{(i,j): y_i = c, x_i(j) = w\}}{\#\{i: y_i = c\} \times F}$$

Exercise: how to extend this to variable-length x_i's (e.g. for text classification)?

Test

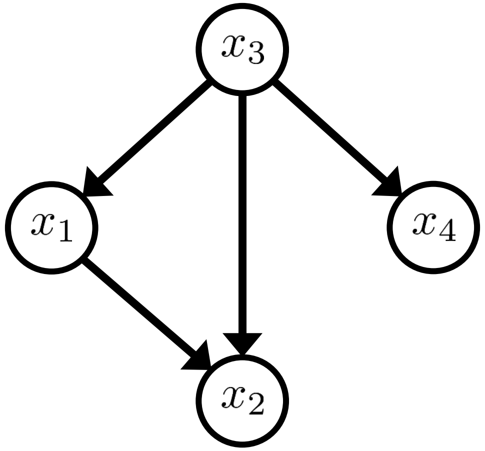
Bayes optimal classification rule with $(\hat{\pi}, \hat{\theta}_0, \hat{\theta}_1)$ (exercise)

Summary

- Probabilistic machine learning recipe
  - Step 1. Modeling
  - Step 2. Training
  - Step 3. Test

Summary

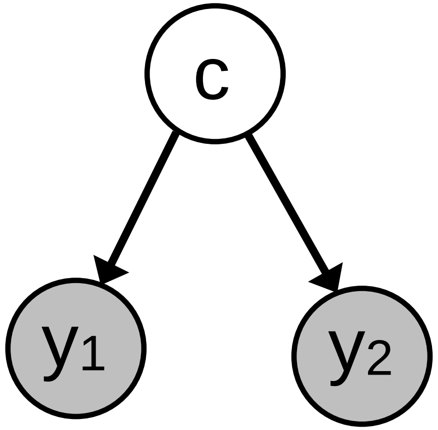
A Bayes Network expresses a unique probability factorization:



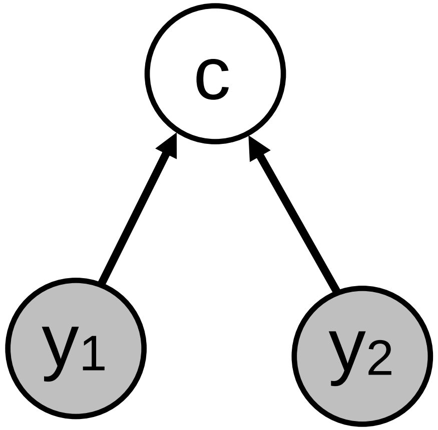
$$p(x) = \prod_{s \in \mathcal{V}} p(x_s \mid x_{\mathrm{Pa}(s)})$$

Parents of node *s*

Inference is performed by Bayes' rule (posterior distribution):

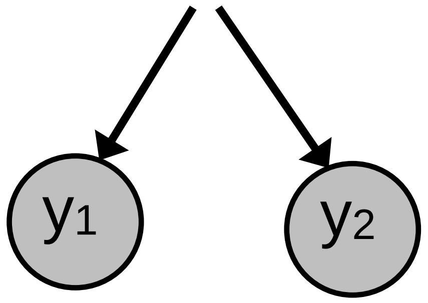


$$p(c \mid y_1, y_2) = \frac{p(c)p(y_1 \mid c)p(y_2 \mid c)}{p(y_1, y_2)}$$



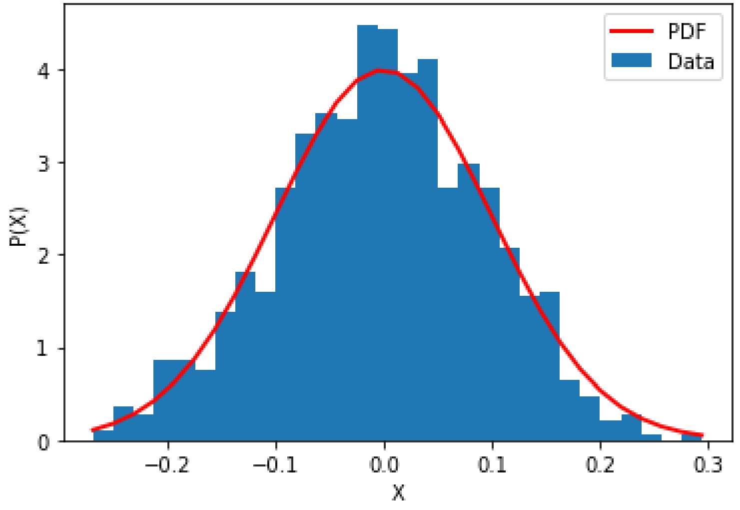
Summary

Hyperparameters must be estimated (e.g. Maximum Likelihood):

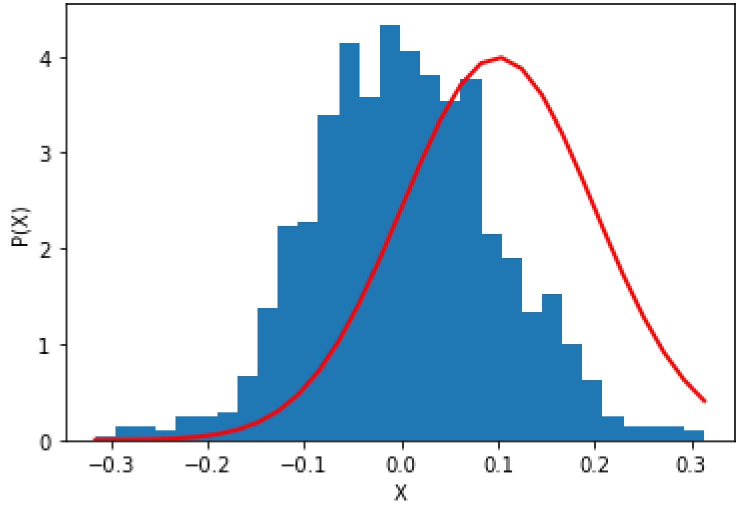


$$\hat{\theta}^{\mathrm{MLE}} = \arg\max_{\theta} \log p(y_1, \ldots, y_n \mid \theta)$$

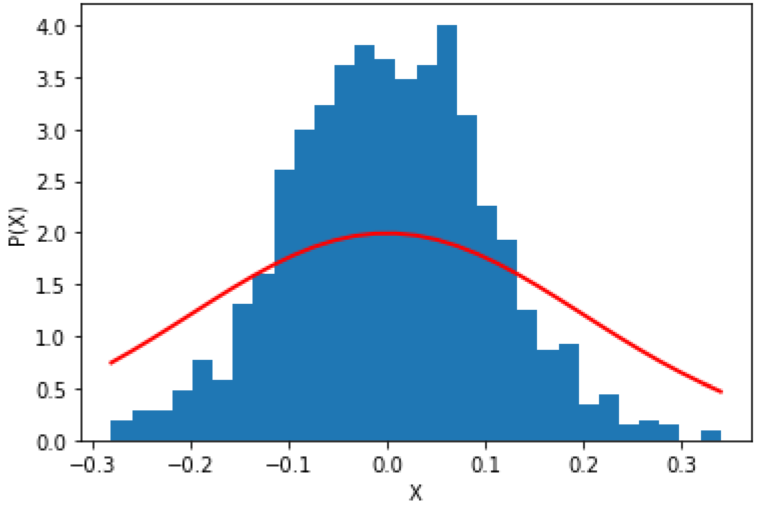
High Likelihood



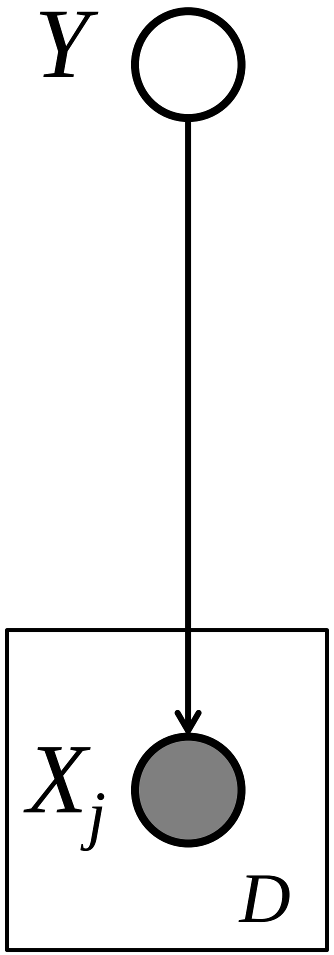
Low Likelihood (mean)



Low Likelihood (variance)



Summary



Naïve Bayes classifier assumes features are *conditionally independent* given class Y:

$$x(j) \perp\!\!\!\perp \left(x(1), \ldots, x(j-1), x(j+1), \ldots, x(D)\right) \mid y$$

Joint distribution factorizes as:

$$p(x, y) = p(y) \prod_{j=1}^{F} p(x(j) \mid y)$$

Allows easier fitting of hyperparameters for *class conditional distributions* (they can be fit independently of each other)

Backup

Summary

Fundamental rules of Probability:

- Law of total probability: $p(Y) = \sum_x p(Y, X = x)$
- Probability chain rule: $p(X \mid Y) = \frac{p(X,Y)}{p(Y)}$
- Conditional probability: $p(X, Y) = p(Y)p(X \mid Y)$

Independence of Random Variables:

- Two RVs are independent if: $p(X = x, Y = y) = p(X = x)p(Y = y)$
- Or: $p(X \mid Y) = p(X)$
- They are *conditionally independent* if:

$$p(X = x, Y = y \mid Z = z) = p(X = x \mid Z = z)p(Y = y \mid Z = z)$$

- Or: $p(X \mid Y, Z) = p(X \mid Z)$

Administrivia

- Homework submission
  - Make sure questions are answered in PDF
  - Match pages to questions
  - Put code in PDF (relevant parts of code at least)
  - Doublecheck your submission
- Midterm Exam
  - Thursday 10/12
  - No coding
  - Probably closed-book

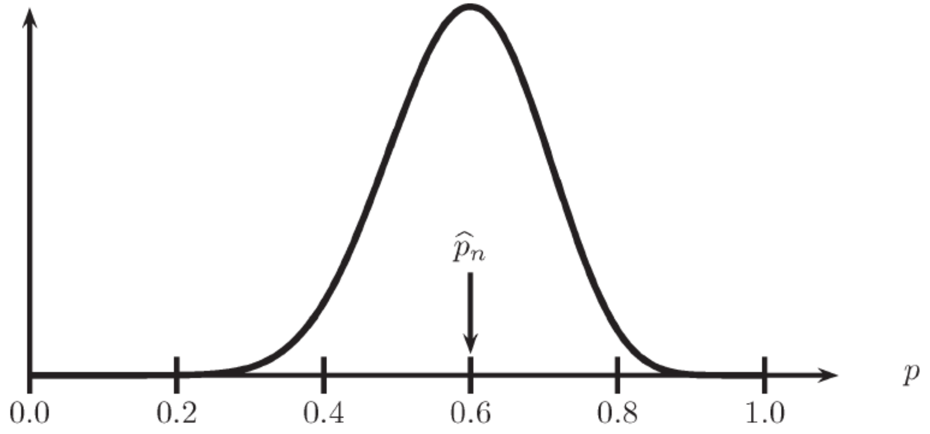
Maximum Likelihood

[Source: Wasserman, L. 2004]

Example Suppose we have N coin tosses with $X_1, \ldots, X_n \sim \text{Bernoulli}(p)$ but we don't know the coin bias p. The likelihood function is,

$$\mathcal{L}_n(p) = \prod_{i=1}^{n} p^{x_i}(1-p)^{1-x_i} = p^S(1-p)^{n-S}$$

where $S = \sum_i x_i$. The log-likelihood is,



Likelihood function for Bernoulli with n=20 and $\sum_i x_i = 12$ heads

$$\log \mathcal{L}_n(p) = S \log p + (n-S)\log(1-p)$$

Set the derivative of $\log \mathcal{L}_n(p)$ to zero and solve,

$$\hat{p}^{\text{MLE}} = S/n = \frac{1}{n}\sum_{i=1}^{n} x_i$$

Maximum likelihood is equivalent to sample mean in Bernoulli

Maximum Likelihood

Maximum Likelihood Estimator (MLE) as the name suggests, maximizes the likelihood function.

$$\hat{\theta}^{\mathrm{MLE}} = \arg\max_{\theta} \mathcal{L}_N(\theta) = \prod_{i=1}^{N} p(x_i; \theta)$$

Intuition: find model θ that is best supported by data