Machine Learning

NYU Shanghai

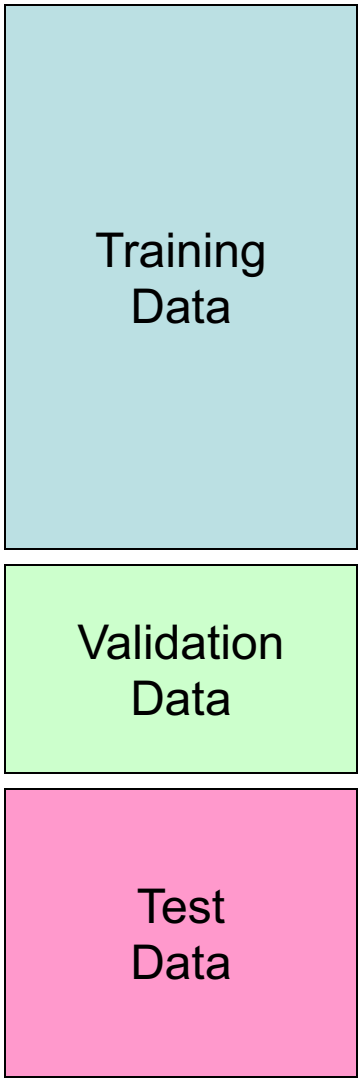
Spring 2020

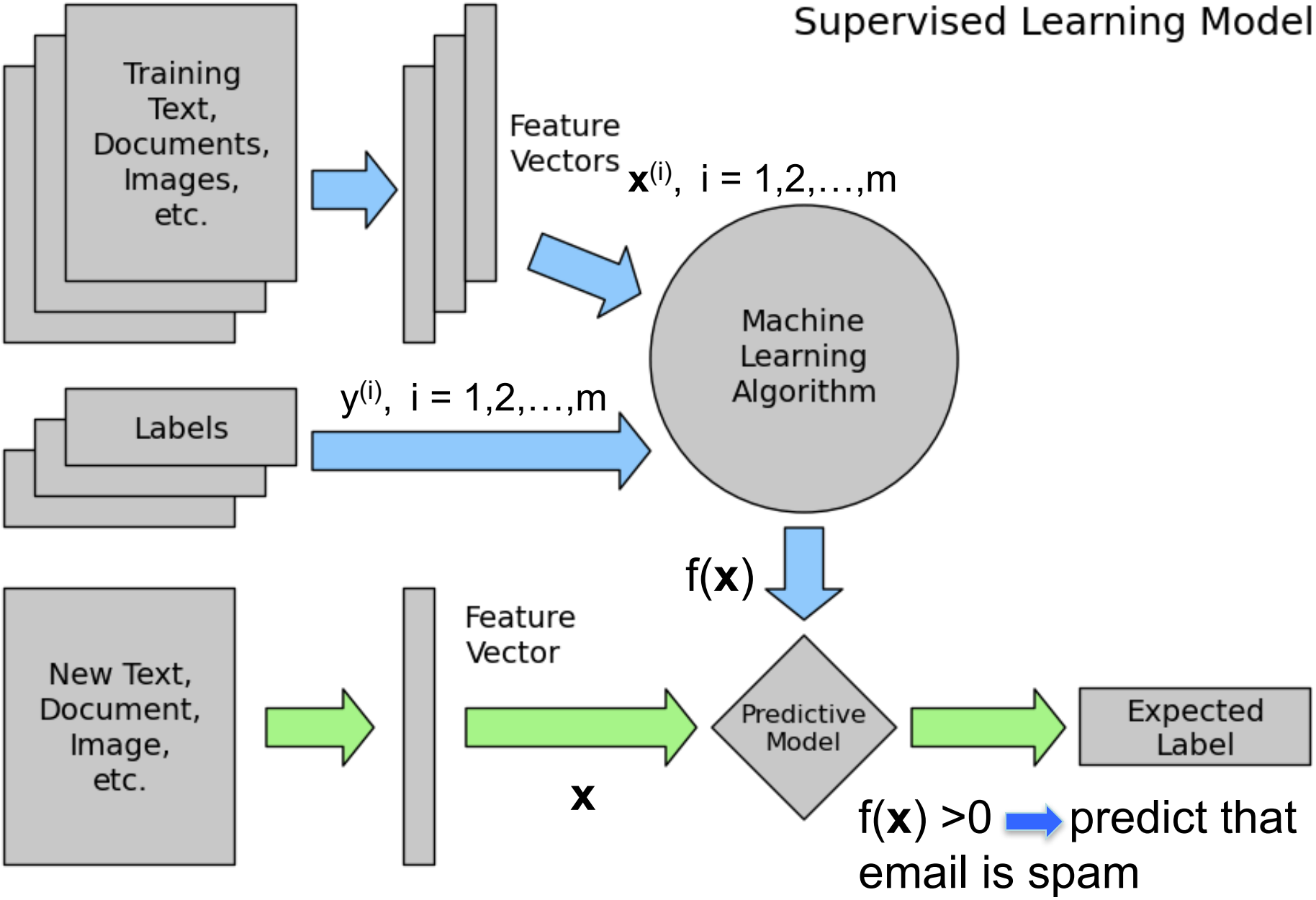
Some Popular Types of Machine Learning

- Supervised vs. unsupervised
  - Supervised often used in practice
  - But… unsupervised is closer to human learning
- Discriminative vs. generative
  - discriminative often used in practice
  - But… generative is more fun
- Deep Learning
  - Deep Learning is just a fancy term for Neural networks ++
- Reinforcement Learning

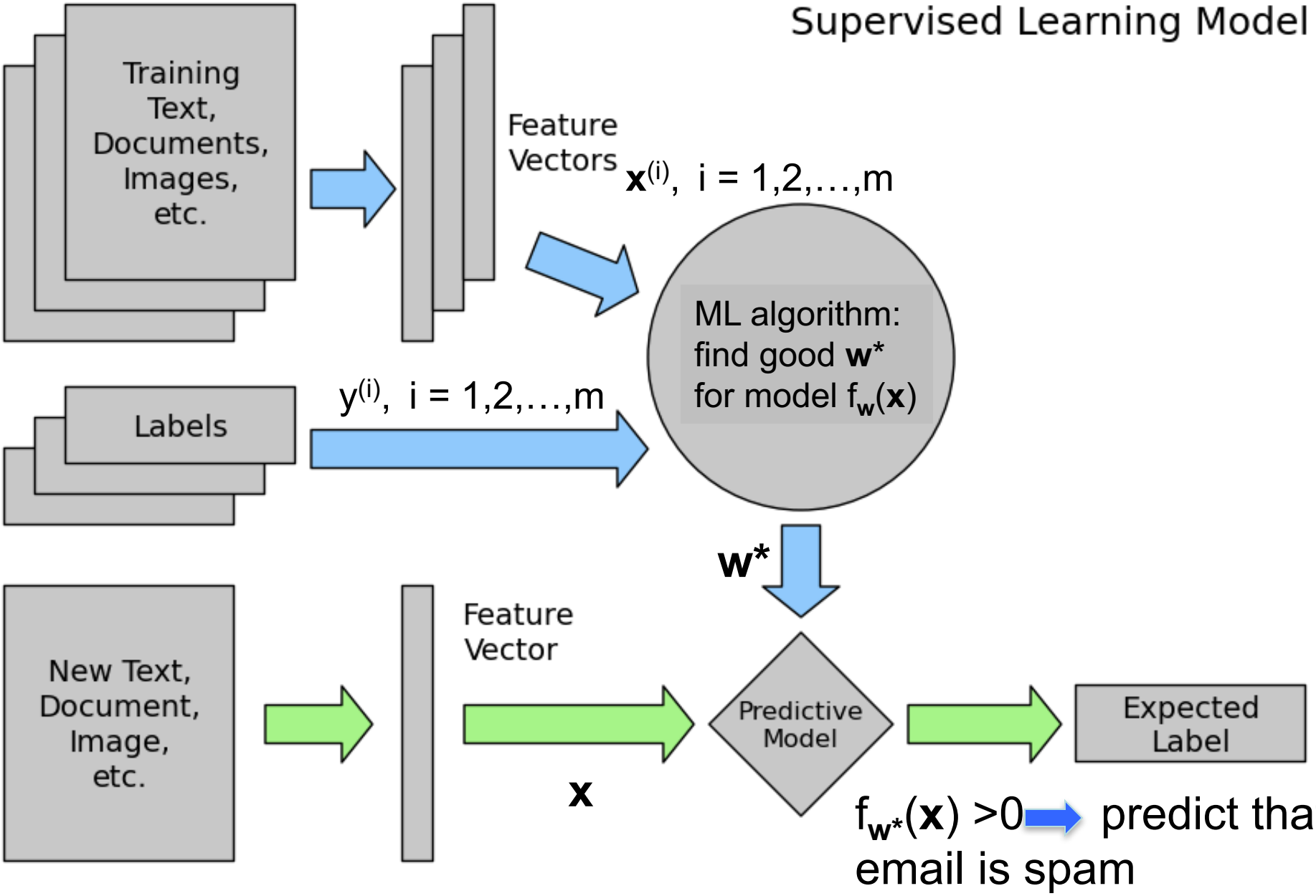
ML Methodology

- Experimentation cycle
  - Select model, features, algorithm
  - Train using **training data** (ie, use algorithm to find good weights)
  - Evaluate using **validation data**: number of instances in validation set predicted correctly.
  - Modify features, hyperparameters
  - Train again….
  - Find best features, hyperparameters
  - ***Avoid Overfitting: to avoid a good performance on training data but a bad performance on validation data***
  - Using bestfeatures and hyperparameters, train with both training data and validation data
  - Compute final accuracy of test data
  - Very important: never "peek" at the test set!




Supervised Learning Model
Training Text, Documents, Images, etc.
Feature Vectors
$\mathbf{x}^{(i)}$, i = 1,2,…,m
Machine Learning Algorithm
Labels
$y^{(i)}$, i = 1,2,…,m
f(**x**)
New Text, Document, Image, etc.
Feature Vector
x
Predictive Model
Expected Label
f(**x**) >0 predict that email is spam

Supervised Learning Model



The Model

- Assumptions:
  - The model is a parameterized function $f_{\mathbf{w}}(\mathbf{x})$, where **w** is a vector of parameters (also called weights).
  - $f_{\mathbf{w}}(\mathbf{x})$ has a specific form. E.g., $f_{\mathbf{w}}(\mathbf{x}) = w_1x_1+w_2x_2+\ldots+w_nx_n$ is a *linear model*.
- Under the assumptions above, the learning process only determines what specific **w** we should use.
- We often use **w*** to denote the best **w**

The perceptron algorithm

- Start with weight vector = $\mathbf{w}$ = (0,0,….0)
- Cycle through the training examples $\mathbf{x}^{(i)}, y^{(i)}$, i=1,2,..,m,1,2,..,m,..

- For i=1,2,..,m,1,2,..,m,…

  Any problem?
  这么强的假设，有没有副作用?

  – if $y^{(i)}(\mathbf{w} \bullet \mathbf{x}^{(i)}) > 0$: (example correctly classified)
  $\mathbf{w} = \mathbf{w}$
  – else: (example incorrectly classified)
  $\mathbf{w} = \mathbf{w} + y^{(i)} \mathbf{x}^{(i)}$

If for some $\mathbf{w}$, get $y^{(i)}(\mathbf{w} \bullet \mathbf{x}^{(i)}) > 0$ for all i=1,…,m, stop and use that $\mathbf{w}$!

what does all this have to do with function approximation?

instead of $F: X \rightarrow Y$,

learn $P(Y \mid X)$

The Joint Distribution

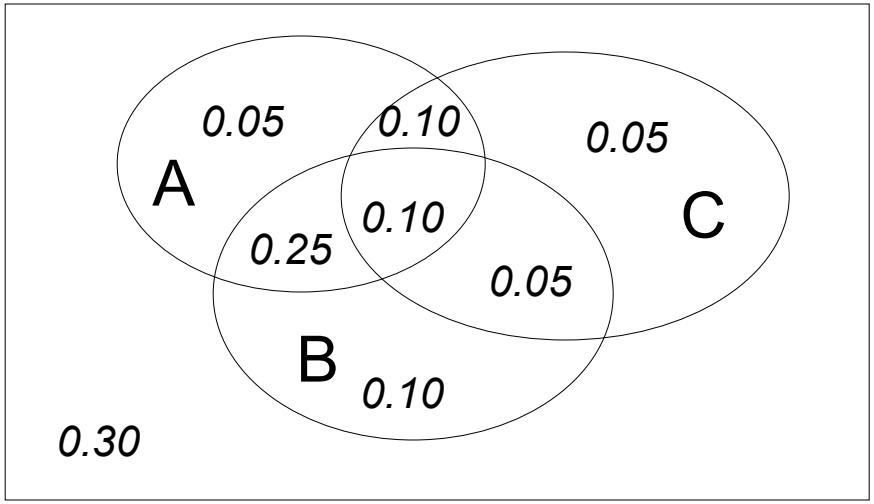
Example: Boolean variables A, B, C

Recipe for making a joint distribution of M variables:

1. Make a truth table listing all combinations of values (M Boolean variables → 2^M rows).

2. For each combination of values, say how probable it is.

A	B	C	Prob
0	0	0	0.30
0	0	1	0.05
0	1	0	0.10
0	1	1	0.05
1	0	0	0.05
1	0	1	0.10
1	1	0	0.25
1	1	1	0.10



[A. Moore]

sounds like the solution to learning F: X → Y, or P(Y | X).

Are we done?

Learning and the Joint Distribution

gender	hours_worked	wealth	
Female	v0:40.5-	poor	0.253122
		rich	0.0245895
	v1:40.5+	poor	0.0421768
		rich	0.0116293
Male	v0:40.5-	poor	0.331313
		rich	0.0971295
	v1:40.5+	poor	0.134106
		rich	0.105933

Suppose we want to learn the function f: <G, H> → W

Equivalently, P(W | G, H)

Solution: learn joint distribution from data, calculate P(W | G, H)

e.g., P(W=rich | G = female, H = 40.5-) =

$$\frac{P(W=r \wedge G=f \wedge H=40-)}{P(\quad G=f \wedge H=40-)} = \frac{.024}{.277} \approx .09$$

[A. Moore]

sounds like the solution to
learning F: X → Y,
or P(Y | X). $2^{10} = 1024$

Main problem: learning P(Y|X)
can require more data than we have

consider learning Joint Dist. with 100 attributes
# of rows in this table? $2^{100} \approx 1000^{10} = 10^{30}$
# of people on earth? 10^9
fraction of rows with 0 training examples? 0.9999

Can we reduce params using Bayes Rule?

Suppose X =<X_1,… X_n>
where X_i and Y are boolean RV's

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$$

How many parameters to define $P(X_1,\ldots X_n \mid Y)$?

$P(X \mid Y=1) \rightarrow 2^n - 1$

$P(X \mid Y=0) \rightarrow 2^n - 1$

How many parameters to define P(Y)?

question: why don't we have to estimate p(x)?

Naïve Bayes

Naïve Bayes assumes

$$P(X_1 \ldots X_n|Y) = \prod_i P(X_i|Y)$$

i.e., that X_i and X_j are conditionally independent given Y, for all $i \neq j$

Naïve Bayes uses assumption that the X_i are conditionally independent, given Y. E.g., $P(X_1|X_2, Y) = P(X_1|Y)$

Given this assumption, then:

$$P(X_1, X_2|Y) = P(X_1|X_2, Y)P(X_2|Y)$$
$$= P(X_1|Y)P(X_2|Y)$$

in general: $P(X_1 \ldots X_n|Y) = \prod_i P(X_i|Y)$

How many parameters to describe $P(X_1 \ldots X_n|Y)$? $P(Y)$?

- Without conditional indep assumption? $2(2^n - 1) + 1$
- With conditional indep assumption? $2n + 1$

Again, any problem?

Naïve Bayes: Subtlety #1

Often the X_i are not really conditionally independent

- We use Naïve Bayes in many cases anyway, and it often works pretty well
  - often the right classification, even when not the right probability (see [Domingos&Pazzani, 1996])

- What is effect on estimated P(Y|X)?
  - Extreme case: what if we add two copies: $X_i = X_k$

$$P(Y=1|X) = P(Y=1)\,P(X_1|Y=1)\,P(X_2|Y=1)\ldots$$

Extreme case: what if we add two copies: $X_i = X_k$

we know it is harmful, what to do?

Naïve Bayes: Subtlety #2

If unlucky, our MLE estimate for $P(X_i \mid Y)$ might be zero.
(for example, X_i = *birthdate*. X_i = *Jan_25_1992*)

$0 = \hat{P}(X_i = 1 \mid S = 1)$

- Why worry about just one parameter out of many?

$$P(Y=1 \mid X_1 \cdots X_n) = \frac{P(Y=1) \prod_i P(X_i \mid Y=1)}{P(X_1 \cdots X_n)}$$

- What can be done to address this?

→ MAP estimate!

Estimating Parameters: Y, X_i discrete-valued

Maximum likelihood estimates:

$$\hat{\pi}_k = \hat{P}(Y = y_k) = \frac{\#D\{Y = y_k\}}{|D|}$$

$$\hat{\theta}_{ijk} = \hat{P}(X_i = x_j | Y = y_k) = \frac{\#D\{X_i = x_j \wedge Y = y_k\}}{\#D\{Y = y_k\}}$$

MAP estimates (Beta, Dirichlet priors):

$$\hat{\pi}_k = \hat{P}(Y = y_k) = \frac{\#D\{Y = y_k\} + (\beta_k - 1)}{|D| + \sum_m (\beta_m - 1)}$$

$$\hat{\theta}_{ijk} = \hat{P}(X_i = x_j | Y = y_k) = \frac{\#D\{X_i = x_j \wedge Y = y_k\} + (\beta_k - 1)}{\#D\{Y = y_k\} + \sum_m (\beta_m - 1)}$$

Only difference: "imaginary" examples

What if we have continuous X_i?

Gaussian Naïve Bayes (GNB): assume

$$p(X_i = x | Y = y_k) = \frac{1}{\sqrt{2\pi\sigma_{ik}^2}} e^{-\frac{1}{2}\left(\frac{x-\mu_{ik}}{\sigma_{ik}}\right)^2}$$

Sometimes assume variance

- is independent of Y (i.e., σ_i),
- or independent of X_i (i.e., σ_k)
- or both (i.e., σ)

Gaussian Naïve Bayes Algorithm – continuous X_i (but still discrete Y)

- Train Naïve Bayes (examples)

  for each value y_k

  estimate* $\pi_k \equiv P(Y = y_k)$

  for each attribute X_i estimate $P(X_i|Y = y_k)$

  - class conditional mean μ_{ik}, variance σ_{ik}

- Classify (X^{new})

$$Y^{new} \leftarrow \arg\max_{y_k} \; P(Y = y_k) \prod_i P(X_i^{new}|Y = y_k)$$

$$Y^{new} \leftarrow \arg\max_{y_k} \; \pi_k \prod_i \mathcal{N}(X_i^{new}; \mu_{ik}, \sigma_{ik})$$

* probabilities must sum to 1, so need estimate only n-1 parameters...

Estimating Parameters: Y discrete, X_i continuous

Maximum likelihood estimates:

$$\hat{\mu}_{ik} = \frac{1}{\sum_j \delta(Y^j = y_k)} \sum_j X_i^j \delta(Y^j = y_k)$$

jth training example

ith feature

kth class

δ()=1 if (y^j=y_k) else 0

$$\hat{\sigma}_{ik}^2 = \frac{1}{\sum_j \delta(Y^j = y_k)} \sum_j (X_i^j - \hat{\mu}_{ik})^2 \delta(Y^j = y_k)$$

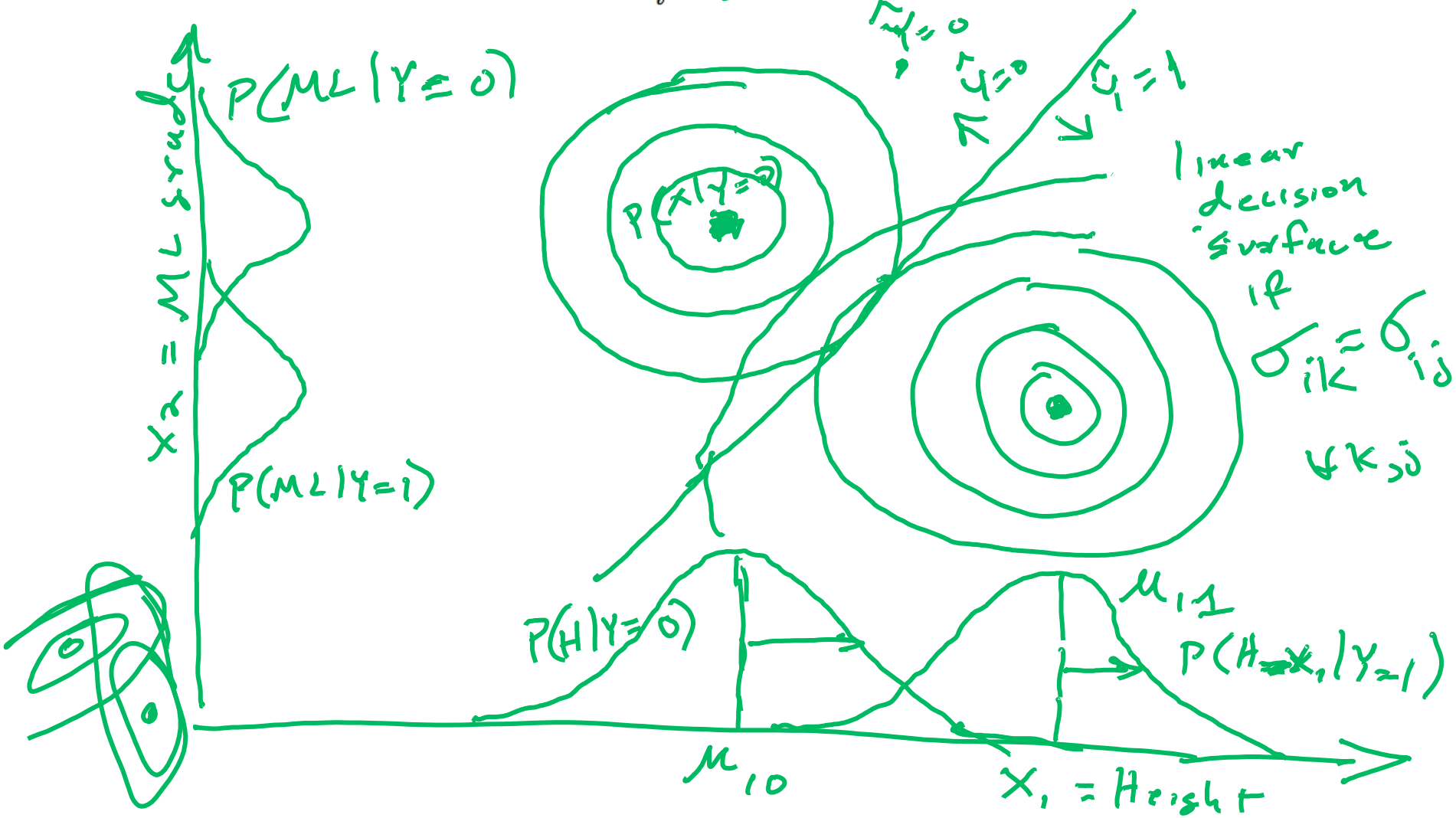
we can have imaginary count here, too.

Gaussian Naïve Bayes – Big Picture

Example: Y= PlayBasketball (boolean), X1=Height, X2=MLgrade

$$Y^{new} \leftarrow \arg\max_{y \in \{0,1\}} P(Y=y) \prod_i P(X_i^{new} | Y=y)$$

assume P(Y=1) = 0.5

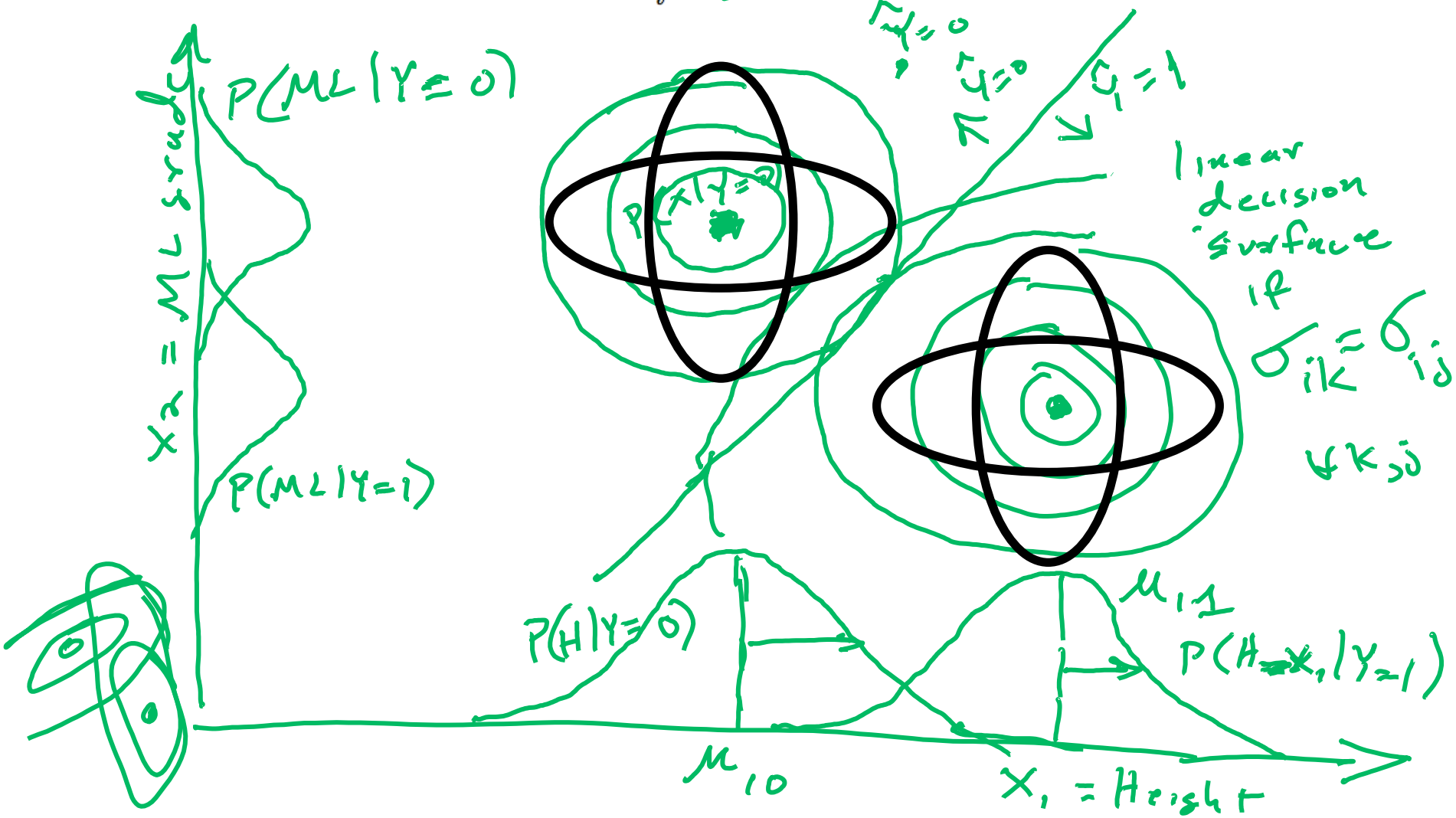


Gaussian Naïve Bayes – Big Picture

Example: Y= PlayBasketball (boolean), X1=Height, X2=MLgrade

$$Y^{new} \leftarrow \arg\max_{y \in \{0,1\}} P(Y=y) \prod_i P(X_i^{new} | Y=y)$$

assume P(Y=1) = 0.5



Logistic Regression

Idea:

- Naïve Bayes allows computing P(Y|X) by learning P(Y) and P(X|Y)

- Why not learn P(Y|X) directly?

- Consider learning f: X → Y, where
  - X is a vector of real-valued features, < X_1 … X_n >
  - Y is boolean
  - assume all X_i are conditionally independent given Y
  - model $P(X_i \mid Y = y_k)$ as Gaussian $N(\mu_{ik}, \sigma_i)$ not σ_{ik}
  - model P(Y) as Bernoulli (π)

- What does that imply about the form of P(Y|X)?

$$P(Y = 1 | X =< X_1, ...X_n >) = \frac{1}{1 + exp(w_0 + \sum_i w_i X_i)}$$

Very convenient!

$$P(Y=1|X=<X_1,...X_n>)=\frac{1}{1+exp(w_0+\sum_i w_i X_i)}$$

implies

$$P(Y=0|X=<X_1,...X_n>)=\frac{exp(w_0+\sum_i w_i X_i)}{1+exp(w_0+\sum_i w_i X_i)}$$

implies

$$\frac{P(Y=0|X)}{P(Y=1|X)}=exp(w_0+\sum_i w_i X_i)$$

linear classification rule!

implies

$$\ln\frac{P(Y=0|X)}{P(Y=1|X)}=w_0+\sum_i w_i X_i$$

Back to Supervised Machine Learning

- $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ "feature vector"
- $f_{\mathbf{w}}(\mathbf{x})$ is a model
- *For example, Linear Model*:
  $f_{\mathbf{w}}(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2 + \ldots + w_n x_n$
  where $\mathbf{w} = (w_0, w_1, w_2, .., w_n)$
- ML algorithm uses training set to find good w^* for $f_{\mathbf{w}}(\mathbf{x})$
- Given new email **x**, predict **x** is SPAM if $f_{\mathbf{w}^*}(\mathbf{x}) > 0$; otherwise not SPAM

Hyperplane

- Let **w =** $(w_1,w_2,..,w_n)$ be any fixed vector and c any scalar
- The equation $w_1u_1 + w_2u_2 + \ldots + w_nu_n = c$ (equivalently **w•u** = c) defines a hyperplane in n-dimensional space.
- **w** is perpendicular to plane
  - The hyperplane cuts R^n into two halves
  - Points **w•u** > c lie on one side of plane
  - Points **w•u** < c lie on the other side

Derive form for $P(Y|X)$ for Gaussian $P(X_i|Y=y_k)$ assuming $\sigma_{ik} = \sigma_i$

$$P(Y=1|X) = \frac{P(Y=1)P(X|Y=1)}{P(Y=1)P(X|Y=1) + P(Y=0)P(X|Y=0)}$$

$$= \frac{1}{1 + \frac{P(Y=0)P(X|Y=0)}{P(Y=1)P(X|Y=1)}}$$

$x = \exp(\ln x)$

$$= \frac{1}{1 + \exp(\ln \frac{P(Y=0)P(X|Y=0)}{P(Y=1)P(X|Y=1)})}$$

$\pi \equiv P(Y=1)$

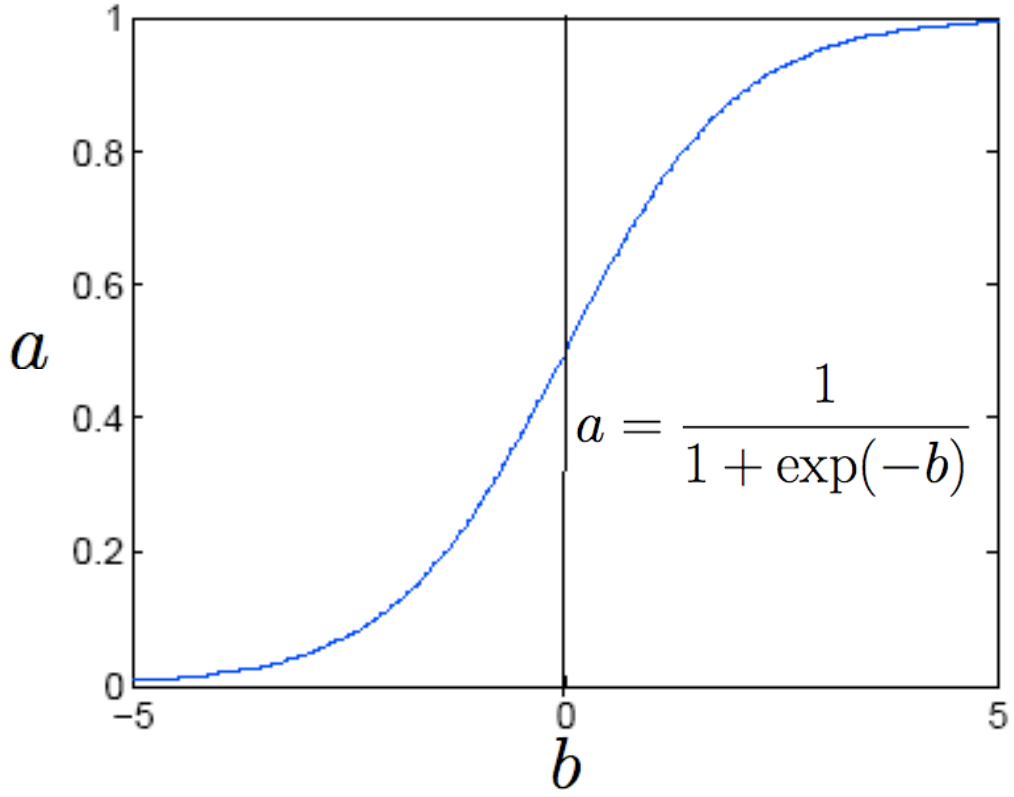
$$= \frac{1}{1 + \exp(\ (\ln \frac{1-\pi}{\pi}) + \sum_i \ln \frac{P(X_i|Y=0)}{P(X_i|Y=1)})}$$

$$P(x \mid y_k) = \frac{1}{\sigma_{ik}\sqrt{2\pi}} \; e^{\frac{-(x-\mu_{ik})^2}{2\sigma_{ik}^2}}$$

$$\sum_i \left(\frac{\mu_{i0} - \mu_{i1}}{\sigma_i^2} X_i + \frac{\mu_{i1}^2 - \mu_{i0}^2)}{2\sigma_i^2} \right)$$

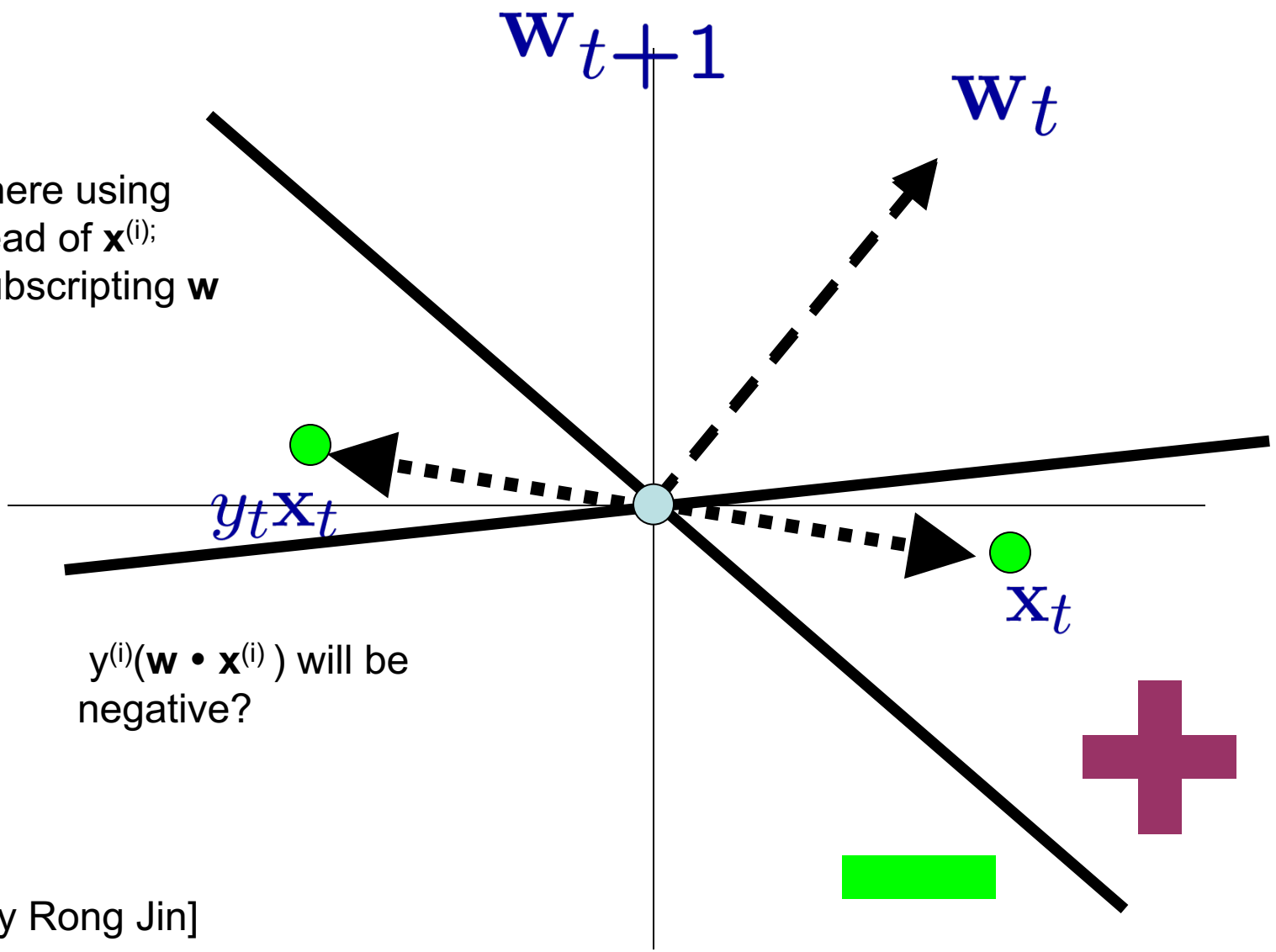
$$P(Y=1|X) = \frac{1}{1 + \exp(w_0 + \sum_{i=1}^{n} w_i X_i)}$$

Logistic function



$$P(Y = 1|X) = \frac{1}{1 + \exp(w_0 + \sum_{i=1}^{n} w_i X_i)}$$

Geometrical Interpretation



Note: here using $\mathbf{x}_t$ instead of $\mathbf{x}^{(i)}$; also subscripting **w**

$y^{(i)}(\mathbf{w} \bullet \mathbf{x}^{(i)})$ will be negative?

[Slide by Rong Jin]

Training Logistic Regression: MCLE

- Choose parameters W=<w_0, ... w_n> to <u>maximize conditional likelihood</u> of training data

where

$$P(Y = 0|X, W) = \frac{1}{1 + exp(w_0 + \sum_i w_i X_i)}$$

$$P(Y = 1|X, W) = \frac{exp(w_0 + \sum_i w_i X_i)}{1 + exp(w_0 + \sum_i w_i X_i)}$$

- Training data D = $\{\langle X^1, Y^1\rangle, \ldots \langle X^L, Y^L\rangle\}$
- Data likelihood = $\prod_l P(X^l, Y^l|W)$
- Data <u>conditional</u> likelihood = $\prod_l P(Y^l|X^l, W)$

$$W_{MCLE} = \arg\max_W \prod_l P(Y^l|W, X^l)$$

Expressing Conditional Log Likelihood

$$l(W) \equiv \ln \prod_l P(Y^l|X^l, W) = \sum_l \ln P(Y^l|X^l, W)$$

$$P(Y = 0|X, W) = \frac{1}{1 + exp(w_0 + \sum_i w_i X_i)}$$

$$P(Y = 1|X, W) = \frac{exp(w_0 + \sum_i w_i X_i)}{1 + exp(w_0 + \sum_i w_i X_i)}$$

$$\begin{aligned} l(W) &= \sum_l Y^l \ln P(Y^l = 1|X^l, W) + (1 - Y^l) \ln P(Y^l = 0|X^l, W) \\ &= \sum_l Y^l \ln \frac{P(Y^l = 1|X^l, W)}{P(Y^l = 0|X^l, W)} + \ln P(Y^l = 0|X^l, W) \\ &= \sum_l Y^l (w_0 + \sum_i^n w_i X_i^l) - \ln(1 + exp(w_0 + \sum_i^n w_i X_i^l)) \end{aligned}$$

Maximizing Conditional Log Likelihood

$$P(Y=0|X,W) = \frac{1}{1+exp(w_0+\sum_i w_i X_i)}$$

$$P(Y=1|X,W) = \frac{exp(w_0+\sum_i w_i X_i)}{1+exp(w_0+\sum_i w_i X_i)}$$

$$\begin{aligned} l(W) &\equiv \ln \prod_l P(Y^l|X^l,W) \\ &= \sum_l Y^l(w_0+\sum_i^n w_i X_i^l) - \ln(1+exp(w_0+\sum_i^n w_i X_i^l)) \end{aligned}$$

Good news: $l(W)$ is concave function of W

Bad news: no closed-form solution to maximize $l(W)$

An example of closed-form solution

The solution of any second-degree polynomial equation can be expressed in terms of its coefficients, using only addition, subtraction, multiplication, division, and square roots, in the familiar quadratic formula: the roots of the polynomial $ax^2 + bx + c$ (with $a \neq 0$) are

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

Maximize Conditional Log Likelihood: Gradient Ascent

$$l(W) \equiv \ln \prod_l P(Y^l|X^l, W)$$

$$= \sum_l Y^l(w_0 + \sum_i^n w_i X_i^l) - \ln(1 + exp(w_0 + \sum_i^n w_i X_i^l))$$

$$\frac{\partial l(W)}{\partial w_i} = \sum_l X_i^l(Y^l - \hat{P}(Y^l = 1|X^l, W))$$

Gradient ascent algorithm: iterate until change < ε

For all i, repeat

$$w_i \leftarrow w_i + \eta \sum_l X_i^l(Y^l - \hat{P}(Y^l = 1|X^l, W))$$

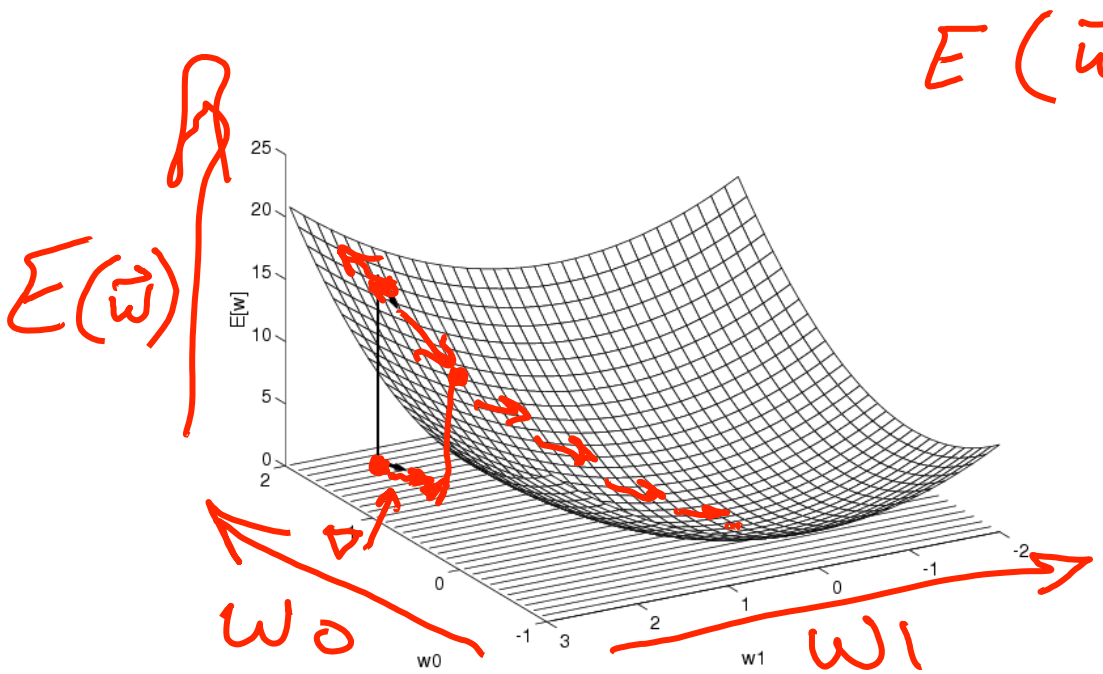
If $f(x) = e^x$, then $f'(x) = e^x$.

If $f(x) = a^x$, $a > 0, a \neq 1$, then $f'(x) = (\ln a) \cdot a^x$.

If $f(x) = \ln x$, then $f'(x) = \frac{1}{x}$.

If $f(x) = \log_a x$, $a > 0, a \neq 1$, then $f'(x) = \frac{1}{(\ln a) \cdot x}$.

Gradient Descent



$E(\vec{w})$

$E(\vec{w})$

$\vec{w}^{(i+1)} = \vec{w}^{(i)} - \nabla E[\vec{w}]$

Gradient

$$\nabla E[\vec{w}] \equiv \left[\frac{\partial E}{\partial w_0}, \frac{\partial E}{\partial w_1}, \cdots \frac{\partial E}{\partial w_n}\right]$$

Training rule:

$$\Delta \vec{w} = -\eta \nabla E[\vec{w}]$$

i.e.,

$$\Delta w_i = -\eta \frac{\partial E}{\partial w_i}$$

MAP estimates and Regularization

- Maximum a posteriori estimate with prior W~N(0,σI)

$$W \leftarrow \arg\max_W \ln[P(W) \prod_l P(Y^l|X^l, W)]$$

$$w_i \leftarrow w_i - \eta\lambda w_i + \eta \sum_l X_i^l (Y^l - \hat{P}(Y^l = 1|X^l, W))$$

called a "regularization" term

- helps reduce overfitting
- keep weights nearer to zero (if P(W) is zero mean Gaussian prior), or whatever the prior suggests
- used very frequently in Logistic Regression

G.Naïve Bayes vs. Logistic Regression

[Ng & Jordan, 2002]

Recall two assumptions deriving form of LR from GNBayes:

1. X_i conditionally independent of X_k given Y
2. $P(X_i \mid Y = y_k) = N(\mu_{ik},\sigma_i)$, ← not $N(\mu_{ik},\sigma_{ik})$

Consider three learning methods:

- GNB (assumption 1 only) -- decision surface can be non-linear
- GNB2 (assumption 1 and 2) – decision surface linear
- LR -- decision surface linear, trained without assumption 1.

Which method works better if we have *infinite* training data, and...

- Both (1) and (2) are satisfied: LR = GNB2 = GNB
- (1) is satisfied, but not (2) : GNB > GNB2, GNB > LR, LR > GNB2
- Neither (1) nor (2) is satisfied: GNB>GNB2, LR > GNB2, LR><GNB

Use log likelihood to avoid underflow:
prod —> sum

how can we know if the step size is reasonable?

Bias, Variance and Error

Bias and Variance

given algorithm that outputs estimate $\hat{\theta}$ for θ, we define:

the bias of the estimator: $E[\hat{\theta}] - \theta$

the variance of estimator: $E[\ (\hat{\theta} - E[\hat{\theta}])^2\ ]$

e.g. $\hat{\theta}^{MLE}$ estimator for probability θ of heads, based on n independent coin flips

$$\hat{\theta}^{MLE} = \frac{\alpha_1}{\alpha_1 + \alpha_0}$$

what is its bias?

variance?

$$Var(\hat{\theta}^{MLE}) = \frac{\theta(1-\theta)}{n}$$

Here, the p is theta

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

$$\ell(p) = \log p \sum_{i=1}^{n} x_i + \log(1-p) \sum_{i=1}^{n} (1-x_i)$$

$$\frac{\partial \ell(p)}{\partial p} = \frac{\sum_{i=1}^{n} x_i}{p} - \frac{\sum_{i=1}^{n} (1-x_i)}{1-p} \overset{\text{set}}{=} 0$$

$$\sum_{i=1}^{n} x_i - p \sum_{i=1}^{n} x_i = p \sum_{i=1}^{n} (1-x_i)$$

$$p = \frac{1}{n} \sum_{i=1}^{n} x_i$$

Bias and Variance $E(\hat{\theta} - \theta)^2$

given algorithm that outputs estimate $\hat{\theta}$ for θ, we define:

the bias of the estimator: $E[\hat{\theta}] - \theta$

the variance of estimator: $E[\ (\hat{\theta} - E[\hat{\theta}])^2\ ]$

which estimator has higher bias? higher variance?

$$\hat{\theta}^{MLE} = \frac{\alpha_1}{\alpha_1 + \alpha_0}$$

$\hat{\theta}^{MLE}_n$ bias $= 0$

Var $= \frac{(1-\theta)\theta}{n}$

$$\hat{\theta}^{MAP} = \frac{\alpha_1 + \beta_1 - 1}{(\alpha_1 + \beta_1 - 1) + (\alpha_0 + \beta_0 - 1)}$$

$\hat{\theta}^{MAP}_n$ bias > 0 for finite n

var is less

$P(|X - E[X]| \geq t) \leq \frac{Var(X)}{t^2}$

$$\mathbb{E}[(\hat{\theta} - \theta)^2] = \mathbb{E}(\hat{\theta}^2 + \theta^2 - 2\theta \cdot \hat{\theta}) = \mathbb{E}(\hat{\theta}^2) + \theta^2 - 2\theta \cdot \mathbb{E}(\hat{\theta})$$

$$\text{remember we have: } \mathbb{E}(\hat{\theta}^2) = \text{Var}(\hat{\theta}) + [\mathbb{E}(\hat{\theta})]^2$$

$$\text{So therefore: } \mathbb{E}[(\hat{\theta} - \theta)^2] = \text{Var}(\hat{\theta}) + \underbrace{[\mathbb{E}(\hat{\theta})]^2 + \theta^2 - 2\theta \cdot \mathbb{E}(\hat{\theta})}_{\text{this is square of bias}} = \text{Var}(\hat{\theta}) + [\mathbb{E}(\hat{\theta}) - \theta]^2$$

Bias – Variance decomposition of error

Reading: Bishop chapter 9.1, 9.2

- Consider simple regression problem f:X→Y

$y = f(x) + \varepsilon$

$h(x) = y = w_0 + w_1 x_1$

deterministic

noise N(0,σ)

Define the expected prediction error:

$$E_D\left[\int_y \int_x (h(x) - f(x))^2 p(y|x) p(x) dy dx\right]$$

expectation over training D

learned estimate of f(x)

https://en.wikipedia.org/wiki/Bias%E2%80%93variance_tradeoff#Bias%E2%80%93variance_decomposition_of_squared_error

Sources of error

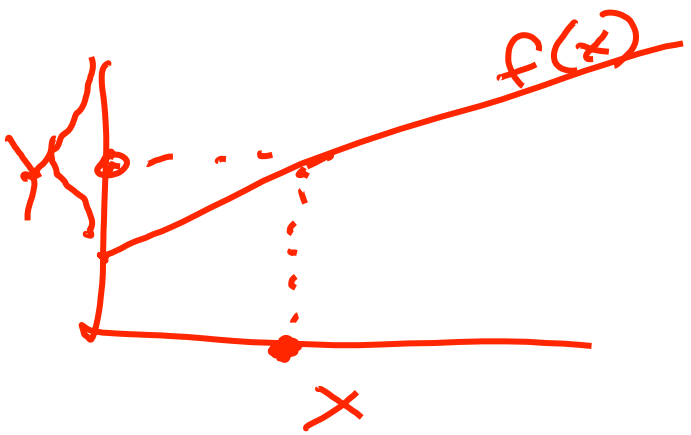
$y = f(x) + \varepsilon$

noise $N(0,\sigma)$

deterministic

What if we have perfect learner, infinite data?

- Our learned h(x) satisfies h(x)=f(x)
- Still have remaining, *unavoidable error*



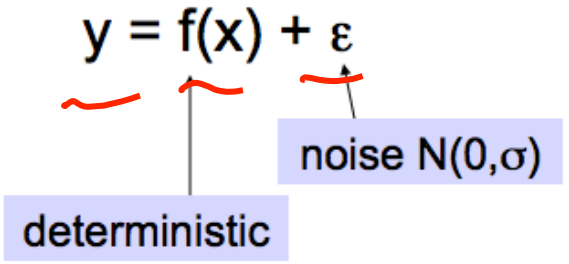
σ^2

Sources of error

- What if we have only n training examples?
- What is our expected error
  - Taken over random training sets of size n, drawn from distribution $D=p(x,y)$

$$E_D\left[\int_y\int_x (h(x)-f(x))^2 p(y|x)p(x)dydx\right]$$

Sources of error



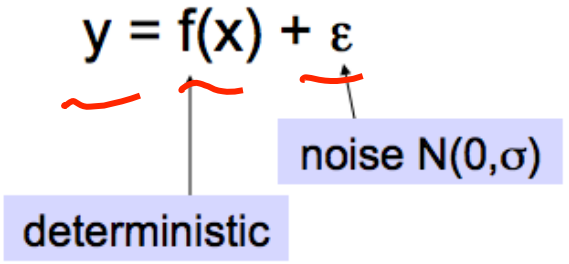
$$E_D\left[\int_y\int_x (h(x) - f(x))^2 p(y|x)p(x)dydx\right]$$

$$= unavoidableError + bias^2 + variance$$

$$bias^2 = \int (E_D[h(x)] - f(x))^2 p(x)dx$$

$$variance = \int E_D[(h(x) - E_D[h(x)])^2]p(x)dx$$

Sources of error



$$E_D\left[\int_y\int_x (h(x) - f(x))^2 p(y|x)p(x)dydx\right]$$

$$= unavoidableError + bias^2 + variance$$

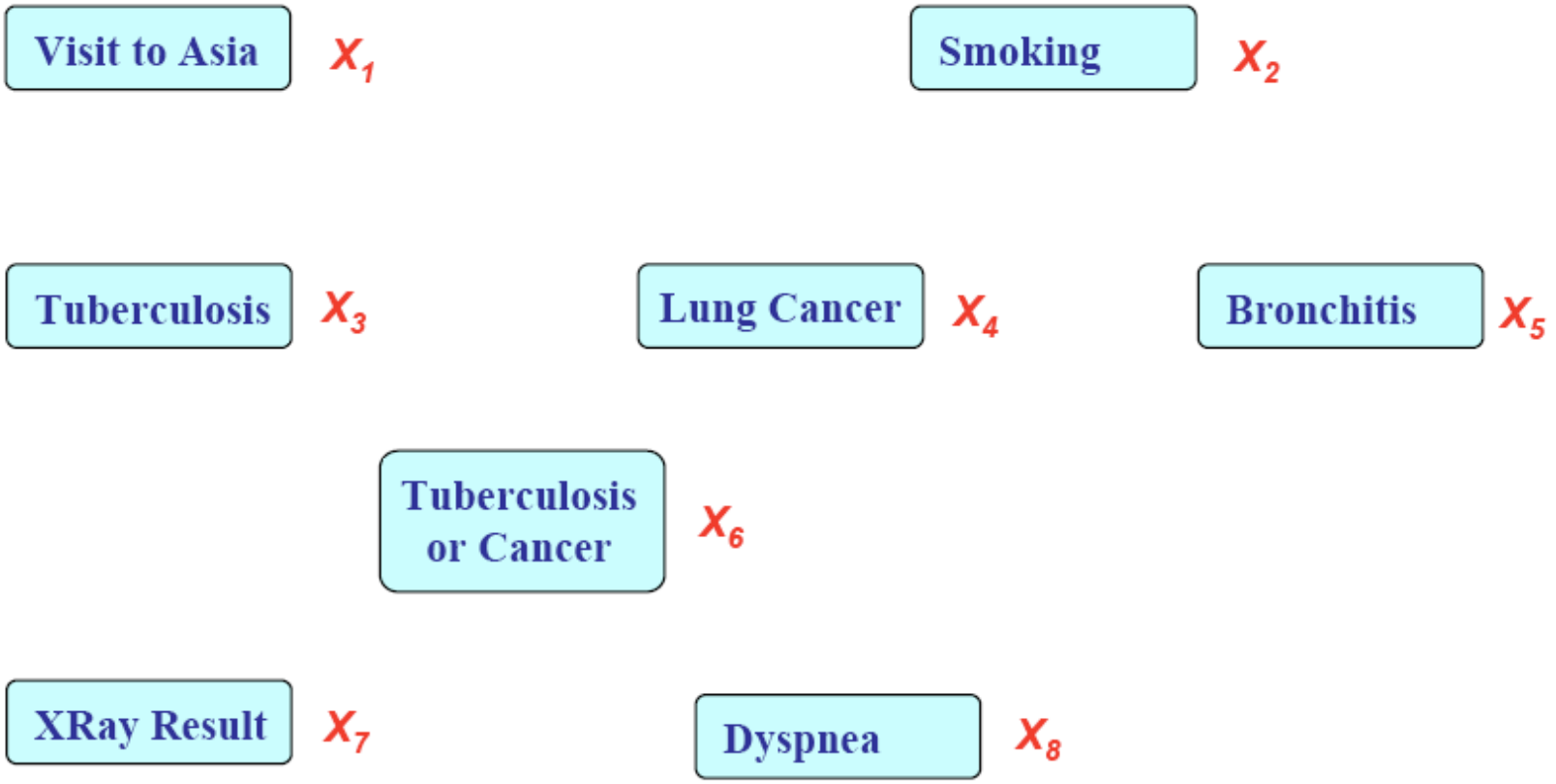
$$bias^2 = \int (E_D[h(x)] - f(x))^2 p(x)dx$$

$$variance = \int E_D[(h(x) - E_D[h(x)])^2]p(x)dx$$

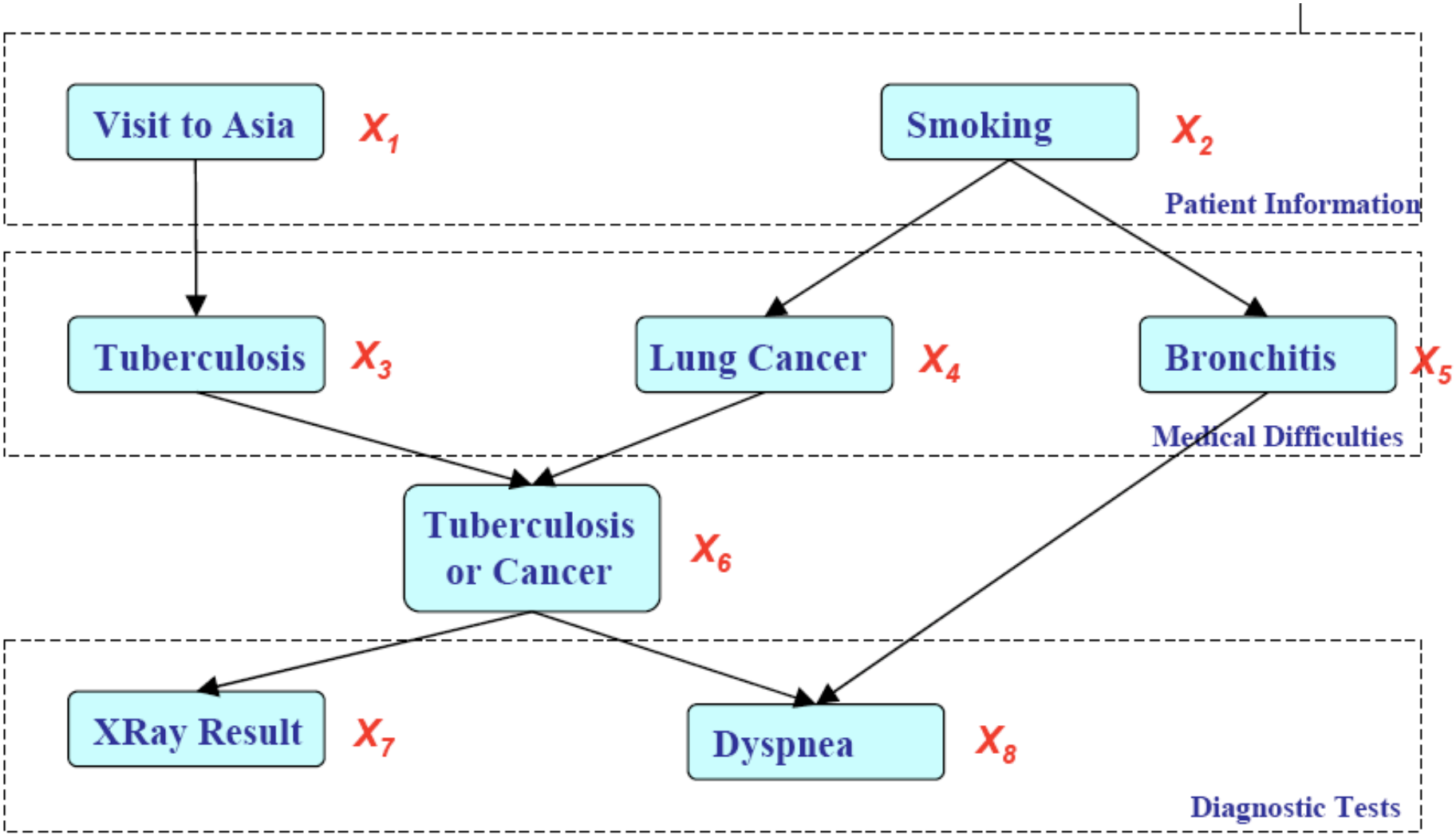
Graphical Models

- Key Idea:
  - Conditional independence assumptions useful
  - but Naïve Bayes is extreme!
  - Graphical models express sets of conditional independence assumptions via graph structure
  - Graph structure plus associated parameters define *joint probability distribution over set of variables*

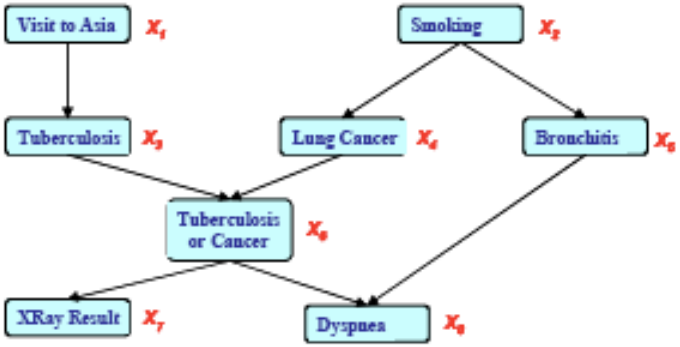
Represent Joint Probability Distribution over Variables



Describe network of dependencies



Bayes Nets define Joint Probability Distribution in terms of this graph, plus parameters



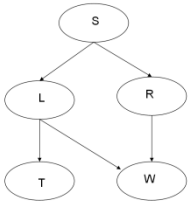
$$P(X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8)$$

$$= P(X_1)\, P(X_2)\, P(X_3|\, X_1)\, P(X_4|\, X_2)\, P(X_5|\, X_2)$$
$$P(X_6|\, X_3, X_4)\, P(X_7|\, X_6)\, P(X_8|\, X_5, X_6)$$

Benefits of Bayes Nets:

- Represent the full joint distribution in fewer parameters, using prior knowledge about dependencies
- Algorithms for inference and learning

Bayesian Networks Definition



Parents	P(W|Pa)	P(¬W|Pa)
L, R	0	1.0
L, ¬R	0	1.0
¬L, R	0.2	0.8
¬L, ¬R	0.9	0.1

A Bayes network represents the joint probability distribution over a collection of random variables

A Bayes network is a directed acyclic graph and a set of conditional probability distributions (CPD's)

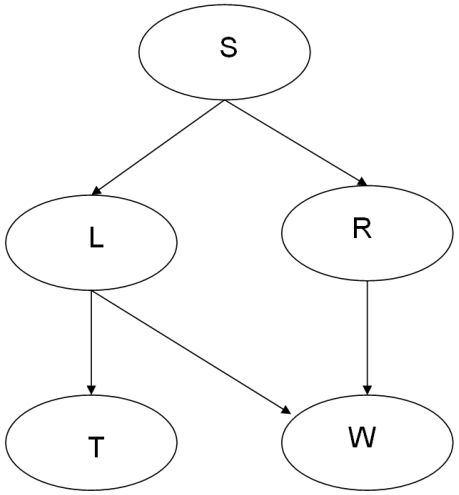
- Each node denotes a random variable
- Edges denote dependencies
- For each node X_i its CPD defines $P(X_i \mid Pa(X_i))$
- The joint distribution over all variables is defined to be

$$P(X_1 \ldots X_n) = \prod_i P(X_i | Pa(X_i))$$

Pa(X) = immediate parents of X in the graph

Bayesian Networks

- CPD for each node X_i describes $P(X_i \mid Pa(X_i))$



Parents	P(W|Pa)	P(¬W|Pa)
L, R	0	1.0
L, ¬R	0	1.0
¬L, R	0.2	0.8
¬L, ¬R	0.9	0.1

W

Chain rule of probability says that in general:

$$P(S, L, R, T, W) = P(S)P(L|S)P(R|S, L)P(T|S, L, R)P(W|S, L, R, T)$$

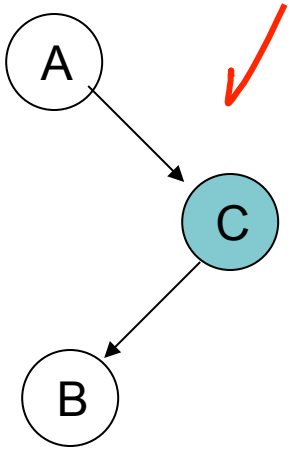
But in a Bayes net: $P(X_1 \ldots X_n) = \prod_i P(X_i|Pa(X_i))$

D-Separation: A formal way to analyze conditional independency

Easy Network 1: Head to Tail

prove A cond indep of B given C?

ie., p(a,b|c) = p(a|c) p(b|c)



$P(A=a)$ simply $P(a)$

$P(a,b|c) = P(a|c)\,P(b|c) \Leftarrow A \perp B \mid C$

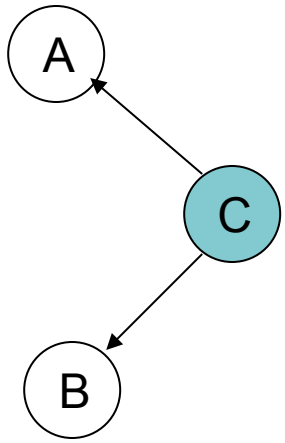
$$P(a,b|c) = \frac{P(a,b,c)}{P(c)} = \frac{P(a)\,P(c|a)}{P(c)}\,P(b|c)$$

$$\frac{P(a,c)}{P(c)} = P(a|c)$$

let's use p(a,b) as shorthand for p(A=a, B=b)

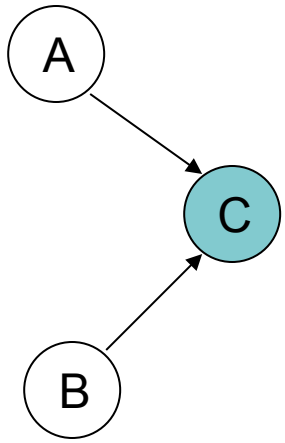
Easy Network 2: Tail to Tail

prove A cond indep of B given C? ie., p(a,b|c) = p(a|c) p(b|c)



let's use p(a,b) as shorthand for p(A=a, B=b)

Easy Network 3: Head to Head



prove A cond indep of B given C? NO!

Summary:

- p(a,b)=p(a)p(b)
- p(a,b|c) NotEqual p(a|c)p(b|c)

Explaining away.

e.g.,

- A=earthquake
- B=breakIn
- C=motionAlarm

X and Y are conditionally independent given Z, **if and only if** X and Y are D-separated by Z.

[Bishop, 8.2.2]

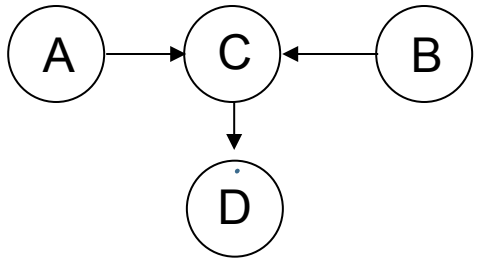
Suppose we have three sets of random variables: X, Y and Z

X and Y are **D-separated** by Z (and therefore conditionally indep, given Z) iff every path from every variable in X to every variable in Y is **blocked**

A path from variable X to variable Y is **blocked** if it includes a node in Z such that either

1. arrows on the path meet either head-to-tail or tail-to-tail at the node and this node is in Z

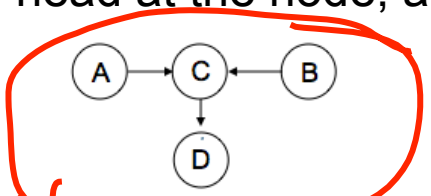
2. or, the arrows meet head-to-head at the node, and neither the node, nor any of its descendants, is in Z



X and Y are **D-separated** by Z (and therefore conditionally indep, given Z) iff every path from any variable in X to any variable in Y is **blocked** by Z

A path from variable A to variable B is **blocked** by Z if it includes a node such that either

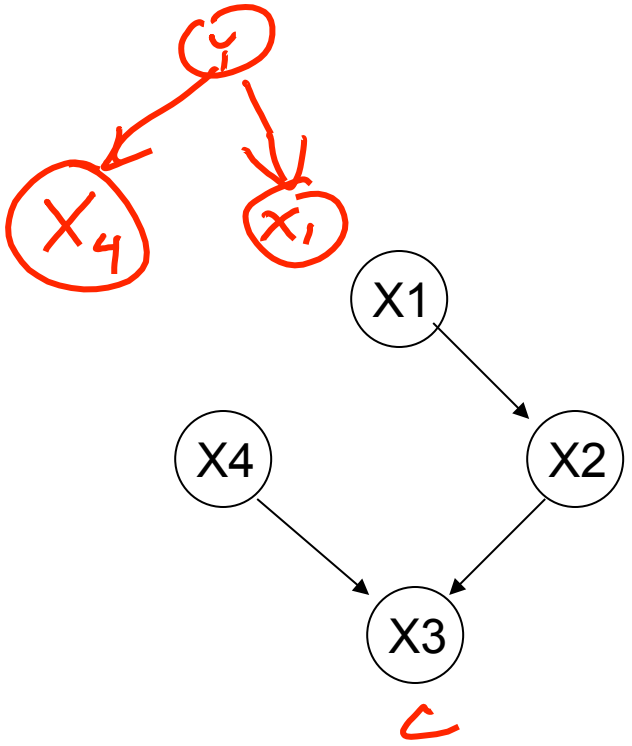
1.arrows on the path meet either head-to-tail or tail-to-tail at the node and this node is in Z 

2.the arrows meet head-to-head at the node, and neither the node, nor any of its descendants, is in Z 

X4 indep of X1 given X3?

X4 indep of X1 given {X3, X2}?

X4 indep of X1 given {}?



X and Y are **D-separated** by Z (and therefore conditionally indep, given Z) iff every path from any variable in X to any variable in Y is **blocked**

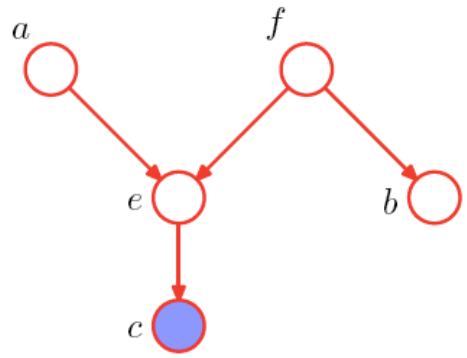
A path from variable A to variable B is **blocked** if it includes a node such that either

1. arrows on the path meet either head-to-tail or tail-to-tail at the node and this node is in Z

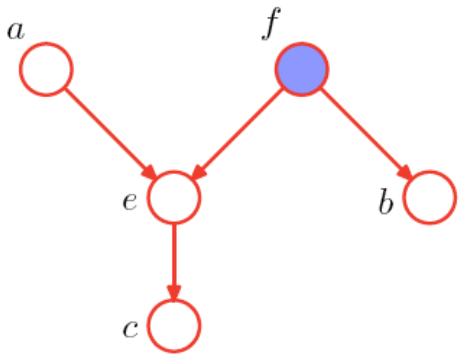
2. or, the arrows meet head-to-head at the node, and neither the node, nor any of its descendants, is in Z

a indep of b given c?

a indep of b given f ?



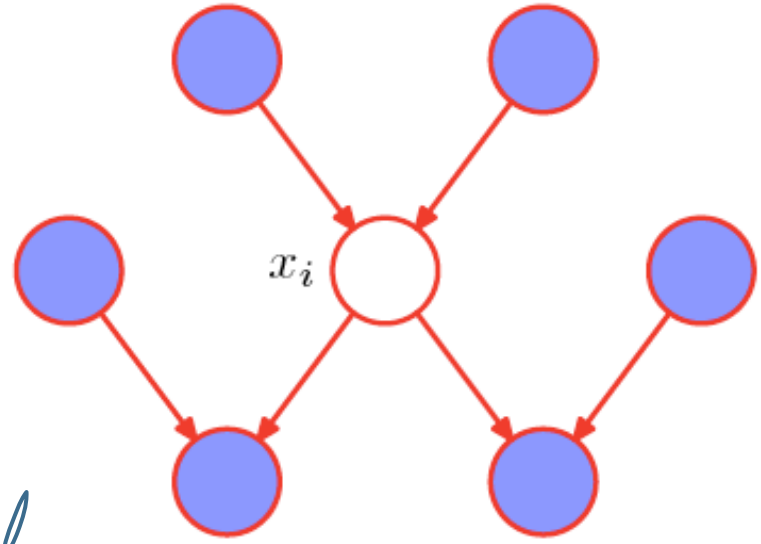
(a)



(b)

Markov Blanket

The Markov blanket of a node $\mathbf{x}_i$ comprises the set of parents, children and co-parents of the node. It has the property that the conditional distribution of $\mathbf{x}_i$, conditioned on all the remaining variables in the graph, is dependent only on the variables in the Markov blanket.



co-parent = other side of X_i's colliders

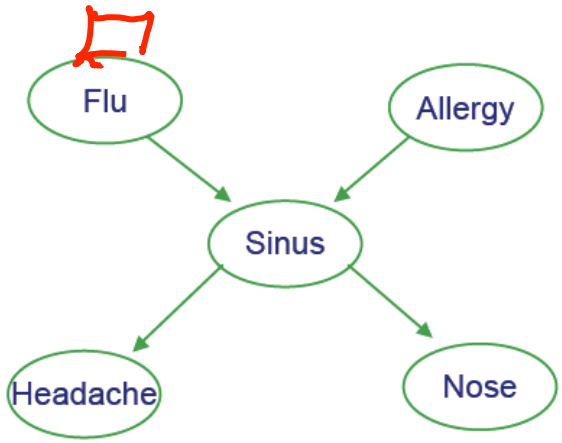
from [Bishop, 8.2]

Inference in Bayes Nets

- In general, intractable (NP-complete)
- For certain cases, tractable
  - Assigning probability to fully observed set of variables
  - Or if just one variable unobserved
  - Or for singly connected graphs (ie., no undirected loops)
    - Variable elimination
    - Belief propagation
- Often use Monte Carlo methods
  - e.g., Generate many samples according to the Bayes Net distribution, then count up the results
  - Gibbs sampling
- Variational methods for tractable approximate solutions

see Graphical Models course 10-708

Prob. of joint assignment: easy



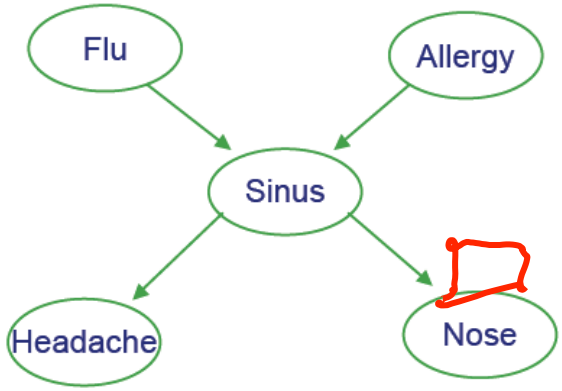
- Suppose we are interested in joint assignment <F=f,A=a,S=s,H=h,N=n>

What is P(f,a,s,h,n)?

$$P(f)\,P(a)\,P(s \mid f, a)\,P(h \mid s)\,P(n \mid s)$$

let's use p(a,b) as shorthand for p(A=a, B=b)

Prob. of marginals: not so easy



- How do we calculate P(N=n) ?

$$P(N=n) = \sum_{f,a,h,s} P(N=n, f, a, h, s)$$

$$f \in F,\ a \in A,\ \ldots$$

we can: 1) sum up the other variables
2) do MCMC sampling

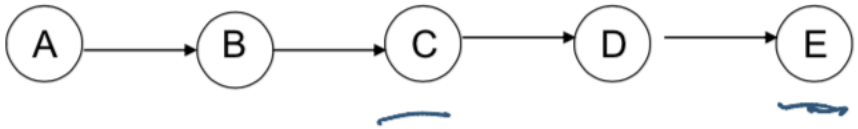
let's use p(a,b) as shorthand for p(A=a, B=b)

Let's do P(C=1)

Prob. of marginals: not so easy

But sometimes the structure of the network allows us to be clever → avoid exponential work

eg., chain



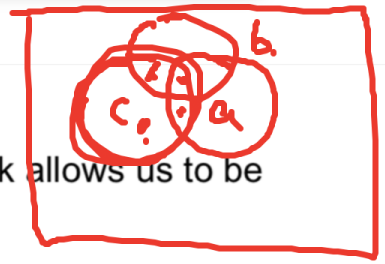
ID: 388-146-648 Stop Share

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save

Let's do P(C=1)

Prob. of marginals: not so easy

But sometimes the structure of the network allows us to be clever → avoid exponential work



eg. chain A → B → C → D → E → 2^k

(a, b, c, d, e)

$\#(C=1) = n$

$\#\text{Total} = 2^n$

$$P(c=1) = \sum_{abde} P(a\,b\,c=1\,d\,e)$$

$$2^2 = \sum_{abde} P(a)P(b|a)P(c|b)P(d|c)P(e|d)$$

$$= \sum_{abde} P(ab)P(c=1|ab)P(de|c=1)$$

$$= \sum_{ab} P(ab)P(c=1|ab)\sum_{de} P(de|c=1)$$

Zoom Group C

From Him Wu to Everyone:
that node is a co-parent?

From Patrick to Everyone:
yes

From Yunjie(Chloe) Song to Ever
not in frequenist

From Patrick to Everyone:
1?

From Ruhao Xin to Everyone:
1?

From Patrick to Everyone:
but since a and c are conditional
given B, do we still need to cons

To: Yuqin Bai (Privately)

Type message here...

Learning of Bayes Nets

- Four categories of learning problems
  - Graph structure may be known/unknown
  - Variable values may be fully observed / partly unobserved

- Easy case: learn parameters for graph structure is *known*, and data is *fully observed*

- Interesting case: graph *known*, data *partly known*

- Gruesome case: graph structure *unknown*, data *partly unobserved*

Learning CPTs from Fully Observed Data

- Example: Consider learning the parameter

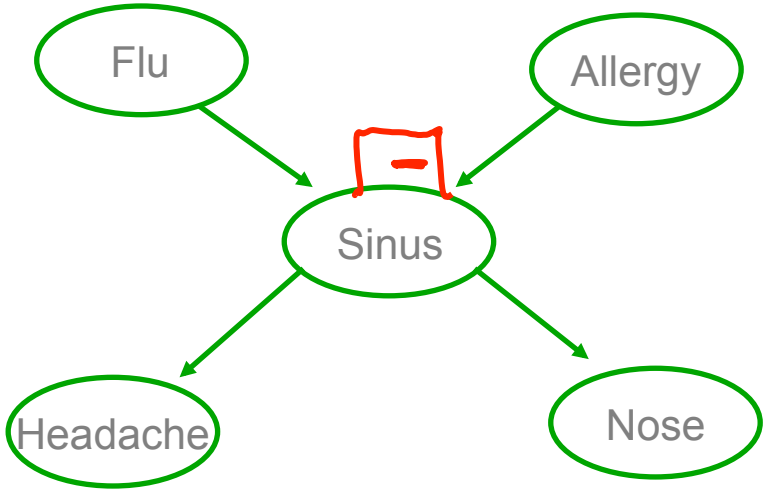
$$\theta_{s|ij} \equiv P(S = 1 | F = i, A = j)$$

- Max Likelihood Estimate is

$$\theta_{s|ij} = \frac{\sum_{k=1}^{K} \delta(f_k = i, a_k = j, s_k = 1)}{\sum_{k=1}^{K} \delta(f_k = i, a_k = j)}$$

k[th] training example

δ(x) = 1 if x=true,
= 0 if x=false



- Remember why?

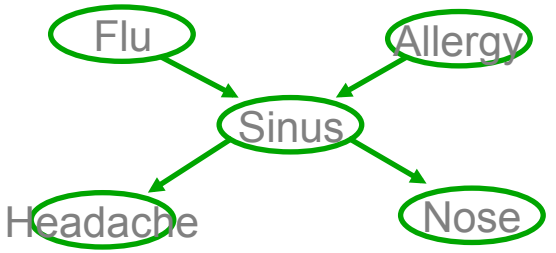
let's use p(a,b) as shorthand for p(A=a, B=b)

MLE estimate of $\theta_{s|ij}$ from fully observed data

- Maximum likelihood estimate

$$\theta \leftarrow \arg\max_{\theta} \log P(data|\theta)$$

- Our case:



$$P(data|\theta) = \prod_{k=1}^{K} P(f_k, a_k, s_k, h_k, n_k)$$

k-th training example

$$P(data|\theta) = \prod_{k=1}^{K} P(f_k)P(a_k)P(s_k|f_k a_k)P(h_k|s_k)P(n_k|s_k)$$

$$\log P(data|\theta) = \sum_{k=1}^{K} \log P(f_k) + \log P(a_k) + \log P(s_k|f_k a_k) + \log P(h_k|s_k) + \log P(n_k|s_k)$$

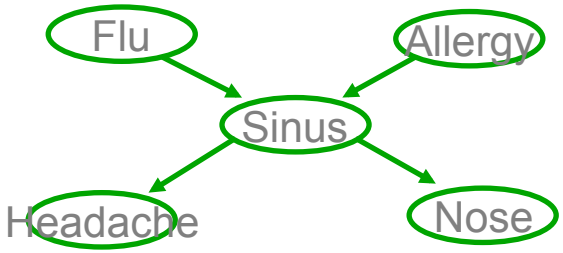
$$\frac{\partial \log P(data|\theta)}{\partial \theta_{s|ij}} = \sum_{k=1}^{K} \frac{\partial \log P(s_k|f_k a_k)}{\partial \theta_{s|ij}}$$

$$\theta_{s|ij} = \frac{\sum_{k=1}^{K} \delta(f_k = i, a_k = j, s_k = 1)}{\sum_{k=1}^{K} \delta(f_k = i, a_k = j)}$$

Estimate θ from partly observed data

- What if FAHN observed, but not S?
- Can't calculate MLE

$$\theta \leftarrow \arg\max_{\theta} \log \prod_{k} P(f_k, a_k, s_k, h_k, n_k|\theta)$$



- Let X be all *observed* variable values (over all examples)
- Let Z be all *unobserved* variable values
- Can't calculate MLE:

$$\theta \leftarrow \arg\max_{\theta} \log P(X, Z|\theta)$$

- WHAT TO DO?

EM Algorithm - Informally

EM is a general procedure for learning from partly observed data

Given observed variables X, unobserved Z (X={F,A,H,N}, Z={S})

Begin with arbitrary choice for parameters θ

Iterate until convergence:

- E Step: estimate the values of unobserved Z, using θ
- M Step: use observed values plus E-step estimates to derive a better θ

Guaranteed to find local maximum.

Each iteration increases $E_{P(Z|X,\theta)}[\log P(X, Z|\theta')]$

EM Algorithm - Precisely

EM is a general procedure for learning from partly observed data

Given observed variables X, unobserved Z (X={F,A,H,N}, Z={S})

Define $Q(\theta'|\theta) = E_{P(Z|X,\theta)}[\log P(X, Z|\theta')]$

next current current M step new

Iterate until convergence:

- E Step: Use X and current θ to calculate P(Z|X,θ)
- M Step: Replace current θ by

$$\theta \leftarrow \arg\max_{\theta'} Q(\theta'|\theta)$$

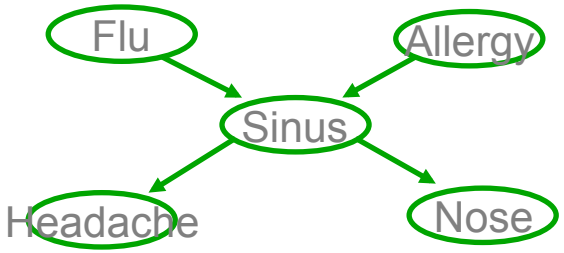
Guaranteed to find local maximum.

Each iteration increases $E_{P(Z|X,\theta)}[\log P(X, Z|\theta')]$

back to this example, say S is Z.

- EM seeks estimate:

$$\theta \leftarrow \arg\max_{\theta} E_{Z|X,\theta}[\log P(X, Z|\theta)]$$



- here, observed X={F,A,H,N}, unobserved Z={S}

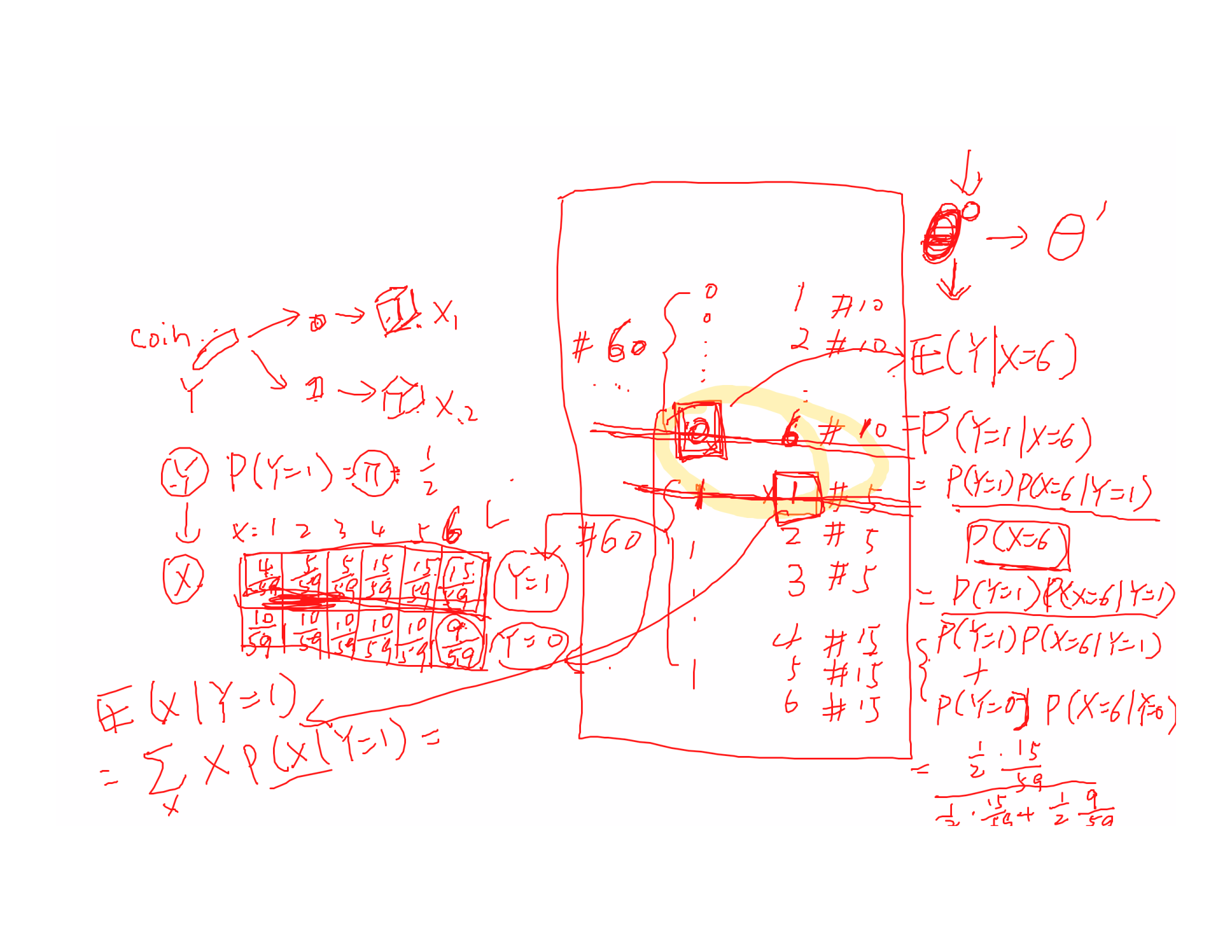
$$\log P(X, Z|\theta) = \sum_{k=1}^{K} \log P(f_k) + \log P(a_k) + \log P(s_k|f_k a_k) + \log P(h_k|s_k) + \log P(n_k|s_k)$$

$$E_{P(Z|X,\theta)} \log P(X, Z|\theta) = \sum_{k=1}^{K} \sum_{i=0}^{1} P(s_k = i|f_k, a_k, h_k, n_k)$$

$$= \frac{P(s_k=0, f, a, h, n)}{P(s_k=0, f a h n) + P(s_k=1, f a h n)}$$

$$[\log P(f_k) + \log P(a_k) + \log P(s_k|f_k a_k) + \log P(h_k|s_k) + \log P(n_k|s_k)]$$

$$E[x] = \sum_i P(x=i)\, i$$


coin
Y
X1
X2
P(Y=1) = π = 1/2
X: 1 2 3 4 5 6
Y=1
Y=0
#60
#10
#5
#15
E(X|Y=1) = Σx x P(x|Y=1) =
θ → θ'
E(Y|X=6)
P(Y=1|X=6)
= P(Y=1)P(X=6|Y=1) / P(X=6)
= P(Y=1)P(X=6|Y=1) / P(Y=1)P(X=6|Y=1) + P(Y=0)P(X=6|Y=0)
= 1/2 · 15/59 / 1/2 · 15/14 + 1/2 · 9/59