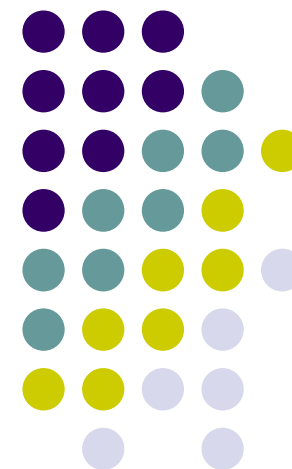


# Advanced Introduction to Machine Learning

10715, Fall 2014

## Introduction & Linear Classifiers



Eric Xing and Barnabas Poczos  
Lecture 1, September 8, 2014

Reading:

© Eric Xing @ CMU, 2014



# Intro to Adv. ML 10-715

- Class webpage:
  - <http://www.cs.cmu.edu/~epxing/Class/10715/>

The screenshot shows a Firefox browser window displaying the class webpage. The page has a navigation menu with links for Home, Description, People, Lectures, Recitations, Homework, and Projects. The main content area features the Carnegie Mellon University logo, the course title 'Advanced Introduction to Machine Learning', and the semester '10-715, Fall 2014'. The instructors listed are Eric Xing and Barnabas Póczos, both from the School of Computer Science at CMU. A green bar highlights the class schedule: Monday and Wednesday, 10:30-11:50am, at location GHC 4211. Another green bar highlights the 'Links' section, which includes links for Grading and Textbooks, Instructors and TAs, Lectures, Recitations Schedule, Course Project, and Homeworks. A third green bar highlights the 'Announcements' section, dated 08/27, which contains information about the class start date, a waiting list, and the class mailing list.

Firefox

10715 Advanced Introduction to Machin... +

www.cs.cmu.edu/~epxing/Class/10715/index.html

Home Description People Lectures Recitations Homework Projects

 **Advanced Introduction to Machine Learning**  
10-715, Fall 2014

[Eric Xing](#), [Barnabas Póczos](#)  
School of Computer Science, Carnegie-Mellon University

**Time: Monday and Wednesday, 10:30-11:50am**  
**Location:** GHC 4211  
**Recitations:** TBA

**Links**

- [Grading and Textbooks](#)
- [Instructors and TAs](#)
- [Lectures](#)
- [Recitations Schedule](#)
- [Course Project](#)
- [Homeworks](#)

**Announcements**

08/27:

- Class begins on Monday 9/8, see you in class!
- IF YOU ARE ON THE WAITING LIST: If the class is fully subscribed, you may still want to consider attending the first few lectures and see how the course develops. We will admit as many students from the waitlist as we can once we know how many registered students drop the course during the first two weeks.
- The class mailing list is [10715-announce@cs](mailto:10715-announce@cs). If you wish to email only the instructors, the email is [10715-instructors@cs](mailto:10715-instructors@cs)
- If you are registered for the course, you have automatically been added to the mail group. If you are for some reason NOT receiving these

# Logistics



- Text book
  - No required book
  - **Reading assignments on class homepage**
  - Optional: David Mackay, **Information Theory, Inference, and Learning Algorithms**
- Mailing Lists:
  - To contact the instructors: [10715-instructors@cs.cmu.edu](mailto:10715-instructors@cs.cmu.edu)
  - Class announcements list: [10715-announce@cs.cmu.edu](mailto:10715-announce@cs.cmu.edu).
- TA:
  - [Kirthivasan Kandasamy, GHC 8015](#)
  - [Veeranjaneyulu Sadhanala, GHC 8005](#)
- Guest Lecturers
  - [Yaoliang Yu](#)
  - [Andrew Wilson](#)
- Class Assistant:
  - [Mallory Deptola, GHC 8001, x8-5527](#)

# Logistics



- 4 homework assignments: 40% of grade
  - Theory exercises
  - Implementation exercises
- **Final project: 40% of grade**
  - Applying machine learning to your research area
    - NLP, IR,, vision, robotics, computational biology ...
  - Outcomes that offer real utility and value
    - Search all the wine bottle labels,
    - An iPhone app for landmark recognition
  - Theoretical and/or algorithmic work
    - a more efficient approximate inference algorithm
    - a new sampling scheme for a non-trivial model ...
  - 3-member team to be formed in the first two weeks, proposal, mid-way report, poster & demo, final report.
- One midterm exams: 20% of grade
  - Theory exercises and/or analysis. Dates already set (no “ticket already booked”, “I am in a conference”, etc. excuse ...)
- Policies ...



# What is Learning

Learning is about seeking a **predictive** and/or **executable** understanding of natural/artificial subjects, phenomena, or activities from ...



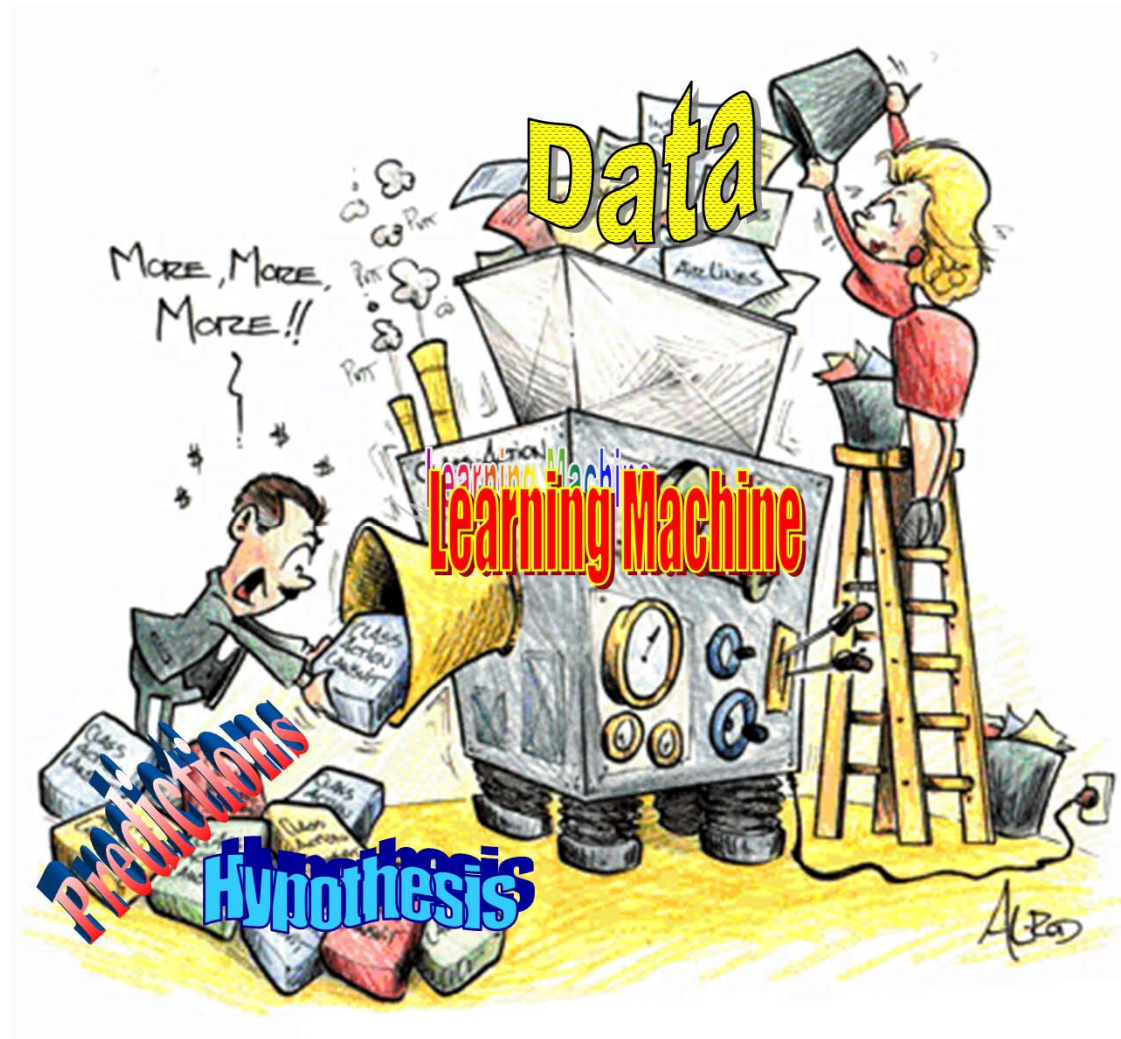
Apoptosis + Medicine

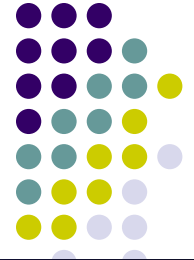
Grammatical rules  
Manufacturing procedures  
Natural laws  
...

Inference:  
what does this mean?  
Any similar article?  
...



# Machine Learning





# What is Machine Learning?

Machine Learning seeks to develop **theories** and **computer systems** for

- representing;
- classifying, clustering and recognizing;
- reasoning under uncertainty;
- predicting;
- and reacting to
- ...

complex, real world data, based on **the system's own experience with data**, and (hopefully) under a **unified model or mathematical framework**, that

- can be formally characterized and analyzed
- can take into account human prior knowledge
- can generalize and adapt across data and domains
- can operate automatically and autonomously
- and can be interpreted and perceived by human.

# Why machine learning?

---



**13 million Wikipedia pages**

**facebook**

**500 million users**

**flickr**

**3.6 billion photos**

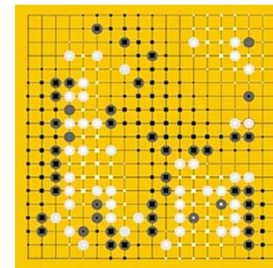
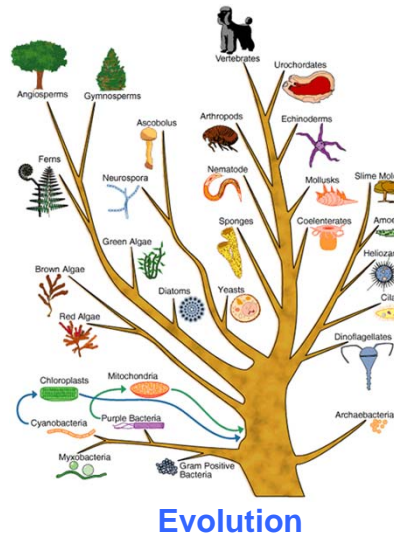
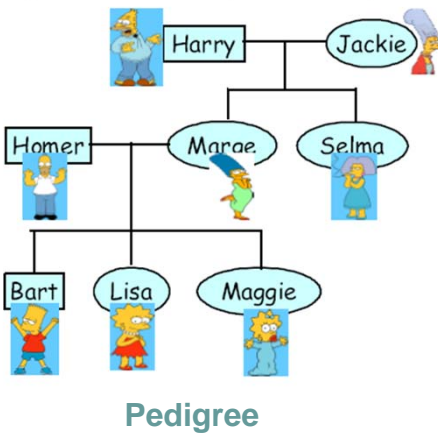
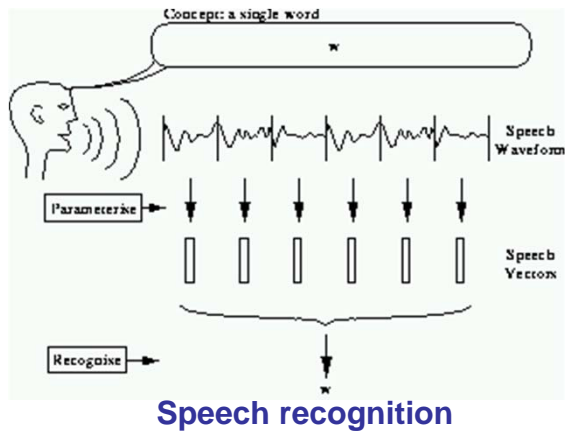
**You Tube**

**24 hours videos uploaded per minute**

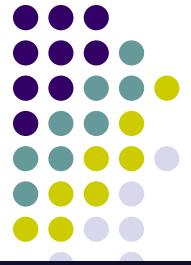




# Machine Learning is Prevalent



# Natural language processing and speech recognition



- Now most pocket **Speech Recognizers** or **Translators** are running on some sort of learning device --- the more you play/use them, the smarter they become!

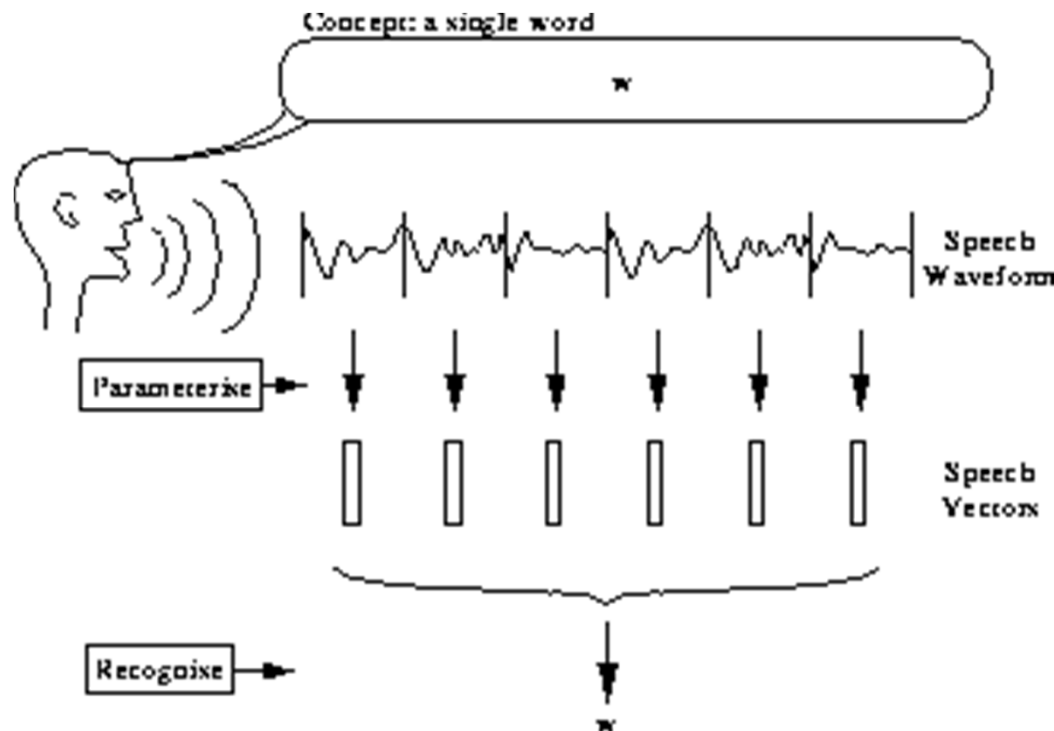
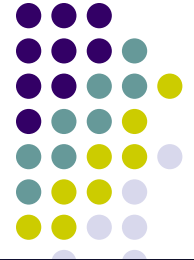
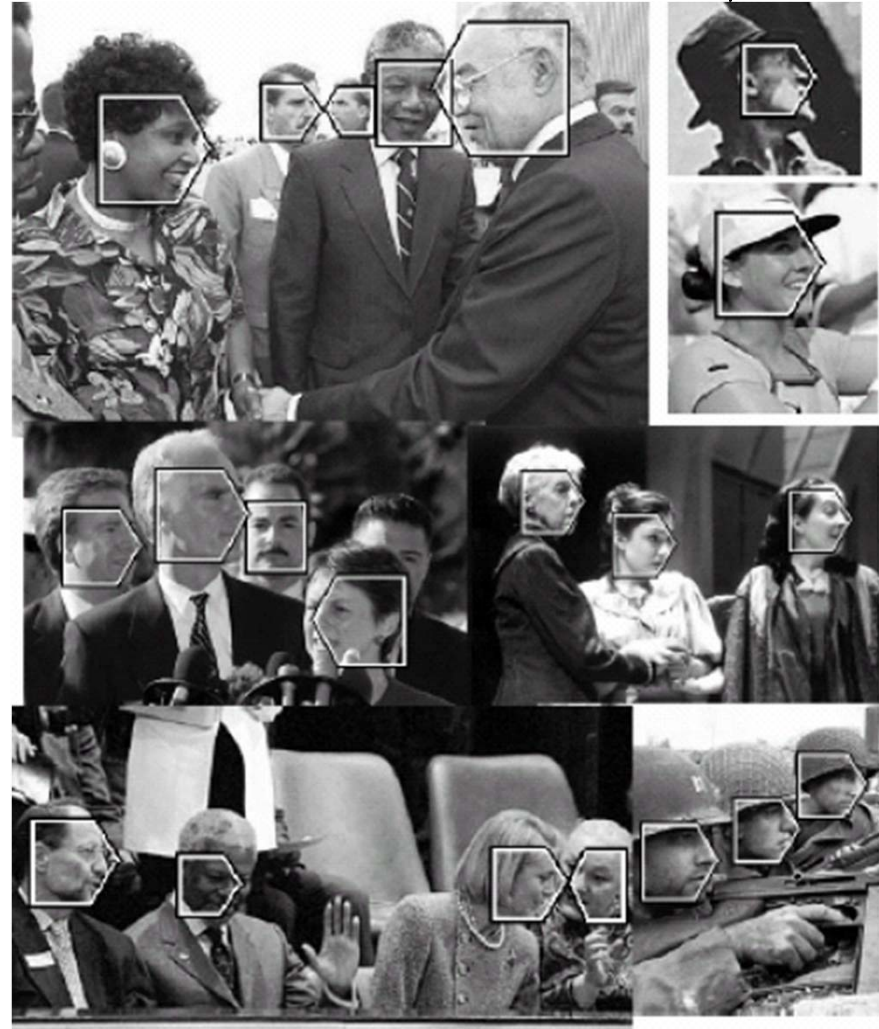


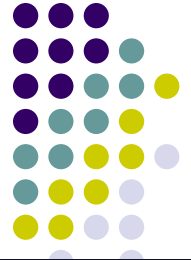
Fig. 1.2 Isolated Word Problem



# Object Recognition

- Behind a security camera, most likely there is a computer that is learning and/or checking!





# Robotic Control

- The **best** helicopter pilot is now a computer!
  - it runs a program that learns how to fly and make acrobatic maneuvers by itself!
  - no taped instructions, joysticks, or things like ...





# Text Mining

- We want:

- Reading, digesting, and categorizing a vast text database is too much for human!



“Arts”	“Budgets”
NEW	MILLION
FILM	TAX
SHOW	PROGRAM
MUSIC	BUDGET
MOVIE	BILLION
PLAY	FEDERAL
MUSICAL	YEAR
BEST	SPENDING
ACTOR	NEW
FIRST	STATE
YORK	PLAN
OPERA	MONEY
THEATER	PROGRAMS
ACTRESS	GOVERNMENT
LOVE	CONGRESS
	TEACHERS
	HIGH
	PUBLIC
	TEACHER
	BENNETT
	MANIGAT
	NAMPHY
	STATE
	PRESIDENT
	ELEMENTARY
	HAITI

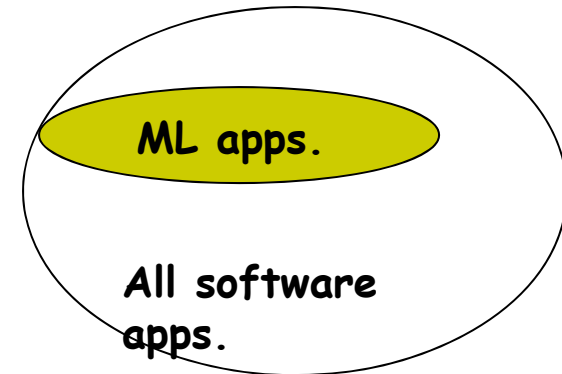
The William Randolph Hearst Foundation will give \$1.25 million to Lincoln Center, Metropolitan Opera Co., New York Philharmonic and Juilliard School. “Our board felt that we had a real opportunity to make a mark on the future of the performing arts with these grants an act every bit as important as our traditional areas of support in health, medical research, education and the social services,” Hearst Foundation President Randolph A. Hearst said Monday in announcing the grants. Lincoln Center’s share will be \$200,000 for its new building, which will house young artists and provide new public facilities. The Metropolitan Opera Co. and New York Philharmonic will receive \$400,000 each. The Juilliard School, where music and the performing arts are taught, will get \$250,000. The Hearst Foundation, a leading supporter of the Lincoln Center Consolidated Corporate Fund, will make its usual annual \$100,000 donation, too.



# Growth of Machine Learning



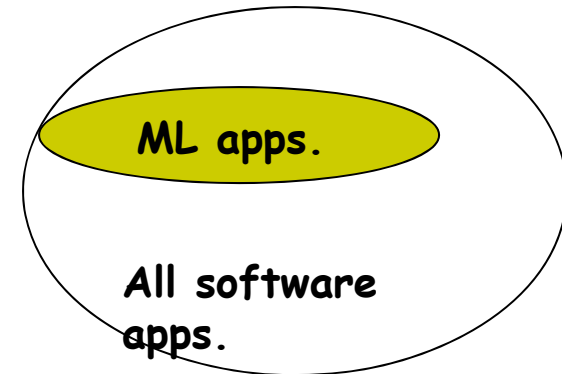
- Machine learning already the preferred approach to
  - Speech recognition, Natural language processing
  - Computer vision
  - Medical outcomes analysis
  - Robot control
  - ...
- This ML niche is growing (why?)





# Growth of Machine Learning

- Machine learning already the preferred approach to
  - Speech recognition, Natural language processing
  - Computer vision
  - Medical outcomes analysis
  - Robot control
  - ...
- This ML niche is growing
  - Improved machine learning algorithms
  - Increased data capture, networking
  - Software too complex to write by hand
  - New sensors / IO devices
  - Demand for **self-customization to user, environment**







# Paradigms of Machine Learning

- Supervised Learning

- Given  $D = \{\mathbf{X}_i, \mathbf{Y}_i\}$ , learn  $f(\cdot) : \mathbf{Y}_i = f(\mathbf{X}_i)$ , s.t.  $D^{\text{new}} = \{\mathbf{X}_j\} \Rightarrow \{\mathbf{Y}_j\}$

- Unsupervised Learning

- Given  $D = \{\mathbf{X}_i\}$ , learn  $f(\cdot) : \mathbf{Y}_i = f(\mathbf{X}_i)$ , s.t.  $D^{\text{new}} = \{\mathbf{X}_j\} \Rightarrow \{\mathbf{Y}_j\}$

- Semi-supervised Learning

- Reinforcement Learning

- Given  $D = \{\text{env, actions, rewards, simulator/trace/real game}\}$

learn  $\text{policy} : e, r \rightarrow a$   
utility :  $a, e \rightarrow r$ , s.t.  $\{\text{env, new real game}\} \Rightarrow a_1, a_2, a_3 \dots$

- Active Learning

- Given  $D \sim G(\cdot)$ , learn  $D^{\text{new}} \sim G'(\cdot)$  and  $f(\cdot)$ , s.t.  $D^{\text{all}} \Rightarrow G'(\cdot), \text{policy}, \{\mathbf{Y}_j\}$



# Machine Learning - Theory

For the learned  $F(; \theta)$

- Consistency (value, pattern, ...)
- Bias versus variance
- Sample complexity
- Learning rate
- Convergence
- Error bound
- Confidence
- Stability
- ...

**PAC Learning Theory**  
(supervised concept learning)

# examples ( $m$ )

error rate ( $\epsilon$ )

representational complexity ( $H$ )

failure probability ( $\delta$ )

$$m \geq \frac{1}{\epsilon} (\ln |H| + \ln(1/\delta))$$



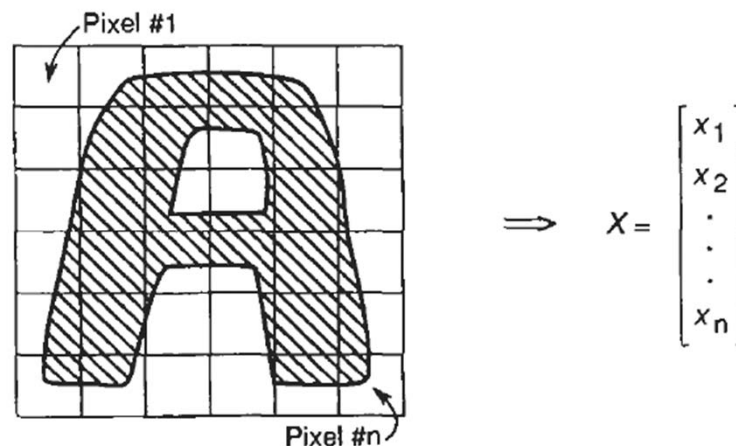
# Elements of Machine Learning

- Here are some important elements to consider before you start:
  - Task:
    - Embedding? Classification? Clustering? Topic extraction? ...
  - Data and other info:
    - Input and output (e.g., continuous, binary, counts, ...)
    - Supervised or unsupervised, of a blend of everything?
    - Prior knowledge? Bias?
  - Models and paradigms:
    - BN? MRF? Regression? SVM?
    - Bayesian/Frequeents ? Parametric/Nonparametric?
  - Objective/Loss function:
    - MLE? MCLE? Max margin?
    - Log loss, hinge loss, square loss? ...
  - Tractability and exactness trade off:
    - Exact inference? MCMC? Variational? Gradient? Greedy search?
    - Online? Batch? Distributed?
  - Evaluation:
    - Visualization? Human interpretability? Perperlexity? Predictive accuracy?
- **It is better to consider one element at a time!**

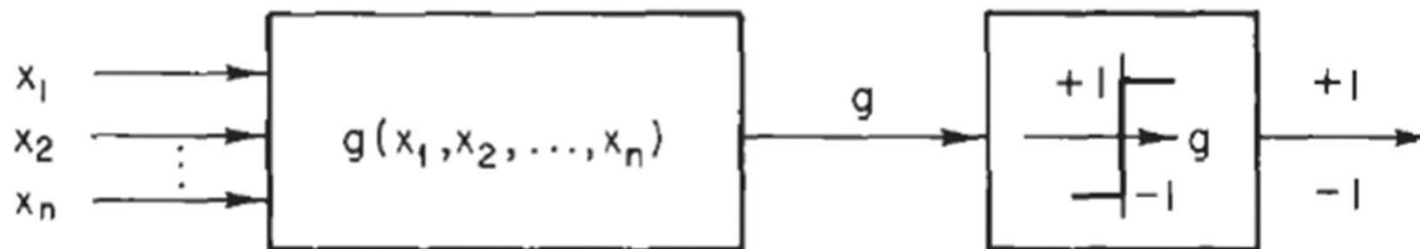


# Classification

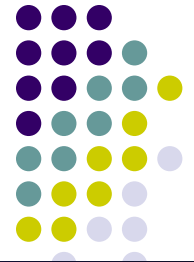
- Representing data:



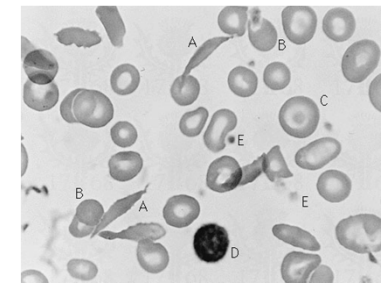
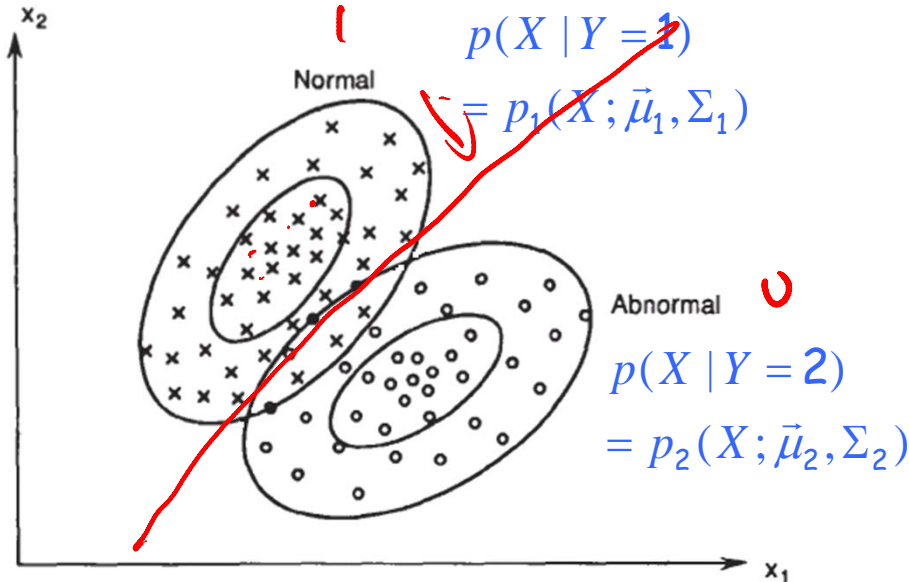
- Hypothesis (classifier)



# Decision-making as dividing a high-dimensional space



- Classification-specific Dist.:  $P(X|Y)$



$$\boxed{P(Y(x))}$$

$$P(Y=1(x)) > P(Y=0(x))$$

$$P(X|Y=1)$$

$$P(X|Y=0)$$

$$P(Y=1)$$

$$P(Y=0)$$

- Class prior (i.e., "weight"):  $P(Y)$



# The Bayes Rule

- What we have just did leads to the following general expression:

$$P(Y | X) = \frac{P(X | Y) p(Y)}{P(X)} = \sum_y P(X, y) = \sum_y P(X|y) P(y)$$

This is Bayes Rule

**Bayes, Thomas (1763)** An essay towards solving a problem in the doctrine of chances. *Philosophical Transactions of the Royal Society of London*, 53:370-418



# The Bayes Decision Rule for Minimum Error



- The *a posteriori* probability of a sample

$$P(Y = i | X) = \frac{p(X | Y = i)P(Y = i)}{p(X)} = \frac{\pi_i p_i(X | Y = i)}{\sum_i \pi_i p_i(X | Y = i)} \equiv q_i(X)$$

- Bayes Test:

$$q_1(x) \geq q_0(x) \quad \frac{p(x|y=1)p(y_1)}{p(x)} \geq \frac{p(x|y=0)p(y_0)}{p(x)}$$

- Likelihood Ratio:

$$l(x) = \frac{p(x|y=1)}{p(x|y=0)} \geq \frac{p(y=0)}{p(y=1)} = 1$$

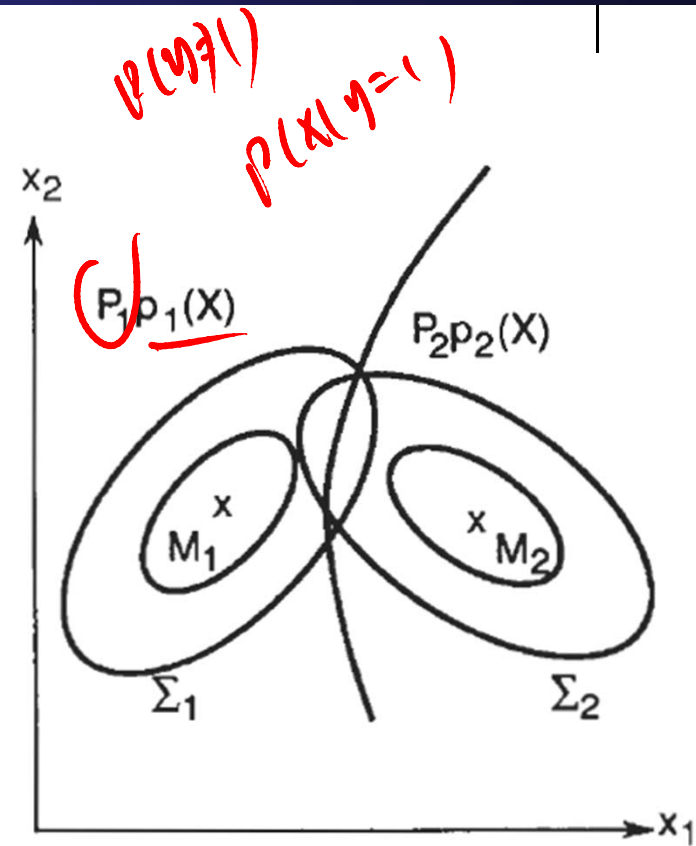
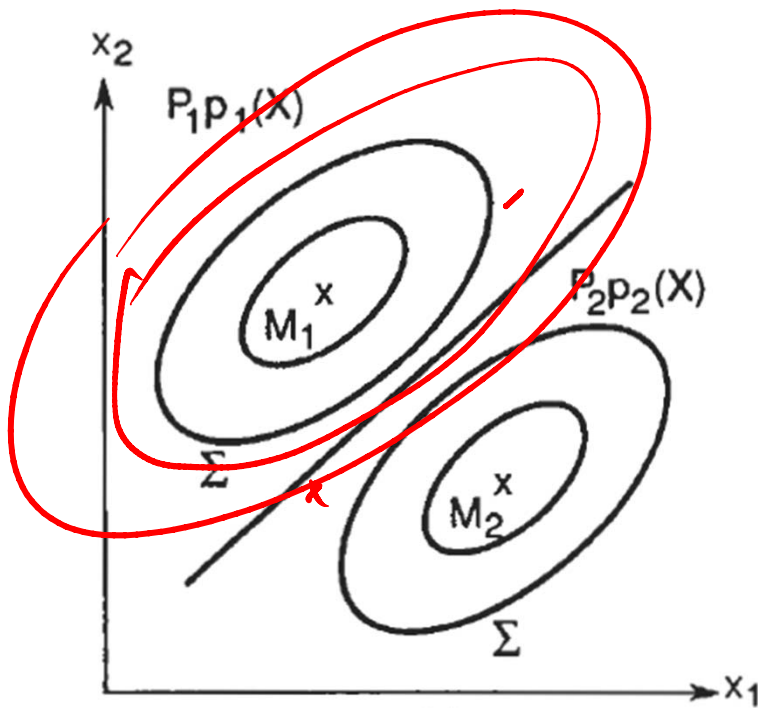
- Discriminant function:

$$h(x) = \log l(x) = \log p(x|y=1) - \log p(x|y=0) \geq$$



# Example of Decision Rules

- When each class is a normal ...



- We can write the decision boundary analytically in some cases ... homework!!





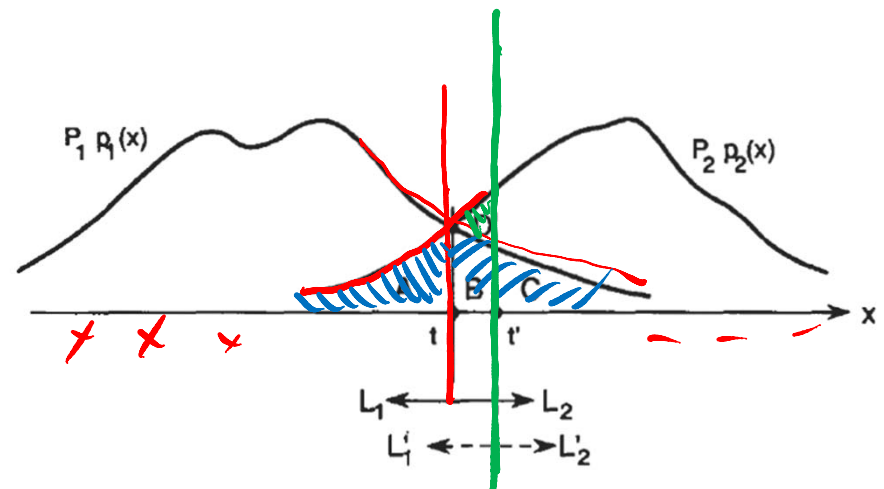
# Bayes Error

- We must calculate the *probability of error*
  - the probability that a sample is assigned to the wrong class
- Given a datum  $X$ , what is the *risk*?

$$r(X) = \min[q_1(X), q_2(X)]$$

- The Bayes error (the expected risk):

$$\begin{aligned}\epsilon &= E[r(X)] = \int r(x)p(x)dx \\ &= \int \min[\pi_1 p_1(x), \pi_2 p_2(x)] dx \\ &= \pi_1 \int_{L_1} p_1(x) dx + \pi_2 \int_{L_2} p_2(x) dx \\ &= \pi_1 \epsilon_1 + \pi_2 \epsilon_2\end{aligned}$$

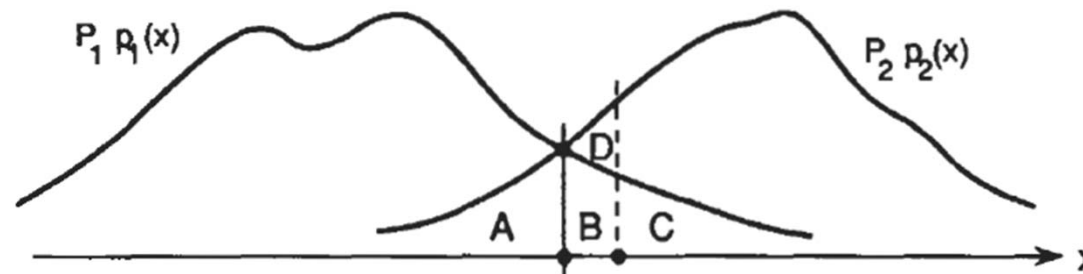




# More on Bayes Error

$\hat{P}(Y)$   
 $\hat{P}(Y|X)$

- Bayes error is the lower bound of probability of classification error



- Bayes classifier is the theoretically best classifier that minimize probability of classification error
- Computing Bayes error is in general a very complex problem. Why?
  - Density estimation:
  - Integrating density function:

$$\epsilon_1 = \int_{\ln(\pi_1/\pi_2)}^{+\infty} p_1(x) dx$$

$$\epsilon_2 = \int_{-\infty}^{\ln(\pi_1/\pi_2)} p_2(x) dx$$



# Learning Classifier

- The decision rule:

$$h(X) = -\ln p_1(X) + \ln p_2(X) \begin{cases} > \ln \frac{\pi_1}{\pi_2} \\ < \ln \frac{\pi_1}{\pi_2} \end{cases}$$

- Learning strategies

- Generative Learning

$$\frac{p(x)}{p(x|y)}$$

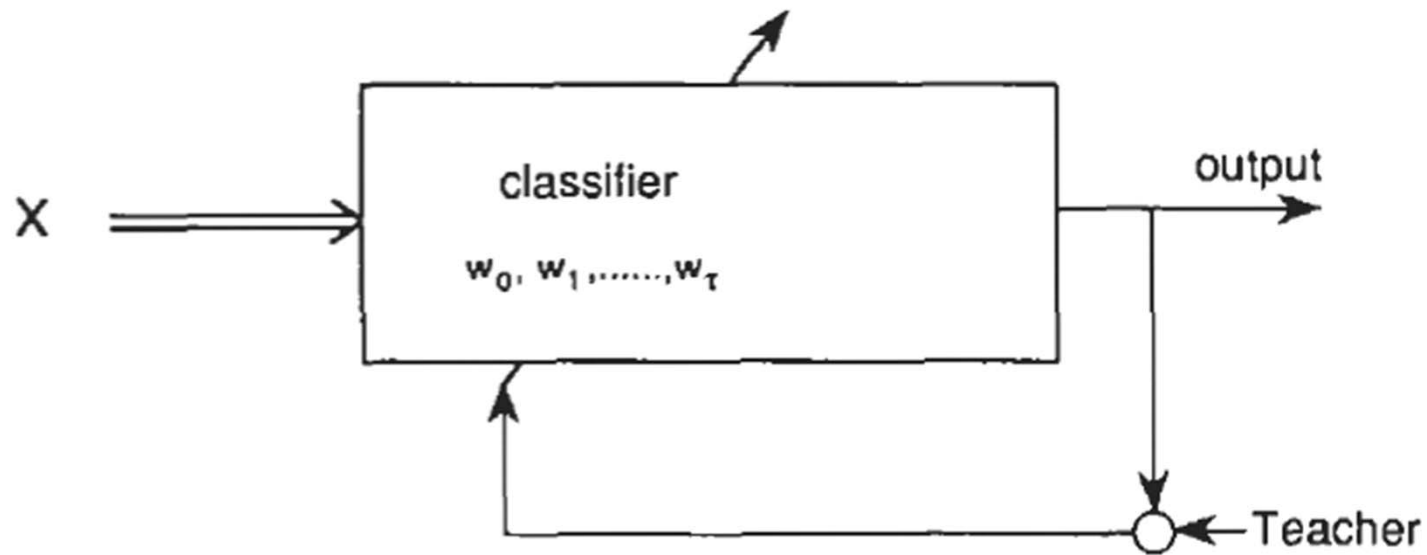
- Discriminative Learning

$$\frac{p(y|x)}{p(y|x) = f(x)}$$

- Instance-based Learning (Store all past experience in memory)

- A special case of nonparametric classifier

# Supervised Learning



- K-Nearest-Neighbor Classifier:  
where the  $h(X)$  is represented by all the data, and by an algorithm



# Learning Bayes Classifier

$$x_n = \begin{bmatrix} x_n^1 \\ \vdots \\ x_n^d \end{bmatrix}$$

$$y_n = [ ]$$

- Training data (discrete case):

X						Y
Sky	Temp	Humid	Wind	Water	Forecst	EnjoySpt
Sunny	Warm	Normal	Strong	Warm	Same	Yes
Sunny	Warm	High	Strong	Warm	Same	Yes
Rainy	Cold	High	Strong	Warm	Change	No
Sunny	Warm	High	Strong	Cool	Change	Yes

$$y_n = 1, 2, 3, \dots$$

$$y_n = \begin{bmatrix} y_n^1 \\ y_n^2 \end{bmatrix}$$

$$\text{if } y_n = 0$$

$$y_n = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\text{if } y_n = 2$$

$$y_n = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

- Learning = estimating  $P(X|Y)$ , and  $P(Y)$

- Classification = using Bayes rule to calculate  $P(Y | X_{\text{new}})$

$$\theta_{ijk} \text{ v } \theta_{i,j,k}$$

$$\begin{bmatrix} \pi_i \\ \tau_{ij} \end{bmatrix}$$

# Parameter learning from *iid* data: The Maximum Likelihood Est.



- Goal: estimate distribution parameters  $\theta$  from a dataset of  $N$  independent, identically distributed (*iid*), fully observed, training cases

$$D = \{x_1, \dots, x_N\}$$

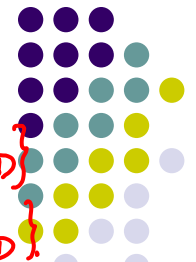
- Maximum likelihood estimation (MLE)
  1. One of the most common estimators
  2. With iid and full-observability assumption, write  $L(\theta)$  as the likelihood of the data:

$$\begin{aligned} L(\theta) &= P(x_1, x_2, \dots, x_N; \theta) \\ &= P(x_1; \theta) P(x_2; \theta), \dots, P(x_N; \theta) \\ &= \prod_{n=1}^N P(x_n; \theta) \end{aligned}$$

3. pick the setting of parameters most likely to have generated the data we saw:

$$\theta^* = \arg \max_{\theta} L(\theta) = \arg \max_{\theta} \log L(\theta)$$

# How hard is it to learn the optimal classifier?



$$P(y|x) \leftarrow P(y) P(x|y)$$

$$X = \{x_1, x_2, \dots, x_D\}$$

$$Y = \{y_1, y_2, \dots, y_D\}$$

- How do we represent these? How many parameters?

- Prior,  $P(Y)$ :

- Suppose  $Y$  is composed of  $k$  classes

$$P(Y) = \prod_{i=1}^D P(y_i | \pi) = \prod_{i=1}^D \prod_{k=1}^K [\pi_k]^{y_i^k} = \prod_{k=1,2} [\pi_k]^{N_k}$$

- Likelihood,  $P(X|Y)$ :

$$= \prod_{k=1,2} [\pi_k]^{N_k}$$

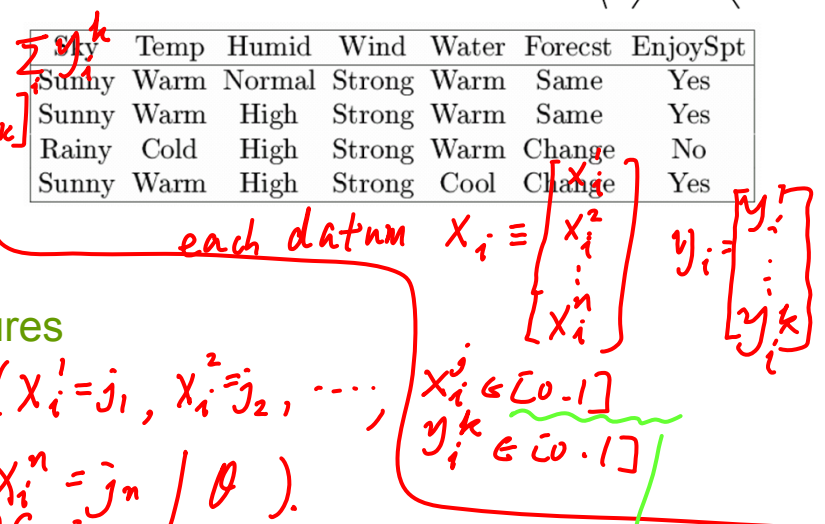
- Suppose  $X$  is composed of  $n$  binary features

$$P(X) = \prod_{i=1}^D P(x_i | \theta) = \prod_{i=1}^D \prod_{j_1 \in \{0,1\}} \prod_{j_2 \in \{0,1\}} \dots \prod_{j_n \in \{0,1\}} P(x_i^1 = j_1, x_i^2 = j_2, \dots, x_i^n = j_n | \theta)$$

$$= \prod_{i=1}^D \prod_{j_1 \in \{0,1\}} \prod_{j_2 \in \{0,1\}} \dots \prod_{j_n \in \{0,1\}} [\theta_{j_1, \dots, j_n}]^{\delta(x_i^1, j_1) \delta(x_i^2, j_2) \dots \delta(x_i^n, j_n)}$$

$$= \prod_{j_1} \prod_{j_2} \dots \prod_{j_n} [\theta_{j_1, \dots, j_n}]^{N_{j_1, \dots, j_n}}$$

	X					Y
	Temp	Humid	Wind	Water	Forest	EnjoySpt
Sunny	Warm	Normal	Strong	Warm	Same	Yes
Sunny	Warm	High	Strong	Warm	Same	Yes
Rainy	Cold	High	Strong	Warm	Change	No
Sunny	Warm	High	Strong	Cool	Change	Yes



this can be a larger set of  $\{1, 2, \dots, J\}$

Complex model! High variance with limited data!!!

we've got to know in advance all  $\theta_{j_1, \dots, j_n}$  for  $\forall [j_1, \dots, j_n]$ .  
 how big is this set?  $\Rightarrow 2^n - 1$  for binary features, and  $J^n - 1$  for  $J$ -nary features.



# Conditional Independence

- $X$  is **conditionally independent** of  $Y$  given  $Z$ , if the probability distribution governing  $X$  is independent of the value of  $Y$ , given the value of  $Z$

$$(\forall i, j, k) P(X = i | Y = j, Z = k) = P(X = i | Z = k)$$

Which we often write

$$P(X | Y, Z) = P(X | Z)$$

- e.g.,

$$P(\textit{Thunder} | \textit{Rain}, \textit{Lightning}) = P(\textit{Thunder} | \textit{Lightning})$$

- Equivalent to:

$$P(X, Y | Z) = P(X | Z)P(Y | Z)$$





# The Naïve Bayes assumption

- Naïve Bayes assumption:
  - Features are conditionally independent given class:

$$\begin{aligned} P(X_1, X_2|Y) &= P(X_1|X_2, Y)P(X_2|Y) \\ &= P(X_1|Y)P(X_2|Y) \end{aligned}$$

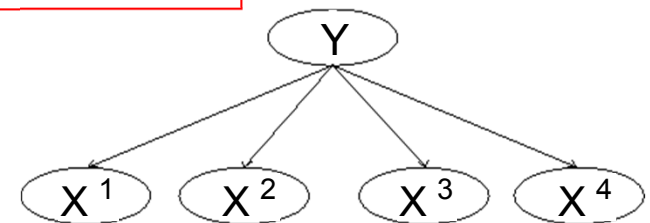
- More generally:

$$P(X^1 \dots X^n | Y) = \prod_i P(X^i | Y)$$

$\Theta_{1..n}$

- How many parameters now?
  - Suppose  $X$  is composed of  $m$  binary features

$2^n - 1 \Rightarrow n$





# The Naïve Bayes Classifier

- Given:
  - Prior  $P(Y)$
  - $m$  conditionally independent features  $\mathbf{X}$  given the class  $Y$
  - For each  $X_n$ , we have likelihood  $P(X_n|Y)$

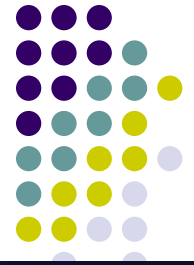
*Handwritten red text:*  $P(x^i|y) \propto \theta_i^{n_i}$

- Decision rule:

$$\begin{aligned} y^* = h_{NB}(\mathbf{x}) &= \arg \max_y P(y) \underbrace{P(x^1, \dots, x^m | y)} \\ &= \arg \max_y P(y) \prod_i P(x^i | y) \end{aligned}$$

- If assumption holds, NB is optimal classifier!

# Gaussian Discriminative Analysis

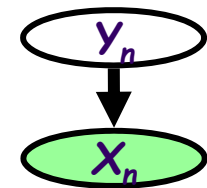


- learning  $f: X \rightarrow Y$ , where
  - $X$  is a vector of real-valued features,  $\mathbf{X}_n = \langle X_n^1, \dots, X_n^m \rangle$
  - $Y$  is an indicator vector
- What does that imply about the form of  $P(Y|X)$ ?
  - The joint probability of a datum and its label is:

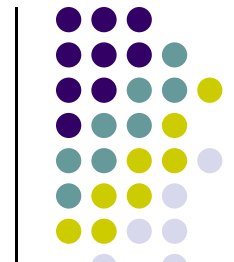
$$\begin{aligned} p(\mathbf{x}_n, y_n^k = 1 | \mu, \sigma) &= p(y_n^k = 1) \times p(\mathbf{x}_n | y_n^k = 1, \mu, \Sigma) \\ &= \pi_k \frac{1}{(2\pi|\Sigma|)^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}_n - \bar{\mu}_k)^T \Sigma^{-1}(\mathbf{x}_n - \bar{\mu}_k)\right\} \end{aligned}$$

- Given a datum  $\mathbf{x}_n$ , we predict its label using the conditional probability of the label given the datum:

$$p(y_n^k = 1 | \mathbf{x}_n, \mu, \Sigma) = \frac{\pi_k \frac{1}{(2\pi|\Sigma|)^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}_n - \bar{\mu}_k)^T \Sigma^{-1}(\mathbf{x}_n - \bar{\mu}_k)\right\}}{\sum_{k'} \pi_{k'} \frac{1}{(2\pi|\Sigma|)^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}_n - \bar{\mu}_{k'})^T \Sigma^{-1}(\mathbf{x}_n - \bar{\mu}_{k'})\right\}}$$



# The A Gaussian Discriminative Naïve Bayes Classifier



- When  $\mathbf{X}$  is multivariate-Gaussian vector:
  - The joint probability of a datum and its label is:

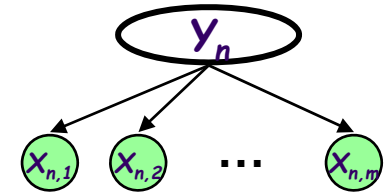
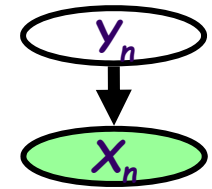
$$\begin{aligned} p(\mathbf{x}_n, y_n^k = 1 | \bar{\boldsymbol{\mu}}, \Sigma) &= p(y_n^k = 1) \times p(\mathbf{x}_n | y_n^k = 1, \bar{\boldsymbol{\mu}}, \Sigma) \\ &= \pi_k \frac{1}{(2\pi|\Sigma|)^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}_n - \bar{\boldsymbol{\mu}}_k)^T \Sigma^{-1}(\mathbf{x}_n - \bar{\boldsymbol{\mu}}_k)\right\} \end{aligned}$$

- The naïve Bayes simplification

$$\begin{aligned} p(\mathbf{x}_n, y_n^k = 1 | \boldsymbol{\mu}, \boldsymbol{\sigma}) &= p(y_n^k = 1) \times \prod_j p(x_n^j | y_n^k = 1, \mu_k^j, \sigma_k^j) \\ &= \pi_k \prod_j \frac{1}{\sqrt{2\pi}\sigma_k^j} \exp\left\{-\frac{1}{2}\left(\frac{x_n^j - \mu_k^j}{\sigma_k^j}\right)^2\right\} \end{aligned}$$

- More generally:  $p(\mathbf{x}_n, y_n | \boldsymbol{\eta}, \boldsymbol{\pi}) = p(y_n | \boldsymbol{\pi}) \times \prod_{j=1}^m p(x_n^j | y_n, \boldsymbol{\eta})$

- Where  $p(. | .)$  is an arbitrary conditional (discrete or continuous) 1-D density





# The predictive distribution

- Understanding the predictive distribution

$$p(y_n^k = \mathbf{1} | x_n, \bar{\mu}, \Sigma, \pi) = \frac{p(y_n^k = \mathbf{1}, x_n | \bar{\mu}, \Sigma, \pi)}{p(x_n | \bar{\mu}, \Sigma)} = \frac{\pi_k N(x_n, | \mu_k, \Sigma_k)}{\sum_{k'} \pi_{k'} N(x_n, | \mu_{k'}, \Sigma_{k'})}$$

$\mu_i$   
 $\sigma_i$   
 $w_i$   
 $w = \sum w_i x_i$   
 $w^T x$   
 $*$

- Under naïve Bayes assumption:

$$p(y_n^k = \mathbf{1} | x_n, \bar{\mu}, \Sigma, \pi) = \frac{\pi_k \exp\left\{-\sum_j \left(\frac{1}{2} \left(\frac{x_n^j - \mu_k^j}{\sigma_k^j}\right)^2 - \log \sigma_k^j - C\right)\right\}}{\sum_{k'} \pi_{k'} \exp\left\{-\sum_j \left(\frac{1}{2} \left(\frac{x_n^j - \mu_{k'}^j}{\sigma_{k'}^j}\right)^2 - \log \sigma_{k'}^j - C\right)\right\}}$$

\*\*

- For two class (i.e.,  $K=2$ ), and when the two classes has the same variance, \*\* turns out to be a **logistic function**

$$p(y_n^1 = \mathbf{1} | x_n) = \frac{1}{1 + \frac{\pi_2 \exp\left\{-\sum_j \left(\frac{1}{2\sigma_j^2} (x_n^j - \mu_2^j)^2 - \log \sigma_j - C\right)\right\}}{\pi_1 \exp\left\{-\sum_j \left(\frac{1}{2\sigma_j^2} (x_n^j - \mu_1^j)^2 - \log \sigma_j - C\right)\right\}}} = \frac{1}{1 + \exp\left\{-\sum_j \left(x_n^j \left[\frac{1}{\sigma_j^2} (\mu_1^j - \mu_2^j)\right] + \frac{1}{\sigma_j^2} ([\mu_1^j]^2 - [\mu_2^j]^2)\right) + \log \frac{(1-\pi_1)}{\pi_1}\right\}}$$

$w_1 x_1$   
 $w_2 x_2$   
 $w_3 x_3$   
 $w^T x$

# The decision boundary

$p(y)$   
 $p(x|y)$

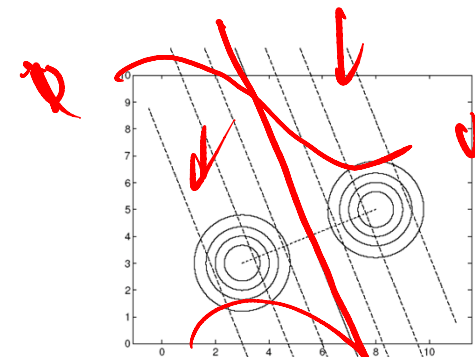


- The predictive distribution

$$p(y_n^1 = \mathbf{1} | x_n) = \frac{1}{1 + \exp\left\{-\sum_{j=1}^M \theta_j x_n^j - \theta_0\right\}} = \frac{1}{1 + e^{-\theta^T x_n}}$$

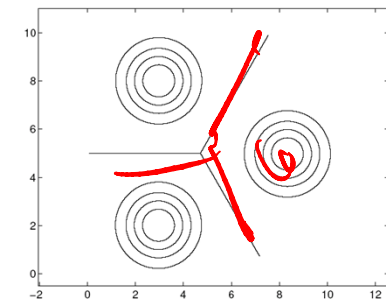
- The Bayes decision rule:

$$\ln \frac{p(y_n^1 = \mathbf{1} | x_n)}{p(y_n^2 = \mathbf{1} | x_n)} = \ln \left( \frac{\frac{1}{1 + e^{-\theta^T x_n}}}{\frac{e^{-\theta^T x_n}}{1 + e^{-\theta^T x_n}}} \right) = \theta^T x_n$$

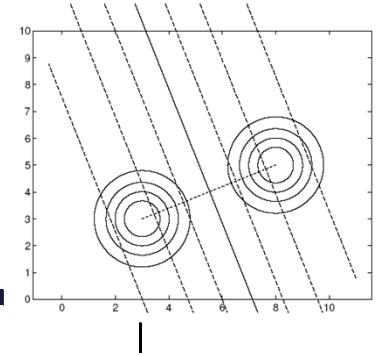


- For multiple class (i.e.,  $K > 2$ ), \* correspond to a softmax function

$$p(y_n^k = \mathbf{1} | x_n) = \frac{e^{-\theta_k^T x_n}}{\sum_j e^{-\theta_j^T x_n}}$$



# Summary: The Naïve Bayes Algorithm



- Train Naïve Bayes (examples)
  - for each\* value  $y_k$
  - estimate  $\pi_k \equiv P(Y = y_k)$
  - for each\* value  $x_{ij}$  of each attribute  $X_i$
  - estimate  $\theta_{ijk} \equiv P(X^i = x_{ij} | Y = y_k)$

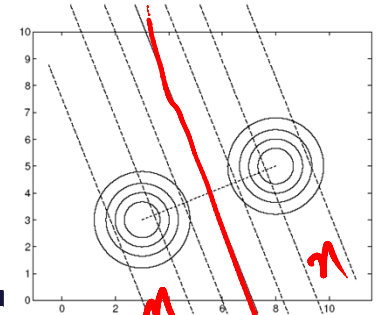
- Classify ( $X_{new}$ )

$$Y^{new} \leftarrow \arg \max_{y_k} P(Y = y_k) \prod_i P(X^i = x_{ij} | Y = y_k)$$

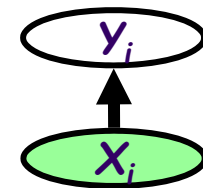
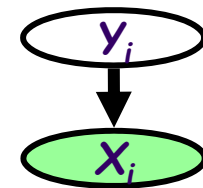
$$Y^{new} \leftarrow \arg \max_{y_k} \pi_k \prod_i \theta_{ijk}$$

# Generative vs. Discriminative Classifiers

$$u(z^n) \rightarrow u(x) \\ u(x^n)$$



- Goal: Wish to learn  $f: X \rightarrow Y$ , e.g.,  $P(Y|X)$
- Generative classifiers (e.g., Naïve Bayes):
  - Assume some functional form for  $P(X|Y)$ ,  $P(Y)$   
This is a '**generative**' model of the data!
  - Estimate parameters of  $P(X|Y)$ ,  $P(Y)$  directly from training data
  - Use Bayes rule to calculate  $P(Y|X=x)$
- Discriminative classifiers:
  - Directly assume some functional form for  $P(Y|X)$   
This is a '**discriminative**' model of the data!
  - Estimate parameters of  $P(Y|X)$  directly from training data



~ sigmoid fun





# Recall the NB predictive distribution

- Understanding the predictive distribution

$$p(y_n^k = \mathbf{1} | x_n, \bar{\mu}, \Sigma, \pi) = \frac{p(y_n^k = \mathbf{1}, x_n | \bar{\mu}, \Sigma, \pi)}{p(x_n | \bar{\mu}, \Sigma)} = \frac{\pi_k N(x_n, | \mu_k, \Sigma_k)}{\sum_{k'} \pi_{k'} N(x_n, | \mu_{k'}, \Sigma_{k'})} \quad *$$

- Under naïve Bayes assumption:

$$p(y_n^k = \mathbf{1} | x_n, \bar{\mu}, \Sigma, \pi) = \frac{\pi_k \exp\left\{-\sum_j \left(\frac{1}{2} \left(\frac{x_n^j - \mu_k^j}{\sigma_k^j}\right)^2 - \log \sigma_k^j - C\right)\right\}}{\sum_{k'} \pi_{k'} \exp\left\{-\sum_j \left(\frac{1}{2} \left(\frac{x_n^j - \mu_{k'}^j}{\sigma_{k'}^j}\right)^2 - \log \sigma_{k'}^j - C\right)\right\}} \quad **$$

- For two class (i.e.,  $K=2$ ), and when the two classes has the same variance, \*\* turns out to be a **logistic function**

$$\begin{aligned} p(y_n^1 = \mathbf{1} | x_n) &= \frac{1}{1 + \frac{\pi_2 \exp\left\{-\sum_j \left(\frac{1}{2\sigma_j^2} (x_n^j - \mu_2^j)^2 - \log \sigma_j - C\right)\right\}}{\pi_1 \exp\left\{-\sum_j \left(\frac{1}{2\sigma_j^2} (x_n^j - \mu_1^j)^2 - \log \sigma_j - C\right)\right\}}} = \frac{1}{1 + \exp\left\{-\sum_j \left(x_n^j \frac{1}{\sigma_j^2} (\mu_1^j - \mu_2^j) + \frac{1}{\sigma_j^2} ([\mu_1^j]^2 - [\mu_2^j]^2)\right) + \log \frac{(1-\pi_1)}{\pi_1}\right\}} \\ &= \frac{1}{1 + e^{-\theta^T x_n}} \end{aligned}$$



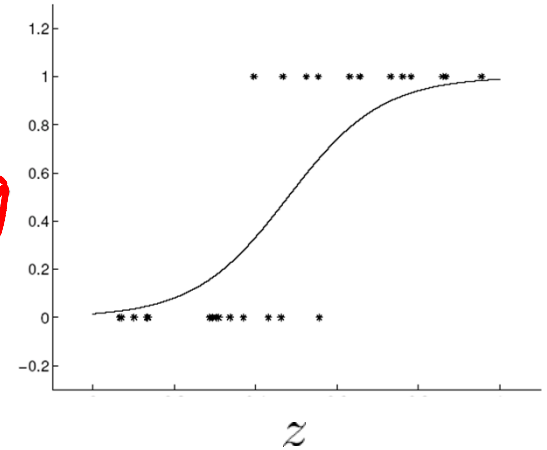
# The logistic function

~~$f(x)$~~



$$g(z) = \frac{1}{1 + e^{-z}}$$

$g$



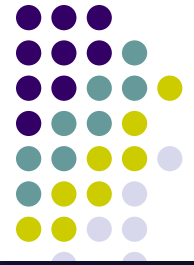
$$z \rightarrow \infty \quad g \rightarrow 1$$

$$z \rightarrow -\infty \quad g \rightarrow 0$$

$$\begin{aligned} \frac{dg}{dz} &= \frac{-1}{(1 + e^{-z})^2} e^{-z} \cdot (-1) \\ &= \frac{1}{1 + e^{-z}} \cdot \frac{e^{-z}}{1 + e^{-z}} = g(1-g) \end{aligned}$$

# Logistic regression (sigmoid classifier)

$y = \begin{cases} 0 \\ 1 \end{cases}$

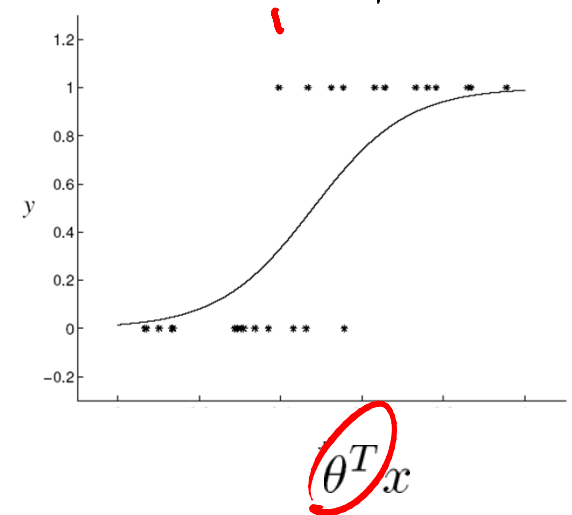


- The condition distribution: a Bernoulli

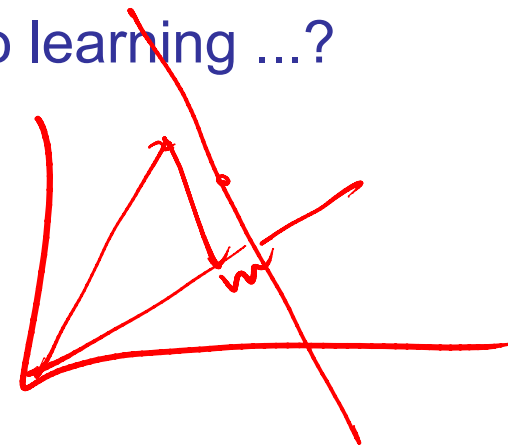
$$p(y|x) = \mu(x)^y (1 - \mu(x))^{1-y}$$

where  $\mu$  is a logistic function

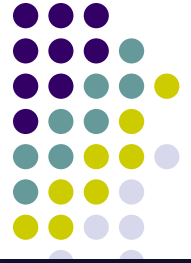
$$\mu(x) = \frac{1}{1 + e^{-\theta^T x}} = p(y = 1 | x)$$



- In this case, learning  $p(y|x)$  amounts to learning ...?
- What is the difference to NB?



# Training Logistic Regression: MCLE



- Estimate parameters  $\theta = \langle \theta_0, \theta_1, \dots, \theta_m \rangle$  to maximize the **conditional likelihood** of training data

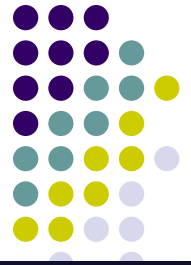
- Training data  $\mathcal{D} = \{(x_1, y_1), \dots, (x_N, y_N)\}$

- Data likelihood =  $\prod_{i=1}^N P(x_i, y_i; \theta)$

- Data conditional likelihood =  $\prod_{i=1}^N P(y_i | x_i; \theta)$

$$\theta = \arg \max_{\theta} \ln \prod_i P(y_i | x_i; \theta)$$

# Expressing Conditional Log Likelihood



$$l(\theta) \equiv \ln \prod_i P(y_i|x_i; \theta) = \sum_i \ln P(y_i|x_i; \theta)$$

- Recall the logistic function:  $\mu = \frac{1}{1 + e^{-\theta^T x}}$

and conditional likelihood:  $P(y|x) = \mu(x)^y(1 - \mu(x))^{1-y}$

$$\begin{aligned} l(\theta) = \sum_i \ln P(y_i|x_i; \theta) &= \sum_i y_i \ln u(x_i) + (1 - y_i) \ln(1 - \mu(x_i)) \\ &= \sum_i y_i \ln \frac{u(x_i)}{1 - \mu(x_i)} + \ln(1 - \mu(x_i)) \\ &= \sum_i y_i \theta^T x_i - \theta^T x_i + \ln(1 + e^{-\theta^T x_i})^{-1} \\ &= \sum_i (y_i - 1) \theta^T x_i + \ln(1 + e^{-\theta^T x_i})^{-1} \end{aligned}$$

# Maximizing Conditional Log Likelihood

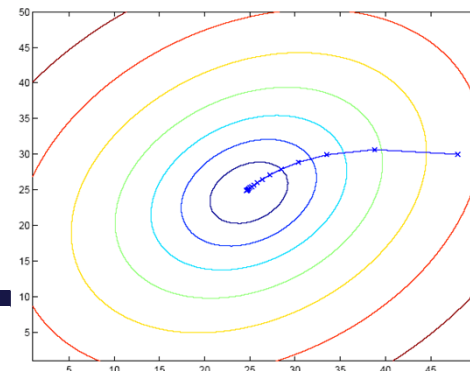
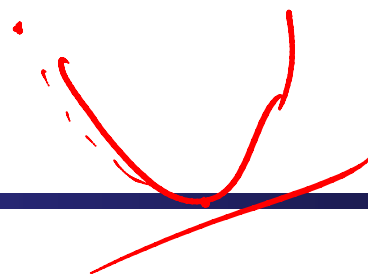


- The objective:

$$\begin{aligned} l(\theta) &= \ln \prod_i P(y_i | x_i; \theta) \\ &= \sum_i (y_i - 1) \theta^T x_i + \ln(1 + e^{-\theta^T x_i})^{-1} \end{aligned}$$

- Good news:  $l(\theta)$  is concave function of  $\theta$
- Bad news: no closed-form solution to maximize  $l(\theta)$

# Gradient Ascent



$$\begin{aligned}
 l(\theta) &= \ln \prod_i P(y_i | x_i; \theta) \\
 &= \sum_i (y_i - 1) \theta^T x_i + \ln(1 + e^{-\theta^T x_i})^{-1} = \sum_i (y_i - 1) \theta^T x_i + \ln \mu(\theta^T x_i)
 \end{aligned}$$

- Property of sigmoid function:

$$\mu = \frac{1}{1 + e^{-t}} \qquad \frac{d\mu}{dt} = \mu(1 - \mu)$$

- The gradient:

$$\frac{\partial l(\theta)}{\partial \theta_j} = \sum_i (y_i - 1) x_i^j + \frac{1}{\mu} \mu(1 - \mu) x_i^j = \sum_i (y_i - \mu) x_i^j$$

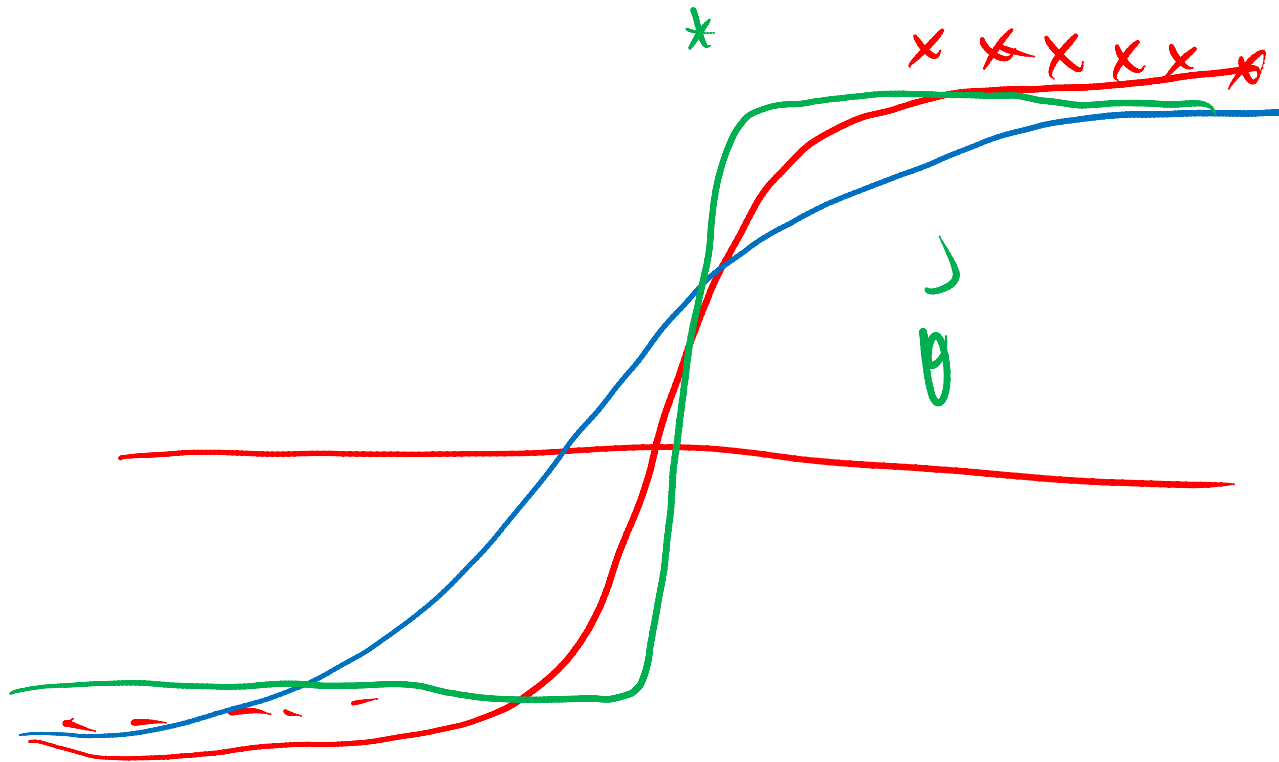
The gradient ascent algorithm iterate until change  $< \epsilon$

For all  $i$ ,  $\theta_j \leftarrow \theta_j + \eta \sum_i (y_i - P(y_i = 1 | x_i; \theta)) x_i^j$

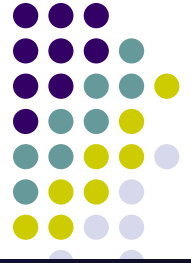
repeat

$\theta^j$   
→

# Overfitting ...







# How about MAP?

- It is very common to use *regularized* maximum likelihood.
- One common approach is to define priors on  $\theta$ 
  - – Normal distribution, zero mean, identity covariance
  - Helps avoid very large weights and overfitting

$$P(y|x; \theta) = \mu(\theta^T x)^y (1 - \mu(\theta^T x))^{1-y}$$

$$P(\theta) = \text{Normal}(0, \lambda^{-1}I)$$

$$l'(\theta) = \sum_i (y_i - 1)\theta^T x_i - \ln \mu(\theta^T x_i) - \frac{\lambda}{2}\theta^T \theta$$

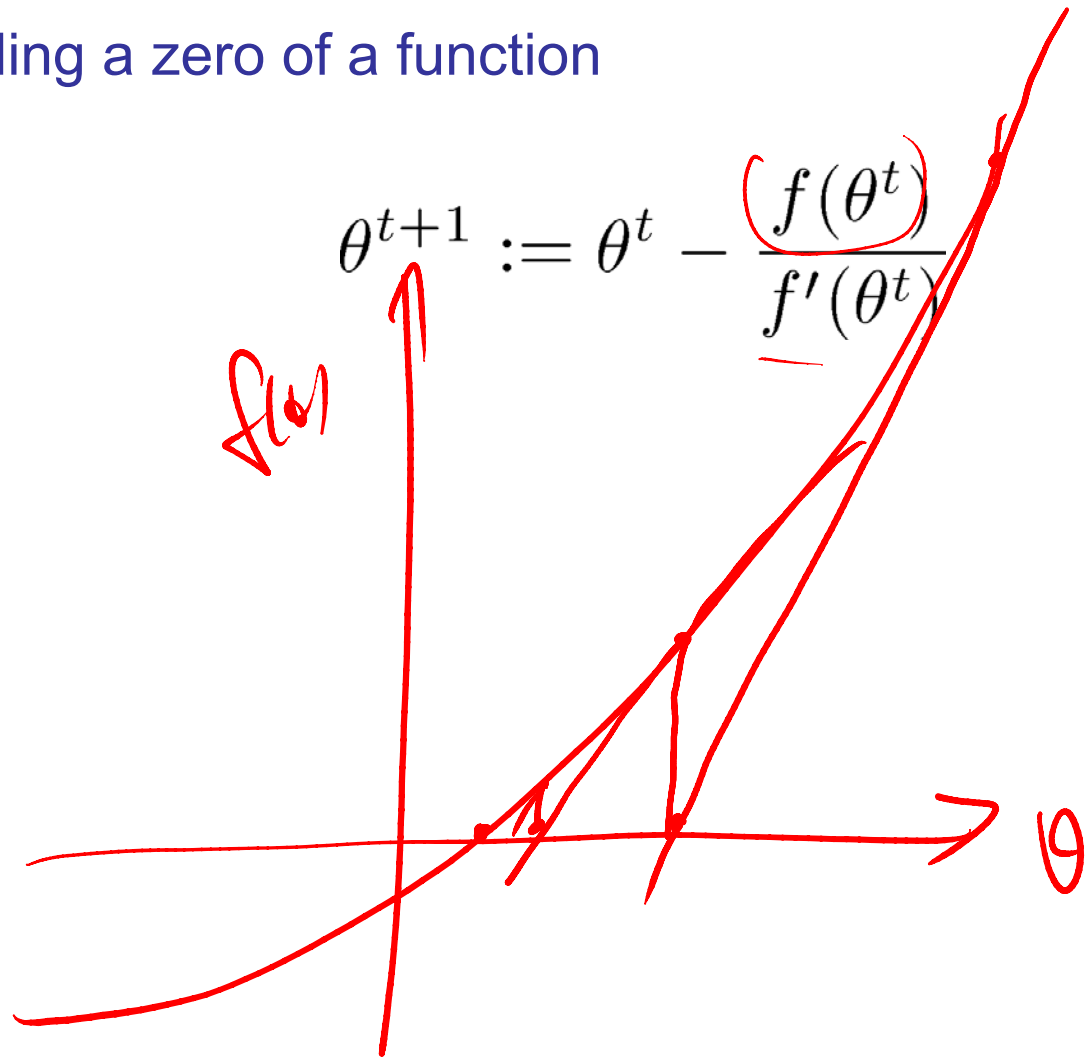
- MAP estimate
- More next lecture ...



# The Newton's method

$\theta \in \mathcal{P}(\theta)$   
m

- Finding a zero of a function





# The Newton's method (con'd)

- To maximize the conditional likelihood  $l(\theta)$ :

$$l(\theta) = \sum_i (y_i - 1)\theta^T x_i + \ln(1 + e^{-\theta^T x_i})$$

since  $l$  is convex, we need to find  $\theta^*$  where  $l'(\theta^*)=0$  !

- So we can perform the following iteration:

$$\theta^{t+1} := \theta^t + \frac{l'(\theta^t)}{l''(\theta^t)}$$

$$\left[ \begin{array}{c} \frac{\partial l}{\partial \theta_i} \\ \vdots \end{array} \right]$$



# The Newton-Raphson method

- In LR the  $\theta$  is vector-valued, thus we need the following generalization:

$$\theta^{t+1} := \theta^t + H^{-1} \nabla_{\theta^t} l(\theta^t)$$

- $\nabla$  is the gradient operator over the function
- $H$  is known as the Hessian of the function

$$\begin{bmatrix} \frac{\partial^2 l}{\partial \theta_1 \partial \theta_1} & \frac{\partial^2 l}{\partial \theta_1 \partial \theta_2} \\ \frac{\partial^2 l}{\partial \theta_2 \partial \theta_1} & \frac{\partial^2 l}{\partial \theta_2 \partial \theta_2} \\ \vdots & \vdots \end{bmatrix}$$



# The Newton-Raphson method

- In LR the  $\theta$  is vector-valued, thus we need the following generalization:

$$\theta^{t+1} := \theta^t + H^{-1} \nabla_{\theta^t} l(\theta^t)$$

- $\nabla$  is the gradient operator over the function

$$\nabla_{\theta} l(\theta) = \sum_i (y_i - u_i) x_i = \mathbf{X}^T (\mathbf{y} - \mathbf{u})$$

- $H$  is known as the Hessian of the function

$$H = \nabla_{\theta} \nabla_{\theta} l(\theta) = \sum_i u_i (1 - u_i) x_i x_i^T = \mathbf{X}^T \mathbf{R} \mathbf{X}$$

$$\text{where } R_{ii} = u_i (1 - u_i)$$

- This is also known as Iterative reweighted least squares (IRLS)

# Iterative reweighted least squares (IRLS)



- Recall in the least square est. in linear regression, we have:

$$\theta = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

which can also be derived from Newton-Raphson

- Now for logistic regression:

$$\begin{aligned} \theta^{t+1} &= \theta^t + H^{-1} \nabla_{\theta^t} l(\theta^t) \\ &= \theta^t - (\mathbf{X}^T \mathbf{R} \mathbf{X})^{-1} \mathbf{X}^T (\mathbf{u} - \mathbf{y}) \\ &= (\mathbf{X}^T \mathbf{R} \mathbf{X})^{-1} \{ \mathbf{X}^T \mathbf{R} \mathbf{X} \theta^t - \mathbf{X}^T (\mathbf{u} - \mathbf{y}) \} \\ &= (\mathbf{X}^T \mathbf{R} \mathbf{X})^{-1} \mathbf{X}^T \mathbf{R} \mathbf{z} \end{aligned}$$

# IRLS



- Recall in the least square est. in linear regression, we have:

$$\theta = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

which can also derived from Newton-Raphson

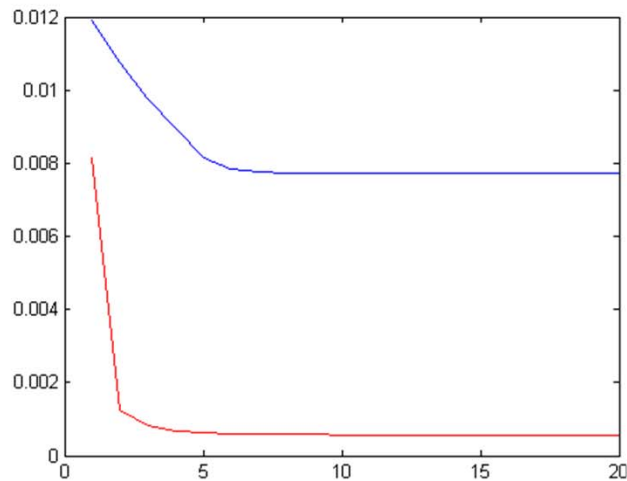
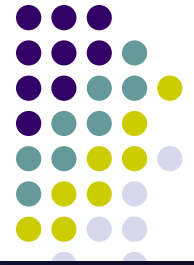
- Now for logistic regression:

$$\theta^{t+1} = (\mathbf{X}^T \mathbf{R} \mathbf{X})^{-1} \mathbf{X}^T \mathbf{R} \mathbf{z}$$

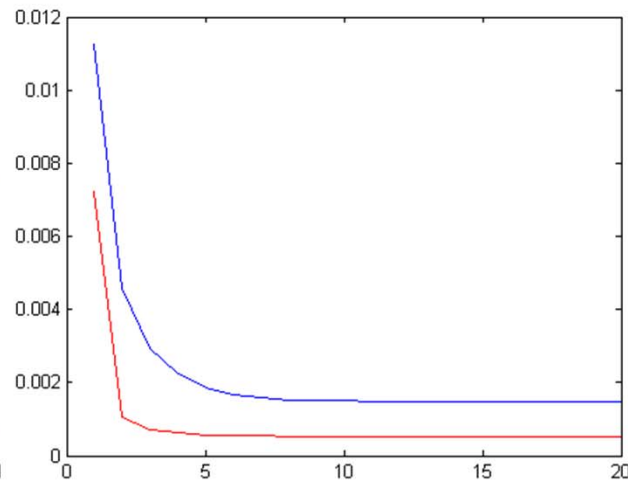
where  $\mathbf{z} = \mathbf{X}\theta^t - \mathbf{R}^{-1}(\mathbf{u} - \mathbf{y})$

and  $R_{ii} = u_i(1 - u_i)$

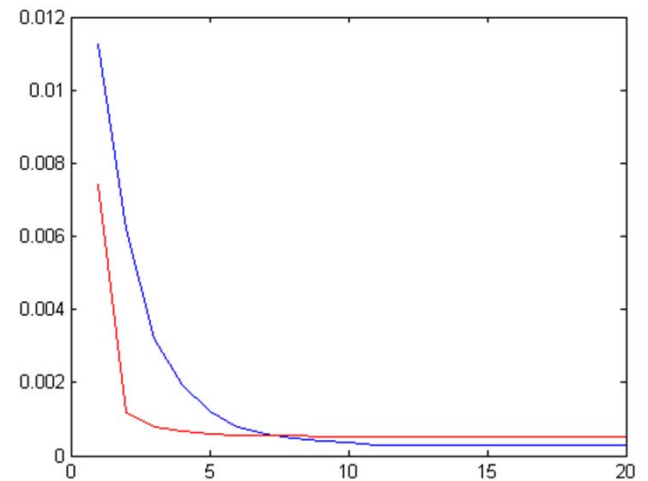
# Convergence curves



**alt.atheism**  
**vs.**  
**comp.graphics**



**rec.autos**  
**vs.**  
**rec.sport.baseball**



**comp.windows.x**  
**vs.**  
**rec.motorcycles**

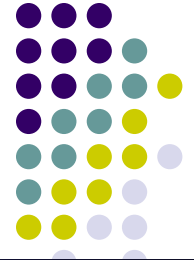
**Legend:** - X-axis: Iteration #; Y-axis: error  
- In each figure, red for **IRLS** and blue for **gradient descent**



# Logistic regression: practical issues



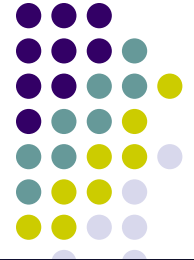
- NR (IRLS) takes  $O(N+d^3)$  per iteration, where  $N$  = number of training cases and  $d$  = dimension of input  $x$ , but converge in fewer iterations
- Quasi-Newton methods, that approximate the Hessian, work faster.
- Conjugate gradient takes  $O(Nd)$  per iteration, and usually works best in practice.
- Stochastic gradient descent can also be used if  $N$  is large c.f. perceptron rule:



# Case Study: Text classification

- Classify e-mails
  - $Y = \{\text{Spam, NotSpam}\}$
- Classify news articles
  - $Y = \{\text{what is the topic of the article?}\}$
- Classify webpages
  - $Y = \{\text{Student, professor, project, ...}\}$
- What about the features  $X$ ?
  - The text!





# Features $X$ are entire document – $X^i$ for $i^{\text{th}}$ word in article

the world of

**TOTAL**



**all about the company**

Our energy exploration, production, and distribution operations span the globe, with activities in more than 100 countries.

At TOTAL, we draw our greatest strength from our fast-growing oil and gas reserves. Our strategic emphasis on natural gas provides a strong position in a rapidly expanding market.

Our expanding refining and marketing operations in Asia and the Mediterranean Rim complement already solid positions in Europe, Africa, and the U.S.

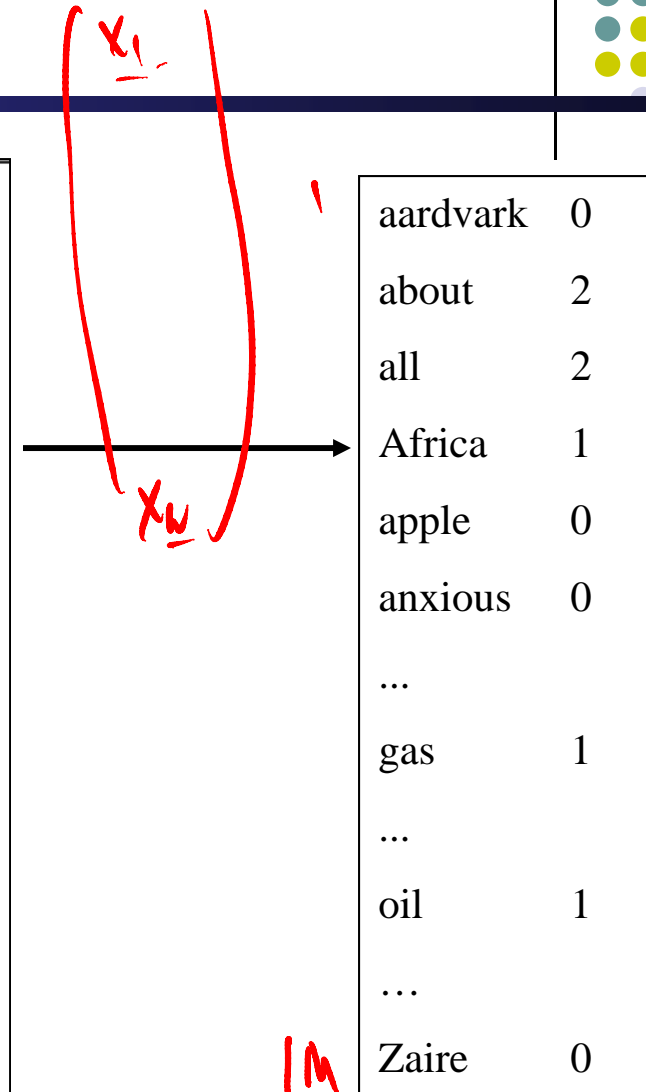
Our growing specialty chemicals sector adds balance and profit to the core energy business.

**All About The Company**

- Global Activities
- Corporate Structure
- TOTAL's Story
- Upstream Strategy
- Downstream Strategy
- Chemicals Strategy
- TOTAL Foundation
- Homepage

*Handwritten red 'i' next to the globe image.*

*Handwritten red 'N' at the bottom right of the page.*





# Bag of words model

- Typical additional assumption – **Position in document doesn't matter**:  $P(X^i=x^i|Y=y) = P(X^k=x^i|Y=y)$ 
  - “Bag of words” model – order of words on the page ignored
  - Sounds really silly, but often works very well!

$$P(y) \prod_{i=1}^{LengthDoc} P(x^i|y) \quad \text{or} \quad P(y) \prod_{k=1}^{LengthVol} P(w^k|y)$$

**When the lecture is over, remember to wake up the person sitting next to you in the lecture room.**



# Bag of words model

- Typical additional assumption – **Position in document doesn't matter**:  $P(X^i=x^i|Y=y) = P(X^k=x^i|Y=y)$ 
  - “Bag of words” model – order of words on the page ignored
  - Sounds really silly, but often works very well!

$$P(y) \prod_{i=1}^{LengthDoc} P(x^i|y) \quad \text{or} \quad P(y) \prod_{k=1}^{LengthVol} P(w^k|y)$$

in is lecture lecture next over person remember room  
sitting the the the to to up wake when you

# NB with Bag of Words for text classification



- Learning phase:
  - Prior  $P(Y)$ 
    - Count how many documents you have from each topic (+ prior)
  - $P(X^i|Y)$ 
    - For each topic, count how many times you saw word in documents of this topic (+ prior)
- Test phase:
  - For each document  $\mathbf{x}_{\text{new}}$ 
    - Use naïve Bayes decision rule

$$h_{NB}(\mathbf{x}_{\text{new}}) = \arg \max_y P(y) \prod_{i=1}^{\text{LengthDoc}} P(x_{\text{new}}^i|y)$$



# Back to our 20 NG Case study

- Dataset

- 20 News Groups (20 classes)
- 61,118 words, 18,774 documents

comp.graphics comp.os.ms-windows.misc comp.sys.ibm.pc.hardware comp.sys.mac.hardware comp.windows.x	rec.autos rec.motorcycles rec.sport.baseball rec.sport.hockey	sci.crypt sci.electronics sci.med sci.space
misc.forsale	talk.politics.misc talk.politics.guns talk.politics.mideast	talk.religion.misc alt.atheism soc.religion.christian

- Experiment:

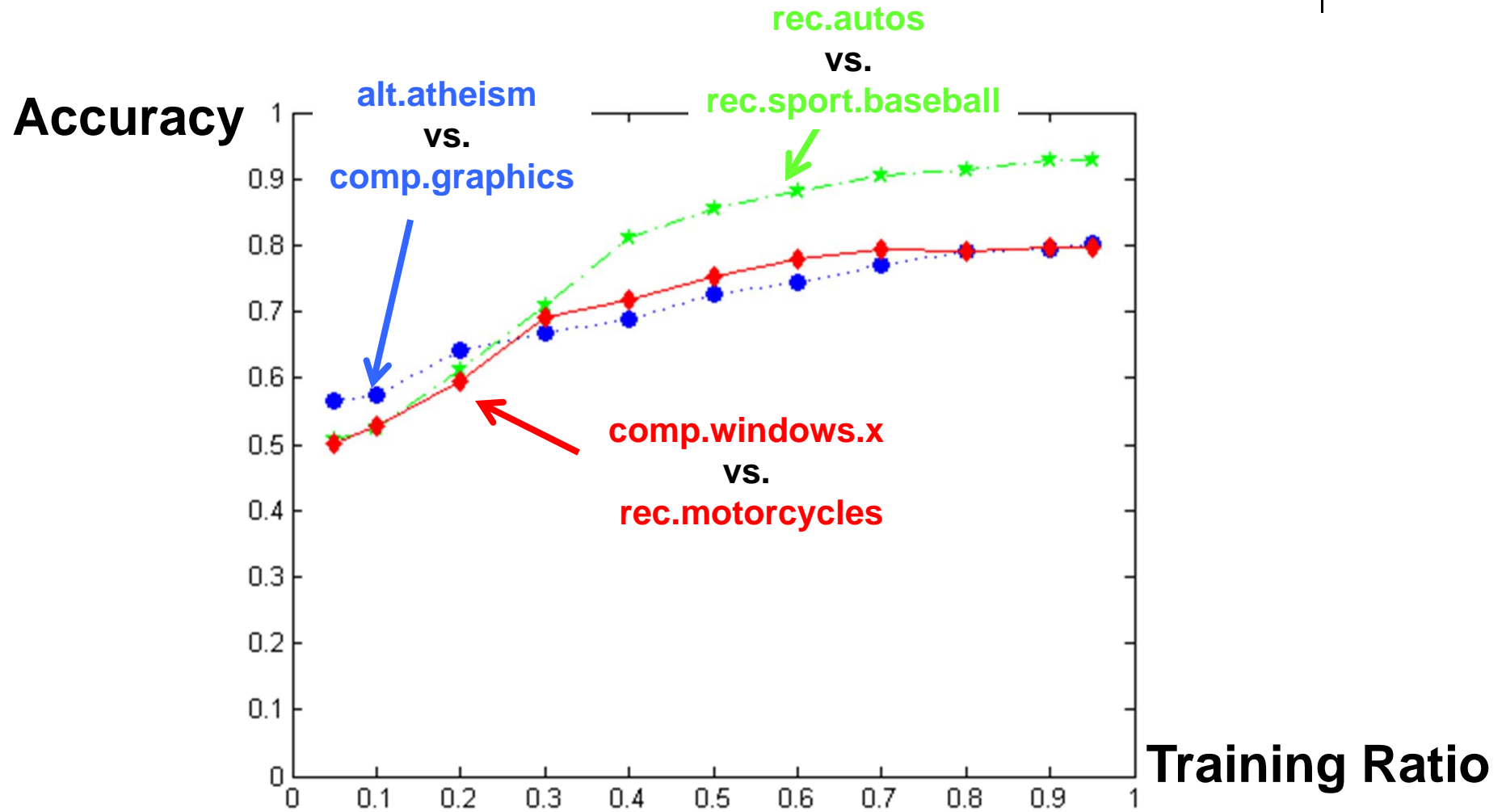
- Solve only a two-class subset: 1 vs 2.
- 1768 instances, 61188 features.
- Use dimensionality reduction on the data (SVD).
- Use 90% as training set, 10% as test set.
- Test prediction error used as accuracy measure.

*Handwritten red notes:*  
mix  
O  
C  
S  
C  
C

$$Accuracy = \frac{\sum_{i \in \text{test set}} I(\text{predict}_i = \text{true label}_i)}{\# \text{ of test samples}}$$



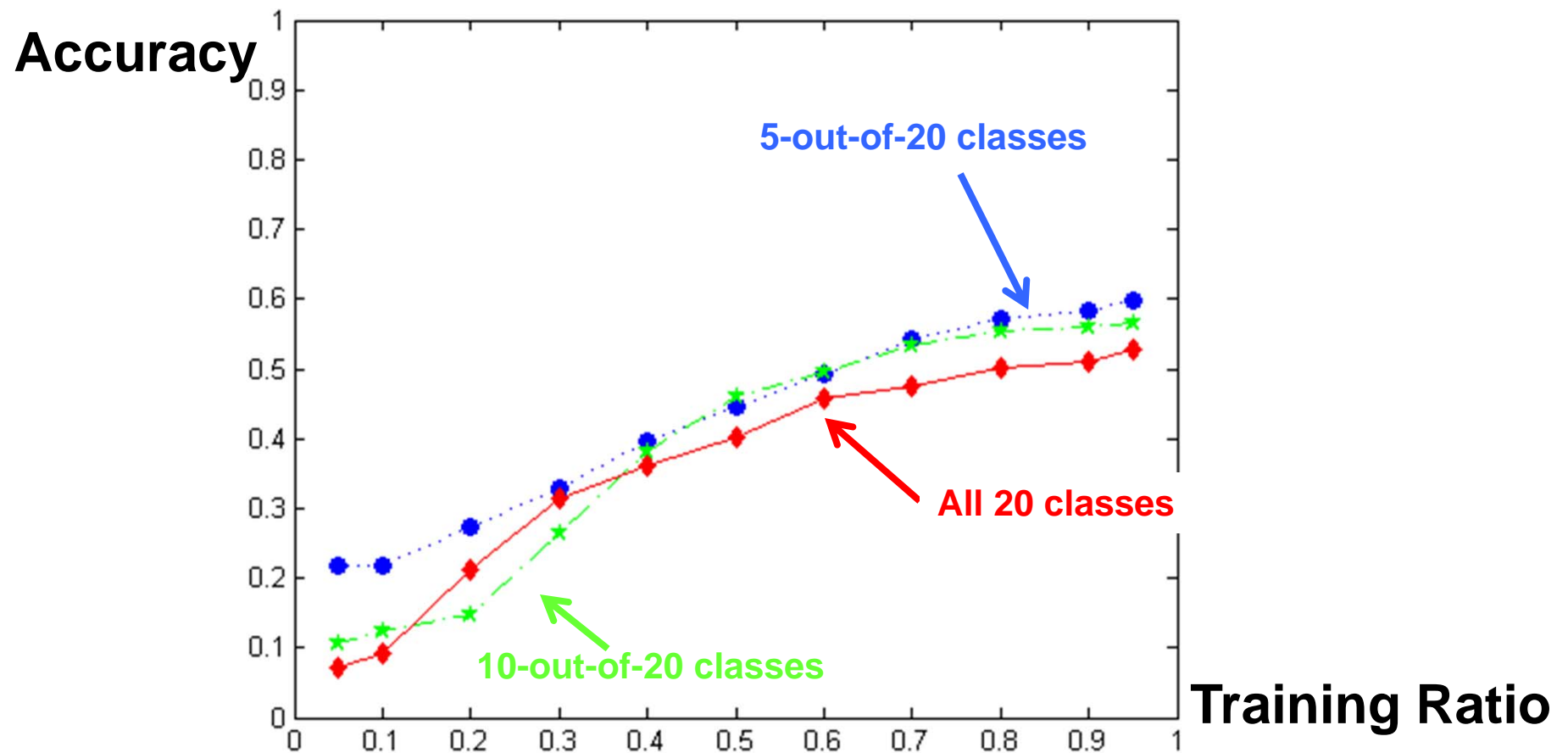
# Results: Binary Classes

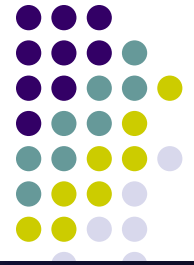






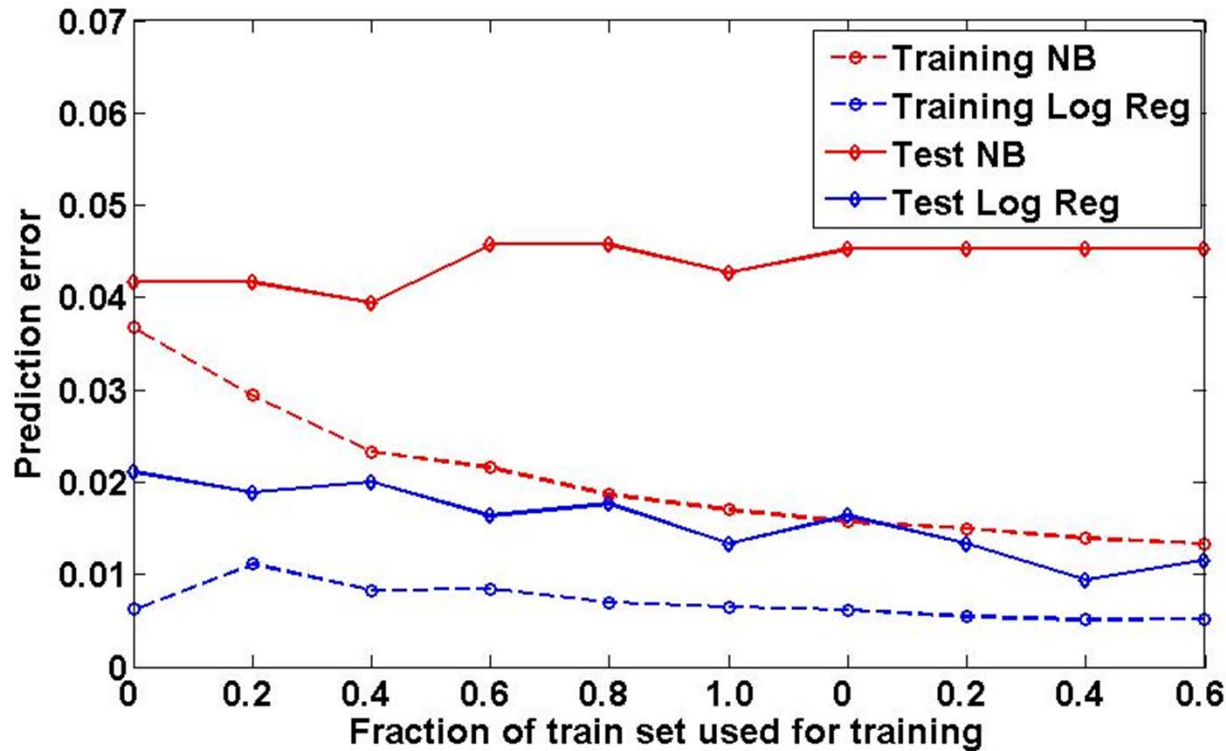
# Results: Multiple Classes





# NB vs. LR

- Versus training size

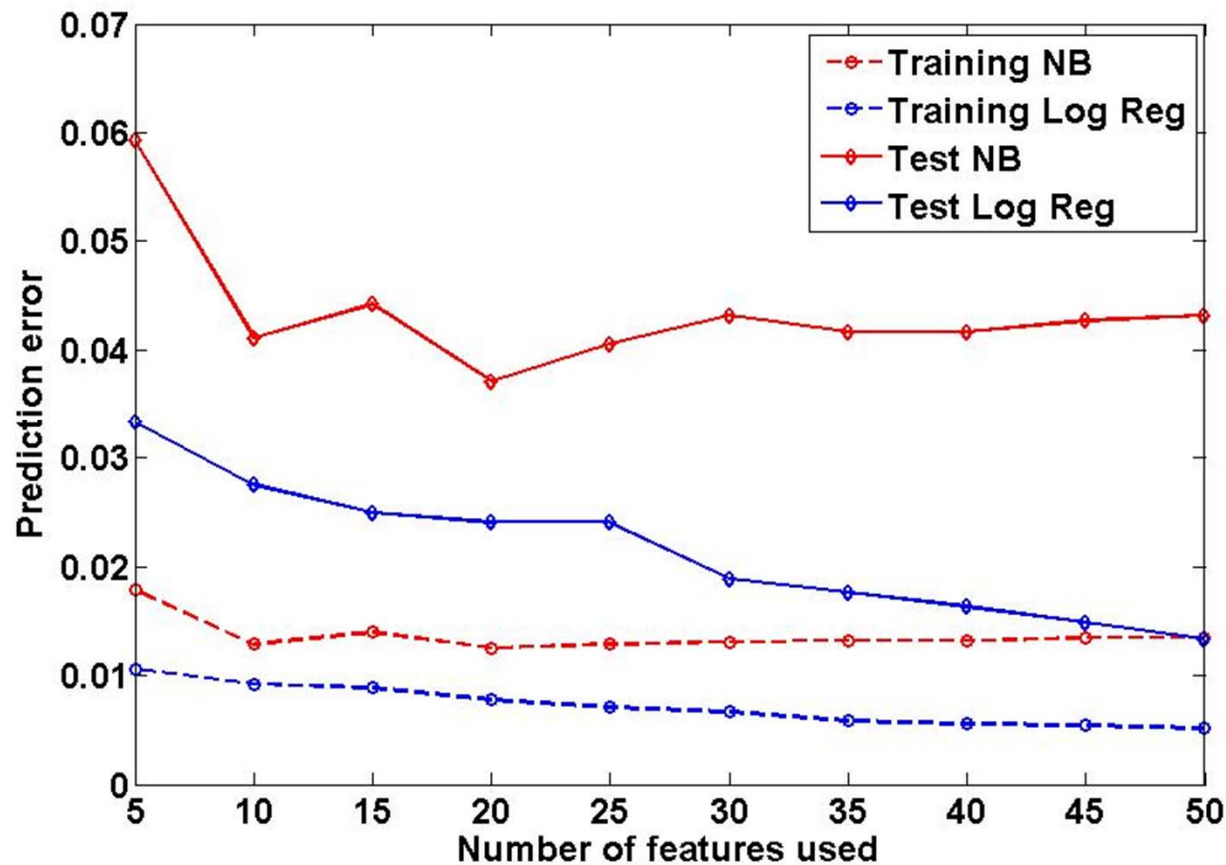


- 30 features.
- A fixed test set
- Training set varied from 10% to 100% of the training set



# NB vs. LR

- Versus model size



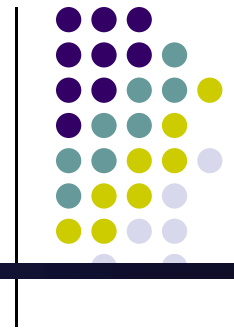
Number of dimensions of the data varied from 5 to 50 in steps of 5

The features were chosen in decreasing order of their singular values

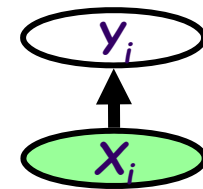
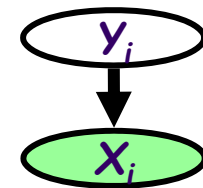
90% versus 10% split on training and test

# Summary:

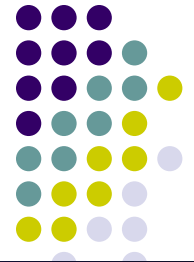
## Generative vs. Discriminative Classifiers



- Goal: Wish to learn  $f: X \rightarrow Y$ , e.g.,  $P(Y|X)$
- Generative classifiers (e.g., Naïve Bayes):
  - Assume some functional form for  $P(X|Y)$ ,  $P(Y)$   
This is a '**generative**' model of the data!
  - Estimate parameters of  $P(X|Y)$ ,  $P(Y)$  directly from training data
  - Use Bayes rule to calculate  $P(Y|X=x)$
- Discriminative classifiers:
  - Directly assume some functional form for  $P(Y|X)$   
This is a '**discriminative**' model of the data!
  - Estimate parameters of  $P(Y|X)$  directly from training data



# Naïve Bayes vs Logistic Regression



- Consider  $Y$  boolean,  $X$  continuous,  $X = \langle X^1 \dots X^m \rangle$
- Number of parameters to estimate:

NB:

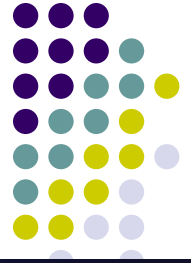
$$p(y | \mathbf{x}) = \frac{\pi_k \exp \left\{ - \sum_j \left( \frac{1}{2\sigma_{k,j}^2} (x_j - \mu_{k,j})^2 - \log \sigma_{k,j} - C \right) \right\}}{\sum_{k'} \pi_{k'} \exp \left\{ - \sum_j \left( \frac{1}{2\sigma_{k',j}^2} (x_j - \mu_{k',j})^2 - \log \sigma_{k',j} - C \right) \right\}}$$

LR:

$$\mu(x) = \frac{1}{1 + e^{-\theta^T x}}$$

- Estimation method:
  - NB parameter estimates are uncoupled
  - LR parameter estimates are coupled

# Naïve Bayes vs Logistic Regression



- Asymptotic comparison (# training examples  $\rightarrow$  infinity)
- when model assumptions correct
  - NB, LR produce identical classifiers
- when model assumptions incorrect
  - LR is less biased – does not assume conditional independence
  - therefore expected to outperform NB

# Naïve Bayes vs Logistic Regression



- Non-asymptotic analysis (see [Ng & Jordan, 2002] )
- convergence rate of parameter estimates – how many training examples needed to assure good estimates?

NB order  $\log m$  (where  $m = \#$  of attributes in  $X$ )

LR order  $m$

- NB converges more quickly to its (perhaps less helpful) asymptotic estimates

# Rate of convergence: logistic regression



- Let  $h_{Dis,m}$  be logistic regression trained on  $n$  examples in  $m$  dimensions. Then with high probability:

$$\epsilon(h_{Dis,n}) \leq \epsilon(h_{Dis,\infty}) + O\left(\sqrt{\frac{m}{n} \log \frac{n}{m}}\right)$$

- Implication: if we want  $\epsilon(h_{Dis,m}) \leq \epsilon(h_{Dis,\infty}) + \epsilon_0$  for some small constant  $\epsilon_0$ , it suffices to pick order  $m$  examples

→ Converges to its asymptotic classifier, in order  $m$  examples

- result follows from Vapnik's structural risk bound, plus fact that the "VC Dimension" of an  $m$ -dimensional linear separators is  $m$



# Rate of convergence: naïve Bayes parameters



- Let any  $\epsilon_1, \delta > 0$ , and any  $n \geq 0$  be fixed.  
Assume that for some fixed  $\rho_0 > 0$ ,  
we have that  $\rho_0 \leq p(y = T) \leq 1 - \rho_0$
- Let  $n = O((1/\epsilon_1^2) \log(m/\delta))$
- Then with probability at least  $1 - \delta$ , after  $n$  examples:

1. For discrete input, 
$$\begin{aligned} |\hat{p}(x_i | y = b) - p(x_i | y = b)| &\leq \epsilon_1 && \text{for all } i \text{ and } b \\ |\hat{p}(y = b) - p(y = b)| &\leq \epsilon_1 \end{aligned}$$

2. For continuous inputs, 
$$\begin{aligned} |\hat{\mu}_{i|y=b} - \mu_{i|y=b}| &\leq \epsilon_1 && \text{for all } i \text{ and } b \\ |\hat{\sigma}_{i|y=b}^2 - \sigma_{i|y=b}^2| &\leq \epsilon_1 \end{aligned}$$

# Some experiments from UCI data sets

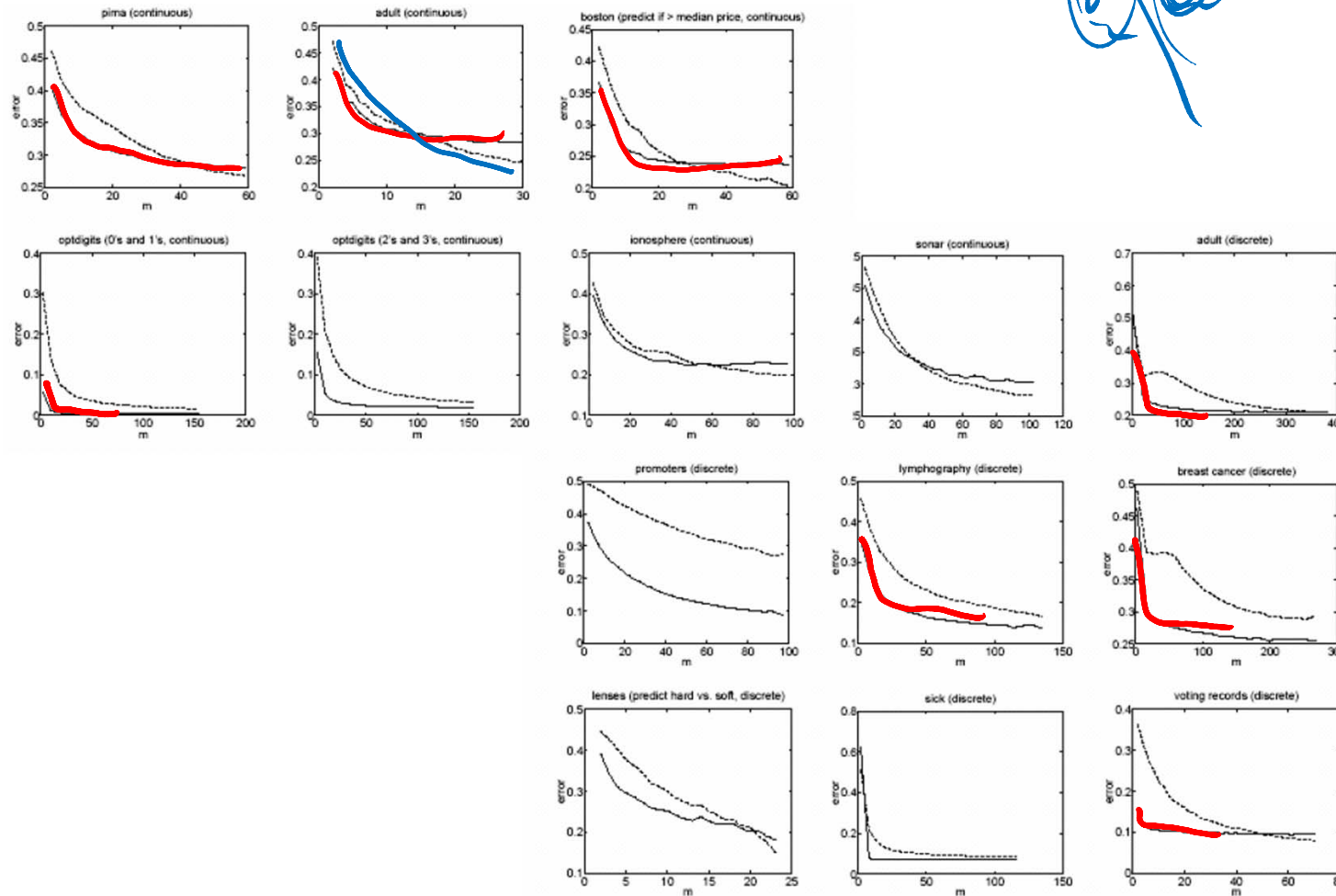
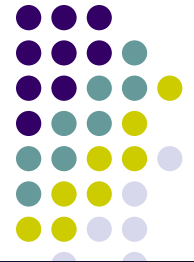


Figure 1: Results of 15 experiments on datasets from the UCI Machine Learning repository. Plots are of generalization error vs.  $m$  (averaged over 1000 random train/test splits). Dashed line is logistic regression; solid line is naive Bayes.  
© Eric Xing @ CMU, 2014



# Take home message

---

- Naïve Bayes classifier
  - What's the assumption
  - Why we use it
  - How do we learn it
- Logistic regression
  - Functional form follows from Naïve Bayes assumptions
  - For Gaussian Naïve Bayes assuming variance
  - For discrete-valued Naïve Bayes too
  - But training procedure picks parameters without the conditional independence assumption
- Gradient ascent/descent
  - – General approach when closed-form solutions unavailable
- Generative vs. Discriminative classifiers
  - – Bias vs. variance tradeoff