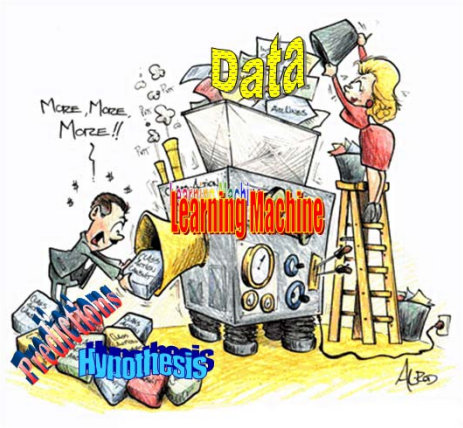
Advanced Introduction to Machine Learning

10715, Fall 2014

Introduction &
Linear Classifiers



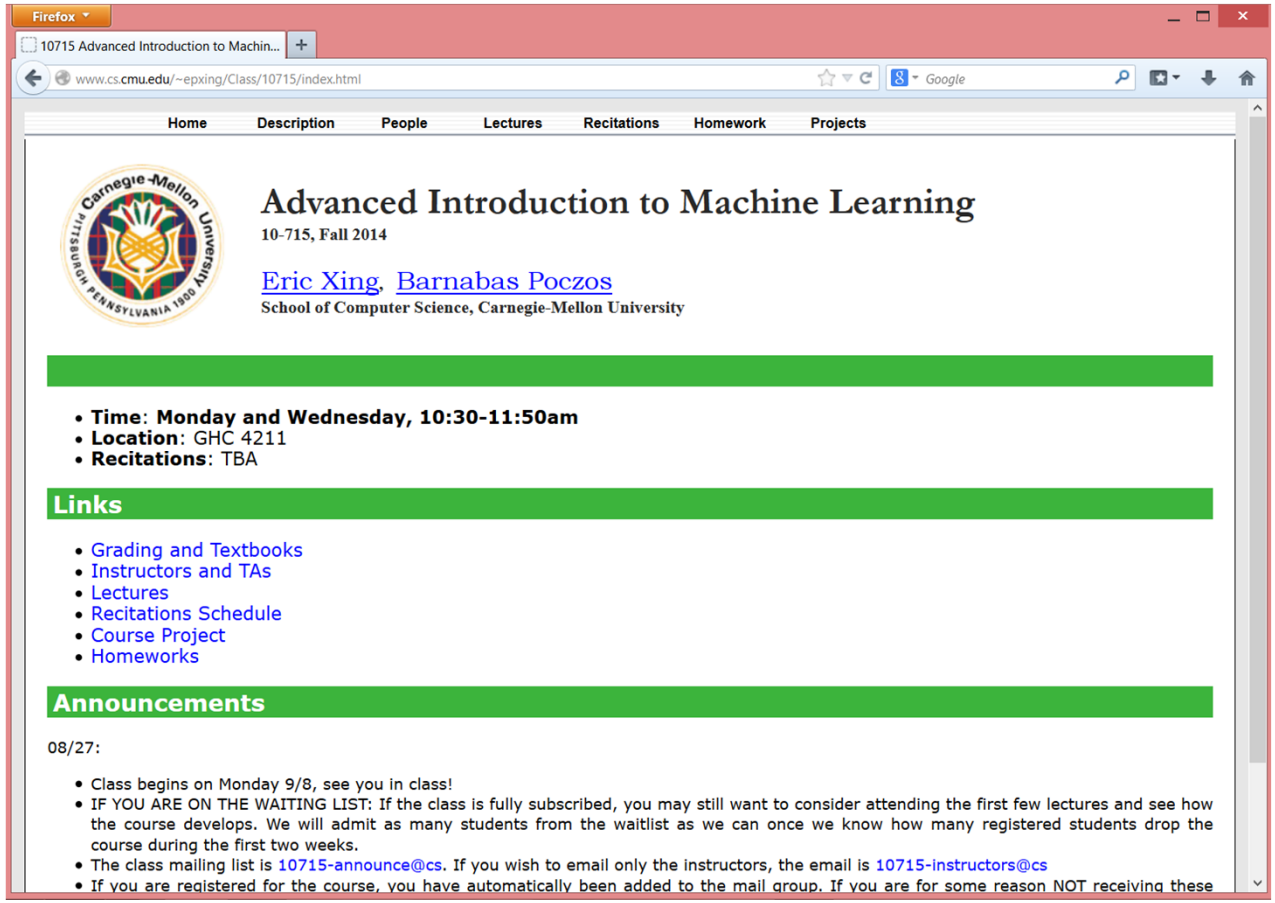
Eric Xing and Barnabas Poczos

Lecture 1, September 8, 2014

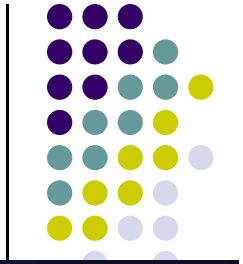
Reading:

Intro to Adv. ML 10-715

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

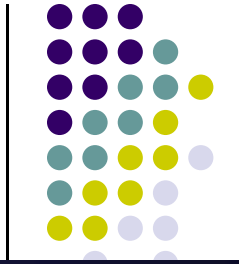


Logistics



- 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
  - Class announcements list: 10715-announce@cs.cmu.edu.
- TA:
  - Kirthevasan Kandasamy, GHC 8015
  - Veeranjaneyulu Sadhanala, GHC 8005
- Guest Lecturers
  - Yaoliang Yu
  - Andrew Wilson
- Class Assistant:
  - Mallory Deptola, GHC 8001, x8-5527

Logistics



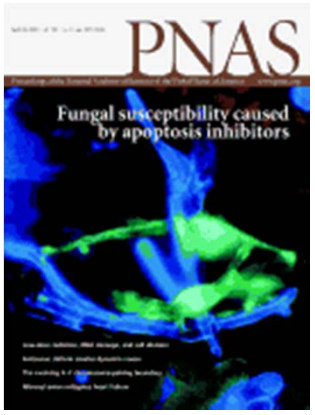
- 4 homework assignments: 40% of grade
  - Theory exercises
  - Implementation exercises
- **Final project: 40% of grade**
  - Applying machine learning to your research area
    - NLP, IR,, vision, robotics, computational biology …
  - Outcomes that offer real utility and value
    - Search all the 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

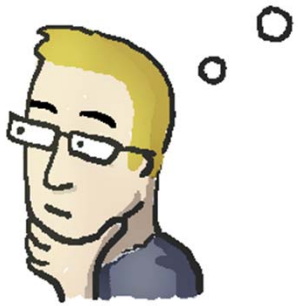
Learning is about seeking a predictive and/or executable understanding of natural/artificial subjects, phenomena, or activities from …

Grammatical rules
Manufacturing procedures
Natural laws
…

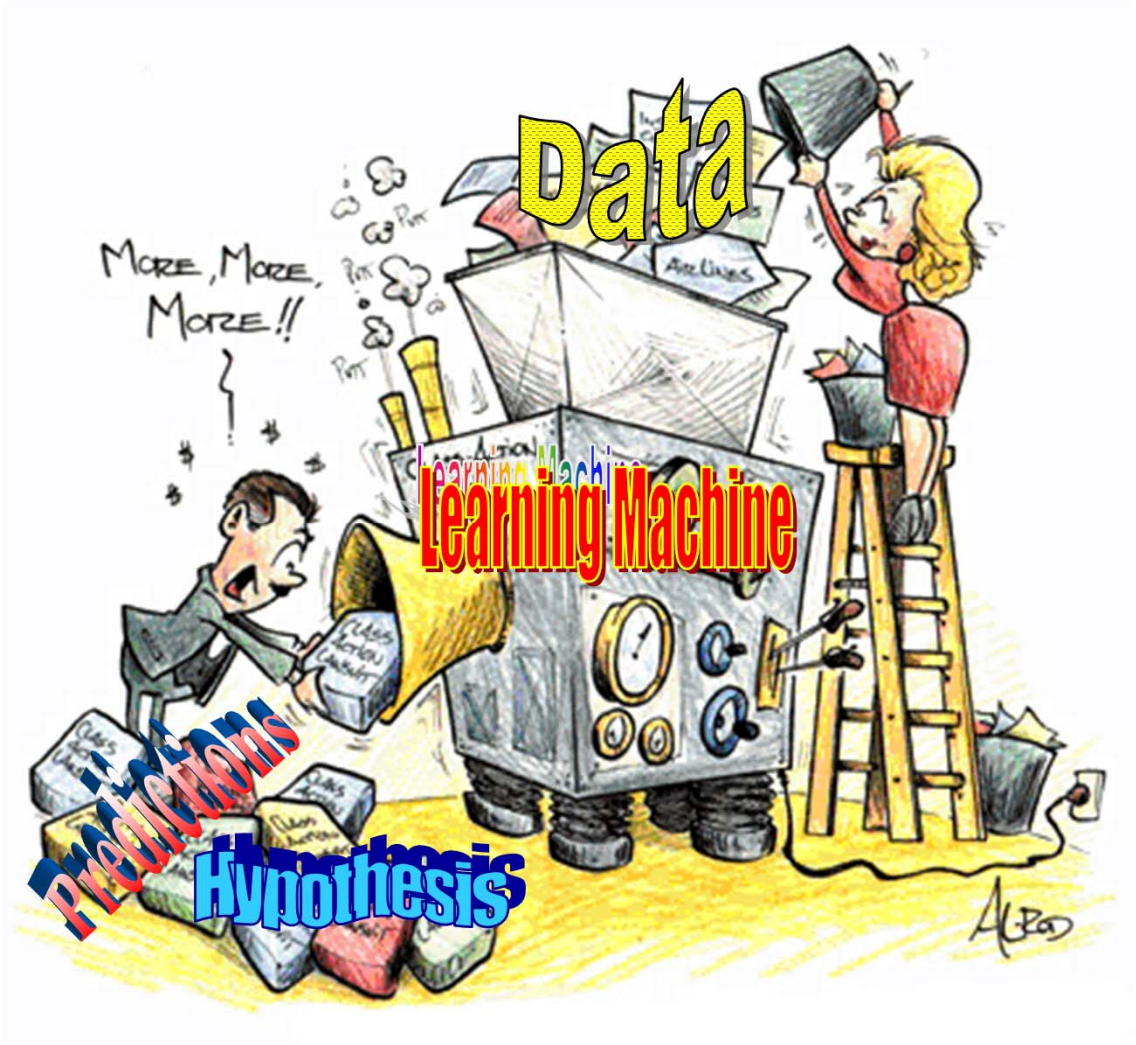
Apoptosis + Medicine



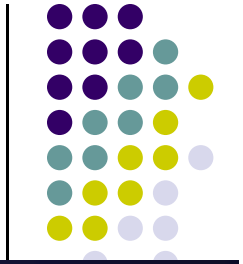
Inference:
what does this mean?
Any similar article?
…



Machine Learning



What is Machine Learning?



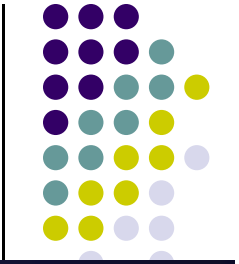
Machine Learning seeks to develop theories and computer systems for

- 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

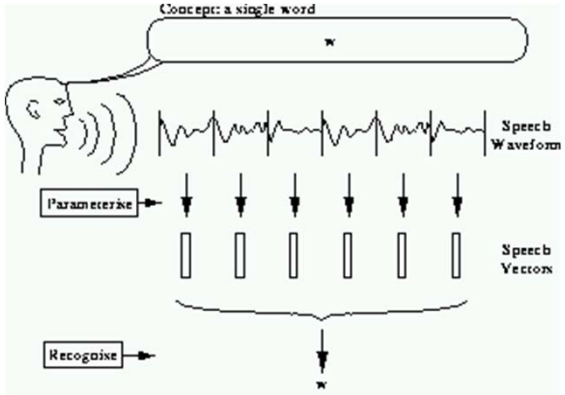
flickr

3.6 billion photos

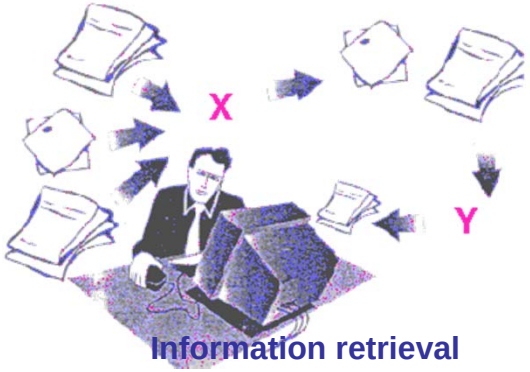
YouTube

24 hours videos uploaded per minute

Machine Learning is Prevalent

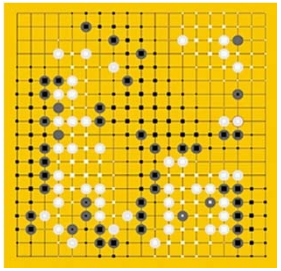


Speech recognition

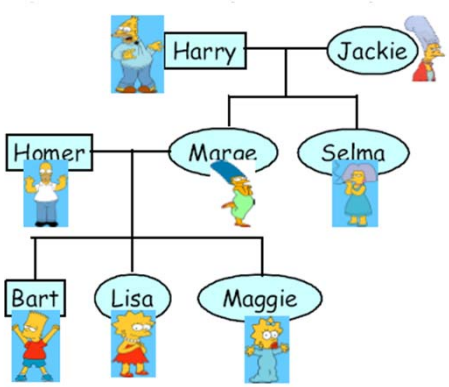


Information retrieval

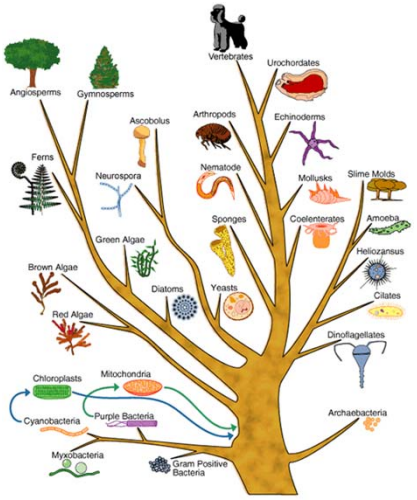
Computer vision



Games



Pedigree

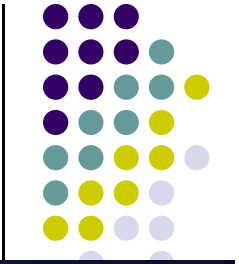


Evolution

Robotic control

Planning

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!

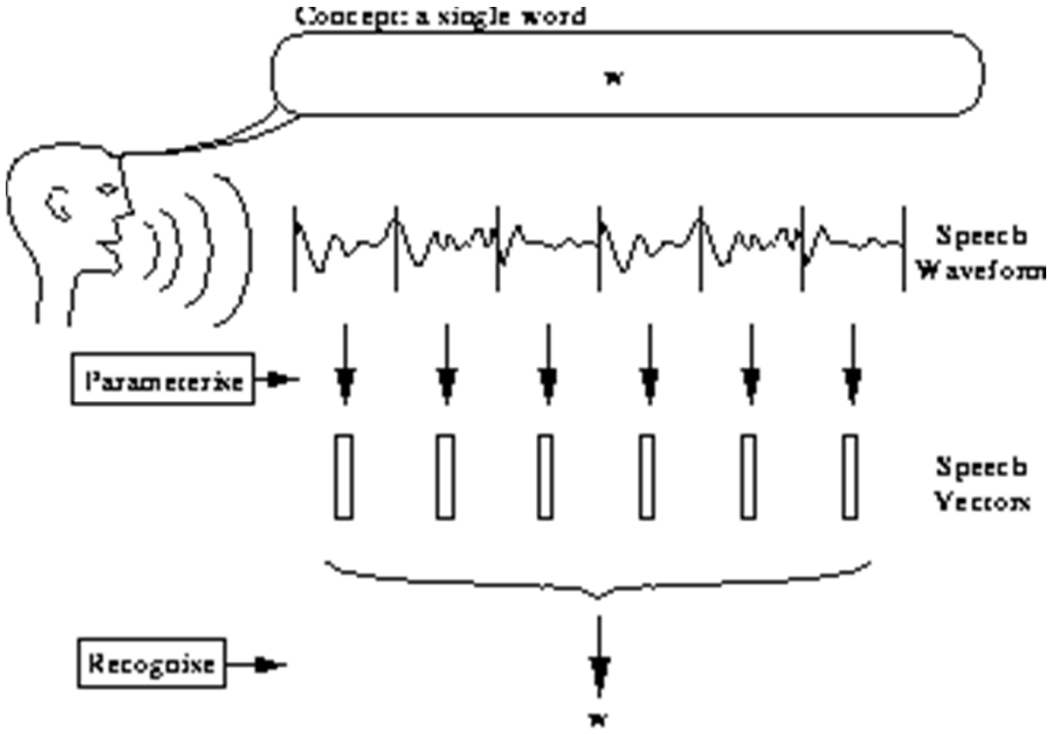
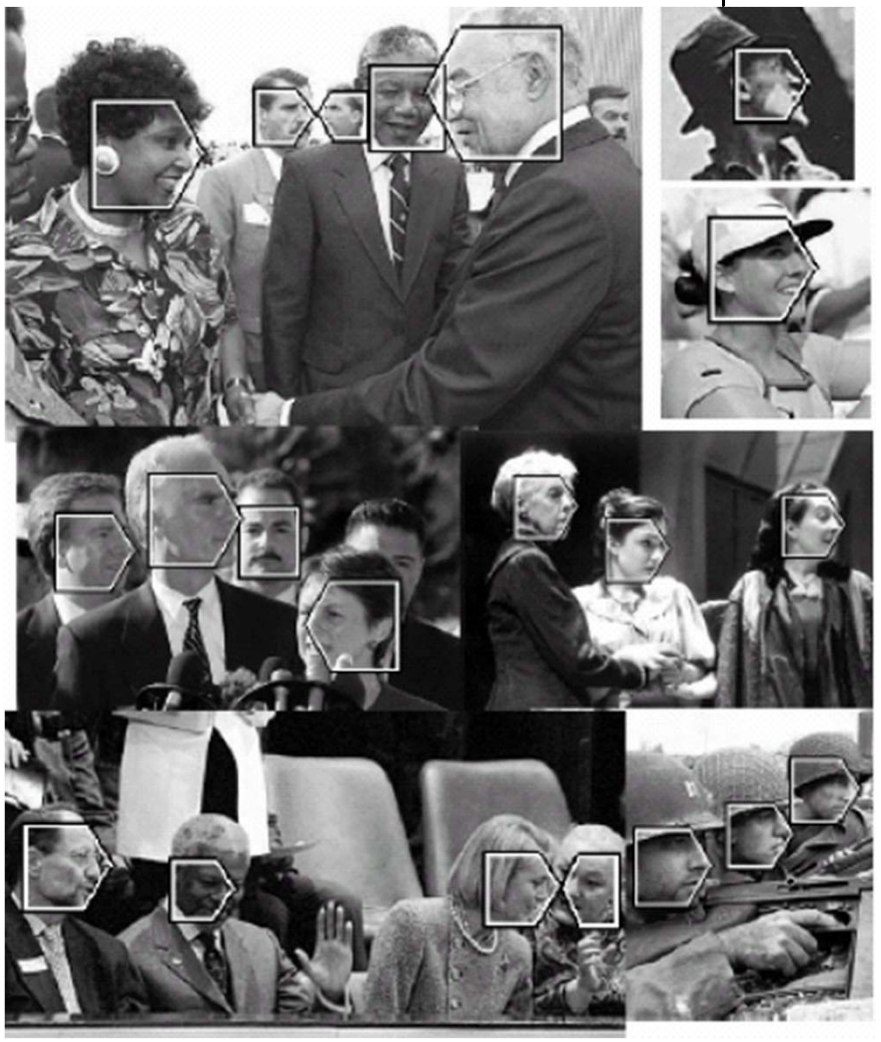


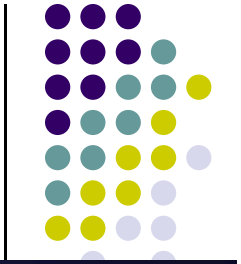
Fig. 1.2 Isolated Word Problem

Object Recognition

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



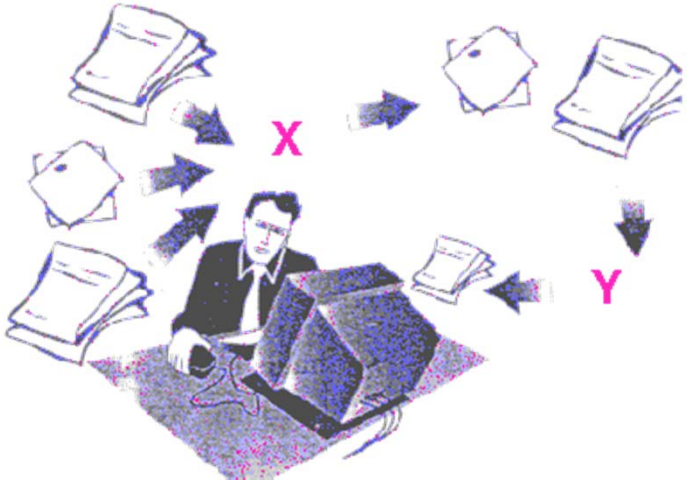
Robotic Control



- 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

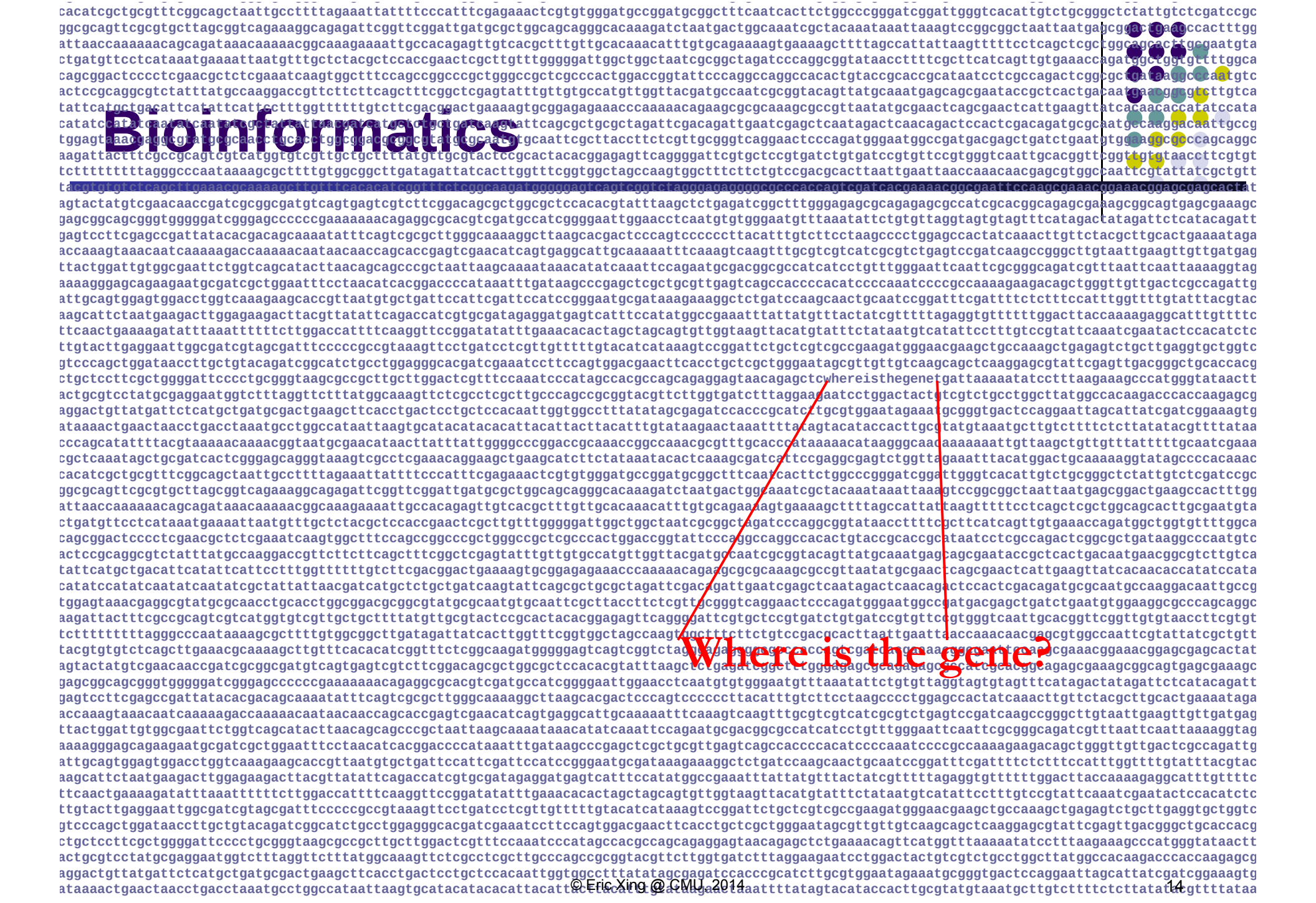
- Reading, digesting, and categorizing a vast text database is too much for human!



- **We want:**

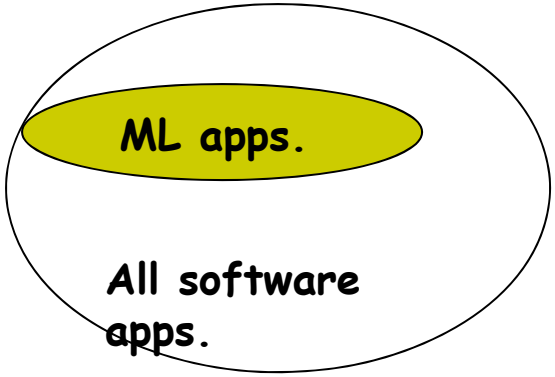
"Arts"	"Budgets"		
NEW	MILLION	YEARS	TEACHERS
FILM	TAX	FAMILIES	HIGH
SHOW	PROGRAM	WORK	PUBLIC
MUSIC	BUDGET	PARENTS	TEACHER
MOVIE	BILLION	SAYS	BENNETT
PLAY	FEDERAL	FAMILY	MANIGAT
MUSICAL	YEAR	WELFARE	NAMPHY
BEST	SPENDING	MEN	STATE
ACTOR	NEW	PERCENT	PRESIDENT
FIRST	STATE	CARE	ELEMENTARY
YORK	PLAN	LIFE	HAITI
OPERA	MONEY		
THEATER	PROGRAMS		
ACTRESS	GOVERNMENT		
LOVE	CONGRESS		

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.


Bioinformatics
Where is the gene?

Growth of Machine Learning

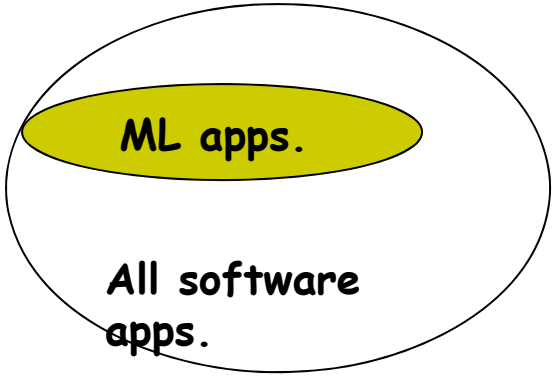
- Machine learning already the preferred approach to
  - Speech recognition, Natural language processing
  - Computer vision
  - Medical outcomes analysis
  - Robot control
  - …



- This ML niche is growing (why?)

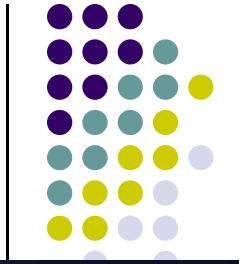
Growth of Machine Learning

- 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. $D^{new} = \{\mathbf{X}_j\} \Rightarrow \{\mathbf{Y}_j\}$
- Unsupervised Learning
  - Given $D = \{\mathbf{X}_i\}$, learn $f(\cdot): \mathbf{Y}_i = f(\mathbf{X}_i)$, s.t. $D^{new} = \{\mathbf{X}_j\} \Rightarrow \{\mathbf{Y}_j\}$
- Semi-supervised Learning
- Reinforcement Learning
  - Given $D = \{\text{env}, \text{actions}, \text{rewards}, \text{simulator/trace/real game}\}$

    learn $\begin{matrix}\text{policy}: e, r \rightarrow a \\ \text{utility}: a, e \rightarrow r\end{matrix}$, s.t. $\{\text{env}, \text{new real game}\} \Rightarrow a_1, a_2, a_3 \ldots$
- Active Learning
  - Given $D \sim G(\cdot)$, learn $D^{new} \sim G'(\cdot)$ and $f(\cdot)$, s.t. $D^{all} \Rightarrow G'(\cdot), \text{policy}, \{\mathbf{Y}_j\}$

Machine Learning - Theory

For the learned F(; θ)

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

PAC Learning Theory

(supervised concept learning)

 # examples (*m*)

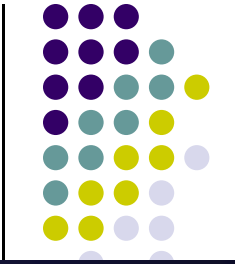
representational complexity (*H*)

error rate (ε)

failure probability (δ)

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

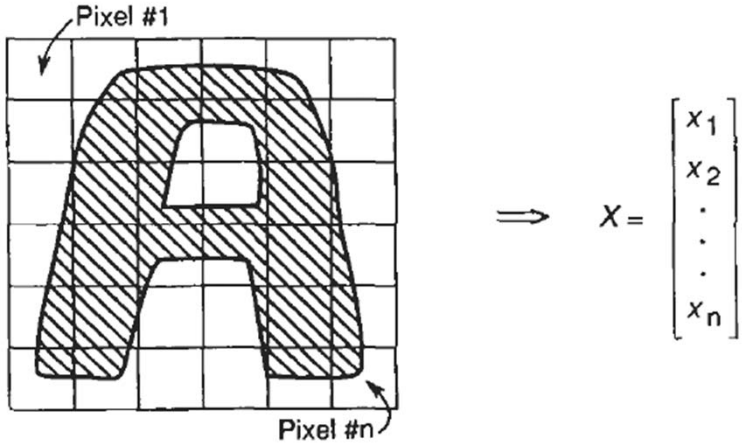
Elements of Machine Learning



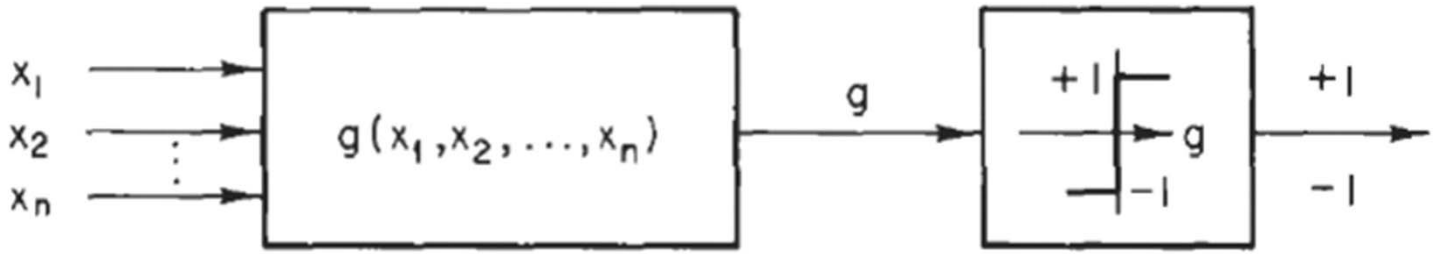
- **Here are some important elements to consider before you start:**
  - Task:
    - Embedding? Classification? Clustering? Topic extraction? …
  - Data and other info:
    - Input and output (e.g., continuous, binary, counts, …)
    - Supervised or unsupervised, of a blend of everything?
    - Prior knowledge? Bias?
  - Models and paradigms:
    - BN? MRF? Regression? SVM?
    - Bayesian/Frequents ? 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

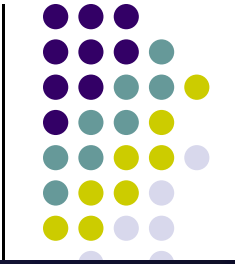
- Representing data:



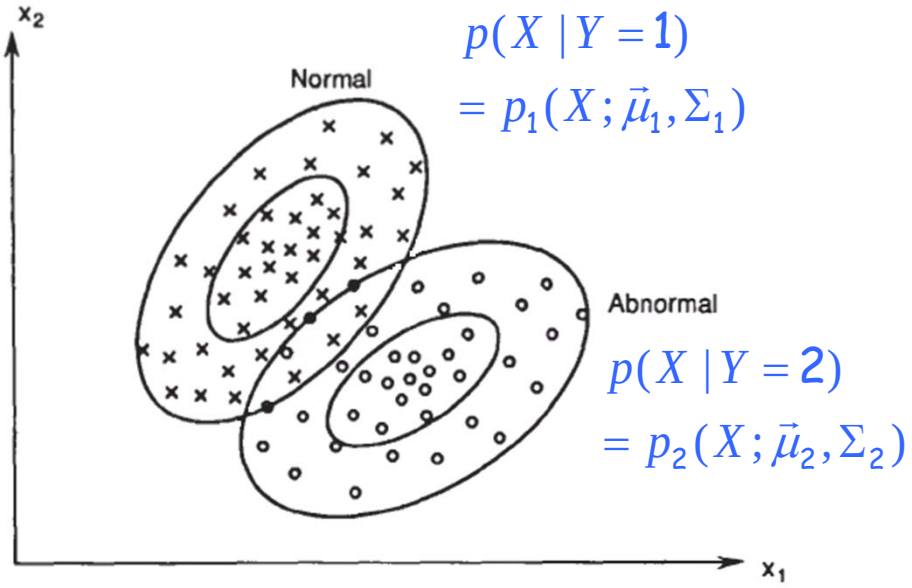
- Hypothesis (classifier)



Decision-making as dividing a high-dimensional space

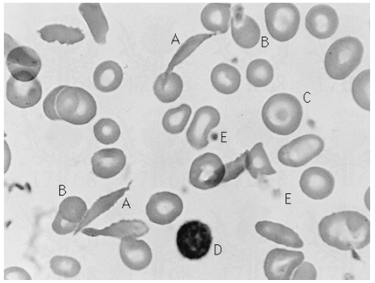


- Classification-specific Dist.: P(X|Y)



$$p(X \mid Y = 1) = p_1(X; \vec{\mu}_1, \Sigma_1)$$

$$p(X \mid Y = 2) = p_2(X; \vec{\mu}_2, \Sigma_2)$$



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

The Bayes Rule

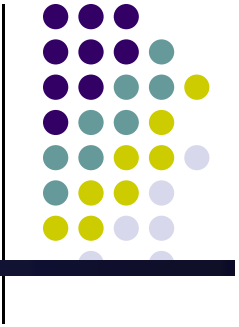
- What we have just did leads to the following general expression:

$$P(Y \mid X) = \frac{P(X \mid Y)\, p(Y)}{P(X)}$$

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

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

- Bayes Test:

- Likelihood Ratio:

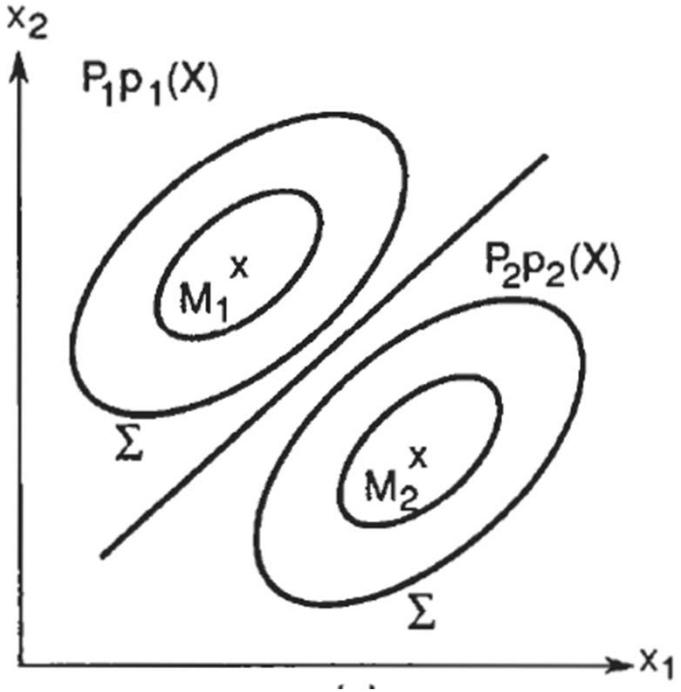
$$\ell(X) =$$

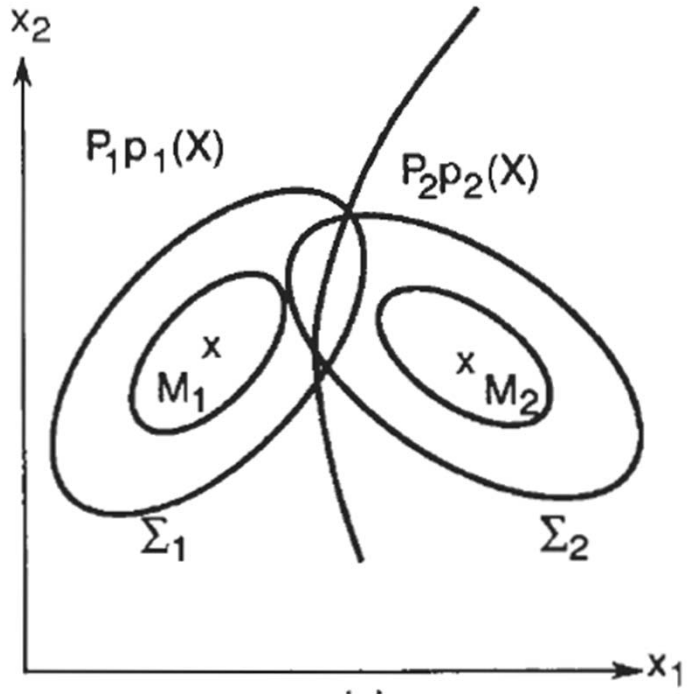
- Discriminant function:

$$h(X) =$$

Example of Decision Rules

- When each class is a normal …





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

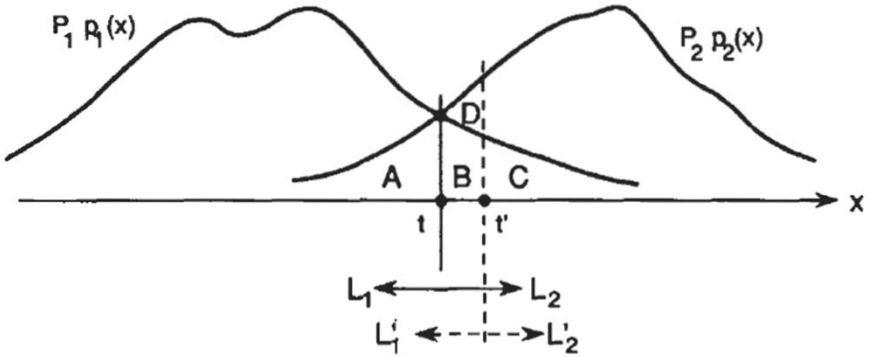
Bayes Error

- 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*?

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

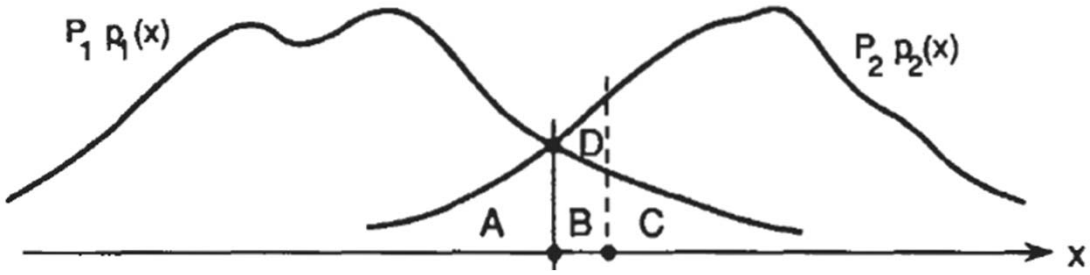
- The Bayes error (the expected risk):

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



More on Bayes Error

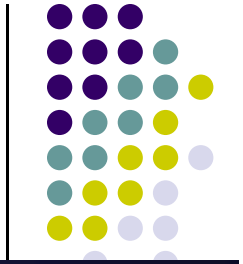
- Bayes error is the lower bound of probability of classification error



- 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:

$$\epsilon_1 = \int_{\ln(\pi_1/\pi_2)}^{+\infty} p_1(x)dx \qquad \epsilon_2 = \int_{-\infty}^{\ln(\pi_1/\pi_2)} p_2(x)dx$$

Learning Classifier

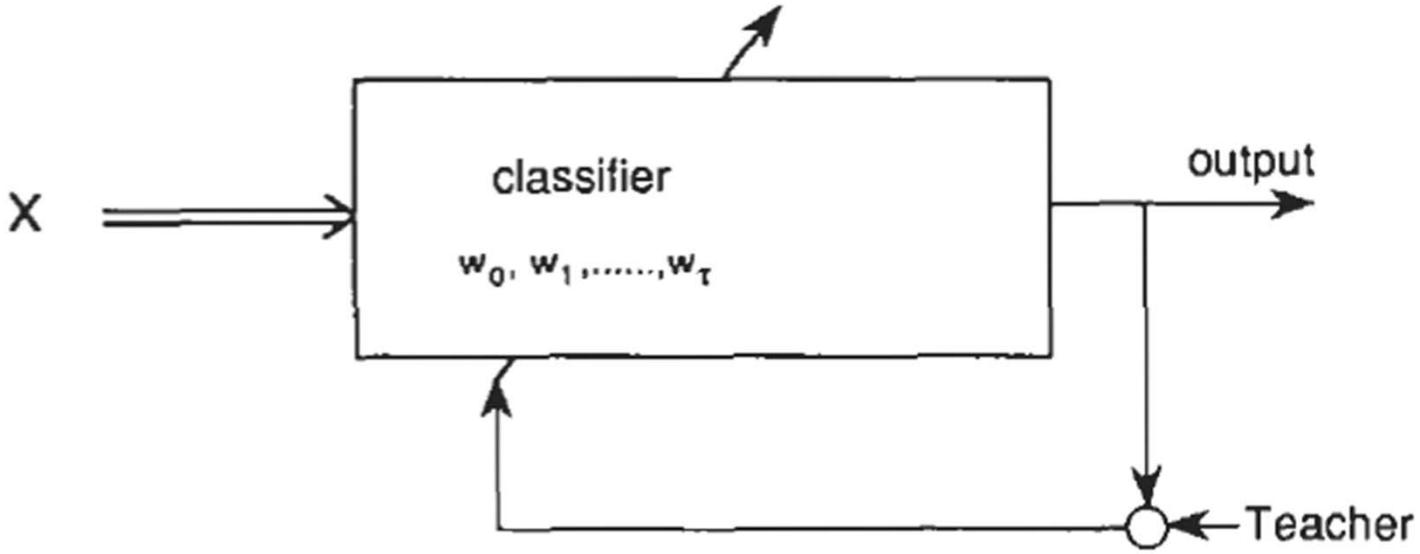


- The decision rule:

$$h(X) = -\ln p_1(X) + \ln p_2(X) \underset{<}{\overset{>}{}} \ln \frac{\pi_1}{\pi_2}$$

- Learning strategies
  - Generative Learning
  - Discriminative Learning
  - Instance-based Learning (Store all past experience in memory)
    - A special case of nonparametric classifier

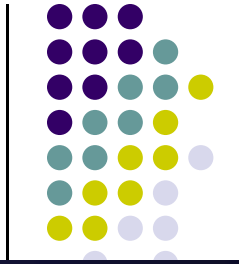
Supervised Learning



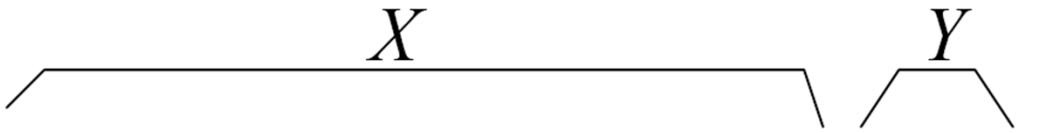
- K-Nearest-Neighbor Classifier:

  where the h(X) is represented by all the data, and by an algorithm

Learning Bayes Classifier



- Training data (discrete case):

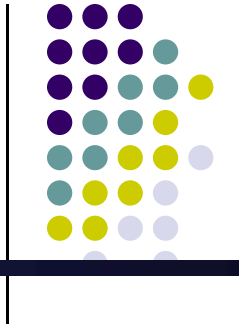


Sky	Temp	Humid	Wind	Water	Forecst	EnjoySpt
Sunny	Warm	Normal	Strong	Warm	Same	Yes
Sunny	Warm	High	Strong	Warm	Same	Yes
Rainy	Cold	High	Strong	Warm	Change	No
Sunny	Warm	High	Strong	Cool	Change	Yes

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

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

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



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

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

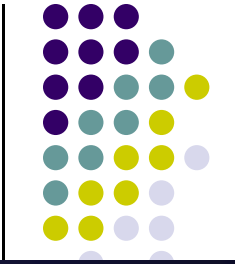
- Maximum likelihood estimation (MLE)
  1. One of the most common estimators
  2. With iid and full-observability assumption, write $L(\theta)$ as the likelihood of the data:

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

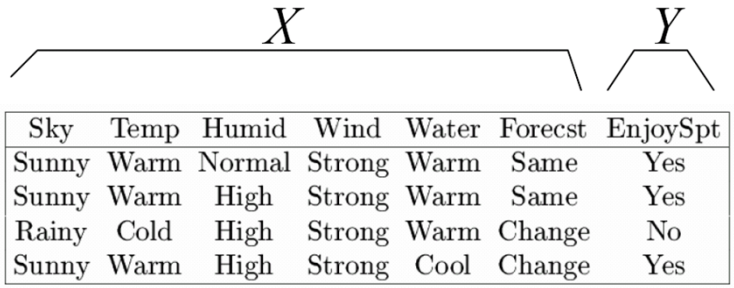
  3. pick the setting of parameters most likely to have generated the data we saw:

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

How hard is it to learn the optimal classifier?



- How do we represent these? How many parameters?
  - Prior, P(Y):
    - Suppose Y is composed of *k* classes

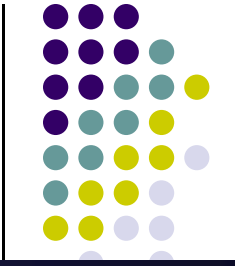


Sky	Temp	Humid	Wind	Water	Forecst	EnjoySpt
Sunny	Warm	Normal	Strong	Warm	Same	Yes
Sunny	Warm	High	Strong	Warm	Same	Yes
Rainy	Cold	High	Strong	Warm	Change	No
Sunny	Warm	High	Strong	Cool	Change	Yes

  - Likelihood, P(**X**|Y):
    - Suppose **X** is composed of *n* binary features

- Complex model ! High variance with limited data!!!

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

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

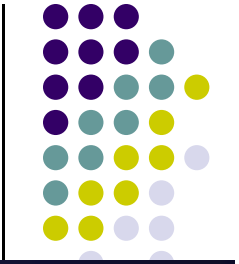
- e.g.,

$$P(Thunder | Rain, Lightning) = P(Thunder | Lightning)$$

- Equivalent to:

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

The Naïve Bayes assumption



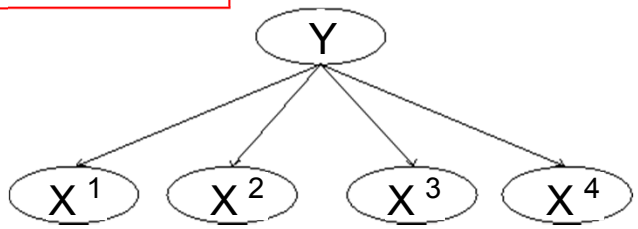
- Naïve Bayes assumption:
  - Features are conditionally independent given class:

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

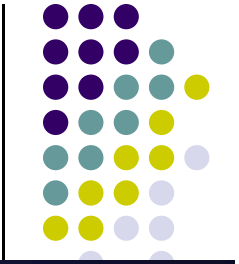
  - More generally:

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

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



The Naïve Bayes Classifier



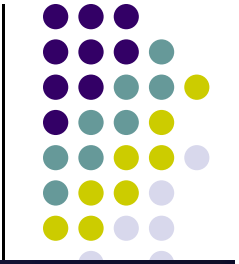
- Given:
  - Prior P(Y)
  - *m* conditionally independent features **X** given the class Y
  - For each X_n, we have likelihood $P(X_n|Y)$

- Decision rule:

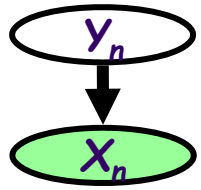
$$y^* = h_{NB}(\mathbf{x}) = \arg\max_y P(y)P(x^1,\ldots,x^m \mid y)$$
$$= \arg\max_y P(y)\prod_i P(x^i|y)$$

- If assumption holds, NB is optimal classifier!

Gaussian Discriminative Analysis



- learning f: X $\rightarrow$ Y, where
  - X is a vector of real-valued features, $\mathbf{X}_n = < X_n^1, \ldots X_n^m >$
  - 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:



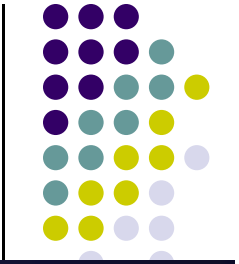
$$p(\mathbf{x}_n, y_n^k = 1 \mid \mu, \sigma) = p(y_n^k = 1) \times p(\mathbf{x}_n \mid y_n^k = 1, \mu, \Sigma)$$

$$= \pi_k \frac{1}{(2\pi|\Sigma|)^{1/2}} \exp\left\{-\tfrac{1}{2}(\mathbf{x}_n - \vec{\mu}_k)^T \Sigma^{-1}(\mathbf{x}_n - \vec{\mu}_k)\right\}$$

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

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

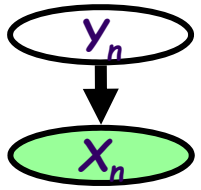
The A Gaussian Discriminative Naïve Bayes Classifier



- When X is multivariate-Gaussian vector:
  - The joint probability of a datum and it label is:

$$p(\mathbf{x}_n, y_n^k = 1 \mid \vec{\mu}, \Sigma) = p(y_n^k = 1) \times p(\mathbf{x}_n \mid y_n^k = 1, \vec{\mu}, \Sigma)$$

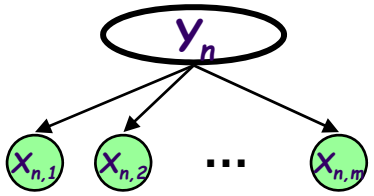
$$= \pi_k \frac{1}{(2\pi|\Sigma|)^{1/2}} \exp\left\{-\tfrac{1}{2}(\mathbf{x}_n - \vec{\mu}_k)^T \Sigma^{-1} (\mathbf{x}_n - \vec{\mu}_k)\right\}$$



  - The naïve Bayes simplification

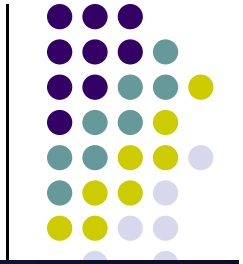
$$p(\mathbf{x}_n, y_n^k = 1 \mid \mu, \sigma) = p(y_n^k = 1) \times \prod_j p(x_n^j \mid y_n^k = 1, \mu_k^j, \sigma_k^j)$$

$$= \pi_k \prod_j \frac{1}{\sqrt{2\pi}\sigma_k^j} \exp\left\{-\tfrac{1}{2}\left(\frac{x_n^j - \mu_k^j}{\sigma_k^j}\right)^2\right\}$$



  - More generally: $p(\mathbf{x}_n, y_n \mid \eta, \pi) = p(y_n \mid \pi) \times \prod_{j=1}^{m} p(x_n^j \mid y_n, \eta)$
    - Where $p(. \mid .)$ is an arbitrary conditional (discrete or continuous) 1-D density

The predictive distribution



- Understanding the predictive distribution

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

- Under naïve Bayes assumption:

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

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

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

$$= \frac{1}{1 + e^{-\theta^T x_n}}$$

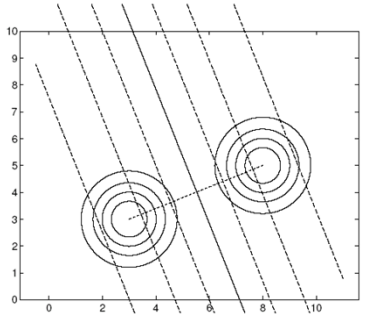
The decision boundary

- The predictive distribution

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

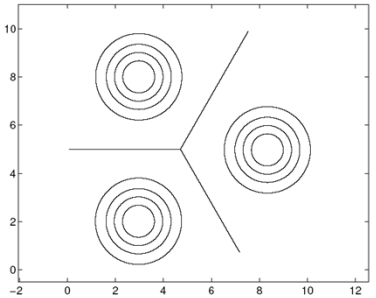
- The Bayes decision rule:

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

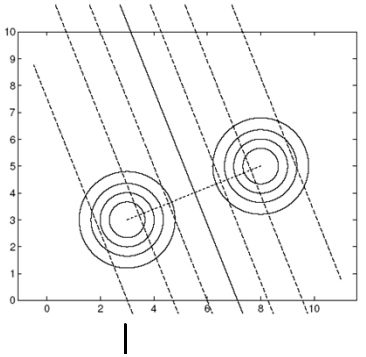


- For multiple class (i.e., $K>2$), * correspond to a softmax function

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



Summary:
The Naïve Bayes Algorithm



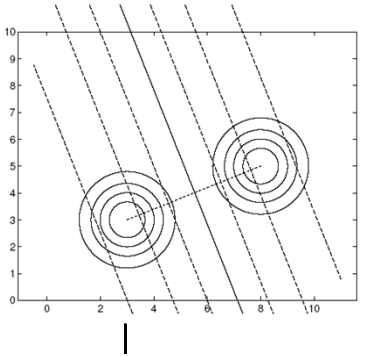
- Train Naïve Bayes (examples)
  - for each* value y_k
  - estimate $\pi_k \equiv P(Y = y_k)$
  - for each* value x_{ij} of each attribute X_i
  - estimate $\theta_{ijk} \equiv P(X^i = x_{ij}|Y = y_k)$

- Classify (X_{new})

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

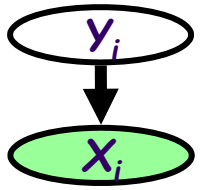
$$Y^{new} \leftarrow \arg\max_{y_k} \pi_k \prod_i \theta_{ijk}$$

Generative vs. Discriminative Classifiers

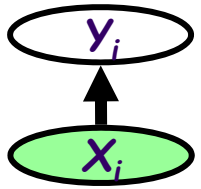


- Goal: Wish to learn f: X $\rightarrow$ Y, e.g., P(Y|X)

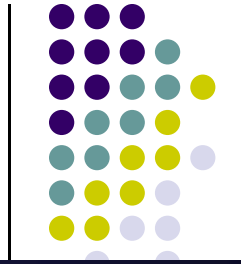
- Generative classifiers (e.g., Naïve Bayes):
  - Assume some functional form for P(X|Y), P(Y)
    This is a '***generative***' model of the data!
  - Estimate parameters of P(X|Y), P(Y) directly from training data
  - Use Bayes rule to calculate P(Y|X= x)



- 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



Recall the NB predictive distribution



- Understanding the predictive distribution

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

- Under naïve Bayes assumption:

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

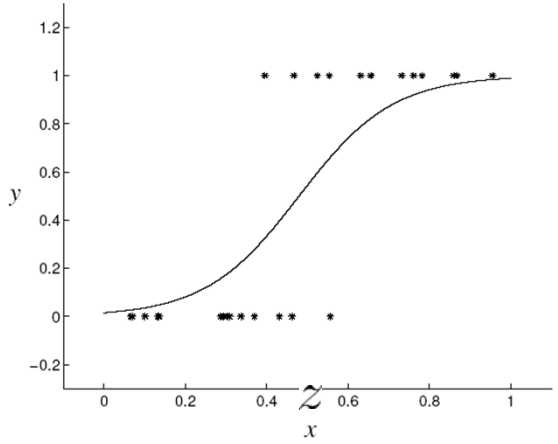
- For two class (i.e., *K*=2), and when the two classes haves the same variance, ** turns out to be a logistic function

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

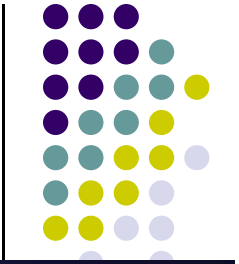
$$= \frac{1}{1 + e^{-\theta^T x_n}}$$

The logistic function

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



Logistic regression (sigmoid classifier)

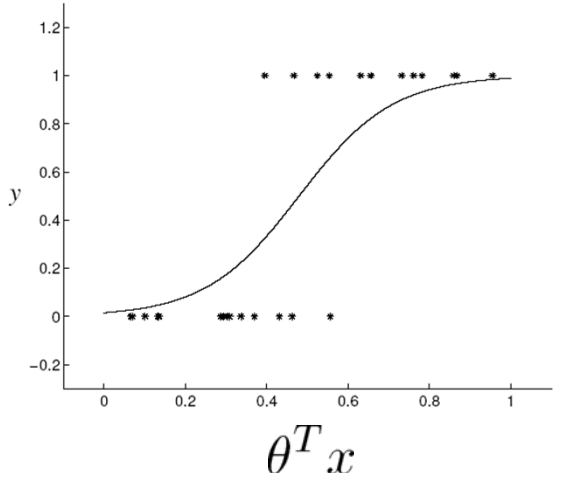


- The condition distribution: a Bernoulli

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

where μ is a logistic function

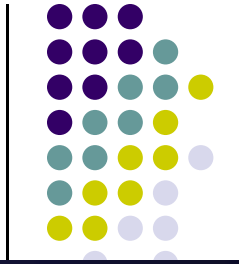
$$\mu(x) = \frac{1}{1+e^{-\theta^T x}} = p(y=1 \mid x)$$



- In this case, learning p(y|x) amounts to learning ...?

- What is the difference to NB?

Training Logistic Regression: MCLE



- Estimate parameters θ=<θ_0, θ_1, ... θ_m> to maximize the **conditional likelihood** of training data

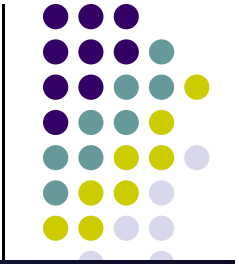
- Training data $\mathcal{D} = \{(x_1, y_1), \ldots, (x_N, y_N)\}$

- 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



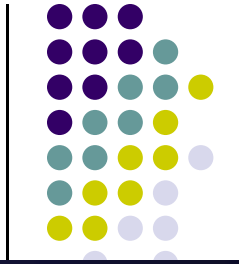
$$l(\theta) \equiv \ln \prod_i P(y_i|x_i;\theta) = \sum_i \ln P(y_i|x_i;\theta)$$

- 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



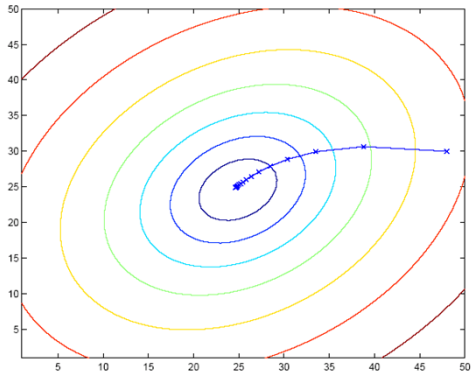
- The objective:

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

- Good news: $l(\theta)$ is concave function of θ

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

Gradient Ascent



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

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

- Property of sigmoid function:

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

- The gradient:

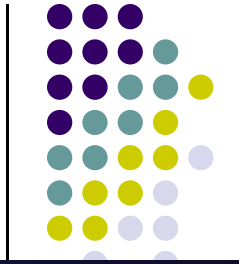
$$\frac{\partial l(\theta)}{\partial \theta_j} =$$

The gradient ascent algorithm iterate until change < ε

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

repeat

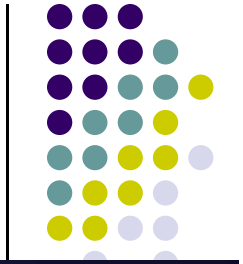
The Newton's method



- Finding a zero of a function

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

The Newton's method (con'd)



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

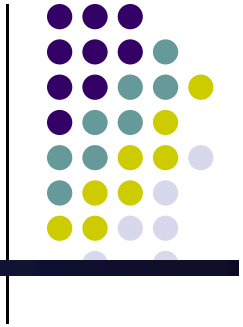
$$l(\theta) = \sum_i (y_i - 1)\theta^T x_i + \ln(1 + e^{-\theta^T x_i})$$

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

- So we can perform the following iteration:

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

The Newton-Raphson method



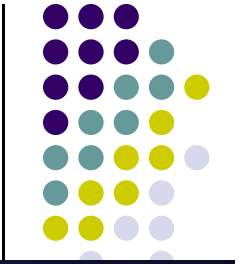
- In LR the θ is vector-valued, thus we need the following generalization:

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

- ∇ is the gradient operator over the function

- H is known as the Hessian of the function

The Newton-Raphson method



- In LR the θ is vector-valued, thus we need the following generalization:

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

- ∇ is the gradient operator over the function

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

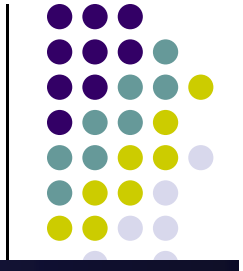
- H is known as the Hessian of the function

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

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

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

Iterative reweighed least squares (IRLS)



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

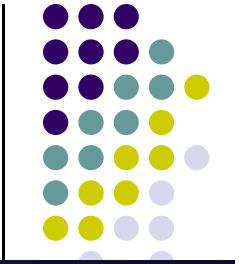
$$\theta = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

which can also 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

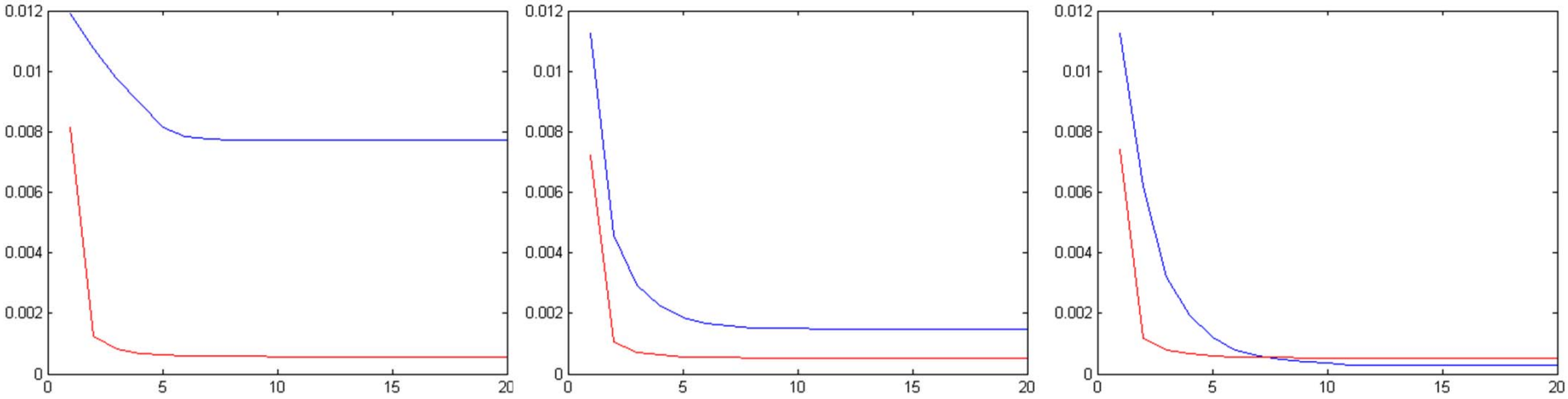
- Now for logistic regression:

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

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

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

Convergence curves



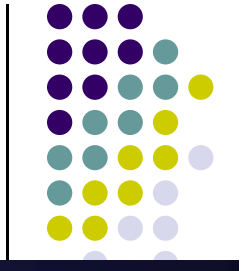
alt.atheism
vs.
comp.graphics

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



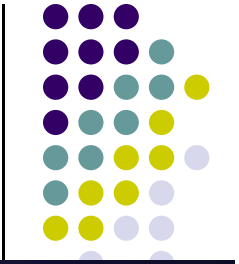
- NR (IRLS) takes $O(N+d^3)$ per iteration, where N = number of training cases and d = dimension of input x, but converge in fewer iterations

- Quasi-Newton methods, that approximate the Hessian, work faster.

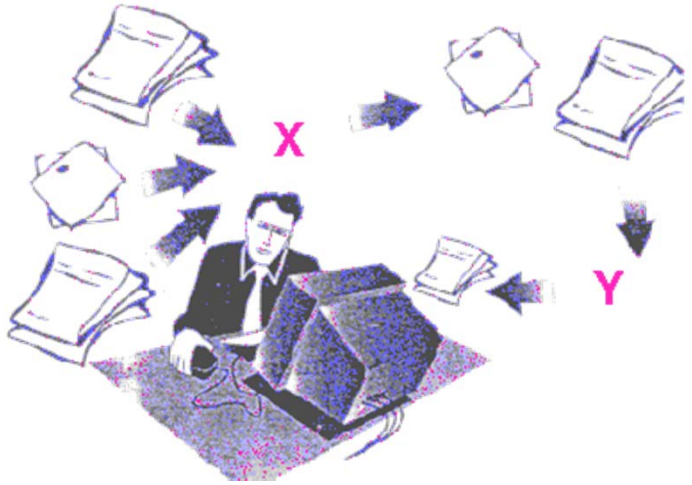
- Conjugate gradient takes $O(Nd)$ per iteration, and usually works best in practice.

- Stochastic gradient descent can also be used if N is large c.f. perceptron rule:

Case Study: Text classification



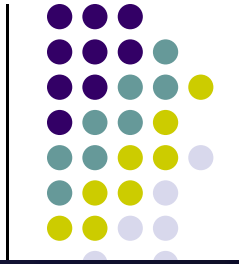
- Classify e-mails
  - Y = {Spam,NotSpam}
- Classify news articles
  - Y = {what is the topic of the article?}
- Classify webpages
  - Y = {Student, professor, project, …}
- What about the features **X**?
  - The text!



Features X are entire document – X^i for i^{th} word in article

aardvark	0
about	2
all	2
Africa	1
apple	0
anxious	0
...	
gas	1
...	
oil	1
…	
Zaire	0

Bag of words model

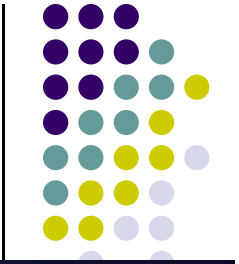


- Typical additional assumption – **Position in document doesn't matter**: $P(X^i=x^i|Y=y) = P(X^k=x^i|Y=y)$
  - "Bag of words" model – order of words on the page ignored
  - Sounds really silly, but often works very well!

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

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

Bag of words model

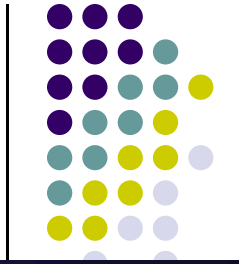


- Typical additional assumption – **Position in document doesn't matter**: $P(X^i=x^i|Y=y) = P(X^k=x^i|Y=y)$
  - "Bag of words" model – order of words on the page ignored
  - Sounds really silly, but often works very well!

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

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

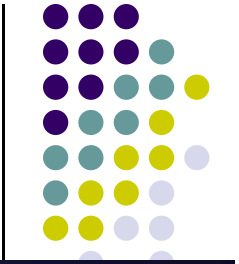
NB with Bag of Words for text classification



- 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 $\mathbf{x}_{new}$
    - Use naïve Bayes decision rule

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

Back to our 20 NG Case study



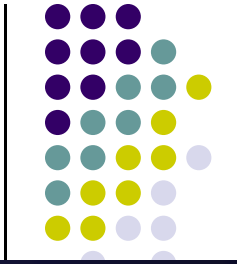
- Dataset
  - 20 News Groups (20 classes)
  - 61,118 words, 18,774 documents

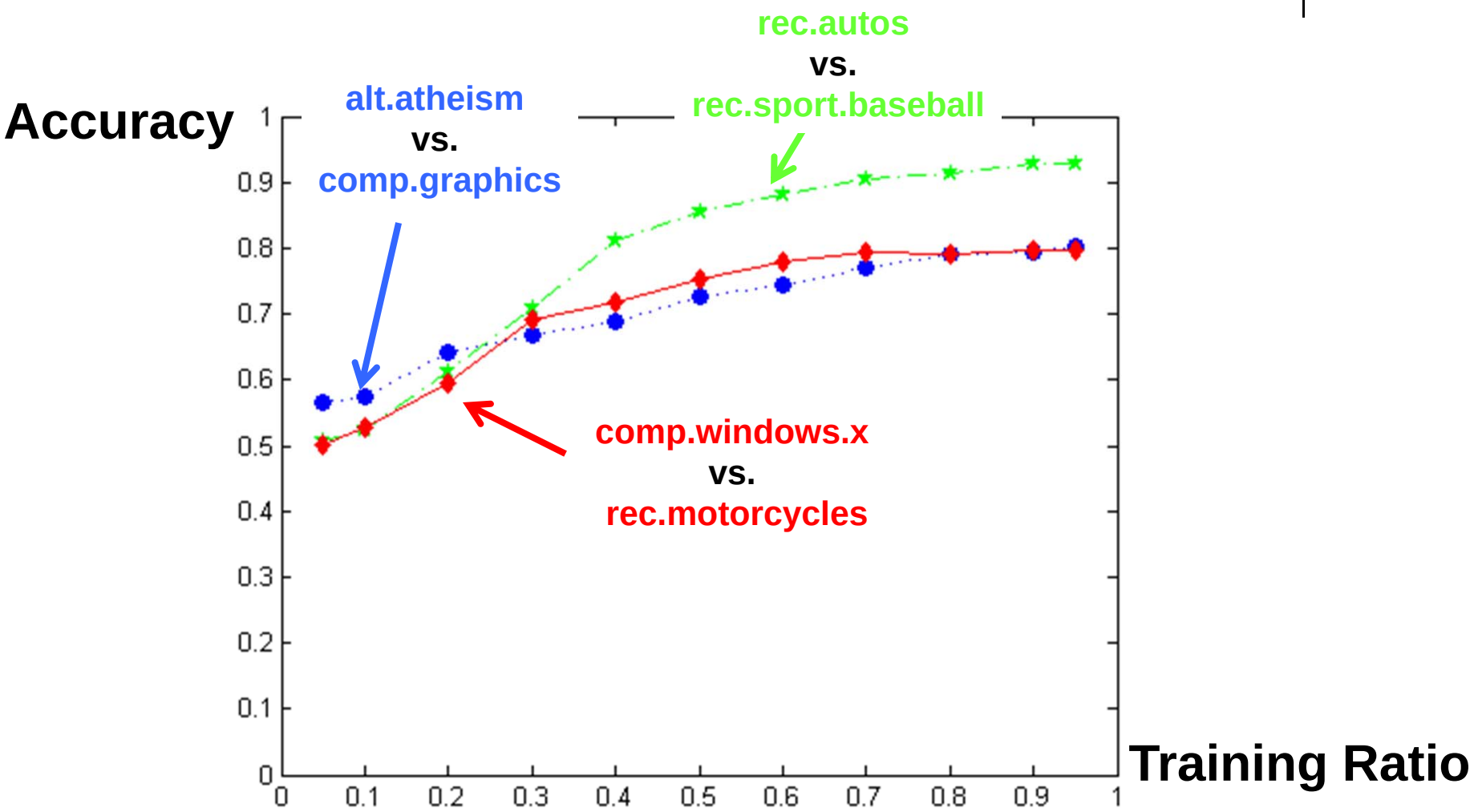
comp.graphics comp.os.ms-windows.misc comp.sys.ibm.pc.hardware comp.sys.mac.hardware comp.windows.x	rec.autos rec.motorcycles rec.sport.baseball rec.sport.hockey	sci.crypt sci.electronics sci.med sci.space
misc.forsale	talk.politics.misc talk.politics.guns talk.politics.mideast	talk.religion.misc alt.atheism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

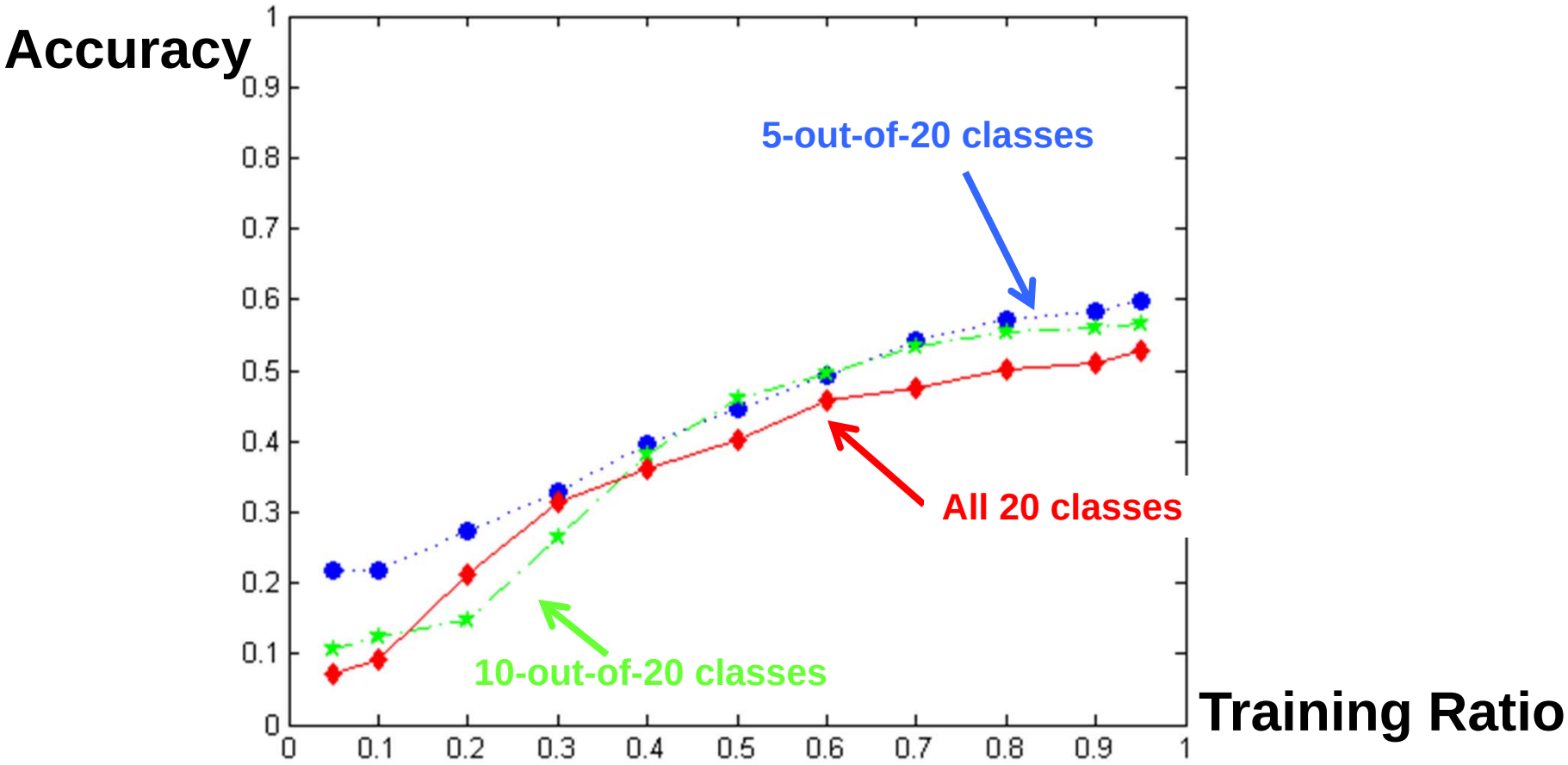
$$Accuracy = \frac{\sum_{i \in test\ set} \mathrm{I}(predict_i == true\ label_i)}{\#\ of\ test\ samples}$$

Results: Binary Classes



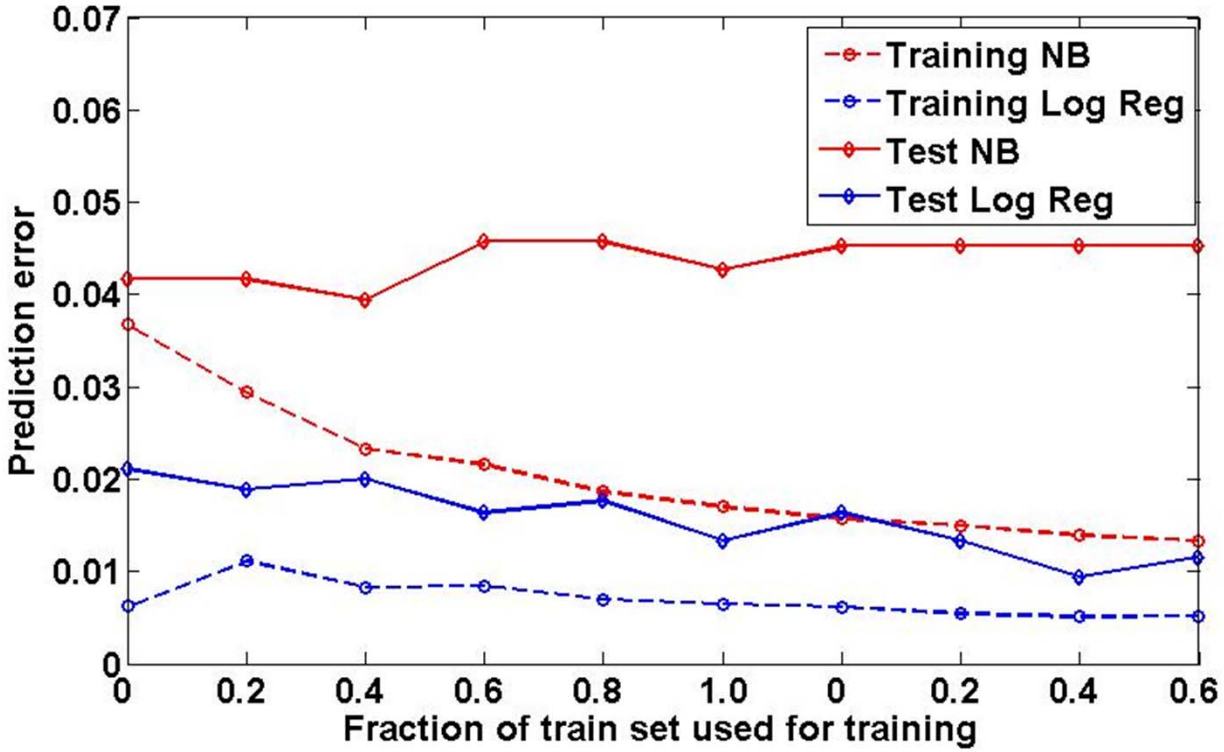


Results: Multiple Classes



NB vs. LR

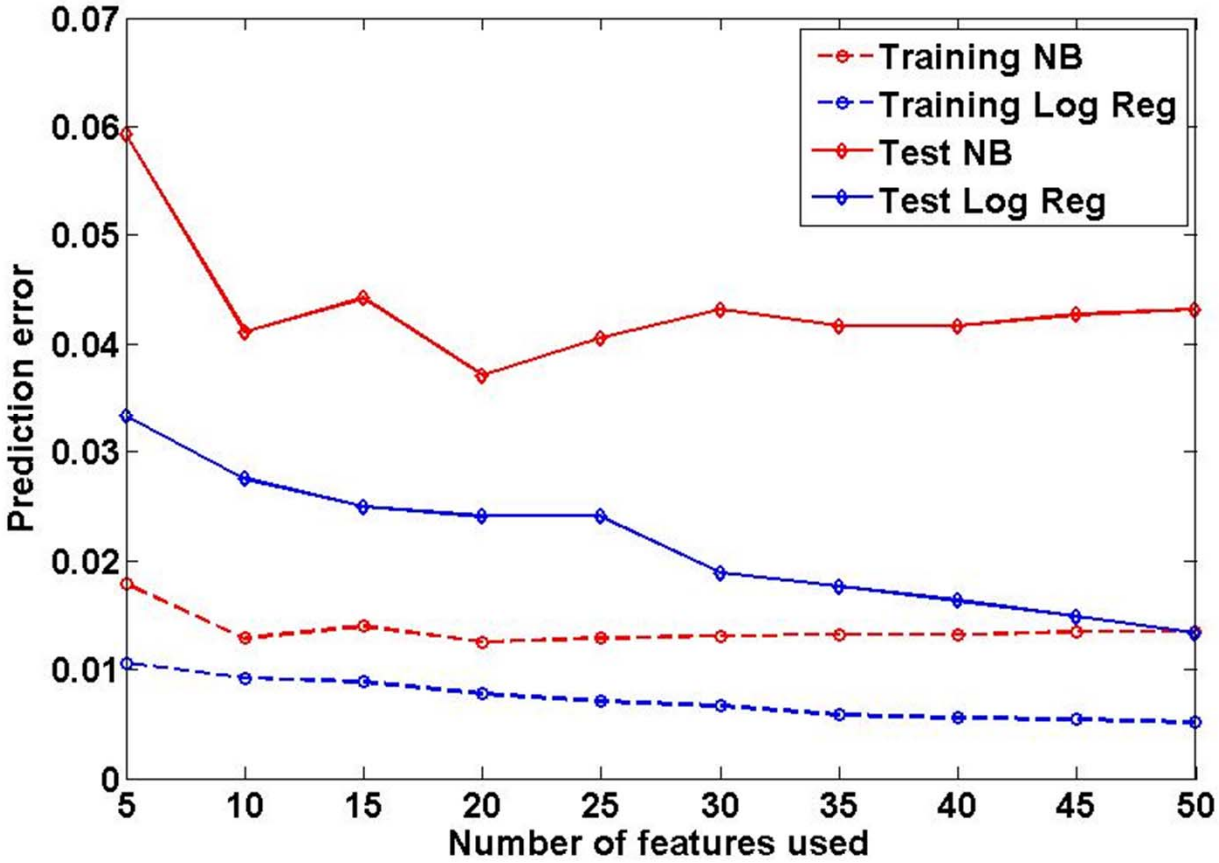
- Versus training size



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

NB vs. LR

- Versus model size

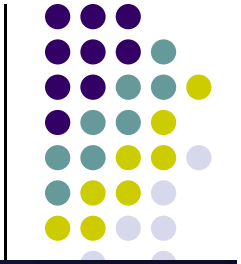


Number of dimensions of the data varied from 5 to 50 in steps of 5

The features were chosen in decreasing order of their singular values

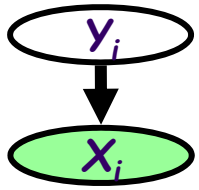
90% versus 10% split on training and test

Summary:
Generative vs. Discriminative Classifiers

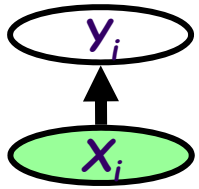


- Goal: Wish to learn f: X $\rightarrow$ Y, e.g., P(Y|X)

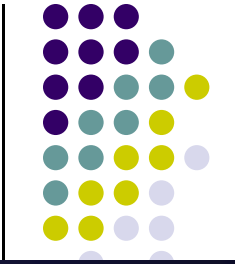
- Generative classifiers (e.g., Naïve Bayes):
  - Assume some functional form for P(X|Y), P(Y)
    This is a '***generative***' model of the data!
  - Estimate parameters of P(X|Y), P(Y) directly from training data
  - Use Bayes rule to calculate P(Y|X= x)



- 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=<X^1 ... X^m>
- Number of parameters to estimate:

NB:

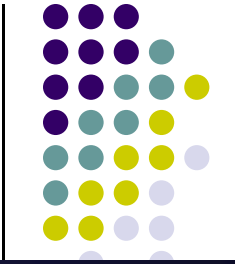
$$p(y \mid \mathbf{x}) = \frac{\pi_k \exp\left\{-\sum_j \left(\frac{1}{2\sigma_{k,j}^2}(x_j - \mu_{k,j})^2 - \log \sigma_{k,j} - C\right)\right\}}{\sum_{k'} \pi_{k'} \exp\left\{-\sum_j \left(\frac{1}{2\sigma_{k',j}^2}(x_j - \mu_{k',j})^2 - \log \sigma_{k',j} - C\right)\right\}} \quad **$$

LR:

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

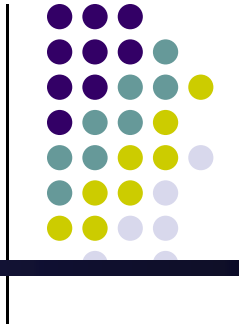
- Estimation method:
  - NB parameter estimates are uncoupled
  - LR parameter estimates are coupled

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])

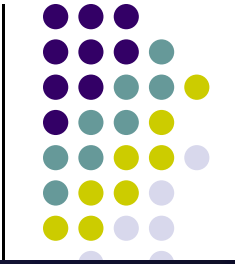
- convergence rate of parameter estimates – how many training examples needed to assure good estimates?

  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:

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

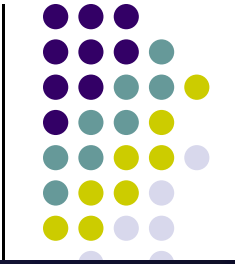
- Implication: if we want $\epsilon(h_{Dis,m}) \leq \epsilon(h_{Dis,\infty}) + \epsilon_0$

  for some small constant ε_0, it suffices to pick order *m* examples

  → Convergences 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 ε_1, δ>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-δ, after n examples:

1. For discrete input, $|\hat{p}(x_i|y = b) - p(x_i|y = b)| \leq \epsilon_1$ for all i and b
   $|\hat{p}(y = b) - p(y = b)| \leq \epsilon_1$
2. For continuous inputs, $|\hat{\mu}_{i|y=b} - \mu_{i|y=b}| \leq \epsilon_1$ for all i and b
   $|\hat{\sigma}^2_{i|y=b} - \sigma^2_{i|y=b}| \leq \epsilon_1$

Some experiments from UCI data sets

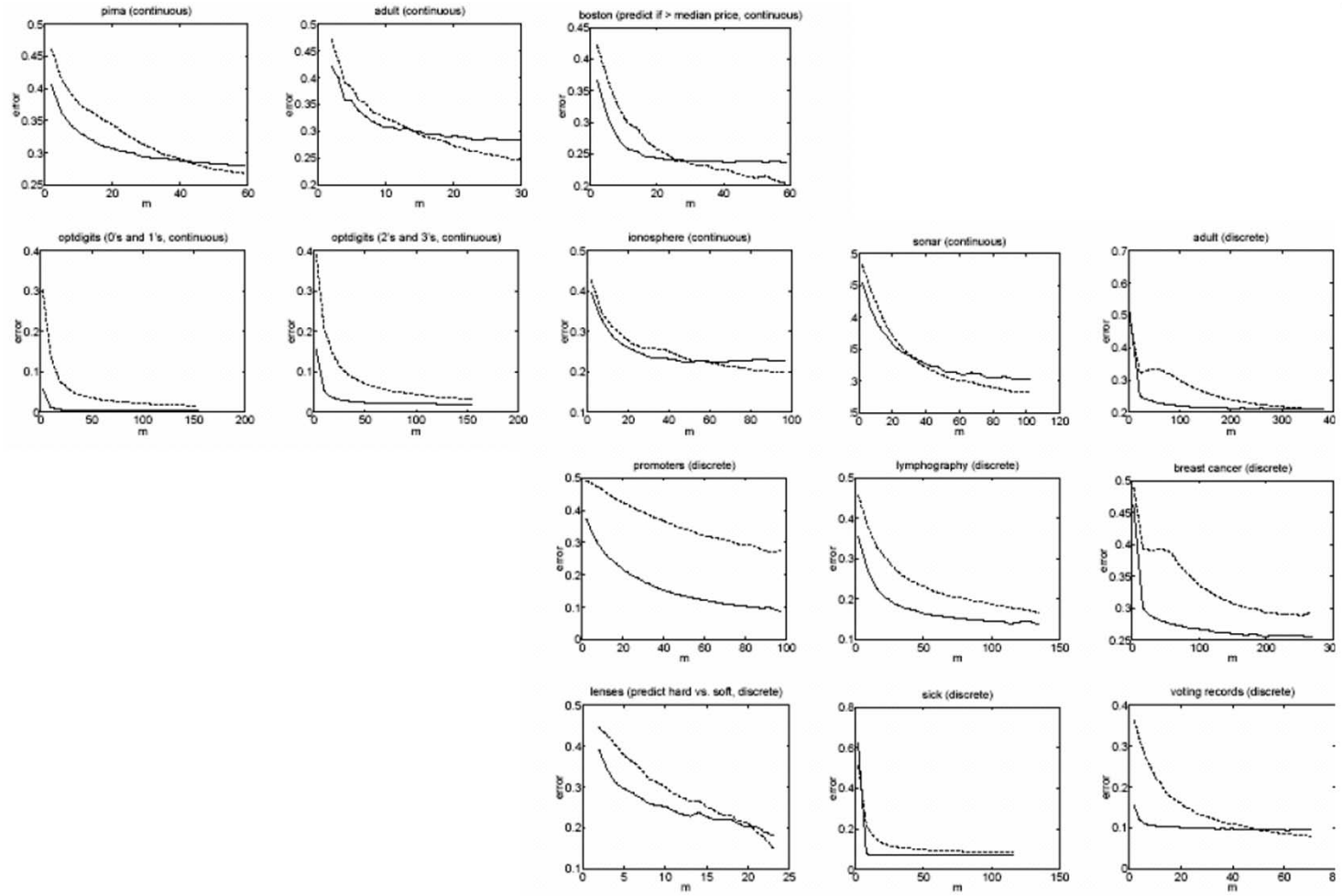
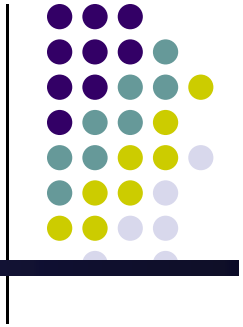


Figure 1: Results of 15 experiments on datasets from the UCI Machine Learnin repository. Plots are of generalization error vs. m (averaged over 1000 rando train/test splits). Dashed line is logistic regression; solid line is naive Bayes.

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