# Machine Learning

10-701/15-781, Fall 2012

## “Nonparametric” methods

## -- instance based learning

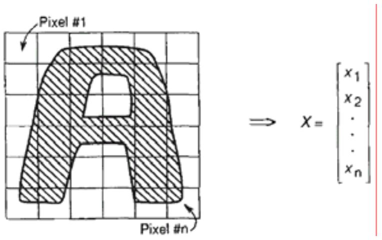
Eric Xing

Lecture 6, September 26, 2012

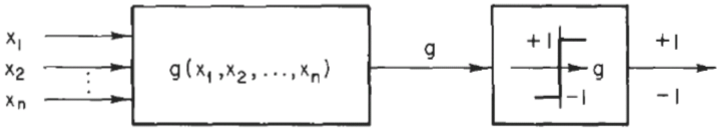
Reading:

# Classification

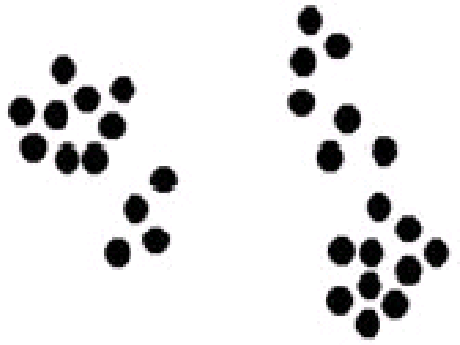
- Representing data:


Pixel #1
Pixel #n
⟹
$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$

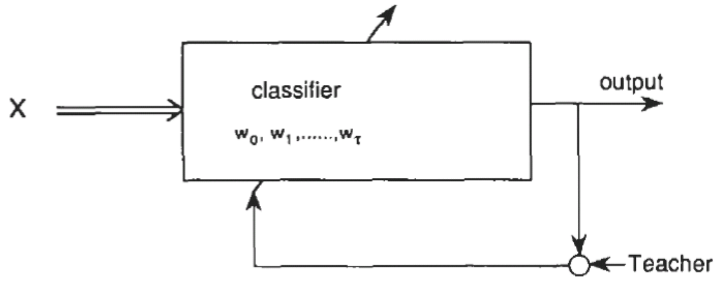
- Hypothesis (classifier)

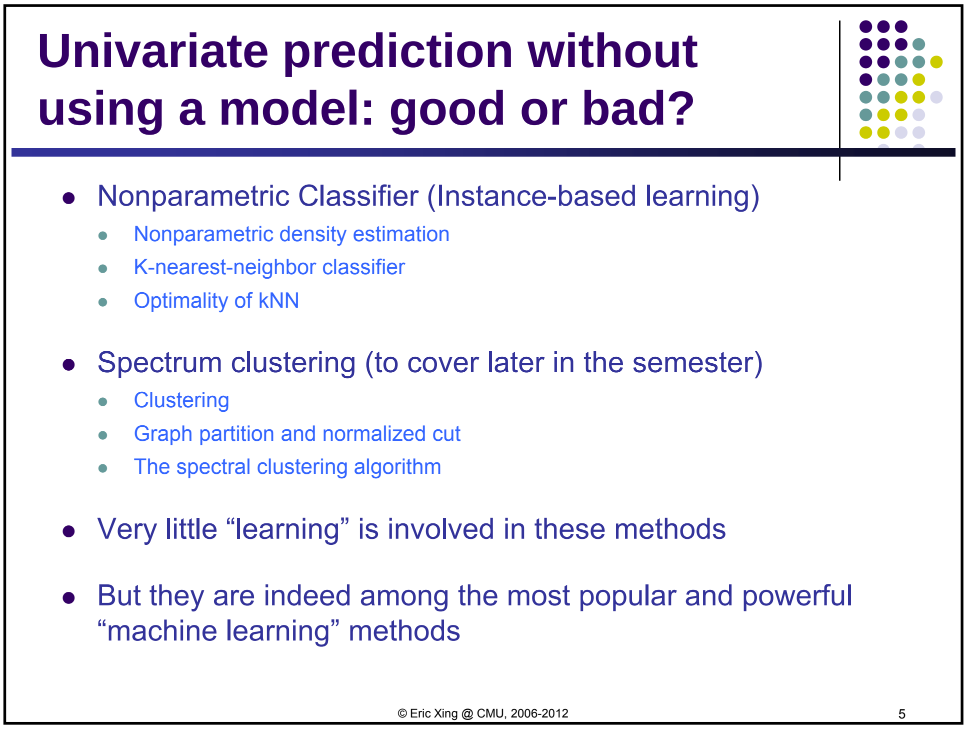


## Clustering



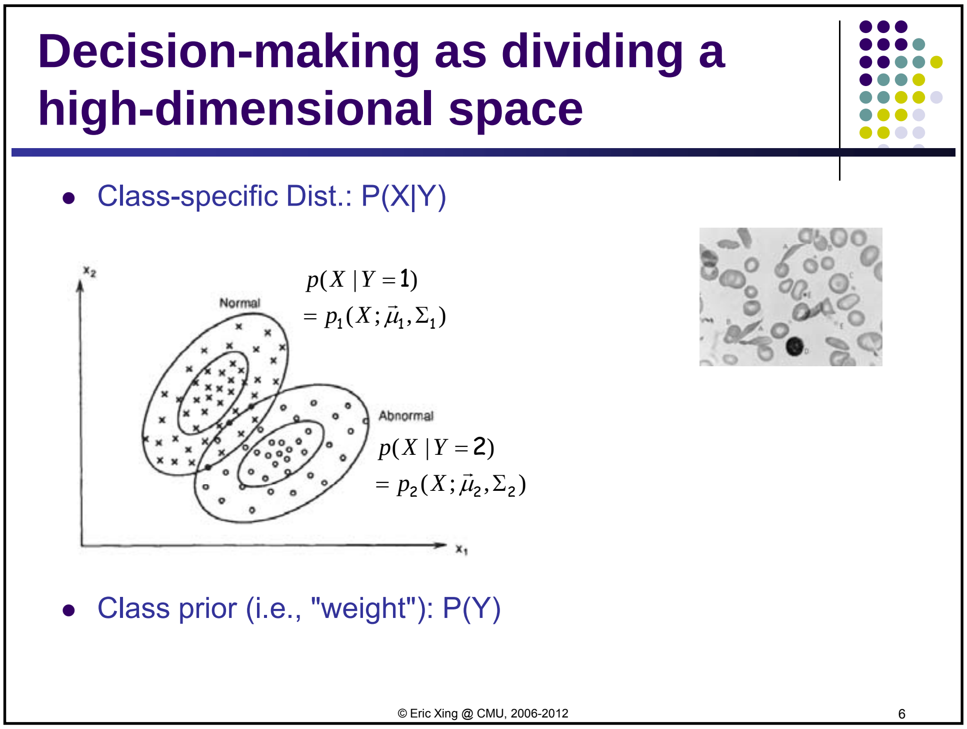
## Supervised vs. Unsupervised Learning





## Univariate prediction without using a model: good or bad?

- Nonparametric Classifier (Instance-based learning)
  - Nonparametric density estimation
  - K-nearest-neighbor classifier
  - Optimality of kNN
- Spectrum clustering (to cover later in the semester)
  - Clustering
  - Graph partition and normalized cut
  - The spectral clustering algorithm
- Very little “learning” is involved in these methods
- But they are indeed among the most popular and powerful “machine learning” methods



## Decision-making as dividing a high-dimensional space

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

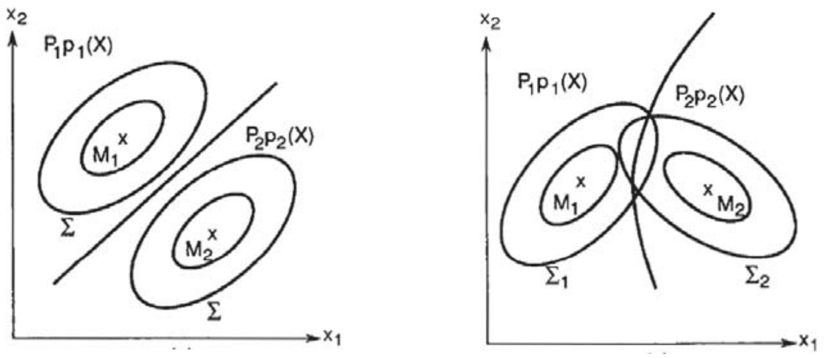
$$p(X \mid Y=1) = p_1(X;\vec{\mu}_1,\Sigma_1)$$

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

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

# Example of Decision Rules

- When each class is a normal …



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

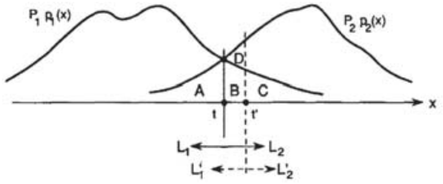
# Bayes Error

- 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*?

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

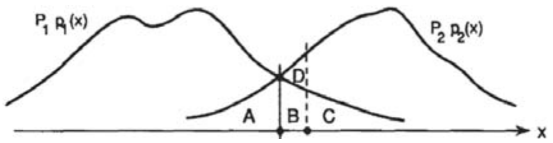
- The Bayes error (the expected risk):

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



## 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:

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

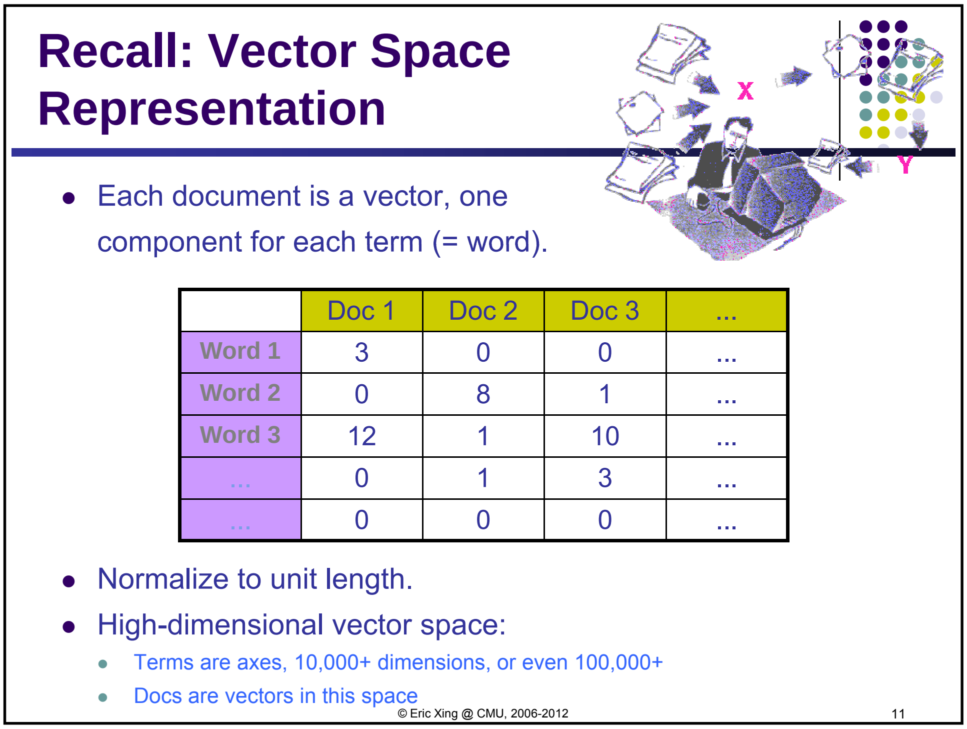
## Learning Classifier

- The decision rule:

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

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

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

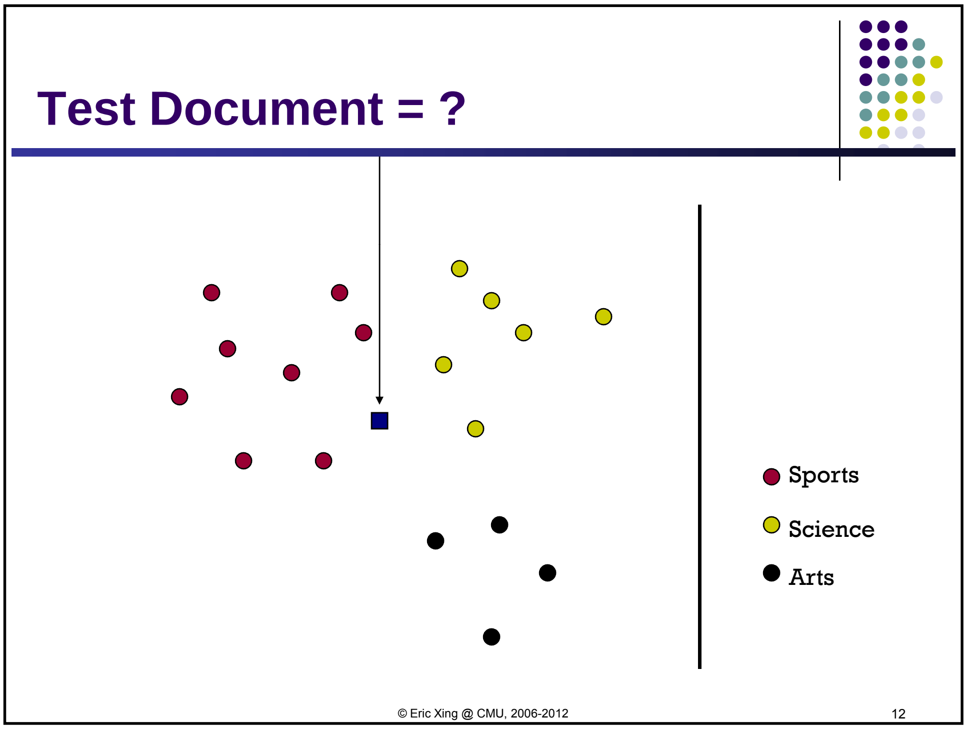


# Recall: Vector Space Representation

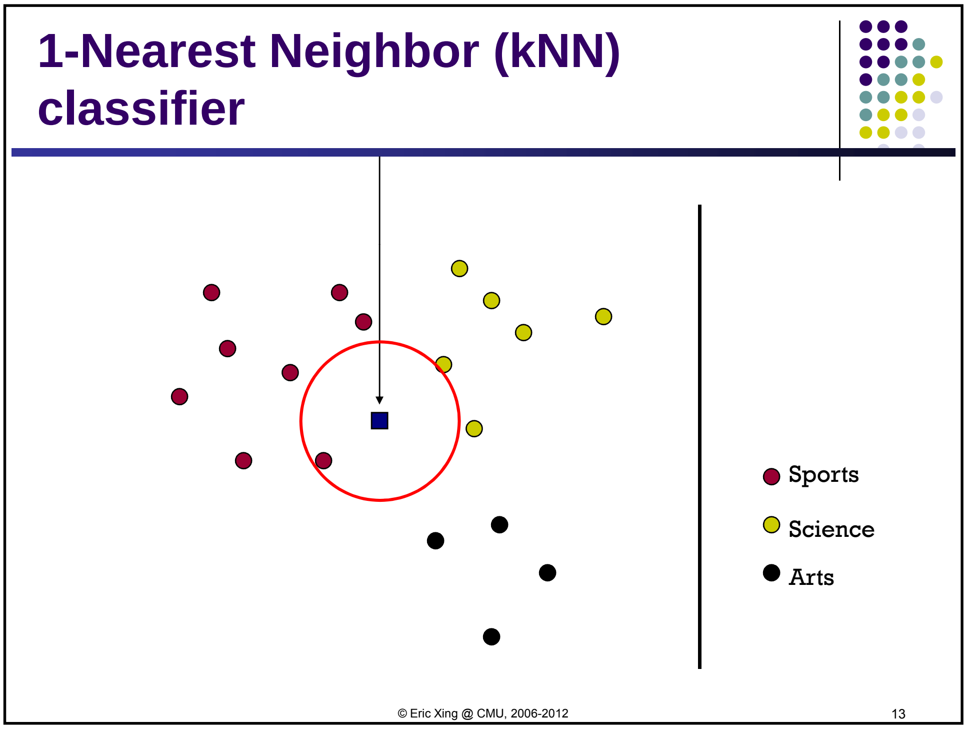
- Each document is a vector, one component for each term (= word).

| | Doc 1 | Doc 2 | Doc 3 | ... |
|---|---|---|---|---|
| Word 1 | 3 | 0 | 0 | ... |
| Word 2 | 0 | 8 | 1 | ... |
| Word 3 | 12 | 1 | 10 | ... |
| ... | 0 | 1 | 3 | ... |
| ... | 0 | 0 | 0 | ... |

- Normalize to unit length.
- High-dimensional vector space:
  - Terms are axes, 10,000+ dimensions, or even 100,000+
  - Docs are vectors in this space



# Test Document = ?

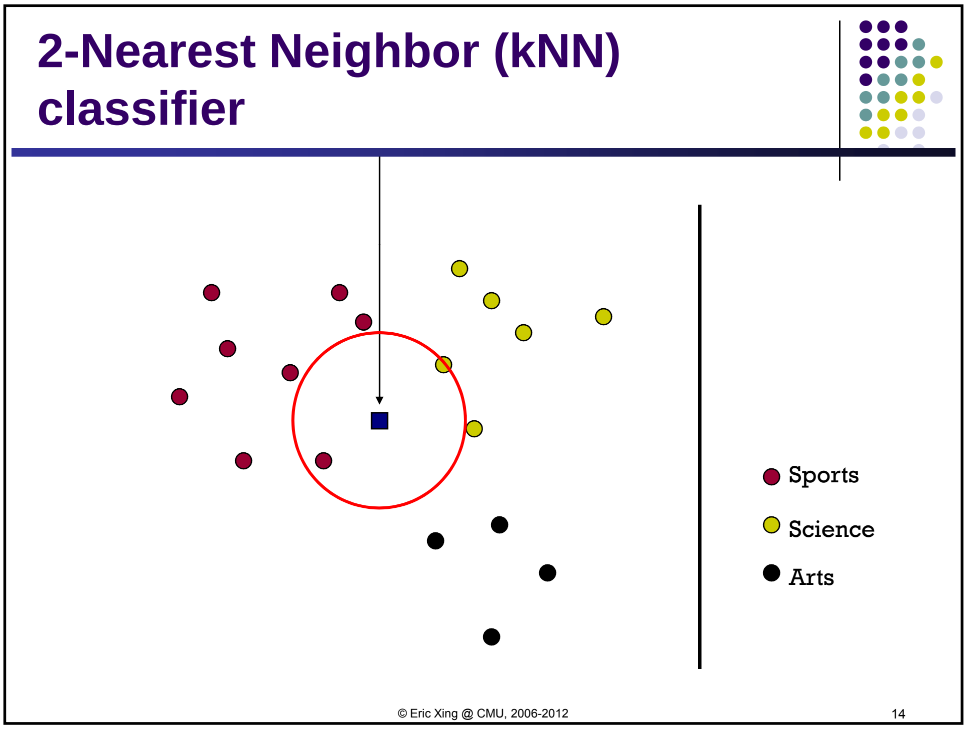


# 1-Nearest Neighbor (kNN) classifier

● Sports

○ Science

● Arts

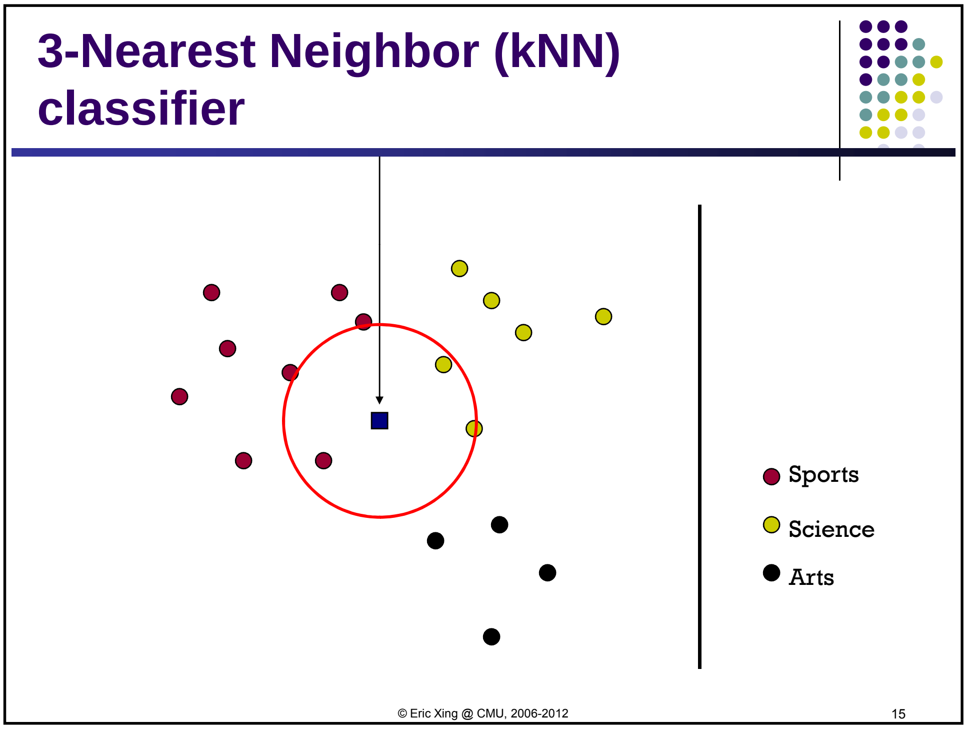


# 2-Nearest Neighbor (kNN) classifier

● Sports

○ Science

● Arts

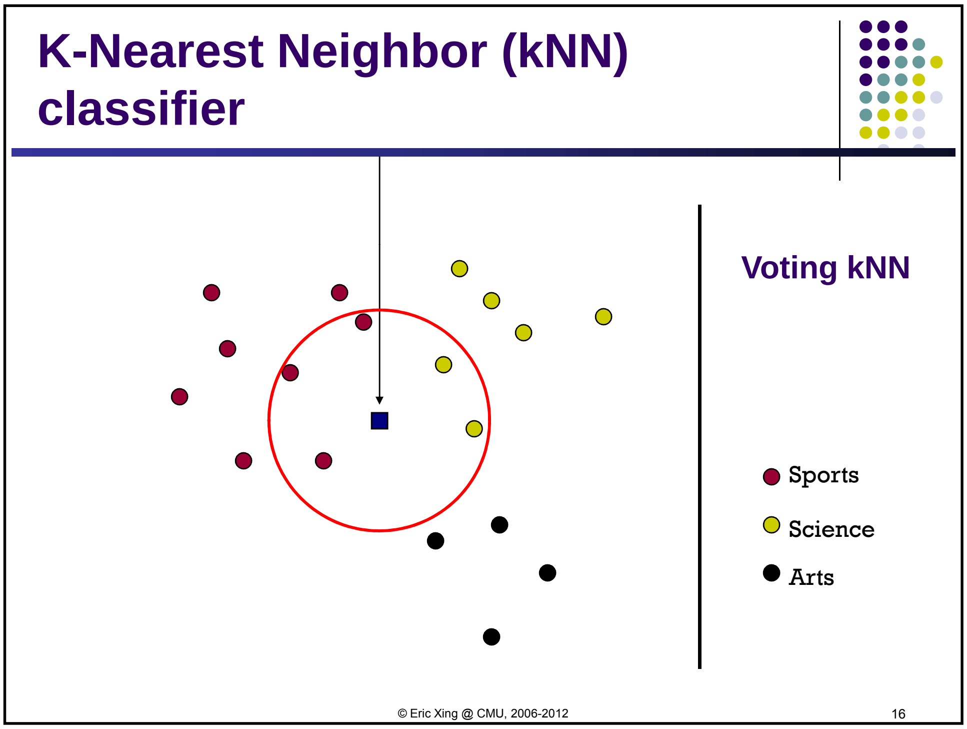


# 3-Nearest Neighbor (kNN) classifier

● Sports

● Science

● Arts



# K-Nearest Neighbor (kNN) classifier

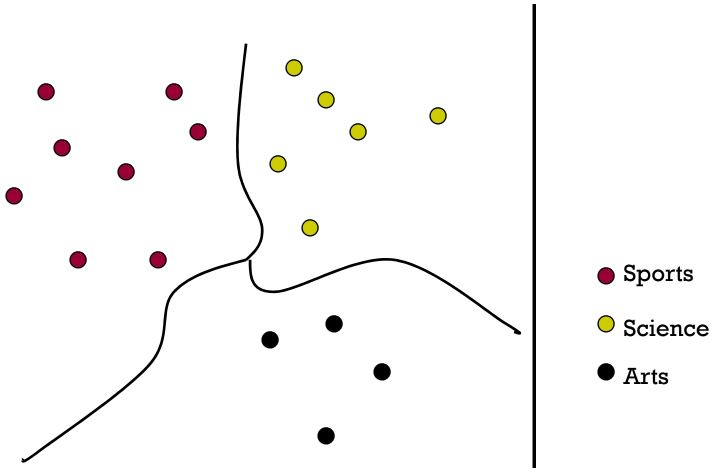
Voting kNN

● Sports

● Science

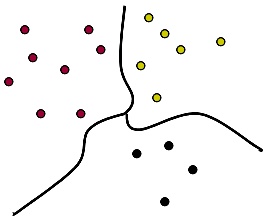
● Arts

# Classes in a Vector Space



# kNN Is Close to Optimal

- Cover and Hart 1967
- Asymptotically, the error rate of 1-nearest-neighbor classification is less than twice the Bayes rate [error rate of classifier knowing model that generated data]
- In particular, asymptotic error rate is 0 if Bayes rate is 0.
- Decision boundary:



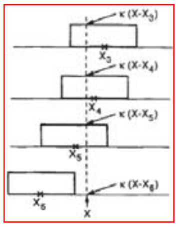
## Where does kNN come from?

- How to estimation $p(X)$ ?

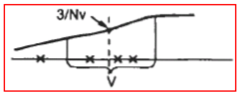
- Nonparametric density estimation
  - Parzen density estimate

    E.g. (Kernel density est.):

$$\hat{p}(X) = \frac{1}{N}\sum_{i=1}^{N}\kappa(X - x_i)$$



More generally: $\hat{p}(X) = \frac{1}{N}\frac{k(X)}{V}$



## Where does kNN come from?

- Nonparametric density estimation
  - Parzen density estimate $\hat{p}(X) = \frac{1}{N}\frac{k(X)}{V}$
  - kNN density estimate $\hat{p}(X) = \frac{1}{N}\frac{(k-1)}{V(X)}$

- Bayes classifier based on kNN density estimator:

$$h(X) \;=\; -\ln\frac{p_1(X)}{p_2(X)} = -\ln\frac{(k_1-1)N_2V_2(X)}{(k_2-1)N_1V_1(X)} \begin{matrix} > \\ < \end{matrix} \ln\frac{\pi_1}{\pi_2}$$

  - Voting kNN classifier

    Pick $K_1$ and $K_2$ implicitly by picking $K_1+K_2=K$, $V_1=V_2$, $N_1=N_2$

# Asymptotic Analysis

- Condition risk: $r_k(X,X_{NN})$
  - Test sample $X$
  - NN sample $X_{NN}$
  - Denote the event $X$ is class I as $X \leftrightarrow I$
  - Assuming $k$=1

$$\begin{aligned} r_1(X, X_{NN}) &= Pr\Big\{\{X \leftrightarrow 1 \ \& \ X_{NN} \leftrightarrow 2\} \text{ or } \{X \leftrightarrow 2 \ \& \ X_{NN} \leftrightarrow 1\} | X, X_{NN}\Big\} \\ &= Pr\Big\{\{X \leftrightarrow 1 \ \& \ X_{NN} \leftrightarrow 2\}\Big\} + Pr\Big\{\{X \leftrightarrow 2 \ \& \ X_{NN} \leftrightarrow 1\} | X, X_{NN}\Big\} \\ &= q_1(X)q_2(X_{NN}) + q_2(X)q_1(X_{NN}) \end{aligned}$$

  - When an infinite number of samples is available, $X_{NN}$ will be so close to $X$

$$r_1^*(X) = 2q_1(X)q_2(X) = 2\xi(X)$$

# Asymptotic Analysis, cont.

- Recall conditional Bayes risk:

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

$$= \frac{1}{2} - \frac{1}{2}\sqrt{1 - 4\xi(X)}$$

$$= \sum_{i=1}^{\infty} \frac{1}{i} \binom{2i-2}{i-1} \xi^i(X)$$

**This is called the MacLaurin series expansion**

- Thus the asymptotic condition risk

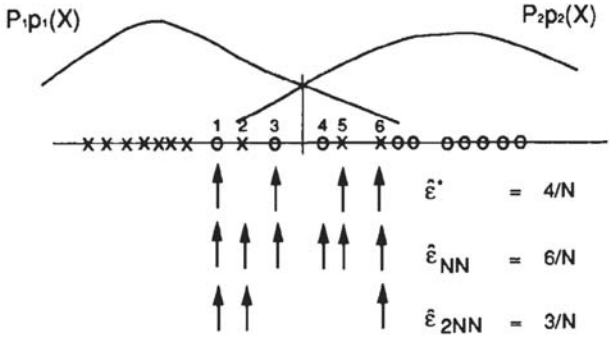
$$r_1^*(X) = 2\xi(X) \leq 2r^*(X)$$

- It can be shown that $\epsilon_1^* \leq 2\epsilon^*$
  - This is remarkable, considering that the procedure does not use any information about the underlying distributions and only the class of the single nearest neighbor determines the outcome of the decision.

# In fact

$$\frac{1}{2}\epsilon^* \leq \epsilon^*_{2NN} \leq \epsilon^*_{4NN} \leq \ldots \leq \epsilon^* \leq \ldots \leq \epsilon^*_{3NN} \leq \epsilon^*_{NN} \leq 2\epsilon^*$$

- Example:



P₁p₁(X)
P₂p₂(X)
1 2 3 4 5 6
x x x x x x x o x o o x x o o o o o o o
$\hat{\varepsilon}^*$ = 4/N
$\hat{\varepsilon}_{NN}$ = 6/N
$\hat{\varepsilon}_{2NN}$ = 3/N

# kNN is an instance of Instance-Based Learning

- What makes an Instance-Based Learner?
  - A distance metric
  - How many nearby neighbors to look at?
  - A weighting function (optional)
  - How to relate to the local points?

# Distance Metric

- Euclidean distance:

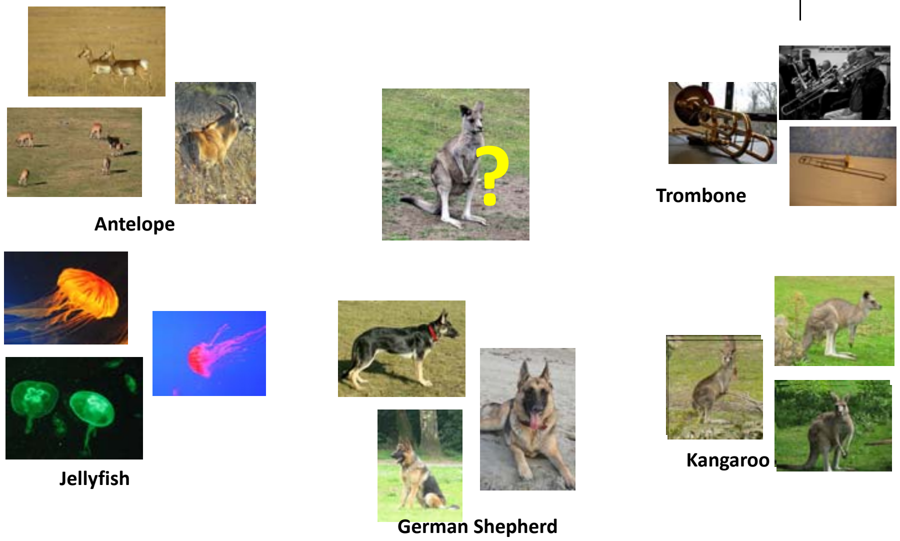
$$D(x,x') = \sqrt{\sum_i \sigma_i^2 (x_i - x_i')^2}$$

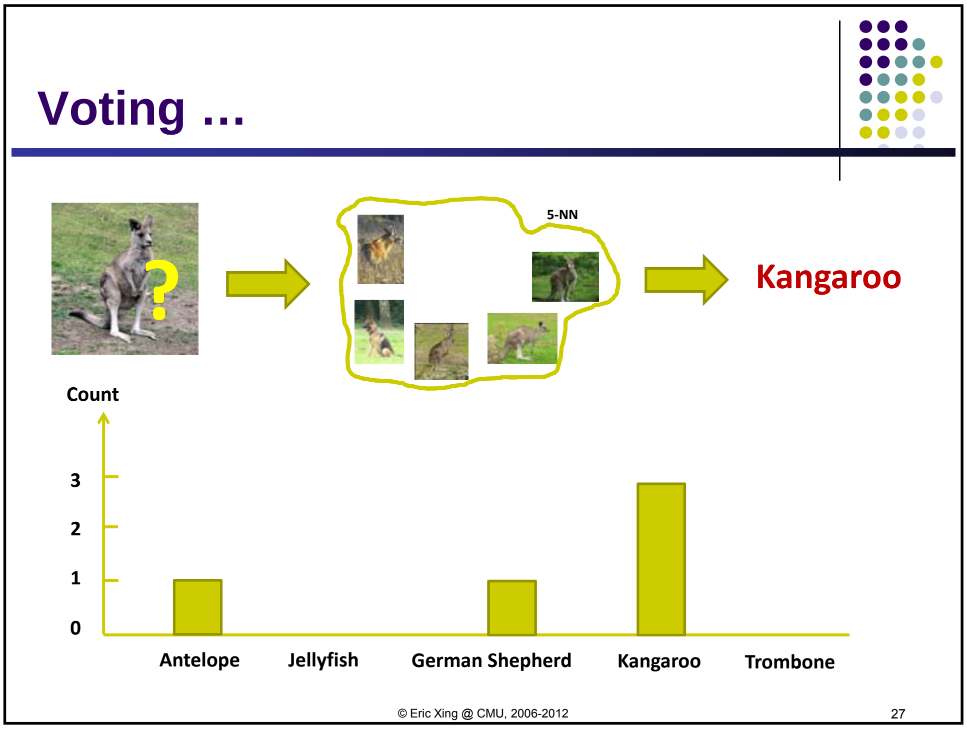
- Or equivalently,

$$D(x,x') = \sqrt{(x-x')^T \Sigma (x-x')}$$

- Other metrics:
  - $L_1$ norm: |x-x'|
  - $L_\infty$ norm: max |x-x'| (elementwise …)
  - Mahalanobis: where Σ is full, and symmetric
  - Correlation
  - Angle
  - Hamming distance, Manhattan distance
  - …

# kNN for image classification: basic set-up





## Voting …

5-NN

Kangaroo

Count

3

2

1

0

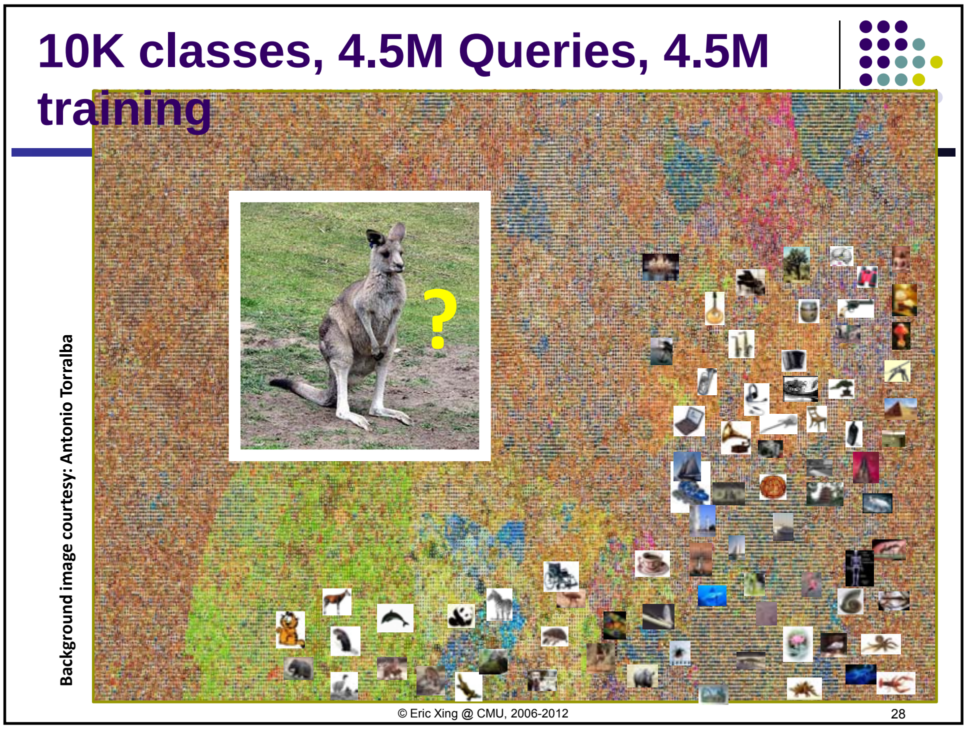
Antelope

Jellyfish

German Shepherd

Kangaroo

Trombone

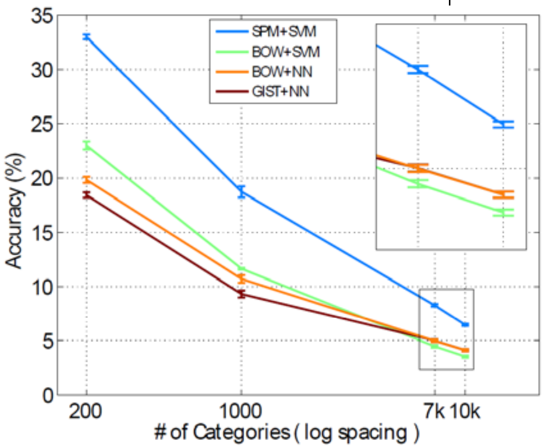


## 10K classes, 4.5M Queries, 4.5M training

Background image courtesy: Antonio Torralba

# KNN on 10K classes

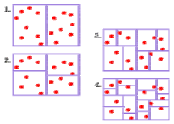
- 10K classes
- 4.5M queries
- 4.5M training
- Features
  - BOW
  - GIST



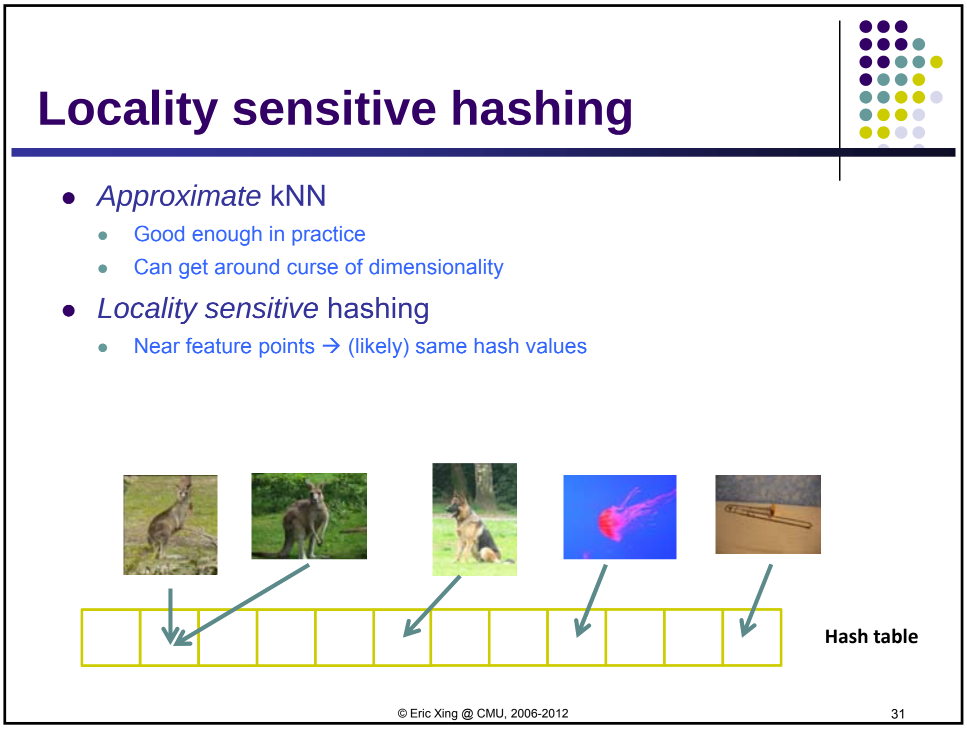
**Deng, Berg, Li & Fei-Fei, ECCV 2010**

# Nearest Neighbor Search in High Dimensional Metric Space

- Linear Search:
  - E.g. scanning 4.5M images!
- k-D trees:
  - axis parallel partitions of the data
  - Only effective in low-dimensional data

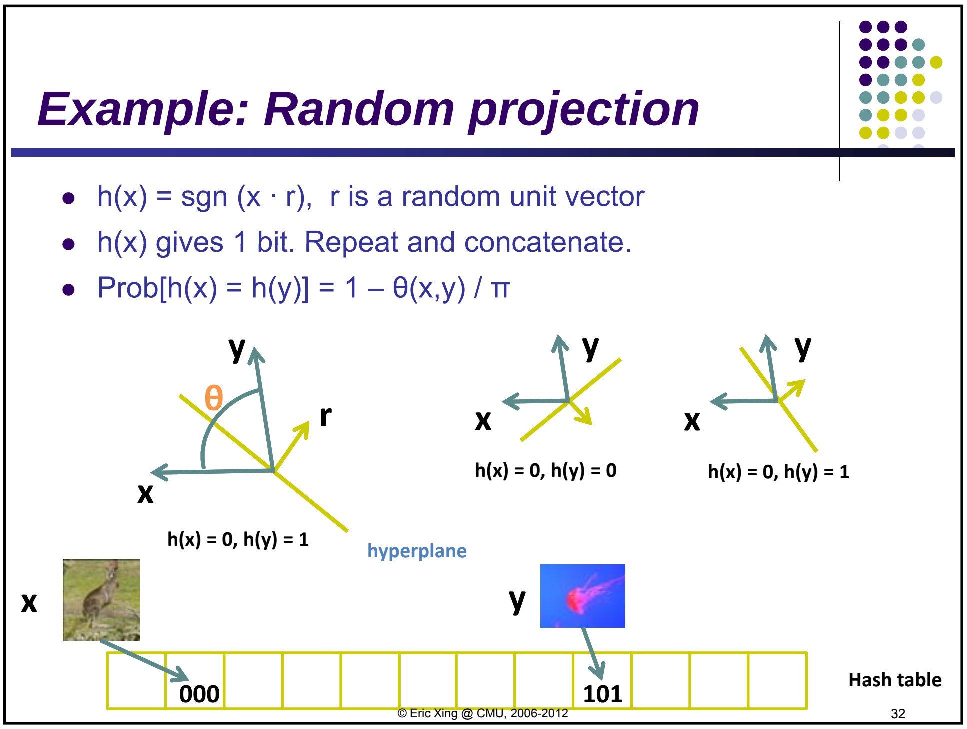


- Large Scale Approximate Indexing
  - Locality Sensitive Hashing (LSH)
  - Spill-Tree
  - NV-Tree
  - All above run on a single machine with all data in memory, and scale to millions of images
- Web-scale Approximate Indexing
  - Parallel variant of Spill-tree, NV-tree on distributed systems,
  - Scale to Billions of images in disks on multiple machines



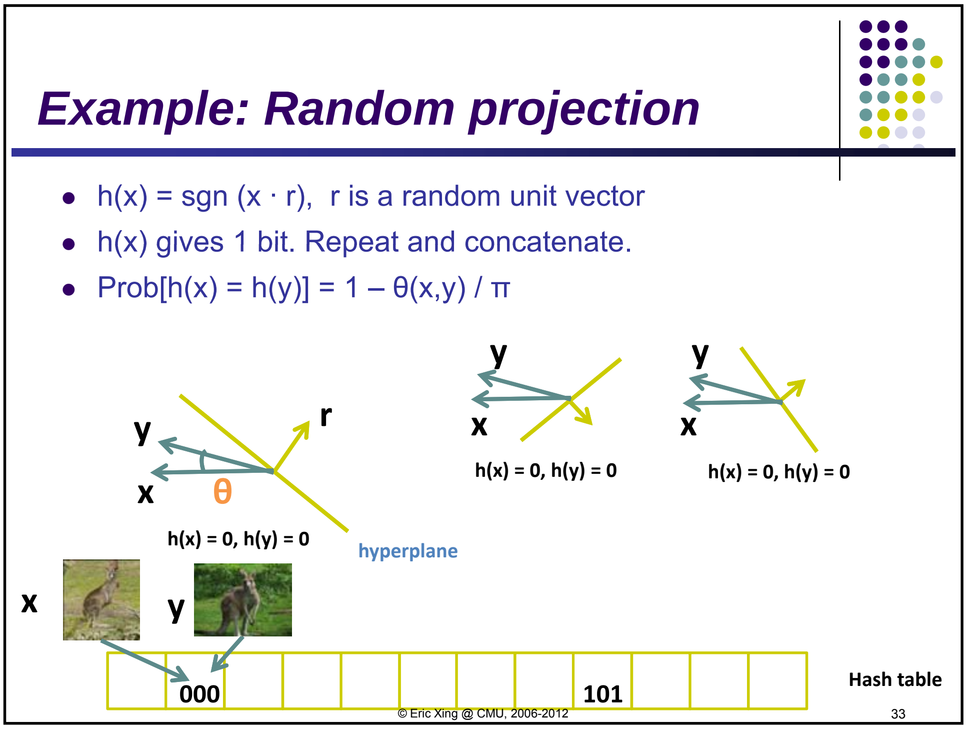
# Locality sensitive hashing

- *Approximate* kNN
  - Good enough in practice
  - Can get around curse of dimensionality
- *Locality sensitive* hashing
  - Near feature points → (likely) same hash values



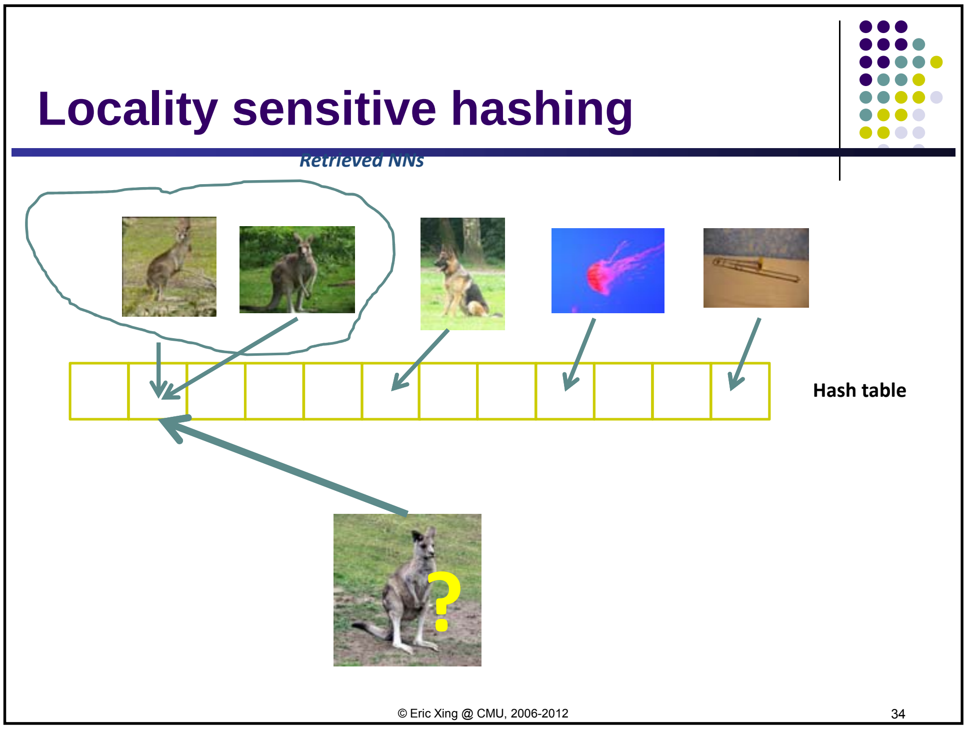
# *Example: Random projection*

- h(x) = sgn (x · r), r is a random unit vector
- h(x) gives 1 bit. Repeat and concatenate.
- Prob[h(x) = h(y)] = 1 – θ(x,y) / π



# *Example: Random projection*

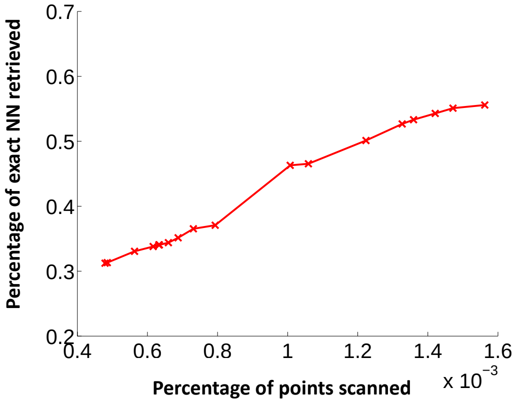
- h(x) = sgn (x · r),  r is a random unit vector
- h(x) gives 1 bit. Repeat and concatenate.
- Prob[h(x) = h(y)] = 1 – θ(x,y) / π



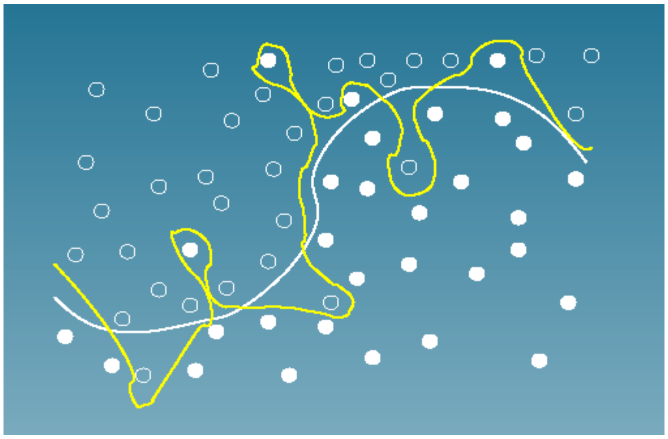
# Locality sensitive hashing

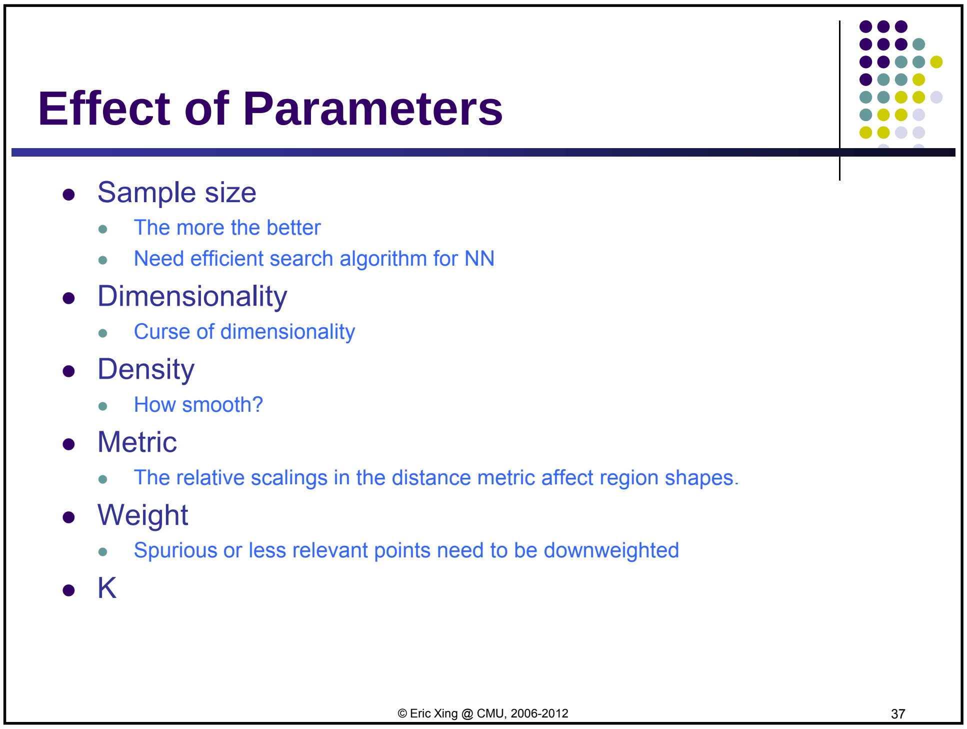
# Locality sensitive hashing

- 1000X speed-up with 50% recall of top 10-NN
- 1.2M images + 1000 dimensions



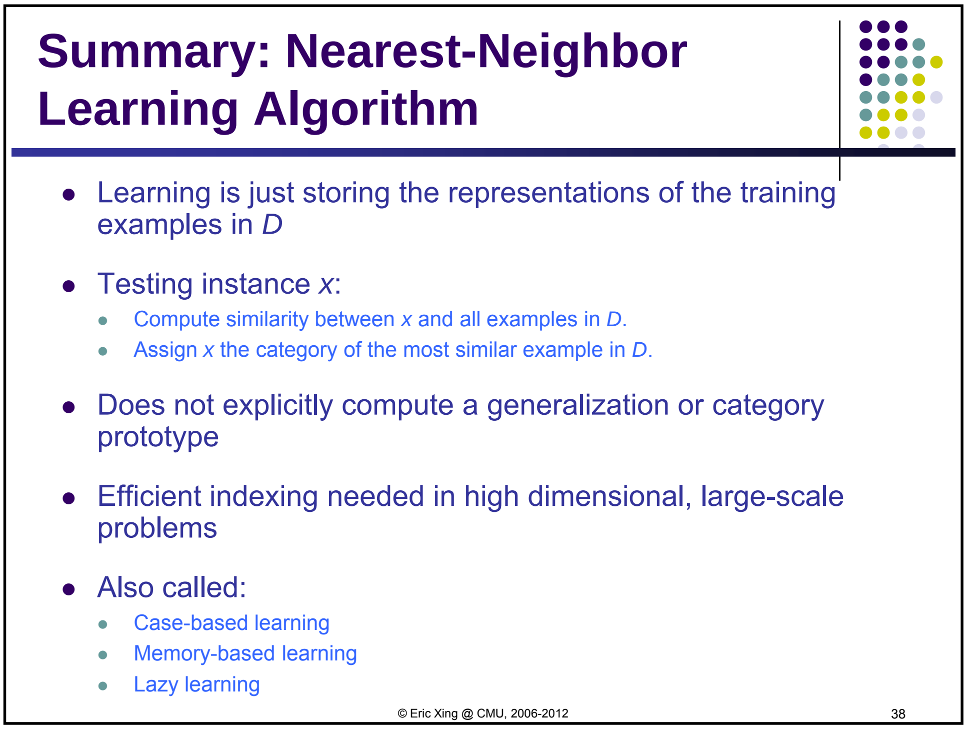
# Is kNN ideal? … more later





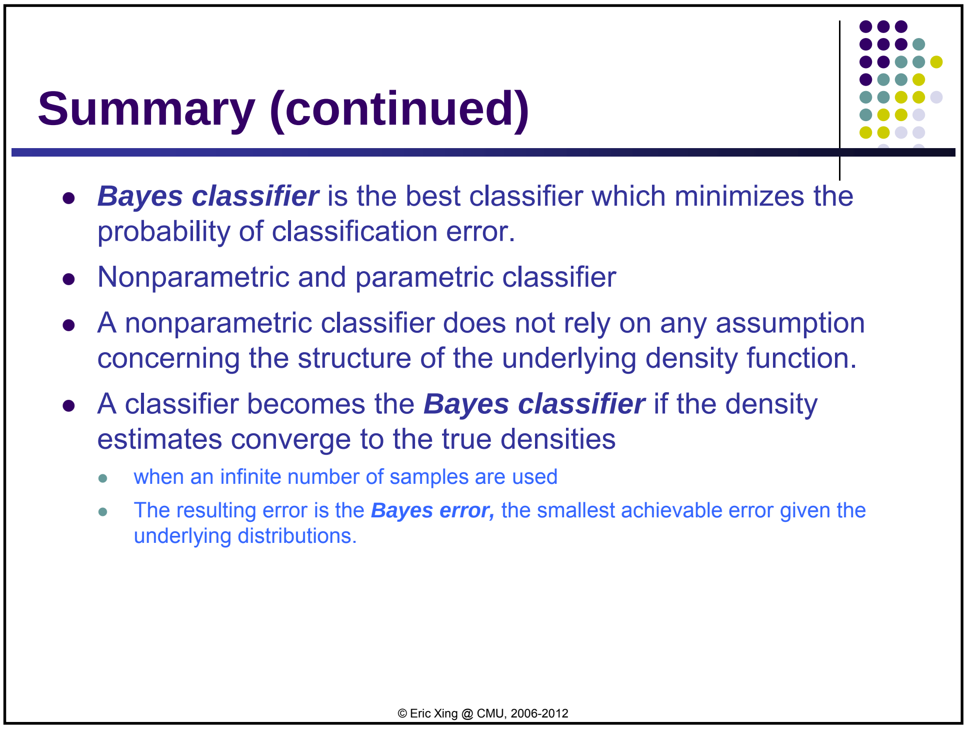
# Effect of Parameters

- Sample size
  - The more the better
  - Need efficient search algorithm for NN
- Dimensionality
  - Curse of dimensionality
- Density
  - How smooth?
- Metric
  - The relative scalings in the distance metric affect region shapes.
- Weight
  - Spurious or less relevant points need to be downweighted
- K



# Summary: Nearest-Neighbor Learning Algorithm

- Learning is just storing the representations of the training examples in *D*
- Testing instance *x*:
  - Compute similarity between *x* and all examples in *D*.
  - Assign *x* the category of the most similar example in *D*.
- Does not explicitly compute a generalization or category prototype
- Efficient indexing needed in high dimensional, large-scale problems
- Also called:
  - Case-based learning
  - Memory-based learning
  - Lazy learning



# Summary (continued)

- ***Bayes classifier*** is the best classifier which minimizes the probability of classification error.
- Nonparametric and parametric classifier
- A nonparametric classifier does not rely on any assumption concerning the structure of the underlying density function.
- A classifier becomes the ***Bayes classifier*** if the density estimates converge to the true densities
  - when an infinite number of samples are used
  - The resulting error is the ***Bayes error,*** the smallest achievable error given the underlying distributions.