Machine Learning

10-701/15-781, Fall 2011

"Nonparametric" methods

Eric Xing

Lecture 2, September 14, 2011

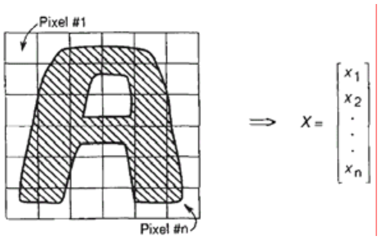
Reading:

Univariate prediction without using a model: good or bad?

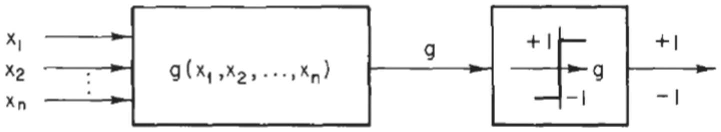
- Nonparametric Classifier (Instance-based learning)
  - Nonparametric density estimation
  - K-nearest-neighbor classifier
  - Optimality of kNN
- Spectrum clustering
  - Clustering
  - Graph partition and normalized cut
  - The spectral clustering algorithm
- Very little "learning" is involved in these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

Classification

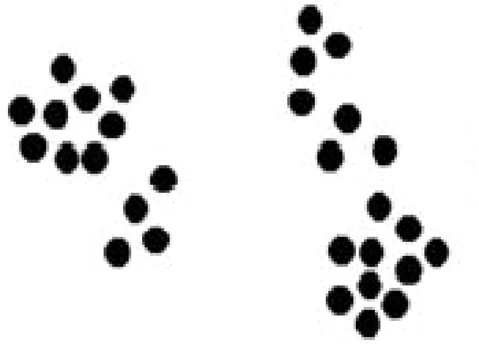
- Representing data:



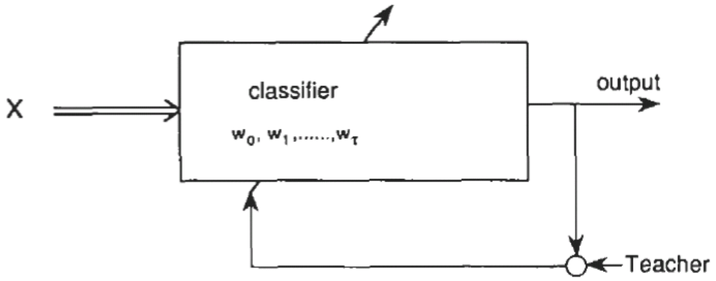
- Hypothesis (classifier)



Clustering

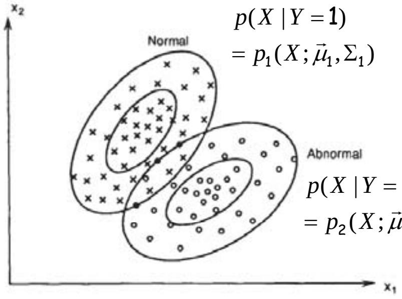


Supervised vs. Unsupervised Learning



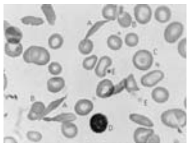
Decision-making as dividing a high-dimensional space

- 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 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:

- Likelihood Ratio:

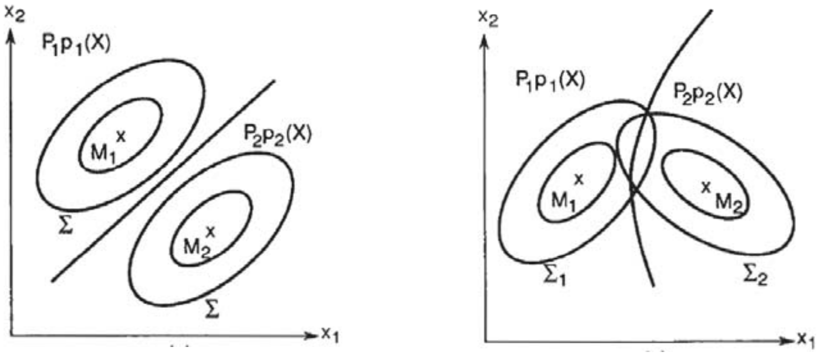
$\ell(X)=$

- 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!!

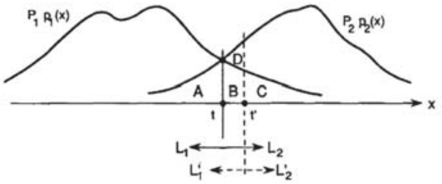
Bayes Error

- 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*?

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

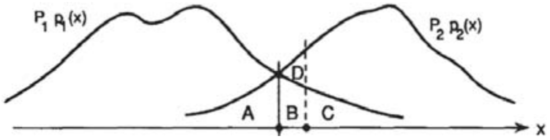
- The Bayes error (the expected risk):

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



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:

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

Learning Classifier

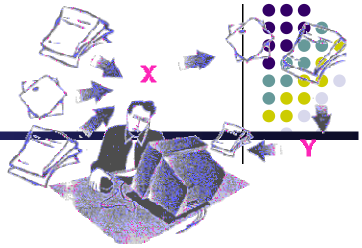
- The decision rule:

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

- 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**

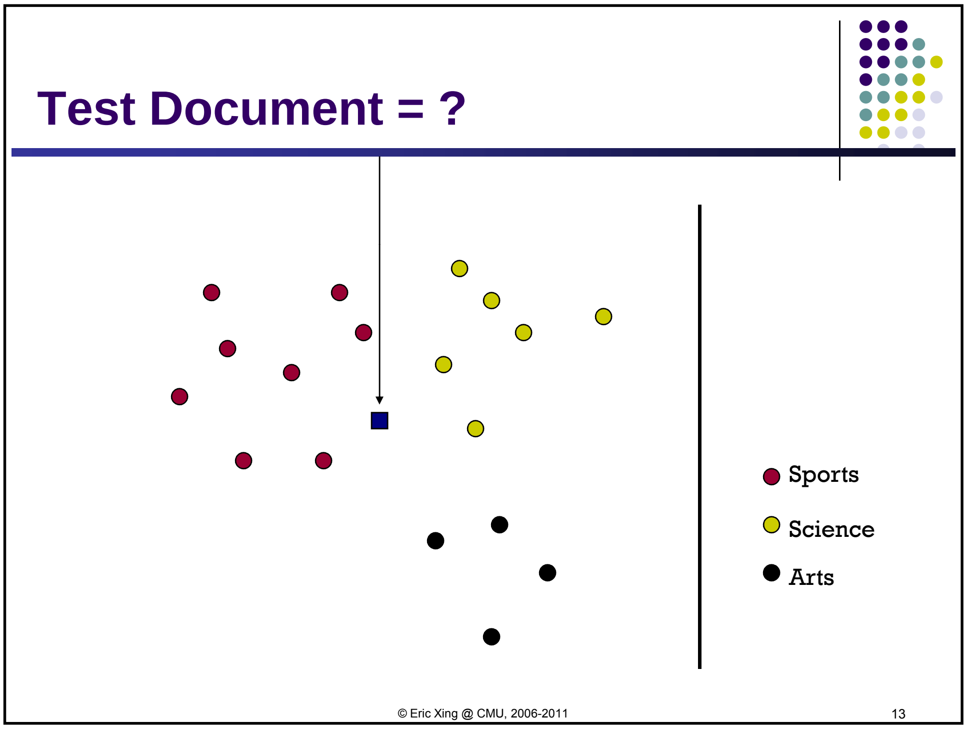
Recall: Vector Space Representation



- 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

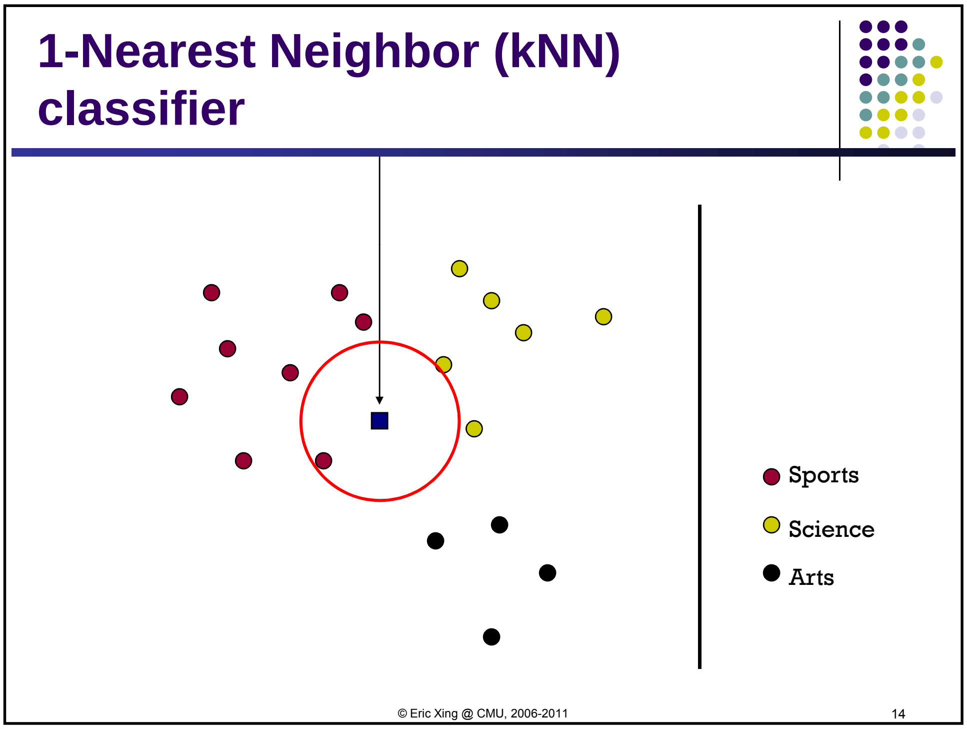


Test Document = ?

● Sports

○ Science

● Arts

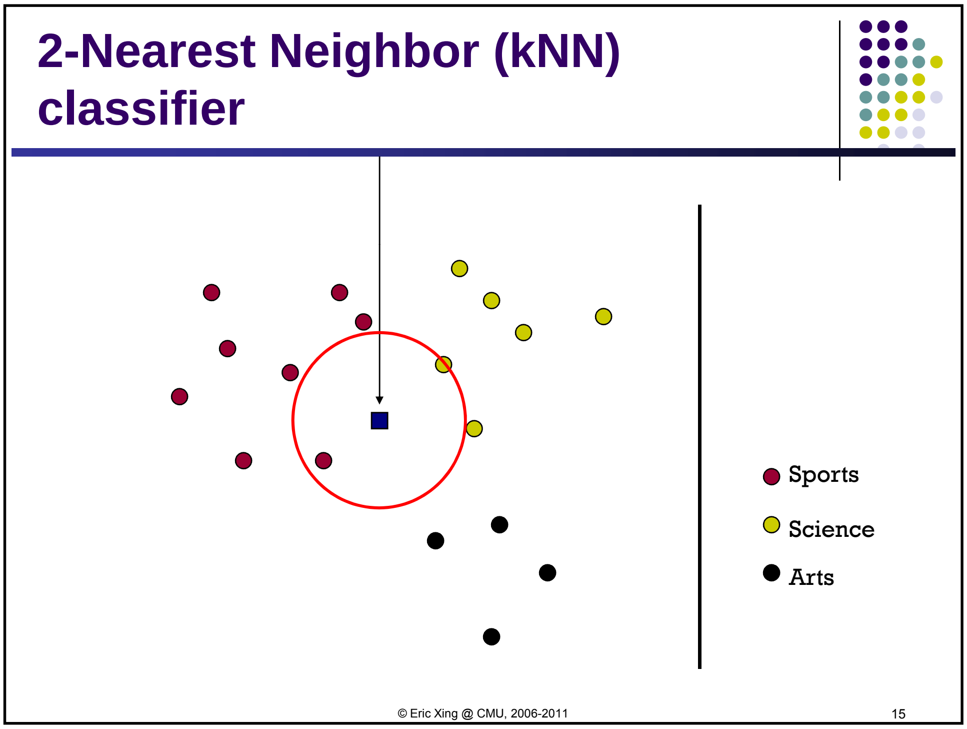


1-Nearest Neighbor (kNN) classifier

● Sports

○ Science

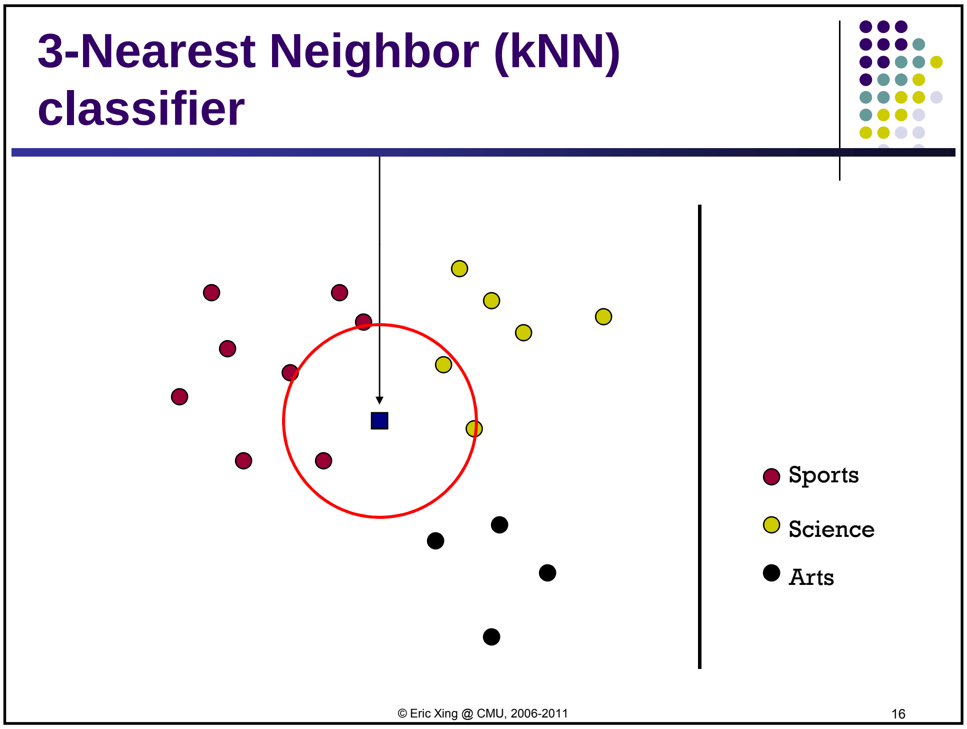
● Arts



2-Nearest Neighbor (kNN) classifier

© Eric Xing @ CMU, 2006-2011

15



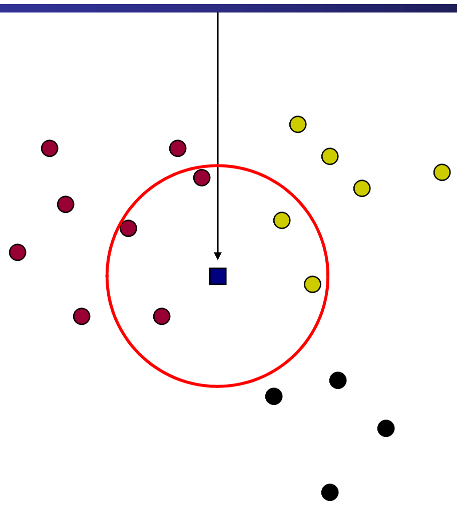
3-Nearest Neighbor (kNN) classifier

© Eric Xing @ CMU, 2006-2011

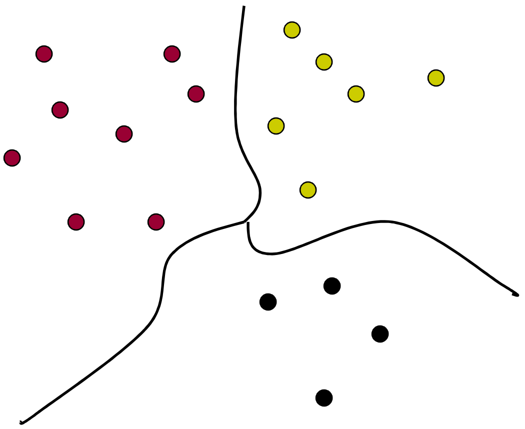
16

K-Nearest Neighbor (kNN) classifier

Voting kNN

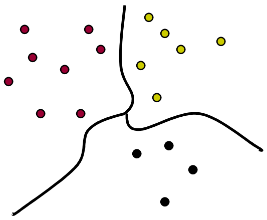


Classes in a Vector Space



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:

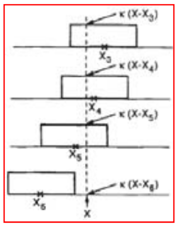


Where does kNN come from?

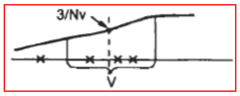
- How to estimation $p(X)$?
- Nonparametric density estimation
  - Parzen density estimate

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:

$$h(X) = -\ln\frac{p_1(X)}{p_2(X)} = -\ln\frac{(k_1-1)N_2V_2(X)}{(k_2-1)N_1V_1(X)} \mathop{\gtrless} \ln\frac{\pi_1}{\pi_2}$$

  - 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

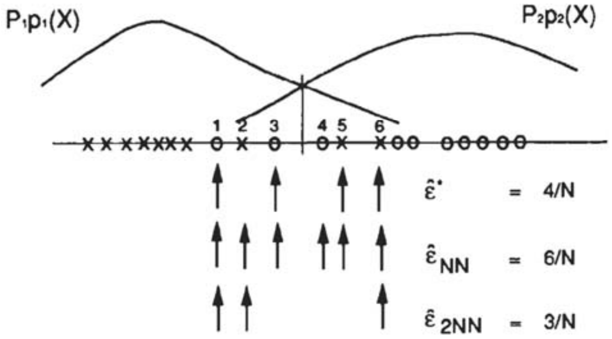
$$r_1^*(X) = 2\xi(X) \leq 2r^*(X)$$

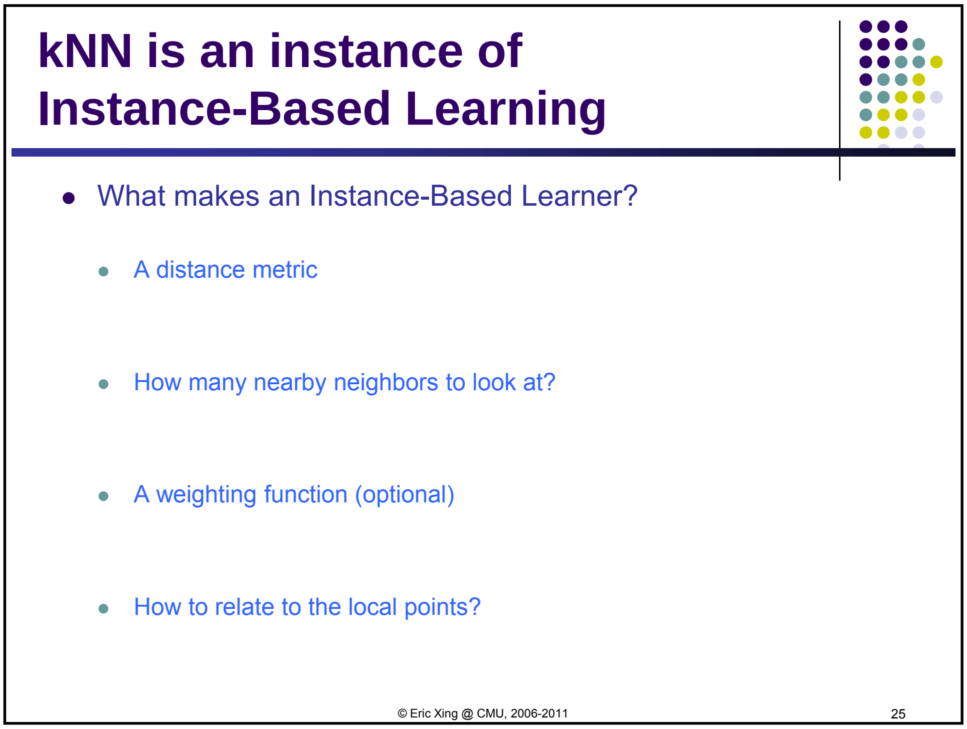
- It can be shown that $\epsilon_1^* \leq 2\epsilon^*$
  - This is remarkable, considering that the procedure does not use any information about the underlying distributions and only the class of the single nearest neighbor determines the outcome of the decision.

In fact

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

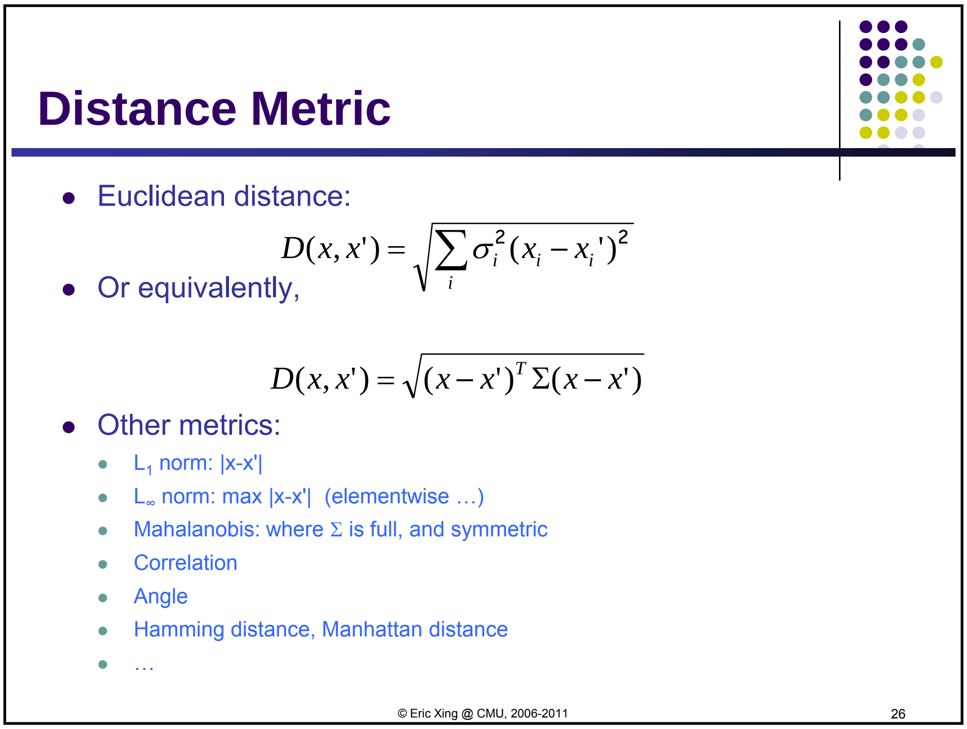
- Example:


$P_1p_1(X)$
$P_2p_2(X)$
1
2
3
4
5
6
x x x xxxx o x o o x xoo ooooo
$\hat{\varepsilon}^*$ = 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

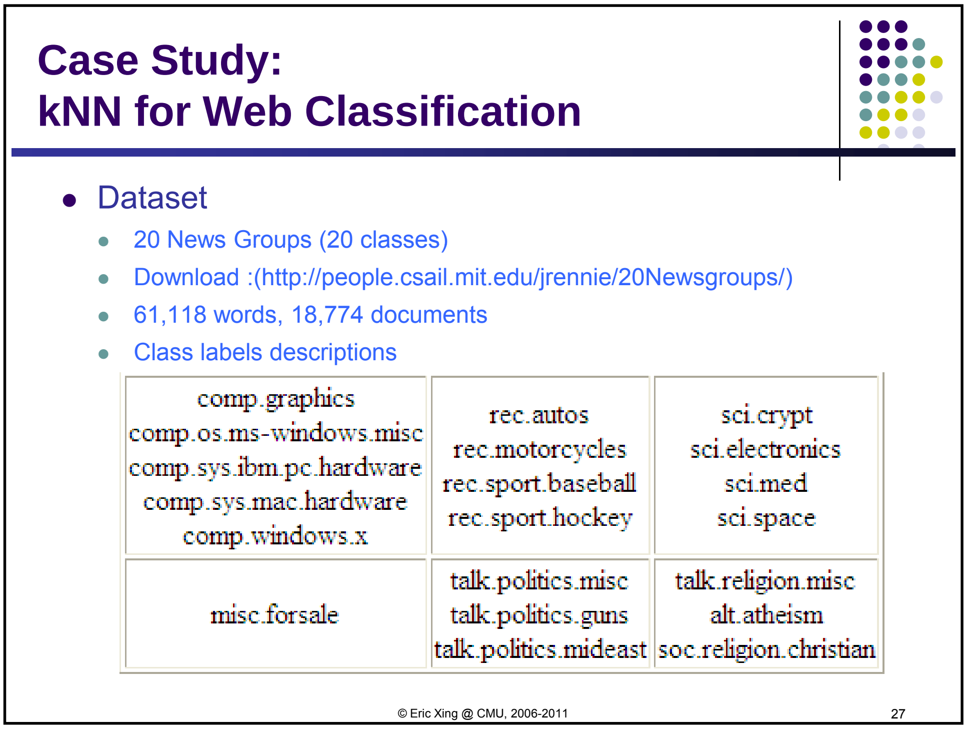
- Euclidean distance:

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

- Or equivalently,

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

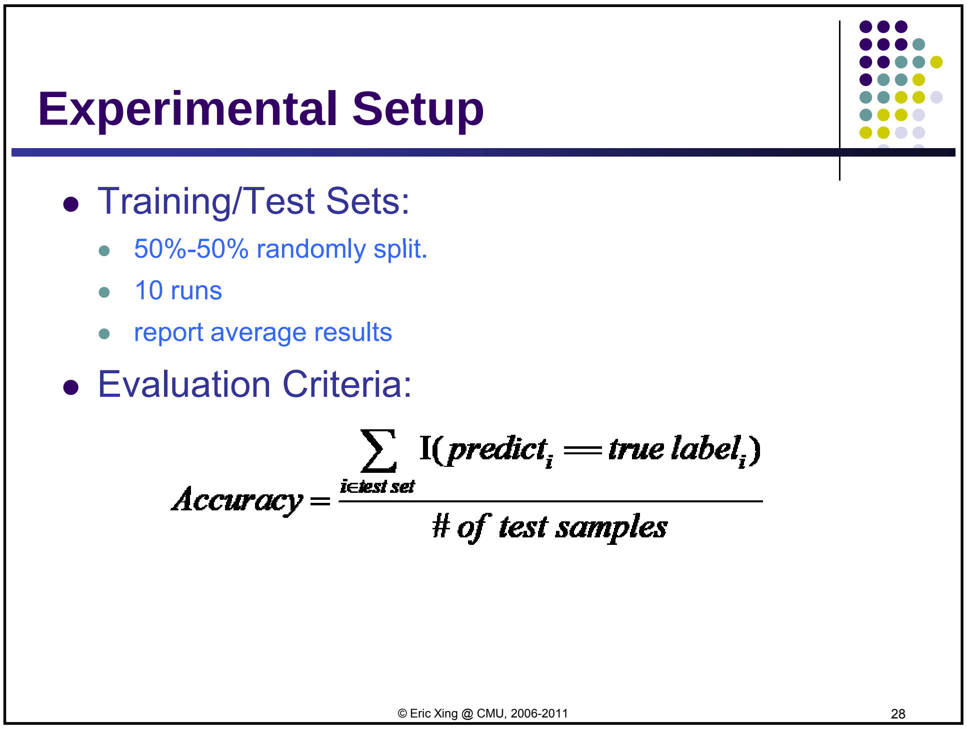
- Other metrics:
  - L_1 norm: |x-x'|
  - L_∞ norm: max |x-x'| (elementwise …)
  - Mahalanobis: where Σ is full, and symmetric
  - Correlation
  - Angle
  - Hamming distance, Manhattan distance
  - …



Case Study: kNN for Web Classification

- Dataset
  - 20 News Groups (20 classes)
  - Download :(http://people.csail.mit.edu/jrennie/20Newsgroups/)
  - 61,118 words, 18,774 documents
  - Class labels descriptions

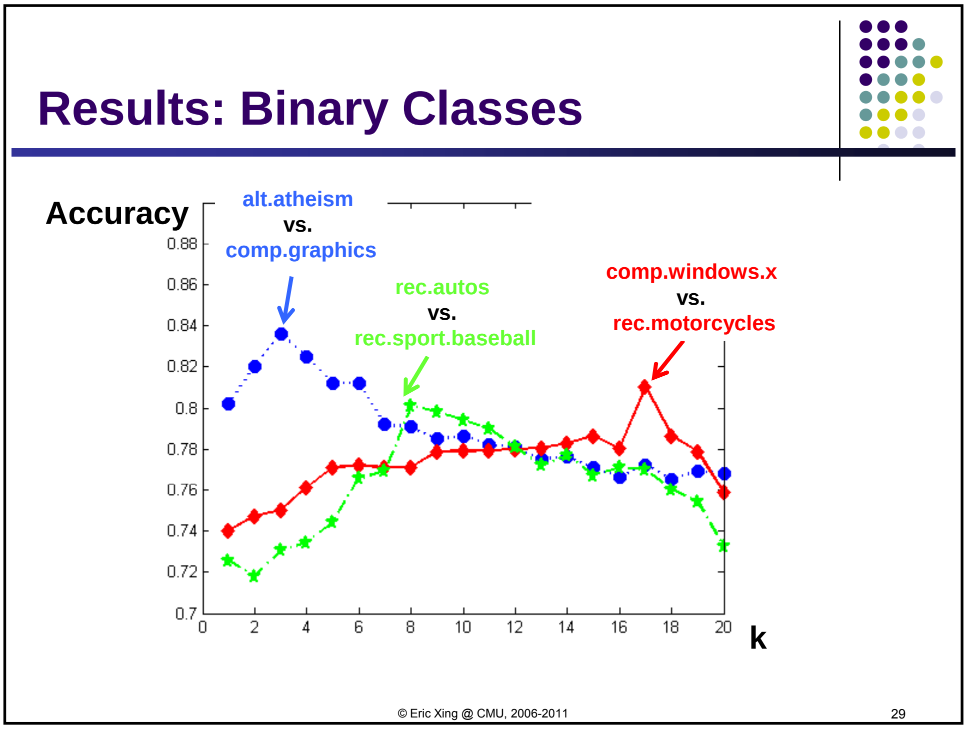
comp.graphics comp.os.ms-windows.misc comp.sys.ibm.pc.hardware comp.sys.mac.hardware comp.windows.x	rec.autos rec.motorcycles rec.sport.baseball rec.sport.hockey	sci.crypt sci.electronics sci.med sci.space
misc.forsale	talk.politics.misc talk.politics.guns talk.politics.mideast	talk.religion.misc alt.atheism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



Experimental Setup

- Training/Test Sets:
  - 50%-50% randomly split.
  - 10 runs
  - report average results
- Evaluation Criteria:

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



Results: Binary Classes

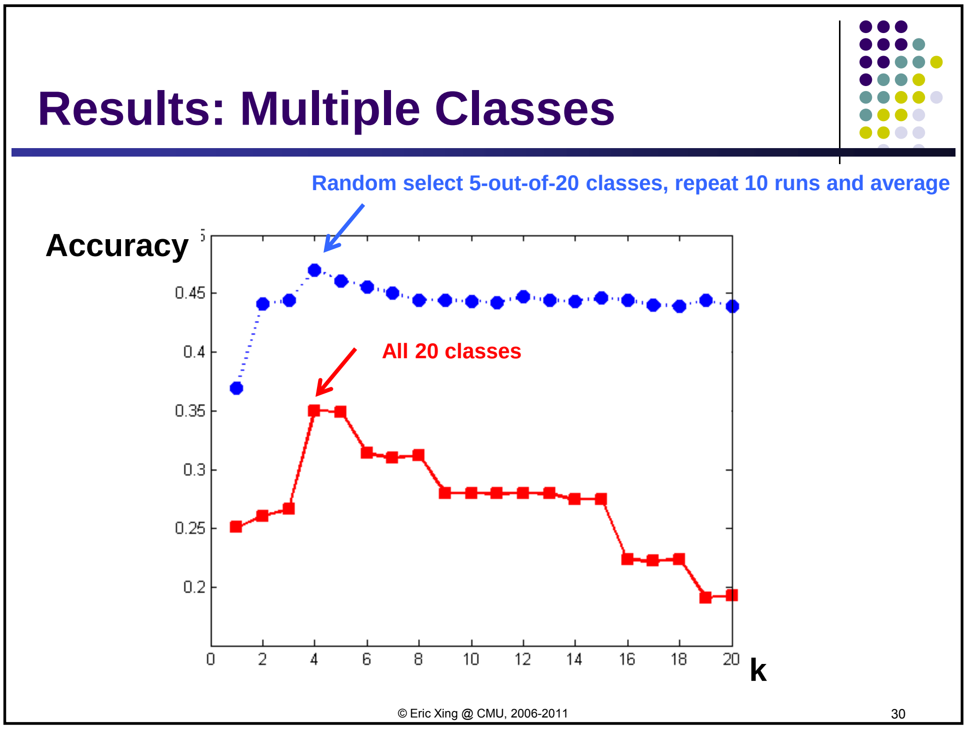
alt.atheism
vs.
comp.graphics

rec.autos
vs.
rec.sport.baseball

comp.windows.x
vs.
rec.motorcycles

© Eric Xing @ CMU, 2006-2011

29



Results: Multiple Classes

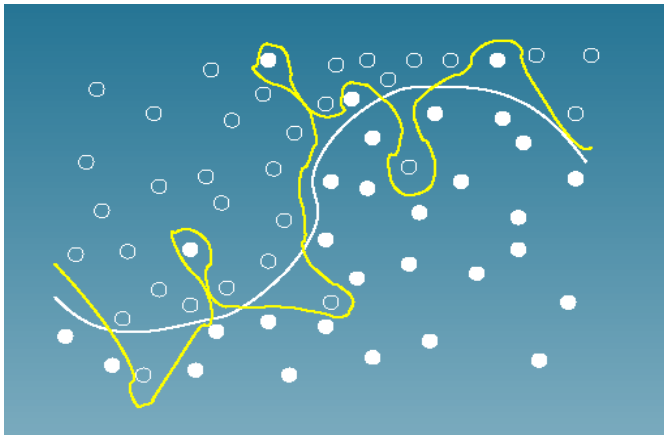
Random select 5-out-of-20 classes, repeat 10 runs and average

All 20 classes

© Eric Xing @ CMU, 2006-2011

30

Is kNN ideal? … more later

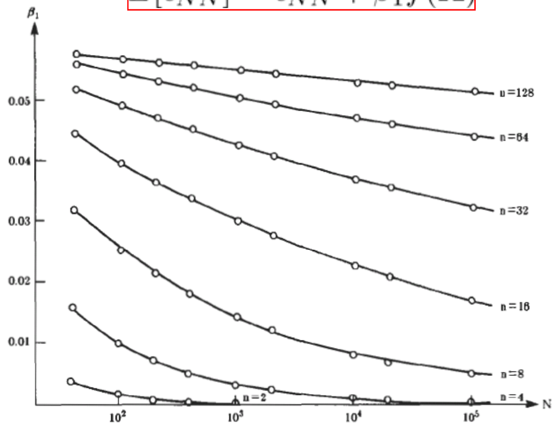


Effect of Parameters

- 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

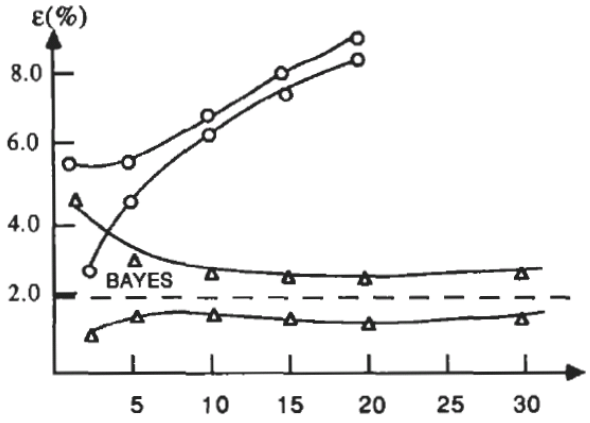
Sample size and dimensionality

$$\boxed{E[\hat{\epsilon}_{NN}] \cong \epsilon_{NN} + \beta_1 f(\mathbf{X})}$$

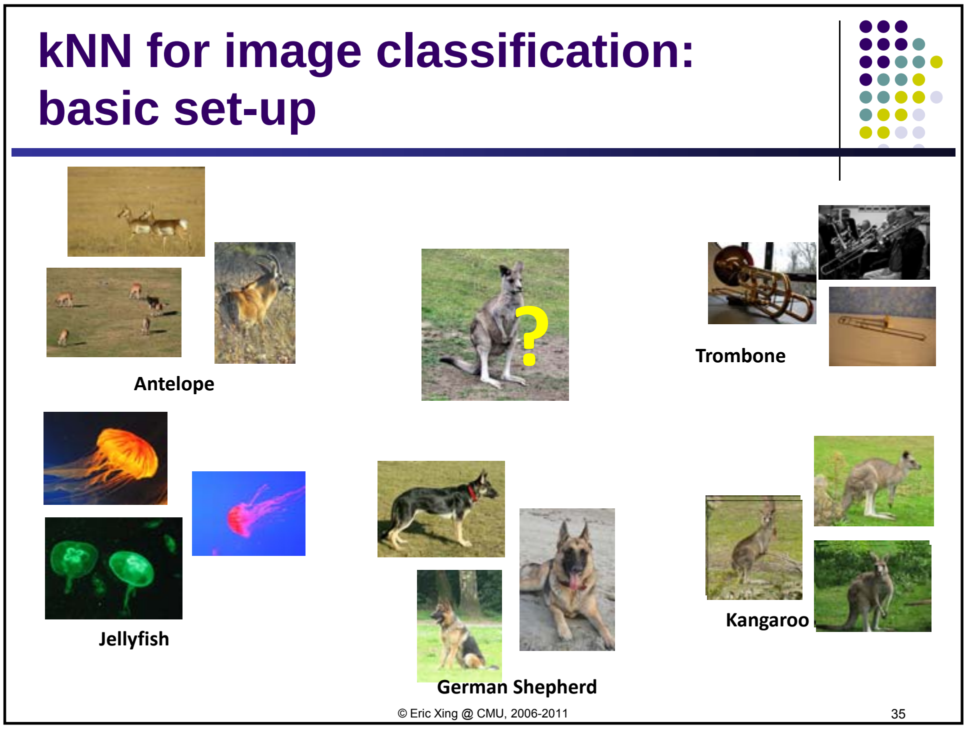


From page 316, Fukumaga

Neighborhood size



From page 350, Fukumaga



kNN for image classification: basic set-up

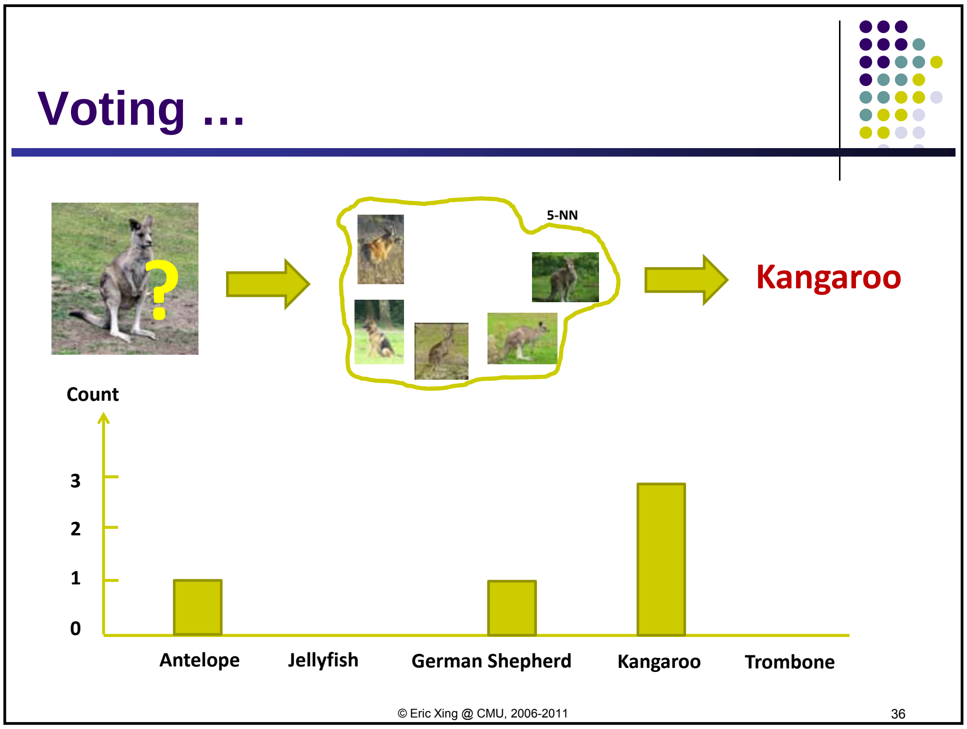
Antelope

Trombone

Jellyfish

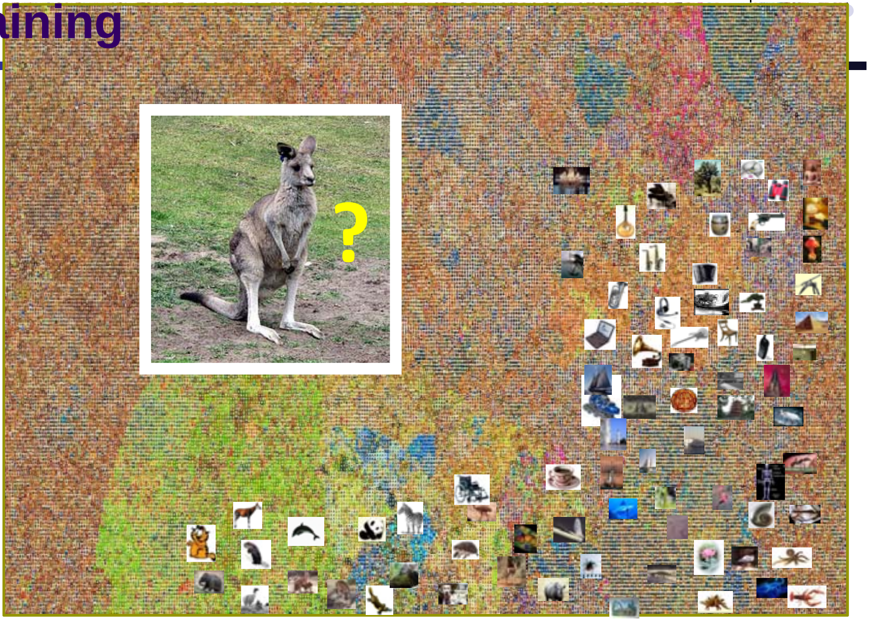
German Shepherd

Kangaroo



Voting …

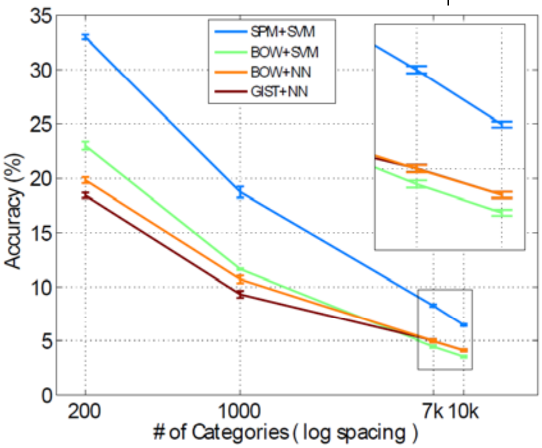
10K classes, 4.5M Queries, 4.5M training



Background image courtesy: Antonio Torralba

KNN on 10K classes

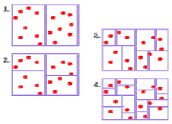
- 10K classes
- 4.5M queries
- 4.5M training
- Features
  - BOW
  - GIST



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

Nearest Neighbor Search in High Dimensional Metric Space

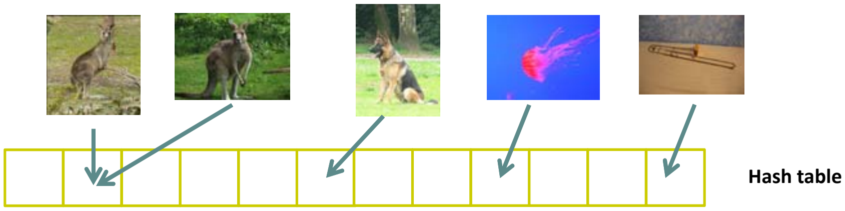
- Linear Search:
  - E.g. scanning 4.5M images!
- k-D trees:
  - axis parallel partitions of the data
  - Only effective in low-dimensional data

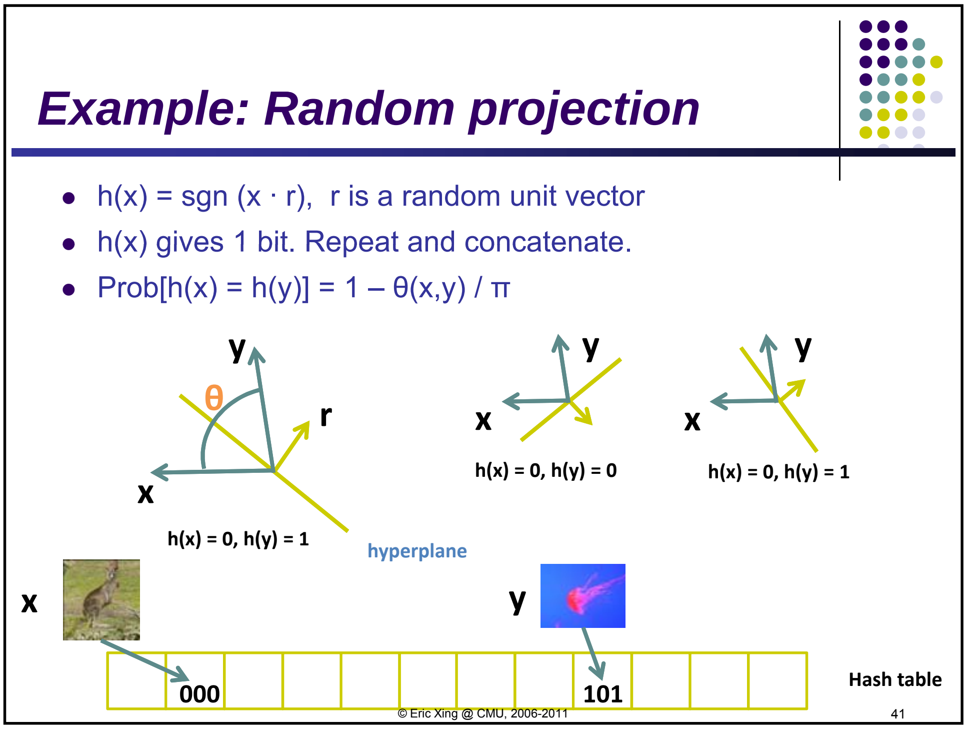


- Large Scale Approximate Indexing
  - Locality Sensitive Hashing (LSH)
  - Spill-Tree
  - NV-Tree
  - All above run on a single machine with all data in memory, and scale to millions of images
- Web-scale Approximate Indexing
  - Parallel variant of Spill-tree, NV-tree on distributed systems,
  - Scale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

Locality sensitive hashing

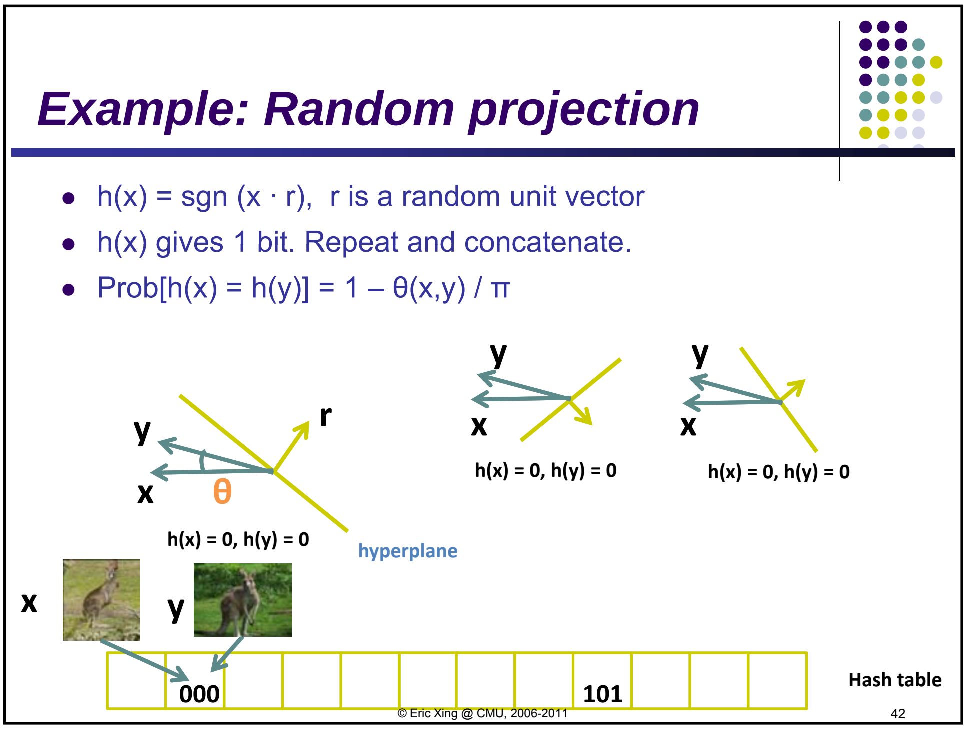
- *Approximate* kNN
  - Good enough in practice
  - Can get around curse of dimensionality
- *Locality sensitive* hashing
  - Near feature points → (likely) same hash values





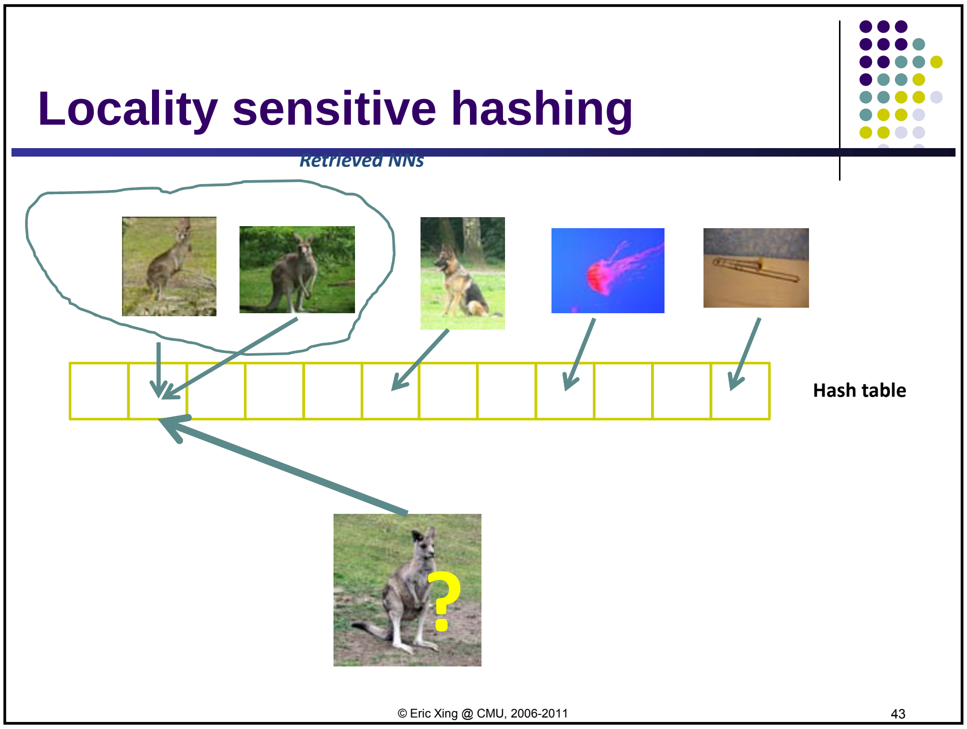
Example: Random projection

- 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

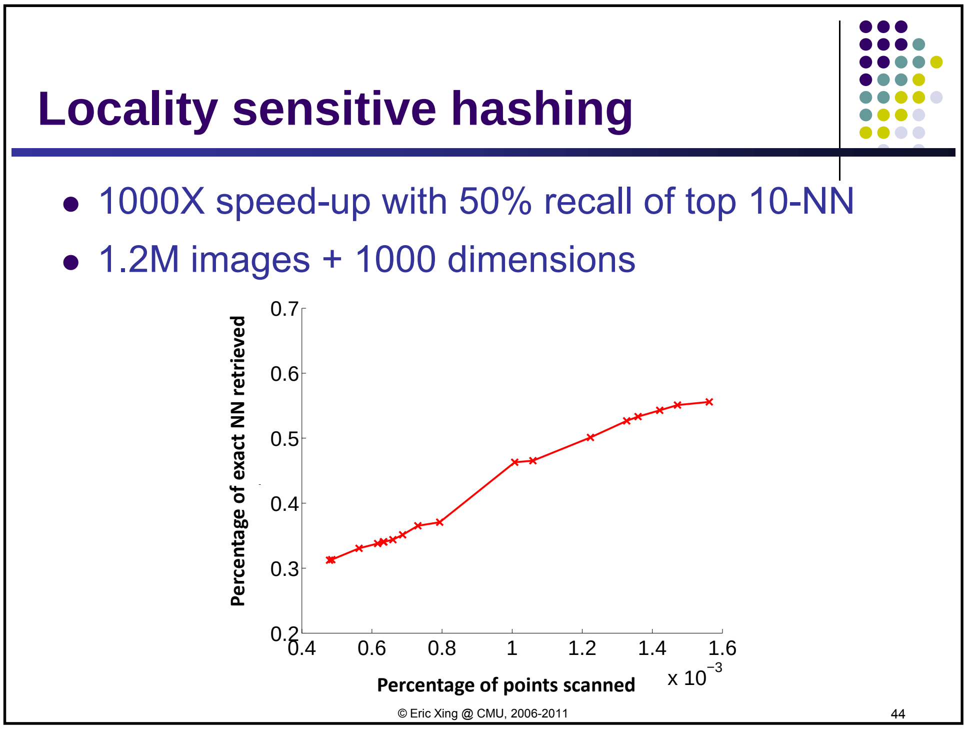


Example: Random projection

- 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

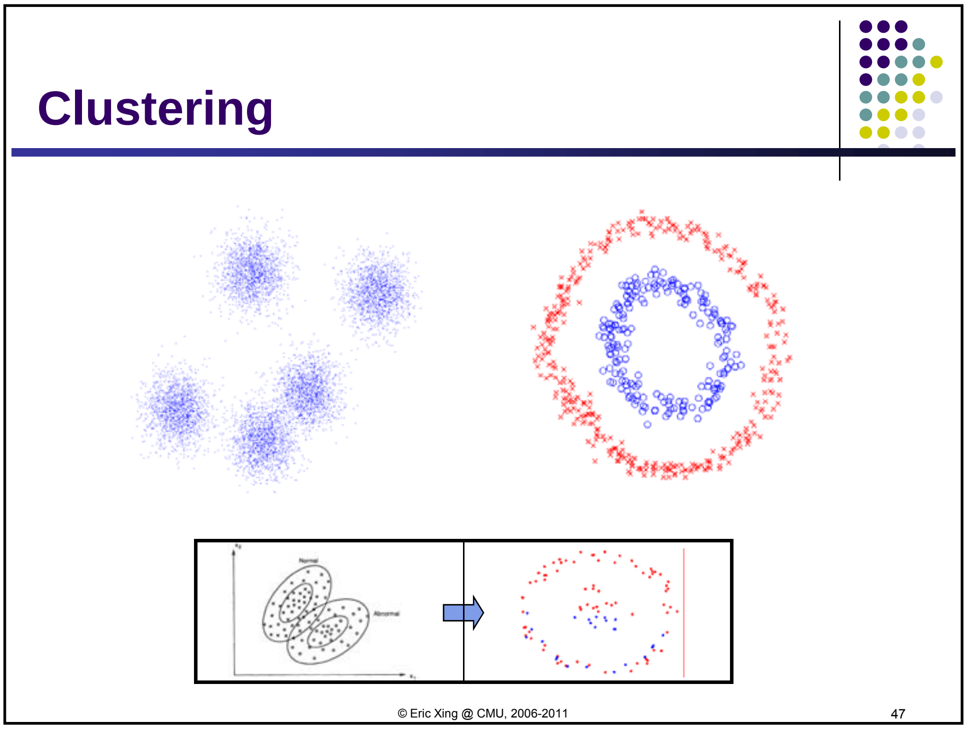
- 1000X speed-up with 50% recall of top 10-NN
- 1.2M images + 1000 dimensions

Summary: Nearest-Neighbor Learning Algorithm

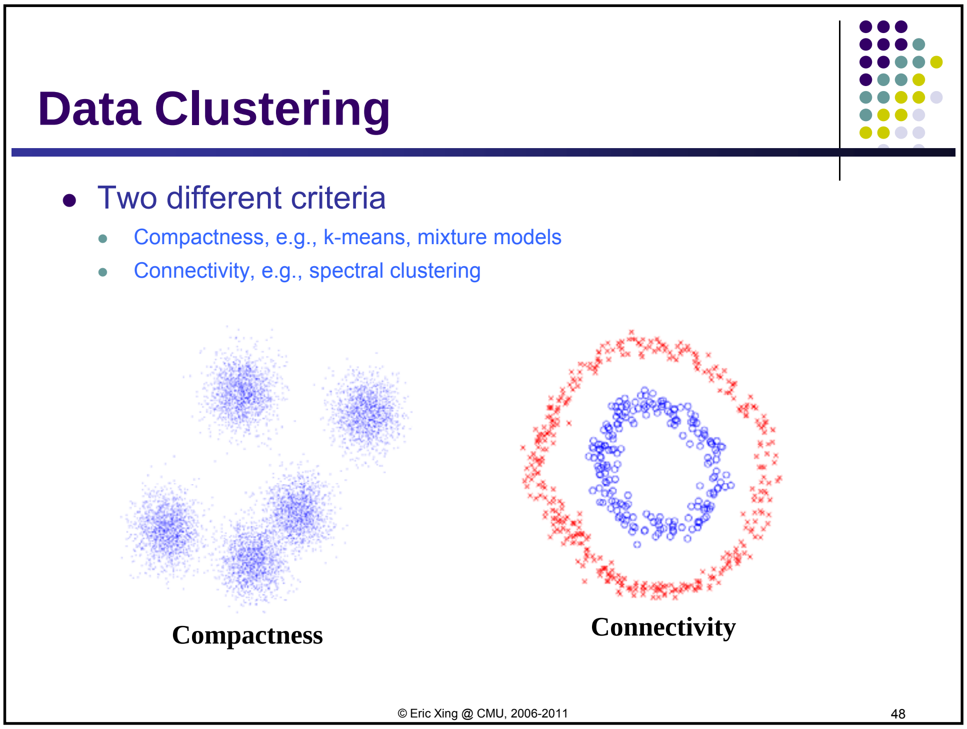
- Learning is just storing the representations of the training examples in *D*
- Testing instance *x*:
  - Compute similarity between *x* and all examples in *D*.
  - Assign *x* the category of the most similar example in *D*.
- Does not explicitly compute a generalization or category prototype
- Efficient 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.



Clustering



Data Clustering

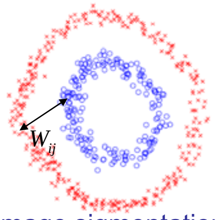
- Two different criteria
  - Compactness, e.g., k-means, mixture models
  - Connectivity, e.g., spectral clustering

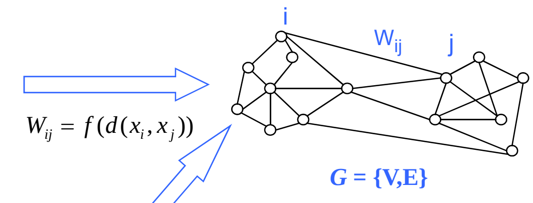
Compactness

Connectivity

Graph-based Clustering

- Data Grouping





$W_{ij} = f(d(x_i, x_j))$

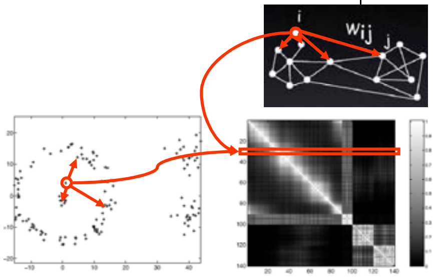
G = {V,E}

- Image sigmentation

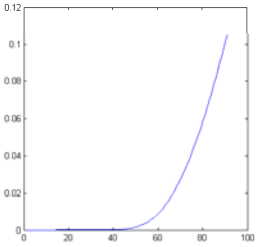
- Affinity matrix: $W = [w_{i,j}]$
- Degree matrix: $D = \mathrm{diag}(d_i)$

Affinity Function

$$W_{i,j} = e^{\frac{-\|X_i - X_j\|_2^2}{\sigma^2}}$$



- Affinities grow as σ grows →
- How the choice of σ value affects the results?
- What would be the optimal choice for σ?

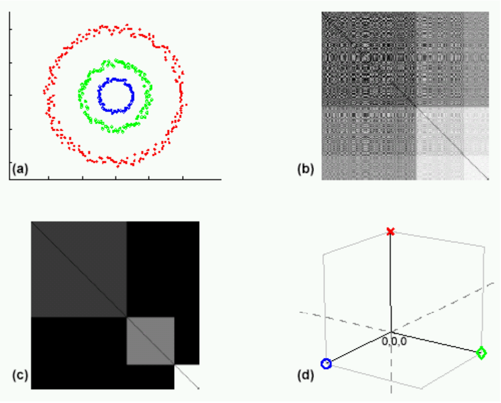


A Spectral Clustering Algorithm

Ng, Jordan, and Weiss 2003

- Given a set of points *S={s$_1$,…s$_n$}*
- Form the affinity matrix $$w_{i,j} = e^{\frac{-\|s_i - s_j\|_2^2}{\sigma^2}} , \quad \forall i \neq j, \quad w_{i,i} = 0$$
- Define diagonal matrix $D_{ii} = \Sigma_{\kappa}\, a_{ik}$
- Form the matrix $$L = D^{-1/2} W D^{-1/2}$$
- Stack the *k* largest eigenvectors of L to for the columns of the new matrix X: $$X = \begin{bmatrix} | & | & & | \\ x_1 & x_2 & \cdots & x_k \\ | & | & & | \end{bmatrix}$$
- Renormalize each of X's rows to have unit length and get new matrix Y. Cluster rows of Y as points in R^k

Why it works?



- K-means in the spectrum space !

More formally …

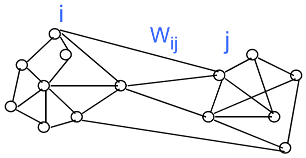
- Spectral clustering is equivalent to minimizing a generalized normalized cut

$$\min \ \mathrm{Ncut}(A_1, A_2 \dots A_k) = \sum_{r=1}^{k} \left(\frac{\mathrm{cut}(A_r, \overline{A}_r)}{d_{A_r}} \right)$$

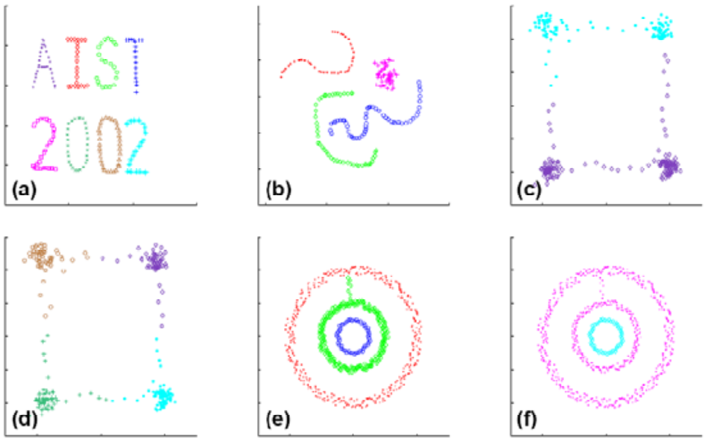
$$\min \ Y^{\mathrm{T}} D^{-1/2} W D^{-1/2} Y$$

$$\text{s.t.} \ \ Y^{\mathrm{T}} Y = I$$

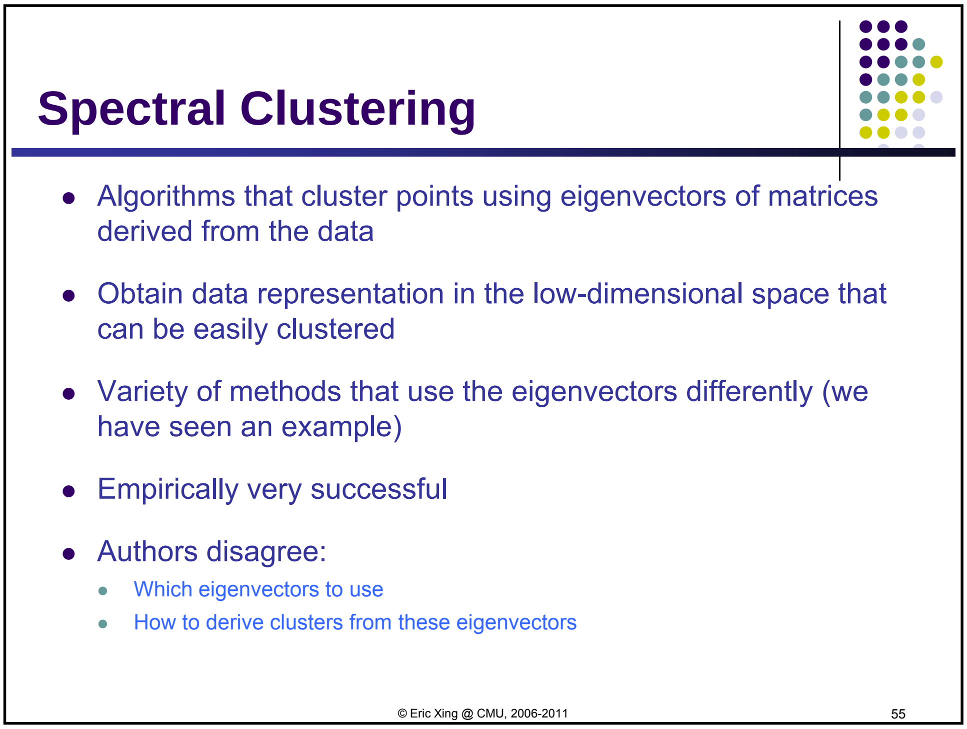
$$Y = \begin{array}{cc} \text{segments} & \\ \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} & \text{pixels} \end{array}$$



Toy examples

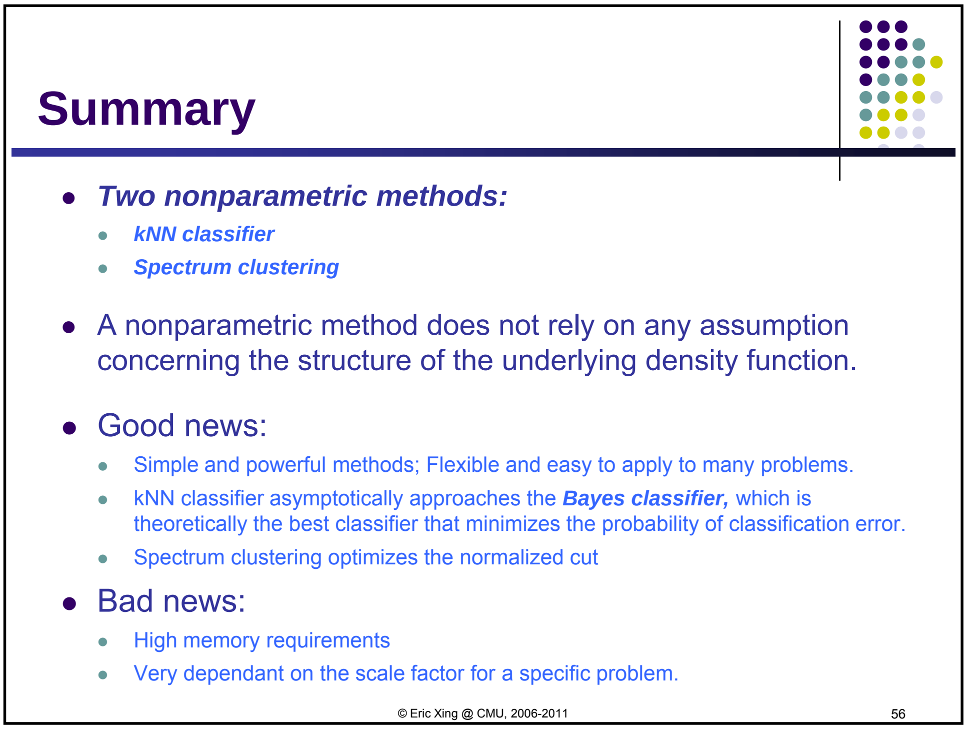


Images from Matthew Brand (TR-2002-42)



Spectral Clustering

- Algorithms that cluster points using eigenvectors of matrices derived from the data
- Obtain data representation in the low-dimensional space that can be easily clustered
- Variety of methods that use the eigenvectors differently (we have seen an example)
- Empirically very successful
- Authors disagree:
  - Which eigenvectors to use
  - How to derive clusters from these eigenvectors



Summary

- ***Two nonparametric methods:***
  - ***kNN classifier***
  - ***Spectrum clustering***
- A nonparametric method does not rely on any assumption concerning the structure of the underlying density function.
- Good news:
  - Simple and powerful methods; Flexible and easy to apply to many problems.
  - kNN classifier asymptotically approaches the ***Bayes classifier,*** which is theoretically the best classifier that minimizes the probability of classification error.
  - Spectrum clustering optimizes the normalized cut
- Bad news:
  - High memory requirements
  - Very dependant on the scale factor for a specific problem.