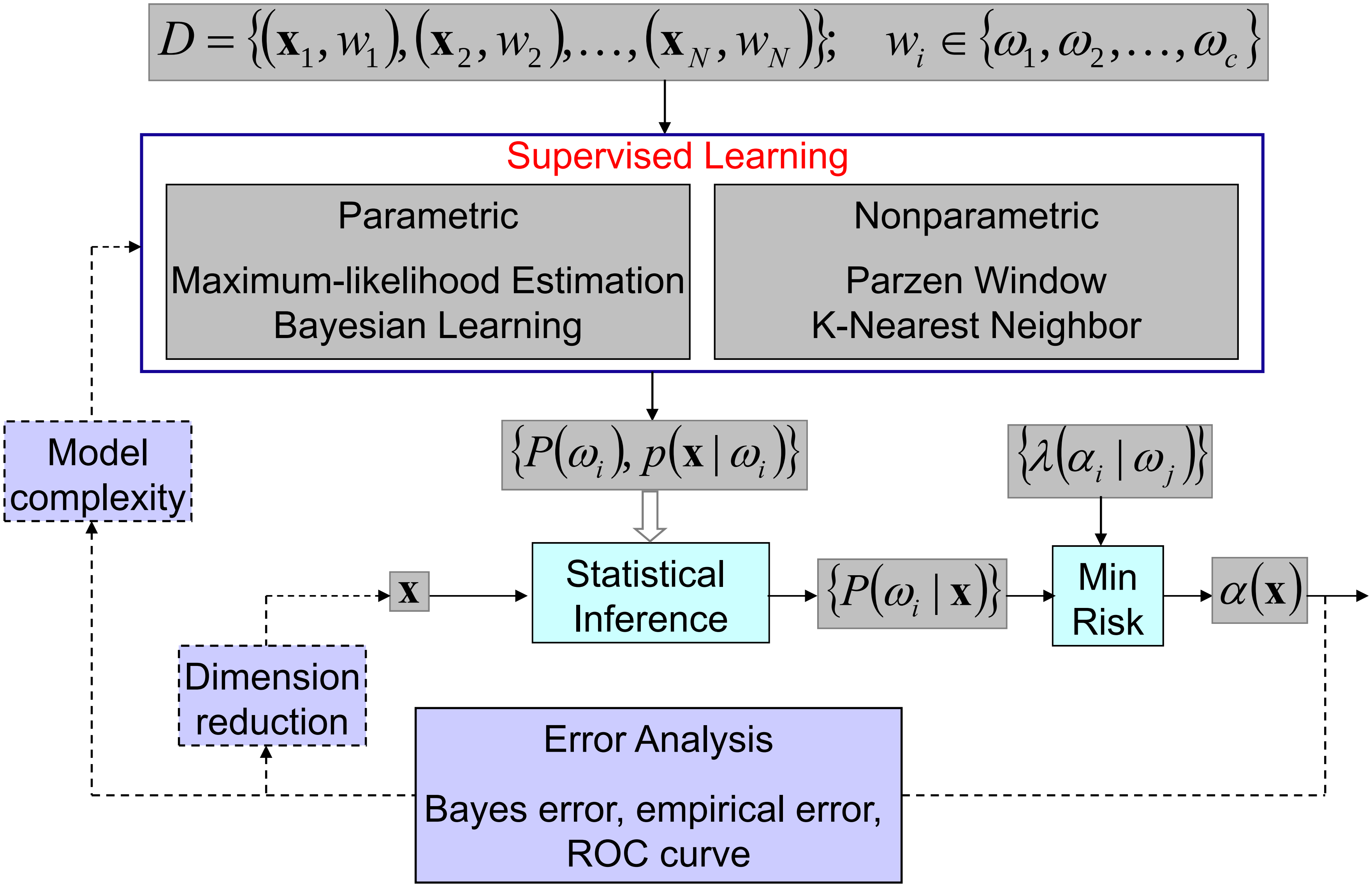
EE5907R: Pattern Recognition

Lecture 6: Review of Part I & Sample Exam Questions

National University of Singapore

System Diagram



Review of Part I

- Bayesian Decision Theory (Chapter 2)
  - –Bayes formula: posterior, likelihood, prior, evidence
  - –Decision rule, decision region, decision boundary
  - –Minimum risk, minimum-error-rate classification
- Parametric Density Estimation (Chapter 3)
  - –Maximum-likelihood Estimation
  - –Bayesian Parameter Estimation
- Nonparametric Techniques (Chapter 4)
  - –Density estimation
    - ✓ Parzen Window
    - ✓ K-Nearest-Neighbor
  - –KNN for classification

Review of Part I (Cont'd)

- Kernel Density Estimation
  - –Univariate KDE
    - ✓ Choosing the bandwidth
    - ✓ Bias-variance tradeoff
  - –Multivariate KDE
- Naïve Bayes Classifier
  - –Features are class-conditionally independent
- Classification Error
  - –Bayes error rate
  - –Error rate of the NN classifier
  - –ROC curve
  - –Confusion matrix

Bayesian Decision Theory

- Bayesian decision theory assumes that the conditional densities $p(x \mid \omega_j)$ and *a priori* probabilities $P(\omega_j)$ are known.

  Posterior = (Likelihood * Prior) / Evidence

  Posterior $\propto$ (Likelihood * Prior)

- It allows us to design optimal classifier
  - Definition of "optimal" depends on the chosen loss function
  - Under the minimum error rate (zero-one loss function)
    - ✓ No prior: ML classifier is optimal
    - ✓ Have prior: MAP classifier is optimal

Discriminant Functions for the Normal Density

- The minimum error-rate classification can be achieved by the discriminant functions

$$g_i(x) = \ln p(x \mid \omega_i) + \ln P(\omega_i)$$

- Case of multivariate normal

$$g_i(\mathbf{x}) = -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^t \boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i) - \frac{d}{2}\ln 2\pi - \frac{1}{2}\ln|\boldsymbol{\Sigma}_i| + \ln P(\omega_i)$$

$$g_i(\mathbf{x}) = -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^t \boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i) - \frac{1}{2}\ln|\boldsymbol{\Sigma}_i| + \ln P(\omega_i)$$

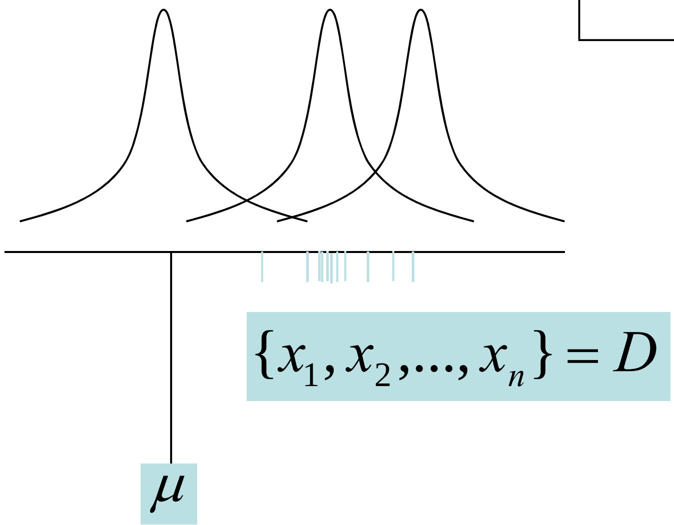
ML Vs. Bayesian

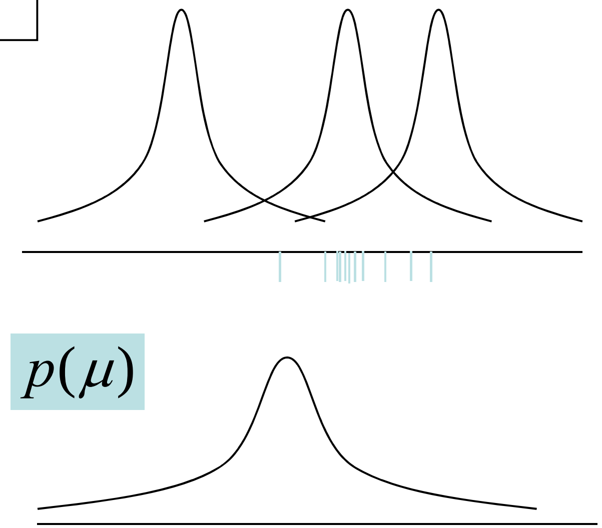
goal: Find $p(x\,|\,\omega_i)$ from D

goal: Find $p(x\,|\,\omega_i, D)$

$p(x\,|\,\omega_i) \sim N(\mu, \sigma^2)$

μ is unknown





sub-goal: Find μ
which maximizes $p(D\,|\,\mu)$

sub-goal: Find $p(\mu\,|\,D)$
and then use

$$p(x\,|\,D) = \int p(x\,|\,\mu, D)\, p(\mu\,|\,D)\, d\mu$$

Parzen Window vs. K-NN

Fix the window size $V = \frac{V_1}{\sqrt{n}}$

$k = k(x)$

$p_n(x) = \frac{k(x)}{V_1\sqrt{n}}$

The number of samples falling in a window can be counted explicitly

$$k(x) = \sum_{i=1}^{n} \varphi\left(\frac{x - x_i}{h}\right)$$

The estimate is always a legitimate density function .

$$p(x) = \frac{k/n}{V}$$

Fix the number of samples inside window $k = k_1\sqrt{n}$

$V = V(x)$

$p_n(x) = \frac{k_1}{V(x)\sqrt{n}}$

The window size does not have an explicit form – it is implicitly determined by the samples.

$$V(x) = 2\left|x - x_k^*\right|,\ x_k^* \in \{x_1, \ldots, x_n\}$$

The integral of the estimate can diverge to infinity.

Kernel Density Estimation

- Univariate KDE

$$p_{\text{KDE}}(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h^d}K\left(\frac{\boldsymbol{x}-\boldsymbol{X}_i}{h}\right)$$

- Multivariate KDE

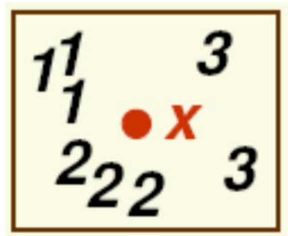
$$p_{\text{KDE}}(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h_1\cdots h_d}K\left(\frac{x_1-X_{i1}}{h_1},\cdots,\frac{x_d-X_{id}}{h_d}\right)$$

- Bias-variance Dilemma
  - Tradeoff between bias and variance
  - Definition of MSE and MISE
- Naïve Bayes

$$g_{i,\text{NB}}(\mathbf{x}) = P(\omega_i)\prod_{k=1}^{d}p(x_k \mid \omega_i)$$

K-NN for Classification

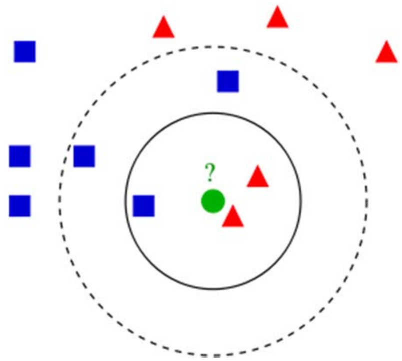
- Place a cell of volume V around x and capture k samples.
- Among these k samples, k_i samples turn out to be labeled ω_i , the posterior estimate is



$$P(\omega_i \mid \mathbf{x}) = k_i / k$$

- For minimum error rate, the most frequently represented category within the cell is selected.

$$\omega^*(\mathbf{x}) = \arg\max_i \{k_1, k_2, \ldots, k_c\}$$



- Selection of k and distance metric

Classification Error

- Probability of Error
  - Bayes error rate

$$P(error) = 1 - P(correct) = 1 - \sum_{i=1}^{c} \int_{R_i} p(\mathbf{x} \mid \omega_i) P(\omega_i) d\mathbf{x}$$

  - Error rate of the NN classifier

$$P^* \le P \le P^* \left(2 - \frac{c}{c-1} P^* \right) \le 2P^*$$

- Error Estimation Methods
  - Resubstitution, Holdout, Leave-one-out, N-fold cross validation, Bootstrap
- Receiver Operating Characteristic (ROC)
- Confusion Matrix

Image Classification

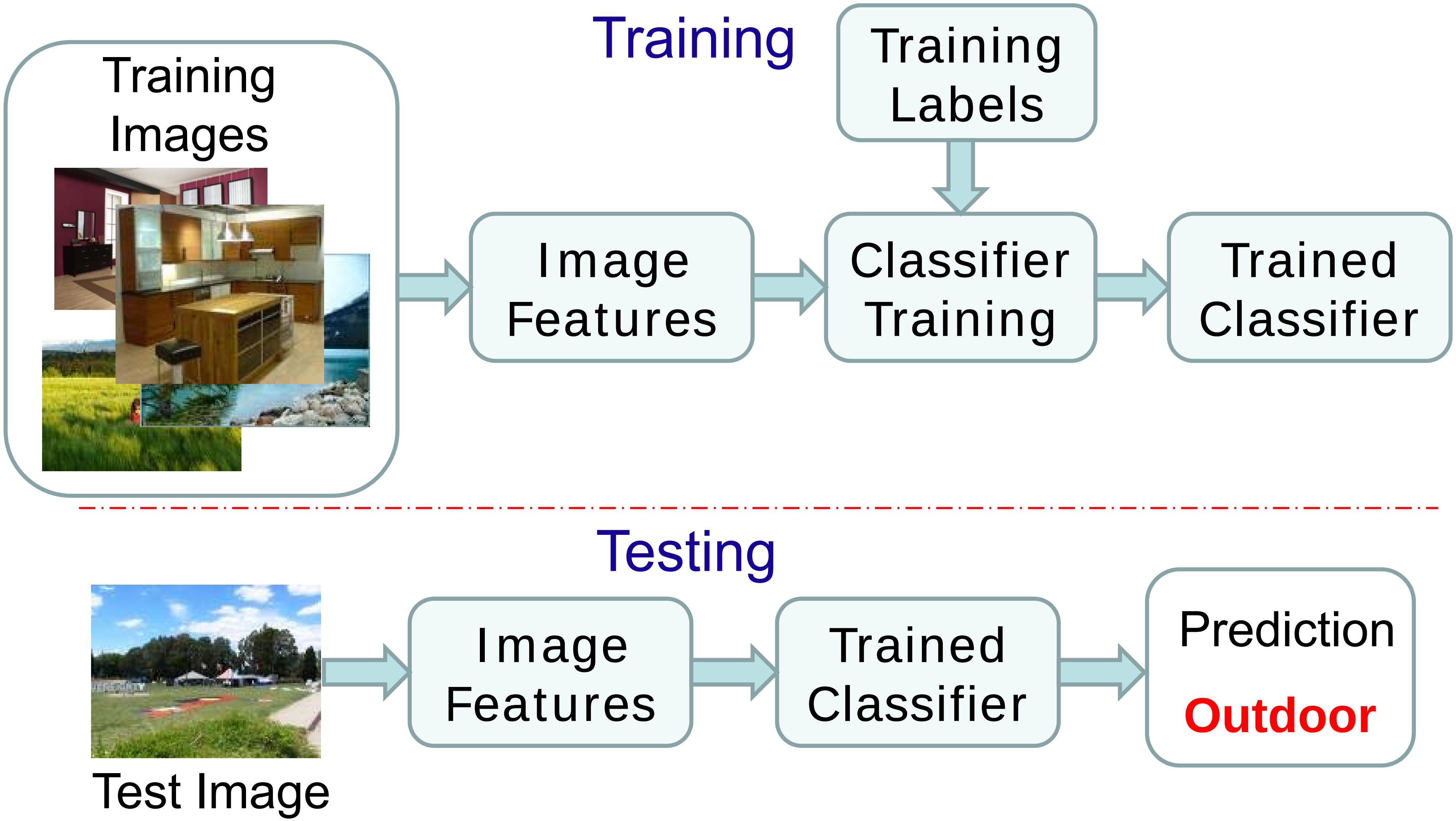
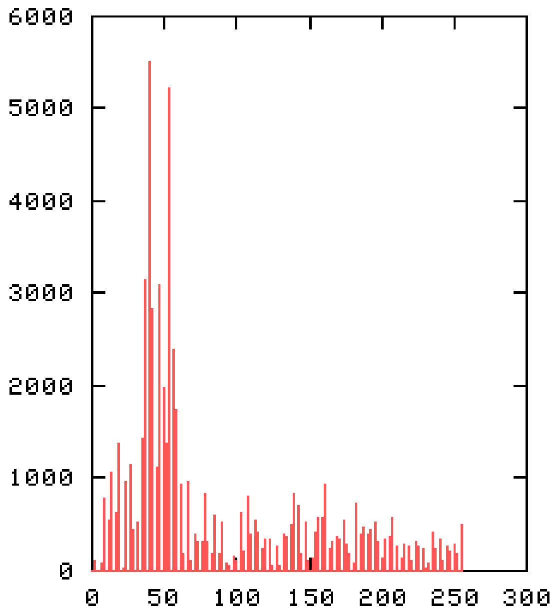


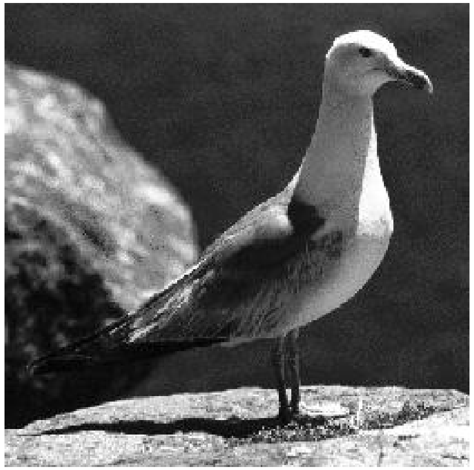
Image Classification: Features

- General Principles
  - Coverage: all relevant info is captured
  - Concision: minimize the number of features
  - Directness: ideal features are independently useful for prediction
- Right features are application-dependent
  - Histograms: color, texture, edge direction, etc.
  - Shape: scene-scale, object-scale, detail-scale
  - Motion: optical flow, tracked points
  - Relationship with other known (or easily detected) objects: distance, size, etc.

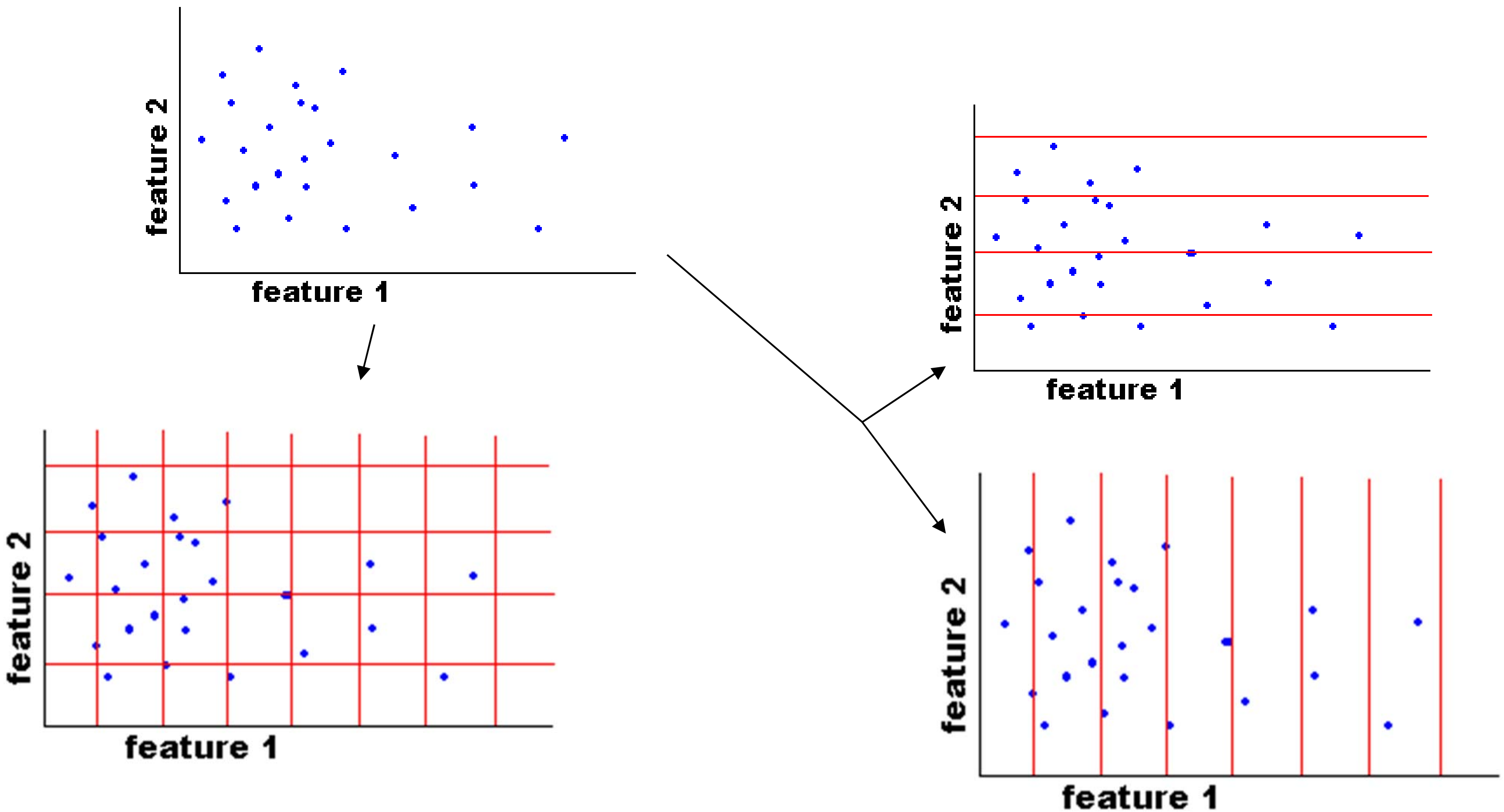
Image Representation

- Image intensity, color, texture, edge direction, etc.
- Histograms: Probability or count of data in each bin





Joint vs. Marginal Histograms



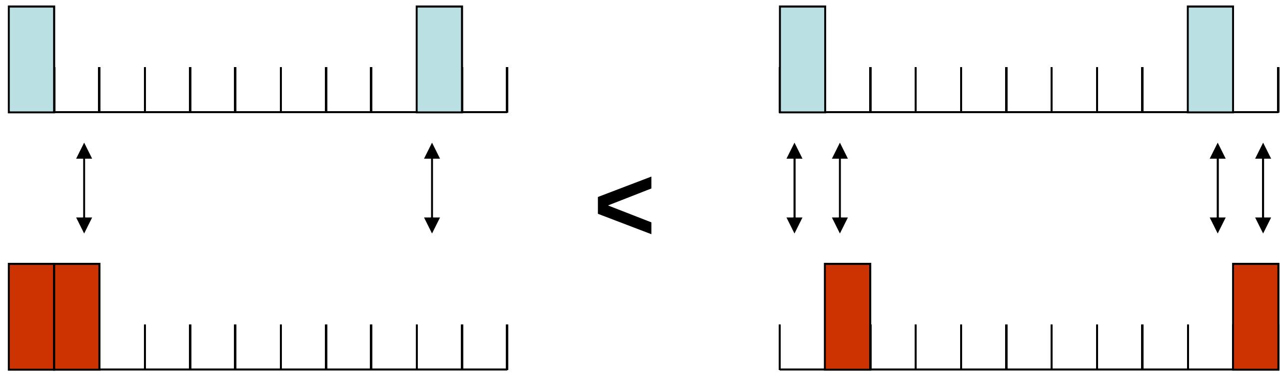
- Requires lots of data
- Loss of resolution to avoid empty bins

- Requires independent features

Minkowski-form Distance

$$D_p(h_1,h_2)=\left[\sum_m \left|h_1(m)-h_2(m)\right|^p\right]^{1/p}$$

- Special cases:
  - L_1: absolute, cityblock, or Manhattan distance
  - L_2: Euclidian distance
- Problem: distance may not reflect the perceived dissimilarity.



Other Distance Measures

- Chi-Squared (χ^2) Distance

$$\chi^2(h_1,h_2)=\frac{1}{2}\sum_{m=1}^{K}\frac{[h_1(m)-h_2(m)]^2}{h_1(m)+h_2(m)}$$

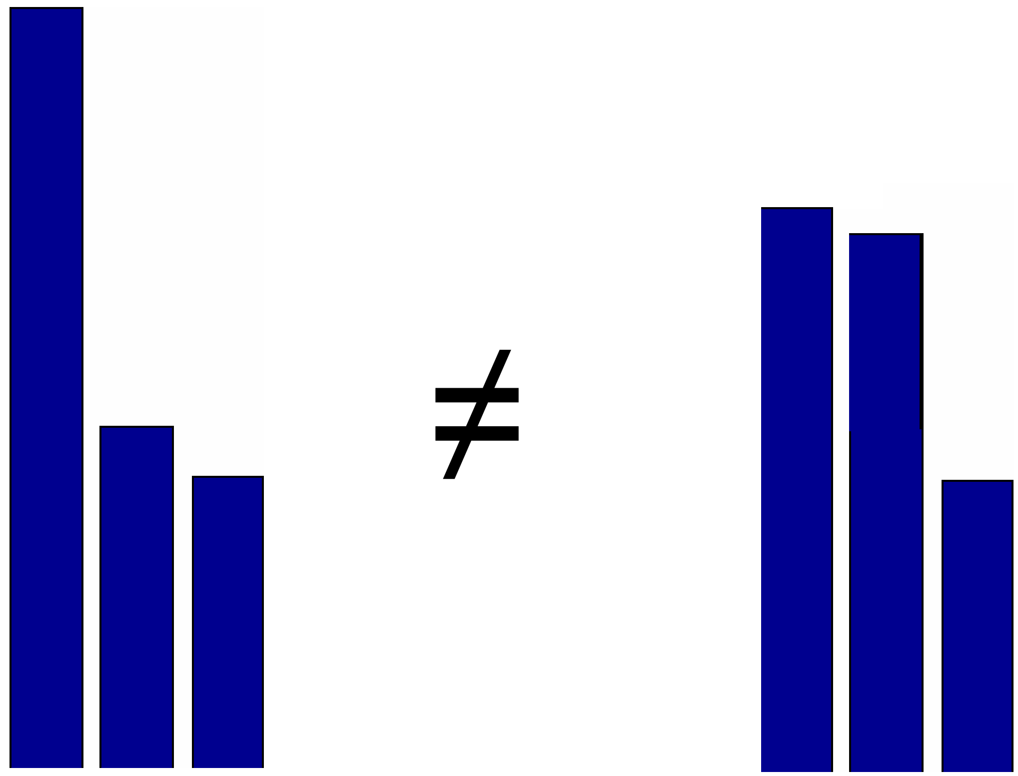
- Kullback-Leibler (KL) Distance

$$D_{KL}(A\parallel B)=-\sum_k P_A(k)\log\frac{P_A(k)}{P_B(k)}$$

- Histogram intersection (for normalized histograms)

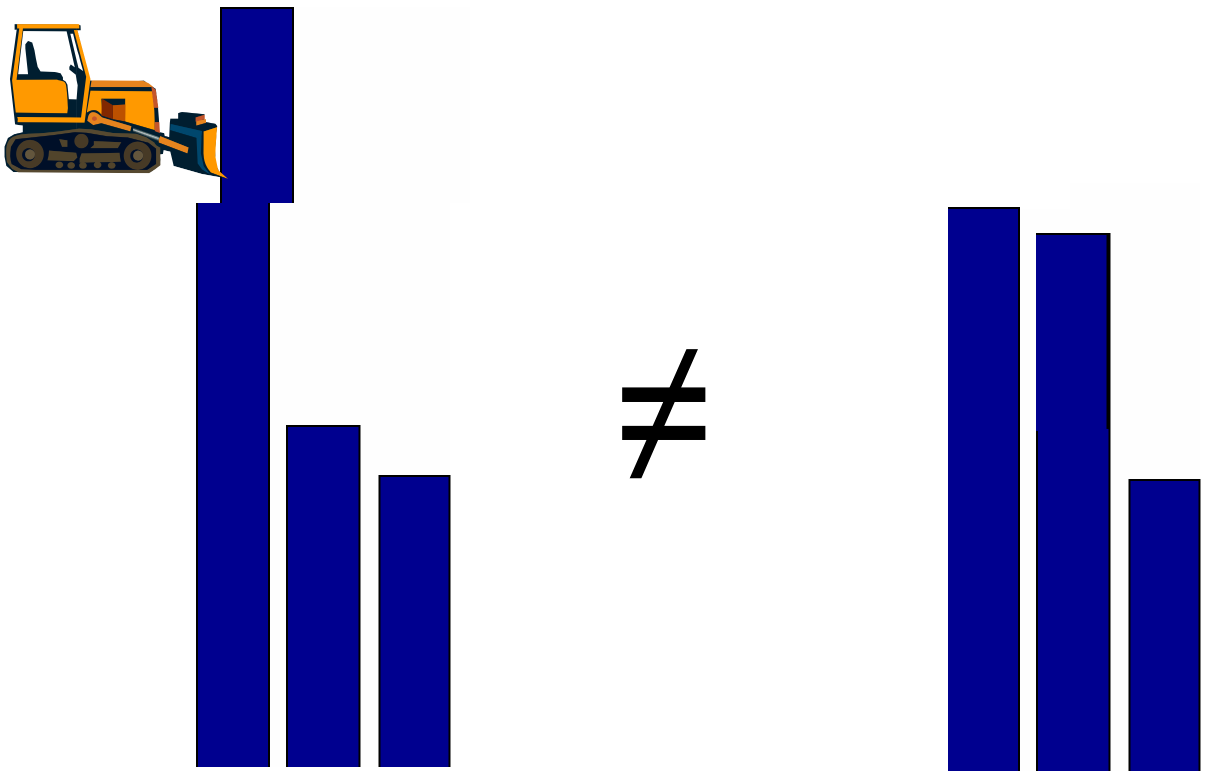
$$D_{\text{int}}(h_1,h_2)=1-\sum_{m=1}^{K}\min\left(h_1(m),h_2(m)\right)$$

Earth Mover's Distance (EMD)



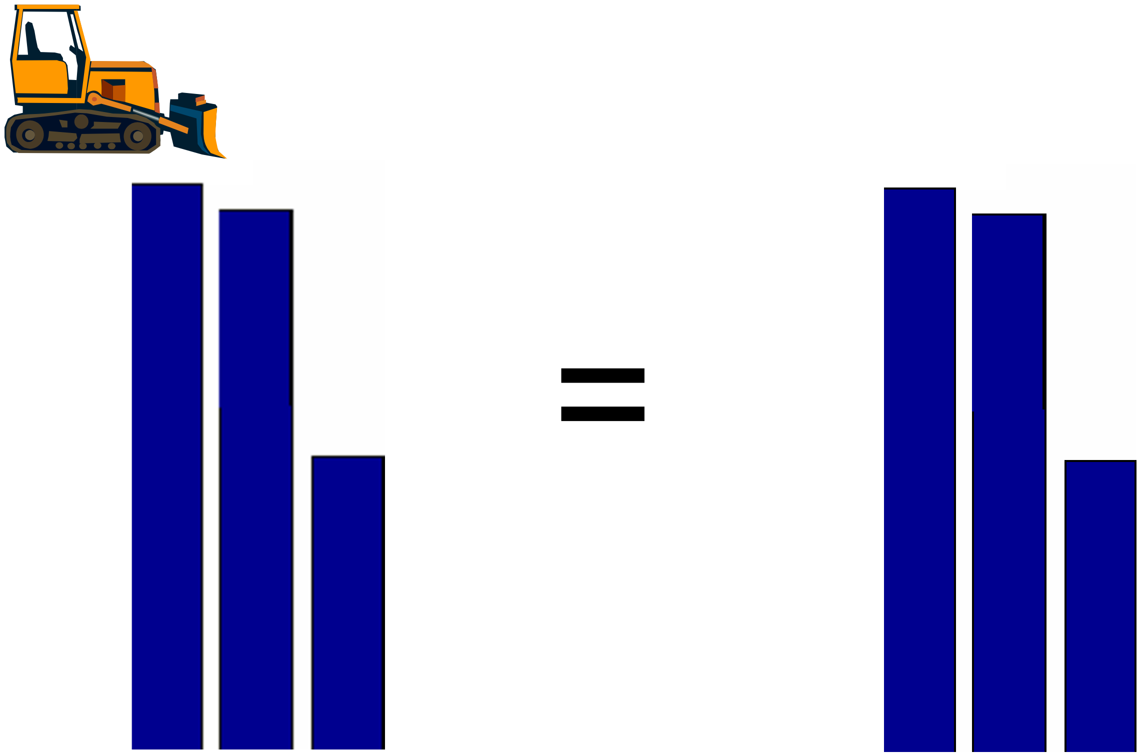
From: Pete Barnum

Earth Mover's Distance (EMD)



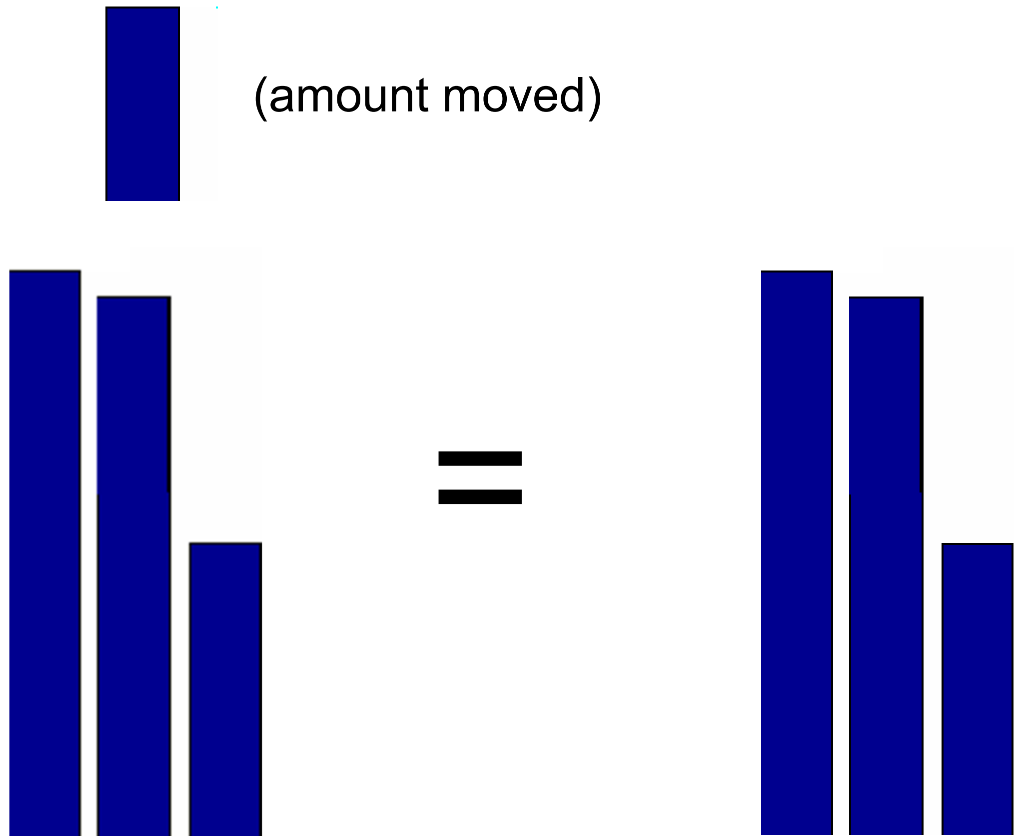
From: Pete Barnum

Earth Mover's Distance (EMD)



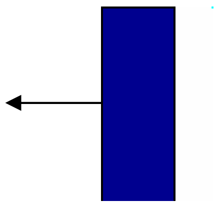
From: Pete Barnum

Earth Mover's Distance (EMD)

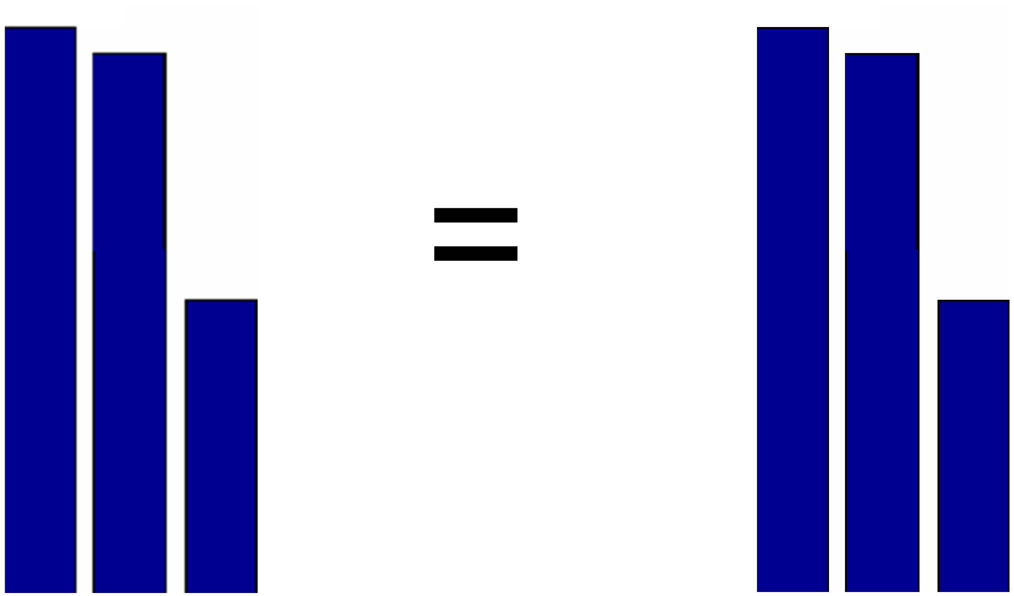


From: Pete Barnum

Earth Mover's Distance (EMD)



(amount moved) * (distance moved) = work



From: Pete Barnum

Earth Mover's Distance (EMD)

$$D_{EMD}(A,B) = \min_F \sum_i \sum_j f_{ij} \cdot d_{ij}$$

$$s.t. \quad f_{ij} \geq 0 \ ; \quad P_B(k) = \sum_i f_{ik} \quad ; \quad P_A(k) \geq \sum_i f_{ki}$$

- Constraints:
  - Move earth only from A to B
  - After the move P_A will be equal to P_B
  - Cannot send more "earth" than there is
- Can be solved using Linear Programming

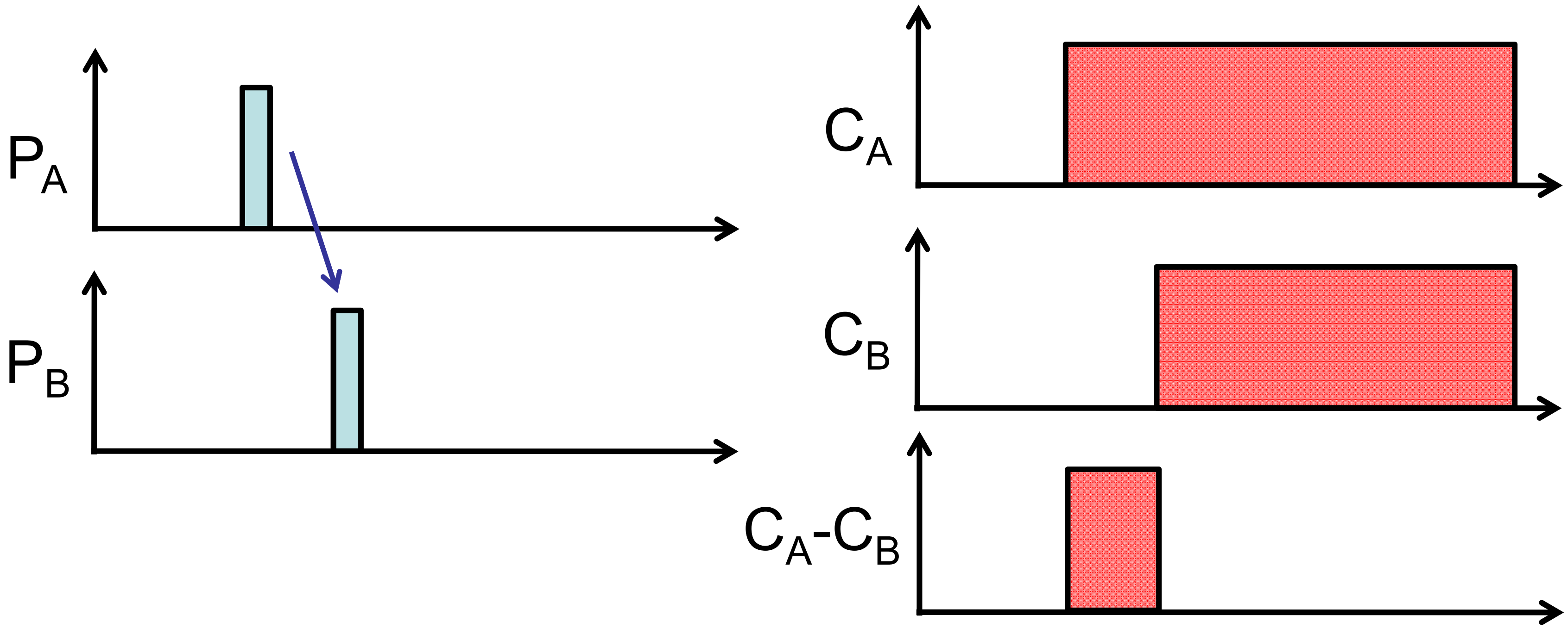
d_{ij} = the "ground distance" between bins i and j.

f_{ij} = amount of data moved from bin *i* to bin *j*

EMD in 1D

- Let C_A and C_B denote the accumulated histograms of image A and B respectively:

$$D_{EMD}(A,B) = \sum_k \left| C_A(k) - C_B(k) \right|$$



Implementation Issues

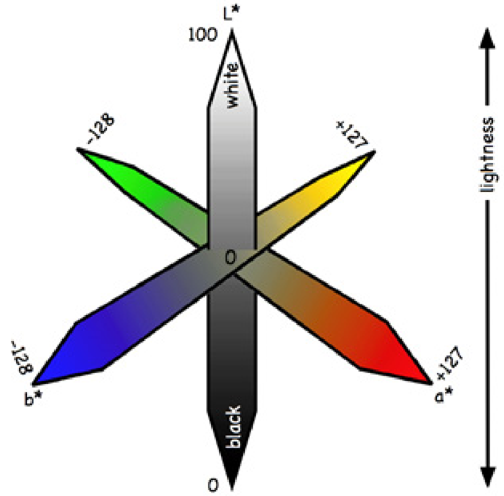
- Quantization
  - –Grids: fast but only applicable with few dimensions
  - –Clustering: slower but can quantize data in higher dimensions

Few Bins
Need less data
Coarser representation

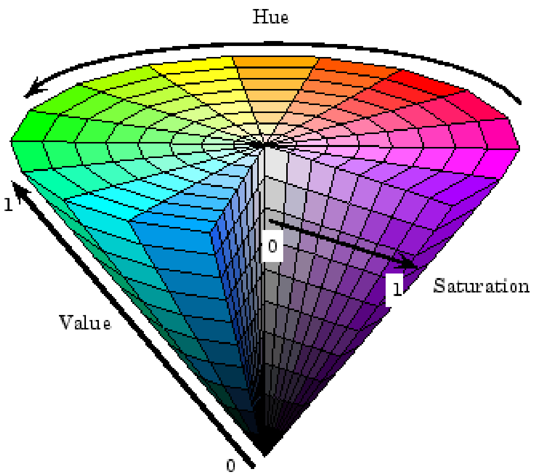
Many Bins
Need more data
Finer representation

- Matching
  - –Histogram intersection or Euclidean are faster
  - –Chi-squared often works better
  - –Earth mover's distance is good when nearby bins represent similar values

Histograms of …

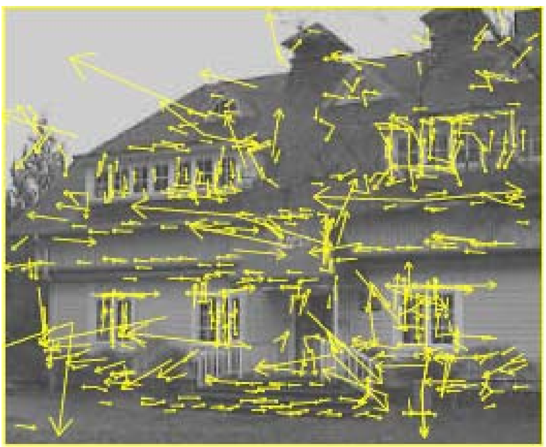


L*a*b* color space



HSV color space

- Color
- Texture
  - Filter banks
  - HOG
- Visual words



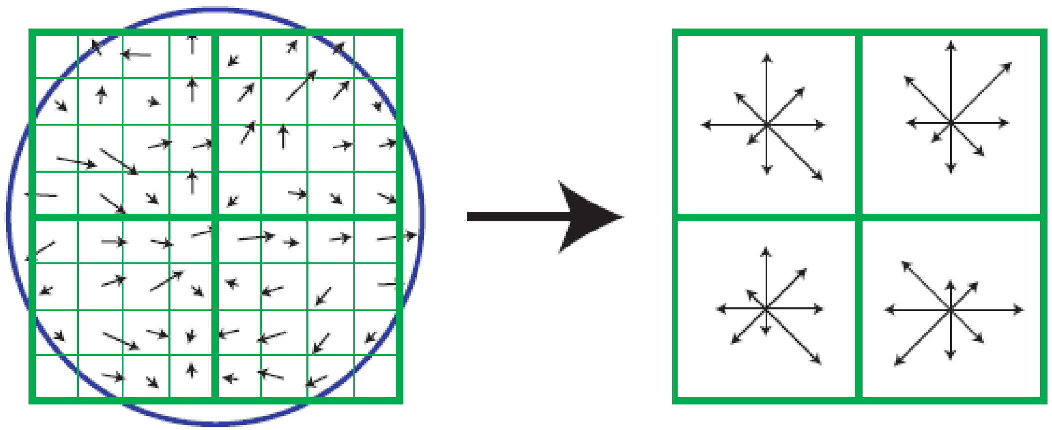


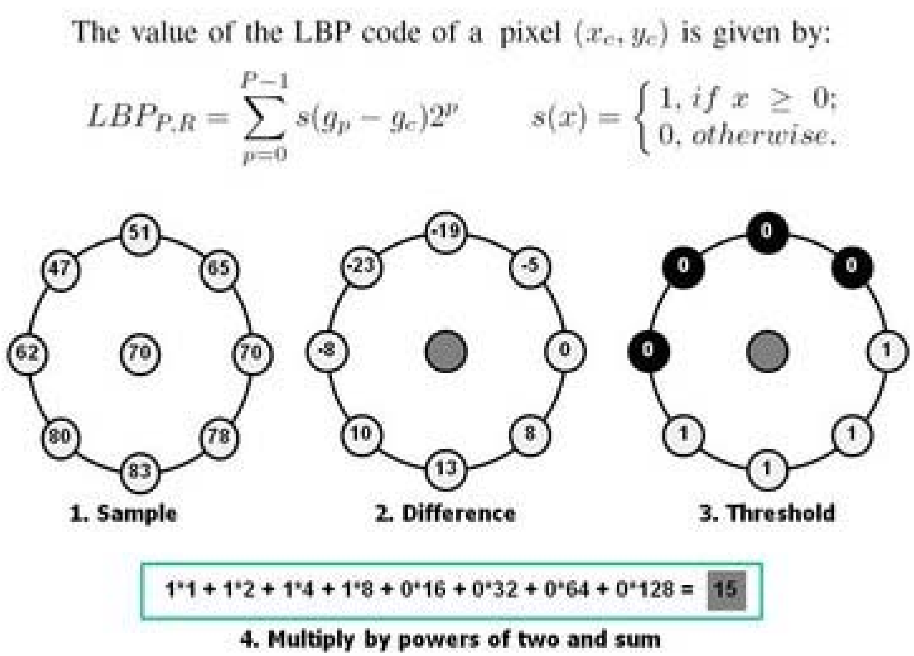
Image gradients

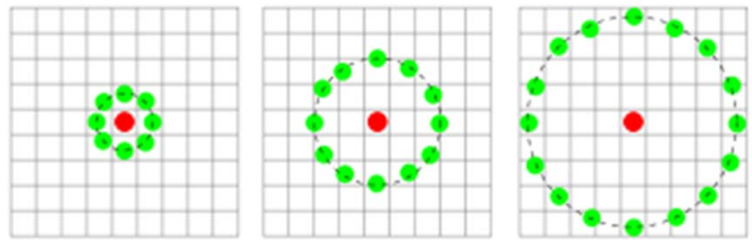
Keypoint descriptor

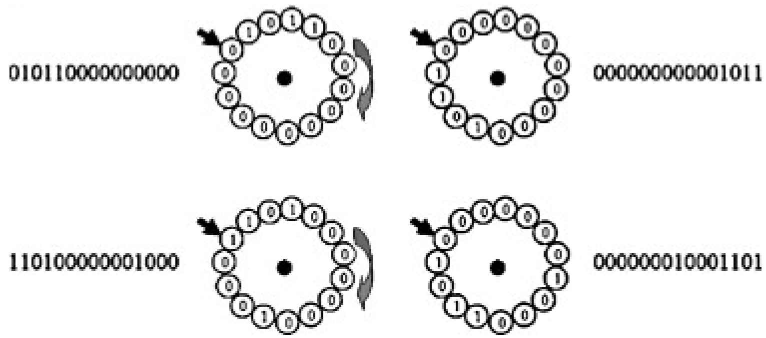
SIFT – Lowe IJCV 2004

Local Binary Patterns

- A simple yet very efficient texture operator which labels the pixels of an image by thresholding the neighborhood of each pixel and considers the result as a binary number.



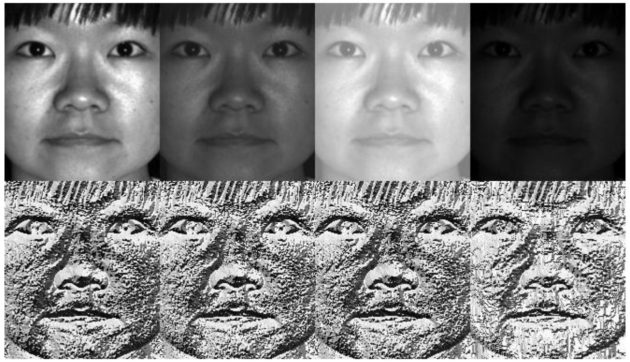


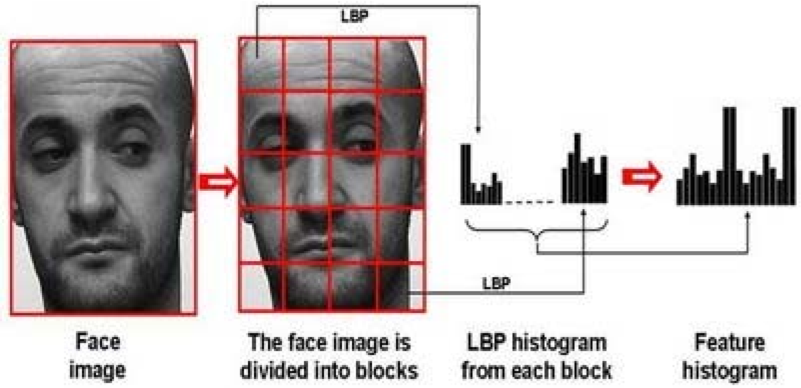


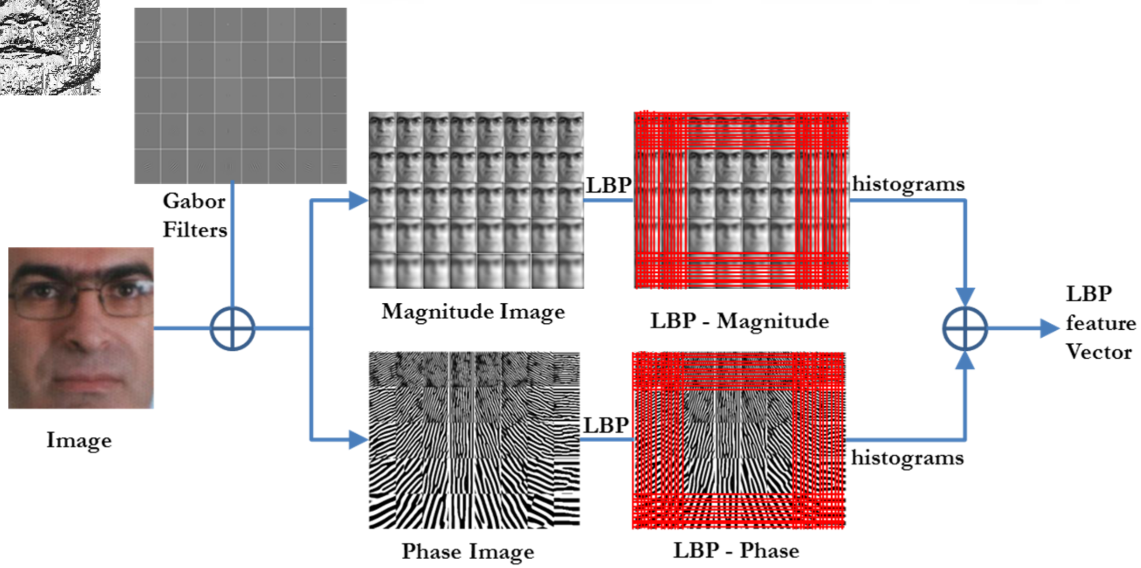
Rotation invariant using circular shift

Local Binary Patterns: Examples

- Texture classification using Local binary patterns
  http://www.youtube.com/watch?gl=SG&hl=en-GB&v=-Ja-vLbHWLc
- Face Recognition







Content-based Image Retrieval

A Bayes Framework for Semantic Classification of Outdoor Vacation Images, by Vailaya et al., 2001.

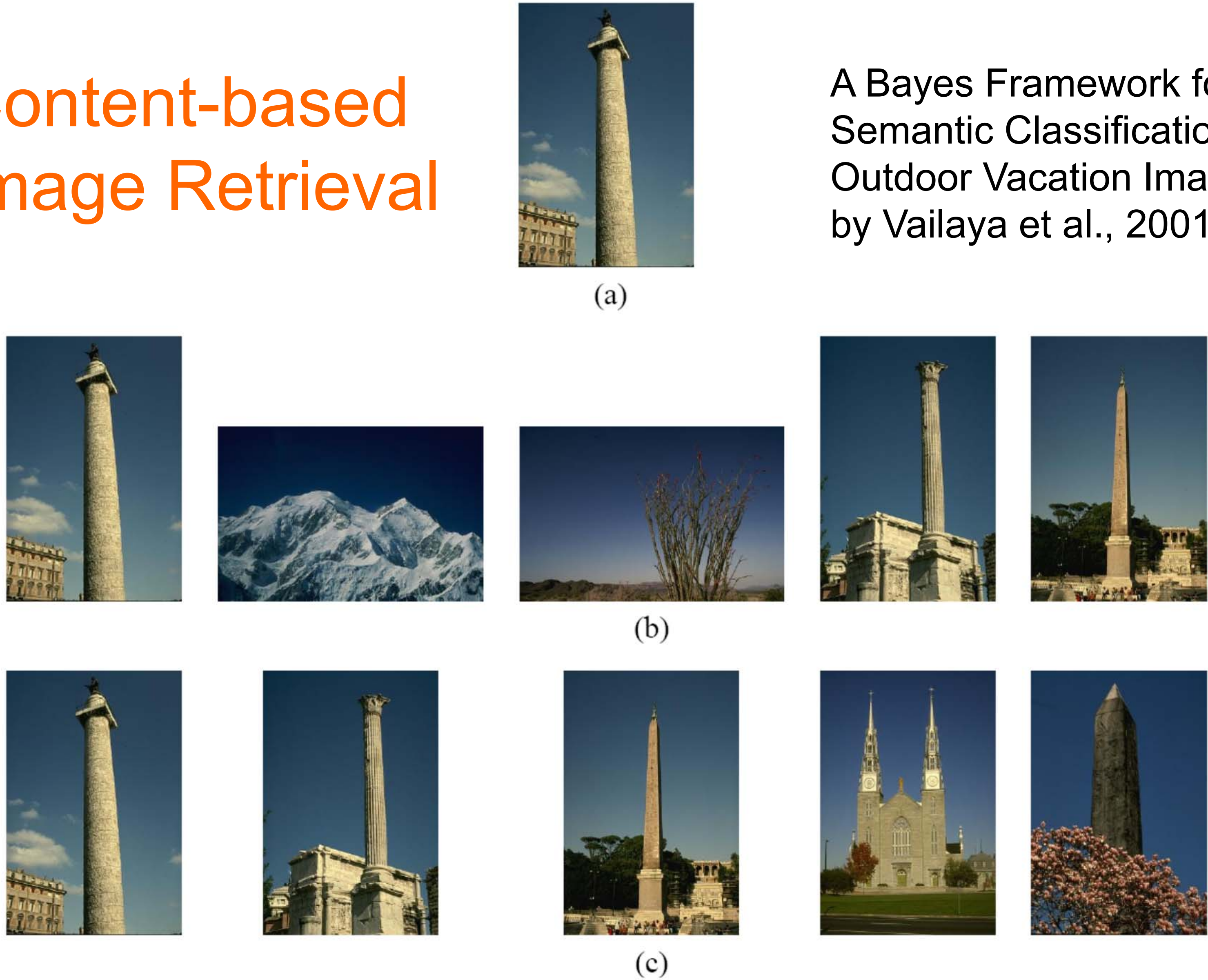
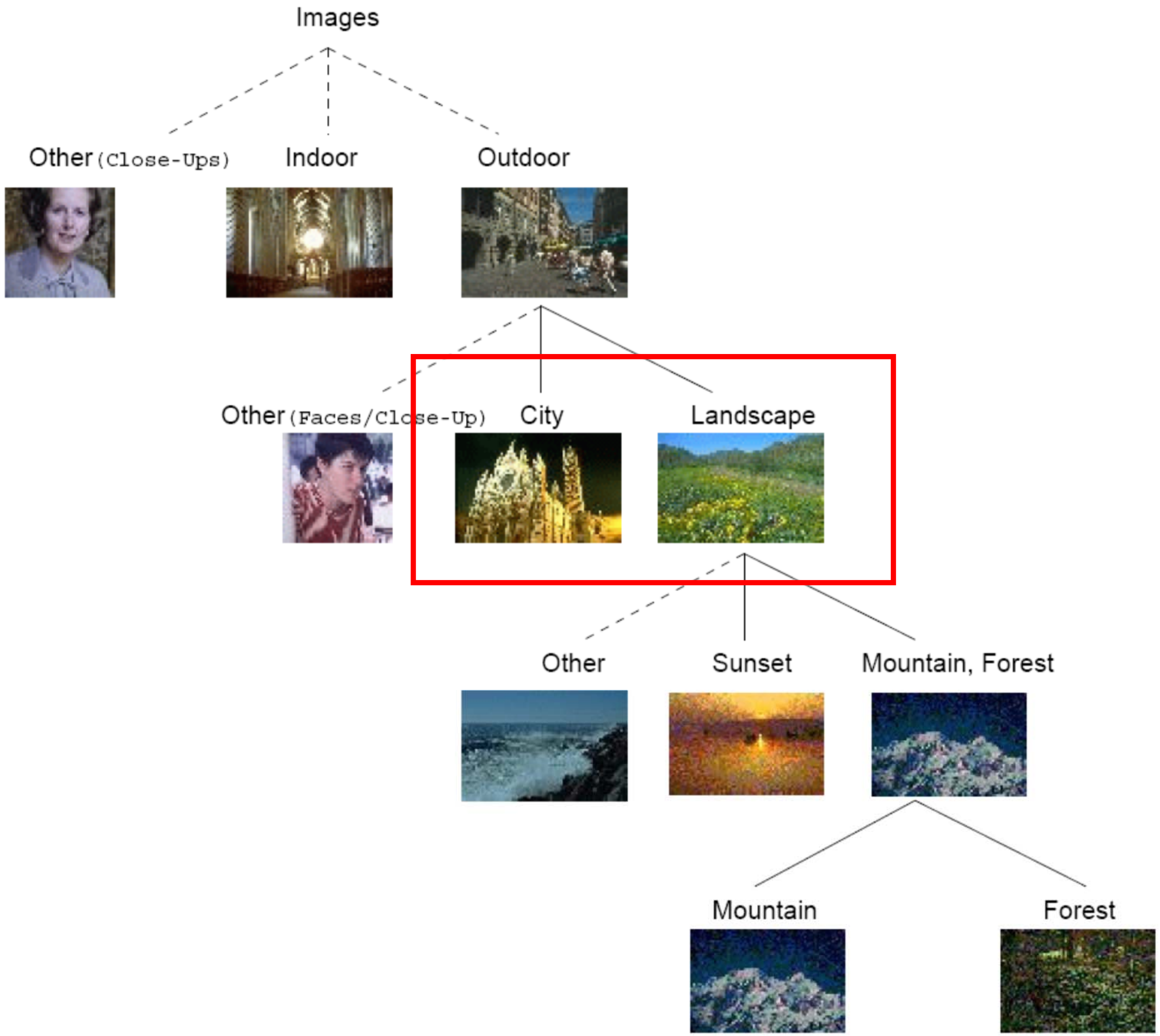


Figure 1: Content-based retrieval results: (a) query image; (b) top-5 retrieved images from a database of 2, 145 city and landscape images; (c) top-5 retrieved images from a database of 760 city images; filtering out landscape images prior to querying clearly improves the retrieval result.

Bayesian Image Classification



Problem and Approach

- City vs. landscape classification
- Database: 2716 images
- Classification accuracy requirement: high!
  - Achieved: 95.3%
- Features used:
  - Color histograms, color coherences, edge direction histograms, and edge direction coherence vectors
- Bayes framework:
  - Vector Quantization (VQ) as a conditional density estimator
  - Classification criterion: Maximum A Posteriori (MAP)

Bayes Framework

- Given image x belongs to one of *K* classes $\Omega = \{\omega_1, \omega_2, \ldots \omega_K\}$
- A priori knowledge $\{P(\omega_1), P(\omega_2), \ldots\ldots P(\omega_K)\}$
- 0/1 loss function $L(\omega_i, \omega_i) = 0, L(\omega_i, \omega_j) = 1, \text{if } i \neq j$
- Decision rule: MAP

Map any observed image into one of the available classes

Classification Rule (MAP)

- Feature extraction: color, edge related

$$\mathbf{y} = \{\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \ldots\ldots \mathbf{y}^{(m)}\}$$

- Assume feature independent:

$$f_{\mathbf{X}}(\boldsymbol{x} \mid \omega) \equiv f_{\mathbf{Y}}(\boldsymbol{y} \mid \omega) = \prod_{i=1}^{M} f_{\mathbf{Y}^{(\mathbf{i})}}(\boldsymbol{y}^{(\boldsymbol{i})} \mid \omega)$$

- Bayes theorem:

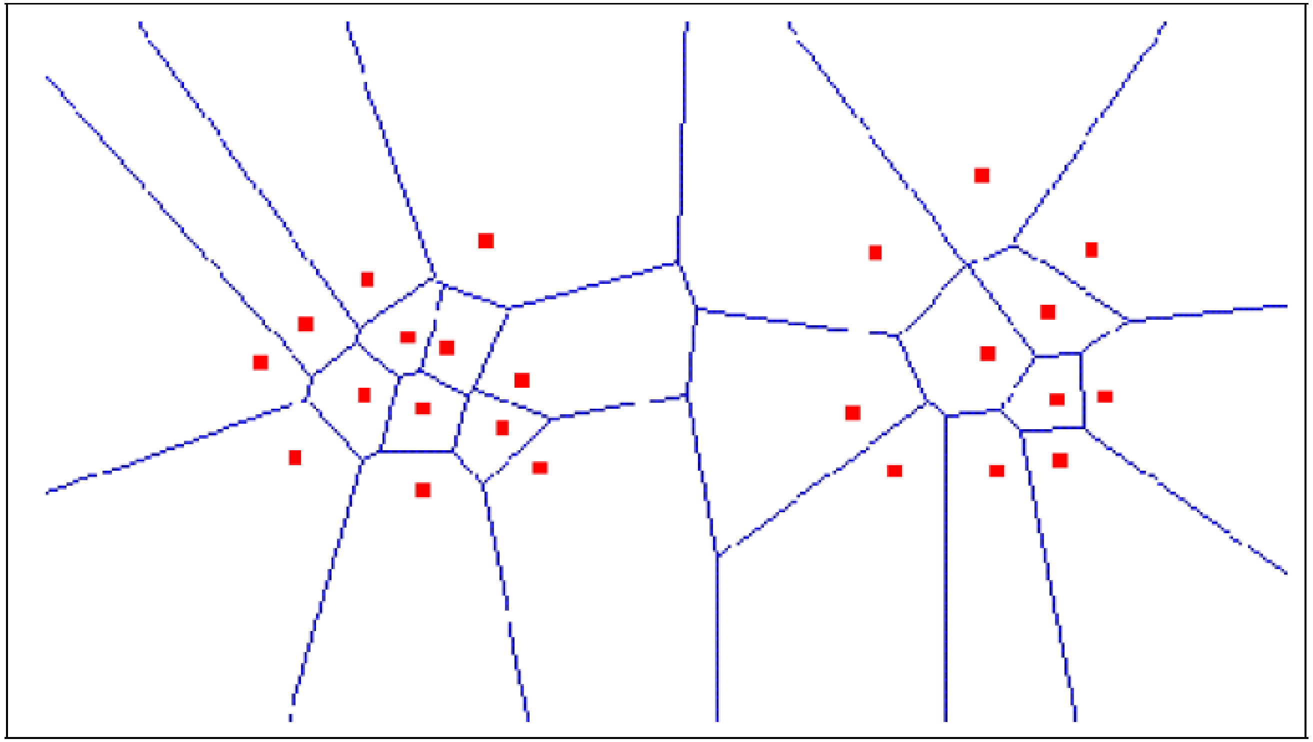
$$P(\omega_j \mid \mathbf{y}) = \frac{f_Y(\mathbf{y} \mid \omega_j) P(\omega_j)}{f_Y(\mathbf{y})}$$

- Classification rule (MAP):

$$\hat{\omega} = \delta(\boldsymbol{x}) = \arg\max_{\omega \in \Omega} \{p(\omega \mid \boldsymbol{y})\} = \arg\max_{\omega \in \Omega} \{f_{\mathbf{Y}}(\boldsymbol{y} \mid \omega)\, p(\omega)\}$$

Vector Quantization

Any input vector y is quantized into one of the q codebook vectors $\mathbf{v}_{\mathbf{j}}^{(\mathbf{i})}$ $(1 \leq j \leq q)$



An input vector is assigned the codebook vector of the cell it falls into.

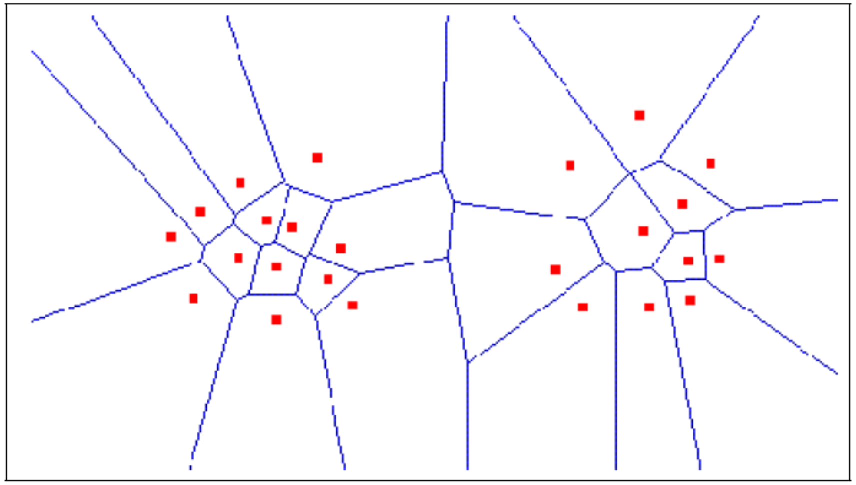
Density Estimation Using VQ

- Class conditional density estimation based on VQ
  - Cell volume is sufficiently small → approximated as a piecewise-constant function over each cell

$$f_{\mathbf{Y}^{(\mathbf{i})}}(\boldsymbol{y}^{(\boldsymbol{i})} \mid \omega) \approx \frac{m_j^{(i)}}{Vol(S_j^{(i)})}$$

$m_j^{(i)}$ ratio of training samples falling into cell $S_j^{(i)}$

$Vol(S_j^{(i)})$ volume of cell $S_j^{(i)}$



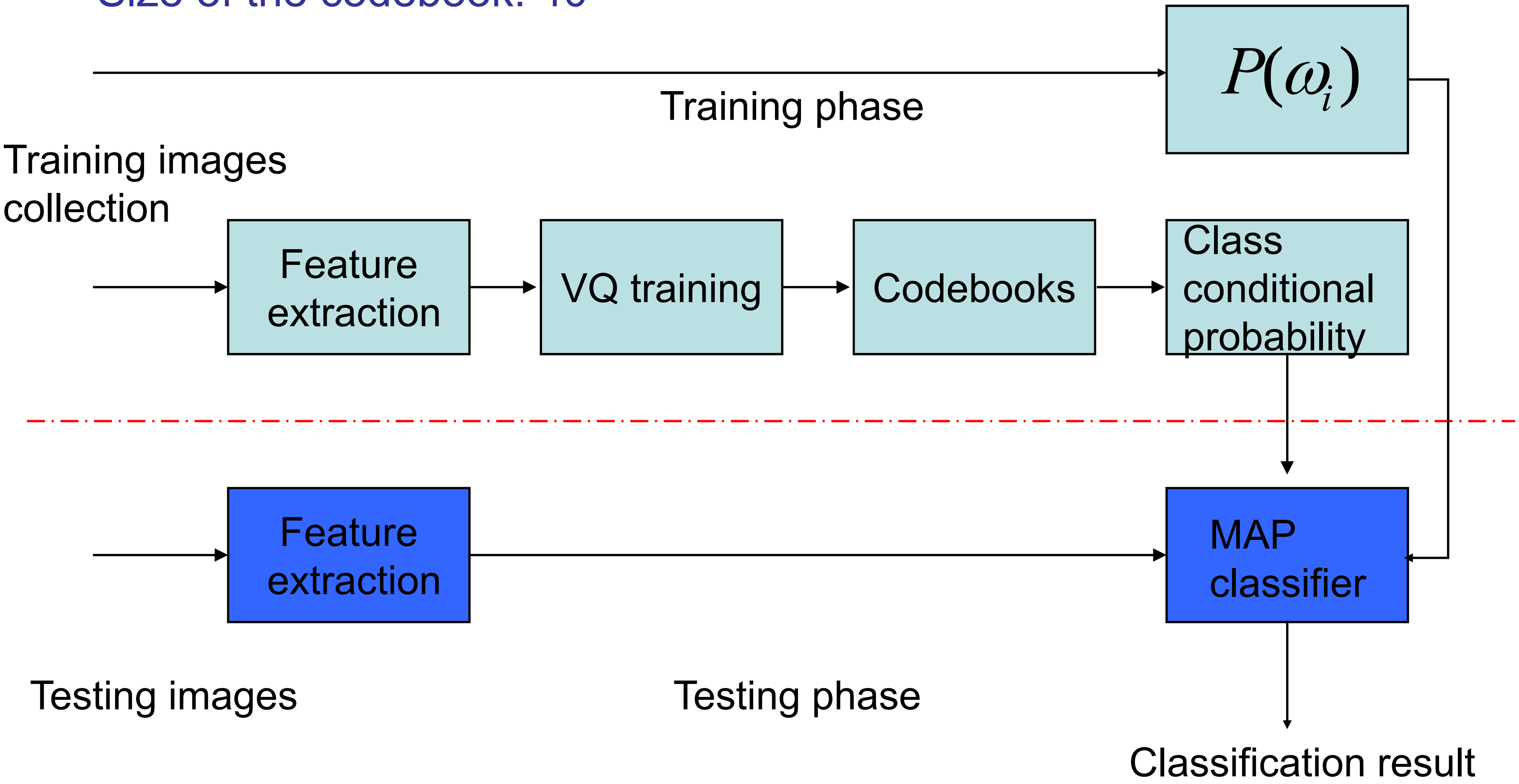
  - Cell volume is NOT sufficiently small → approximated using a kernel-based approach

$$f_{\mathbf{Y}^{(\mathbf{i})}}(\boldsymbol{y}^{(\boldsymbol{i})} \mid \omega) \propto \sum_{j=1}^{q} m_j^{(i)} * \exp(-\|\boldsymbol{y}^{(\boldsymbol{i})} - \boldsymbol{v}_{\boldsymbol{j}}^{(\boldsymbol{i})}\|^2/2)$$

System Architecture

Features: Color Histogram, Edge Direction Histogram, etc.

Size of the codebook: 40



Experimental Results: City vs. Landscape

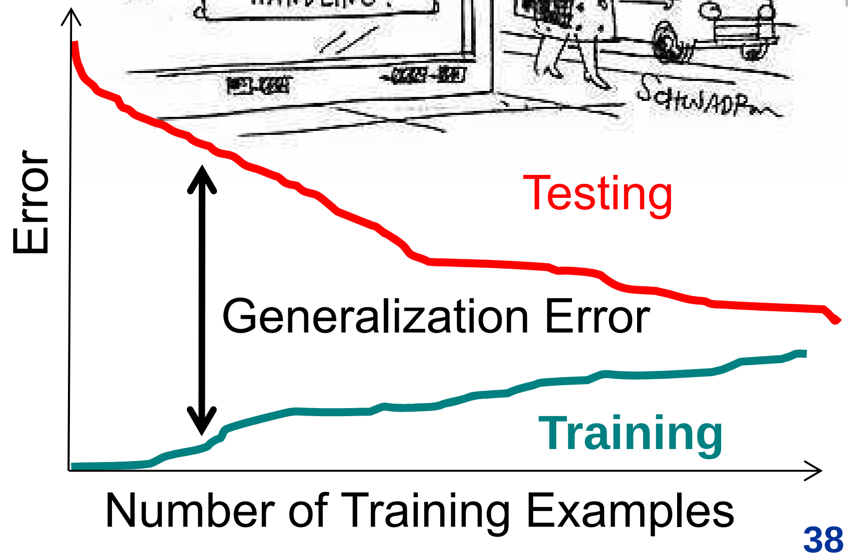
Test Data	EDH	EDCV	CH	CCV	EDH & CH	EDH & CCV	EDCV & CH	EDCV & CCV
Training Set	94.7	96.7	83.7	83.5	94.8	95.4	96.4	96.9
Test Set	92.0	92.7	75.4	76.0	92.5	92.8	93.4	93.8
Entire Database	93.4	94.7	79.6	79.8	93.7	94.1	94.9	95.3

Table 2: Classification accuracies (in %) for city vs. landscape classification problem; the features are abbreviated as follows: edge direction histogram (EDH), edge direction coherence vector (EDCV), color histogram (CH), and color coherence vector (CCV).

Edge information is important for detecting city images!

Summary

- Choose right classifiers based on your needs
  - –Accuracy, speed, memory constraints
  - –Samples available
- Performance evaluation is critical for design of pattern recognition systems.
- It may require many iterative steps to meet applications' needs.
  - –Design, implementation, and evaluation



Online Resources

- Standard/open databases for performance benchmarking of algorithms and systems
  - Face Recognition: http://www.face-rec.org/
  - Objects and Scenes: http://web.mit.edu/torralba/www/database.html
  - Content-based Image Retrieval: http://www.cs.washington.edu/research/imagedatabase/
  - Cross Language Image Retrieval: http://imageclef.org/
  - UC Irvine Machine Learning Repository: http://archive.ics.uci.edu/ml/
  - CV online: http://homepages.inf.ed.ac.uk/rbf/CVonline/
  - Online Demonstrations for Statistical Pattern Recognition
  - Many more…

Sample Exam Questions

Question 1

Consider a two-category classification problem with two-dimensional feature vector $\mathbf{x} = [x_1 \quad x_2]^T$. The two categories are ω_1 and ω_2, and the feature vector is Gaussian distributed with mean vectors and covariance matrices given by

$$\boldsymbol{\mu}_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \qquad \boldsymbol{\Sigma}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\boldsymbol{\mu}_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad \boldsymbol{\Sigma}_2 = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}$$

Assume that the two classes are equally likely. Derive the Bayes decision rule and sketch the decision boundary.

Solution to Question 1

The inverse covariance matrices are:

$$\boldsymbol{\Sigma}_1^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad \boldsymbol{\Sigma}_2^{-1} = \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix}$$

Since the priors and the determinants are the same, the discriminant functions are:

$$g_i(\mathbf{x}) = -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^{\mathrm{T}} \boldsymbol{\Sigma}_i^{-1} (\mathbf{x}-\boldsymbol{\mu}_i)$$

As there are only two alternative decisions, we can use a single discriminant function

$$g(\mathbf{x}) = g_1(\mathbf{x}) - g_2(\mathbf{x}) = -\frac{1}{2}\left[(\mathbf{x}-\boldsymbol{\mu}_1)^{\mathrm{T}} \boldsymbol{\Sigma}_1^{-1} (\mathbf{x}-\boldsymbol{\mu}_1) - (\mathbf{x}-\boldsymbol{\mu}_2)^{\mathrm{T}} \boldsymbol{\Sigma}_2^{-1} (\mathbf{x}-\boldsymbol{\mu}_2)\right]$$

Solution to Question 1

Single discriminant function:

$$g(\mathbf{x}) = -\frac{1}{2}\left[(\mathbf{x}-\boldsymbol{\mu}_1)^{\mathrm{T}}\boldsymbol{\Sigma}_1^{-1}(\mathbf{x}-\boldsymbol{\mu}_1) - (\mathbf{x}-\boldsymbol{\mu}_2)^{\mathrm{T}}\boldsymbol{\Sigma}_2^{-1}(\mathbf{x}-\boldsymbol{\mu}_2)\right]$$

$$= -\frac{1}{2}\left\{\left[x_1^2 + (x_2 - 1)^2\right] - \left[2x_1^2 - 2x_1x_2 + x_2^2\right]\right\}$$

$$= -\frac{1}{2}\left[-x_1^2 + 2x_1x_2 - 2x_2 + 1\right] \propto (x_1 - 1)(x_1 + 1 - 2x_2)$$

The optimal classifier should

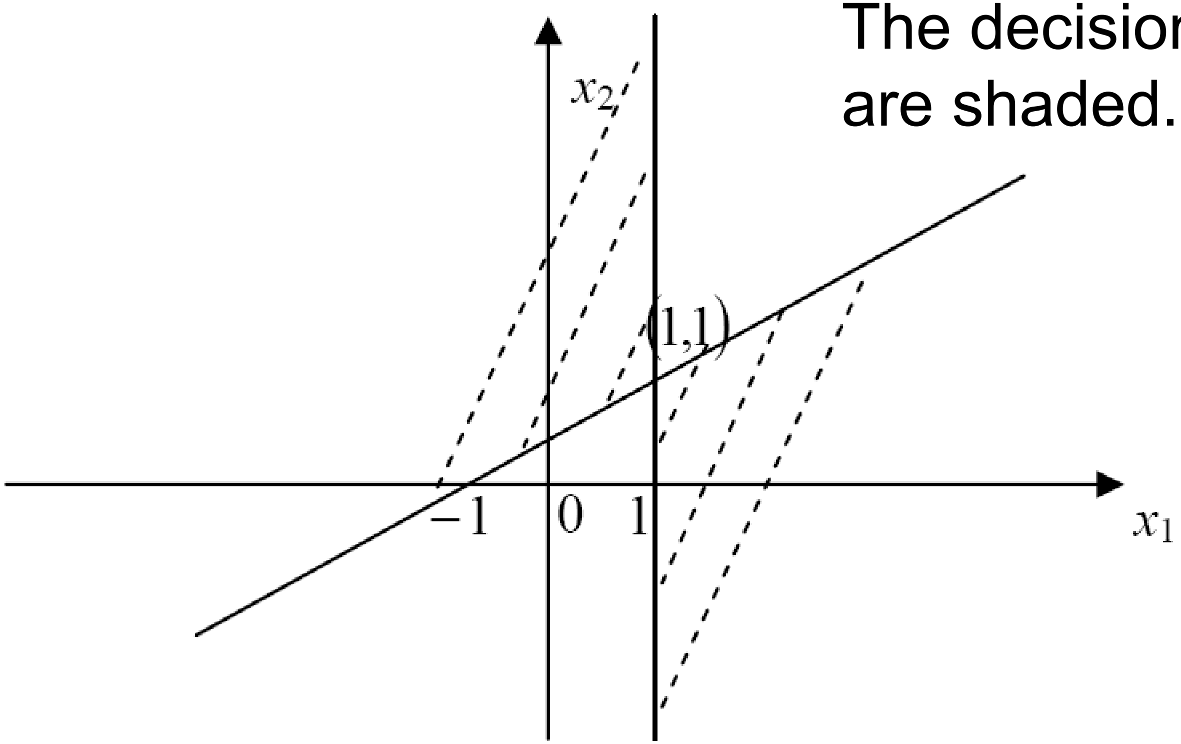
decide ω_1 if $(x_1 - 1)(x_1 + 1 - 2x_2) > 0$

decide ω_2 otherwise.

Solution to Question 1

The decision boundary consists of two lines

$$x_1 - 1 = 0 \quad \text{and} \quad x_1 + 1 - 2x_2 = 0$$



The decision regions for ω_1 are shaded.

Question 2

- Suppose that *N* samples $x_1, x_2 \ldots, x_N$ are drawn independently according to the following probability density function:

$$p(x \mid \theta) = \begin{cases} 2\theta x \exp\left(-\theta x^2\right) & x \geq 0; \\ 0 & \text{otherwise.} \end{cases}$$

- Here the parameter $\theta > 0$ is fixed but unknown. Derive the maximum likelihood estimate of the parameter θ

Solution to Question 2

- Find the likelihood $p(D \mid \theta) = p(x_1, x_2, \ldots, x_N \mid \theta)$

$$p(x_1, x_2, \ldots, x_n \mid \theta) = \prod_{k=1}^{N} p(x_k \mid \theta)$$

$$p(x \mid \theta) = \begin{cases} 2\theta x \exp(-\theta x^2) & x \geq 0; \\ 0 & \text{otherwise.} \end{cases}$$

- Log-likelihood

$$l(\theta) = \sum_{k=1}^{N} \ln p(x_k \mid \theta)$$

$$l(\theta) = \sum_{k=1}^{N} \left[\ln(2\theta x_k) - \theta x_k^2 \right]$$

Solution to Question 2 (Cont'd)

- Log-likelihood $l(\theta) = \sum_{k=1}^{N} \left[\ln(2\theta x_k) - \theta x_k^2 \right]$

- Taking derivative

$$\nabla_\theta l(\theta) = \sum_{k=1}^{N} \left(\frac{1}{\theta} - x_k^2 \right)$$

$$= \frac{N}{\theta} - \sum_{k=1}^{N} x_k^2$$

Let $\nabla_\theta l(\theta) = 0 \Rightarrow \frac{N}{\theta} = \sum_{k=1}^{N} x_k^2 \Rightarrow \widehat{\theta} = \frac{N}{\sum_{k=1}^{N} x_k^2}$

Question 3

- Briefly describe how one computes the k-nearest-neighbor estimate $\hat{p}_{knn}(x)$
- Given 4 data points:

$$x_1 = 2, \quad x_2 = 4, \quad x_3 = 5, \quad x_4 = 7$$

- For $k=2$, sketch $\hat{p}_{knn}(x)$

Solution to Question 3

- The kNN estimate is

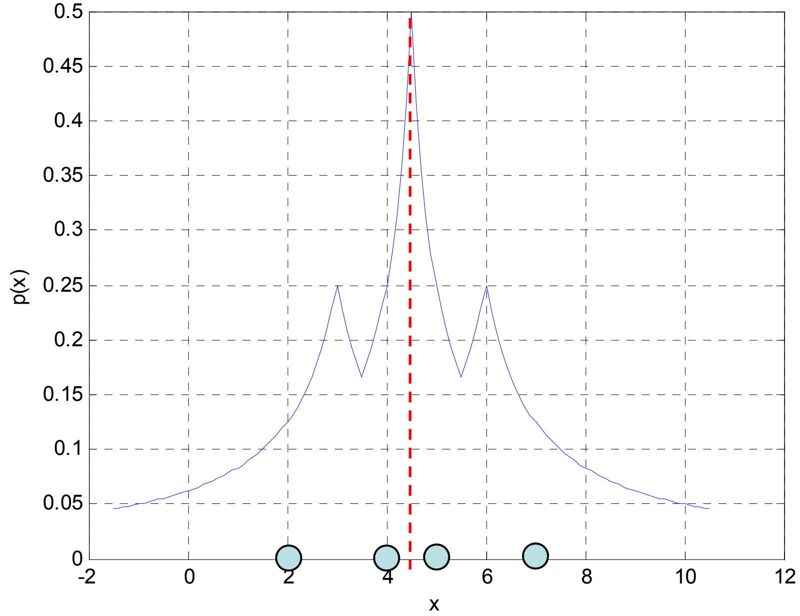
$$\hat{p}_{knn}(x) = \frac{k/N}{V}$$

- Given 4 data points: $x_1 = 2, \quad x_2 = 4, \quad x_3 = 5, \quad x_4 = 7$

- The Sketch for $k=2$

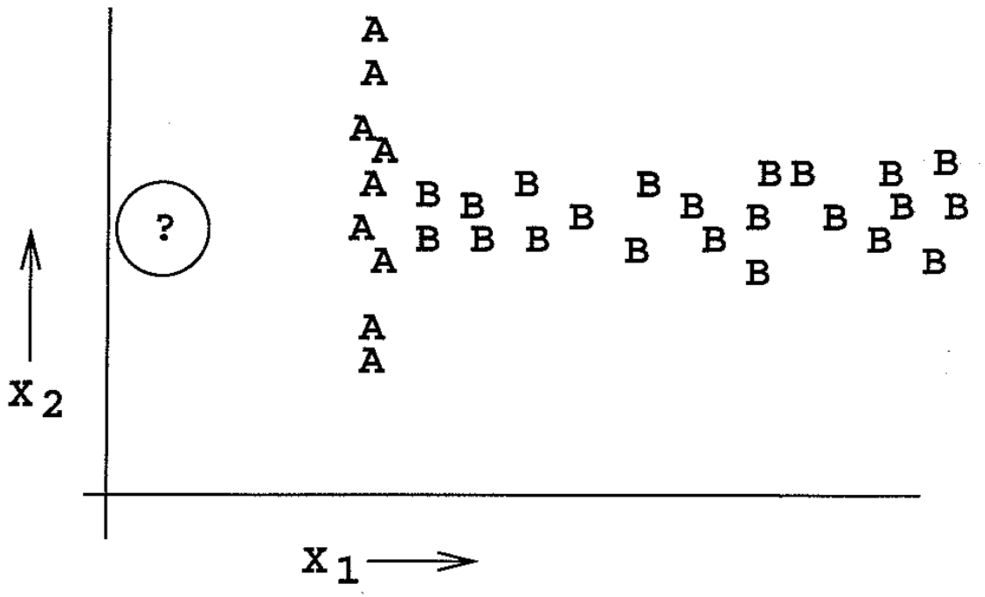
$$N = 4$$

$$\hat{p}_{knn}(x) = \frac{1/2}{V}$$

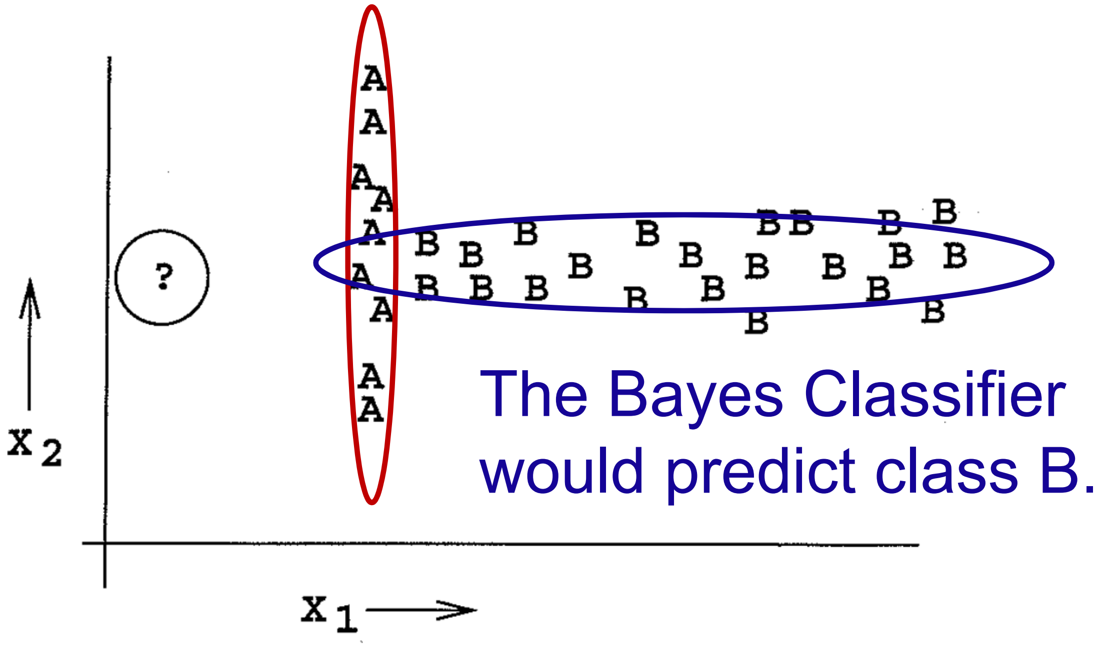


Question 4

Suppose you train a Bayes Classifier for a two-class (i.e., classes A and B) problem, assuming that the class-conditional probability density function is Gaussian. The training examples are shown in the figure below. Which class would the Bayes Classifier predict for a test sample at the location indicated by the question mark? (Answer by reasoning – no need to calculate out anything.)



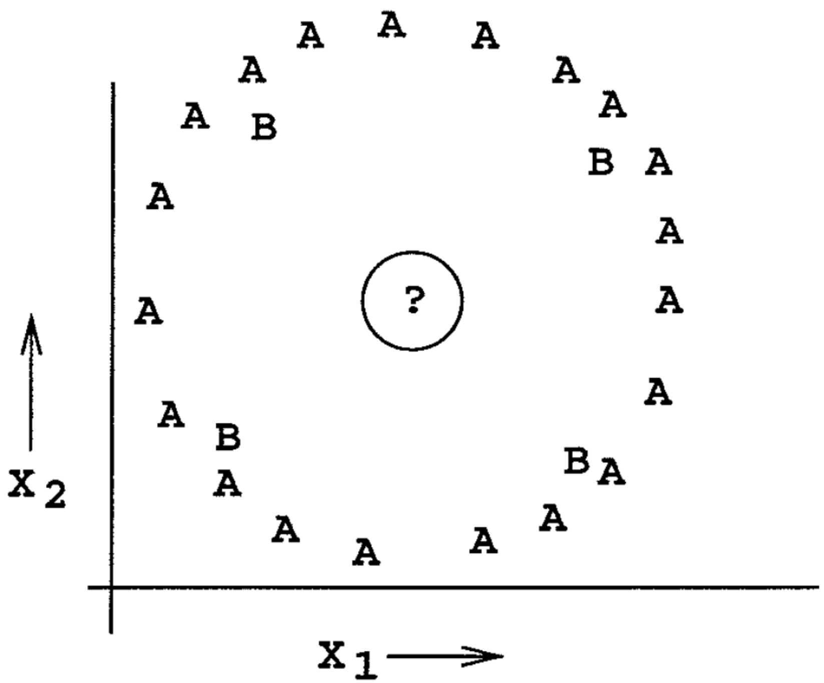
Solution to Question 4



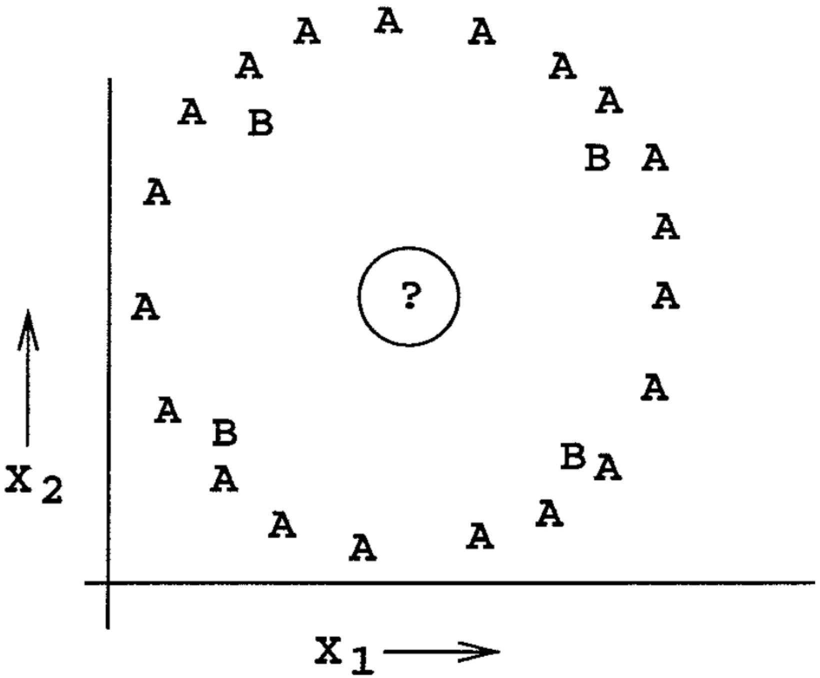
In terms of the Mahalanobis distance, the test sample is much closer to the center of the Gaussian distribution of class B, than to the center of the Gaussian distribution of class A. In addition, the estimated prior probability of class B is also larger than that of class A.

Question 5

Suppose you train a Bayes Classifier for a two-class (i.e., classes A and B) problem, assuming that the class-conditional probability density function is Gaussian. The training examples are shown in the figure below. Which class would the Bayes Classifier predict for a test sample at the location indicated by the question mark?



Solution to Question 5

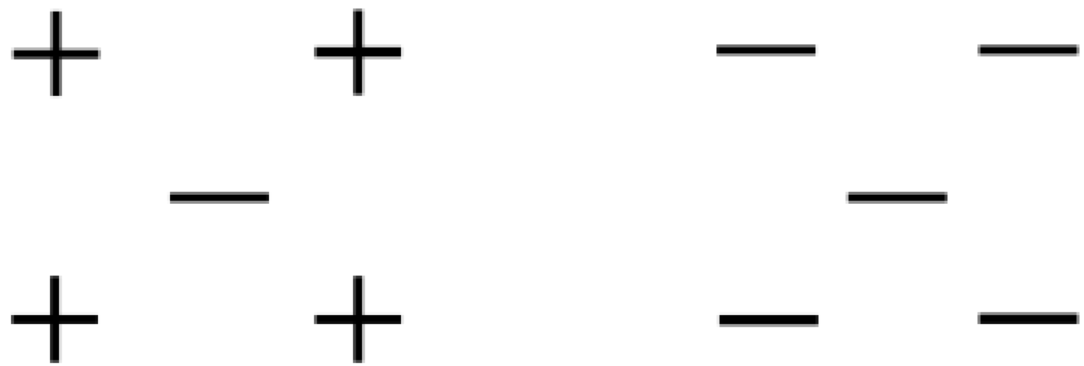


The Bayes Classifier would predict class A.

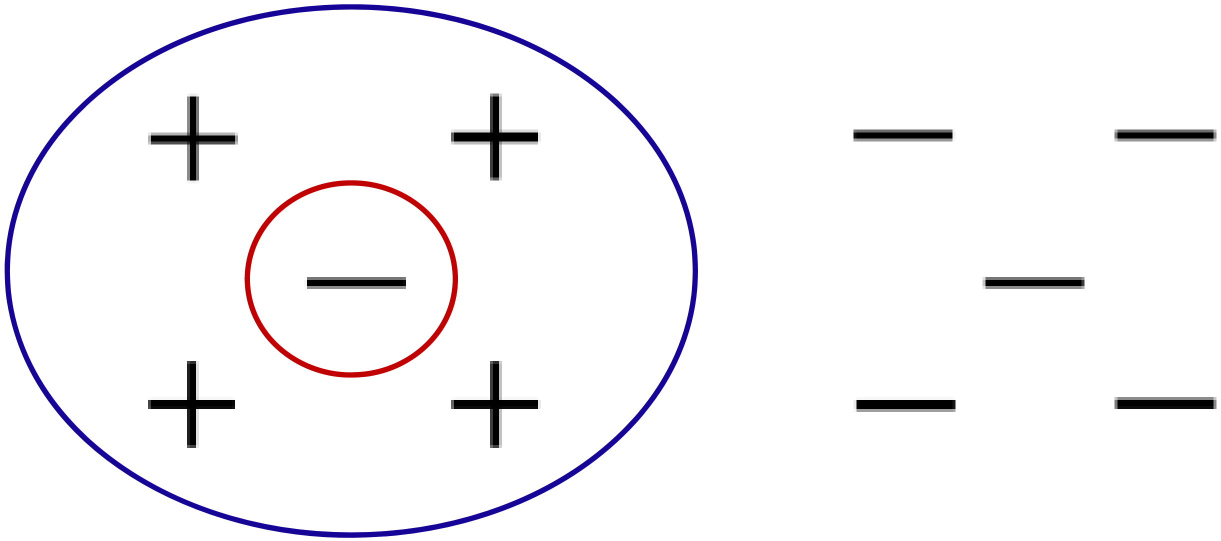
Both classes have the same center and similar standard deviation. Hence, their estimated probability density functions are almost the same. At the center, the density of class B is slightly higher than that of class A. However, the estimated prior probability of class A is much larger than that of class B. Based on the MAP rule, the Bayes classifier would predict class A.

Question 6

The figure below shows samples from two classes marked respectively by “+” and “—“. If you use the 1-NN rule to perform classification, what is the leave-one-out cross validation error? If instead you use the 3-NN rule to perform classification, will the leave-one-out cross validation error increase or decrease?



Solution to Question 6



For 1-NN, the leave-one-out cross validation error is 5/10.
For 3-NN, the leave-one-out cross validation error is 1/10.
Hence, if the 3-NN rule is used to perform classification, the leave-on-out cross validation error will decrease.

Question 7

Suppose a bank classifies customers as either good or bad credit risks. Based on historical data, the bank has observed that 1% of good credit risks and 10% of bad credit risks overdraw their accounts in any given month. A new customer opens a checking account at this bank. By checking with a credit bureau, the bank determines there is a 70% chance that the customer will turn out to be a good credit risk.

a) Suppose that this customer's account is overdrawn in the first month. How does this alter the bank's opinion of this customer's creditworthiness?

b) Given a), what would be the bank's opinion of the customer's creditworthiness at the end of the second month if there was not an overdraft in the second month?

Solution to Question 7

There are two categories:

ω_1: The customer is a good credit risk.
ω_2: The customer is a bad credit risk.

Priors: $P(\omega_1) = 0.7$ and $P(\omega_2) = 0.3$

Likelihoods: $P(\text{overdrawn} \mid \omega_1) = 0.01$

$$P(\text{overdrawn} \mid \omega_2) = 0.1$$

$$P(\omega_1 \mid \text{overdrawn}) = \frac{P(\text{overdrawn} \mid \omega_1)P(\omega_1)}{P(\text{overdrawn} \mid \omega_1)P(\omega_1) + P(\text{overdrawn} \mid \omega_2)P(\omega_2)}$$

$$P(\omega_1 \mid \text{overdrawn}) = \frac{0.01 \times 0.7}{0.01 \times 0.7 + 0.1 \times 0.3} = 0.1892$$

Solution to Question 7

There are two categories:

ω_1: The customer is a good credit risk.
ω_2: The customer is a bad credit risk.

Given a), the priors become: $P(\omega_1) = 0.1892$ and $P(\omega_2) = 0.8108$

Likelihoods: $P(\text{overdrawn} \mid \omega_1) = 0.01$

$$P(\text{overdrawn} \mid \omega_2) = 0.1$$

$$P(\omega_1 \mid \text{not overdrawn}) = \frac{P(\text{not overdrawn} \mid \omega_1)P(\omega_1)}{P(\text{not overdrawn} \mid \omega_1)P(\omega_1) + P(\text{not overdrawn} \mid \omega_2)P(\omega_2)}$$

$$P(\omega_1 \mid \text{not overdrawn}) = \frac{0.99 \times 0.1892}{0.99 \times 0.1892 + 0.9 \times 0.8108} = 0.2043$$