Note:

- Homework #4 submission closed
- Homework #5 (Last one ☺)
  - On Lecture 10-11, Due in one weeks, on *Canvas* now.
  - Submit your PDF file by 4/22 Tuesday 4pm
- Literature survey study tips
  - **Do not procrastinate it.**
  - Ask questions if any
- Fast Prototyping Exercise #3 on LDA classification starts today
  - https://bidal.sfsu.edu/~kazokada/csc872/PD3.pdf

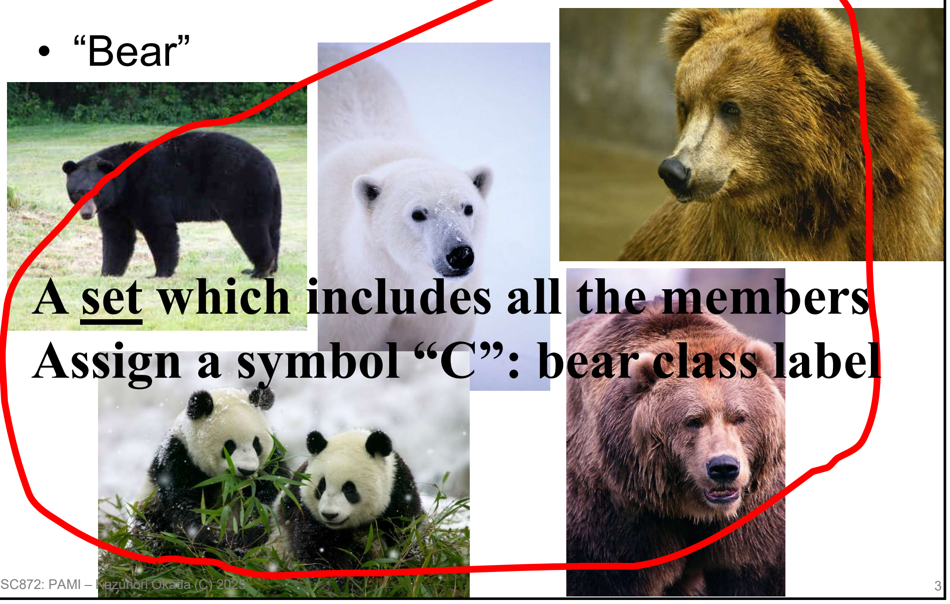
Classification & Learning (ML)

CSC 872

Pattern Analysis and Machine Intelligence

What is a Class?

- “Bear”



A <u>set</u> which includes all the members
Assign a symbol “C”: bear class label

Classification

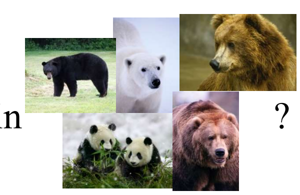
- Classification as an action: ***<u>Prediction</u>*** (test)

Is x =  a bear ? (Probability and/or ***Yes or No***

$x \in C$?

(training)

- Classification as learning: ***<u>Knowledge Extraction</u>***

What are common in  ?

A Learning Machine Interpretation

- Learning Machine: ***y* = *f*(*x*)** $X = (x_1, x_2)$
- **Input**: sample attributes features
  - E.g., x_1 = hairiness , x_2 = size
- **Output**: class membership
  - E.g., y = "+1" if bear, "-1" otherwise

member of C   not a member of C

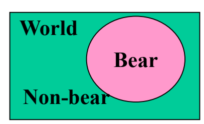
5

Types: Binary vs Multi-Labels

- **Binary** Classification
  - classify the world into *C*=bear or *C**=non-bear

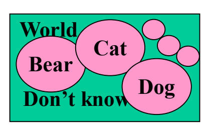
$K=2$

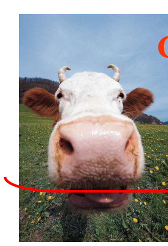
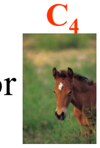
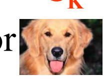
Is x =  a bear **or not** ?



- **Multi-label** Classification
  - Classify the world into *K* classes

$K>2$



Is x =   C_1 or  C_2 or  C_3 or  C_4 or  C_K ?

K

(K+1) or not

6

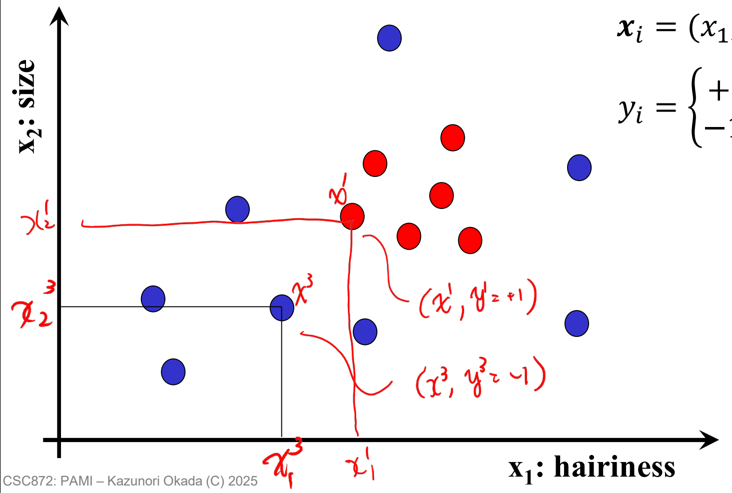
Training Data: X

- Supervised learning

$$X = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$$

$$\boldsymbol{x}_i = (x_1, x_2)^t = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

$$y_i = \begin{cases} +1 \;\; 1 & \text{if bear} \\ -1 \;\; 0 & \text{otherwise} \end{cases}$$



Training Data: X

Binary Classification
is $x \in C$?

- Supervised learning

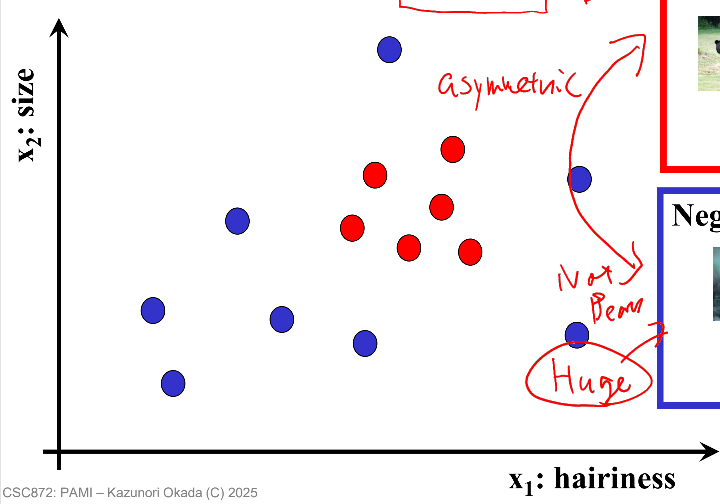
C

Bear

asymmetric

Not Bear

Huge



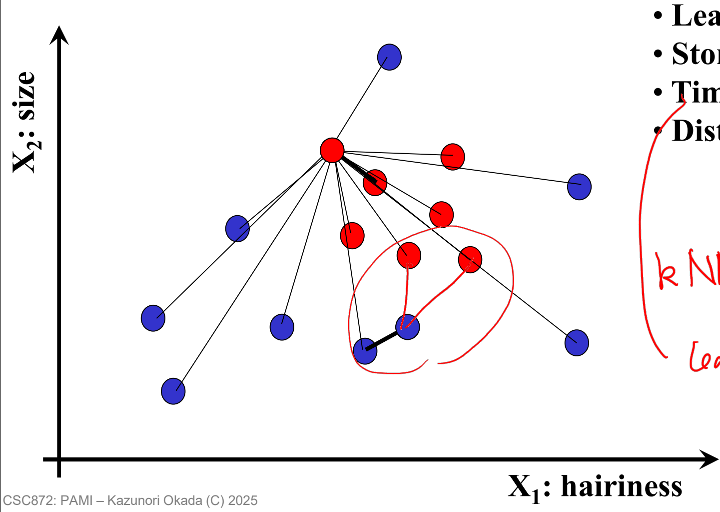
Pos cases

Positive samples

Negative samples

NEG cases

PF: Nearest-Neighbor Classifier



- Learning?
- Store all samples
- Time complexity?
- Distance function?

N=13

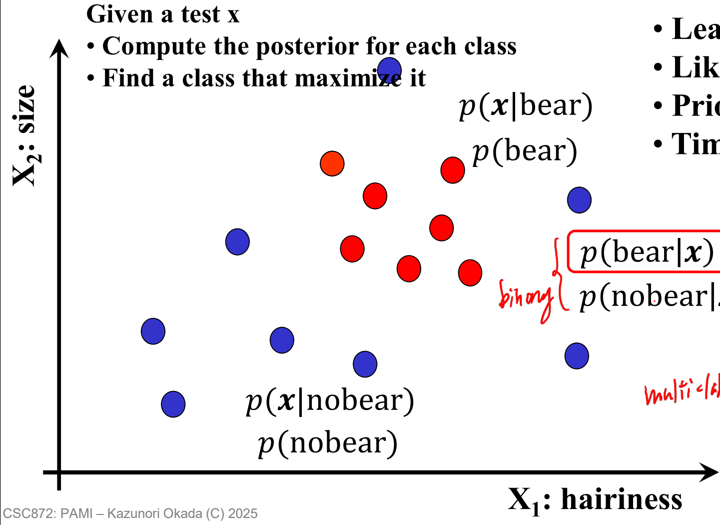
k NN classifier

Leading Factor is # of Training samples

PF: Probabilistic Classifier

- Bayesian MAP classifier: $y = \mathrm{argmax}_y p(y|\boldsymbol{x})$

Given a test x
- **Compute the posterior for each class**
- **Find a class that maximize it**



- **Learning?**
- **Likelihood function**
- **Prior distribution**
- **Time complexity?**

K=2

binary $\left\{ \begin{array}{l} \boxed{p(\text{bear}|\boldsymbol{x}) = 0.8} \\ p(\text{nobear}|\boldsymbol{x}) = 0.2 \end{array} \right.$

multi-class $p(C_1|x)$
$p(C_2|x)$
$\vdots$
$p(C_K|x)$

Probabilistic Classifiers

- Naïve Bayes Classifier
  - x_1 and x_2 are independent

$p(x_1, x_2) = p(x_1)\,p(x_2)$

- Gaussian Bayes Classifier
  - Data likelihood function takes a form of Gaussian

$$p(y|x_1, x_2) = \frac{p(y|x_1)p(y|x_2)}{p(y)}$$

- Advantages
  - **Rejection**: no learned classes fit the data
  - **Changing utility function**: different risk factors
  - **Compensating for imbalance class**: scaled likelihood trick
  - **Combining models**: sensor fusion
- Disadvantages
  - Difficult to specify priors
  - Inefficiency

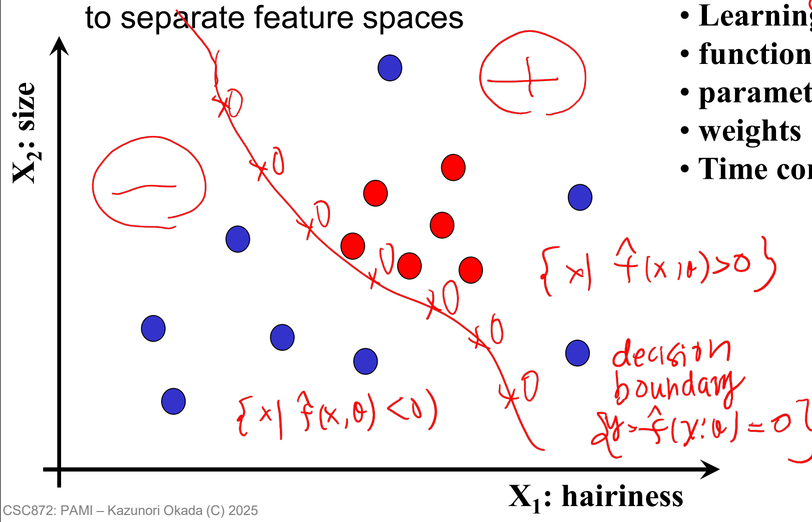
PF: Discriminative Classifiers

- Learning **Discriminant function**: $y = \hat{f}(\boldsymbol{x}; \boldsymbol{\theta})$ to separate feature spaces

is Parameterised Function
= learning machine
= agent function!

- Learning?
- function ?
- parameters
- weights
- Time complexity?

D=2



STOP: Linear Classifier

2D

3D Plane

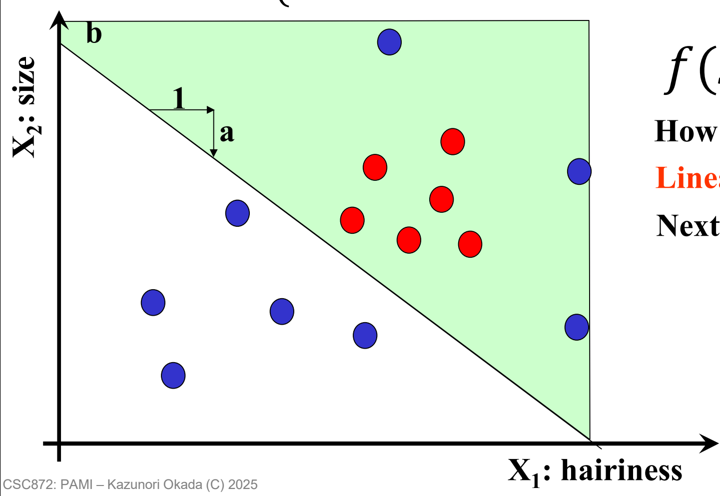
ND hyperplane

$x_2 = a x_1 + b$

$\Leftrightarrow x_2 - a x_1 + b = 0$

LDA!

$$f(\boldsymbol{x}) = \begin{cases} +1 & \text{if } x_2 - a x_1 - b \geq 0 \\ -1 & \text{otherwise} \end{cases}$$



$f(\boldsymbol{x}; a, b)$

How to learn this?

Linear Regression !!!

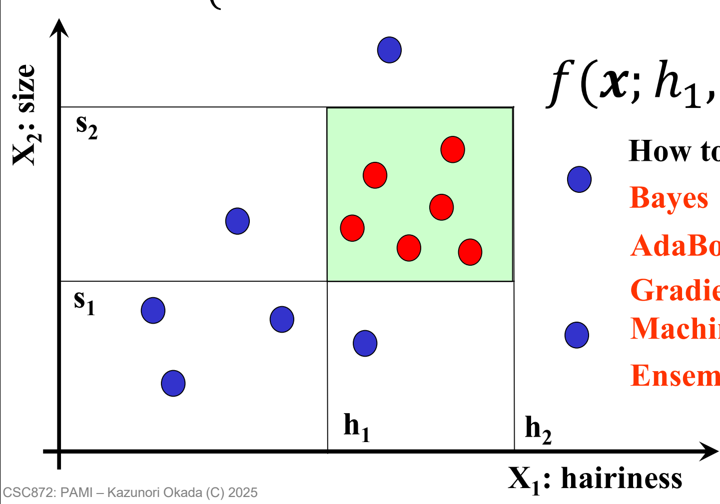
Next lecture !!!

13

Linear Classifier

Piecewise Linear

$$f(\boldsymbol{x}) = \begin{cases} +1 & \text{if } h_1 \leq x_1 \leq h_2 \text{ and } s_1 \leq x_2 \leq s_2 \\ -1 & \text{otherwise} \end{cases}$$



$f(\boldsymbol{x}; h_1, h_2, s_1, s_2)$

How to learn threshold?

Bayes

AdaBoost Boosting

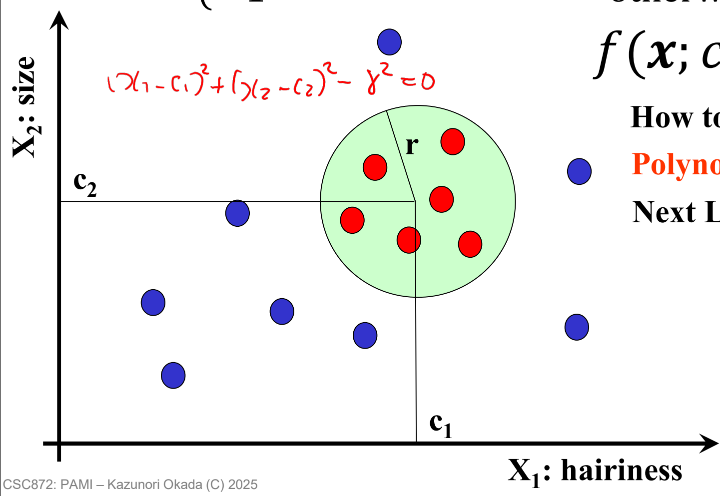
Gradient Boosting Machine

Ensemble Classifiers

14

Non-Linear Classifier Circular Classifier

$$f(\boldsymbol{x}) = \begin{cases} +1 & \text{if } (x_1 - c_1)^2 + (x_2 - c_2)^2 \le r^2 \\ -1 & \text{otherwise} \end{cases}$$



$$f(\boldsymbol{x}; c_1, c_2, r)$$

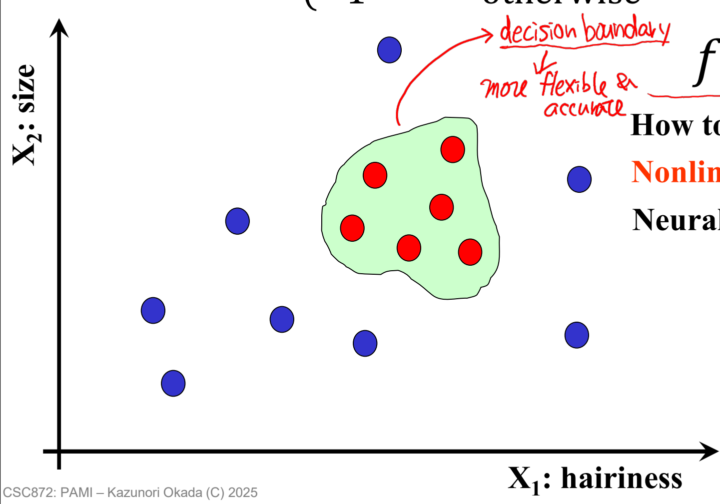
How to learn this?

Polynomial Regression

Next Lecture !!!

Non-Linear Classifier

$$f(\boldsymbol{x}) = \begin{cases} +1 & \text{if } \hat{f}(\boldsymbol{x}; \boldsymbol{w}) \ge 0 \\ -1 & \text{otherwise} \end{cases}$$



$$f(\boldsymbol{x}; \boldsymbol{w})$$ # of Params ↑

How to learn weights?

Nonlinear Regression

Neural Network

learning gets more difficult!

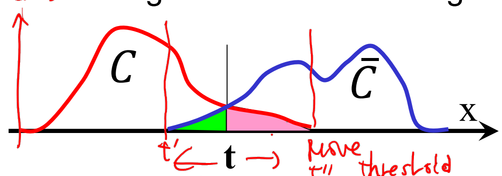
False Positive / False Negative

TP+FN=P
TN+FP=N

- More detailed way to interpret the Test Errors w/ histograms of Pos and Neg cases

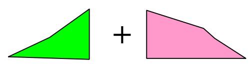
CNTs

A = #TP + #TN + #FP + #FN
= TP + TN + FP + FN



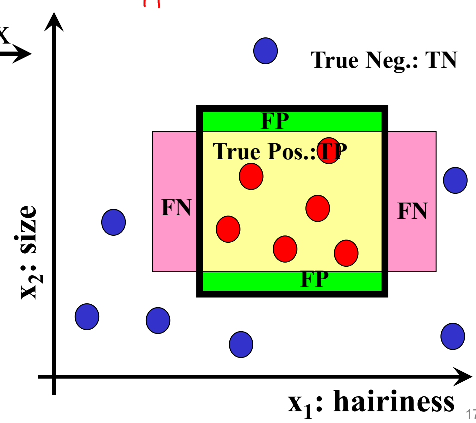
t' ← t → move t'' threshold

$$f(x) = \begin{cases} +1 & \text{if } x \le t \\ -1 & \text{otherwise} \end{cases}$$



False Positives (FP)

False Negatives (FN)

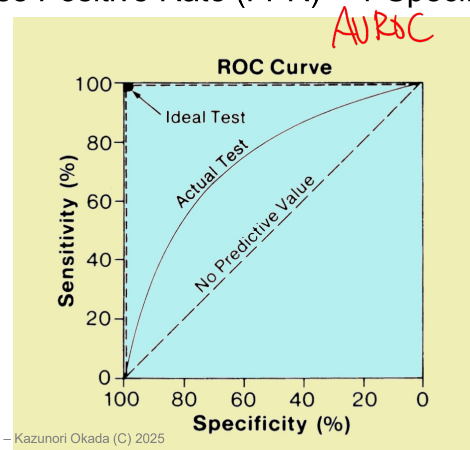


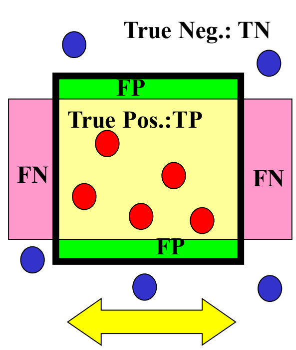
ROC Curve

Specificity = $\frac{TN}{N}$

- Receiver Operating Characteristics (ROC)
- True Positive Rate (TPR) = Sensitivity = TP/(TP+FN) = TP/P
- False Positive Rate (FPR) = 1-Specificity = FP/(TN+FP) = FP/N

AUROC



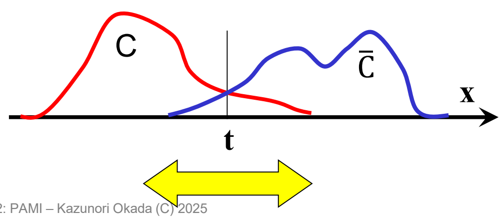


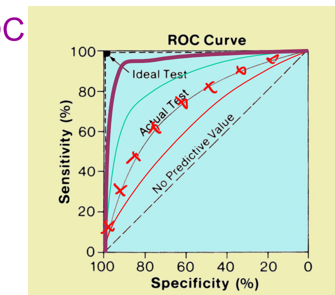
Model Selection with ROC

Given a training and test datasets (Tr, Te)

- Train a classifier with a certain threshold
- Categorizes test errors to TP,TN,FP,FN
- Iterate this with different threshold/weight values
- Produce a ROC curve
- Do the same for different algorithm
- Pick the algorithm that gave best ROC

AUROC → 1





Recall Eigen Face

- Linear Feature Extraction
  - A **linear transformation (projection)** from n-Dim input to m-Dim feature space

F

1000

$$10 \begin{pmatrix} y_1 \\ \cdot \\ \cdot \\ y_m \end{pmatrix} = \begin{pmatrix} f_{11} & \cdot & \cdot & \cdot & f_{1n} \\ \cdot & \cdot & \cdot & \cdot & \\ \cdot & \cdot & \cdot & \cdot & \\ f_{m1} & \cdot & \cdot & \cdot & f_{mn} \end{pmatrix} \begin{pmatrix} x_1 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{pmatrix}$$

$$y = Fx$$

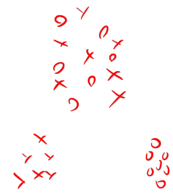
$$m < n$$

10 1000

- PCA: eigenvectors are rows of F: basis vectors!
- Reduce dimensionality in feature space to handle "curse of dimensionality"
- Remove redundant features

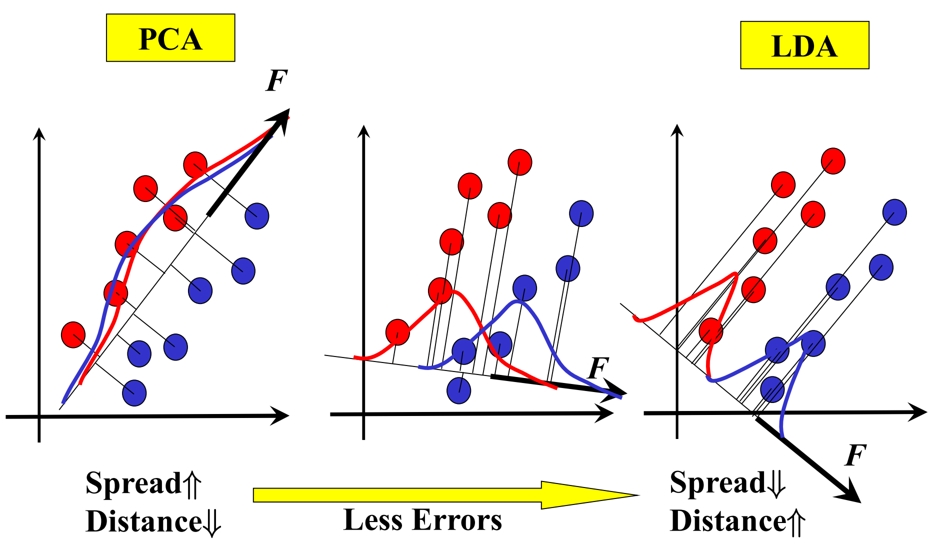
Linear Discriminant Analysis: LDA

- Suppose K classes in training data
- Find a linear basis set *F* to optimize the extracted feature for the purpose of classification



- How to do this supervised learning?
- Essentially
  - Minimize average sample spreads within classes
  - Maximize average distances between classes

2D Examples



Definitions

Binary Classification $K = 2$

- Suppose K classes

$\{\{c_1\}, \cdots, \{c_k\}\}$

$\{\{pos\}, \{neg\}\}$

$\boldsymbol{\mu_k}$: mean of k–th class

$\boldsymbol{\mu} = \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{\mu_k}$: mean of X

M_k: the number of sampels in k–th class

Within-Class Scatter Matrix

$$\boldsymbol{S_w} = \sum_{k=1}^{K}\sum_{i=1}^{M_k}(\boldsymbol{x}_i - \boldsymbol{\mu}_k)(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^t$$

Between-Class Scatter Matrix

$$\boldsymbol{S_b} = \sum_{k=1}^{K} M_k(\boldsymbol{\mu}_k - \boldsymbol{\mu})(\boldsymbol{\mu}_k - \boldsymbol{\mu})^t$$

How to Find F from X

- PF: Find F such that

$$\text{maximize } \frac{\boldsymbol{w}^t \boldsymbol{S_b} \boldsymbol{w}}{\boldsymbol{w}^t \boldsymbol{S_w} \boldsymbol{w}}$$

- Solution of a **generalized eigenvalue problem**

$$\boldsymbol{S_b}\boldsymbol{w}_m = \lambda_m \boldsymbol{S_w}\boldsymbol{w}_m$$
$$F = (\boldsymbol{w}_1, .., \boldsymbol{w}_M)^t$$
$$M = K - 1$$

eig

- Solution of a corresponding **eigenvalue problem**

$$(\boldsymbol{S_w^{-1}}\boldsymbol{S_b})\boldsymbol{w}_m = \lambda_m \boldsymbol{w}_m$$

Are we done?

- Solve **eig(inv(S_w)$\times$$S_b$)** ?
- What if $\mathbf{S_w}$ is singular so **inv(S_w)** cannot be computed?
- This happens a lot! (#data < Dimensionality)

- Soln1:
  - PCA+LDA, $\boldsymbol{y = F_{LDA}F_{PCA}x}$
- Soln2:
  - Regularize it: $C' = C + \sigma I$ (find σ by cross validation)
  - **Make C a full rank so that it can be inverted**

- Once you find F, then transform X by F then **use your choice of other classifier (e.g., Bayes Classifier)**

Summary

- Classification & Learning
  - What is a Class?
  - Nearest Neighbor Classifier
  - Probabilistic Classifier
  - Discriminant Classifier
  - FP/FN/ROC
  - Linear Discriminant Analysis
- Next
  - Learning & Regression
  - Linear Regression
  - Least Squares Estimation & Maximum Likelihood Regression
  - Non-linear Regression
  - Optimization Techniques