Supervised learning II

M. Vazirgiannis

JANUARY 2015

Outline

- **Logistic Regression**
- Perceptron
- Linear Discriminant Analysis (LDA)

How to tackle e learning problem

Given data, a real-world classification problem, and constraints, you need to determine:

- classifier to use
- optimization method to employ
- loss function to minimize
- features to take from the data
- evaluation metric to use

Binary classification

Assume

- users have visited pages and some of them visited advertisement *ad_1* (the target variable)

- *User U_5 = < 1,....1>*

Objective: predict if the user (based on her past visits) will click the ad or not

User_id	Seen_p1	..	Seen_pn	VISITED ad1
1	1	..	1	0
2	0	..	0	1
3	1	..	1	1
4	1	..	0	0
5	1	..	1	?

Logistic regression

- Fits binary classification problems
- Output in [0,1]
- Need a function that takes data vector and produces as output 0/1

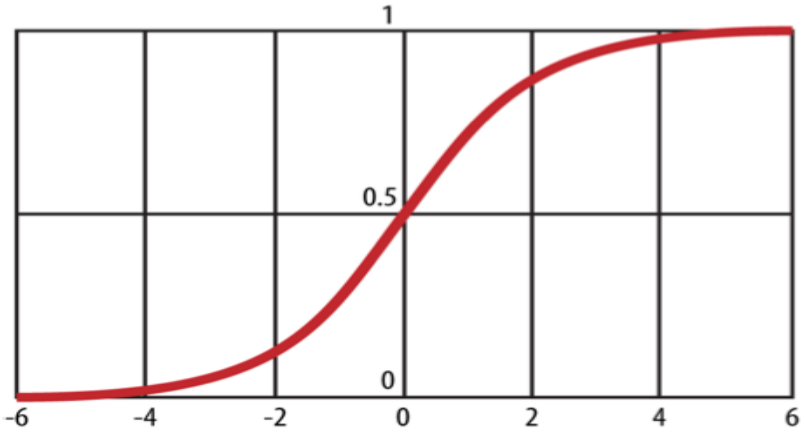
Logit function

The *logit* function takes as input values p in [0,1] and transform them to real value, i.e:

$$logit(p) = log(\frac{p}{1-p}) = log(p) - log(1-p)$$

The *inverse logit* function takes as input values *along the real line* and transforms them in the range [0,1], i.e.:

$$P(t) = logit^{-1}(t) = \frac{1}{1+e^{-t}} = \frac{e^t}{1+e^t}$$



Modeling Prediction of a click

- Assume class c_i (1 if clicked, 0 otherwise)
- x_i the input data for user i

$$P(c_i|x_i) = logit^{-1}(\alpha + \beta^\tau x_i)^{c_i} \star (1 - logit^{-1}(\alpha + \beta^\tau x_i))^{1-c_i}$$

- As c is either 0 or 1 :
- ci=1: $P(c_i = 1|x_i) = \frac{1}{1+e^{-(\alpha+\beta^\tau x_i)}} = logit^{-1}(\alpha + \beta^\tau x_i)$

- ci=0: $P(c_i = 0|x_i) = 1 - logit^{-1}(\alpha + \beta^\tau x_i)$

Logistic regression model.

logit(P(<user *i* clicks on the ad>) = linear function of the features (URLs that user *i* visited).

$$logit(P(c_i|x_i)) = (\alpha + \beta^\tau x_i)$$

Logistic regression model.

- make this a linear model for ci
- take the log of the *odds ratio*:

$$log(P(c_i = 1|x_i)/(1 - P(c_i = 1|x_i))) = \alpha + \beta^\tau x_i$$

- Or: $log(P(c_i = 1|x_i)) = \alpha + \beta^\tau x_i$

- α: base rate, unconditional probability of c=1 (click)

- β: $P(c_i = 1) = \frac{1}{1+e^{-a}}$
  - slope of the logit function
  - Vectors with dimension equal to the one of the feature vector
  - Determines the relevance of features (i.e. pages visited) to the likelihood of the ad being clicked

Estimating α, β

- Parameters: $\theta = \{a, b\}$
- Likelihood:

$$L(\theta|X_1, X_2, ...X_n) = P(X|\theta) = P(X_1|\theta)...P(X_n|\theta)$$

- Xi users, are independent…

$$\theta_{MLE} = argmax_\theta \prod_{i=1}^{n} P(X_i|\theta)$$

- Setting $p_i = \frac{1}{1+e^{-(\alpha+\beta^t x_i)}}$

$$\theta_{MLE} = argmax_\theta \prod_{i=1}^{n} p^{c_i}(1-p_i)^{1-c_i}$$

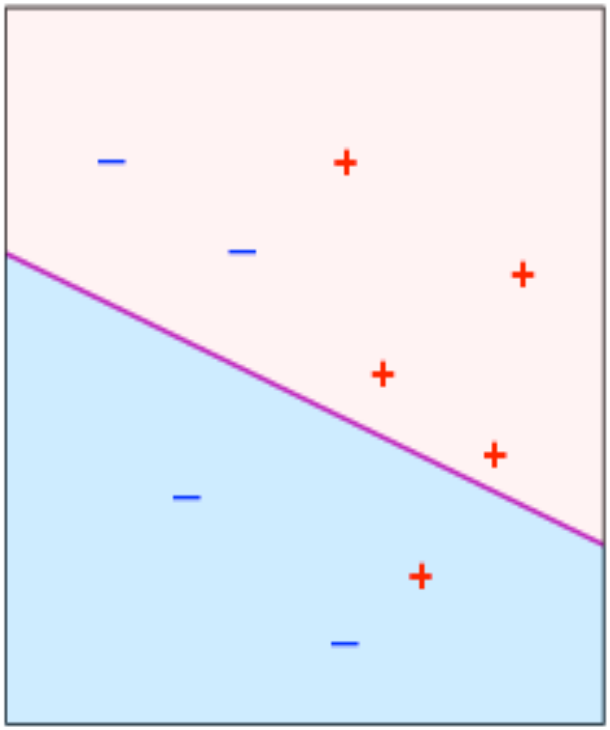
Maximizing likelihood

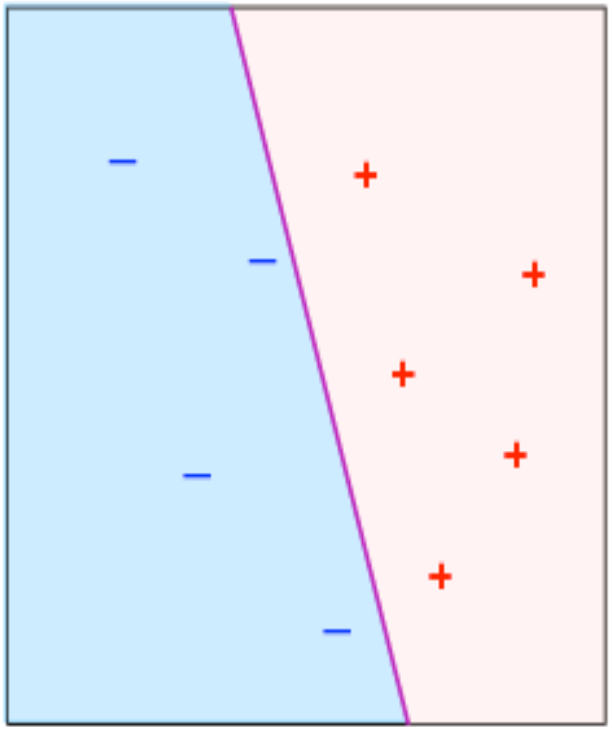
- Newton method
- Stochastic gradient descent
  - Update parameters for each point visited.

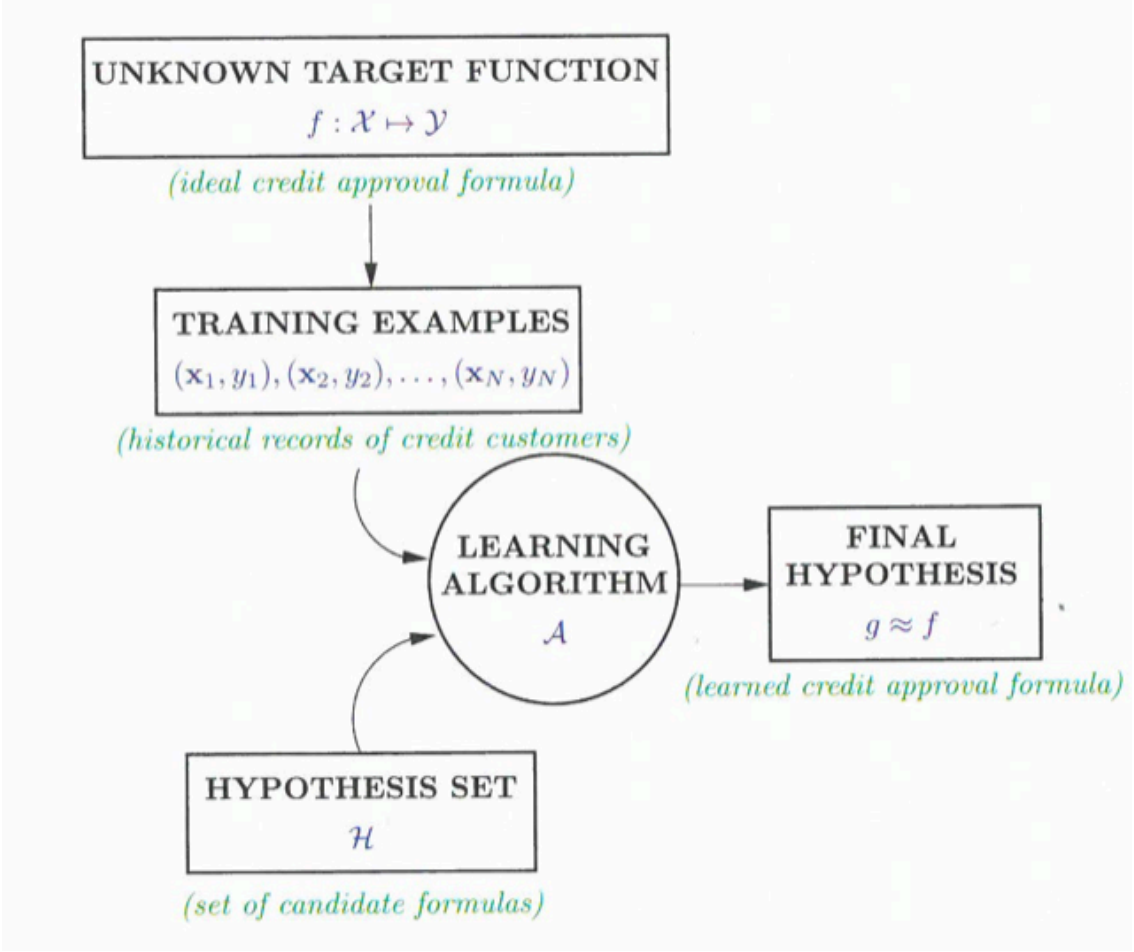
Outline

- Logistic Regression
- **Perceptron**
- Linear Discriminant Analysis (LDA)

Simple learning problem







From “Learning from Data”,Y. Mostafa et al. 2012

Components of learning

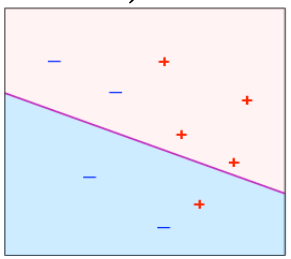
- Assume the credit approval problem
- Let
  - input x the customer information
  - an unknown target function $f: X->Y$, Y is yes/no for approval
  - Let $D=\{(x1, y1) \ldots (x1, y1)\}$ a training set of points
  - Learning algorithm uses D to pick a formula $g: X->Y$ approximating f
  - g is chosen from a set of functions H (Hypotheses set) and the objective is to learn its parameters values (i.e. polynomial coefficients) to fit f.

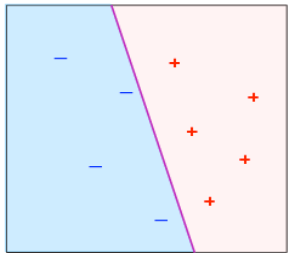
Simple learning problem

- Assume the credit scoring case:
- Let $X = (x_1, ...x_d)$ user data
- $h(X) = \sum_i^d w_i x_i$ the evaluation
- approve credit if: $\sum_i^d w_i x_i > thershold$
- disapprove credit if: $\sum_i^d w_i x_i < thershold$

There fore: $h(x) = sign((\sum_i^d w_i x_i) + b)$

Perceptron





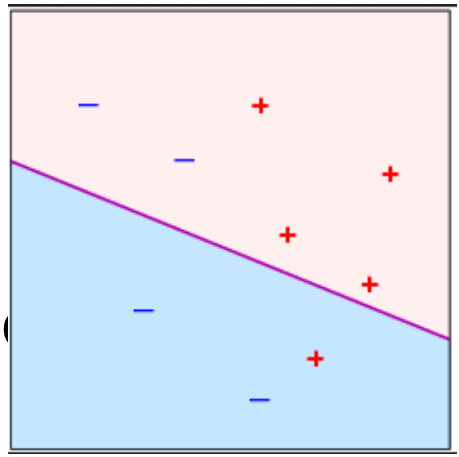
Perceptron learning algorithm

$$h(x) = sign((w^T x)$$

- Assume data set is linearly separable
- Algorithm
  - starts with values w_0
  - Take (x_t, y_t) a miss-classified example fr
    $(x_1, y_1), (x_2, y_2)...(x_n, y_n)$
  - Since $sign(w^t x_t) = -1$ the update rule is:

    $$w^{t+1} = w^t + y_t x_t$$

  - Moves the boundary in the direction of classifying x_t correctly



Perceptron learning algorithm

- Continue with further iterations with other misclassified points (mp)
- Algorithms converges to the solution *h* regardless of
  - the order of mps consideration
  - The initial w_0 vector

Proof of convergence at:

http://www.cs.columbia.edu/~mcollins/courses/6998-2012/notes/perc.converge.pdf

Outline

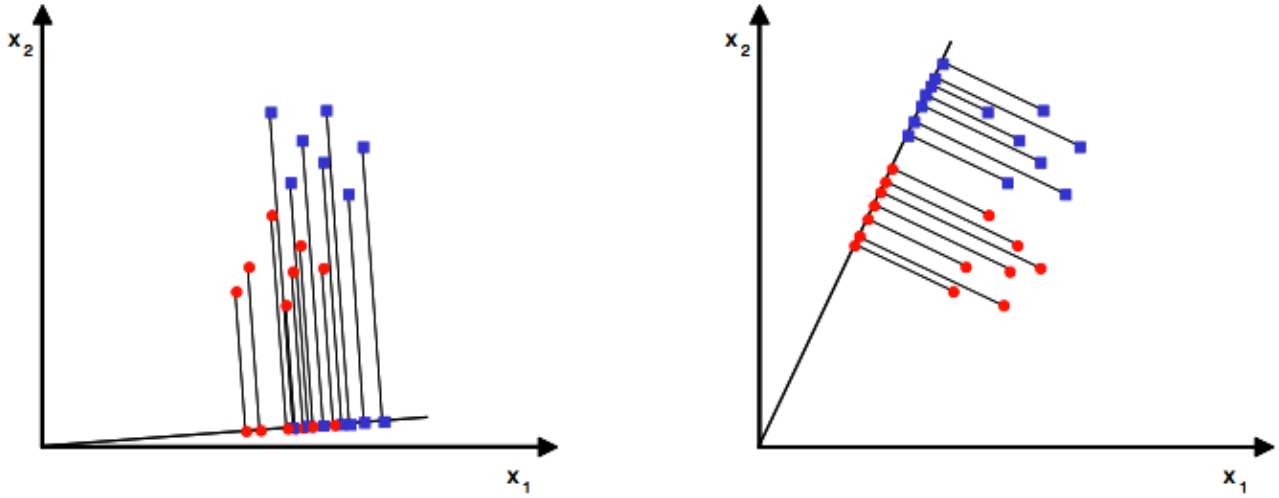
- Logistic Regression
- Perceptron
- **Linear Discriminant Analysis (LDA)**

Linear discriminants analysis

- Linear discriminant analysis, two classes
- Linear discriminant analysis, C classes

LDA – two classes

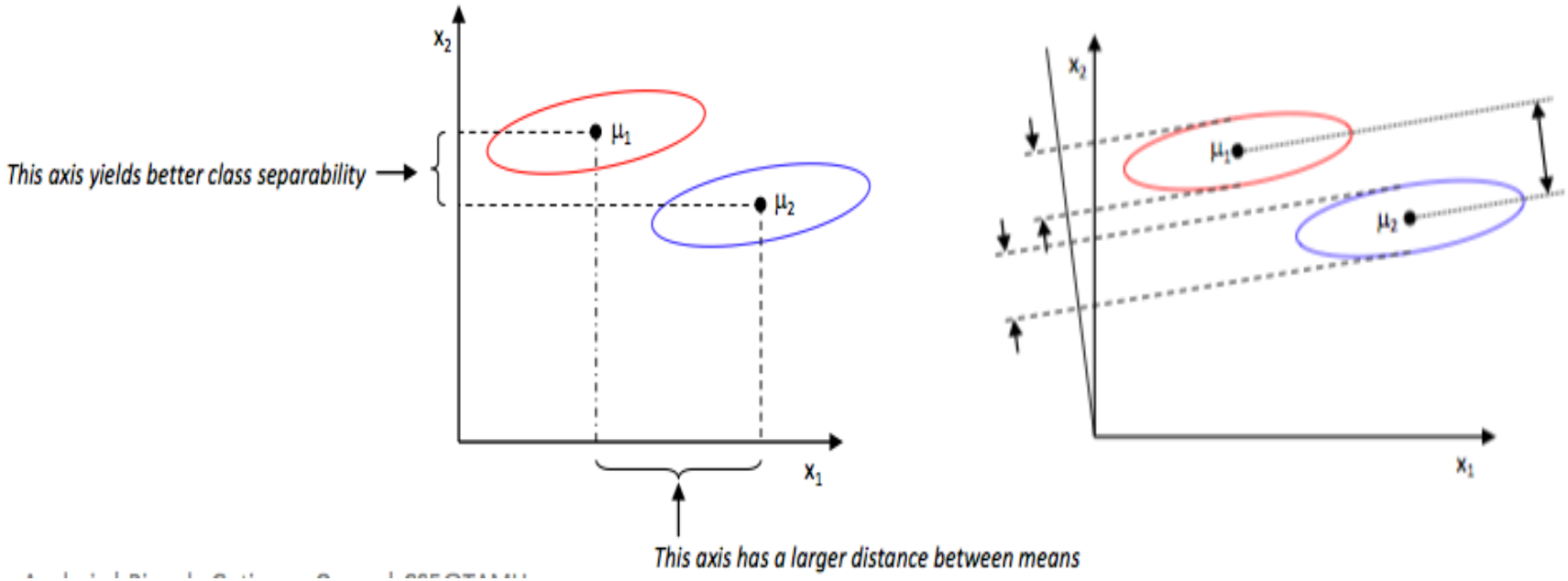
- LDA aims to project data in a lower dimensional space that preserves as much of the class discriminatory information as possible
- Assume $X = (x_1, x_2, \ldots x_n)$ d-dimensional that belong to either of the classes C1 and C2.
- We search for $y = w^T X$ that best separates the data of C1 and C2.



LDA – two classes

The projection should

- maximize the distance of the class centers
- minimize the in class variance



LDA two classes – Fischer's criterion

- *Distance* between projected classes centers $\hat{\mu_1}\hat{\mu_2}$:

$$|\hat{\mu_1} - \hat{\mu_2}| = \frac{1}{|C_1|}\sum_{C_1} y_i^{C_1} - \frac{1}{|C_2|}\sum_{C_2} y_i^{C_2} = \frac{1}{|C_1|}\sum_{C_1} w^T x_i^{C_1} - \frac{1}{|C_2|}\sum_{C_2} w^T x_i^{C_2}$$

- Within class variance $\hat{S}_i = \sum_{y \in C_i} (y - \hat{\mu}_i)^2$
- Total *within class scatter* for the projected data

$$\hat{S}_1 + \hat{S}_2$$

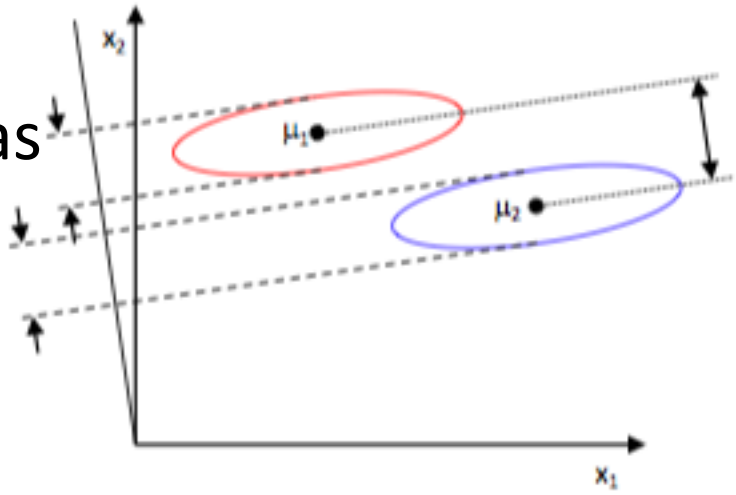
Fischer's linear discriminant

linear function of $y = w^T X$

maximizing: $J(w) = \frac{|\hat{\mu_1} - \hat{\mu_2}|^2}{\hat{S}_1 + \hat{S}_2}$

Therefore, searching for a projection where same class points are

- projected very close to each
- projected means are as farther as possible



Optimization – within class scatter

Within class scatter matrix in feature space x

$$S_w = S_1 + S_2$$

Where $S_i = \sum_{x \in C_i} (x - \mu_i)(x - \mu_i)^T$

Within class scatter matrix in projected space y

$$\hat{S}_1^{\,2} + \hat{S}_2^{\,2} = w^T S_w w$$

Where $\hat{S}_i = w^T S_i w$

between class scatter in projected space y

$$|\hat{\mu_1} - \hat{\mu_2}|^2 = w^T(\mu_1 - \mu_2)(\mu_1 - \mu_2)^T w = w^T S_B w$$

Fishers Criterion optimization

$$J(W) = \frac{w^T S_B w}{w^T S_W w}$$

To find maximum

$$\frac{d}{dw} J(W) = 0 \implies S_w^{-1} S_B w - Jw = 0$$

Solving the generalized eigenvalue problem:

$$S_w^{-1} S_B w = Jw$$

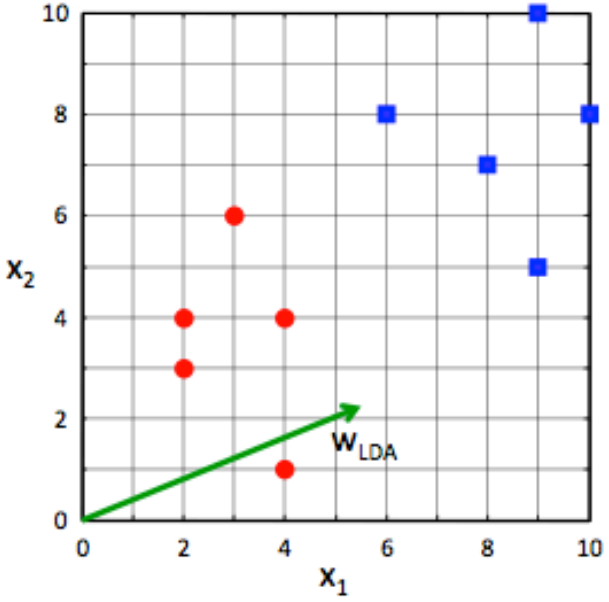
We find:

$$w^* = argmax[\frac{w^T S_B w}{w^T S_W w}] = S_w^{-1}(\mu_1 - \mu_2)$$

An example[1]

Compute the LDA projection for the following 2D dataset

$$X1 = \{(4,1), (2,4), (2,3), (3,6), (4,4)\}$$
$$X2 = \{(9,10), (6,8), (9,5), (8,7), (10,8)\}$$



SOLUTION (by hand)

- The class statistics are

$$S_1 = \begin{bmatrix} .8 & -.4 \\ & 2.64 \end{bmatrix} \quad S_2 = \begin{bmatrix} 1.84 & -.04 \\ & 2.64 \end{bmatrix}$$

$$\mu_1 = [3.0\ 3.6]^T; \quad \mu_2 = [8.4\ 7.6]^T$$

- The within- and between-class scatter are

$$S_B = \begin{bmatrix} 29.16 & 21.6 \\ & 16.0 \end{bmatrix} \quad S_W = \begin{bmatrix} 2.64 & -.44 \\ & 5.28 \end{bmatrix}$$

- The LDA projection is then obtained as the solution of the generalized eigenvalue problem

$$S_W^{-1} S_B v = \lambda v \Rightarrow \left| S_W^{-1} S_B - \lambda I \right| = 0 \Rightarrow \begin{vmatrix} 11.89 - \lambda & 8.81 \\ 5.08 & 3.76 - \lambda \end{vmatrix} = 0 \Rightarrow \lambda = 15.65$$

$$\begin{bmatrix} 11.89 & 8.81 \\ 5.08 & 3.76 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = 15.65 \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \Rightarrow \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} .91 \\ .39 \end{bmatrix}$$

- Or directly by

$$w^* = S_W^{-1}(\mu_1 - \mu_2) = [-.91\ -.39]^T$$

LDA C classes [1]

Fisher's LDA generalizes gracefully for C-class problems

- Instead of one projection y, we will now seek $(C-1)$ projections $[y_1, y_2, \dots y_{C-1}]$ by means of $(C-1)$ projection vectors w_i arranged by columns into a projection matrix $W = [w_1|w_2|\dots|w_{C-1}]$:

$$y_i = w_i^T x \Rightarrow y = W^T x$$

Derivation

- The within-class scatter generalizes as

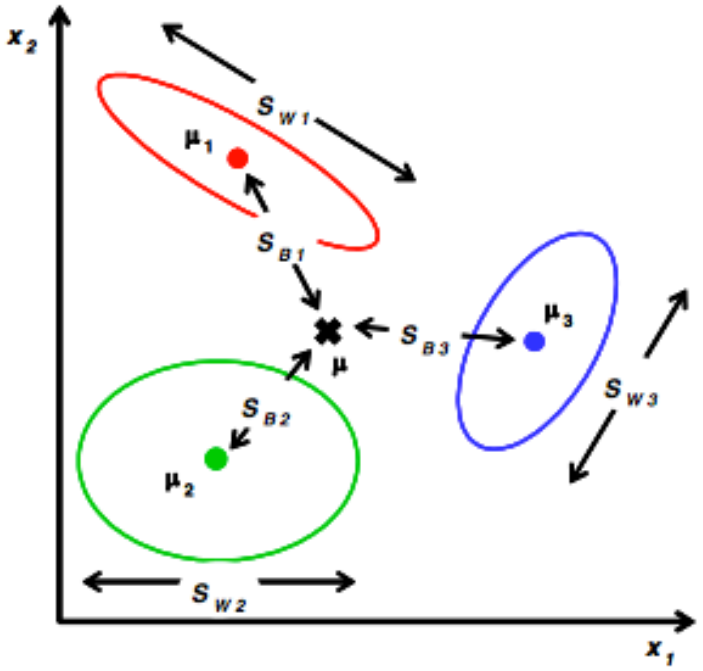
$$S_W = \sum_{i=1}^{C} S_i$$

  - where $S_i = \sum_{x \in \omega_i} (x - \mu_i)(x - \mu_i)^T$ and $\mu_i = \frac{1}{N_i} \sum_{x \in \omega_i} x$

- And the between-class scatter becomes

$$S_B = \sum_{i=1}^{C} N_i (\mu_i - \mu)(\mu_i - \mu)^T$$

  - where $\mu = \frac{1}{N} \sum_{\forall x} x = \frac{1}{N} \sum_{i=1}^{C} N_i \mu_i$



- Matrix $S_T = S_B + S_W$ is called the <u>total scatter</u>

LDA C classes

- mean vector and scatter matrices for the projected samples

$$\tilde{\mu}_i = \frac{1}{N_i}\sum_{y\in\omega_i} y \qquad \tilde{S}_W = \sum_{i=1}^{C}\sum_{y\in\omega_i}(y-\tilde{\mu}_i)(y-\tilde{\mu}_i)^T$$

$$\tilde{\mu} = \frac{1}{N}\sum_{\forall y} y \qquad \tilde{S}_B = \sum_{i=1}^{C} N_i(\tilde{\mu}_i-\tilde{\mu})(\tilde{\mu}_i-\tilde{\mu})^T$$

- Thus generalizing the two class problem: $\tilde{S}_W = W^T S_W W$, $\tilde{S}_B = W^T S_B W$
- Searching for a projection maximizing the ratio of *between-class/within-class scatter*.
- projection is no longer a scalar ($C-1$ dimensions), use the determinant of the scatter matrices to obtain a scalar objective function

$$J(W) = \frac{|\tilde{S}_B|}{|\tilde{S}_W|} = \frac{|W^T S_B W|}{|W^T S_W W|}$$

LDA C classes

- seek the projection matrix $W*$ maximizing this ratio
- optimal projection matrix $W*$ is the one whose columns are the eigenvectors corresponding to the largest eigenvalues of the following generalized eigenvalue problem

$$W^* = [w_1^* | w_2^* | \dots w_{C-1}^*] = \arg\max \frac{|W^T S_B W|}{|W^T S_W W|} \Rightarrow (S_B - \lambda_i S_W) w_i^* = 0$$

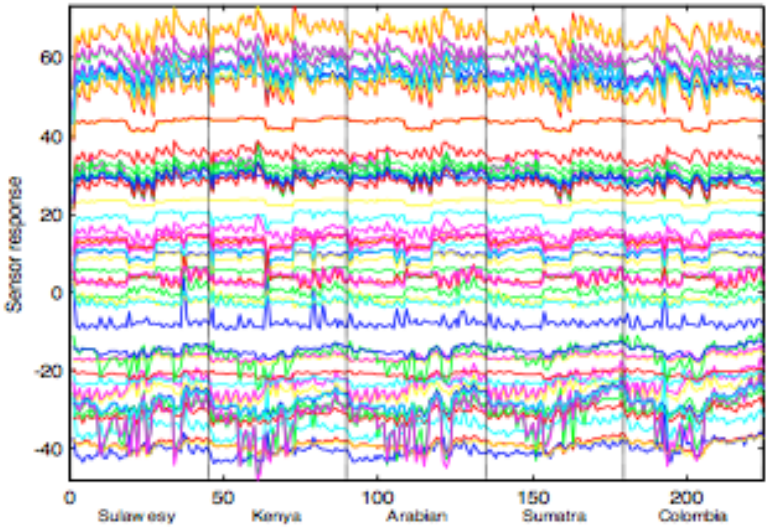
LDA vs. PCA

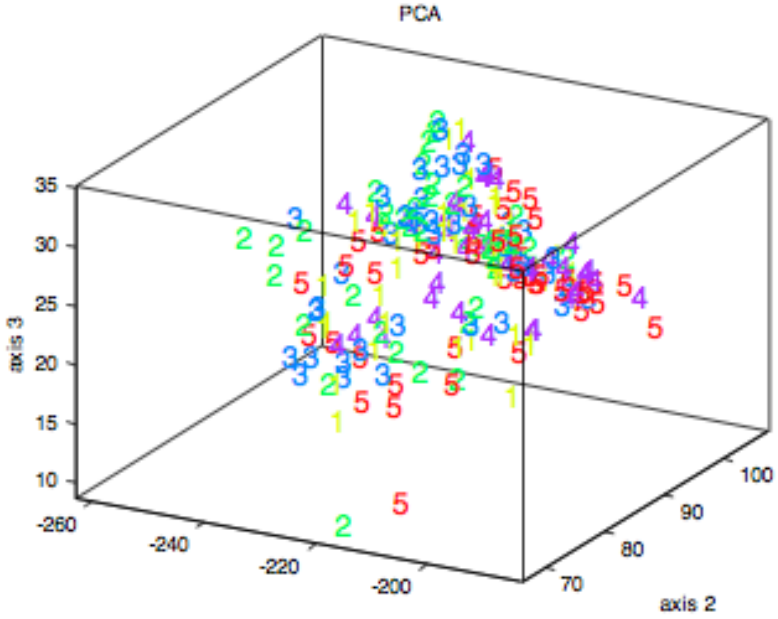
This example illustrates the performance of PCA and LDA on an odor recognition problem

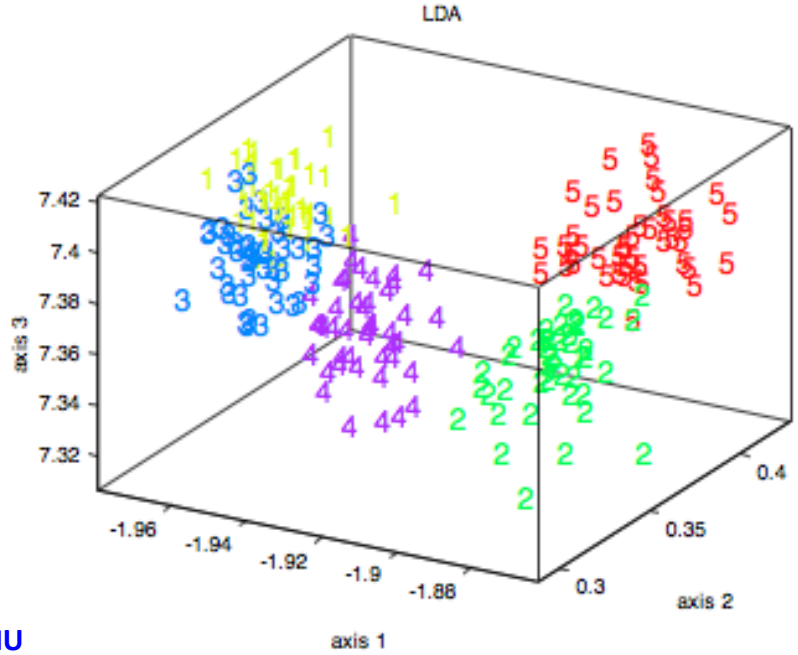
- Five types of coffee beans were presented to an array of gas sensors
- For each coffee type, 45 "sniffs" were performed and the response of the gas sensor array was processed in order to obtain a 60-dimensional feature vector

Results

- From the 3D scatter plots it is clear that LDA outperforms PCA in terms of class discrimination
- This is one example where the discriminatory information is not aligned with the direction of maximum variance







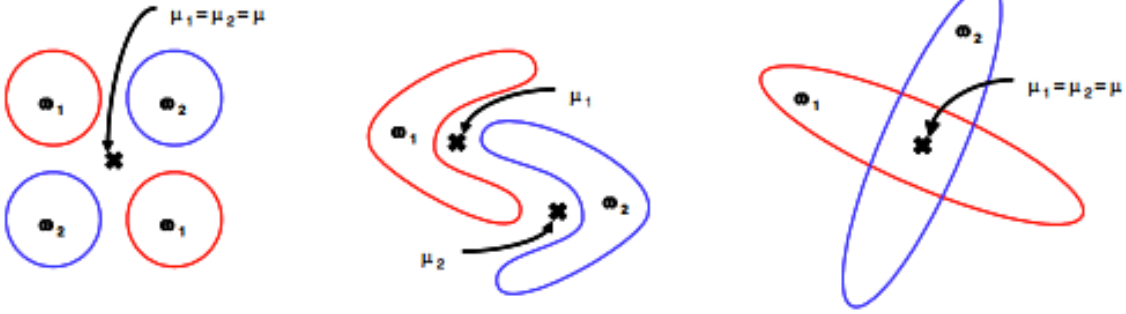
Limitations of LDA

LDA produces at most $C - 1$ feature projections

- If the classification error estimates establish that more features are needed, some other method must be employed to provide those additional features

LDA is a parametric method (it assumes unimodal Gaussian likelihoods)

- If the distributions are significantly non-Gaussian, the LDA projections may not preserve complex structure in the data needed for classification



LDA will also fail if discriminatory information is not in the mean but in the variance of the data