PA196: Pattern Recognition

3. Linear discriminants

Vlad Popovici
popovici@iba.muni.cz

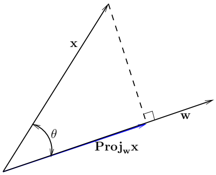
Institute of Biostatistics and Analyses
Masaryk University, Brno

Outline

Reminder - scalar product

- scalar (dot, inner) product of two vectors: $\mathbf{x}, \mathbf{w} \in \mathbb{R}^d : \mathbf{w} \cdot \mathbf{x} = \langle \mathbf{w}, \mathbf{x} \rangle = \mathbf{w}^t \mathbf{x} = \sum_{i=1}^{d} w_i x_i \in \mathbb{R}$
- $\cos \theta = \frac{\langle \mathbf{w}, \mathbf{x} \rangle}{||\mathbf{w}|| \, ||\mathbf{x}||}$
- $\langle \mathbf{w}, \mathbf{x} \rangle = 0 \iff \mathbf{w} \perp \mathbf{x}$
- projection of $\mathbf{x}$ on $\mathbf{w}$ is

$$\text{Proj}_{\mathbf{w}} \mathbf{x} = \frac{\langle \mathbf{x}, \mathbf{w} \rangle}{||\mathbf{w}||} \frac{\mathbf{w}}{||\mathbf{w}||} = \frac{\langle \mathbf{x}, \mathbf{w} \rangle}{||\mathbf{w}||^2} \mathbf{w}$$



Outline

General problem

- we consider the binary classification problem ($K = 2$)
- without loss of generality, we let the labels of the classes be ± 1
- we are given a set $\mathcal{X} \times \mathcal{Y} = \{(\mathbf{x}_i, y_i) | i = 1, \dots, n\} \subset \mathbb{R}^d \times \{-1, +1\}$
- the goal is to find the parameters of the classifier such that the number of misclassified points is minimized
- let the discriminant function have the form

$$h(\mathbf{x}) = \mathbf{w}^t \mathbf{x} + w_0 = \langle \mathbf{w}, \mathbf{x} \rangle + w_0 = w_0 + \sum_{i=1}^{d} w_i x_i$$

- note that $\mathbf{x}$ can be replaced with $\phi(\mathbf{x})$! (we'll discuss this later)
- the classifier is

$$\text{sign}(h(\mathbf{x})) = \text{sign}(\langle \mathbf{w}, \mathbf{x} \rangle + w_0)$$

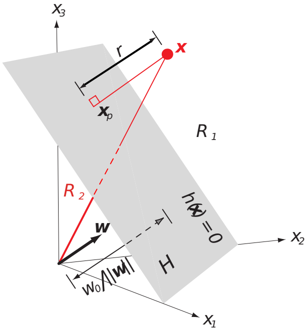
- an error: if $\text{sign}(\langle \mathbf{w}, \mathbf{x}_i \rangle + w_0) \neq y_i$; in other words: if $y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + w_0) < 0 \Leftrightarrow y_i h(\mathbf{x}_i) < 0$
- the *risk of misclassification (error)* is

$$R(h) = \Pr[Y \neq \text{sign}(h(X))]$$

  where (X, Y) is a random pair of observations
- the *empirical risk* is the estimation of the risk on a given set of points:

$$\hat{R}_n(h) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}_{\{y_i \neq \text{sign}(h(\mathbf{x}_i))\}} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}_{y_i h(\mathbf{x}_i) < 0}$$

- you need $n \geq d + 1$ points for learning the classifier



The linear decision boundary H, where $h(\mathbf{x}) = \mathbf{w}^t\mathbf{x} + w_0 = 0$, separates the feature space into two half-spaces R_1 (where $h(\mathbf{x}) > 0$) and R_2 (where $h(\mathbf{x}) < 0$). From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Outline

Margins

Functional Margin

The *functional margin* of a point $\mathbf{x}_i$ with respect to a hyperplane $\mathbf{w}$ is defined to be

$$\gamma_i = y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + w_0) = y_i h(\mathbf{x}_i)$$

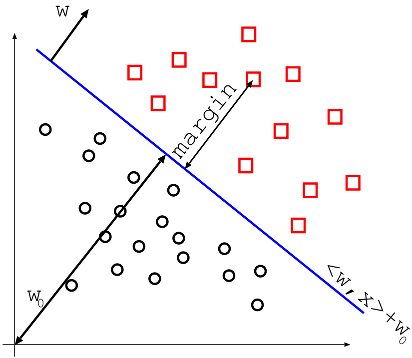
Geometric Margin

The *geometric margin* of a point $\mathbf{x}_i$ with respect to a hyperplane $\mathbf{w}$ is defined to be

$$\gamma_i = y_i \left(\left\langle \frac{\mathbf{w}}{||\mathbf{w}||}, \mathbf{x}_i \right\rangle + \frac{w_0}{||\mathbf{w}||} \right) = y_i \frac{h(\mathbf{x}_i)}{||\mathbf{w}||}$$

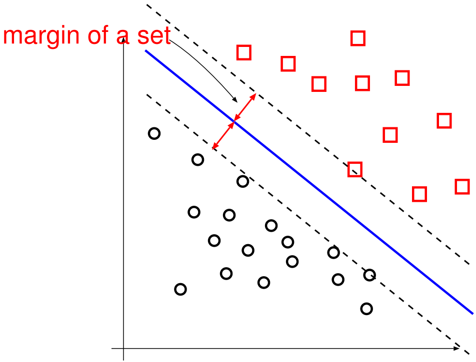
→ Geometric margin is the normalized functional margin.

Margin of a point



Margin of a set (of points)

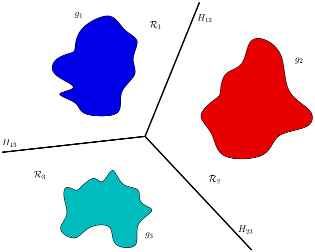
The maximum margin among all (hyper)planes is the margin of a set of points. The corresponding hyperplane is called maximum margin hyperplane.



Outline

Generalization to multi-class problems

- a multi-class problem can be decomposed in a series of two-class problems: 1-vs-all or 1-vs-1
- or, one can use K (no. of classes) discriminant fn. $h_i(\mathbf{x})$ and build classifiers of the form: assign $\mathbf{x}$ to class i if $h_i(\mathbf{x}) > h_j(\mathbf{x})$ for all $i \neq j$
- this defines $K(K-1)/2$ hyperplanes $H_{ij} : h_i(\mathbf{x}) - h_j(\mathbf{x}) = 0$
- in practice, there are usually less hyperplanes that form the decision surface

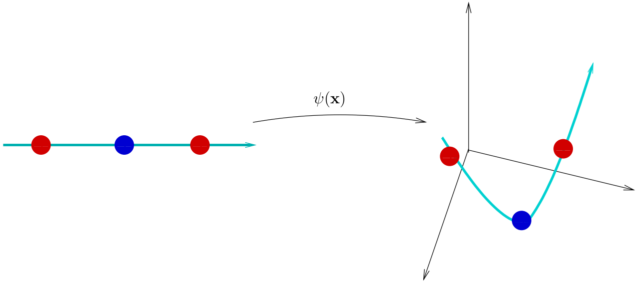


Generalized linear discriminants

Consider a function $\psi : \mathbb{R}^d \to \mathbb{R}^{\hat{d}}$. The discriminant function

$$g(\mathbf{x}) = \langle \mathbf{a}, \psi(\mathbf{x}) \rangle = \sum_{i=1}^{\hat{d}} a_i \psi_i(\mathbf{x})$$

is a linear function in **a** (but not in **x**).
Example: let $\mathbf{x} = x \in \mathbb{R}$ and let $\psi(x) = [1, x, x^2]^t \in \mathbb{R}^3$.



Remarks:

- a problem which is not linearly separable in $\mathbb{R}^d$ may become linearly separable in $\mathbb{R}^{\hat{d}}$
- $\psi = ?$
- finding the coefficients in $\mathbb{R}^{\hat{d}}$ requires much more training points!
- the decision surface, when projected back into $\mathbb{R}^d$ (by ψ^{-1}) is non-linear

- a convenient (but trivial) transformation: "normalization" of the notation
- take $\psi(\mathbf{x}) = y[1, \mathbf{x}]^t$. This allows us to write

$$\gamma = yh(\mathbf{x}) = y(\langle \mathbf{w}, \mathbf{x} \rangle + w_0) = \langle \mathbf{a}, \mathbf{z} \rangle$$

  where $\mathbf{a} = [w_0, \mathbf{w}]^t$ and $\mathbf{z} = y[1, \mathbf{x}]^t$
- the problem becomes: find $\mathbf{a}$ such that

$$\langle \mathbf{a}, \mathbf{z} \rangle > 0$$

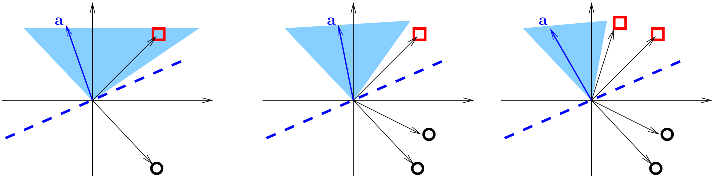
  i.e. all the margins are positive
- the decision surface $\hat{H}$ in $\mathbb{R}^{d+1}$, defined by $\langle \mathbf{a}, \mathbf{z} \rangle = 0$, corresponds to a hyperplane passing through the origin of the $\mathbf{z}$–space

Outline

- consider we are given the set $\{(\mathbf{x}_i, y_i)\}$ with $y_i = \pm 1$
- with the previous "normalized" notation, the set is linearly separable if

$$\langle \mathbf{a}, \mathbf{z}_i \rangle > 0, \quad \forall i = 1, \ldots, n$$

- the solution $\mathbf{a}$ is constrained by each point $\mathbf{z}_i$



- under current conditions, the solution is not unique!
- solutions on the boundary of the solution space may be too sensitive → you can use the condition $\langle \mathbf{a}, \mathbf{z}_i \rangle \geq \xi > 0$

Outline

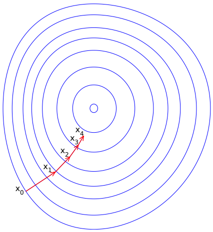
General approach

- let $J(\mathbf{a})$ be a criterion function that measures the "suitability" of a candidate solution $\mathbf{a}$
- by convention, the solution to the classification problem is obtained as
$$\mathbf{a}^* = \arg\min_{\mathbf{a}} J(\mathbf{a})$$
- usually, J is chosen to be continuous (at least in a neighborhood of the solution) and differentiable

Gradient descent

$$\mathbf{a}_{k+1} = \mathbf{a}_k - \eta_k \nabla J(\mathbf{a}_k)$$

- the negative gradient, $-\nabla J(\mathbf{a})$ is locally the steepest descent towards a (local) minimum
- η_k is a *line search* parameter or *learning rate*
- start with some $\mathbf{a}_0$ and iterate until $|\eta_k \nabla J(\mathbf{a}_k)| < \theta$



Using Taylor's 2nd order approximation:

$$J(\mathbf{a}) \approxeq J(\mathbf{a}_k) + \nabla J(\mathbf{a} - \mathbf{a}_k) + \frac{1}{2}(\mathbf{a} - \mathbf{a}_k)^t \mathbf{H}(\mathbf{a} - \mathbf{a}_k),$$

where $\mathbf{H}$ is the *Hessian matrix* $\mathbf{H} = \left[\frac{\partial^2 J}{\partial a_i \partial a_j}\right]_{ij}$, one can find the optimal learning rate as

$$\eta_k = \frac{||\nabla J||^2}{(\nabla J)^t \mathbf{H}(\nabla J)}.$$

Note: if J is quadratic, then η_k is a constant.

Newton's method

$$\mathbf{a}_{k+1} = \mathbf{a}_k - \mathbf{H}^{-1}(\nabla J)$$

- works well for quadratic objective functions
- problems if the Hessian is singular
- no need to invert $\mathbf{H}$: solve the system $\mathbf{Hs} = -\nabla J$ and update the solution $\mathbf{a}_{k+1} = \mathbf{a}_k + \mathbf{s}$

Outline

The perceptron

- criterion: find $\mathbf{a}^*$ (or, equivalently, $\mathbf{w}^*$ and w_0^*) that minimize

$$J(\mathbf{a}) = -\sum_{i \in \mathbb{I}} \gamma_i = -\sum_{i \in \mathbb{I}} \langle \mathbf{a}, \mathbf{z}_i \rangle$$

  where $\mathbb{I}$ is the set of indices of misclassified points
- note: since $\gamma_i < 0$ for all misclassified points, $J(\mathbf{a}) \geq 0$, reaching 0 when all points are correctly classified
- it is easy to see that

$$\nabla_{\mathbf{a}} J(\mathbf{a}) = -\sum_{i \in \mathbb{I}} \mathbf{z}_i$$

- using *gradient descent* we get the updating iterations of the form

$$\mathbf{a}_{k+1} = \mathbf{a}_k + \eta_k \mathbf{z}_i$$

- the perceptron in guaranteed to converge in a finite number of iterations, *if the training set is separable* - Novikoff's thm
- from Novikoff's thm. the number of mistakes the perceptron makes is upper bounded by

$$\left(\frac{2R}{\gamma}\right)^2$$

where R is the radius of the sphere containing the data points, i.e. $R = \max_i \|\mathbf{x}_i\|$

Perceptron algorithm (batch perceptron)

Input: A separable training set $\mathcal{X} \times \mathcal{Y}$ and a stop criterion θ

Output: $\mathbf{a}_k$ such that $\gamma_i > 0, \forall i$ and k is the number of mistakes

$\mathbf{a}_0 \leftarrow \mathbf{0}$, $k \leftarrow 0$, $\eta_0 \leftarrow$ some initial value

repeat

    for $i = 1$ **to** n **do**

        if $\gamma_i = \langle \mathbf{a}_k, \mathbf{z}_i \rangle < 0$ **then**

            $\mathbf{a}_{k+1} \leftarrow \mathbf{a}_k + \eta_k \mathbf{z}_i$

            $k \leftarrow k + 1$

        end if

    end for

until $|\eta_k \sum_{i \in \mathbb{I}_k} \mathbf{z}_i| < \theta$

What about η_k? There are different "schedules" for modifying it...

- conditions: $\eta_k \geq 0$, $\lim_{m\to\infty} \sum_{k=1}^{m} \eta_k = \infty$ and

$$\lim_{m\to\infty} \frac{\sum_{k=1}^{m} \eta_k^2}{\left(\sum_{k=1}^{m} \eta_k\right)^2} = 0$$

- $\eta_k = \text{constant} > 0$
- $\eta_k \propto \frac{1}{k}$

- let **a** be the solution of the perceptron algorithm
- it is easy to see that $\mathbf{a} = \sum_{i=1}^{n} \alpha_i \mathbf{z}_i$ where

$$\alpha_i = \begin{cases} 0, \text{if point } i \text{ was always correctly classified} \\ > 0, \propto \text{the number of times point } i \text{ was misclassified} \end{cases}$$

- α_i can be seen as the importance (or contribution) of $\mathbf{z}_i$ to the classification rule
- the discriminant function can be rewritten as

$$\begin{aligned} h(\mathbf{x}) &= \langle \mathbf{a}, \mathbf{z} \rangle \\ &= \left\langle \sum_{i=1}^{n} \alpha_i \mathbf{z}_i, \mathbf{z} \right\rangle \\ &= \sum_{i=1}^{n} \alpha_i \langle \mathbf{z}_i, \mathbf{z} \rangle \end{aligned}$$

- this is the dual form of the perceptron algorithm

Dual formulation of the perceptron algorithm

Input: A training set $\mathcal{X} \times \mathcal{Y}$
Output: $\alpha = [\alpha_1, \dots, \alpha_n]$

```
α ← 0
repeat
    for i = 1 to n do
        if γ_i = (Σ_{j=1}^{n} α_j⟨z_j, z_i⟩) ≤ 0 then
            α_i ← α_i + 1
        end if
    end for
until no mistakes
```

Dual representation - remarks

- in dual representation, the only way data is involved in the algorithm/formula is through the dot products $\langle \mathbf{z}_i, \mathbf{z}_j \rangle$
- this property is valid for a large class of methods
- the dot products for the data can be computed offline, and stored in a *Gram matrix* $G = [\langle \mathbf{z}_i, \mathbf{z}_j \rangle]_{ij}$
- similarly, to predict the class of a new point $\mathbf{x}$, just (some of) the products $\langle \mathbf{z}, \mathbf{z}_i \rangle$ are needed

Relaxation procedures

Another objective function:

$$J_r(\mathbf{a}) = \frac{1}{2} \sum_{i \in \mathbb{I}} \frac{(\langle \mathbf{a}, \mathbf{z}_i \rangle - \xi)^2}{||\mathbf{z}_i||^2}$$

- it is smooth and has a continuous gradient function
- the term ξ is introduced to avoid the solution on the boundary of the solution space
- $||\mathbf{z}||^2$ is a normalization term to avoid J_r being dominated by the largest vectors
- 1/2 is merely to make the gradient nicer...

$$\nabla J_r = \sum_{i \in \mathbb{I}} \frac{\langle \mathbf{a}, \mathbf{z}_i \rangle - \xi}{||\mathbf{z}_i||^2} \mathbf{z}_i$$

Algorithms:

- *batch relaxation with margin*: update step:

$$\mathbf{a}_{k+1} = \mathbf{a}_k + \eta_k \sum_{i \in \mathbb{I}_k} \frac{\xi - \langle \mathbf{a}_k, \mathbf{z}_i \rangle}{||\mathbf{z}_i||^2} \mathbf{z}_i$$

- *single-sample relaxation with margin*: update step (for each misclassified sample $\mathbf{z}_i$):

$$\mathbf{a}_{k+1} = \mathbf{a}_k + \eta_k \frac{\xi - \langle \mathbf{a}_k, \mathbf{z}_i \rangle}{||\mathbf{z}_i||^2} \mathbf{z}_i$$

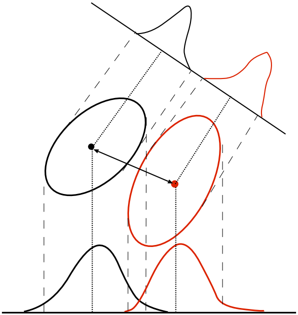
- if $\eta_k < 1$: *underrelaxation*; if $\eta_k > 1$: *overrelaxation*

Outline

Fisher criterion

Objective

Find the hyperplane $(\mathbf{w}, w_0)$ on which the projected data is maximally separated.



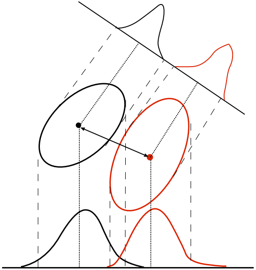
- the lenght of the projection of a vector $\mathbf{z}$ onto $\mathbf{w}$ is $\frac{\langle \mathbf{w},\mathbf{z}\rangle}{||\mathbf{w}||}$
- projection of the difference vector between the means of the two classes (taking $||\mathbf{w}|| = 1$):

$$|\langle \mathbf{w}, (\mu_{+1} - \mu_{-1})\rangle|$$

- maximize the difference, *relative* to the projected pool variance (scatter):

$$\frac{1}{n_{+1} + n_{-1}}(s_{+1}^2 + s_{-1}^2)$$

- $s_{\cdot}^2 = \sum_i(\langle \mathbf{w}, \mathbf{x}_i\rangle - \langle \mathbf{w}, \mu_{\cdot}\rangle)^2$ where the sum is over the elements in either class



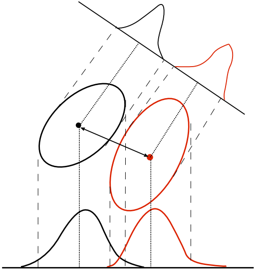
- the lenght of the projection of a vector **z** onto **w** is $\frac{\langle \mathbf{w}, \mathbf{z} \rangle}{||\mathbf{w}||}$
- projection of the difference vector between the means of the two classes (taking $||\mathbf{w}|| = 1$):

$$|\langle \mathbf{w}, (\mu_{+1} - \mu_{-1}) \rangle|$$

- maximize the difference, *relative* to the projected pool variance (scatter):

$$\frac{1}{n_{+1} + n_{-1}} (s_{+1}^2 + s_{-1}^2)$$

- $s_{\cdot}^2 = \sum_i (\langle \mathbf{w}, \mathbf{x}_i \rangle - \langle \mathbf{w}, \mu_{\cdot} \rangle)^2$ where the sum is over the elements in either class



Objective: maximize

$$J(\mathbf{w}) = \frac{|\langle \mathbf{w}, \mu_{+1} \rangle - \langle \mathbf{w}, \mu_{-1} \rangle|^2}{s_{+1}^2 + s_{-1}^2}$$

Fisher criterion

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} J(\mathbf{w}) = \arg\max_{\mathbf{w}} \frac{\mathbf{w}^t \mathbf{S}_b \mathbf{w}}{\mathbf{w}^t \mathbf{S}_w \mathbf{w}}$$

where

- $\mathbf{S}_b = (\mu_{+1} - \mu_{-1})(\mu_{+1} - \mu_{-1})^t \leftarrow$ *between-class scatter matrix*
- $\mathbf{S}_w = \sum_{i \in I_{+1}} (\mathbf{x}_i - \mu_{+1})(\mathbf{x}_i - \mu_{+1})^t + \sum_{i \in I_{-1}} (\mathbf{x}_i - \mu_{-1})(\mathbf{x}_i - \mu_{-1})^t$ $\leftarrow$ *within-class scatter matrix*
- $\mathbf{S}_w$ is proportional to sample covariance matrix for the pooled data

- $J_{\mathbf{w}}$ is also known as *Rayleigh quotient*
- the solution has the form

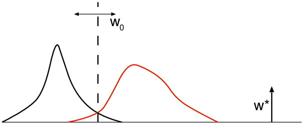
$$\mathbf{w}^* \propto \mathbf{S}_w^{-1}(\mu_{+1} - \mu_{-1})$$

  and it defines the direction of *Fisher's linear discriminant*
- the classification of d–dimensional points is transformed into a classification of one-dimensional points

- no assumption on the underlying distributions was made in finding $\mathbf{w}^*$
- the complete form of the linear discriminant is

$$\langle \mathbf{w}, \mathbf{x} \rangle + w_0 = 0$$

- to find w_0 one can, for example:
  - assume $p(\mathbf{x}| \pm 1)$ to be Gaussians: this leads to the previously seen formulas for w_0 (see Ch. 2)
  - try to find a value optimal for the training set



Outline

Outline

Linear regression problem

Find $\mathbf{a} = ([w_0, \mathbf{w}]^t)$ such that

$$b_i = \langle \mathbf{a}, \mathbf{z}_i \rangle, \quad i = 1, 2, \ldots, n$$

for some fixed positive constants b_i. In matrix notation, solve the linear system

$$\mathbf{Za} = \mathbf{b}$$

for $\mathbf{a}$.

- $\mathbf{Z}$ is a $n \times (d+1)$–dimensional matrix (*design matrix*), $\mathbf{a}$ is a $(d+1)$–elements vector.
- $\mathbf{b}$ is a n–elements vector (*response vector*)
- usually $n > d + 1$, so the system is *overdetermined* $\rightarrow$ no exact solution

- define the *error vector*

$$\mathbf{e} = \mathbf{Z}\mathbf{a} - \mathbf{b}$$

- *minimum squared error* criterion:

$$\text{minimize } J_s(\mathbf{a}) = ||\mathbf{e}||^2 = \sum_{i=1}^{n} (\langle \mathbf{a}, \mathbf{z}_i \rangle - b_i)^2$$

- at the minimum, the gradient $\nabla J_s = 2\mathbf{Z}^t(\mathbf{Z}\mathbf{a} - \mathbf{b})$ is zero $\Rightarrow \mathbf{a} = (\mathbf{Z}^t\mathbf{Z})^{-1}\mathbf{Z}^t\mathbf{b} = \mathbf{Z}^\dagger\mathbf{b}$, where $\mathbf{Z}^\dagger$ is the *pseudoinverse* of $\mathbf{Z}$
- the solution depends on **b** and different choices lead to various properties of the solution

Relation to Fisher's linear discriminant

- by properly choosing the class coding, one can show that MSE approach is equivalent to FDA
- $b_i = \frac{n}{n_{+1}}$ for the class "+1" (with n_{+1} elements) and $b_j = \frac{n}{n_{-1}}$ for the class "-1" (with n_{-1} elements)
- the MSE criterion for $\mathbf{a} = [w_0, \mathbf{w}]$ leads to

$$\mathbf{w} \propto n S_w^{-1}(\mu_{+1} - \mu_{-1})$$

  which is the direction of FDA
- additionally, it gives a value for the threshold: $w_0 = -\mu^t \mathbf{w}$ (μ is the grand mean vector)
- the decision rule becomes: if $\mathbf{w}^t(\mathbf{x} - \mu) > 0$ classify $\mathbf{x}$ as belonging to the first class

Relation with Bayesian classifier

- let the Bayesian discriminant be

$$h_0(\mathbf{x}) = P(g_1|\mathbf{x}) - P(g_2|\mathbf{x})$$

- the samples are assumed to be drawn *independently and identically distributed* from the underlying distribution

$$p(\mathbf{x}) = p(\mathbf{x}|g_1)P(g_1) + p(\mathbf{x}|g_2)P(g_2)$$

- MSE becomes

$$\epsilon^2 = \int (\langle \mathbf{a}, \mathbf{z} \rangle - h_0(\mathbf{x}))^2 \, p(\mathbf{x}) \, d\mathbf{x}$$

- $\rightarrow$ the solution to MSE problem, **a**, generates *an approximation* of the Bayesian discriminant
- $p(\mathbf{x}) =?$
- main problem of MSE: places more emphasis on points with high $p(\mathbf{x})$ instead of point near to the discrimination surface
- $\rightarrow$ the "best" approximation of Bayes decision does not necessarily minimize the probability of error

Numerical considerations on the LS problem

Using the pseudo-inverse is not the best technique, from a numerical stability perspective:

- computing $\mathbf{Z}^t\mathbf{Z}$ and $\mathbf{Z}^t\mathbf{b}$ may lead to information loss due to approximations in floating-point computations
- the conditioning of the system is worsen: $\text{cond}(\mathbf{Z}^t\mathbf{Z}) = [\text{cond}(\mathbf{Z})]^2$

Normally, a *matrix factorization* is used for improved numerical stability: QR, SVD,...

QR factorization

The $n \times m$ (with $m > n$) matrix $\mathbf{Z}$ can be factorized as

$$\mathbf{Z} = \mathbf{QR}$$

where

- $\mathbf{Q}$ is an *orthogonal matrix*: $\mathbf{Q}^t\mathbf{Q} = \mathbf{I} \Leftrightarrow \mathbf{Q}^{-1} = \mathbf{Q}^t$
- $\mathbf{R}$ is an *upper triangular matrix*

With this, the solution $\mathbf{a}$ to our problem is the solution of the *triangular system* (solved by backsubstitution):

$$\mathbf{Ra} = \mathbf{Q}^t\mathbf{b}$$

A statistical perspective

A linear model (linear regression) problem:

$$E[\mathbf{b}] = \mathbf{Za}, \qquad \text{under the assumption } \mathrm{Cov}(b) = \sigma^2 I$$

It can be shown that the *best linear unbiased estimator* is

$$\hat{\mathbf{a}} = (\mathbf{Z}^t\mathbf{Z})^{-1}\mathbf{Z}^t\mathbf{b} = \mathbf{R}^{-1}\mathbf{Q}^t\mathbf{b}$$

for a decomposition $\mathbf{Z} = \mathbf{QR}$. Then: $\hat{\mathbf{b}} = \mathbf{QQ}^t\mathbf{b}$. (Gauss-Markov thm.: LS estimator has the lowest variance among all unbiased linear estimators.) Also,

$$\mathrm{Var}(\hat{\mathbf{a}}) = (\mathbf{Z}^t\mathbf{Z})^{-1}\sigma^2 = (\mathbf{R}^t\mathbf{R})^{-1}\sigma^2$$

where $\sigma^2 = \|\mathbf{b} - \hat{\mathbf{b}}\|^2/(n - d - 1)$.

Outline

- the MSE criterion, $J_s(a) = \sum_{i=1}^{n}(\langle \mathbf{a}, \mathbf{z}_i \rangle - b_i)^2$ can also be minimized by gradient descent method
- since

$$\nabla J_s = 2\mathbf{Z}^t(\mathbf{Za} - \mathbf{b})$$

  the update rule becomes

$$\mathbf{a}_1 = \text{some value}$$
$$\mathbf{a}_{k+1} = \mathbf{a}_k + \eta_k \mathbf{Z}^t(\mathbf{Za}_k - \mathbf{b})$$

- if $\eta_k = \eta_1 / k$, the procedure convergest to a limiting value for $\mathbf{a}$ satistifying

$$\mathbf{Z}^t(\mathbf{Za} - \mathbf{b}) = 0$$

- this algorithm yields a solution even if $\mathbf{Z}^t\mathbf{Z}$ is singular or badly conditioned

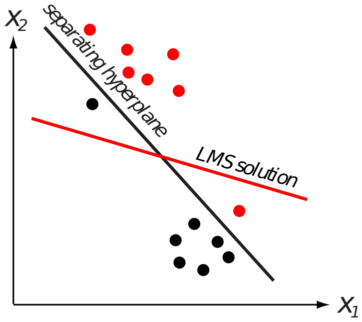
The Widrow-Hoff (or LMS) algorithm implements sequential gradient descent. (In signal processing: least mean squares filter - adaptive filtering...)

Input: A training set $(\mathbf{X}, \mathbf{y})$

Output: $\mathbf{a}$ - approximate MSE solution

1: initialize $\mathbf{a}, \mathbf{b}, \eta_1, \theta$ and $k \leftarrow 0$
2: **repeat**
3: $\quad k \leftarrow (k+1)n$
4: $\quad \mathbf{a} \leftarrow \mathbf{a} + \eta_k (b_k - \langle \mathbf{a}, \mathbf{z}_k \rangle)\mathbf{z}_k$
5: $\quad \eta_k \leftarrow \eta_1 / k$
6: **until** $|\eta_k (b_k - \langle \mathbf{a}, \mathbf{z}_k \rangle)\mathbf{z}_k| < \theta$

[DHS - Fig.5.17]



Outline

- consider $\mathbf{b} = \mathbf{Za}$ be the *margins* (instead of fixed labels)
- idea: adjust both the coefficients **a** and the margins **b** such that $\mathbf{b} > 0$ (each margin should be positive)
- formally: find **a** and $\mathbf{b} > 0$ such that

$$J_s(\mathbf{a}, \mathbf{b}) = \|\mathbf{Za} - \mathbf{b}\|^2$$

  becomes 0
- use a modified gradient descent, with gradient taken w.r.t. **a** and **b**