Linear Classifiers II, and Tangent Space for $k - NN$

Tomáš Svoboda

Vision for Robots and Autonomous Systems, Center for Machine Perception
Department of Cybernetics
Faculty of Electrical Engineering, Czech Technical University in Prague

May 12, 2022

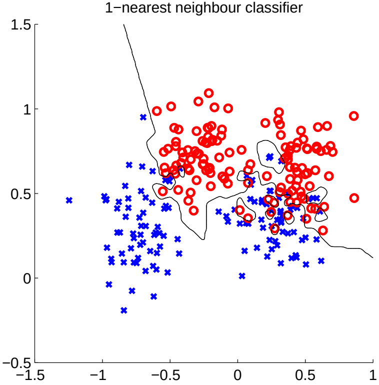
Outline

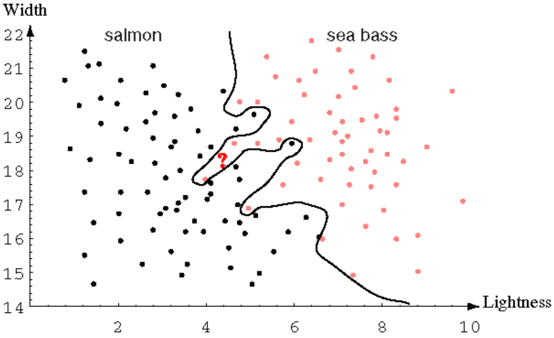
- $k - NN$, Tangent distance measure, invariance to rotation
- Better etalons by applying Fischer linear discriminator analysis.
- LSQ formulation of the learning task.

K-Nearest neighbors classification

For a query $\mathbf{x}$:

- Find K nearest $\mathbf{x}$ from the training (labeled) data.
- Classify to the class with the most exemplars in the set above.



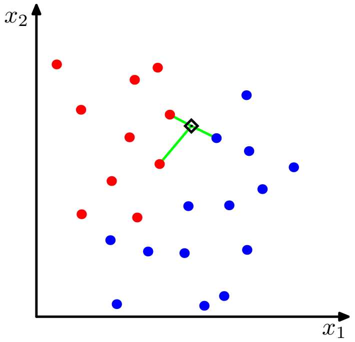


$K-$ Nearest Neighbor and Bayes $j^* = \operatorname{argmax}_j P(s_j|\mathbf{x})$

Assume data:

- N points $\mathbf{x}$ in total.
- N_j points in s_j class. Hence, $\sum_j N_j = N$.

We want to classify $\mathbf{x}$. Draw a sphere centered at $\mathbf{x}$ containing K points irrespective of class. V is the volume of this sphere. $P(s_j|\mathbf{x}) = ?$



$$P(s_j|\mathbf{x}) = \frac{P(\mathbf{x}|s_j)P(s_j)}{P(\mathbf{x})}$$

K_j is the number of points of class s_j among the K nearest neighbors.

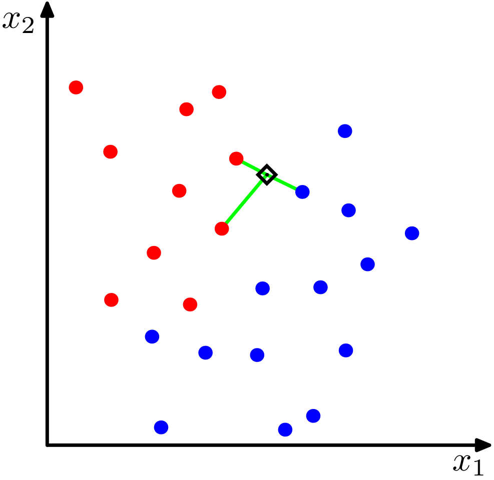
$$P(s_j) = \frac{N_j}{N}$$

$$P(\mathbf{x}) = \frac{K}{NV}$$

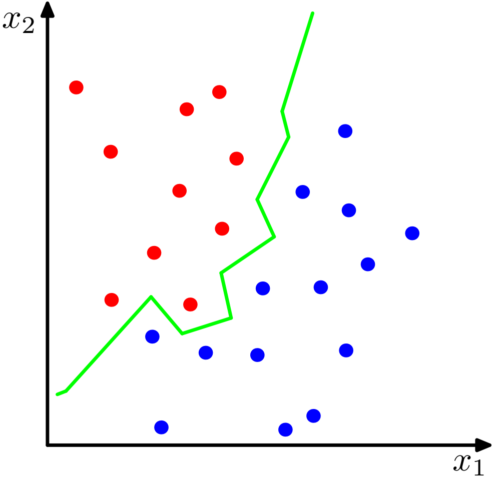
$$P(\mathbf{x}|s_j) = \frac{K_j}{N_j V}$$

$$P(s_j|\mathbf{x}) = \frac{P(\mathbf{x}|s_j)P(s_j)}{P(\mathbf{x})} = \frac{K_j}{K}$$

NN classification example



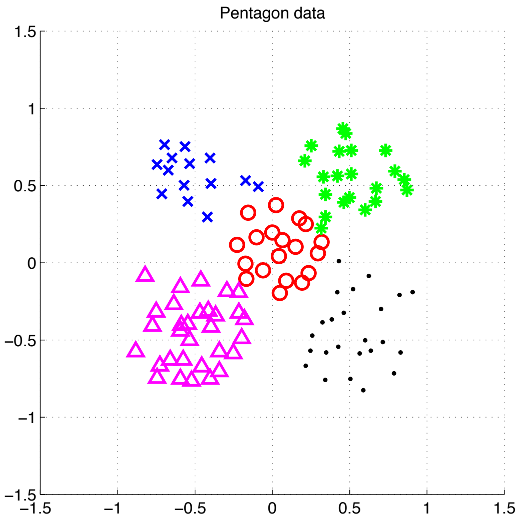
(a)

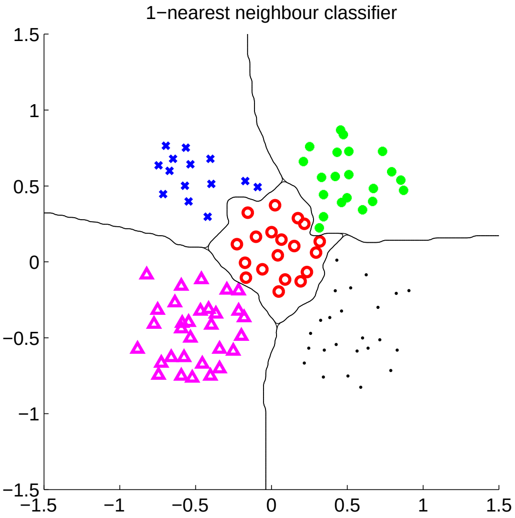


(b) [1]

[1] Figs from [1]

NN classification example





What is *nearest*? Metrics for NN classification ...

Metrics : a function D which is

- nonnegative,
- reflexive,
- symmetrical,
- satisfying triangle inequality:

$$D(\mathbf{x}_1, \mathbf{x}_2) \geq 0$$
$$D(\mathbf{x}_1, \mathbf{x}_2) = 0 \text{ iff } \mathbf{x}_1 = \mathbf{x}_2$$
$$D(\mathbf{x}_1, \mathbf{x}_2) = D(\mathbf{x}_1, \mathbf{x}_2)$$
$$D(\mathbf{x}_1, \mathbf{x}_2) + D(\mathbf{x}_2, \mathbf{x}_3) \geq D(\mathbf{x}_1, \mathbf{x}_3)$$

$D(\mathbf{x}_1, \mathbf{x}_2) = \|\mathbf{x}_1 - \mathbf{x}_2\|$ just fine, but

...

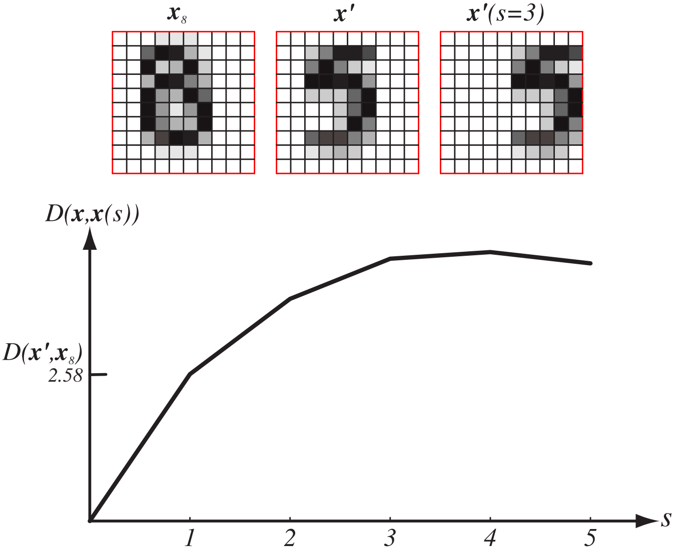
What is *nearest*? Metrics for NN classification . . .

Metrics : a function D which is

- nonnegative,
- reflexive,
- symmetrical,
- satisfying triangle inequality:

$$D(\mathbf{x_1}, \mathbf{x_2}) \geq 0$$
$$D(\mathbf{x_1}, \mathbf{x_2}) = 0 \text{ iff } \mathbf{x_1} = \mathbf{x_2}$$
$$D(\mathbf{x_1}, \mathbf{x_2}) = D(\mathbf{x_1}, \mathbf{x_2})$$
$$D(\mathbf{x_1}, \mathbf{x_2}) + D(\mathbf{x_2}, \mathbf{x_3}) \geq D(\mathbf{x_1}, \mathbf{x_3})$$

$D(\mathbf{x_1}, \mathbf{x_2}) = \|\mathbf{x_1} - \mathbf{x_2}\|$ just fine, but . . .



Invariance to geometrical transformations? (figure from [2])

Tangent space

Consider continuous tranformation:
e.g. rotation or translation not mirror reflection.

$\mathbf{x} = [x_1, x_2]^\top$ move along manifold $\mathcal{M}$
α is a tranformation parameter (e.g. angle)
Tangent vector $\boldsymbol{\tau}$ is a linearization

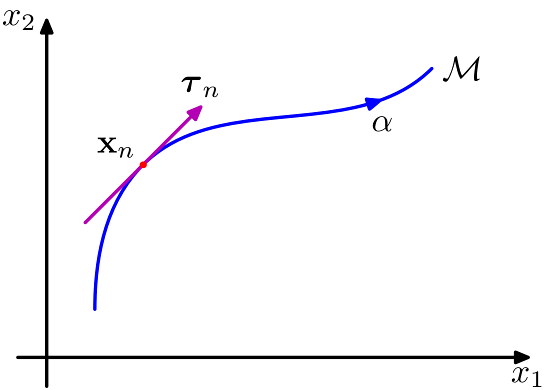
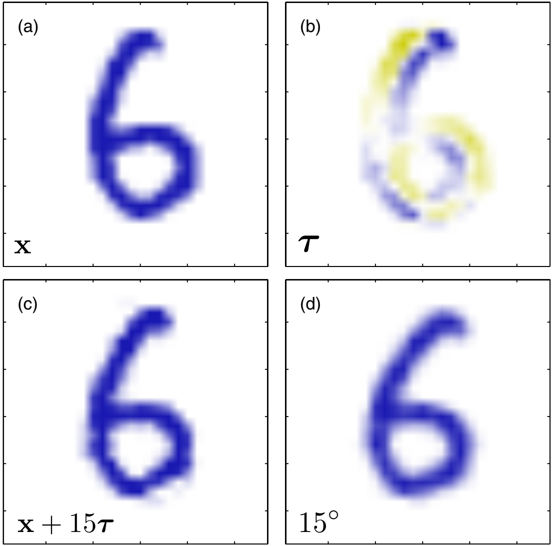
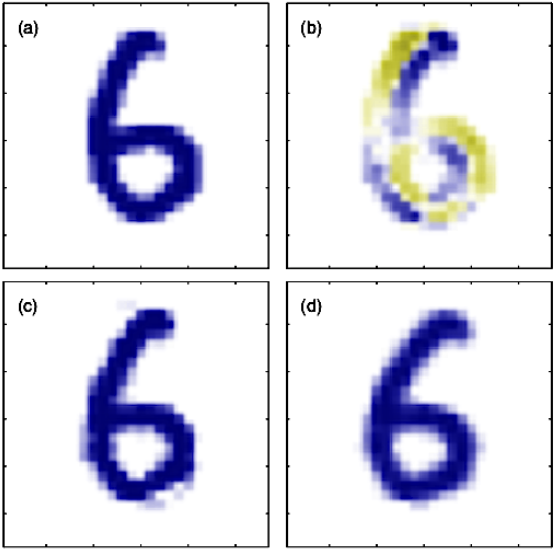


Figure from [1], slightly adapted

Approximating image rotation by adding tangent vector





Figures from [1], slighly adapted.

Combining more transformations

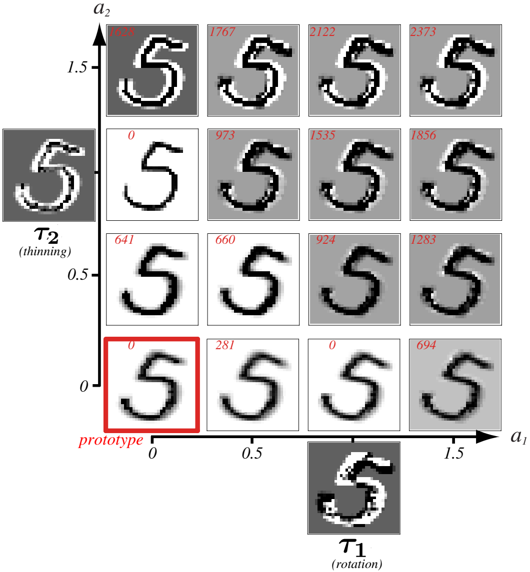
Approximate derivative by difference.
For all exemplars $\mathbf{x}'$
and all r tranformations $\mathcal{F}_i$

- $\boldsymbol{\tau_i} = \mathcal{F}_i(\mathbf{x}', \alpha_i) - \mathbf{x}'$

For each exemplar we have $d \times r$ matrix $\mathtt{T}$

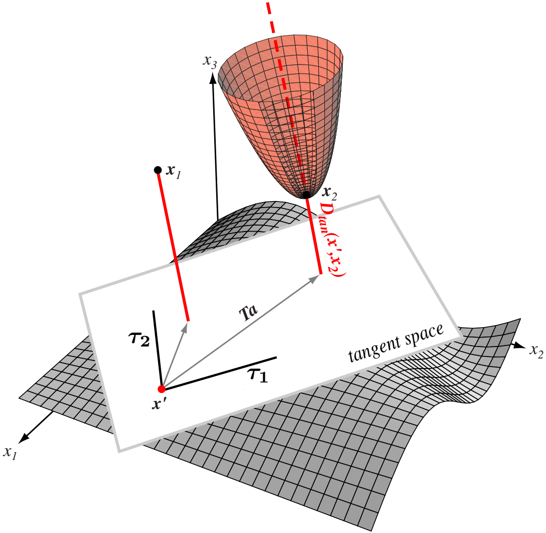
$$\mathtt{T} = [\boldsymbol{\tau_1}, \boldsymbol{\tau_2}, \cdots, \boldsymbol{\tau_r}]$$

Grouping coefficients $\mathbf{a} = [a_1, a_2]^\top$
Right image visualizes $\mathbf{x}' + \mathtt{T}\mathbf{a}$



Figures from [2], slighly adapted.

Minimizing distance to tangent space



$$D_{tan}(\mathbf{x}', \mathbf{x}) = \min_{\mathbf{a}} \|(\mathbf{x}' + \mathtt{T}\mathbf{a}) - \mathbf{x}\|$$

Gradient descent will do.

Figures from [2], slighly adapted.

Linear classifiers II

$$g(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + w_0$$

Decide s_1 if $g(\mathbf{x}) > 0$ and s_2 if $g(\mathbf{x}) < 0$

Figure from [2]

Linear classifiers II

$$g(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + w_0$$

Decide s_1 if $g(\mathbf{x}) > 0$ and s_2 if $g(\mathbf{x}) < 0$

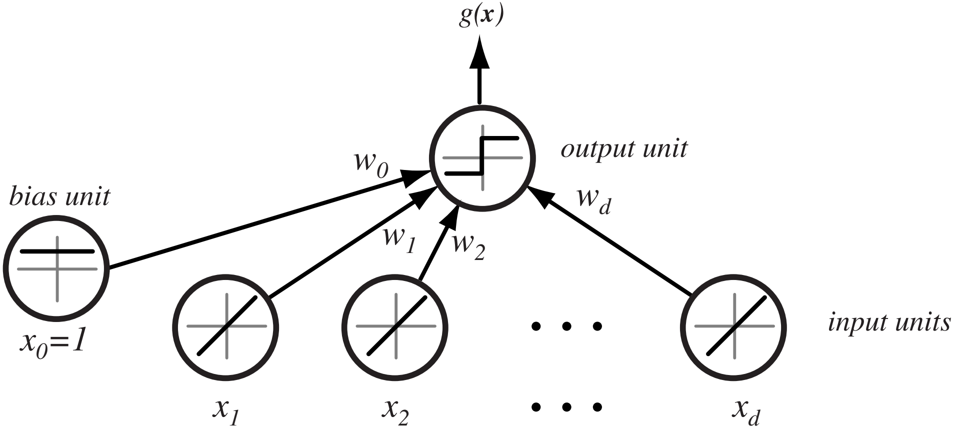
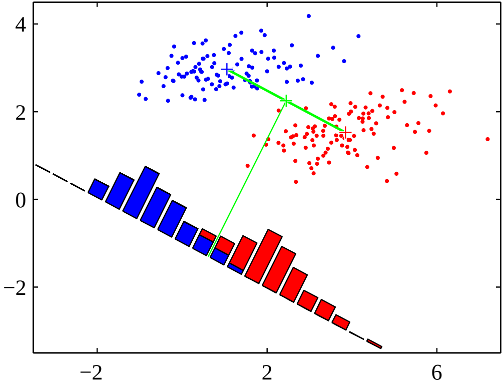
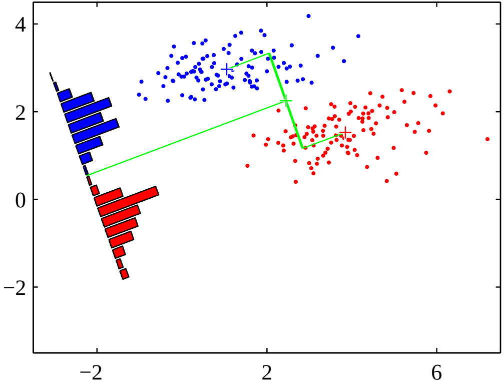


Figure from [2]

Fischer linear discriminant

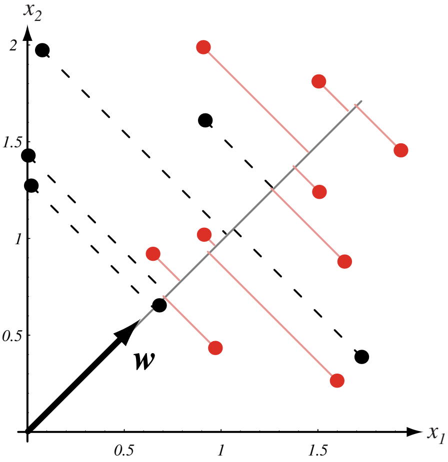




- Dimensionality reduction
- Maximize distance between means, ...
- ...and minimize within class variance. (minimize overlap)

Figures from [1]

Projections to lower dimensions $y = \mathbf{w}^\top \mathbf{x}$



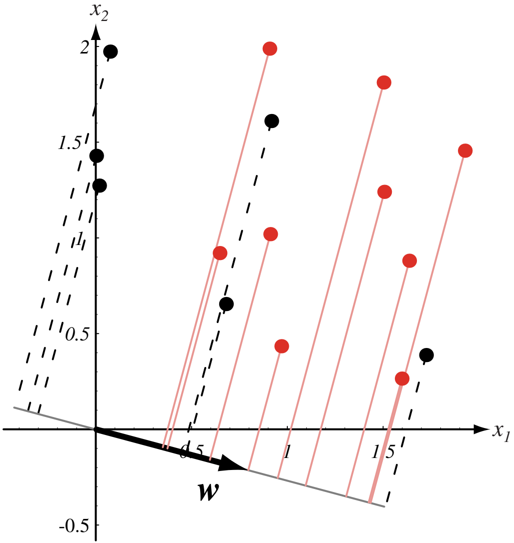


Figure from [2]

Projection to lower dimension $\mathbf{y} = \mathtt{W}^\top \mathbf{x}$

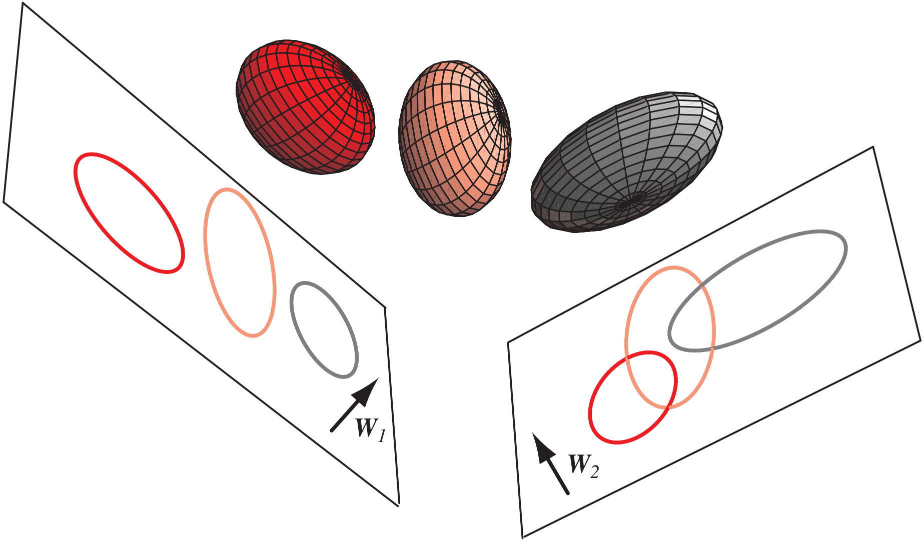
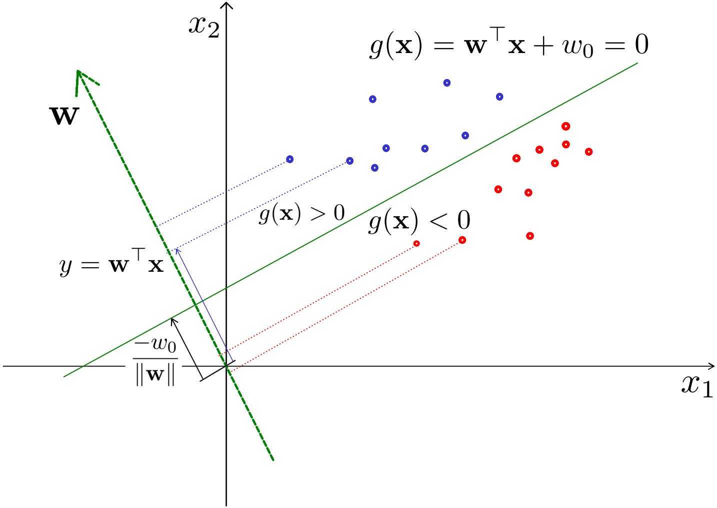
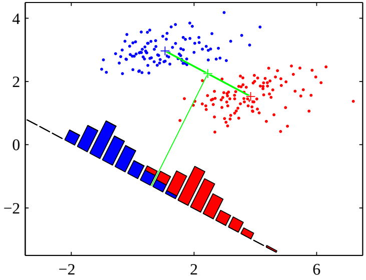


Figure from [2]

Finding the best projection $y = \mathbf{w}^\top \mathbf{x}$, $y \geq -w_0 \Rightarrow C_1$, otherwise C_2





Finding the best projection $y = \mathbf{w}^\top \mathbf{x}$, $y \geq -w_0 \Rightarrow C_1$, otherwise C_2

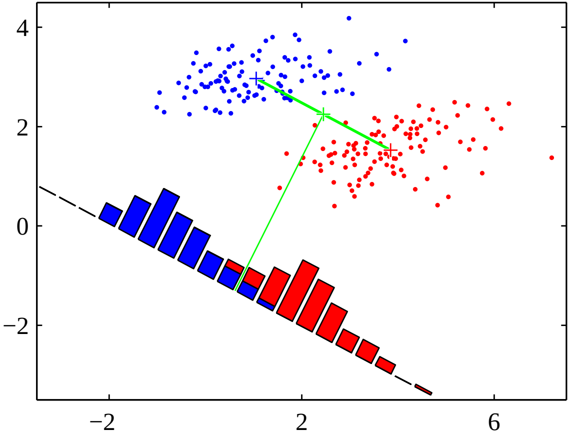
$$m_2 - m_1 = \mathbf{w}^\top(\mathbf{m}_2 - \mathbf{m}_1)$$

Within class scatter of projected samples

$$s_i^2 = \sum_{y \in C_i} (y - m_i)^2$$

Fischer criterion:

$$J(\mathbf{w}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2}$$



Finding the best projection $y = \mathbf{w}^\top \mathbf{x}$, $y \geq -w_0 \Rightarrow C_1$, otherwise C_2

$$m_2 - m_1 = \mathbf{w}^\top(\mathbf{m}_2 - \mathbf{m}_1)$$

$$s_i^2 = \sum_{y \in C_i} (y - m_i)^2$$

$$J(\mathbf{w}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2}$$

$$\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} = 0$$

$$\mathrm{S}_i = \sum_{x \in C_i} (\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^\top$$

$$\mathrm{S}_W = \mathrm{S}_1 + \mathrm{S}_2$$

$$\mathrm{S}_B = (\mathbf{m}_2 - \mathbf{m}_1)(\mathbf{m}_2 - \mathbf{m}_1)^\top$$

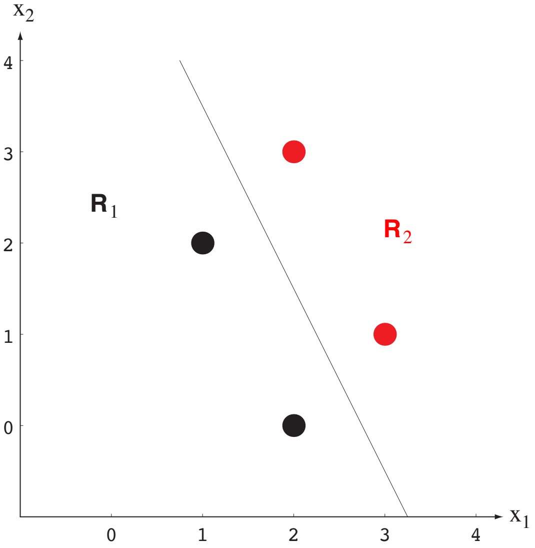
$$J(\mathbf{w}) = \frac{\mathbf{w}^\top \mathrm{S}_B \mathbf{w}}{\mathbf{w}^\top \mathrm{S}_W \mathbf{w}}$$

LSQ approach to linear classification

$$\mathbf{w} = \begin{bmatrix} w_0 \\ \mathbf{w} \end{bmatrix}$$

$$\mathtt{X}\mathbf{w} = \mathbf{b}$$

$$J(\mathbf{w}) = \|\mathtt{X}\mathbf{w} - \mathbf{b}\|^2$$



LSQ approach, better margins **b**?

$$\mathtt{X} = \begin{bmatrix} \mathtt{1}_1 & \mathtt{X}_1 \\ -\mathtt{1}_2 & -\mathtt{X}_2 \end{bmatrix}$$

$$\mathbf{b} = \begin{bmatrix} \frac{n}{n_1}\mathtt{1}_1 \\ \frac{n}{n_2}\mathtt{1}_2 \end{bmatrix}$$

References I

Further reading: Chapter 4 of [1], or chapter 3 and 5 of [2].

[1] Christopher M. Bishop.
Pattern Recognition and Machine Learning.
Springer Science+Bussiness Media, New York, NY, 2006.
PDF freely downloadable.

[2] Richard O. Duda, Peter E. Hart, and David G. Stork.
Pattern Classification.
John Wiley & Sons, 2nd edition, 2001.