Linear Classifiers II

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 26, 2021

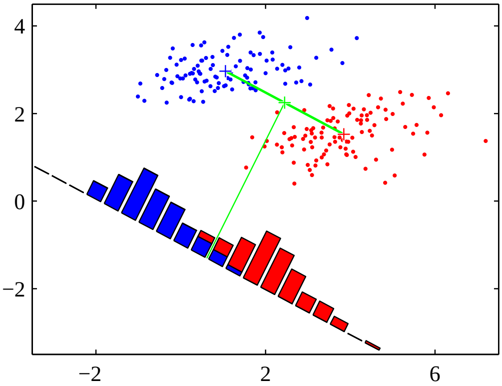
Notes

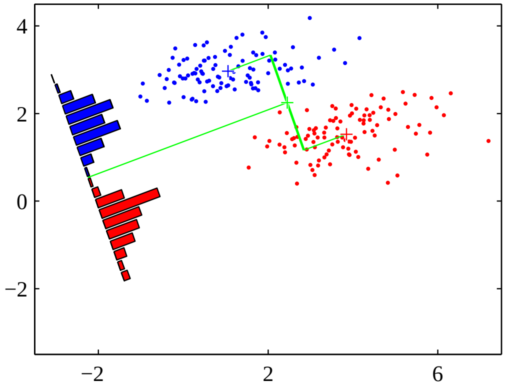
Linear Classifiers - supplement lecture

- Supplement to the lecture about learning Linear Classifiers (perceptron, . . .)
- Better etalons by applying Fischer linear discriminator analysis.
- LSQ formulation of the learning task.

Notes

Fischer linear discriminant





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

Figures from [1]

Notes

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

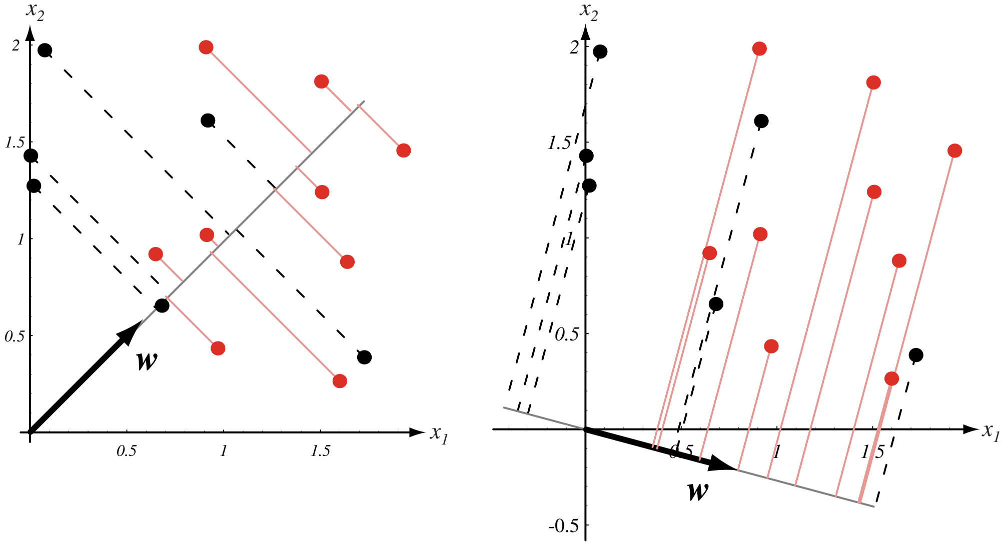


Figure from [2]

Notes

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

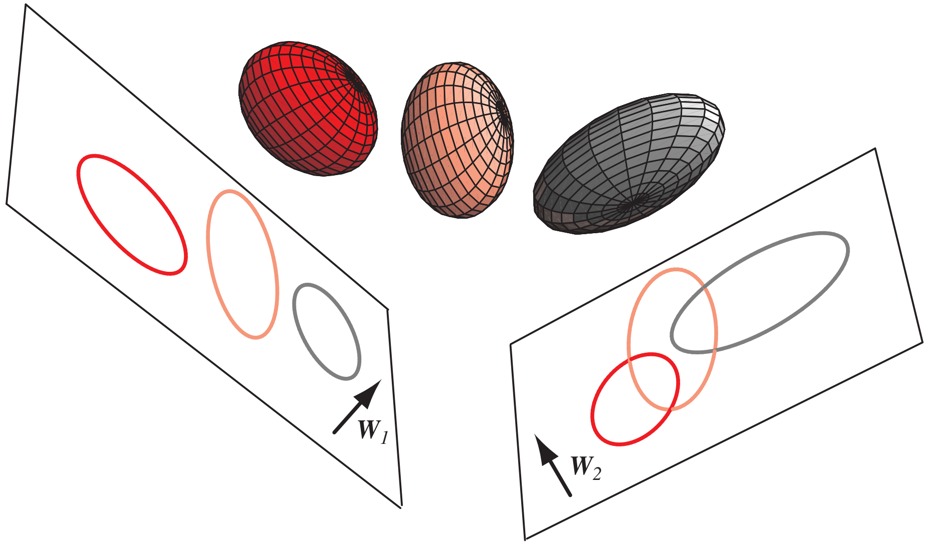
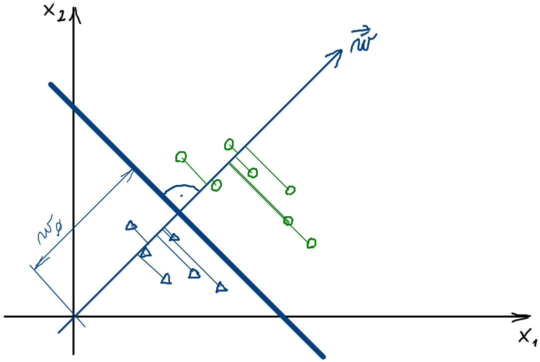
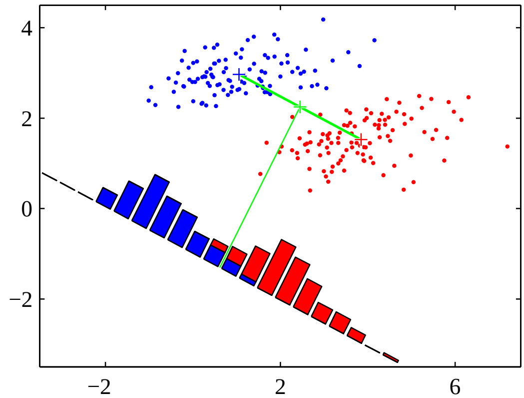


Figure from [2]

Notes

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





Notes

This is just to make sure we understand geometric meaning of $\mathbf{w}$, w_0 and the separating hyperplane. Remind the vector notation $\mathbf{w}$ means the same as $\vec{w}$.

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

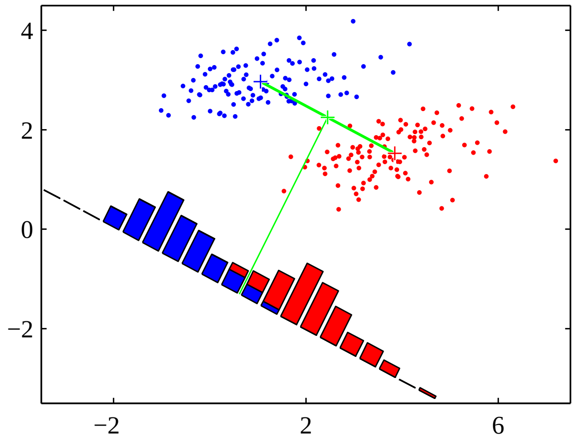
$$m_2 - m_1 = \mathbf{w}^\top(\mathbf{m}_2 - \mathbf{m}_1)$$

Within class scatter of projected samples

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

Fischer criterion:

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



Notes

Fischer criterion, max or min?

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

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

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

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

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

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

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

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

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

Notes

$\mathtt{S}_B$ stands for the *between* class scatter matrix. Remind

$$\left(\frac{f}{g}\right)' = \frac{f'g - fg'}{g^2}$$

hence we seek:

$$2\mathtt{S}_B\mathbf{w}(\mathbf{w}^\top \mathtt{S}_W \mathbf{w}) = (\mathbf{w}^\top \mathtt{S}_B \mathbf{w})2\mathtt{S}_W\mathbf{w}$$

the expressions within bracket are (unknown) scalars

$$\mathtt{S}_B\mathbf{w} = \lambda \mathtt{S}_W \mathbf{w}$$

leading to eigenvalue problem

$$\mathtt{S}_W^{-1}\mathtt{S}_B\mathbf{w} = \lambda \mathbf{w}$$

However, $\mathtt{S}_B\mathbf{w}$ is always in direction $(\mathbf{m}_2 - \mathbf{m}_1)$, and scale is not important

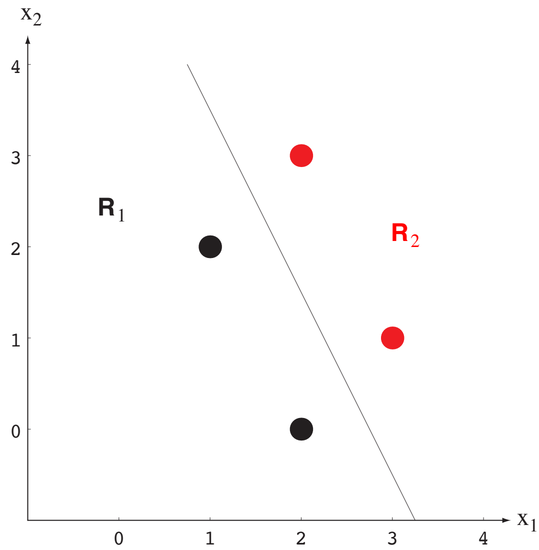
$$\mathbf{w} = \mathtt{S}_W^{-1}(\mathbf{m}_2 - \mathbf{m}_1)$$

LSQ approach to linear classification

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

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

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



Notes

Write dimensions to each symbol, n may stand for the number of points, d for dimensionality of the feature space.
Solving

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

yields $\mathbf{w} = (\mathtt{X}^\top \mathtt{X})^{-1}\mathtt{X}^\top \mathbf{b}$ Try to solve the above figure. We are looking for a separating hyperplane

$$\mathbf{w}^\top \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix} = 0$$

and we want points in training set distant from the hyperplane

$$\mathtt{X} = \begin{bmatrix} 1 & 1 & 2 \\ 1 & 2 & 0 \\ -1 & -3 & -1 \\ -1 & -2 & -3 \end{bmatrix}$$

$$\mathbf{b} = [1\ 1\ 1\ 1]^\top$$

Linear least squares not guaranteed to correctly classify everything on the training set. It's objective function not perfect for classification. Margins $\mathbf{b}$ were set quite arbitrarily.

Outliers can shift the decision boundary.

LSQ approach, better margins $\mathbf{b}$?

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

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

Notes

After some derivation it can be shown the LSQ solution is equivalent to Fisher linear discriminant
instert into intermediate result when solving $\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} = 0$

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

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.

Notes