Closing Session

Andreas Wichert
Department of Computer Science and Engineering
Técnico Lisboa

1

Understand your data

- Univariate data analysis
- Gaussian Distribution
- Relative Probability
- Bivariate data statistics
- Correlation (Pearson correlation)
- Spearman Rank (correlation)

2

Symbolical Machine learning

- Decision Trees
  - ID3, greedy search with heuristic (*algorithm*)
  - Heuristic function: **Shannon Entropy**
    - What dies the formula indicate? How close the distribution is to uniform distribution

$$H = -\sum_{t}^{K} p(x_t) \cdot \log_2 p(x_t) = -\sum_{x \in X} p(x) \cdot \log_2 p(x).$$

3

Model Evaluation

- Confusion Matrix for binary Classifier

true/actual/target

predicted		P	N	
	P	True Positives (TP)	False Positives (FP)	TP+FP
	N	False Negatives (FN)	True Negatives (TN)	FN+TN
		P=TP+FN	N=FP+TN	All=P+N

Recall/sensitivity
% of positive observations predicted as positive

$$Recall = \frac{TP}{P} = \frac{TP}{TP+FN}$$

Precision
% of positive observations among the observations predicted as positive

$$Precision = \frac{TP}{TP + FP}$$

4

Balanced Measure

- Precision and Recall have to be simultaneously interpreted.
- We can combine both values with the *harmonic mean*

$$F = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

- Both values are evenly weighted.
- This measure is also called the balanced measure.

5

Bayes' Rule

$$p(h_k|D) = \frac{p(D|h_k) \cdot p(h_k)}{p(D)} = \frac{p(D, h_k)}{p(D)}$$

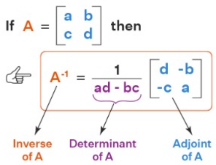
- *$p(h_k)$ is called the* ***prior*** *(before)*
  - *For example, what is the probability of some illness in Portugal*
- *$p(D|h_k)$ is called* ***likelihood*** *and can can be easily estimated*
  - *For example, what is the probability that some illness generates some symptoms?*
  - *$p(D,h_k)$ is called* ***joint distribution***
- *$p(h_k|D)$ is called* ***posterior probability***

6

Bayesian optimal classifier: Multivariate Gaussian

Approximate a multivariate Gaussian distribution using the following points: $\{(-2,2)^T, (-1,3)^T, (0,1)^T, (-2,1)^T\}$

- $\mu = \frac{1}{4}\left(\begin{bmatrix}-2\\2\end{bmatrix} + \begin{bmatrix}-1\\3\end{bmatrix} + \begin{bmatrix}0\\1\end{bmatrix} + \begin{bmatrix}-2\\1\end{bmatrix}\right) = \begin{bmatrix}-1.25\\1.75\end{bmatrix}$
- $c_{12} = c_{21} = \frac{(-2+1.25)(2-1.75)+(-1+1.25)(3-1.75)+(0+1.25)(1-175)+(-2+1.25)(1-1.75)}{3} = -0.83$
- $c_{11} = \frac{(-2+1.25)^2+(-1+1.25)^2+(0+1.25)^2+(-2+1.25)^2}{3} = 0.92$
- $c_{22} = \frac{(2-1.75)^2+(3-1.75)^2+(1-175)^2+(1-1.75)^2}{3} = 0.92$
- $\Sigma = \begin{pmatrix}c_{11} & c_{21}\\ c_{12} & c_{22}\end{pmatrix} = \begin{pmatrix}0.92 & -0.083\\ -0.083 & 0.92\end{pmatrix}$.
- $\Sigma^{-1} = \begin{pmatrix}1.1 & 0.1\\ 0.1 & 1.1\end{pmatrix}$. Det(Σ)=| Σ|=0.833Type equation here.



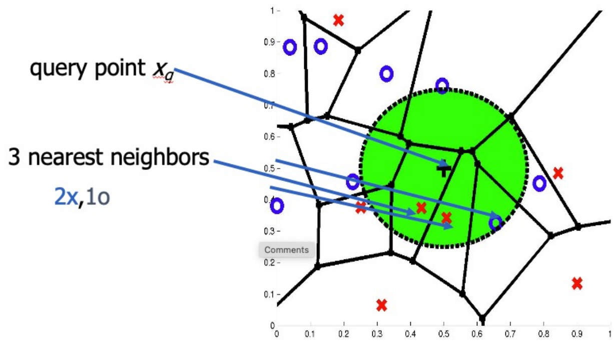
$$N(\mathbf{x}|\boldsymbol{\mu},\Sigma) = \frac{1}{(2\pi)^{2/2}\sqrt{0.083}} exp\left(-\frac{1}{2}\left(\begin{bmatrix}x_1\\x_2\end{bmatrix} - \begin{bmatrix}-1.25\\1.75\end{bmatrix}\right)^T \begin{bmatrix}1.1 & 0.1\\0.1 & 1.1\end{bmatrix}\left(\begin{bmatrix}x_1\\x_2\end{bmatrix} - \begin{bmatrix}-1.25\\1.75\end{bmatrix}\right)\right)$$

7

K Nearest Neighbour

- The cost of the learning process is *0*, all the cost is in the computation of the prediction
- This kind learning is also known as *lazy learning*

3-Nearest Neighbors



8

Error Functions

- **Error Mnimization**

$$E(\mathbf{w}) = \frac{1}{2} \cdot \sum_{\eta=1}^{N} (t_\eta - y(\mathbf{x}_\eta, \mathbf{w}))^2 = \frac{1}{2} \cdot \|\mathbf{t} - \mathbf{y}\|^2$$

$$E(\mathbf{w}) = \frac{1}{2} \cdot \sum_{\eta=1}^{N} \left(t_\eta - \mathbf{w}^T \cdot \mathbf{x}_\eta)\right)^2 = \frac{1}{2} \cdot \sum_{\eta=1}^{N} \left(t_\eta - \mathbf{x}_\eta^T \cdot \mathbf{w})\right)^2$$

$$E(\mathbf{w}) = \frac{1}{2} \cdot \|\mathbf{t} - X \cdot \mathbf{w}\|^2 = \frac{1}{2} \cdot (\mathbf{t} - X \cdot \mathbf{w})^T (\mathbf{t} - X \cdot \mathbf{w})$$

9

Least-Squares Estimation

We set the gradient of $E(\mathbf{w})$ to zero with the gradient operator

$$\nabla = \left[\frac{\partial}{\partial w_1}, \frac{\partial}{\partial w_2}, \cdots, \frac{\partial}{\partial w_D}\right]^T$$

$$\nabla E(\mathbf{w}) = \left[\frac{\partial E}{\partial w_1}, \frac{\partial E}{\partial w_2}, \cdots, \frac{\partial E}{\partial w_D}\right]^T$$

$$\nabla E(\mathbf{w}) = \nabla \left(\frac{1}{2} \cdot (\mathbf{t} - X \cdot \mathbf{w})^T \cdot (\mathbf{t} - X \cdot \mathbf{w})\right) = 0$$

10

With

$$\Phi_{\eta,j} = \phi_j(\mathbf{x}_\eta)$$

- Dimensions change since the dimension are not determined by the dimension of the vector **x** which is D
- The number of the is M-1

$$\begin{pmatrix} y_1 \\ \vdots \\ y_\eta \\ \vdots \\ y_N \end{pmatrix} = \begin{pmatrix} 1 & \phi_{1,1} & \phi_{1,2} & \cdots & \phi_{1,M-1} \\ 1 & \phi_{2,1} & \phi_{2,2} & \cdots & \phi_{2,M-1} \\ \vdots & \vdots & \ddots & \vdots & \\ 1 & \phi_{N,1} & \phi_{N,2} & \cdots & \phi_{N,M-1} \end{pmatrix} \cdot \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_j \\ \vdots \\ w_{M-1} \end{pmatrix}$$

with $\Phi^\dagger$ is Moore-Penrose or the pseudo-inverse of Φ as before with

$$\Phi^\dagger = \left(\Phi^T \cdot \Phi\right)^{-1} \cdot \Phi^T$$

11

With

$$\Phi_{\eta,j} = \phi_j(\mathbf{x}_\eta)$$

- Dimensions change since the dimension are not determined by the dimension of the vector **x** which is D
- The number of the is M-1

$$\begin{pmatrix} y_1 \\ \vdots \\ y_\eta \\ \vdots \\ y_N \end{pmatrix} = \begin{pmatrix} 1 & \phi_{1,1} & \phi_{1,2} & \cdots & \phi_{1,M-1} \\ 1 & \phi_{2,1} & \phi_{2,2} & \cdots & \phi_{2,M-1} \\ \vdots & \vdots & \ddots & \vdots & \\ 1 & \phi_{N,1} & \phi_{N,2} & \cdots & \phi_{N,M-1} \end{pmatrix} \cdot \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_j \\ \vdots \\ w_{M-1} \end{pmatrix}$$

with $\Phi^\dagger$ is Moore-Penrose or the pseudo-inverse of Φ as before with

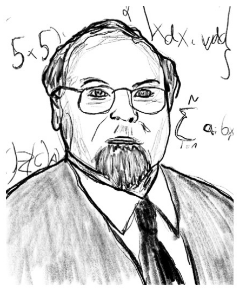
$$\Phi^\dagger = \left(\Phi^T \cdot \Phi\right)^{-1} \cdot \Phi^T$$

12

Quadratic Function

- Now we can define the quadratic function, minimising it is equivalent to maximising $\mathbf{w}_{MAP}(N)$

$$E(\mathbf{w}) = \frac{1}{2} \cdot \sum_{\eta=1}^{N} (t_\eta - \mathbf{w}^T \cdot \mathbf{x}_\eta)^2 + \frac{\lambda}{2} \|\mathbf{w}\|^2$$



- We set the gradient of $E(\mathbf{w})$ to zero with the gradient operator

$$\nabla E(\mathbf{w}) = \nabla \left(\frac{1}{2} \cdot (\mathbf{t} - X \cdot \mathbf{w})^T \cdot (\mathbf{t} - X \cdot \mathbf{w}) + \frac{\lambda}{2} \mathbf{w}^T \mathbf{w} \right) = 0$$

$$-2 \cdot X^T \cdot \mathbf{t} + 2 \cdot X^T \cdot X \cdot \mathbf{w} + 2 \cdot \lambda \cdot \mathbf{w} = 0$$

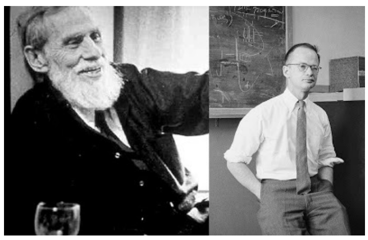
$$-X^T \cdot \mathbf{t} + X^T \cdot X \cdot \mathbf{w} + \lambda \cdot \mathbf{w} = 0$$

$$X^T \cdot \mathbf{t} = \left(X^T \cdot X + \lambda \cdot I \right) \cdot \mathbf{w}$$

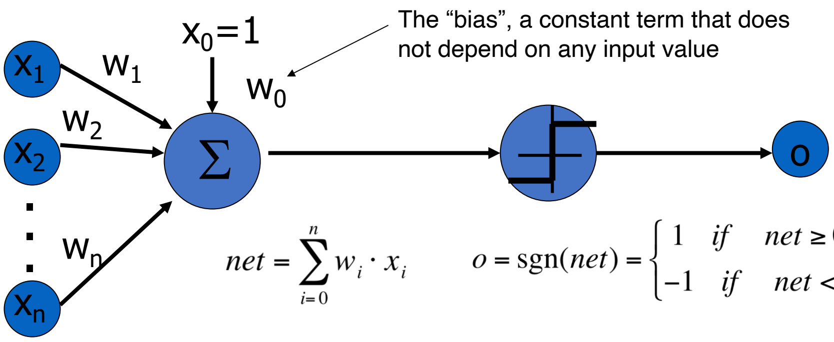
$$\left(X^T \cdot X + \lambda \cdot I \right)^{-1} \cdot X^T \cdot \mathbf{t} = \mathbf{w}$$

13

Perceptron (1957)



- Linear threshold unit (LTU)



$$net = \sum_{i=0}^{n} w_i \cdot x_i$$

$$o = \mathrm{sgn}(net) = \begin{cases} 1 & if \quad net \geq 0 \\ -1 & if \quad net < 0 \end{cases}$$

McCulloch-Pitts model of a neuron (1943)

14

In this example a linearly separable training set is described by four vectors

$$\mathbf{x}_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \mathbf{x}_2 = \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \mathbf{x}_3 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \mathbf{x}_4 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

and the corresponding targets

$$t_1 = -1, t_2 = 1, t_3 = 1, t_4 = -1.$$

The weights are initialized to 1 and the learning rate η for simplicity is set to 1 as well

$$w_0 = 1, w_1 = 1, w_2 = 1, \quad \eta = 1$$

15

$$o_1 = sgn(1 \cdot 0 + 1 \cdot 0 + 1) = sgn(1) = 1; \quad \delta_1 = -2; \quad \mathbf{w} = \begin{pmatrix} -1 \\ 1 \\ 1 \end{pmatrix},$$

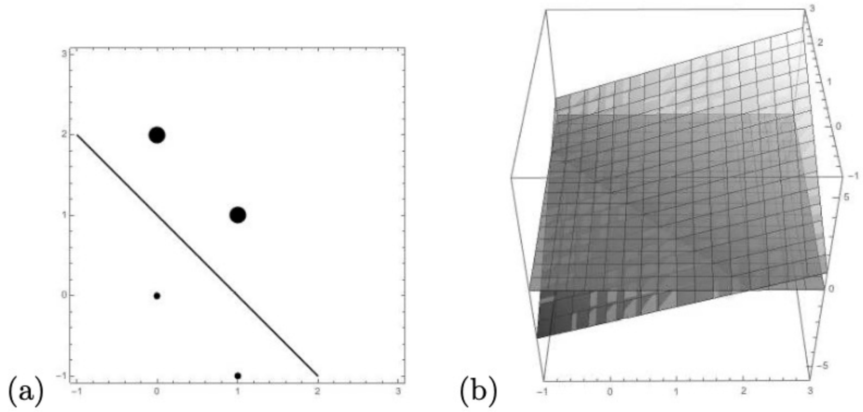
$$o_2 = sgn(1 \cdot 0 + 1 \cdot 2 - 1) = sgn(1) = 1; \quad \delta_2 = 0; \quad \mathbf{w} = \begin{pmatrix} -1 \\ 1 \\ 1 \end{pmatrix},$$

$$o_3 = sgn(1 \cdot 1 + 1 \cdot 1 - 1) = sgn(1) = 1; \quad \delta_3 = 0; \quad \mathbf{w} = \begin{pmatrix} -1 \\ 1 \\ 1 \end{pmatrix},$$

$$o_4 = sgn(1 \cdot 1 + 1 \cdot (-1) - 1) = sgn(-1) = -1; \quad \delta_4 = 0; \quad \mathbf{w} = \begin{pmatrix} -1 \\ 1 \\ 1 \end{pmatrix}$$

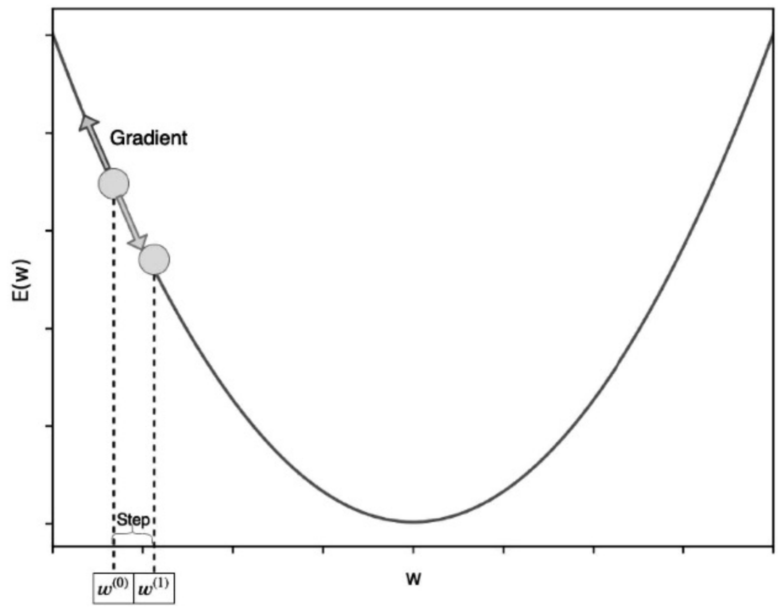
For additional epochs, the weights do not change

16



(a) The two classes 1 (indicated by a big point) and −1 (indicated a small point) are separated by the line $-1+x_1+x_2 = 0$. (b) The hyperplane $-1+x_1+x_2 = y$ defines the line for y = 0.

17



18

Continuous activation functions - Squared Error

For continuous activation function $\phi()$

$$o_k = \phi\left(\sum_{j=0}^{D} w_j \cdot x_{k,j}\right).$$

we can define as well the update rule for gradient decent with the differential

$$\frac{\partial E}{\partial w_j} = \sum_{k=1}^{N}(t_k - o_k) \cdot \frac{\partial}{\partial w_j}\left(t_k - \phi\left(\sum_{j=0}^{D} w_j \cdot x_{k,j}\right)\right)$$

$$\frac{\partial E}{\partial w_j} = \sum_{k=1}^{N}(t_k - o_k) \cdot \left(-\phi'\left(\sum_{j=0}^{D} w_j \cdot x_{k,j}\right) \cdot x_{k,j}\right)$$

$$\frac{\partial E}{\partial w_j} = -\sum_{k=1}^{N}(t_k - o_k) \cdot \phi'\left(\sum_{j=0}^{D} w_j \cdot x_{k,j}\right) \cdot x_{k,j}.$$

19

- The **softmax function** is used in various multi class classification methods, such as multinomial logistic regression (also known as softmax regression) with the prediction

$$\sigma(net_{ks}) = \frac{\exp(net_{ks})}{\sum_{t=1}^{K} \exp(net_{kt})}$$

20

Logistic Regression Algorithm

Given a training set (sample)

$$Data = \{(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), \cdots, (\mathbf{x}_k, \mathbf{y}_k), \cdots, (\mathbf{x}_N, \mathbf{y}_N)\}$$

with $\mathbf{y}_k$ represented as vectors of dimension K. During the training each neuron is trained individually withs its target value y_{kt}

$$y_{kt} \in \{0, 1\}, \quad \sum_{t=1}^{K} y_{kt} = 1$$

the goal of the algorithm is to correctly classify the test set (population) into K classes $C_1 = 100\cdots$, $C_2 = 010\cdots$, $C_3 = 001\cdots$, $\cdots$

21

Sigmoid Unit versus Logistic Regression

Sigmoid Unit is with target, should be positive (between zero and one):

$$\Delta w_j = \eta \cdot \alpha \cdot \sum_{k=1}^{N} (t_k - o_k) \; \sigma(net_{k,j}) \cdot (1 - \sigma(net_{k,j})) \; x_{k,j}$$

Logistic Regression is with target $t_k \in \{0, 1\}$

$$\Delta w_j = \eta \cdot \sum_{k=1}^{N} (t_k - o_k) \cdot x_{k,j}.$$

If we assume $\alpha = 1$ then the difference between sigmoid unit and the logistic regression that was derived by maximising the negative logarithm of the likelihood is

$$\sigma(net_{k,j}) \cdot (1 - \sigma(net_{k,j})) \geq 0$$

the step size in the direction of gradient. Does it mean that Sigmoid Unit converge faster?

22

Linear Unit versus Logistic Regression

Target can be any value and can be solved by closed-form solution, by pseudo inverse

$$o_k = \sum_{j=0}^{D} w_j \cdot x_{k,j}$$

Target $t_k \in \{0,1\}$ cannot be solved by closed-form solution

$$o_k = \frac{1}{1 + e^{(-\alpha \cdot (\sum_{j=0}^{N} w_j \cdot x_{k,j}))}}$$

$$\Delta w_j = \eta \cdot \sum_{k=1}^{N} (t_k - o_k) \cdot x_{k,j}.$$

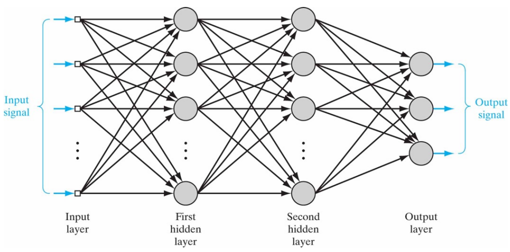
Logistic Regression as well as the sigmoid unit gives a better decision boundary.

For Sigmoid (Logistic) distant points from the decision boundary have the same impact

23

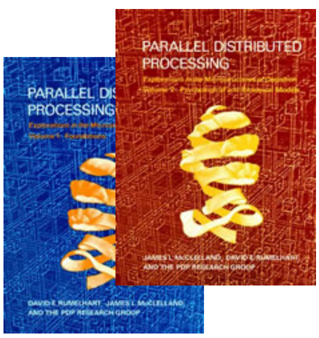
Back-propagation (1980)

- Back-propagation is a learning algorithm for multi-layer neural networks



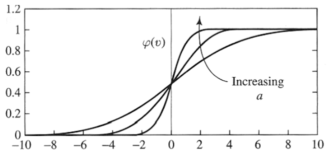
Parallel Distributed Processing - Vol. 1
Foundations
David E. Rumelhart, James L. McClelland and the PDP Research Group

What makes people smarter than computers? These volumes by a pioneering neurocomputing.....



24

- We have to use a nonlinear differentiable activation function in **hidden units**
  - Examples:



$$f(x) = \sigma(x) = \frac{1}{1 + e^{(-\alpha \cdot x)}}$$

$$f^{'}(x) = \sigma^{'}(x) = \alpha \cdot \sigma(x) \cdot (1 - \sigma(x))$$

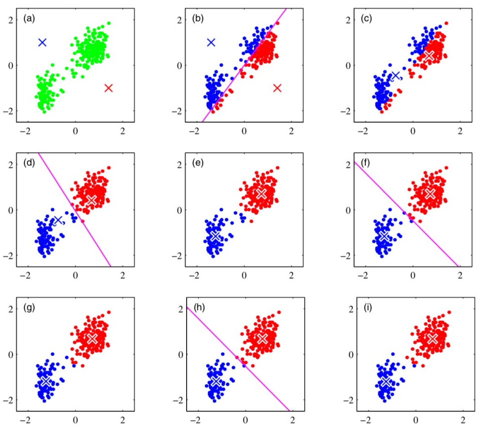
$$f(x) = \tanh(\alpha \cdot x)$$

$$f^{'}(x) = \alpha \cdot (1 - f(x)^2)$$

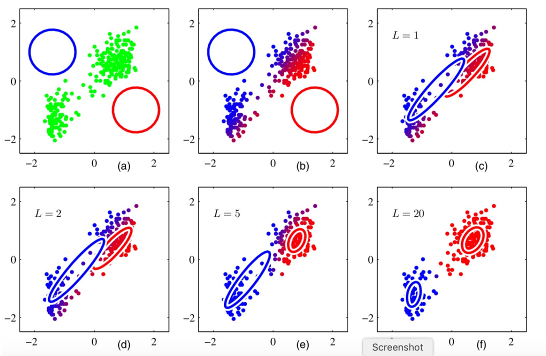
25

Clustering

- K-Means

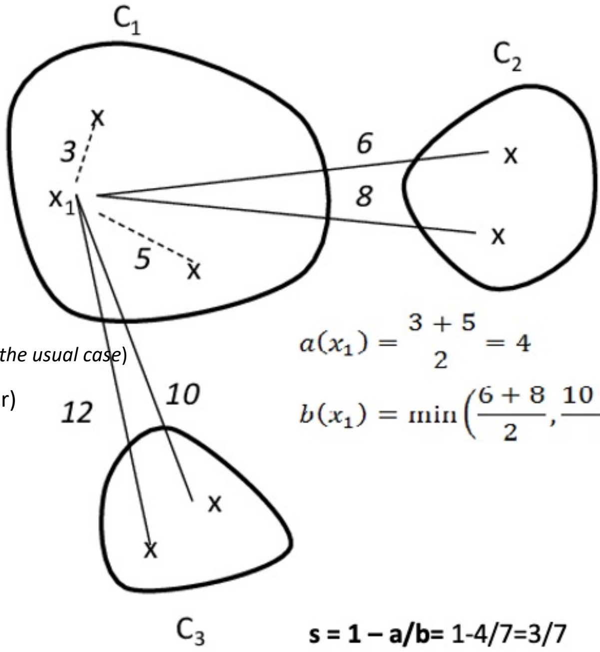


EM: Expectation Maximization algorithm
(Bayes in E step)



26

Validity index Silhouette



s = 1 – a/b if a < b, (*or s = b/a - 1 if a ≥ b, not the usual case*)

between -1 and 1 (the closer to 1 the better)

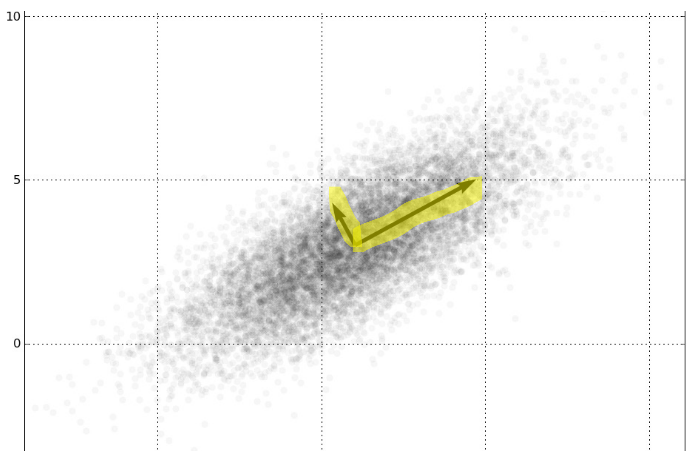
$$a(x_1) = \frac{3+5}{2} = 4$$

$$b(x_1) = \min\left(\frac{6+8}{2}, \frac{10+12}{2}\right) = 7$$

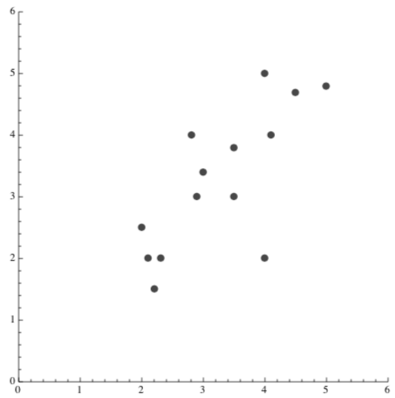
s = 1 – a/b= 1-4/7=3/7

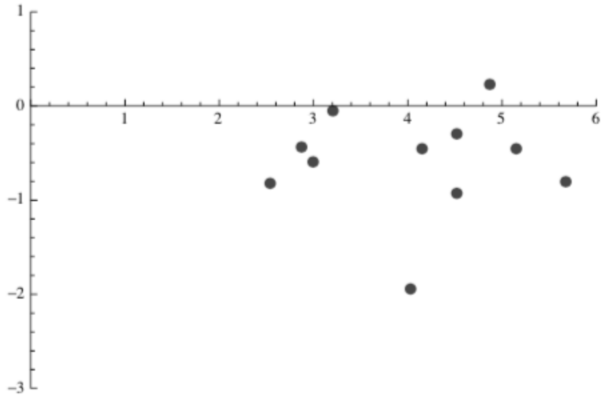
27

PCA

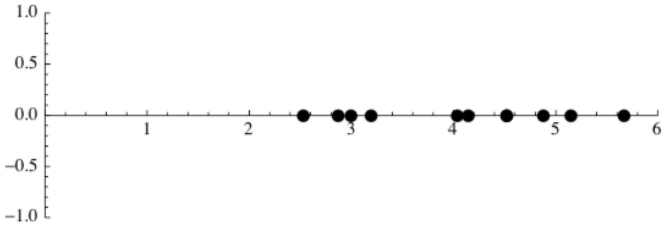


28





$$\mathbf{y} = U^\top \cdot \mathbf{x} = \begin{pmatrix} 0.611454 & 0.79128 \\ -0.79128 & 0.611454 \end{pmatrix} \cdot \mathbf{x},$$



$$\mathbf{z} = W^\top \cdot \mathbf{x} = \begin{pmatrix} 0.611454 & 0.79128 \end{pmatrix} \cdot \mathbf{x}$$

29

Preparation:

- Practical Lectures
- Home works
- Prepare a **“cheat sheet”**

- Open Book Exam, Calculator
  - Careful: **Organize your knowledge**, you will have **no time** for search
- No ebook Reader, Computer, Smartphone, Smartwatch etc..

30