COM2004/3004
Data Driven Computing

Week 4a: Linear Classifiers

Autumn Semester

Overview

Week 4 – this week

- Today: introducing linear classifiers
- Tutorial: examples Bayes classification, linear classifiers
- Lab Class: training and using a classifier with Python
- Lecture: the Perceptron linear classifier

Next Week

- Lecture: instance-based classifiers
- Lab Class: K-nearest neighbour classification with Python
- Lecture: more on instance-based classifiers, exam questions

Overview

Linear classifiers

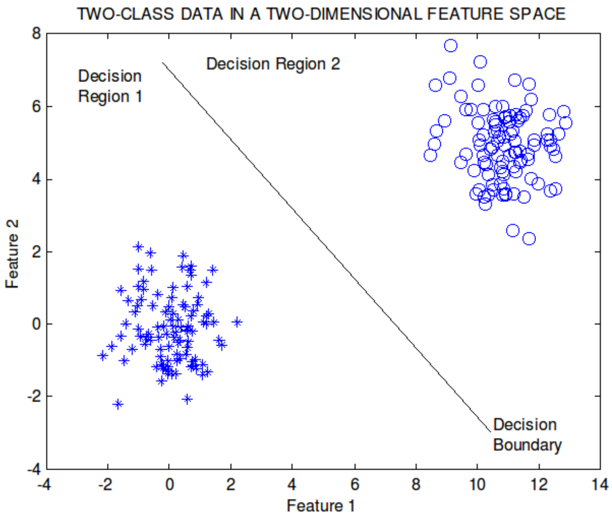
- decision boundaries
- linear classifiers
- Perceptron
- Perceptron algorithm
- classification by Perceptron
- XOR problem

Decision Boundaries

What is a decision boundary?

- Consider a 2-class and 2-feature problem.
  - e.g. classifying male vs. female using height and weight
- Consider the 2-D feature space
  - i.e. the plane showing weight on one axis and height on the other
- A classifier will output class ω_1 in some regions and class ω_2 in others.
- The line separating these regions is the decision boundary.
- The decision boundary is a property of the classifier.

Decision Boundaries

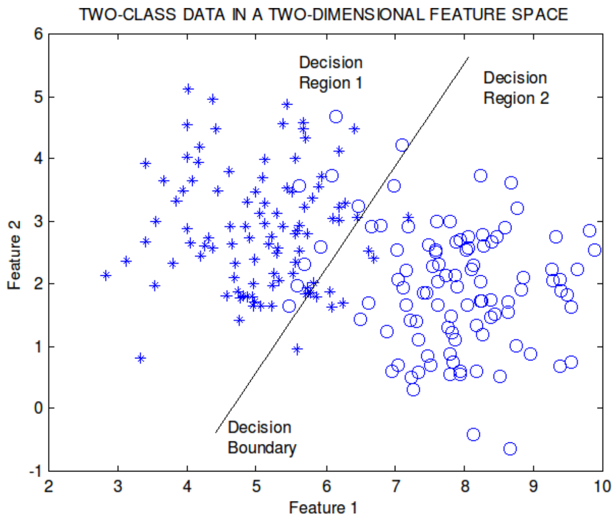


2-class data with a linear decision boundary

Decision Boundaries

- Different types of classifier can draw different types of decision boundary.
- A classifier that can only draw straight line boundaries is called a linear classifier
- If the data sets can be separated by a straight line they are said to be linearly separable
- Linear classifiers can be used to classifier linearly separable data.
- Note,
  - in 2-D a linear decision boundary forms a straight line
  - in 3-D a linear decision boundary forms a flat plane
  - in N-D it forms a flat $(N-1)$-D hyperplane

Decision Boundaries



data that is not linearly separable

Decision Boundaries

- Consider the Bayesian classifiers with $p(x|\omega)$ modelled using Gaussian distributions that we discussed last week.
- Are they linear classifiers?
- To find out we need to examine their decision boundaries.
- For simplicity we will consider the 2-D case.

Video demo: Gaussians.mov

Bayesian Classifiers

- M-class classification task:

$$P(\omega_i|\boldsymbol{x}) - P(\omega_j|\boldsymbol{x}) = 0$$

  is the surface separating the regions R_i and R_j
- define a discriminant function:

$$g_i(\boldsymbol{x}) \equiv f(P(\omega_i|\boldsymbol{x}))$$

  (note) $f(\bullet)$ can be any monotonic function, meaning that

$$P(\omega_i|\boldsymbol{x}) < P(\omega_j|\boldsymbol{x}) \iff g_i(\boldsymbol{x}) < g_j(\boldsymbol{x})$$

- the decision test:

$$\boldsymbol{x} \text{ belongs to } \omega_i \text{ if } g_i(\boldsymbol{x}) > g_j(\boldsymbol{x}) \ \forall j \neq i$$

- the decision boundary between class i and j: $g_i(\boldsymbol{x}) - g_j(\boldsymbol{x}) = 0$

Bayesian Classifiers

Multivariate normal case

- pdf for a class ω_i:

$$p(\boldsymbol{x}|\omega_i) = \frac{1}{(2\pi)^{\frac{l}{2}}|\Sigma_i|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)^T\Sigma_i^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i)\right)$$

  with the mean $\boldsymbol{\mu}_i$ and the covariance Σ_i
- a discriminant function for the Bayesian classifier:

$$\begin{aligned} g_i(\boldsymbol{x}) &= \ln P(\omega_i|\boldsymbol{x}) \\ &= -\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)^T\Sigma_i^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i) + c_i + \ln P(\omega_i) - \ln p(\boldsymbol{x}) \end{aligned}$$

  with some constant c_i

Bayesian Classifiers

- the discriminant function:

$$\begin{aligned} g_i(\boldsymbol{x}) &= \ln P(\omega_i|\boldsymbol{x}) \\ &= \ln \frac{p(\boldsymbol{x}|\omega_i)P(\omega_i)}{p(\boldsymbol{x})} \\ &= \ln p(\boldsymbol{x}|\omega_i) + \ln P(\omega_i) - \ln p(\boldsymbol{x}) \\ &= -\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)^T\Sigma_i^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i) \underbrace{-\frac{l}{2}\ln(2\pi) - \frac{1}{2}\ln|\Sigma_i|}_{c_i} \\ &\quad + \ln P(\omega_i) - \ln p(\boldsymbol{x}) \end{aligned}$$

  - logarithmic function $\ln(\bullet)$ is monotonic
  - can ignore $\ln p(\boldsymbol{x})$ from the calculation
  - can also ignore $\ln P(\omega_i)$ for equiprobable classes

Bayesian Classifiers

Minimum distance classifiers

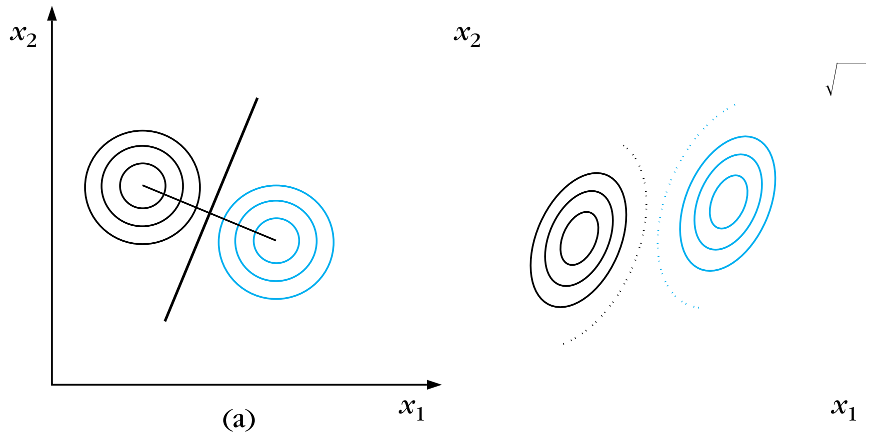
- the discriminant function for equiprobable classes, with the same covariance matrix for each class:

$$g_i(\boldsymbol{x}) = -\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)^T \Sigma^{-1} (\boldsymbol{x}-\boldsymbol{\mu}_i)$$

  - if $\Sigma = \sigma^2 I$: maximum $g_i(\boldsymbol{x})$ implies the minimum Euclidean distance: $d_e = ||\boldsymbol{x}-\boldsymbol{\mu}_i||$
  - non-diagonal Σ: maximum $g_i(\boldsymbol{x})$ implies the minimum Mahalanobis distance:

$$d_m = \left((\boldsymbol{x}-\boldsymbol{\mu}_i)^T \Sigma^{-1} (\boldsymbol{x}-\boldsymbol{\mu}_i)\right)^{\frac{1}{2}}$$

Bayesian Classifiers



curves of (a) equal Euclidean distance and (b) equal Mahalanobis distance

Bayesian Classifiers

Example

- in a two-class two-dimensional classification task, feature vectors are generated by two normal distributions with the mean vectors $\boldsymbol{\mu}_1 = (0,0)^T$ and $\boldsymbol{\mu}_2 = (3,3)^T$, sharing the same covariance matrix

$$\Sigma = \begin{pmatrix} 1.1 & 0.3 \\ 0.3 & 1.9 \end{pmatrix}$$

- Consider the point $(1.0, 2.2)^T$
- $(1.0, 2.2)^T$ is closer to $\boldsymbol{\mu}_2 = (3,3)^T$ using the Euclidean distance

Bayesian Classifiers

Example (continue)

- Mahalanobis distances of $(1.0, 2.2)^T$ from two means:

$$\begin{aligned} d_m^2(\boldsymbol{\mu}_1, \boldsymbol{x}) &= (\boldsymbol{x} - \boldsymbol{\mu}_1)^T \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_1) \\ &= (1.0, 2.2) \begin{pmatrix} 0.95 & -0.15 \\ -0.15 & 0.55 \end{pmatrix} \begin{pmatrix} 1.0 \\ 2.2 \end{pmatrix} = 2.952 \\ d_m^2(\boldsymbol{\mu}_2, \boldsymbol{x}) &= (\boldsymbol{x} - \boldsymbol{\mu}_2)^T \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_2) \\ &= (-2.0, -0.8) \begin{pmatrix} 0.95 & -0.15 \\ -0.15 & 0.55 \end{pmatrix} \begin{pmatrix} -2.0 \\ -0.8 \end{pmatrix} = 3.672 \end{aligned}$$

hence $(1.0, 2.2)^T$ is assigned to the class ω_1

Bayesian Classifiers

Considering the equal covariance case

$$g_j(\boldsymbol{x}) = -\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_j)^T \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_j)$$
$$g_k(\boldsymbol{x}) = -\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_k)^T \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_k)$$

- The decision boundary is given by

$$g_j(\boldsymbol{x}) = g_k(\boldsymbol{x})$$

- The squared terms in $\boldsymbol{x}$ will cancel leaving something of the form

$$\boldsymbol{\mu}_c^T \Sigma^{-1} \boldsymbol{x} - \frac{1}{2}\boldsymbol{\mu}_c^T \Sigma^{-1} \boldsymbol{\mu}_c + C = 0$$

- This is the equation of a straight line - i.e. when the covariances are equal the decision boundaries become linear

16 / 23

Linear Classifiers

A linear classifier

- a linear classifier is a mapping which partitions feature space using a linear function (a straight line, or a hyperplane)
- it is one of the simplest classifiers we can imagine
  - in the 2-dimensional feature space the decision boundary, separating the two classes, is a straight line

17 / 23

Linear Classifiers

- a 2-class task with ω_1 and ω_2 in the l-dimensional feature space
- decision hyperplane

$$g(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x} + w_0 = 0$$

where

$$\begin{aligned} \boldsymbol{x} = \{x_1, x_2, \ldots, x_l\}^T & \quad \text{feature vector} \\ \boldsymbol{w} = \{w_1, w_2, \ldots, w_l\}^T & \quad \text{weight vector} \\ w_0 & \quad \text{threshold} \end{aligned}$$

(note) $\boldsymbol{w}^T \boldsymbol{x} = w_1 x_1 + w_2 x_2 + \ldots + w_l x_l$

Linear Classifiers

- e.g. for a 2-D problem
- decision boundary is a straight line defined by

$$g(\boldsymbol{x}) = w_1 x_1 + w_2 x_2 + w_0 = 0$$

i.e.

$$x_2 = -\frac{w_1}{w_2} x_1 - \frac{w_0}{w_2}$$

- How many parameters?
  - 2-D 2-class Gaussian classifier? (3 + 2) x 2 + 1 = 11
  - 2-D 2-class linear classifier? 3 ... really 2?

Linear Classifiers

- suppose that $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are on the plane $g(\boldsymbol{x}) = 0$:

$$\boldsymbol{w}^T(\boldsymbol{x}_1 - \boldsymbol{x}_2) = 0 \quad \forall \boldsymbol{x}_1, \boldsymbol{x}_2$$

- the weight vector $\boldsymbol{w}$ is orthogonal to the decision hyperplane because the difference $\boldsymbol{x}_1 - \boldsymbol{x}_2$ lies on the plane

Linear Classifiers

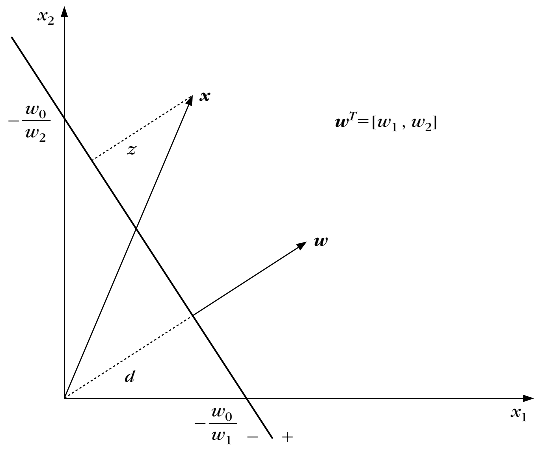


figure shows case where $w_1 > 0; \quad w_2 > 0; \quad w_0 < 0$

Linear Classifiers

in the figure

$$d = \frac{|w_0|}{||\boldsymbol{w}||}; \quad z = \frac{|g(\boldsymbol{x})|}{||\boldsymbol{w}||}$$

- $||\boldsymbol{w}|| = \sqrt{w_1^2 + w_2^2}$ in the 2-dimensional case
- $|g(\boldsymbol{x})|$ is a measure of the Euclidean distance of the point $\boldsymbol{x} = \{x_1, x_2\}$ from the plane
- on one side of the plane $g(\boldsymbol{x})$ takes positive values; negative on the other side
- the plane passes through the origin when $w_0 = 0$

(note) $||\bullet||$ norm; $|\bullet|$ absolute value, determinant

22 / 23

Summary

23 / 23

Summary

- □ Linear classifiers have linear decision boundaries
- □ We can analyse decision boundaries using discriminant functions
- □ Gaussian classifiers aren't in general linear...
  - ... but they become so when all class share the same covariance matrix
- □ A linear classifier can be expressed using an vector orthogonal to the boundary and an offset
  - ... 3 parameters, but only really 2 free parameters
- □ Next lecture we'll see ways of learning the parameters from labelled data.

23 / 23