EML'24 – Lecture 4
Classification I

ISLR 4, ESL 4

Prof. Isabel Valera
31 October 2024

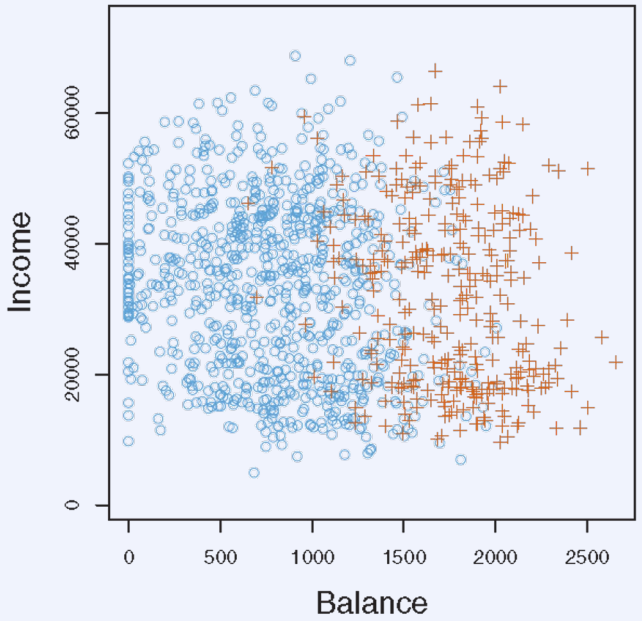
SIC Saarland Informatics Campus

Classification Overview

In classification, we want to predict categorical outputs

Example will someone pay back their loan? yes or no?

- inputs: annual **income**, monthly **balance**, **student** status



Classification Overview

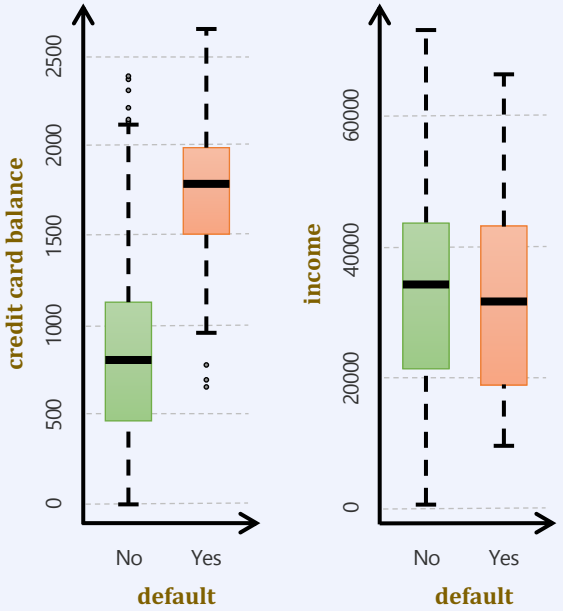
In classification, we want to predict categorical outputs

Example will someone pay back their loan? yes or no?

- inputs: annual **income**, monthly **balance**, **student** status

More examples

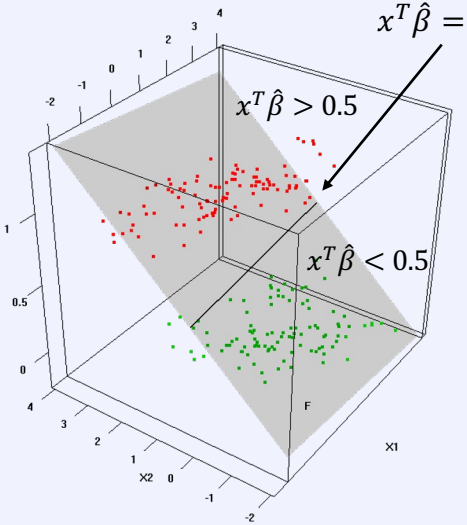
- identify whether an email is a spam email
- classify which out of k diseases a patient has given symptoms
- decide whether a transaction is fraudulent based on transaction history, location, IP, DNS, etc.
- identify disease-causing mutations based on DNA sequences from patients with and without a given disease (feature selection)

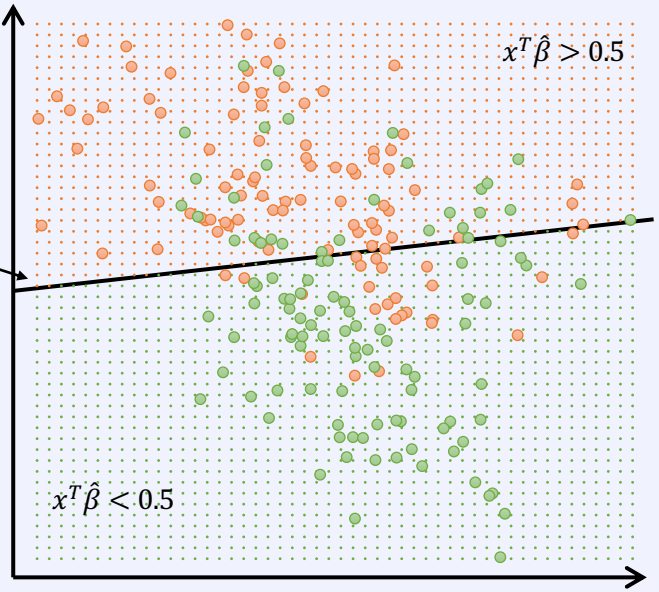


Why not just do linear regression?

Linear regression can actually work for binary classification

- simply code $Y = \begin{cases} 0 & \text{if } \textbf{green} \\ 1 & \text{if } \textbf{red} \end{cases}$





Why not just do linear regression?

Linear regression does not generalize to more than two classes

For example, which coding when we have three classes?

- $Y = \begin{cases} 0 & \text{if green} \\ 1 & \text{if red} \\ 2 & \text{if blue} \end{cases}$ or $Y = \begin{cases} 0 & \text{if red} \\ 1 & \text{if blue} \\ 2 & \text{if green} \end{cases}$ or $Y = \begin{cases} 0 & \text{if red} \\ 3 & \text{if blue} \\ 9 & \text{if green} \end{cases}$?
- each imposes a different ordering, and different distances between classes

A regression model tries to respect the ordering and numbers representing the classes

- unless we know that the labels are metric, we should not impose one as this introduces undue bias
- also, for more than two classes linear-regression has a problem called masking (ESL page 105)

Logistic Regression

Example Credit **default** data

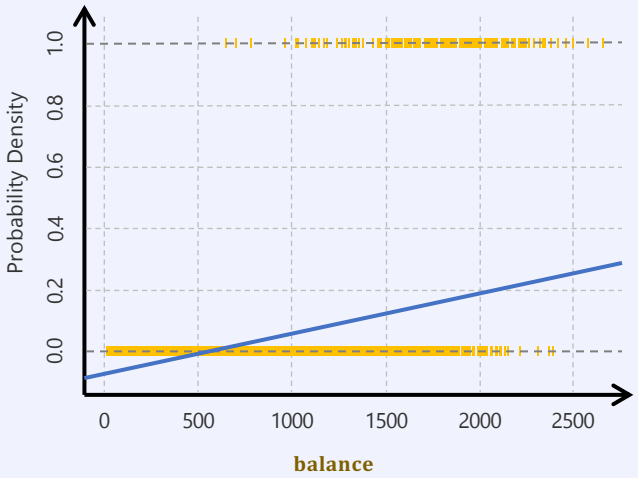
- univariate model, e.g.

$$\Pr(\textbf{default} = \text{yes} \mid \textbf{balance})$$

- simple linear regression models this as

$$f(X) = \beta_0 + \beta_1 X$$

- which leads to values outside $[0,1]$



Logistic Regression

Example Credit **default** data

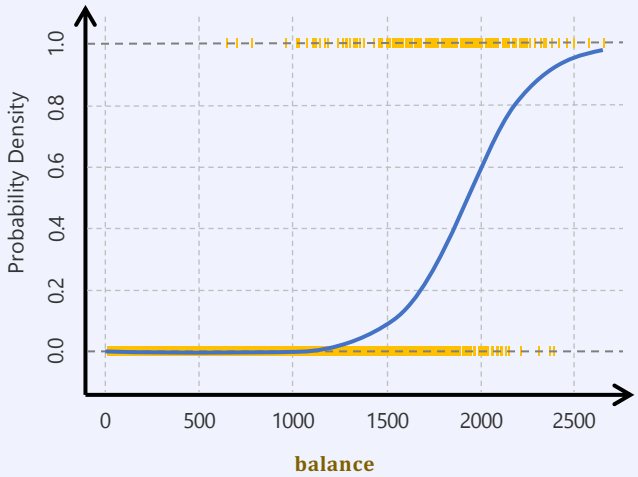
- univariate model, e.g.

$$\Pr(\textbf{default} = \text{yes} \mid \textbf{balance})$$

- simple linear regression models this as

$$f(X) = \beta_0 + \beta_1 X$$

- which leads to values outside $[0,1]$



We can map these into $[0,1]$ using the logistic function

$$p(Y = 1|X) = \frac{e^{f(X)}}{1 + e^{f(X)}} = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}$$

probability that $Y = \textbf{yes} = 1$

Logistic Regression

Example Credit **default** data

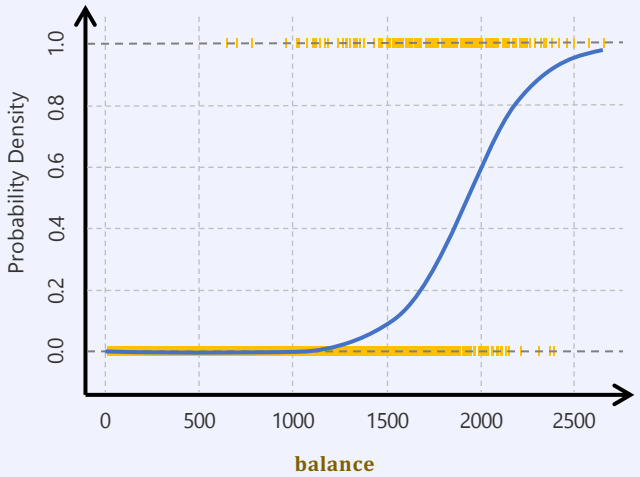
- univariate model, e.g.

$$\Pr(\textbf{default} = \text{yes} \mid \textbf{balance})$$

- simple linear regression models this as

$$f(X) = \beta_0 + \beta_1 X_1$$

- which leads to values outside $[0,1]$



We can map these into $[0,1]$ using the logistic function

probability that $Y = \textbf{yes} = 1$ ⟶

$$p(Y = 1|X) = \frac{e^{f(X)}}{1 + e^{f(X)}} = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}$$

- not only are all values now sensible, we also have the

odds ratio as $\frac{p(Y=1|X)}{1-p(Y=1|X)} = e^{\beta_0 + \beta_1 X}$, and the log-odds(logit) as $\log\left(\frac{p(Y=1|X)}{1-p(Y=1|X)}\right) = \beta_0 + \beta_1 X$

Interpreting a Logistic Model

If we increase X by one unit, we

- add β_1 to the log-odds
- multiply the odds by e^{β_1}

$$\log\left(\frac{p(Y=1|X)}{1-p(Y=1|X)}\right) = \beta_0 + \beta_1 X$$

$$\frac{p(Y=1|X)}{1-p(Y=1|X)} = e^{\beta_0+\beta_1 X}$$

Effect on $p(X)$ is non-linear

- if $\beta_1 > 0$, adding X increases $p(Y=1|X)$
- if $\beta_1 < 0$, adding X decreases $p(Y=1|X)$

$$p(Y=1|X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}$$

Estimating the Coefficients of Logistic Regression

Maximum Likelihood

- generative approach to find the model that is least surprised to see the given data
- the (conditional) likelihood function to maximize is

$$p(y_1, y_2, \dots, y_n \mid \beta_0, \beta_1, x_1, x_2, \dots, x_n) = \prod_{i:y_i=1} p(y_i = 1 \mid x_i) \prod_{i:y_i=0} (1 - p(y_i = 1 \mid x_i))$$

- equivalent, but often more practical, is to maximize the log-likelihood

$$\ell(\beta_0, \beta_1) = \sum_{i:y_i=1} \log p(y_i = 1 \mid x_i) + \sum_{i:y_i=0} \log(1 - p(y_i = 1 \mid x_i))$$

- equivalent, is to minimize the negative log-likelihood (NLL)

We can maximize the likelihood function using nonlinear gradient-descent (Newton-Raphson)

- the intercept only adjusts the average of the fitted probabilities to the proportions of 1s in the data
- in each step we do linear regression, and can hence apply all types of linear model analysis we know

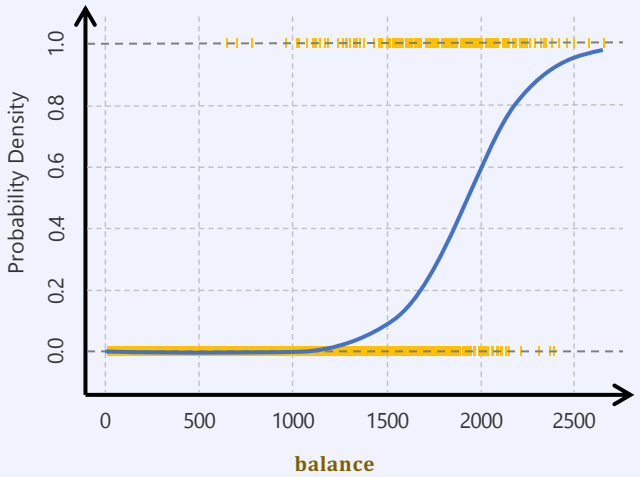
Example Single Continuous Predictor

Probabilities of **default** given **balance**

$$\hat{p}(\textbf{default}|1000) = \frac{e^{-10.6513+0.0055\times1000}}{1+e^{-10.6513+0.0055\times1000}} = 0.00576$$

$$\hat{p}(\textbf{default}|\,2000) = \frac{e^{-10.6513+0.0055\times2000}}{1+e^{-10.6513+0.0055\times2000}} = 0.586$$

- if we increase **balance** by 1 EUR, this
- increases the log odds of defaulting by $\mathbf{0.0055}$
- multiplies the odds of defaulting by $e^{0.0055} = 1.0055\%$



	Coefficient	Std. error	Z-statistic	p-value
intercept	-10.653	0.3612	-29.5	<0.0001
balance	0.0055	0.0002	24.9	<0.0001

Example Single Binary Predictor

Probabilities of **default** given **student**

$$\hat{p}(\textbf{default}|\ \textbf{student} = \text{yes}) = \frac{e^{-3.5041+0.40409\times1}}{1+e^{-3.5041+0.40409\times1}} = 0.00431$$

$$\hat{p}(\textbf{default}\ |\ \textbf{student} = \text{no}) = \frac{e^{-3.5041+0.40409\times0}}{1+e^{-3.5041+0.40409\times0}} = 0.00292$$

	Coefficient	Std. error	Z-statistic	p-value
intercept	-3.5041	0.0707	-49.55	<0.0001
student	0.4049	0.1150	3.52	0.0004

Multiple Logistic Regression

The multivariate logistic regression model is defined as

- $\log\left(\frac{p(Y=1|X)}{1-p(Y=1|X)}\right) = \beta_0 + \beta_1 X + \cdots \beta_p X_p$ with $\quad p(Y=1|X) = \frac{e^{\beta_0+\beta_1 X+\cdots+\beta_p X_p}}{1+e^{\beta_0+\beta_1 X+\cdots+\beta_p X_p}}$

Example predicting **default** based on **balance**, **income**, and **student**

$$\hat{p}(\textbf{default} \mid \textbf{student} = \text{yes}, \textbf{balance} = 1{,}500, \textbf{income} = 40) = \frac{e^{-10.869+0.00574\times1{,}500+0.003\times40-0.6468\times1}}{1+e^{-10.869+0.00574\times1{,}500+0.003\times40-0.6468\times1}} = 0.058$$

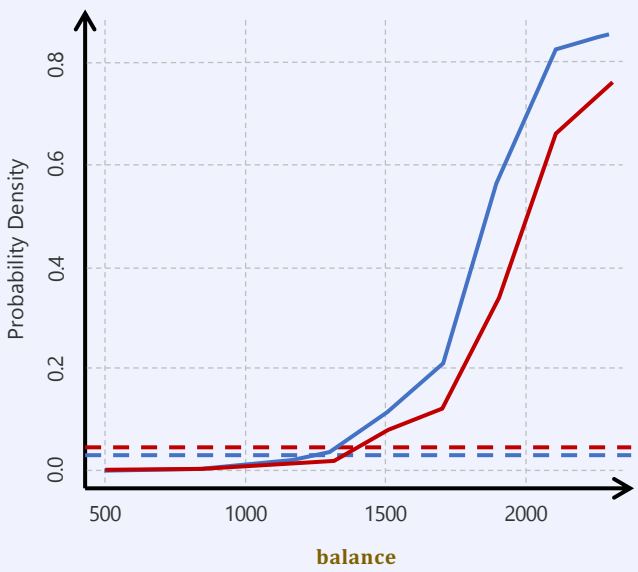
$$\hat{p}(\textbf{default} \mid \textbf{student} = \text{no}, \textbf{balance} = 1{,}500, \textbf{income} = 40) = \frac{e^{-10.869+0.00574\times1{,}500+0.003\times40-0.6468\times0}}{1+e^{-10.869+0.00574\times1{,}500+0.003\times40-0.6468\times0}} = 0.105$$

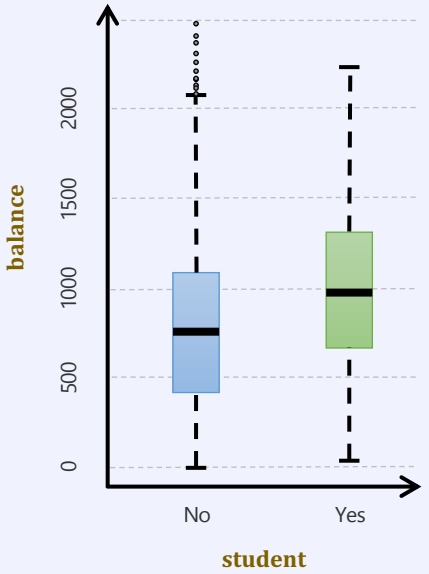
	Coefficient	Std. error	Z-statistic	p-value
intercept	-10.8690	0.4923	-22.08	<0.0001
balance	0.0057	0.0002	24.74	<0.0001
income	0.0030	0.0082	0.37	0.7115
student [yes]	-0.6468	0.2362	-2.74	0.0062

Example Confounding in Logistic Regression

Why is the **student** coefficient positive in the univariate and negative in the multivariate model?

- confounding!
- students have higher **balance**
- students **default** at higher **balance**
- for a fixed value of **balance** and **income**, a **student** is less likely to default than a nonstudent!





average default rate nonstudent
average default rate student
nonstudent
student

The Sound of Machine Learning that is just

logistic regression…

(credits @MaartenvSmeden)

Classification Discriminative vs. Generative

	Discriminative	Generative
Output for an input x	estimate $\hat{g}(x)$ of class $g(x)$	probability distribution $\{p_g(x) \mid g \in G\}$, $p_g(x)$ is the probability that x belongs to class g
Main idea	the classifier returns an estimate of the output, which discriminates between different classes	the classifier generates the output with some probability
Performance measure	loss function that measures the deviation between estimate and output, e.g. 0-1 loss	(log-)likelihood of the estimator generating the output $\sum_{i=1}^{n} \log p_{g_i}(x)$
Optimization problem	Minimize the loss function	Maximize the likelihood

Bayes Classifier

Bayesian Methods

- Bayes' formula

Probability of the input, given the output, i.e. class density

Posterior (probability of the output given the input) → $\Pr(Y \mid X) = \dfrac{\Pr(X \mid Y)\Pr(Y)}{\Pr(X)}$ ← Prior probability of the output

Prior probability of the input

- $\Pr(X)$ is a normalizing constant that only depends on the input data and often need not be computed

Bayes classifier for K classes

- use Bayes' formula to determine posterior density per class $\Pr(Y = k \mid X = x)$

$$p_k(x) = \Pr(Y = k \mid X = x) = \frac{\pi_k f_k(x)}{\sum_{\ell=1}^{K} \pi_\ell f_\ell(x)}$$

class density

- we compute $p_k(x)$ by estimating the class prior probabilities π_k and the class densities $f_k(X)$
- we estimate the prior class probabilities from data, $\pi_k = \frac{1}{n}\sum_{i=1}^{n} I(y_i = k)$
- we need to determine the probability density for point x for a class k
- we then classify each point to its most probable class

The Bayes classifier - example

Model **assumptions**

- every class is Gaussian-distributed

$$f_k(x) = \frac{1}{\sqrt{2\pi}\sigma_k}\exp\left(-\frac{1}{2\sigma_k^2}(x-\mu_k)^2\right)$$

- all classes have the same variance

$$\sigma_1^2 = \sigma_2^2 = \cdots = \sigma_k^2 = \sigma^2$$

The **Bayesian classifier** now becomes

$$p_k(x) = \frac{\pi_k f_k(x)}{\sum_{\ell=1}^{K}\pi_\ell f_\ell(x)} = \frac{\pi_k \frac{1}{\sqrt{2\pi}\sigma_k}\exp\left(-\frac{1}{2\sigma_k^2}(x-\mu_k)^2\right)}{\sum_{l=1}^{K}\pi_l \frac{1}{\sqrt{2\pi}\sigma^2}\exp\left(-\frac{1}{2\sigma^2}(x-\mu_l)^2\right)}$$

- the logarithm of the numerator

$$-\frac{x^2}{2\sigma^2} + x\cdot\frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log\pi_k - \log\left(\sqrt{2\pi}\sigma\right)$$

Linear Discriminant Analysis

The Bayes-optimal choice is to classify $\boldsymbol{x}$ to the class with the largest discriminant

- the discriminant of a class $\boldsymbol{k}$ is the log-probability that cancels in the log-odds

$$\log\left(\frac{p_k(x)}{p_l(x)}\right) = \delta_k(x) - \delta_l(x)$$

- where

$$\delta_k(x) = x \cdot \frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log \pi_k$$

  is the log-numerator from previous slide with the class-independent terms removed

Example Linear Discriminant Analysis

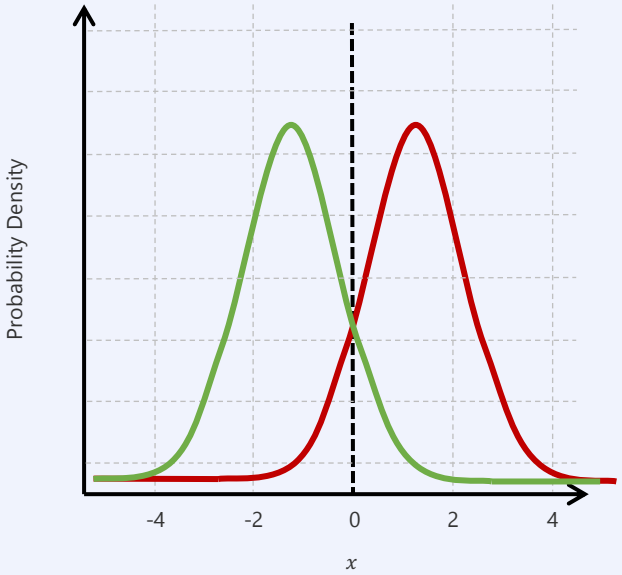
If $\pi_1 = \pi_2$ we classify an observation x to class 1 if

$$2x(\mu_1 - \mu_2) > \mu_1^2 - \mu_2^2$$

- the Bayes decision boundary is the set of points for which both discriminants are equal, i.e.

$$x = \frac{\mu_1^2 - \mu_2^2}{2(\mu_1 - \mu_2)} = \frac{\mu_1 + \mu_2}{2}$$

- the figure shows two 1D normal density functions.
- the dashed line represents the Bayes decision boundary, at which an observation is equally likely to belong to either class



$\mu_1 = -1.25$
$\mu_2 = 1.25$
$\sigma_1 = \sigma_2 = 1$

Multivariate LDA

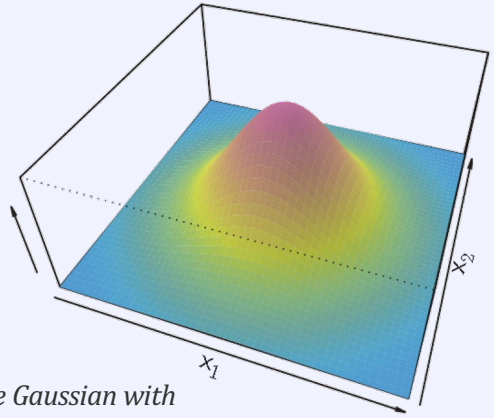
Model assumptions

- each class is a multivariate Gaussian
- the covariance matrix is the same for all classes

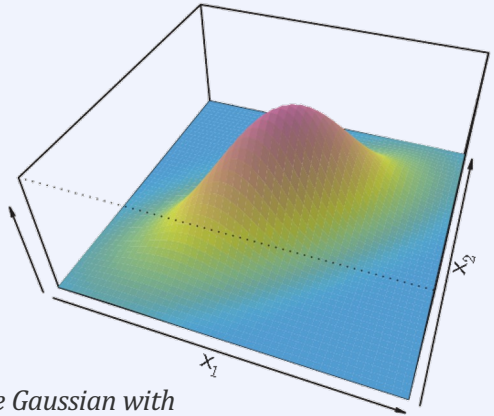
$$f_{k(x)} = \frac{1}{(2\pi)^{\frac{p}{2}}|\mathbf{\Sigma}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x-\mu_k)^T\mathbf{\Sigma}^{-1}(x-\mu_k)\right)$$

$$\delta_k(x) = x^T\mathbf{\Sigma}^{-1}\mu_k - \frac{1}{2}\mu_k^T\mathbf{\Sigma}^{-1}\mu_k + \log\pi_k$$

- $\mathbf{\Sigma}$ is the $p \times p$ covariance matrix of the inputs $\mathbf{\Sigma} = \mathrm{Cov}(x)$



Multivariate Gaussian with two uncorrelated predictors



Multivariate Gaussian with two correlated predictors (0.7)

Multivariate LDA

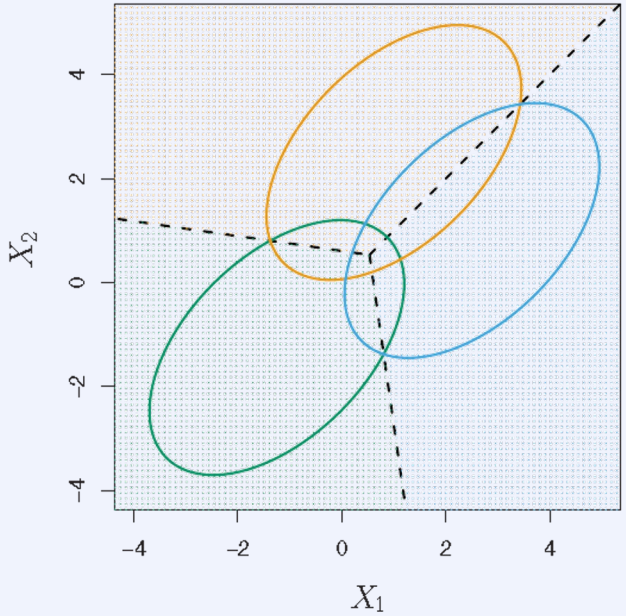
Model assumptions

- each class is a multivariate Gaussian
- the covariance matrix is the same for all classes

$$f_{k(x)} = \frac{1}{(2\pi)^{\frac{p}{2}}|\mathbf{\Sigma}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x-\mu_k)^T\mathbf{\Sigma}^{-1}(x-\mu_k)\right)$$

$$\delta_k(x) = x^T\mathbf{\Sigma}^{-1}\mu_k - \frac{1}{2}\mu_k^T\mathbf{\Sigma}^{-1}\mu_k + \log\pi_k$$

- $\mathbf{\Sigma}$ is the $p \times p$ covariance matrix of the inputs $\mathbf{\Sigma} = \mathrm{Cov}(x)$



2D synthetic data example with three classes. Ellipses contain 95% of the class probability mass, the Bayes decision boundaries are dashed

Multivariate LDA

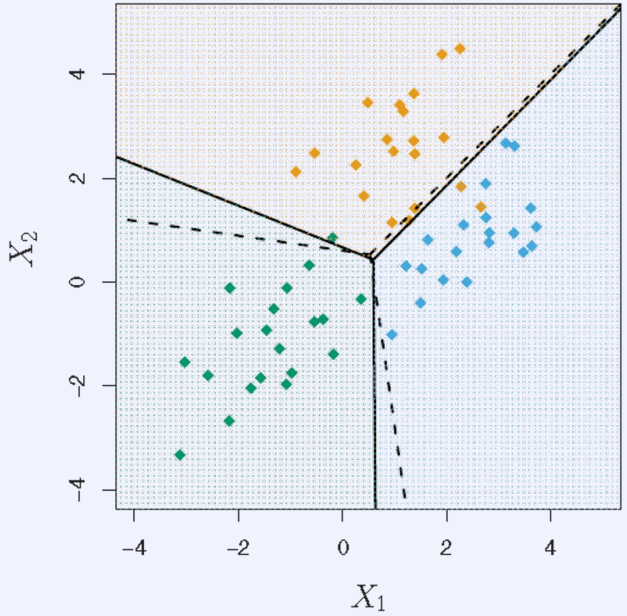
Model assumptions

- each class is a multivariate Gaussian
- the covariance matrix is the same for all classes

$$f_{k(x)} = \frac{1}{(2\pi)^{\frac{p}{2}}|\mathbf{\Sigma}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x-\mu_k)^T\mathbf{\Sigma}^{-1}(x-\mu_k)\right)$$

$$\delta_k(x) = x^T\mathbf{\Sigma}^{-1}\mu_k - \frac{1}{2}\mu_k^T\mathbf{\Sigma}^{-1}\mu_k + \log\pi_k$$

- $\mathbf{\Sigma}$ is the $p \times p$ covariance matrix of the inputs $\mathbf{\Sigma} = \mathbf{Cov}(x)$
- model is fitted using sample estimates similar to the 1D case
- μ easy, but $\mathbf{\Sigma}$ is the hardest to estimate



LDA fit of data set comprising 20 samples from each class, decision boundary in black

Quadratic Discriminant Analysis (QDA)

In QDA every class has its own covariance matrix

$$f_{k(x)} = \frac{1}{(2\pi)^{\frac{p}{2}}|\mathbf{\Sigma}_k|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x-\mu_k)^T\mathbf{\Sigma}_k^{-1}(x-\mu_k)\right)$$

$$\delta_k(x) = -\frac{1}{2}x^T\mathbf{\Sigma}_k^{-1}x + x^T\mathbf{\Sigma}_k^{-1}\mu_k - \frac{1}{2}\mu_k^T\mathbf{\Sigma}_k^{-1}\mu_k + \log\pi_k$$

- class boundaries are now quadratic curves
- we fit a different covariance matrix estimate per class
- LDA has $(2K+p+1)p/2$ parameters,
- QDA has $Kp(p+3)/2$ parameters

Example

- for $p = 4, K = 2$, LDA has 18 parameters, QDA has 28 parameters
- for $p = 8, K = 2$, LDA has 52 parameters, QDA has 88 parameters

Quadratic Discriminant Analysis (QDA)

We give up the assumption that the covariances of all classes are all the same

For QDA we have

$$f_{k(x)} = \frac{1}{(2\pi)^{\frac{p}{2}}|\boldsymbol{\Sigma}_k|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x-\mu_k)^T\boldsymbol{\Sigma}_k^{-1}(x-\mu_k)\right)$$

$$\delta_k(x) = -\frac{1}{2}x^T\boldsymbol{\Sigma}_k^{-1}x + x^T\boldsymbol{\Sigma}_k^{-1}\mu_k - \frac{1}{2}\mu_k^T\boldsymbol{\Sigma}_k^{-1}\mu_k + \log\pi_k$$

For LDA we had

$$f_{k(x)} = \frac{1}{(2\pi)^{\frac{p}{2}}|\boldsymbol{\Sigma}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x-\mu_k)^T\boldsymbol{\Sigma}^{-1}(x-\mu_k)\right)$$

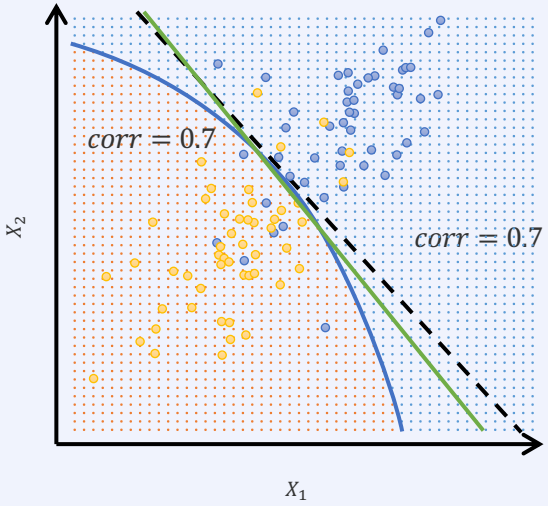
$$\delta_k(x) = x^T\boldsymbol{\Sigma}^{-1}\mu_k - \frac{1}{2}\mu_k^T\boldsymbol{\Sigma}^{-1}\mu_k + \log\pi_k$$

V

Example LDA vs. QDA

Two-class problem with $\Sigma_1 = \Sigma_2$

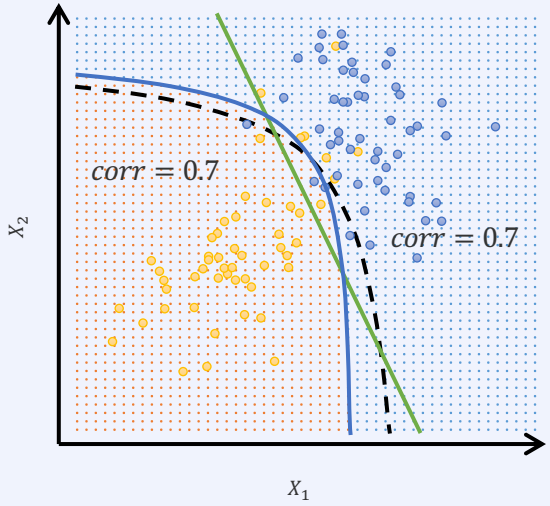
QDA overtrains



Bayes decision boundary
LDA decision boundary
QDA decision boundary

Two-class problem with $\Sigma_1 \neq \Sigma_2$

LDA overtrains



Bayes decision boundary
LDA decision boundary
QDA decision boundary