Lecture 2: Model-based classification

Felix Held, Mathematical Sciences

MSA220/MVE440 Statistical Learning for Big Data

28th March 2019

CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

Reprise: Statistical Learning (I)

Regression

- Theoretically best regression function for squared error loss

$$\widehat{f}(\mathbf{x}) = \mathbb{E}_{p(y|\mathbf{x})}[y]$$

- Approximate (1) or make model-assumptions (2)
  1. k-nearest neighbour regression

$$\mathbb{E}_{p(y|\mathbf{x})}[y] \approx \frac{1}{k} \sum_{\mathbf{x}_{i_l} \in N_k(\mathbf{x})} y_{i_l}$$

  2. linear regression (viewpoint: generalized linear models (GLM))

$$\mathbb{E}_{p(y|\mathbf{x})}[y] \approx \mathbf{x}^T \boldsymbol{\beta}$$

Reprise: Statistical Learning (II)

Classification

- Theoretically best classification rule for 0-1 loss and K possible classes

$$\hat{c}(\mathbf{x}) = \underset{1 \leq i \leq K}{\arg\max}\, p(i|\mathbf{x})$$

- Approximate (1) or make model-assumptions (2)
  1. k-nearest neighbour classification

$$p(i|\mathbf{x}) \approx \frac{1}{k} \sum_{\mathbf{x}_l \in N_k(\mathbf{x})} \mathbb{1}(i_l = i)$$

  2. Instead of approximating $p(i|\mathbf{x})$ from data, can we make sensible model assumptions instead?

Amendment: kNN methods

There are two choices to make when implementing a kNN method

1. The metric to determine a neighbourhood
   - e.g. Euclidean/ℓ_2 norm, Manhattan/ℓ_1 norm, max norm, ...
2. The number of neighbours, i.e. k

The choice of metric changes the underlying local model of the method while k is a tuning parameter.

Model-based classification

Classification as regression

- Consider a two-class problem, with $i_l = 0$ or $i_l = 1$
- Instead of 0-1 loss, use square error loss, i.e.

$$\mathbb{E}_{p(i|\mathbf{x})}[i] = 0 \cdot p(0|\mathbf{x}) + 1 \cdot p(1|\mathbf{x}) = p(1|\mathbf{x})$$

  Note that i has a discrete distribution.
- Linear regression model assumption

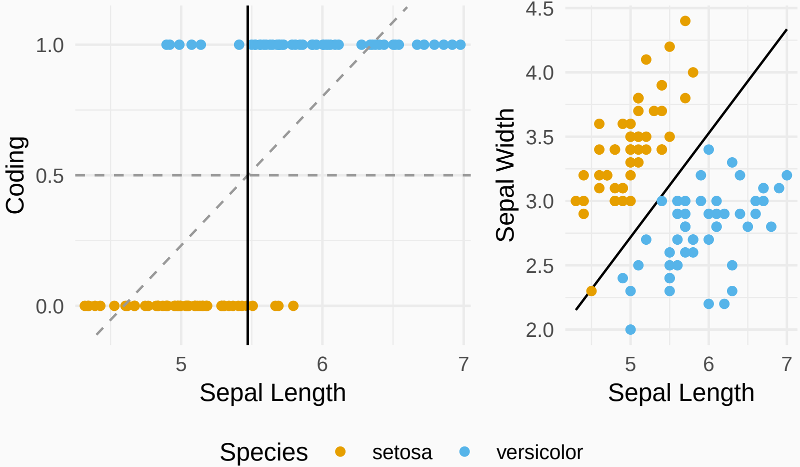
$$p(1|\mathbf{x}) = \mathbb{E}_{p(i|\mathbf{x})}[i] \approx \mathbf{x}^T \boldsymbol{\beta}$$

- Since we are approximating $p(1|\mathbf{x})$ and $p(0|\mathbf{x}) = 1 - p(1|\mathbf{x}) \approx 1 - \mathbf{x}^T \boldsymbol{\beta}$, we indirectly specified a model approximation for Bayes' rule as well

$$c(\mathbf{x}) = \begin{cases} 0 & \mathbf{x}^T \boldsymbol{\beta} \leq \frac{1}{2} \\ 1 & \text{otherwise} \end{cases}$$

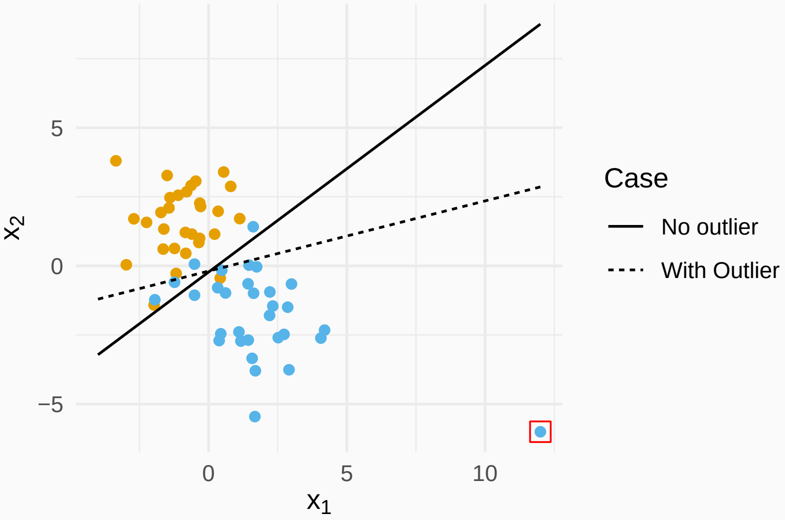
  Note that $\mathbf{x}^T \boldsymbol{\beta} = \frac{1}{2}$ defines the decision boundary

0-1 regression



The solid black lines show the **decision boundary**.

0-1 regressions and outliers



Dummy encoding for categorical variables

In regression, when a predictor x is **categorical**, i.e. takes one of K values, it is common to use a **dummy encoding**.

Example:

$$x = 1 \to z = (1, 0, 0)$$
$$x = 2 \to z = (0, 1, 0)$$
$$x = 3 \to z = (0, 0, 1)$$

Idea

Turn a classification problem into a regression problem by representing the class outcomes i_l in the training data $(i_l, \mathbf{x}_l)$ as vectors in dummy encoding.

Multiple classes

- This creates a sequence of 0-1 regressions (see blackboard). If there are K classes then

$$z_l^{(1)} := \mathbb{1}(i_l = 1) \to p(z^{(1)} = 1|\mathbf{x}) \approx \mathbf{x}^T \boldsymbol{\beta}^{(1)}$$
$$\vdots$$
$$z_l^{(K)} := \mathbb{1}(i_l = K) \to p(z^{(K)} = 1|\mathbf{x}) \approx \mathbf{x}^T \boldsymbol{\beta}^{(K)}$$

- Note that

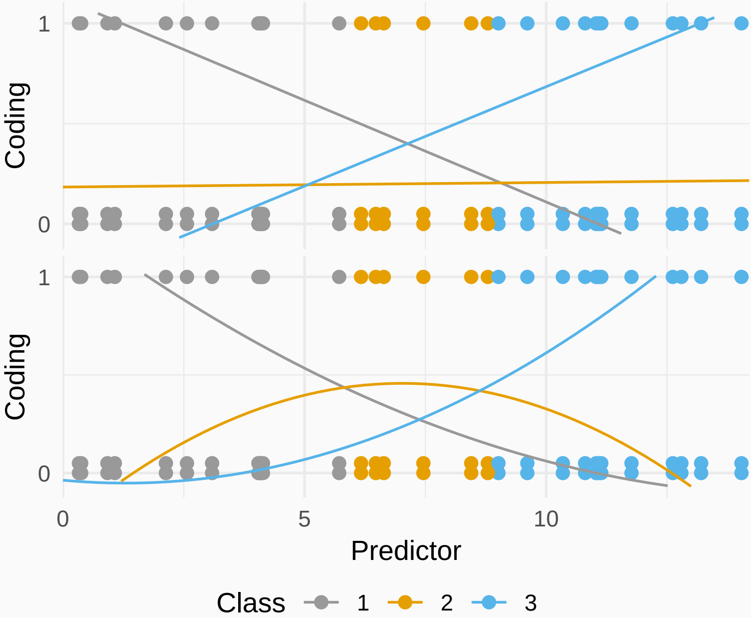
$$p(i|\mathbf{x}) = p(z^{(i)} = 1|\mathbf{x}) \approx \mathbf{x}^T \boldsymbol{\beta}^{(i)}$$

- Classification rule

$$c(x) = \underset{1 \le i \le K}{\arg\max}\, p(i|\mathbf{x}) \approx \underset{1 \le i \le K}{\arg\max}\, \mathbf{x}^T \boldsymbol{\beta}^{(i)}$$

  Decision boundaries are defined by $c(x) = \mathbf{x}^T \beta^{(i)} = \mathbf{x}^T \beta^{(j)}$ for $i \neq j$

Multiple 0-1 regressions



Problems with 0-1 regression

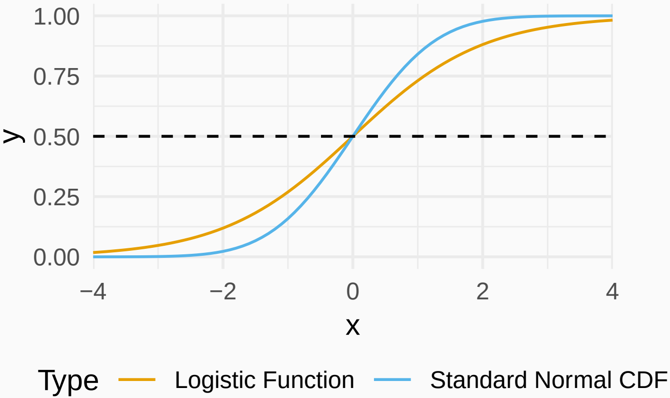
Observations:

1. $\mathbf{x}^T\boldsymbol{\beta}$ is unbounded but models a probability $p(i|\mathbf{x}) \in [0,1]$
2. Only values of $\mathbf{x}^T\boldsymbol{\beta}$ around 0.5 (for binary classification) or close to the maximal value (for multiple classes) are really of interest.
3. Sensitive to points far away from the boundary (outliers)
4. **Masking:** Classes can get buried among other classes (adding polynomial predictors can sometimes help, but this is arbitrary and data dependent)

Inspiration from GLM

Can we transform $\mathbf{x}^T\boldsymbol{\beta}$ such that the transformed values are in $[0,1]$, are similar to the original values when close to 0.5 and insensitive outliers far away from the boundary?

Logistic function and Normal Distribution CDF



Logistic (sigmoid) function

$$\sigma(x) = \frac{\exp(x)}{1 + \exp(x)}$$

Standard Normal CDF

$$\Phi(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{z^2}{2}\right) \mathrm{d}z$$

Logistic and probit regression

- We arrive at **logistic regression** when assuming

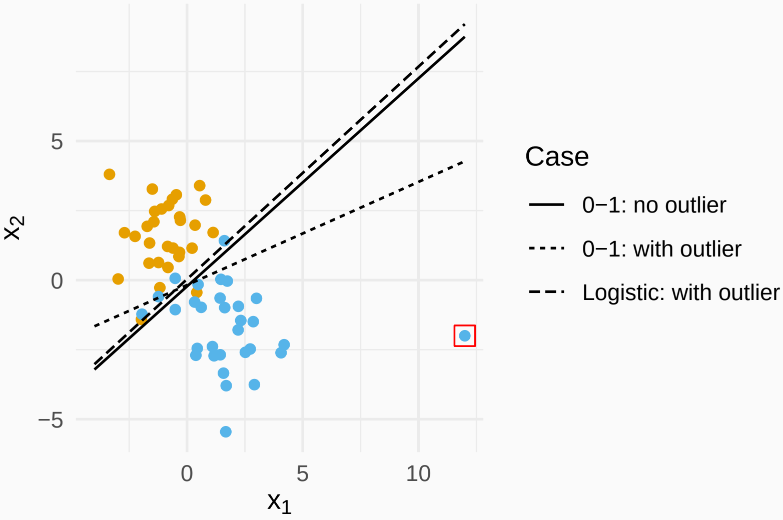
  $$p(1|\mathbf{x}) = \mathbb{E}_{p(i|\mathbf{x})}[i] = \sigma^{-1}\left(\mathbf{x}^T \boldsymbol{\beta}\right)$$

  or **probit regression** when assuming

  $$p(1|\mathbf{x}) = \mathbb{E}_{p(i|\mathbf{x})}[i] = \Phi^{-1}\left(\mathbf{x}^T \boldsymbol{\beta}\right)$$

- Parameters can be estimated by **iteratively reweighted least squares** (Details in ESL Ch. 4.4.1)
- **A warning:** Problematic situation in two-class case (occurs seldom in practice)
  - Assume two classes can be separated perfectly in one or more predictors
  - Logistic regression tries to fit a step-like function, which forces the intercept to $-\infty$ and the corresponding predictor coefficient to $+\infty$.

Logistic regression and outliers



Multi-class logistic regression

- In case of $K > 2$ classes, using dummy encoding for the outcome leads again to a series of regression problems.
- **Requirement:** Probabilities should be modelled, i.e. in $p(i|\mathbf{x}) \in [0,1]$ for each class and $\sum_i p(i|\mathbf{x}) = 1$
- **Softmax function:** $\boldsymbol{\sigma} : \mathbb{R}^K \mapsto [0,1]^K$

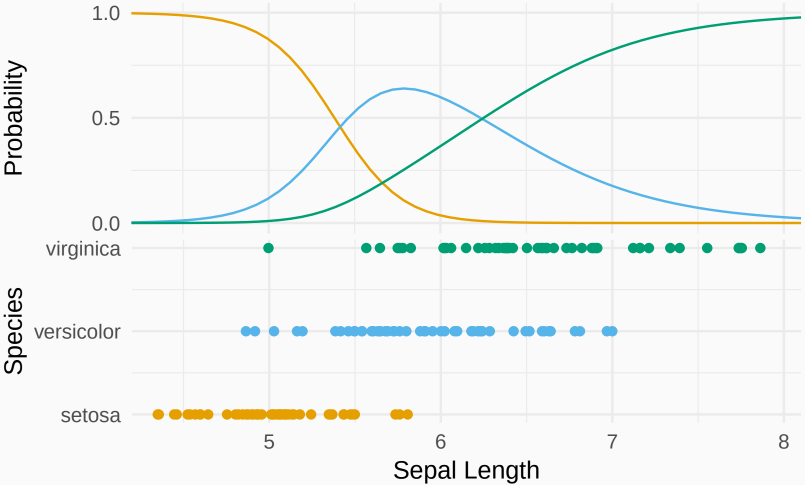
$$\boldsymbol{\sigma}_j(\mathbf{z}) = \frac{e^{z_j}}{\sum_{l=1}^{K} e^{z_l}} \quad \Leftrightarrow \quad \boldsymbol{\sigma}_j(\mathbf{z}) = \frac{e^{(z_j - z_K)}}{1 + \sum_{l=1}^{K-1} e^{(z_l - z_K)}}$$

- Model now:

$$p(i|\mathbf{x}) = \frac{e^{\mathbf{x}^T \boldsymbol{\beta}^{(i)}}}{\sum_{l=1}^{K} e^{\mathbf{x}^T \boldsymbol{\beta}^{(i)}}} \quad \text{or} \quad p(i|\mathbf{x}) = \frac{e^{\mathbf{x}^T (\boldsymbol{\beta}^{(l)} - \boldsymbol{\beta}^{(K)})}}{1 + \sum_{l=1}^{K-1} e^{\mathbf{x}^T (\boldsymbol{\beta}^{(l)} - \boldsymbol{\beta}^{(K)})}}$$

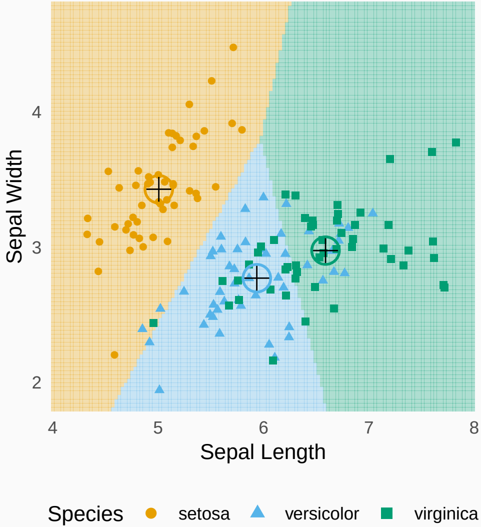
- This method has many names: softmax regression, multinomial logistic regression, maximum entropy classifier, ...

Multi-class logistic regression: An example



Classification with focus on the feature/predictor space

Motivation for a different viewpoint: Nearest centroids



Determine mean predictor vector per class

$$\widehat{\mu}_i = \frac{1}{n_i} \sum_{i_l = i} \mathbf{x}_l$$

where

$$n_i = \sum_{l=1}^{n} \mathbb{1}(i_l = i)$$

and classify points to the class who's mean is closest.

A change of scenery

Summary

- Classification can be approached through regression and approximation of $\mathbb{E}_{p(i|\mathbf{x})}[i]$
- Indirectly we approximated $p(i|\mathbf{x})$ and were able to use Bayes' rule

Observation: Good predictors group by class in feature space

Change of focus: Let's model the density of $\mathbf{x}$ conditionally on i instead!

How? **Bayes' law**

The setting of Discriminant Analysis

Apply Bayes' law

$$p(i|\mathbf{x}) = \frac{p(\mathbf{x}|i)p(i)}{\sum_{j=1}^{K} p(\mathbf{x}|j)p(j)}$$

Instead of specifying $p(i|\mathbf{x})$ we can specify

$$p(\mathbf{x}|i) \quad \text{and} \quad p(i)$$

The main assumption of Discriminant Analysis (DA) is

$$p(\mathbf{x}|i) \sim N(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$$

where $\boldsymbol{\mu}_i \in \mathbb{R}^p$ is the mean vector for class i and $\boldsymbol{\Sigma}_i \in \mathbb{R}^{p \times p}$ the corresponding covariance matrix.

Finding the parameters of DA

- Notation: Write $p(i) = \pi_i$ and consider them as unknown parameters
- Given data $(i_l, \mathbf{x}_l)$ the likelihood maximization problem is

$$\underset{\boldsymbol{\mu},\boldsymbol{\Sigma},\boldsymbol{\pi}}{\arg\max} \prod_{l=1}^{n} N(\mathbf{x}_l | \boldsymbol{\mu}_{i_l}, \boldsymbol{\Sigma}_{i_l}) \pi_{i_l} \quad \text{subject to} \quad \sum_{i=1}^{K} \pi_i = 1.$$

- Can be solved using a Lagrange multiplier (try it!) and leads to

$$\hat{\pi}_i = \frac{n_i}{n}, \quad \text{with} \quad n_i = \sum_{l=1}^{n} \mathbb{1}(i_l = i)$$

$$\hat{\boldsymbol{\mu}}_i = \frac{1}{n_i} \sum_{i_l = i} x_l$$

$$\hat{\boldsymbol{\Sigma}}_i = \frac{1}{n_i - 1} \sum_{i_l = i} (x_l - \hat{\boldsymbol{\mu}}_i)(x_l - \hat{\boldsymbol{\mu}}_i)^T$$

Performing classification in DA

Bayes' rule implies the classification rule

$$c(\mathbf{x}) = \arg\max_{1 \le i \le K} N(\mathbf{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)\pi_i$$

Note that since $\log$ is strictly increasing this is equivalent to

$$c(\mathbf{x}) = \arg\max_{1 \le i \le K} \delta_i(\mathbf{x})$$

where

$$\begin{aligned}\delta_i(\mathbf{x}) &= \log N(\mathbf{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) + \log \pi_i \\ &= \log \pi_i - \frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1}(\mathbf{x} - \boldsymbol{\mu}_i) - \frac{1}{2} \log |\boldsymbol{\Sigma}_i| \quad (+\,C)\end{aligned}$$

This is a quadratic function in $\mathbf{x}$.

Different levels of complexity

- This method is called **Quadratic Discriminant Analysis (QDA)**
- **Problem:** Many parameters that grow quickly with dimension
  - $K-1$ for all π_i
  - $p \cdot K$ for all $\boldsymbol{\mu}_i$
  - $p(p+1)/2 \cdot K$ for all $\boldsymbol{\Sigma}_i$ (most costly)
- **Solution:** Replace covariance matrices $\boldsymbol{\Sigma}_i$ by a pooled estimate

$$\widehat{\boldsymbol{\Sigma}} = \sum_{i=1}^{K} \widehat{\boldsymbol{\Sigma}}_i \frac{n_i - 1}{n - K} = \frac{1}{n-K} \sum_{i=1}^{K} \sum_{i_l = i} (x_l - \widehat{\boldsymbol{\mu}}_i)(x_l - \widehat{\boldsymbol{\mu}}_i)^T$$

- **Simpler correlation and variance structure:** All classes are assumed to have the same correlation structure between features

Performing classification in the simplified case

As before, consider

$$c(\mathbf{x}) = \underset{1 \leq i \leq K}{\arg\max}\, \delta_i(\mathbf{x})$$

where

$$\delta_i(\mathbf{x}) = \log \pi_i + \mathbf{x}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_i - \frac{1}{2} \boldsymbol{\mu}_i^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_i \quad (+\, C)$$

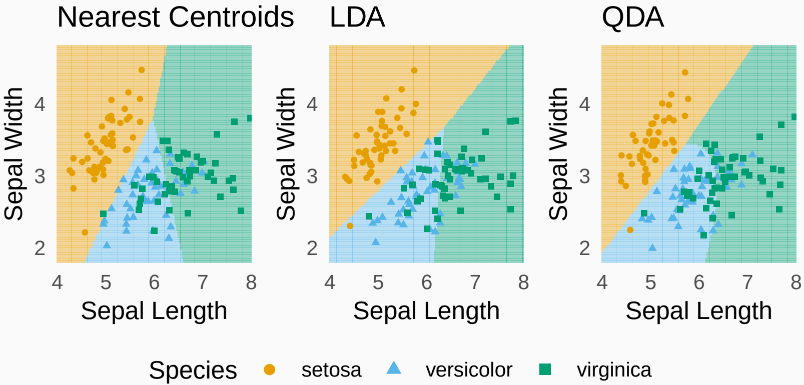
This is a linear function in $\mathbf{x}$. The method is therefore called **Linear Discriminant Analysis (LDA)**.

Even more simplifications

Other simplifications of the correlation structure are possible

- Ignore all correlations between features but allow different variances, i.e. $\mathbf{\Sigma}_i = \mathbf{\Lambda}_i$ for a diagonal matrix $\mathbf{\Lambda}_i$ (**Diagonal QDA** or **Naive Bayes' Classifier**)
- Ignore all correlations and make feature variances equal, i.e. $\mathbf{\Sigma}_i = \mathbf{\Lambda}$ for a diagonal matrix $\mathbf{\Lambda}$ (**Diagonal LDA**)
- Ignore correlations and variances, i.e. $\mathbf{\Sigma}_i = \sigma^2 \mathbf{I}_{p\times p}$ (**Nearest Centroids adjusted for class frequencies π_i**)

Examples of LDA and QDA



Decision boundaries can be found with

$$N(\mathbf{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)\pi_i = N(\mathbf{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)\pi_j \quad \text{for} \quad i \neq j$$

and $\boldsymbol{\Sigma}_i = \boldsymbol{\Sigma}$ for LDA and $\boldsymbol{\Sigma}_i = \sigma^2 \mathbf{I}_{p\times p}$ for Nearest Centroids.

Take-home message

- Classification can be achieved through the point-of-view of regression
- Modelling the conditional densities of features instead of classes leads to Discriminant Analysis (DA)
- There is a range of assumptions in DA about the correlation structure in feature space → trade-off between stability and flexibility