Lecture 2: Model-based classification

Felix Held, Mathematical Sciences

MSA220/MVE440 Statistical Learning for Big Data

26th March 2020

Recall: Statistical Learning (I)

Regression

- Theoretically best regression function for squared error loss

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

- Can be solved **data-driven** (1) or by **making 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 (with implied constant β_0 and $x_0 = 1$)

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

Learning in Statistical Learning

Learn a model from data by minimizing expected prediction error determined by a loss function.

Expected prediction error

$$J(f) = \mathbb{E}_{p(\mathbf{x},y)}\left[L(y, f(\mathbf{x}))\right] = \mathbb{E}_{p(\mathbf{x})}\left[\mathbb{E}_{p(y|\mathbf{x})}\left[L(y, f(\mathbf{x}))\right]\right]$$

How is this solved in practice, given a training sample $(y_l, \mathbf{x}_l)$ for $l = 1, \ldots, n$?

1. Approximate $\mathbb{E}_{p(\mathbf{x},y)}\left[L(y, f(\mathbf{x}))\right]$ from the training sample,

$$\text{i.e.} \quad J(f) \approx \frac{1}{n}\sum_{l=1}^{n} L(y_l, f(\mathbf{x}_l)) \rightarrow \text{minimize w.r.t. } f$$

2. Find optimal theoretical solution (e.g. $\mathbb{E}_{p(y|\mathbf{x})}[y]$) and approximate it instead,

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

A third alternative

- Theoretically best regression function for squared error loss

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

  if we allow **all functions!**
- Instead of approximating the overall optimal solution, we can restrict the class of allowed functions.
- **Example:** Restrict to class of **linear functions**, i.e.

$$f \in \{\mathbf{x} \mapsto \mathbf{x}^\top \boldsymbol{\beta} : \boldsymbol{\beta} \in \mathbb{R}^{p+1}\}$$

- Combined with squared error loss, the function minimizing (empirical) expected prediction error for the class of linear functions uses the standard least squares estimates $\hat{\boldsymbol{\beta}}$

Recall: Statistical Learning (II)

Classification

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

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

- Can be solved **data-driven** (1) or by **making 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?

Model-based classification

A model for binary classification

Consider binary classification with $i = 0$ or $i = 1$.

We want to model $p(i|\mathbf{x})$ and since $p(0|\mathbf{x}) + p(1|\mathbf{x}) = 1$, it is enough to model one of the probabilities.

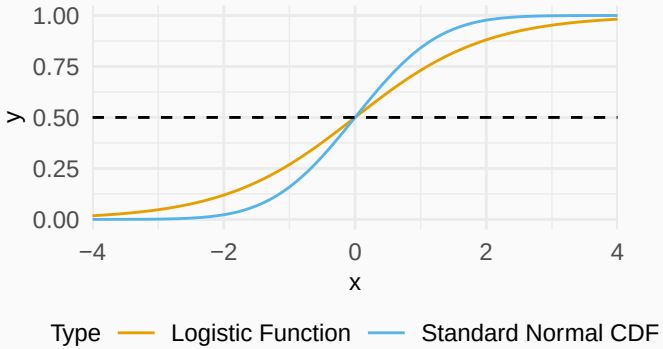
Bernoulli model:

- Let $p(1|\mathbf{x}) = \theta \in (0, 1)$, then $p(0|\mathbf{x}) = 1 - \theta$.
- Given responses i_l for $l = 1, \dots, n$ we can estimate the maximum likelihood estimate of θ
- Specifies a model approximation for **Bayes' rule**

$$c(\mathbf{x}) = \underset{i \in \{0,1\}}{\arg\max}\, p(i|\mathbf{x}) = \begin{cases} 0 & \theta \leq \frac{1}{2} \\ 1 & \text{otherwise} \end{cases}$$

How can we include predictors $\mathbf{x}_l$?

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/probit models

Include a **linear predictor** after transformation, i.e.

logistic model

$$p(1|\mathbf{x}) \approx \sigma(\mathbf{x}^\top \boldsymbol{\beta})$$

probit model

$$p(1|\mathbf{x}) \approx \Phi(\mathbf{x}^\top \boldsymbol{\beta})$$

with corresponding Bayes' rule

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

since $\sigma(\mathbf{x}^\top \boldsymbol{\beta}) \leq 1/2$ for $\mathbf{x}^\top \boldsymbol{\beta} \leq 0$ and analogous for the probit model.

How are the regression coefficients determined?

The maximum likelihood estimates of $\boldsymbol{\beta}$ in the **logistic regression model** can be determined from the log-likelihood (with $i_l^* = 2i_l - 1$)

$$l(\boldsymbol{\beta}) = \sum_{l=1}^{n} i_l \log(\sigma(\mathbf{x}_l^\top \boldsymbol{\beta})) + (1 - i_l) \log(1 - \sigma(\mathbf{x}_l^\top \boldsymbol{\beta}))$$

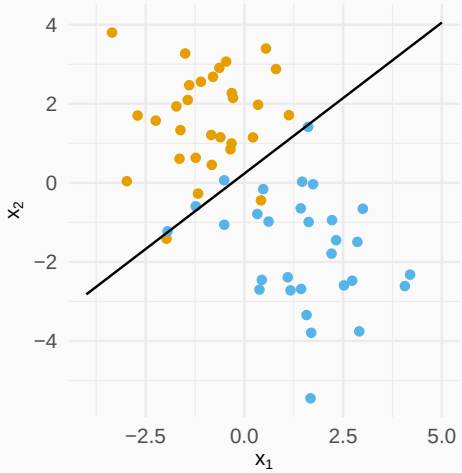
$$= \sum_{l=1}^{n} i_l^* \mathbf{x}_l^\top \boldsymbol{\beta} - \log\left(1 + \exp(i_l^* \mathbf{x}_l^\top \boldsymbol{\beta})\right)$$

The gradient is (with $\sigma(-x) = -\sigma(x)$)

$$\nabla_\beta l(\boldsymbol{\beta}) = \sum_{l=1}^{n} i_l^* \mathbf{x}_l - \sigma(i_l^* \mathbf{x}_l^\top \boldsymbol{\beta}) i_l^* \mathbf{x}_l = \sum_{l=1}^{n} \mathbf{x}_l (i_l^* - \sigma(\mathbf{x}_l^\top \boldsymbol{\beta}))$$

which can be used in **gradient ascent** or leads to a **iteratively reweighted least squares** problem via the **Newton-Raphson algorithm**. (Details in ESL Ch. 4.4.1)

Example: Logistic regression



Decision boundary (black line)

$$\beta_0 + \mathbf{x}^\top \boldsymbol{\beta} = 0$$
$$\Leftrightarrow x_2 = (-\beta_0 - \beta_1 x_1)/\beta_2$$

How can logistic regression be extended to $K > 2$ classes?

Multi-class logistic regression (I)

- Assume there are $K > 2$ classes.
- **Requirement:** Probabilities need to be modelled, i.e. $p(i|\mathbf{x}) \in (0,1)$ for each class and $\sum_i p(i|\mathbf{x}) = 1$, and dependence on predictors $\mathbf{x}_l$ should be included
- Use a categorical/multinomial model with

$$p(1|\mathbf{x}) = \theta_1, \dots, p(K-1|\mathbf{x}) = \theta_{K-1}, p(K|\mathbf{x}) = \theta_K$$

  where $\theta_j \in (0,1)$ and $\sum_j \theta_j = 1$.
- **Softmax function:** $\boldsymbol{\sigma} : \mathbb{R}^K \mapsto [0,1]^K$ for $j = 1, \dots, K-1$

$$[\boldsymbol{\sigma}(\mathbf{z})]^{(j)} = \frac{e^{z_j}}{\sum_{r=1}^{K} e^{z_r}} = \frac{e^{z_j - z_K}}{1 + \sum_{r=1}^{K-1} e^{z_r - z_K}}, \quad [\boldsymbol{\sigma}(\mathbf{z})]^{(K)} = \frac{1}{1 + \sum_{r=1}^{K-1} e^{z_r - z_K}},$$

  Note that $K-1$ inputs are enough to determine the softmax function and $z_K = 0$ could be imposed without loss of generality.

Multi-class logistic regression (II)

- Use that only $K-1$ parameters are necessary and model for $i = 1, \dots, K-1$

$$p(i|\mathbf{x}) = \frac{e^{\mathbf{x}^\top \boldsymbol{\beta}_i}}{1 + \sum_{r=1}^{K-1} e^{\mathbf{x}^\top \boldsymbol{\beta}_r}}, \quad p(K|\mathbf{x}) = \frac{1}{1 + \sum_{r=1}^{K-1} e^{\mathbf{x}^\top \boldsymbol{\beta}_r}}$$

- This method has many names: multi-class logistic regression, softmax regression, multinomial logistic regression, maximum entropy classifier, ...
- Note that for any $i \in \{1, \dots, K-1\}$

$$\log \frac{p(i|\mathbf{x})}{p(K|\mathbf{x})} = \mathbf{x}^\top \boldsymbol{\beta}_i$$

the **log-odds** of class i vs K. Class K is called the **reference class**.

Multi-class logistic regression (III)

Model for $i = 1, \dots, K-1$

$$p(i|\mathbf{x}) = \frac{e^{\mathbf{x}^\top \boldsymbol{\beta}_i}}{1 + \sum_{r=1}^{K-1} e^{\mathbf{x}^\top \boldsymbol{\beta}_r}}, \quad p(K|\mathbf{x}) = \frac{1}{1 + \sum_{r=1}^{K-1} e^{\mathbf{x}^\top \boldsymbol{\beta}_r}}$$

- Bayes rule

$$c(x) = \underset{i=1,\dots,K}{\arg\max}\, p(i|\mathbf{x}) = \begin{cases} K & \text{if } \mathbf{x}^\top \boldsymbol{\beta}_i < 0 \text{ for all } i = 1, \dots, K-1 \\ \arg\max_i \mathbf{x}^\top \boldsymbol{\beta}_i & \text{otherwise} \end{cases}$$

- Decision boundaries are found through $\mathbf{x}^\top \boldsymbol{\beta}_i = \mathbf{x}^\top \boldsymbol{\beta}_j$ and $\mathbf{x}^\top \boldsymbol{\beta}_i = 0$ for all $i, j = 1, \dots, K-1$.

Multi-class logistic regression models can be estimated with a **Newton-Raphson algorithm**, **coordinate descent**, **neural networks**, … (see ESL Ch. 4.4.1 for some pointers)

The most over-used dataset in the world

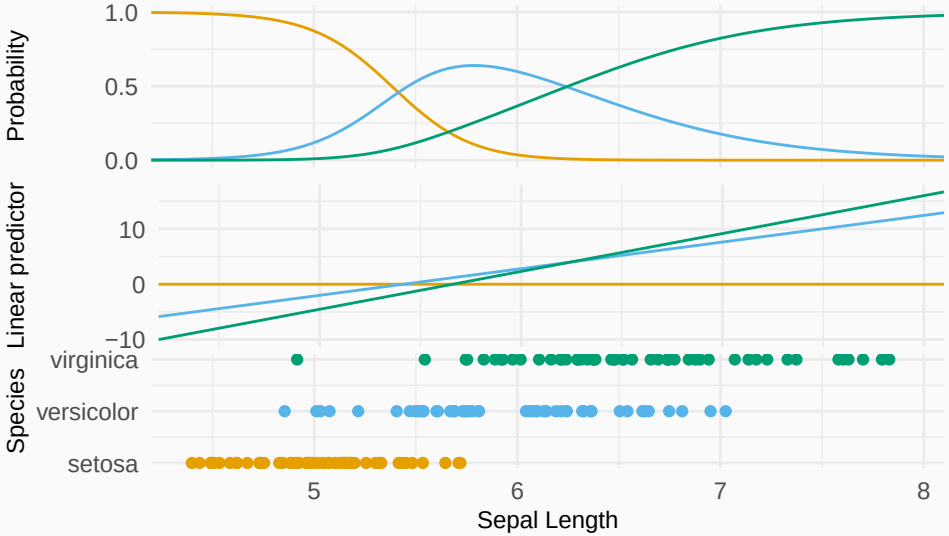
Iris flower data set

Measurements on iris flowers[1] collected by Edgar Anderson (published 1936)

- Three species: *iris setosa*, *iris virginica*, and *iris versicolor*
- 150 samples (50 for each species)
- Four features: Length and width of the sepals and petals in centimeters

[1] https://en.wikipedia.org/wiki/Iris_flower_data_set#/media/File:Iris_versicolor_3.jpg

Multi-class logistic regression: An example

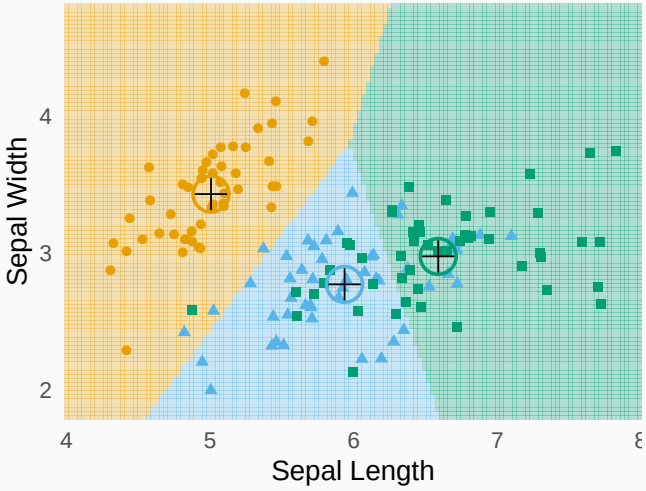


A warning: Problematic situation in two-class case (occurs seldom in practice)

- Assume two classes can be separated perfectly by a line/hyperplane in predictor space. The classes are then called **linearly separable**.
- In this situation, logistic regression tries to fit a step-like function, which forces the intercept to $-\infty$ and the corresponding predictor coefficient to $+\infty$.

Classification with focus on the feature/predictor space

Motivation for a different viewpoint: Nearest centroids



Determine mean predictor vector per class

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

where

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

and classify points to the class whose mean is closest.

A change of scenery

So far

Classification problems can be solved by approximating $p(i|\mathbf{x})$ and applying Bayes' rule

- in a data-driven way, such as kNN,
- by a transformed regression model, as in logistic/probit regression

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

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

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

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

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

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

Performing classification in DA

Bayes' rule implies the classification rule

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

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

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

where

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

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

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

- **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

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

where

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

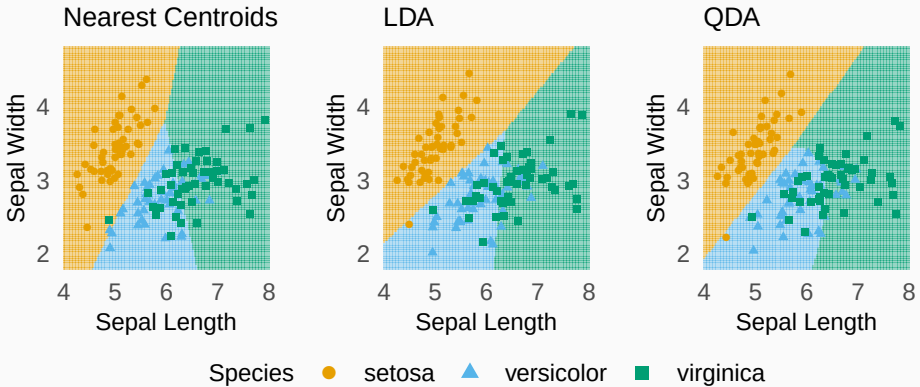
This is a linear function in $\mathbf{x}$. The method is therefore called **Linear Discriminant Analysis (LDA)**.

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 transformed 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 numerical stability and flexibility