Introduction to Machine Learning

Chapter 5: Introduction to Classification

Bernd Bischl, Christoph Molnar, Daniel Schalk, Fabian Scheipl

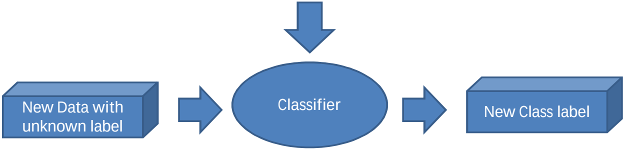
Department of Statistics – LMU Munich

CLASSIFICATION

We want to assign new observations to known categories according to criteria learned from a training set.

Our Data

Sepal Length	Sepal Width	Petal Length	Petal Width	Species
5.1	3.5	1.4	0.2	setosa
5.9	3.0	5.1	1.8	virginica



Sepal Length	Sepal Width	Petal Length	Petal Width	Species
5.4	3.3	3.2	1.1	???

CLASSIFICATION

Assume we are given a *classification problem*:

$$
\begin{aligned}
x &\in \mathcal{X} && \text{feature vector} \\
y &\in \mathcal{Y} = \{1, \ldots, g\} && \textit{categorical} \text{ output variable (label)} \\
\mathcal{D} &= \left\{ \left(x^{(1)}, y^{(1)}\right), \ldots, \left(x^{(n)}, y^{(n)}\right) \right\} && \text{observations of } x \text{ and } y
\end{aligned}
$$

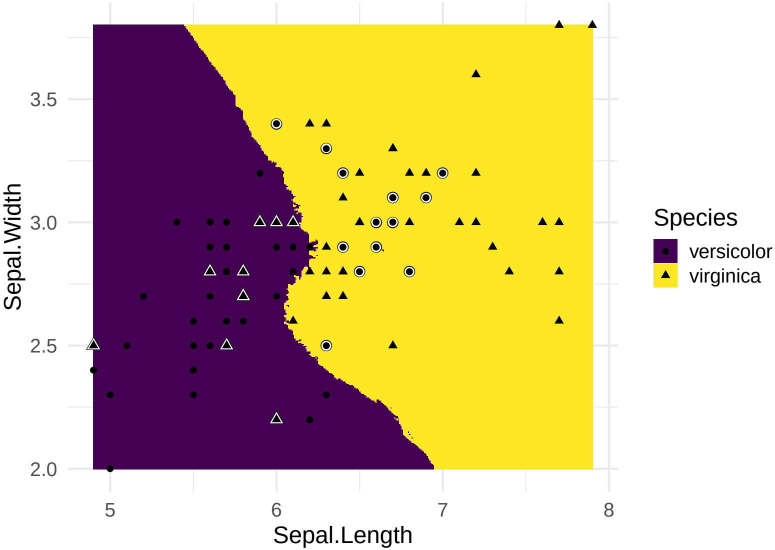
Classification usually means to construct g discriminant functions $f_1(x), \ldots f_g(x)$, so that we choose our class as

$$h(x) = \arg\max_k f_k(x)$$

for $k = 1, 2, \ldots, g$.

This divides the feature space into g *decision regions* $\{x \in \mathcal{X} | h(x) = k\}$. These regions are separated by the *decision boundaries* where ties occur between these regions.

CLASSIFICATION



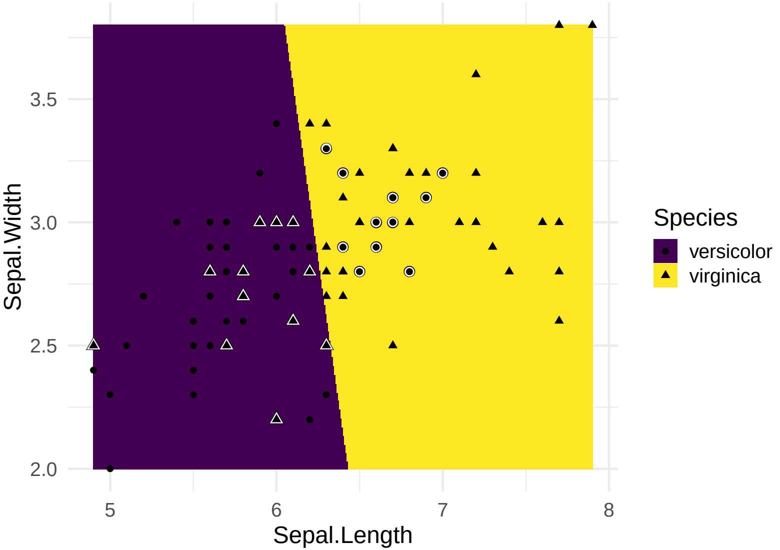
LINEAR CLASSIFIER

If these functions $f_k(x)$ can be specified as linear functions, we will call the classifier a *linear classifier*. We can then write a decision boundary as $x^T\theta = 0$, which is a hyperplane separating two classes.

If only 2 classes exist (**binary classification**), we can simply use a single discriminant function $f(x) = f_1(x) - f_2(x)$ (note that it would be more natural here to label the classes with {+1, -1} or {0, 1}).

Note that all linear classifiers can represent non-linear decision boundaries in our original input space if we include *derived features* like higher order interactions, polynomials or other transformations of x in the model.

LINEAR CLASSIFIER



CLASSIFICATION APPROACHES

Two fundamental approaches exist to construct classifiers:
The **generative approach** and the **discriminant approach**.

They tackle the classification problem from different angles:

- *Generative* classification approaches assume a data generating process in which the distribution of the features x is different for the various classes of the output y, and try to learn these conditional distributions:
  "Which y tends to have x like these?"

- *Discriminant* approaches use *empirical risk minimization* based on a suitable loss function:
  "What is the best prediction for y given these x?"

GENERATIVE APPROACH

The *generative approach* models $p(x|y=k)$, usually by making some assumptions about the structure of these distributions, and employs the Bayes theorem:

$$\pi_k(x) = \mathbb{P}(y=k|x) = \frac{\mathbb{P}(x|y=k)\mathbb{P}(y=k)}{\mathbb{P}(x)} \propto p(x|y=k)\pi_k$$

to allow the computation of $\pi_k(x)$.
The discriminant functions are then $\pi_k(x)$ or $\log p(x|y=k) + \log \pi_k$.
Prior class probabilities π_k are easy to estimate from the training data.

Examples:

- Naive Bayes classifier
- Linear discriminant analysis (generative, linear)
- Quadratic discriminant analysis (generative, not linear)

Note: LDA and QDA have 'discriminant' in their name, but are generative models! (…sorry.)

GENERATIVE APPROACH

Representation: Conditional feature distributions $p(x|y = k)$ and prior label probabilities π_k.
Often restricted to certain kinds of distributions (e.g. $\mathcal{N}(\mu, \Sigma)$) depending on the specific method, representation then via the distributions' parameters.

Optimization: Often analytic solutions (LDA, QDA); density estimation (Naive Bayes).

Evaluation: Classification loss functions. Typically: negative log posterior probability.

DISCRIMINANT APPROACH

The *discriminant approach* tries to optimize the discriminant functions directly, usually via empirical risk minimization.

$$\hat{f} = \arg\min_{f \in H} \mathcal{R}_{\text{emp}}(f) = \arg\min_{f \in H} \sum_{i=1}^{n} L\left(y^{(i)}, f\left(x^{(i)}\right)\right).$$

Examples:

- Logistic regression (discriminant, linear)
- kNN classifier (discriminant, not linear)

Representation and optimization depend on the specific learner.
Evaluation via classification loss functions.