CLASSIFICATION

Assumptions and Terminology

In a **classification problem**, we record measurements $\mathbf{x}_1, \mathbf{x}_2, \ldots$.

We assume:

1. All measurements can be represented as elements of a Euclidean $\mathbb{R}^d$.
2. Each $\mathbf{x}_i$ belongs to exactly one out of K categories, called **classes**. We express this using variables $y_i \in [K]$, called **class labels**:

$$y_i = k \quad \Leftrightarrow \quad \text{"}\mathbf{x}_i \text{ in class } k\text{"}$$

3. The classes are characterized by the (unknown!) joint distribution of (X, Y), whose density we denote $p(x, y)$. The conditional distribution with density $p(x|y = k)$ is called the **class-conditional distribution** of class k.
4. The only information available on the distribution p is a set of example measurements *with* labels,

$$(\tilde{\mathbf{x}}_1, \tilde{y}_1), \ldots, (\tilde{\mathbf{x}}_n, \tilde{y}_n) \ ,$$

called the **training data**.

Definition

A **classifier** is a function

$$f : \mathbb{R}^d \longrightarrow [K] \ ,$$

i.e. a function whose argument is a measurement and whose output is a class label.

Learning task

Using the training data, we have to estimate a good classifier. This estimation procedure is also called **training**.

A good classifier should generalize well to new data. Ideally, we would like it to perform with high accuracy on data sampled from p, but all we know about p is the training data.

Simplifying assumption

We first develop methods for the two-class case (K=2), which is also called **binary classification**. In this case, we use the notation

$$y \in \{-1, +1\} \qquad \text{instead of} \qquad y \in \{1, 2\}$$

Supervised vs. unsupervised

Fitting a model using labeled data is called **supervised learning**. Fitting a model when only $\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_n$ are available, but no labels, is called **unsupervised learning**.

Types of supervised learning methods

- Classification: Labels are discrete, and we estimate a classifier $f : \mathbb{R}^d \longrightarrow [K]$,
- Regression: Labels are real-valued ($y \in \mathbb{R}$), and we estimate a continuous function $f : \mathbb{R}^d \longrightarrow \mathbb{R}$. This functions is called a **regressor**.

Algorithm

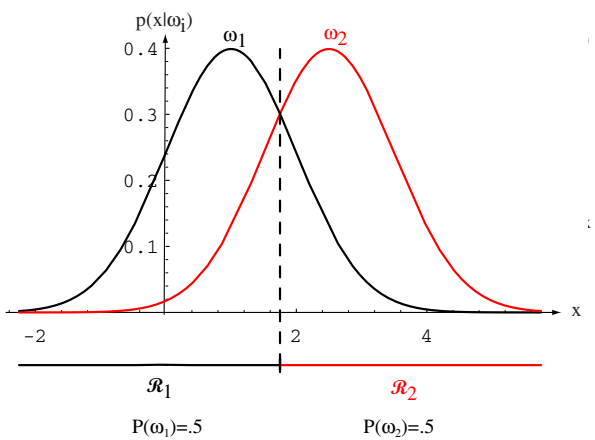
1. On training data, fit a Gaussian into each class (by MLE).
   Result: Densities $g(\mathbf{x}|\mu_\oplus, \Sigma_\oplus)$ and $g(\mathbf{x}|\mu_\ominus, \Sigma_\ominus)$
2. Classify a new point $\mathbf{x}$ according to which density assigns larger value:

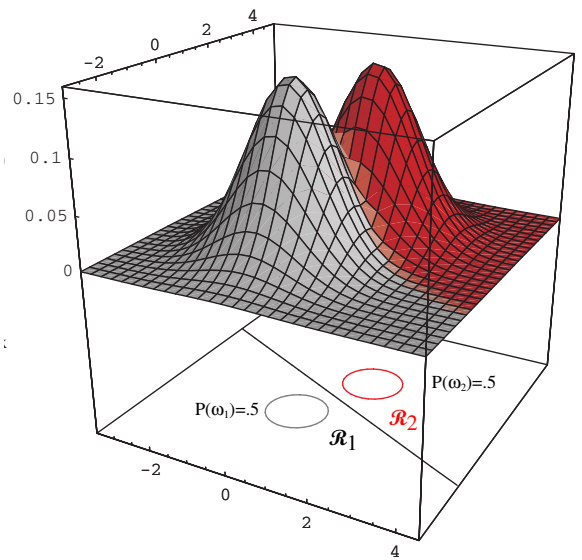
$$y_i := \begin{cases} +1 & \text{if } g(\mathbf{x}|\mu_\oplus, \Sigma_\oplus) > g(\mathbf{x}|\mu_\ominus, \Sigma_\ominus) \\ -1 & \text{otherwise} \end{cases}$$

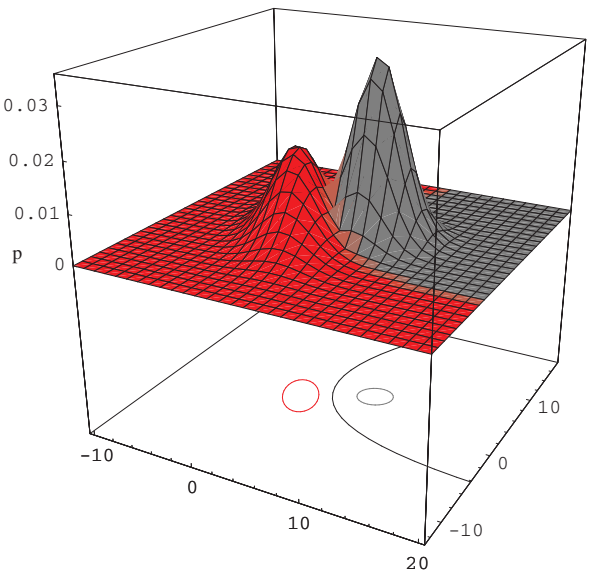
Resulting classifier

- Hyperplane if $\Sigma_\oplus$=$\Sigma_\ominus$ = constant $\cdot$ diag$(1, \ldots, 1)$ ("isotropic" Gaussians).
- Curved surface otherwise.

A Very Simple Classifier







Possible weakness

1. Distributional assumption.
2. Density estimates emphasize main bulk of data. Critical region for classification is at decision boundary, i.e. region between classes.

Consequence

- Classification algorithms focus on class boundary.
- Technically, this means: We focus on estimating a good decision surface (e.g. a hyperplane) between the classes; we do *not* try to estimate a distribution.

Our program in the following

- First develop methods for the linear case, i.e. separate two classes by a hyperplane.
- Then: Consider methods that do not require the decision surface (= the boundary between classes) to be linear (= a straight line or plane).
- Dealing with more than two classes.

Definition

A **loss function** is a function

$$L : [K] \times [K] \longrightarrow [0, \infty) \ ,$$

which we read as

$$L : (\text{true class label } y, \ \text{classifier output } f(x)) \longmapsto \ \text{loss value} \ \ .$$

Example: The two most common loss functions

1. The **0-1 loss** is used in classification. It counts mistakes:

$$L^{\text{0-1}}(y, f(\mathbf{x})) = \begin{cases} 0 & f(\mathbf{x}) = y \\ 1 & f(\mathbf{x}) \neq y \end{cases}$$

2. **Squared-error loss** is used in regression:

$$L^{\text{se}}(y, f(\mathbf{x})) := \|y - f(\mathbf{x})\|_2^2$$

   Its value depends on how far off we are: Small errors hardly count, large ones are very expensive.

Motivation

It may be a good strategy to allow (even expensive) errors for values of $\mathbf{x}$ which are very unlikely to occur

Definition

The **risk** $R(f)$ of a classifier f is its *expected loss under p*. If you prefer equations:

$$R(f) := \mathbb{E}_p[L(y, f(\mathbf{x})] = \int L(y, f(\mathbf{x}))p(\mathbf{x}, y)d\mathbf{x}dy = \sum_{y=1}^{K} \int L(y, f(\mathbf{x}))p(\mathbf{x}, y)d\mathbf{x} \ .$$

When we train f, we do not know p, and have to approximate R using the data:

The **empirical risk** $\hat{R}_n(f)$ is the plug-in estimate of $R(f)$, evaluated on the training sample $(\tilde{\mathbf{x}}_1, \tilde{y}_1), \ldots, (\tilde{\mathbf{x}}_n, \tilde{y}_n)$:

$$\hat{R}_n(f) := \frac{1}{n} \sum_{i=1}^{n} L(\tilde{y}_i, f(\tilde{\mathbf{x}}_i))$$