CENTRALE NANTES

LS2N LABORATOIRE DES SCIENCES DU NUMÉRIQUE DE NANTES

Linear classification

APSTA - LEARN

Diana Mateus

Table of contents

Background: Linear Models

Linear Regression

Data:

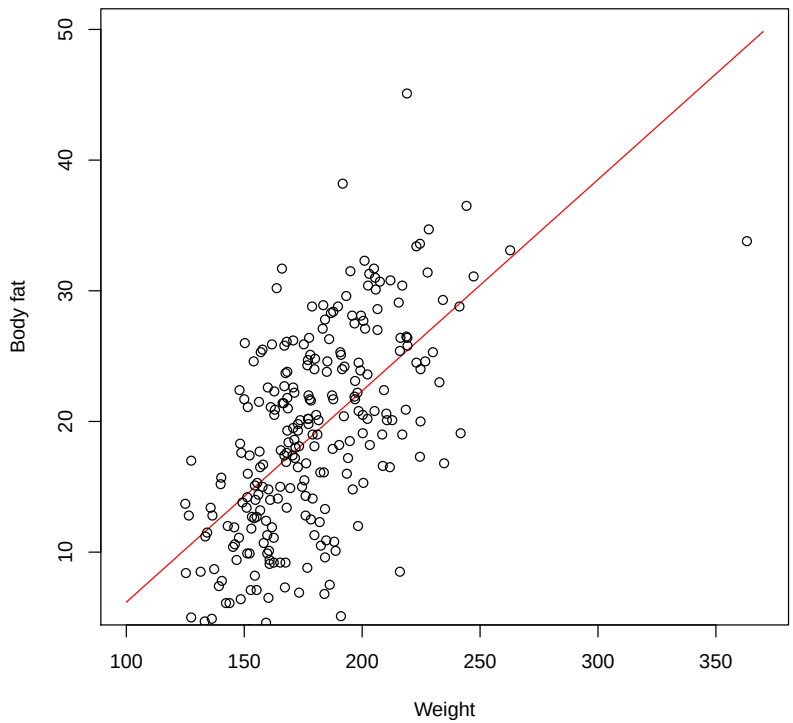
- $\mathbf{x}_i$ is a data point.
- y_i is a target value.
- each $\mathbf{x}_i$ has m features.

$$\mathbf{x}_i = (x_{i1}, \ldots, x_{im})^\top$$

- there are n such data points

$\rightarrow$ Data matrix $\mathbf{X}$ (size $n \times m$)

$\rightarrow$ Target vector $\mathbf{y}$ (size $n \times 1$)



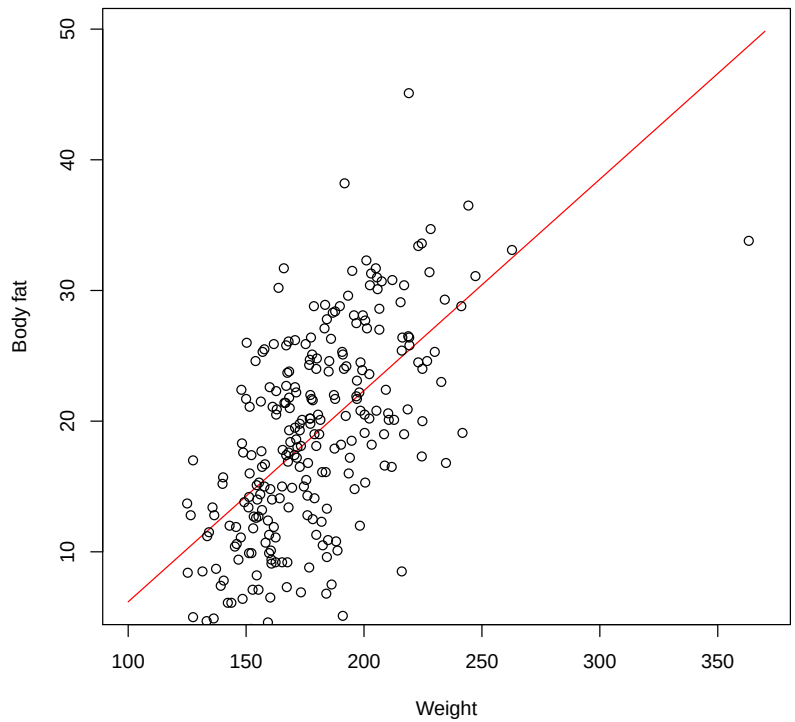
Linear Regression

Definition (Linear Regression)

$$y_i = w_0 + w_1 x_{i1} + \ldots + w_m x_{im}$$

$$y_i = w_0 + \mathbf{x}_i^\top \mathbf{w}$$

- w_i are **coefficients** or weights of each features.
- w_0 denotes the **intercept**.



Linear Regression

Ordinary Least Squares Estimation

Definition (Residual Sum of Squares; RSS)

$$\mathrm{RSS}(w_0, \ldots, w_m) = \sum_{i=1}^{n} (y_i - \hat{f}(\mathbf{x}_i))^2$$

- RSS gives the total loss over the whole training set
- Choose the coefficients $w_0, \ldots, w_m$ such that the total loss according to RSS is **minimized**.

Linear Regression

Ordinary Least Squares Estimation

- Set the partial derivative of RSS to zero
- In matrix form:

$$\mathrm{RSS}(\mathbf{w}) = (\mathbf{y} - \mathbf{X}\mathbf{w})^\top (\mathbf{y} - \mathbf{X}\mathbf{w}) \tag{1}$$

$$\frac{\partial \mathrm{RSS}(\mathbf{w})}{\partial \mathbf{w}} = -2\mathbf{X}^\top (\mathbf{y} - \mathbf{X}\mathbf{w}). \tag{2}$$

- **Note**: where $\mathbf{w} = (w_0, \ldots, w_m)^\top$ and the first column of $\mathbf{X}$ contains only 1 to accommodate the intercept w_0, i.e. $\mathbf{X}$ is a $n \times m + 1$ matrix.

Linear Models – Ordinary Least Squares Estimation

Definition (Ordinary Least Squares Estimate)

$$\hat{\mathbf{w}} = \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \mathbf{y}$$

- The minimum of the loss function is **unique**.
- Estimates of the coefficients can be obtained in **closed form** solution and therefore no optimization is required.
- $\mathbf{X}$ must have full column rank $\Rightarrow \mathbf{X}^\top \mathbf{X}$ is positive definite.
- Prediction (regression) is performed by

$$\hat{f}(x_1, \ldots, x_m) = \hat{w}_0 + \hat{w}_1 x_1 + \ldots + \hat{w}_m x_m$$

Classification

Background: Linear Models

Ordinary Least Squares

Classification

Logistic Regression

Linear Classifiers

Optimal Separating Hyperplanes

The geometric margin

Linear SVMs

Non-linear Support Vector Machiness

Classification– Definitions

- **Training sample** $\mathbf{x}_i$ consists of m **features** $(x_{i1}, \ldots, x_{im})^\top$
- To each training sample $\mathbf{x}_i$ is associated a training **output** y_i.
- **Training set** $\mathcal{T} = \{\mathbf{x}_i, y_i\}_{i=1}^n$.
- Data matrix $\mathbf{X}$ with the i-th sample in the i-th row
- $\mathbf{y} = (y_1, \ldots, y_n)^\top$ the vector of all outputs.

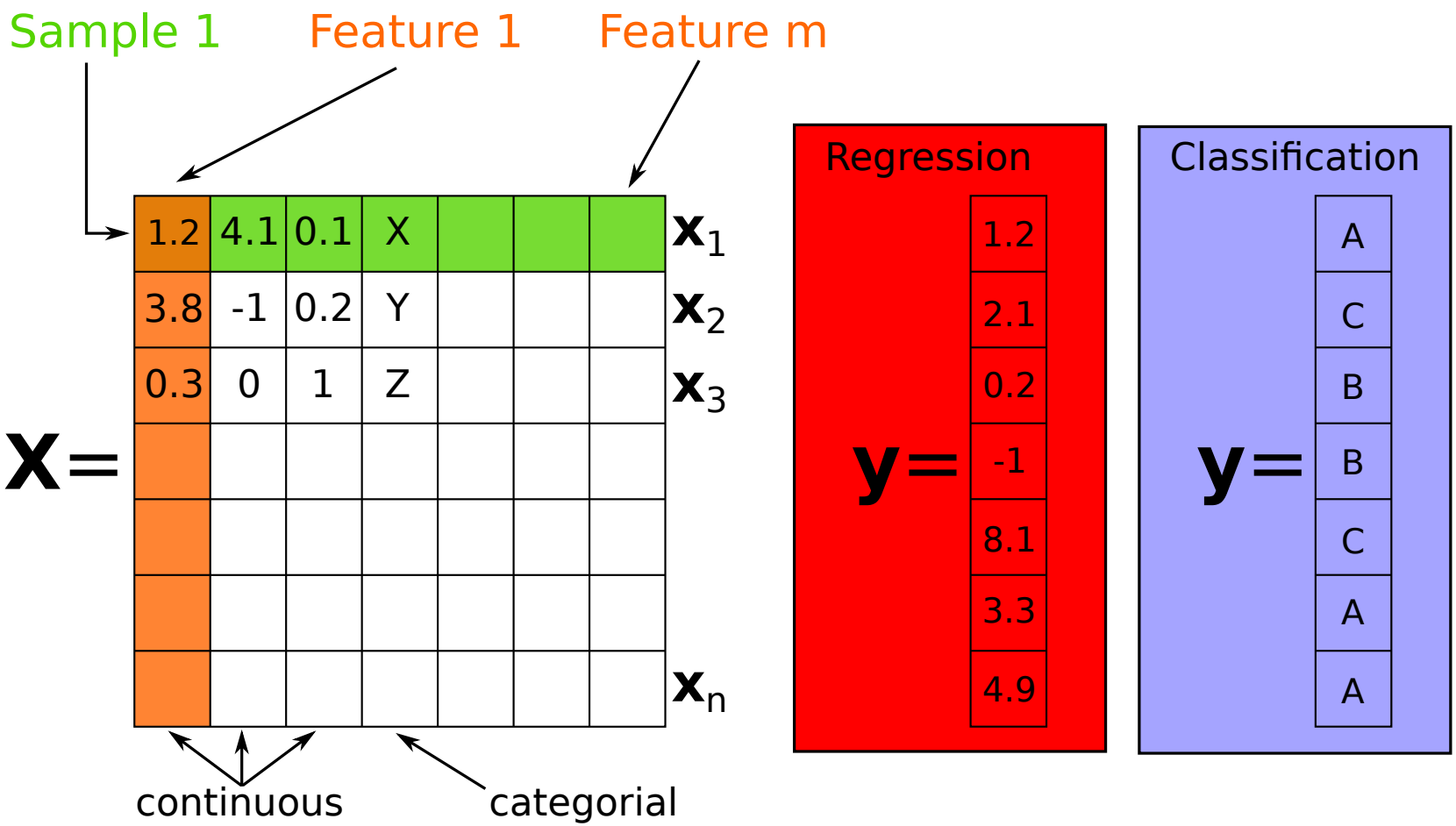
Classification– Definitions

- **Training sample** $\mathbf{x}_i$ consists of m **features** $(x_{i1}, \ldots, x_{im})^\top$
- To each training sample $\mathbf{x}_i$ is associated a training **output** y_i.
- **Training set** $\mathcal{T} = \{\mathbf{x}_i, y_i\}_{i=1}^n$.
- Data matrix $\mathbf{X}$ with the i-th sample in the i-th row
- $\mathbf{y} = (y_1, \ldots, y_n)^\top$ the vector of all outputs.

Question:

What is the difference with regression?

Classification – Definitions

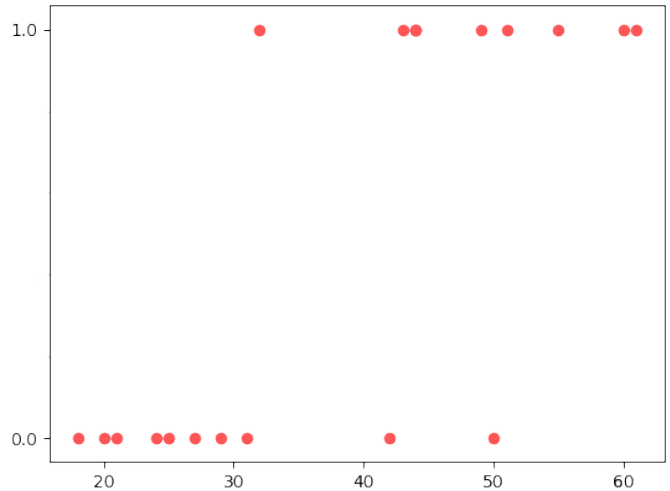


Classification – Definitions

- Each **feature** can be:
  - **continuous** (a number).
  - **discrete** (from a predefined set of values).

- The **output** can be:
  - continuous, we perform regression.
  - discrete, we perform classification.

Linear Classification – Definitions

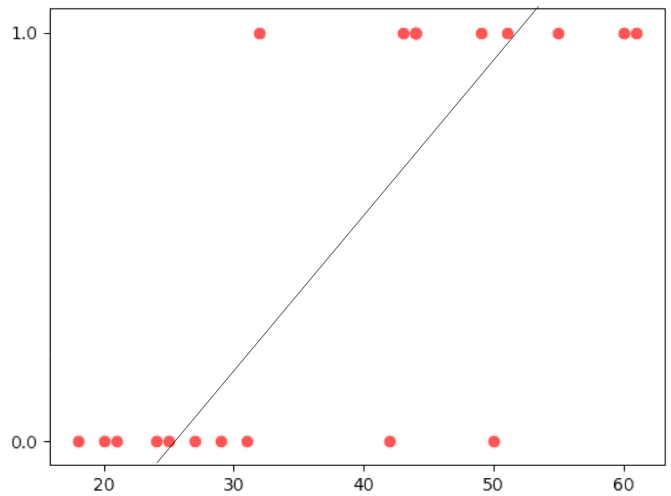


Classification Goal

Find a **decision function** f such that: $f(\mathbf{x}_i) = y_i$

- Assume a **linear model** for f.

Linear Classification – Definitions

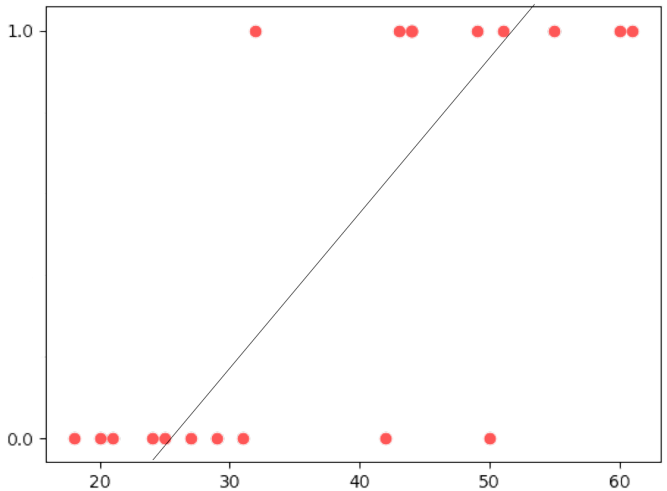


Classification Goal

Find a **decision function** f such that: $f(\mathbf{x}_i) = y_i$

- Assume a **linear model** for f.
- We solved linear **regression** with least squares ...

Linear Classification – Definitions

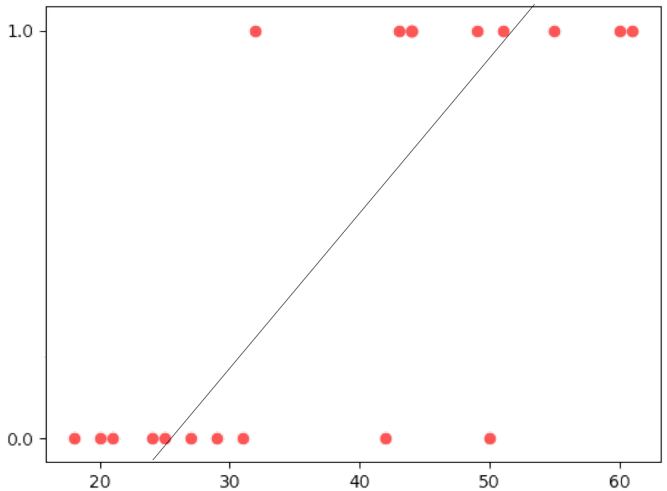


Classification Goal

Find a **decision function** f such that: $f(\mathbf{x}_i) = y_i$

- Assume a **linear model** for f.
- We solved linear **regression** with least squares ...
- Problem?

Linear Classification – Definitions

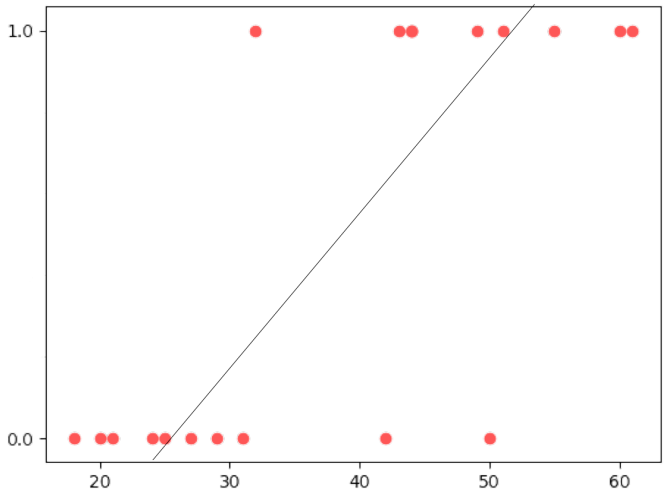


Classification Goal

Find a **decision function** f such that: $f(\mathbf{x}_i) = y_i$

- Assume a **linear model** for f.
- We solved linear **regression** with least squares ...
- Problem? in **classification** the outcome y_i is **categorical**.

Linear Classification – Definitions

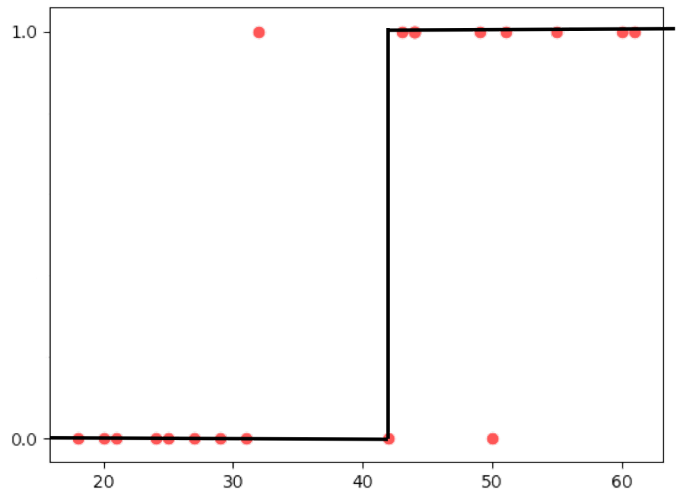


Classification Goal

Find a **decision function** f such that: $f(\mathbf{x}_i) = y_i$

- Assume a **linear model** for f.
- We solved linear **regression** with least squares ...
- Problem? in **classification** the outcome y_i is **categorical**.
- Naive solution: use a **threshold**.

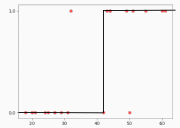
Linear Classification



Classification Goal

Find a **decision function** f such that: $f(\mathbf{x}_i) = y_i$

how do we optimize for such function?



Classification Goal

Find a **decision function** f such that: $f(\mathbf{x}_i) = y_i$

how do we optimize for such function?

The straightforward way to approximate this ideal function is to use two line segments to fit the dots, which are also referred to as training data points. However, to be learnable, we want to use a differentiable function to do the fitting instead of the two line segments.

Logistic Regression

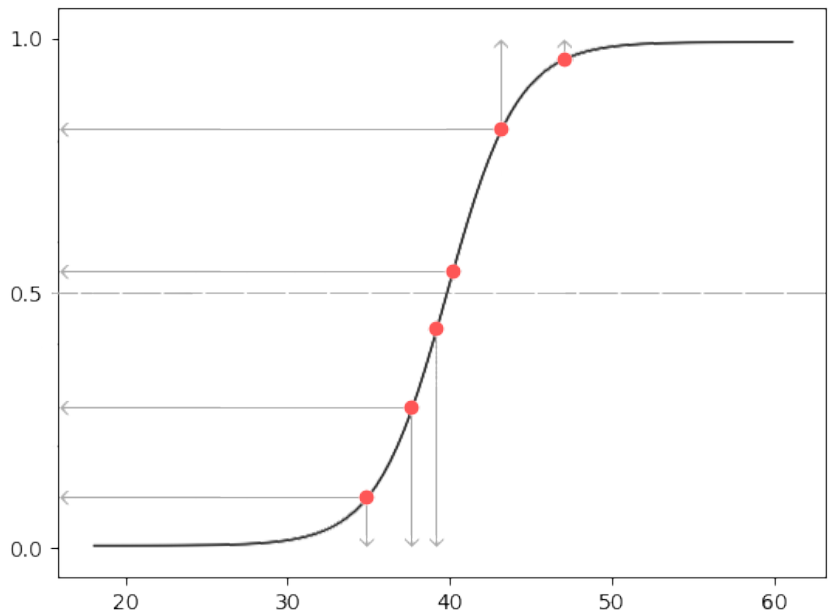
Pass the initial line through a **sigmoid function** ensuring that $0 \leq f(\mathbf{x}) \leq 1$

$$f(\mathbf{x}_i) = \text{sigm}(\mathbf{w}^\top \mathbf{x}_i) \quad \text{where} \quad \text{sigm}(\eta) = \frac{1}{1+e^{-\eta}} = \frac{e^\eta}{1+e^\eta}$$

The sigmoid function is also known as logistic function.

Prediction model:

$$f(\mathbf{x}_i) = \frac{1}{1 + e^{-(\mathbf{w}^\top \mathbf{x}_i)}}$$



Logistic Regression

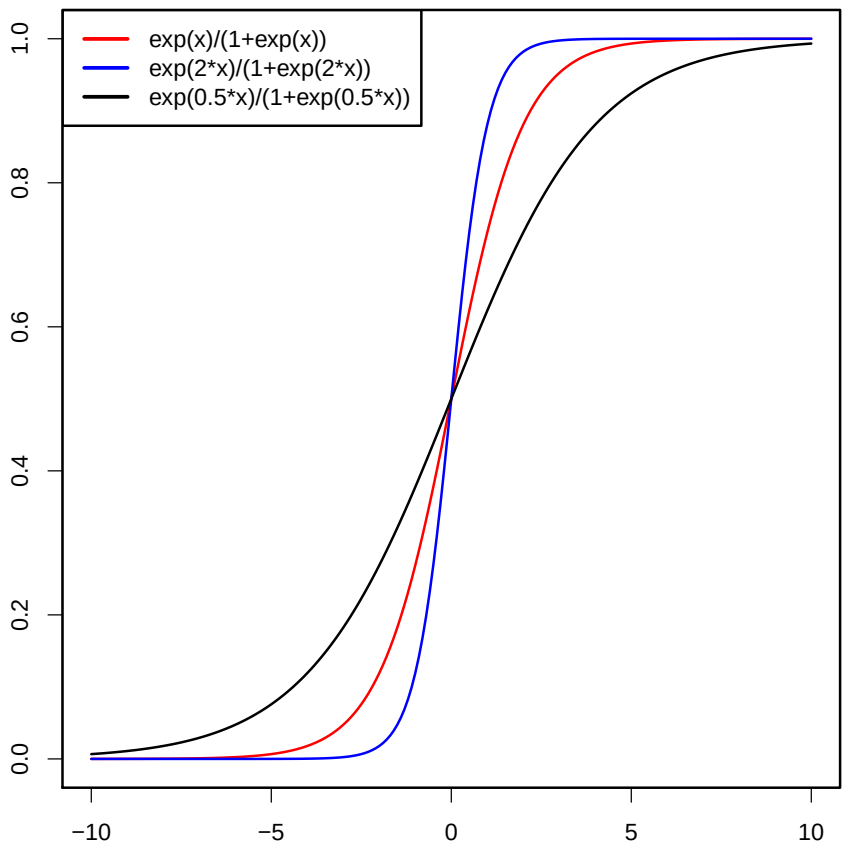
Pass the initial line through a **sigmoid function** ensuring that $0 \leq f(\mathbf{x}) \leq 1$

$$f(\mathbf{x}_i) = \text{sigm}(\mathbf{w}^\top \mathbf{x}_i) \quad \text{where} \quad \text{sigm}(\eta) = \frac{1}{1+e^{-\eta}} = \frac{e^\eta}{1+e^\eta}$$

The sigmoid function is also known as logistic function.

Prediction model:

$$f(\mathbf{x}_i) = \frac{1}{1 + e^{-(\mathbf{w}^\top \mathbf{x}_i)}}$$



Logistic Regression

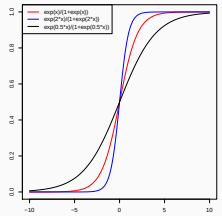
Pass the initial line through a **sigmoid function** ensuring that $0 \leq f(\mathbf{x}) \leq 1$

$f(\mathbf{x}_i) = \text{sigm}(\mathbf{w}^\top \mathbf{x}_i)$ where $\text{sigm}(\eta) = \frac{1}{1+e^{-\eta}} = \frac{e^\eta}{1+e^\eta}$

The sigmoid function is also known as logistic function.

Prediction model:

$$f(\mathbf{x}_i) = \frac{1}{1 + e^{-(\mathbf{w}^\top \mathbf{x}_i)}}$$



The Logistic function, which is also referred to as sigmoid function, can be employed here. Logistic function is a monotonic, continuous function between 0 and 1

Logistic Regression

Pass the initial line through a **sigmoid function** ensuring that $0 \leq f(\mathbf{x}) \leq 1$

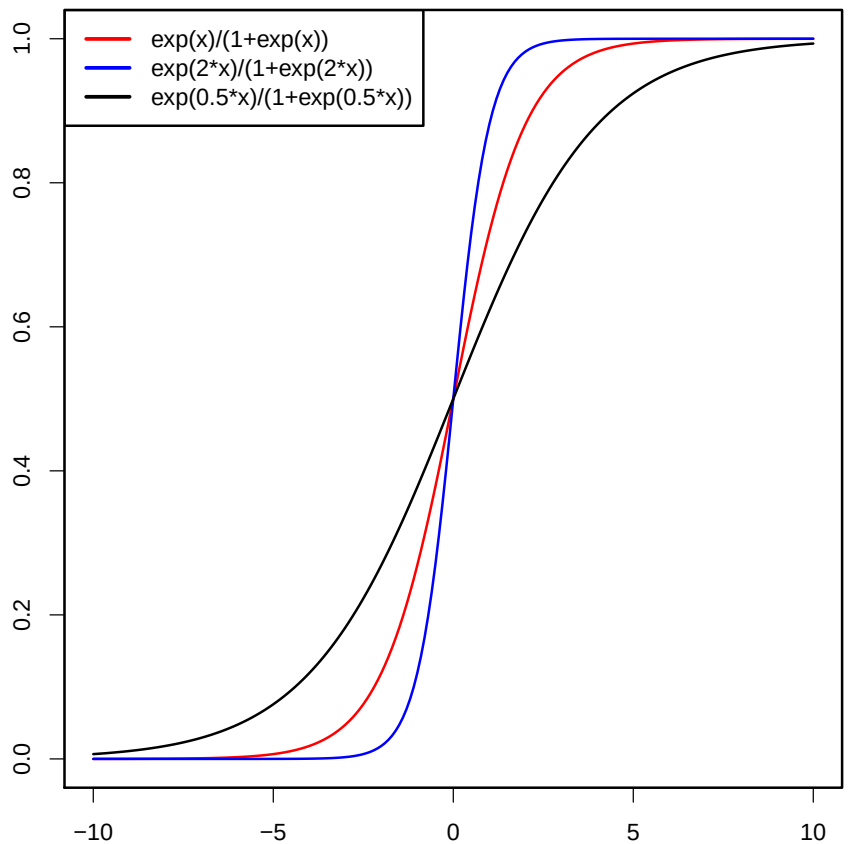
$$f(\mathbf{x}_i) = \text{sigm}(\mathbf{w}^\top \mathbf{x}_i) \quad \text{where} \quad \text{sigm}(\eta) = \frac{1}{1+e^{-\eta}} = \frac{e^\eta}{1+e^\eta}$$

The sigmoid function is also known as logistic function.

Prediction model:

$$f(\mathbf{x}_i) = \frac{1}{1 + e^{-(\mathbf{w}^\top \mathbf{x}_i)}}$$

Logistic regression is a form of **probabilistic classification**!



Logistic Regression – Formally

- In the case of a **binary** classification problem $y_i \in \{0, 1\}$:
  - $y_i = 1$ is the **positive** class,
  - $y_i = 0$ is the **negative** class.
- The probability of variables taking one of two possible outcomes can be modelled with the discrete **Bernoulli** distribution:

$$P(y = 1) = q^y (1 - q)^{(1-y)}.$$

  with q the probability of the class 1 and $(1 - p)$ that of class 0
- The prediction function represents the **posterior probability** π_i of belonging to the positive class, given sample $\mathbf{x}_i$

$$\pi_i = P(y_i = 1 | x_{i1}, \ldots, x_{im})$$

Logistic Regression – Maximum Likelihood Estimation

The likelihood of the training set is modeled as the product of Bernoulli events:

Definition (Likelihood function)

$$L(w_0, \mathbf{w}) = \prod_{i=1}^{n} P(y_i|\mathbf{x}_i) = \prod_{i=1}^{n} \pi_i^{y_i}(1-\pi_i)^{1-y_i}$$

The estimate of the parameters $\mathbf{w}$ can be derived by maximizing

Definition (Maximum Likelihood Estimate; MLE)

$$\hat{\mathbf{w}} = \arg\max_{w_0, \mathbf{w}} L(w_0, \mathbf{w})$$

Solve with (weighted) Iterative Least Squares

Logistic Regression – Maximum Likelihood Estimation

The likelihood of the training set is modeled as the product of Bernoulli events:

Definition (Likelihood function)

$$L(w_0, \mathbf{w}) = \prod_{i=1}^{n} P(y_i|\mathbf{x}_i) = \prod_{i=1}^{n} \pi_i^{y_i}(1-\pi_i)^{1-y_i}$$

The estimate of the parameters $\mathbf{w}$ can be derived by maximizing

Definition (Maximum Likelihood Estimate; MLE)

$$\hat{\mathbf{w}} = \arg\max_{w_0, \mathbf{w}} L(w_0, \mathbf{w})$$

Solve with (weighted) Iterative Least Squares

This formulation can be extended to multiple categorical output cases.

Homework: Find the solution of the max likelihood problem.

Logistic regression optimization in action

logistic regression optimization video

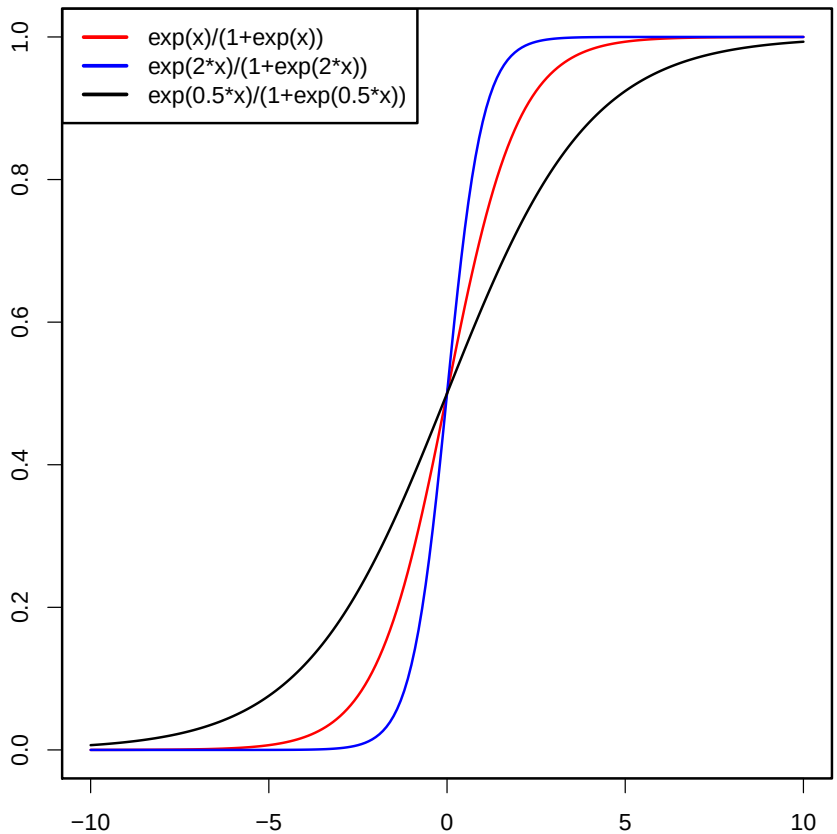
Logistic Regression – Response and link function

- The logistic function is also called **response function**.

$$\text{sigm}(\eta) = \frac{1}{1+e^{-\eta}} = \frac{e^{\eta}}{1+e^{\eta}}$$

- Its inverse is known as the **logit** or **link function**

$$\text{logit}(x) = \log\left(\frac{x}{1-x}\right)$$



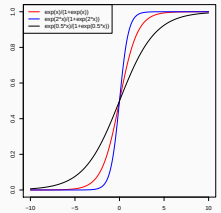
Logistic Regression – Response and link function

- The logistic function is also called **response function**.

$$\mathrm{sigm}(\eta) = \frac{1}{1+e^{-\eta}} = \frac{e^{\eta}}{1+e^{\eta}}$$

- Its inverse is known as the **logit** or **link function**

$$\mathrm{logit}(x) = \log\left(\frac{x}{1-x}\right)$$



Verify the inverse by replacing the logit function inside the sigmoid.

$$\mathrm{sigm}(\mathrm{logit}(x))\frac{\exp\left(\ln\left(\frac{x}{1-x}\right)\right)}{1+\exp\left(\ln\left(\frac{x}{1-x}\right)\right)}$$

Logistic Regression – Log odds

The logit or link function $\text{logit}(x) = \ln\left(\frac{x}{1-x}\right)$ can be used to define the **Log odds**, which are an alternate way of expressing probabilities.

The **log odds** of a prediction $\hat{y} = 1$ are:

$$\text{logit}(P(\hat{y} = 1)) = \ln\left(\frac{P(\hat{y} = 1)}{P(\hat{y} = 0)}\right)$$

Log odss are linked to the linear model by the following expression:

$$\text{logit}(P(\hat{y}_i = 1)) = \mathbf{w}^\top \mathbf{x}_i$$

such that each w_j can be **interpreted** as the contribution of x_{ij} to the log odds.

Logistic Regression – Log odds

The logit or link function $\text{logit}(x) = \ln\left(\frac{x}{1-x}\right)$ can be used to define the **Log odds**, which are an alternate way of expressing probabilities.

The **log odds** of a prediction $\hat{y} = 1$ are:

$$\text{logit}(P(\hat{y} = 1)) = \ln\left(\frac{P(\hat{y} = 1)}{P(\hat{y} = 0)}\right)$$

Log odss are linked to the linear model by the following expression:

$$\text{logit}(P(\hat{y}_i = 1)) = \mathbf{w}^\top \mathbf{x}_i$$

such that each w_j can be **interpreted** as the contribution of x_{ij} to the log odds.

$$\begin{aligned}
P(y=1) &= \frac{e^{\mathbf{w}^T\mathbf{x}}}{1+e^{\mathbf{w}^T\mathbf{x}}} \\
P(y=1)(1+e^{\mathbf{w}^T\mathbf{x}}) &= e^{\mathbf{w}^T\mathbf{x}} \\
P(y=1) + P(y=1)e^{\mathbf{w}^T\mathbf{x}} &= e^{\mathbf{w}^T\mathbf{x}} \\
P(y=1) &= e^{\mathbf{w}^T\mathbf{x}}(1-P(y=1)) \\
\frac{P(y=1)}{(1-P(y=1))} &= e^{\mathbf{w}^T\mathbf{x}} \\
\ln\left(\frac{P(y=1)}{P(y=0)}\right) &= \mathbf{w}^T\mathbf{x}
\end{aligned}$$

Logistic Regression – Log-Odds Ratio

Definition (Log-Odds ratio)

The coefficient w_i represents the **log-odds ratio** of the i-th feature

- $w_i > 0 \Leftrightarrow$ Odds increase
- $w_i < 0 \Leftrightarrow$ Odds decrease
- $w_i = 0 \Leftrightarrow$ Odds remain unchanged
- This becomes very handy to assess which feature has the largest influence, especially if the goal is to predict which patients are diseased based on clinical features.

Logistic Regression – Example

Birth weight data contains data from 189 births to determine which of these factors were risk factors for low birth weight (< 2.5 kg)

Feature	w / log-odds ratio	Chance
(Intercept)	0.924910	
Age	-0.042784	decreased
Mother's weight (pounds)	-0.015436	decreased
Race = White	0	
Race = Black	1.168452	increased
Race = Other	0.814620	increased
Previous premature labour	1.333970	increased
History of hypertension	1.740511	increased
Smoking during pregnancy	0.858332	increased

Logistic Regression – Log odds

Advantages

- low computational demand.
- highly interpretable.
- does not require features to be scaled.
- probabilistic output.
- easy to implement and fast training.
- good baseline model.

Disadvantages

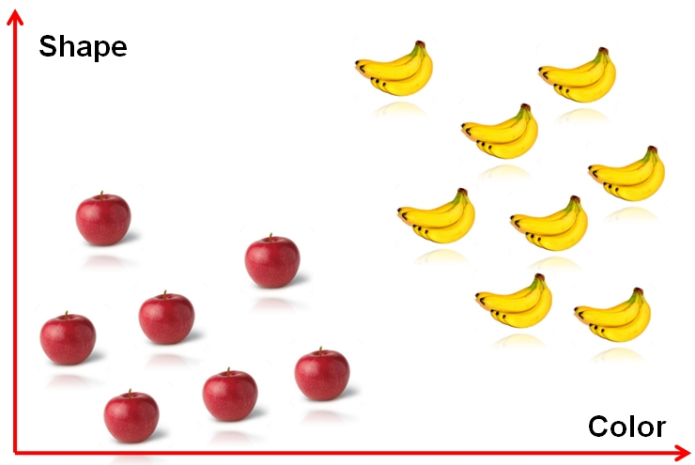
- Works better with when removing unrelated and correlated attributes, so it is dependend on feature engineering.
- the decision surface is linear.
- its low complexity may lead to overfitting.

Binary classification problem

Input: Objects from 2 classes:  = -1,  = 1

Goal: Find a **decision function** f such that :

$$f(\mathbf{x}_i) = y_i,$$
$$\text{for all } i \in \{1, \cdots, n\}$$



$f($  $) = \texttt{'apple'} = -1$

Binary classification problem

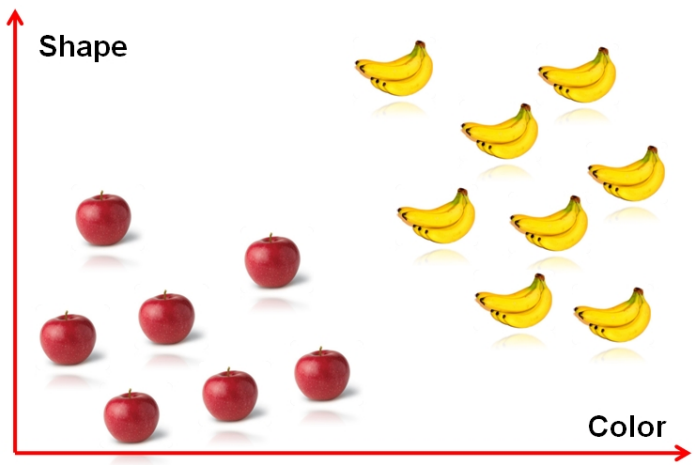
Input: Objects from 2 classes:  = -1,  = 1

Goal: Find a **decision function** f such that :

$$f(\mathbf{x}_i) = y_i,$$
$$\text{for all } i \in \{1, \cdots, n\}$$

Question:

how?

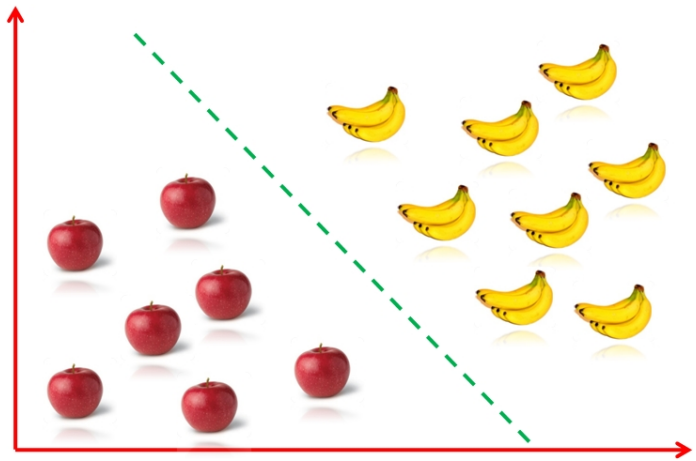


$f($  $) = \texttt{'apple'} = -1$

Binary classification problem – Generalized Linear Classifier

- Given training data $\{\mathbf{x}_i,\ y_i\}_{i\in\{1,\cdots,n\}}$
- Find a **hyperplane** h that separates the data points in 2 classes:

$$h(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}_i + w_0$$

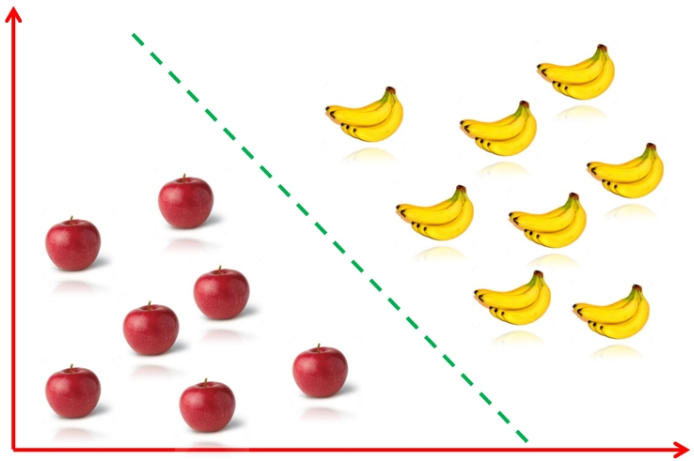


- The line of interest is created when the hyperplane cuts the feature space $(h = 0)$

Binary classification problem – Generalized Linear Classifier

- Given training data $\{\mathbf{x}_i,\ y_i\}_{i\in\{1,\cdots,n\}}$
- Find a **hyperplane** h that separates the data points in 2 classes:

$$h(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}_i + w_0$$

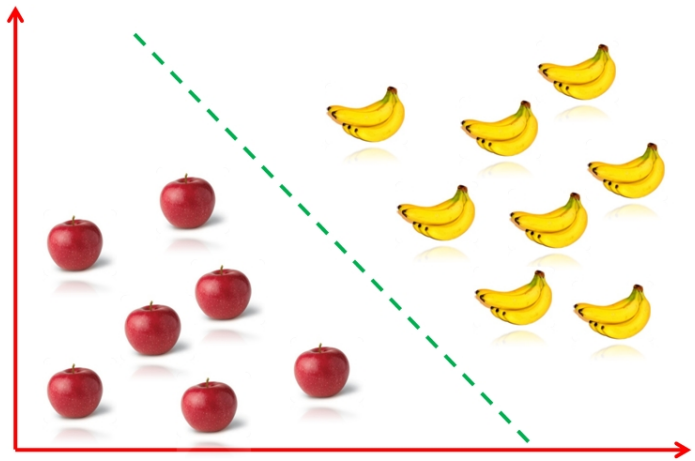


- The line of interest is created when the hyperplane cuts the feature space $(h = 0)$
- $w_0 = 0$
  $\mathbf{w} = [2, -1]$
  $\{\mathbf{x}_i\} = \{[2, 0], [0, 2], [2, 4]\}$
  $h(\mathbf{x}_i) = ?$

Binary classification problem – Generalized Linear Classifier

- Given training data $\{\mathbf{x}_i,\ y_i\}_{i\in\{1,\cdots,n\}}$
- Find a **hyperplane** h that separates the data points in 2 classes:

$$h(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}_i + w_0$$



- The line of interest is created when the hyperplane cuts the feature space $(h = 0)$

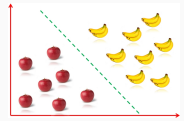
- How to link these values with the desired y_i?

Binary classification problem – Generalized Linear Classifier

- Given training data $\{\mathbf{x}_i, y_i\}_{i\in\{1,\cdots,n\}}$
- Find a **hyperplane** h that separates the data points in 2 classes:

$$h(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}_i + w_0$$

- The line of interest is created when the hyperplane cuts the feature space ($h = 0$)



- How to link these values with the desired y_i?

Use the signs of the hyperplane equation to make predictions.

$$\{h(x_i)\} = \{4, -2, 0\}$$

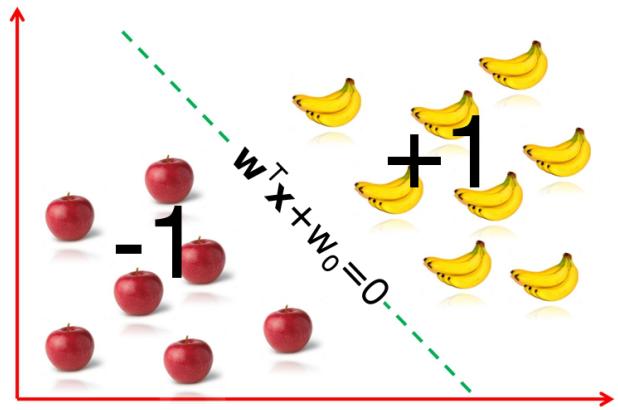
Binary classification problem – Generalized Linear Classifier

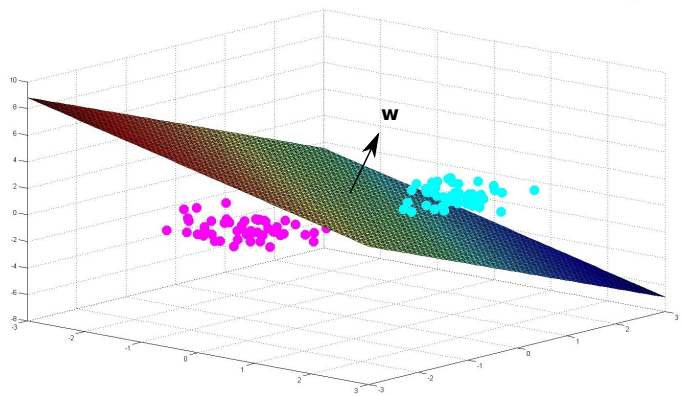
- Given training data $\{\mathbf{x}_i,\ y_i\}_{i\in\{1,\cdots,n\}}$
- Find hyperplane:

$$h(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + w_0$$

Subject to constraints for all i:

$$\mathbf{w}^\top \mathbf{x}_i + w_0 > 0 \text{ if } y_i = +1$$
$$\mathbf{w}^\top \mathbf{x}_i + w_0 < 0 \text{ if } y_i = -1$$





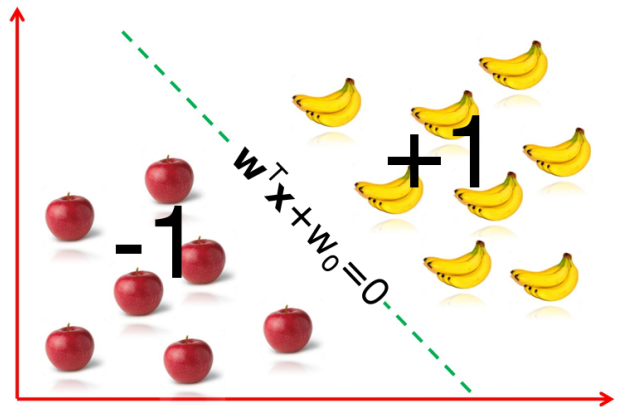
Binary classification problem – Generalized Linear Classifier

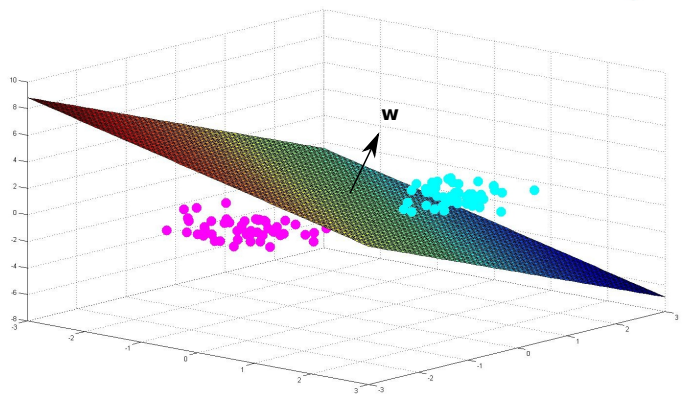
- Given training data $\{\mathbf{x}_i,\ y_i\}_{i\in\{1,\cdots,n\}}$
- Find hyperplane:

$$h(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + w_0$$

Subject to constraints for all i:

$$\mathbf{w}^\top \mathbf{x}_i + w_0 > 0 \text{ if } y_i = +1$$
$$\mathbf{w}^\top \mathbf{x}_i + w_0 < 0 \text{ if } y_i = -1$$





- Another way to write it?

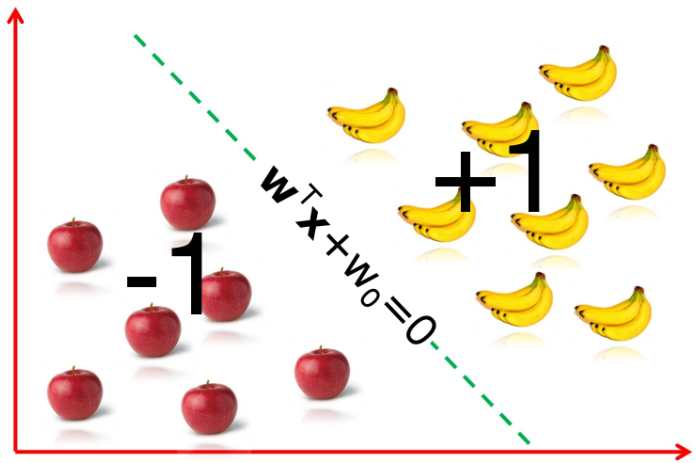
Binary classification problem – Generalized Linear Classifier

- Given training data $\{\mathbf{x}_i,\ y_i\}_{i\in\{1,\cdots,n\}}$
- Find hyperplane:

$$h = \mathbf{w}^\top \mathbf{x} + w_0 = 0$$

Subject to constraints for all i:

$$\mathbf{w}^\top \mathbf{x}_i + w_0 > 0 \text{ if } y_i = +1$$
$$\mathbf{w}^\top \mathbf{x}_i + w_0 < 0 \text{ if } y_i = -1$$



Decision function $f(\mathbf{x}) = \text{sign}(\mathbf{w}^\top \mathbf{x} + w_0)$

Binary classification problem – Generalized Linear Classifier

Given training data

$$\{\mathbf{x}_i,\ y_i\}_{i\in\{1,\cdots,n\}}$$

Decision function

$$f(\mathbf{x}) = \text{sign}(\mathbf{w}^\top \mathbf{x} + w_0)$$

And then?

Binary classification problem – Generalized Linear Classifier

Given training data

$$\{\mathbf{x}_i,\ y_i\}_{i\in\{1,\cdots,n\}}$$

Decision function

$$f(\mathbf{x}) = \text{sign}(\mathbf{w}^\top \mathbf{x} + w_0)$$

And then?

Find the optimal parameters $\hat{\mathbf{w}}^*$ and $\hat{w}_0^*$ which minimize the classification errors by using a loss function.

Binary classification problem – Generalized Linear Classifier

Loss

Definition

Empirical Risk Minimization

$$[\hat{\mathbf{w}}, \hat{w}_0] = \text{argmin}_{\mathbf{w},w_0} \left(\frac{1}{n} \sum_{i=1}^{n} \mathbf{1} \left(y_i \neq \text{sign}(\mathbf{w}^T \mathbf{x}_i + w_0) \right) \right)$$

- $\mathbf{1}(\cdot)$ is the indicator function
- Can be solved if data are linearly separable.
- **perceptron algorithm**
  https://www.cs.utexas.edu/~teammco/misc/perceptron/

Definition

Empirical Risk Minimization

$$[\hat{\mathbf{w}}, \hat{w}_0] = \operatorname{argmin}_{\mathbf{w}, w_0} \left(\frac{1}{n} \sum_{i=1}^{n} \mathbf{1} \left(y_i \neq \operatorname{sign}(\mathbf{w}^T \mathbf{x}_i + w_0) \right) \right)$$

- $\mathbf{1}(\cdot)$ is the indicator function
- Can be solved if data are linearly separable.
- **perceptron algorithm**
  https://www.cs.utexas.edu/~teammco/misc/perceptron/

The Perceptron algorithm is a simple **mistake-driven online algorithm**.

1. Start with a zero weight vector and process each training example in turn.
2. If the current weight vector classifies the current example incorrectly, move the weight vector in the right direction.
3. If weights stop changing, stop.

Homework: How is the perceptron algorithm implemented in practice?

Binary classification problem – Generalized Linear Classifier

Given training data

$$\{\mathbf{x}_i,\ y_i\}_{i\in\{1,\cdots,n\}}$$

Decision function:

$$f(\mathbf{x}) = \text{sign}(\mathbf{w}^\top \mathbf{x} + w_0)$$

Optimal parameters $\hat{\mathbf{w}}^*$ and $\hat{w}_0^*$.

And then?

Binary classification problem – Generalized Linear Classifier

Given training data

$$\{\mathbf{x}_i,\ y_i\}_{i\in\{1,\cdots,n\}}$$

Decision function:

$$f(\mathbf{x}) = \text{sign}(\mathbf{w}^\top \mathbf{x} + w_0)$$

Optimal parameters $\hat{\mathbf{w}}^*$ and $\hat{w}_0^*$.

Prediction:

$$\hat{f}(\mathbf{x}) = \text{sign}(\hat{\mathbf{w}}^{*\top} \mathbf{x} + \hat{w}_0^*)$$

Binary classification problem – Generalized Linear Classifier

Loss

Definition

Empirical Risk Minimization 0-1 Loss

$$[\hat{\mathbf{w}}, \hat{w}_0] = \text{argmin}_{\mathbf{w}, w_0} \left(\frac{1}{n} \sum_{i=1}^{n} \mathbf{1} \left(y_i \neq \text{sign}(\mathbf{w}^T \mathbf{x}_i + w_0) \right) \right)$$

- Can be solved if data are linearly separable. E.g. perceptron algorithm
- Problem?

Binary classification problem – Generalized Linear Classifier

Loss

Definition

Empirical Risk Minimization 0-1 Loss

$$[\hat{\mathbf{w}}, \hat{w}_0] = \text{argmin}_{\mathbf{w}, w_0} \left(\frac{1}{n} \sum_{i=1}^{n} \mathbf{1} \left(y_i \neq \text{sign}(\mathbf{w}^T \mathbf{x}_i + w_0) \right) \right)$$

- Can be solved if data are linearly separable. E.g. perceptron algorithm
- Problem?
  - if data is not linearly separable $\rightarrow$ NP-hard!
  - Loss 0-1 is not convex and its gradient is either 0 or undefined.
  - Different **w** solutions may have the same loss.

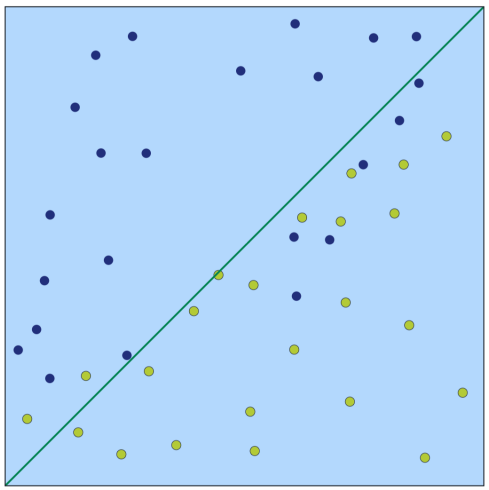
Binary classification problem – Generalized Linear Classifier

Loss

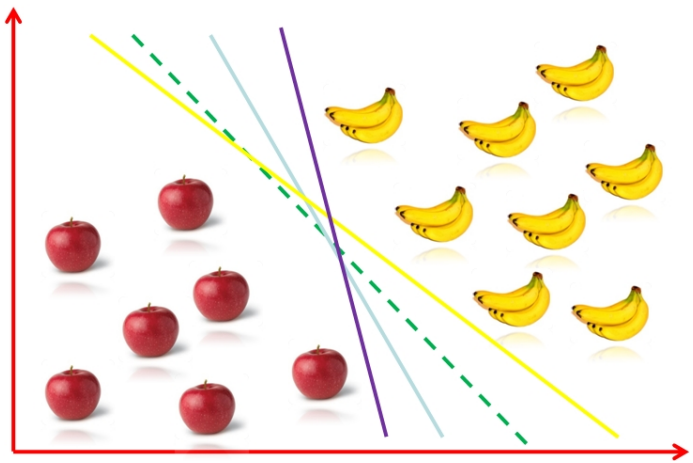
Definition

Empirical Risk Minimization

$$[\hat{\mathbf{w}}, \hat{w}_0] = \text{argmin}_{\mathbf{w},w_0} \left(\frac{1}{n} \sum_{i=1}^{n} \mathbf{1} \left(y_i \neq \text{sign}(\mathbf{w}^T \mathbf{x}_i + w_0) \right) \right)$$



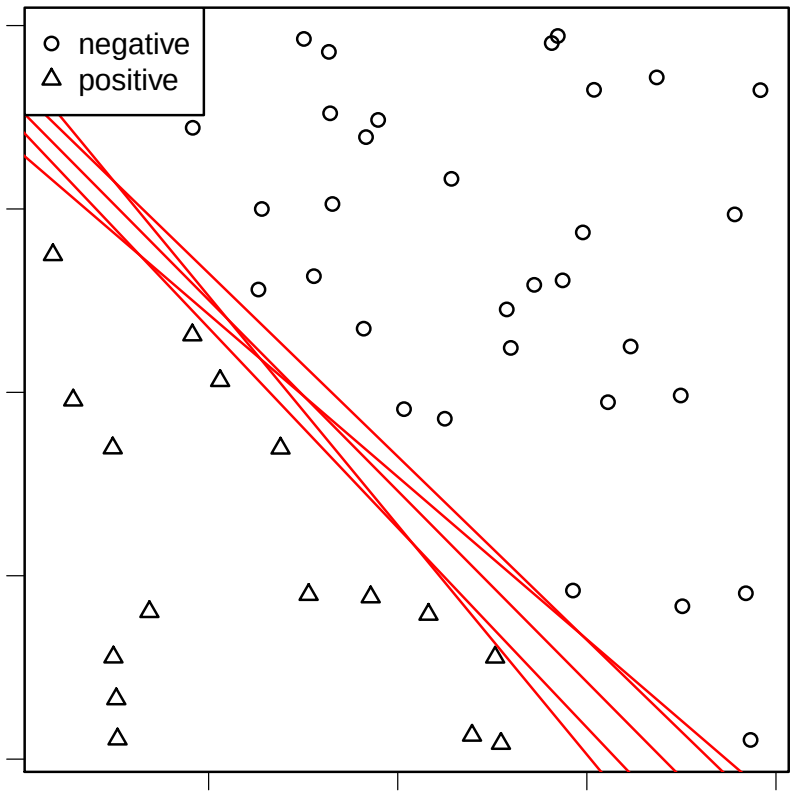
NP-hard



trop de solutions?

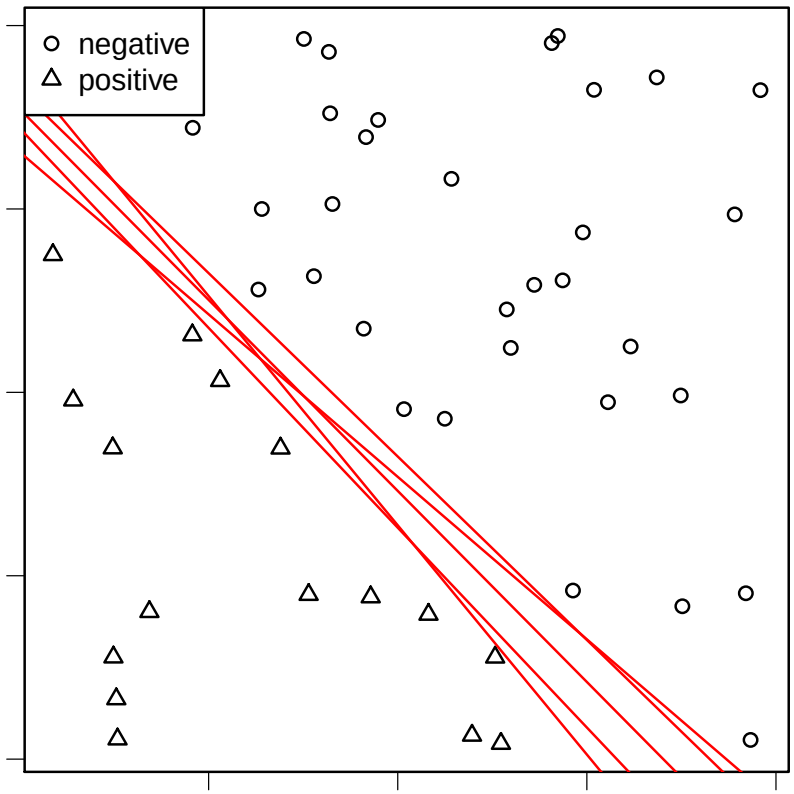
Optimal Separating Hyperplanes

- Consider a binary classification problem where two classes are optimally separable.
- A lot of hyperplanes solve this problem but which one is the best?
- Solution?



Optimal Separating Hyperplanes

- Consider a binary classification problem where two classes are optimally separable.
- A lot of hyperplanes solve this problem but which one is the best?
- **Intuition:** the **margin** separating both classes has to be **maximized**.



The geometric margin

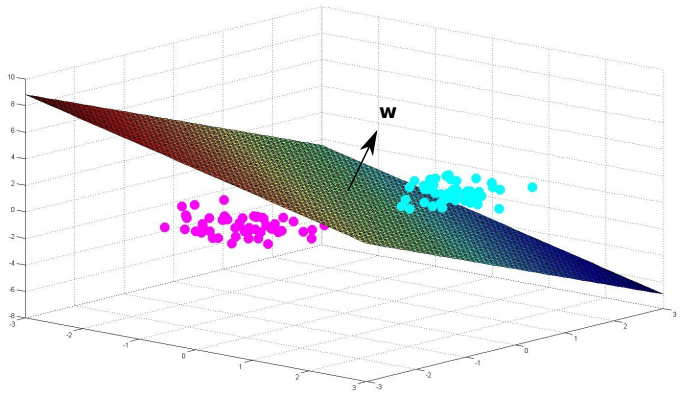
- Hyperplane equation:

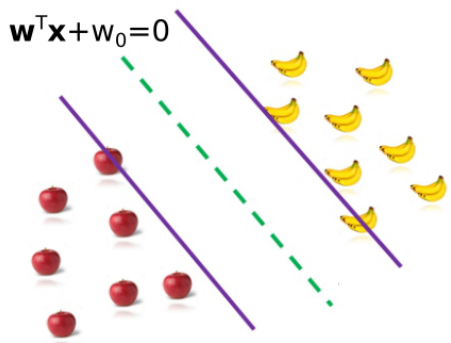
  $h(\mathbf{x}) = w_0 + \mathbf{x}^\top \mathbf{w}$

- To find the decision boundary:

  $h(\mathbf{x}) = 0$

- $\mathbf{w}$ is orthogonal to the hyperplane.
  - $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^m$ two points on the hyperplane,
  - $(\mathbf{x}_1 - \mathbf{x}_2)^\top \mathbf{w} = 0$





The geometric margin

- Hyperplane equation:

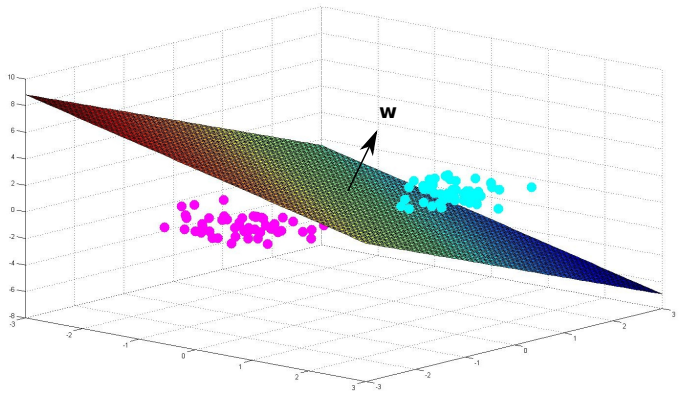
  $h(\mathbf{x}) = w_0 + \mathbf{x}^\top \mathbf{w}$

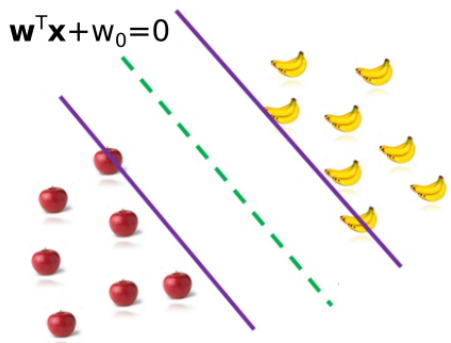
- To find the decision boundary:

  $h(\mathbf{x}) = 0$

- $\mathbf{w}$ is orthogonal to the hyperplane.
  - $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^m$ two points on the hyperplane,
  - $(\mathbf{x}_1 - \mathbf{x}_2)^\top \mathbf{w} = 0$

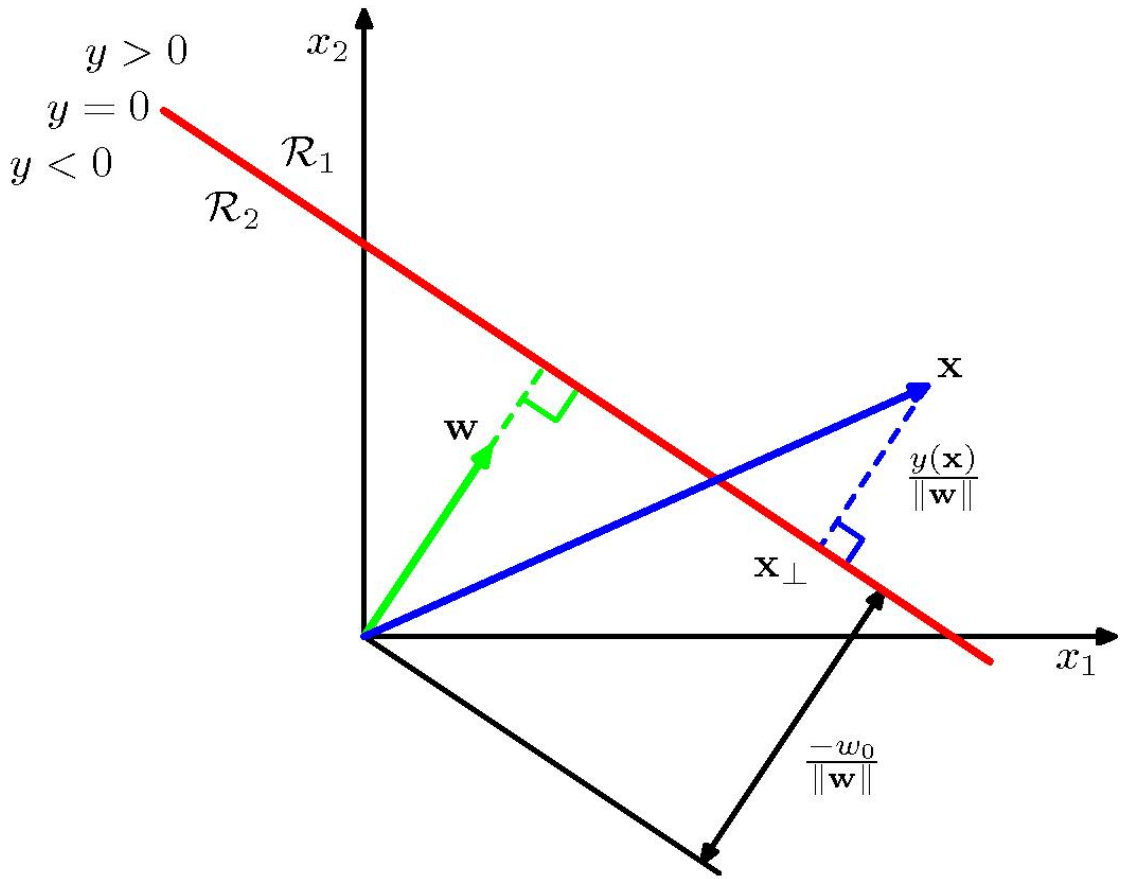
- Signed distance of a point to a hyperplane?





The geometric margin

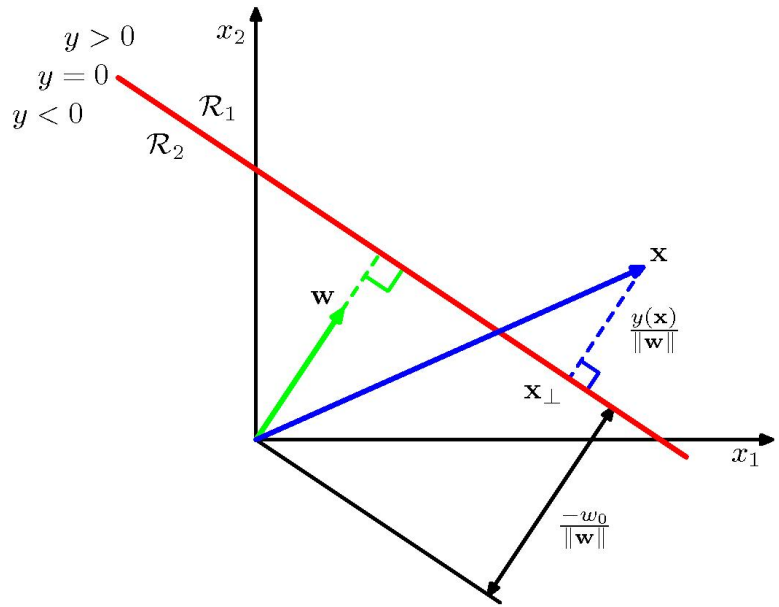
Signed distance of a point to a hyperplane?



The geometric margin

- $\mathbf{x}_{i\perp}$ is the projection of $\mathbf{x}_i$ onto the hyperplane
- $\tilde{\mathbf{x}}_i = (\mathbf{x}_i - \mathbf{x}_{i\perp})$ a vector from the hyperplane to point $\mathbf{x}_i$
- Project $\tilde{\mathbf{x}}_i$ onto $\mathbf{w}$

$$\text{proj}_{\mathbf{w}}\tilde{\mathbf{x}}_i = \frac{\tilde{\mathbf{x}}_i \cdot \mathbf{w}}{\|\mathbf{w}\|} = \frac{(\mathbf{x}_i - \mathbf{x}_{i\perp})^\top \mathbf{w}}{\|\mathbf{w}\|}$$



Bishop, 2006

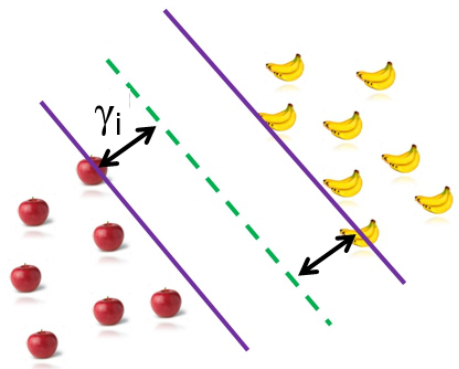
Since $\mathbf{x}_{i\perp}$ on the plane $w_0 + \mathbf{x}_{i\perp}^\top \mathbf{w} = 0$

$$\text{proj}_{\mathbf{w}}\tilde{\mathbf{x}}_i = \frac{\mathbf{x}_i^\top \mathbf{w} - \cancel{\mathbf{x}_{i\perp}^\top \mathbf{w}}}{\|\mathbf{w}\|} = \frac{\mathbf{x}_i^\top \mathbf{w} + w_0}{\|\mathbf{w}\|}$$

The geometric margin

Margin: Signed distance of a point to the hyperplane:

$$\gamma_i = \frac{y_i \left(\mathbf{w}^\top \mathbf{x}_i + w_0\right)}{\|\mathbf{w}\|}$$

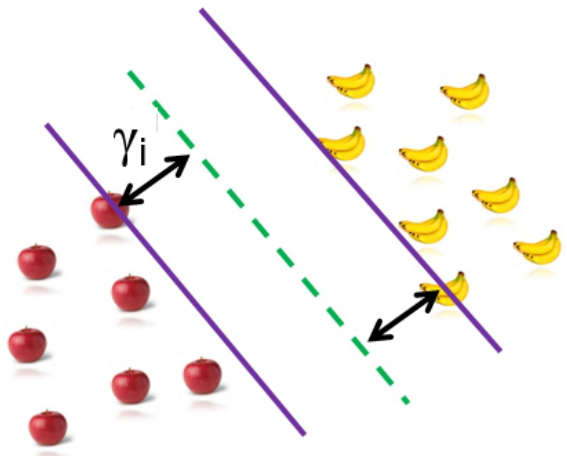


The geometric margin

Definition

Margin: smallest distance over all points

$$\gamma = \min_i (\gamma_i) = \min_i \left(\frac{y_i \left(\mathbf{w}^\top \mathbf{x}_i + w_0 \right)}{\|\mathbf{w}\|} \right)$$



Optimal Separating Hyperplanes

Question

How to define the classification problem in terms of the margin?

Optimal Separating Hyperplanes

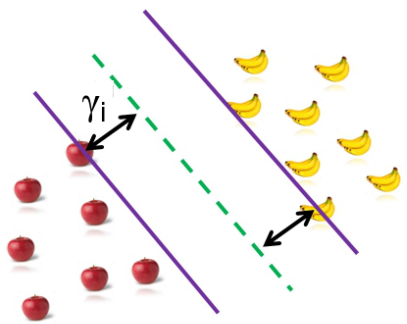
Goal

$$\mathbf{max}_{w_0,\mathbf{w}}\ \gamma$$

$$\mathbf{s.t.}\ \frac{1}{\|\mathbf{w}\|} y_i(\mathbf{w}^\top \mathbf{x}_i + w_0) \geq \gamma, \quad i = 1, \ldots, n$$

Find a hyperplane that

- separates the two classes
- maximizes the distance to the closest point from either class

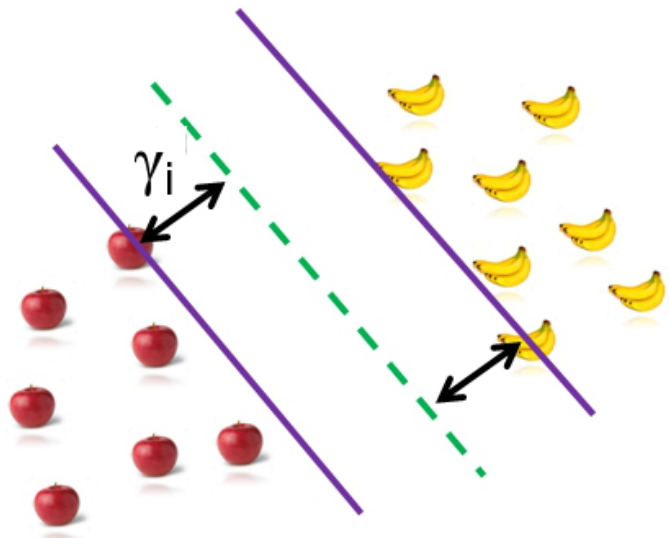


Optimal Separating Hyperplanes

Goal

$$\max_{w_0, \mathbf{w}} \gamma$$

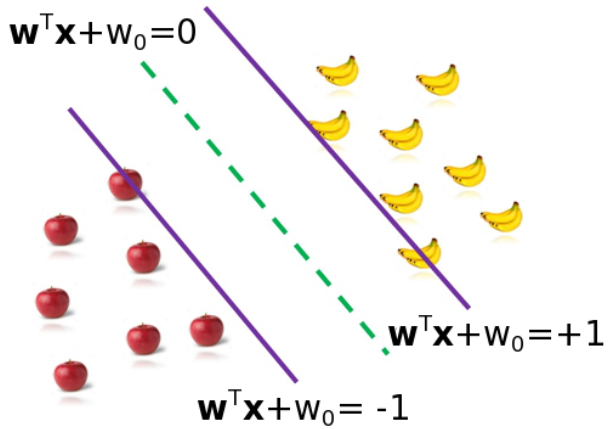
$$\text{s.t.} \quad \frac{1}{\|\mathbf{w}\|} y_i(\mathbf{w}^\top \mathbf{x}_i + w_0) \geq \gamma, \quad i = 1, \ldots, n$$



Ex:bananas and apples

Optimal Separating Hyperplanes

s.t. $\frac{1}{\|\mathbf{w}\|} y_i(\mathbf{w}^\top \mathbf{x}_i + w_0) \geq \gamma, \quad i = 1, \ldots, n$



Since changing the scaling of $\mathbf{w}$ and w_0 does not change the margin, we can arbitrarily set: $y_i \left(\mathbf{w}^\top \mathbf{x}_i + w_0\right) \geq 1$

Margin

$$\gamma = \frac{1}{\|\mathbf{w}\|}$$

which leads to an easier equivalent problem.

Optimal Separating Hyperplanes

$$\max \frac{1}{\|\mathbf{w}\|} \text{ is equivalent to } \min \|\mathbf{w}\|^2$$

Maximum margin problem

- **Minimize** $\frac{1}{2}\|\mathbf{w}\|^2$
- **subject to**: $y_i\left(\mathbf{w}^\top \mathbf{x}_i + w_0\right) \geq 1$ for all $i = 1, \ldots, n$

Optimal Separating Hyperplanes – Optimization

Maximum margin problem

- **Minimize** $\frac{1}{2}\|\mathbf{w}\|^2$
- **subject to**: $y_i\left(\mathbf{w}^\top \mathbf{x}_i + w_0\right) \geq 1$ for all $i = 1, \ldots, n$

- The functional to miminize $f(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2$ is convex,
- The **constraints** $1 - y_i\left(\mathbf{w}^\top \mathbf{x}_i + w_0\right) \leq 0$ are **linear**.

Can be solved with convex optimization, e.g. using **quadratic programming**! (`quadprog` in both python and matlab)

Optimal Separating Hyperplanes – Optimization

Maximum margin problem

- **Minimize** $\frac{1}{2}\|\mathbf{w}\|^2$
- **subject to**: $y_i\left(\mathbf{w}^\top \mathbf{x}_i + w_0\right) \geq 1$ for all $i = 1, \ldots, n$

Construct the **Lagrangian** :

Lagrangian

$$\mathcal{L}(\mathbf{w}, w_0, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 + \sum_{i=1}^{n} \alpha_i \left(1 - y_i(\mathbf{w}^\top \mathbf{x}_i + w_0)\right)$$

Optimal Separating Hyperplanes – Optimization

It is a Constrained Optimization with an inequality constraints.

With g_i the constraint $g_i(\mathbf{w}) = 1 - y_i(\mathbf{w}^\top \mathbf{x}_i + w_0)$.

the solution $\hat{\mathbf{w}}^*, w_0^*, \alpha_i^* \forall i$ has to verify the **KKT Conditions:**

1. Lagrangian Stationarity: $\frac{\partial \mathcal{L}(\hat{\mathbf{w}}^*, \hat{w}_0^*, \alpha^*)}{\partial \mathbf{w}} = 0$ & $\frac{\partial \mathcal{L}(\hat{\mathbf{w}}^*, \hat{w}_0^*, \alpha^*)}{\partial w_0} = 0$
2. Primal Feasibility: $g_i(\mathbf{w}^*) \leq 0$
3. Dual Feasibility: $\alpha_i^* \geq 0$
4. Complementary Slackness: $\alpha_i^* g_i(\mathbf{w}^*) = 0$

KKT conditions establish a generalization of Lagrange multipliers, for inequality constraints.

Optimal Separating Hyperplanes – Dual

Primal

$$p^* = \min_{\mathbf{w}, w_0} \mathcal{P}(\mathbf{w})$$

$$= \min_{\mathbf{w}, w_0} \max_{\alpha : \alpha_i \geq 0} \mathcal{L}(\mathbf{w}, w_0, \alpha)$$

Dual

$$d^* = \max_{\alpha : \alpha_i \geq 0} \mathcal{D}(\alpha)$$

$$= \max_{\alpha : \alpha_i \geq 0} \min_{\mathbf{w}, w_0} \mathcal{L}(\mathbf{w}, w_0, \alpha)$$

Optimal Separating Hyperplanes – Dual

Primal

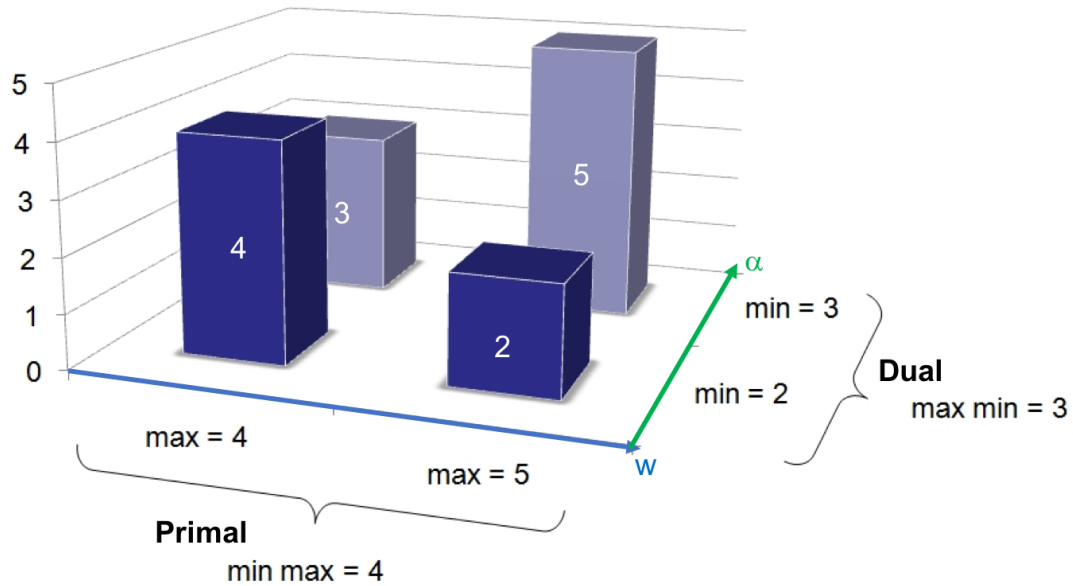
$$p^* = \min_{\mathbf{w}, w_0} \mathcal{P}(\mathbf{w})$$

$$= \min_{\mathbf{w}, w_0} \max_{\alpha : \alpha_i \geq 0} \mathcal{L}(\mathbf{w}, w_0, \alpha)$$

Dual

$$d^* = \max_{\alpha : \alpha_i \geq 0} \mathcal{D}(\alpha)$$

$$= \max_{\alpha : \alpha_i \geq 0} \min_{\mathbf{w}, w_0} \mathcal{L}(\mathbf{w}, w_0, \alpha)$$



Optimal Separating Hyperplanes – Dual

Primal

$$p^* = \min_{\mathbf{w}, w_0} \mathcal{P}(\mathbf{w})$$

$$= \min_{\mathbf{w}, w_0} \max_{\alpha : \alpha_i \geq 0} \mathcal{L}(\mathbf{w}, w_0, \alpha)$$

Dual

$$d^* = \max_{\alpha : \alpha_i \geq 0} \mathcal{D}(\alpha)$$

$$= \max_{\alpha : \alpha_i \geq 0} \min_{\mathbf{w}, w_0} \mathcal{L}(\mathbf{w}, w_0, \alpha)$$

Relation formulations primale/duale

$$d^* \leq p^*$$

if the **KKT conditions** as well as the **strong duality** conditions are verified, which is the case: $d^* = p^*$

Optimal Separating Hyperplanes – Dual

To find the dual of our problem, we need to express the problem in terms of the dual variables α. To do so, we look at the **minimal** conditions of the Lagrangian with respect to $\mathbf{w}$ and w_0:

$$\mathcal{L}(\mathbf{w}, w_0, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 + \sum_{i=1}^{n} \alpha_i \left(1 - y_i(\mathbf{w}^\top \mathbf{x}_i + w_0)\right)$$

- $\frac{\partial \mathcal{L}}{\partial \mathbf{w}} =$
- $\frac{\partial \mathcal{L}}{\partial w_0} =$

By plugging $\mathbf{w}$ in the Lagrangian (and after some simplifications:

Dual

$$\mathcal{D}(\alpha) =$$

Optimal Separating Hyperplanes – Dual

To find the dual of our problem, we need to express the problem in terms of the dual variables α. To do so, we look at the **minimal** conditions of the Lagrangian with respect to $\mathbf{w}$ and w_0:

$$\mathcal{L}(\mathbf{w}, w_0, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 + \sum_{i=1}^{n} \alpha_i \left(1 - y_i(\mathbf{w}^\top \mathbf{x}_i + w_0)\right)$$

- $\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i = 0$

  $\Rightarrow \mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$

- $\frac{\partial \mathcal{L}}{\partial w_0} = \sum_{i=1}^{n} \alpha_i y_i = 0$

By plugging $\mathbf{w}$ in the Lagrangian (and after some simplifications:

Dual

$$\mathcal{D}(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,k=1}^{n} y_i y_k \alpha_i \alpha_k \mathbf{x}_i^\top \mathbf{x}_k$$

s.t $\sum_{i=1}^{n} \alpha_i y_i = 0$

Optimal Separating Hyperplanes – Dual

To find the dual of our problem, we need to express the problem in terms of the dual variables α. To do so, we look at the **minimal** conditions of the Lagrangian with respect to $\mathbf{w}$ and w_0:

$$\mathcal{L}(\mathbf{w}, w_0, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 + \sum_{i=1}^{n} \alpha_i \left(1 - y_i(\mathbf{w}^\top \mathbf{x}_i + w_0)\right)$$

- $\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i = 0$
  $\Rightarrow \mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$
- $\frac{\partial \mathcal{L}}{\partial w_0} = \sum_{i=1}^{n} \alpha_i y_i = 0$

By plugging $\mathbf{w}$ in the Lagrangian (and after some simplifications:

Dual

$$\mathcal{D}(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,k=1}^{n} y_i y_k \alpha_i \alpha_k \mathbf{x}_i^\top \mathbf{x}_k$$

s.t $\sum_{i=1}^{n} \alpha_i y_i = 0$

Multiplying sums should be done with different indices. The constraint on the sum of $\sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i = 0$

Optimal Separating Hyperplanes – Dual

The advantages of the dual formulation are:

- The α_i enable **identifying the support vectors** (the training points that are responsible for classifying a new point).
- The formulation in terms of a scalar product between data points allows using the **kernel trick**
- The primal and the dual problems are same for convex quadratic programming problems with linear inequality constraints (there is **no duality gap**).
- The dual problem is simpler to optimize

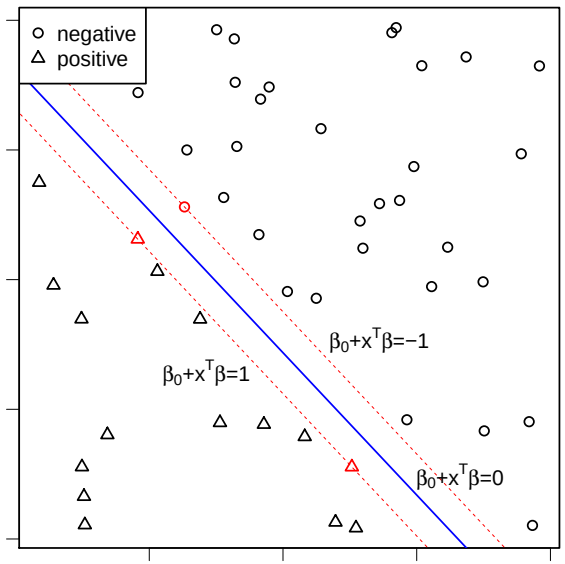
Optimal Separating Hyperplanes

After optimization the resultant hyperplane is defined by weights $\hat{\mathbf{w}}$

$$\hat{\mathbf{w}}^* = \sum_i^n \hat{\alpha}_i^* y_i \mathbf{x}_i$$

Notice that $\hat{\mathbf{w}}$ are a **linear combination** of the data points !

$$\hat{h}^*(\mathbf{x}) = \sum_{i=1}^n \hat{\alpha}_i^* y_i \mathbf{x}_i^\top \mathbf{x} + \hat{w}_0^*$$



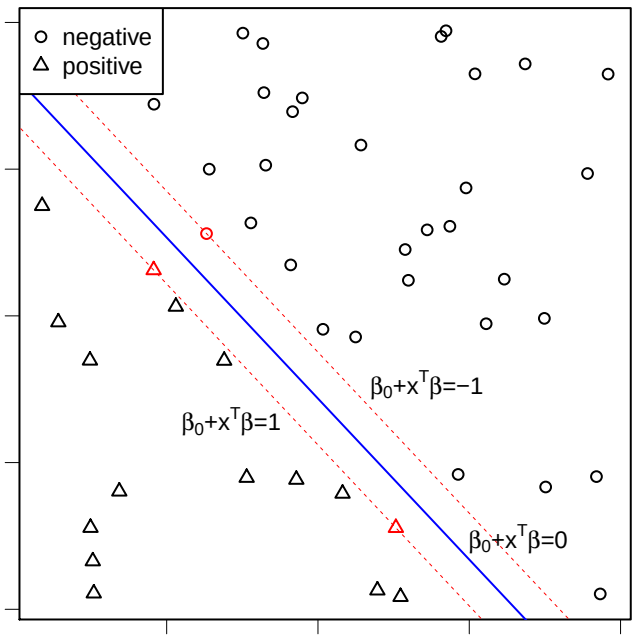
Optimal Separating Hyperplanes – Support Points

$$\hat{h}^*(\mathbf{x}) = \sum_{i=1}^{n} \hat{\alpha}_i^* y_i \mathbf{x}_i^\top \mathbf{x} + \hat{w}_0^*$$

$\hat{\alpha}_i^* \geq 0$ are **weights** such that

- $\hat{\alpha}_i^* \neq 0$ if $\mathbf{x}_i$ **support vector**
- $\hat{\alpha}_i^* = 0$ otherwise.

The solution only depends on the support points not on the whole data set!!!

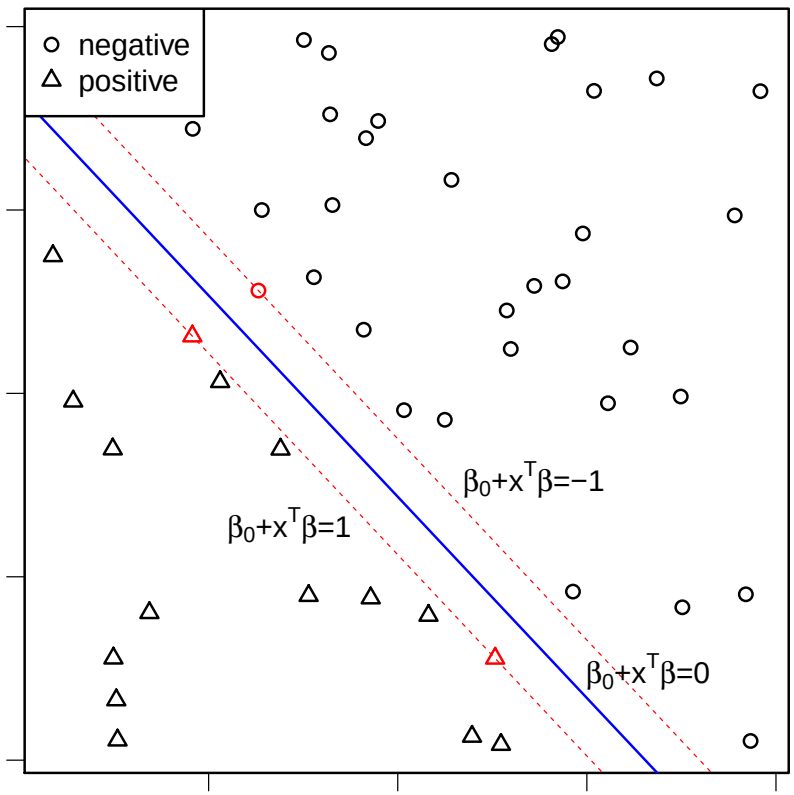


Optimal Separating Hyperplanes – Prediction

A new sample is classified by

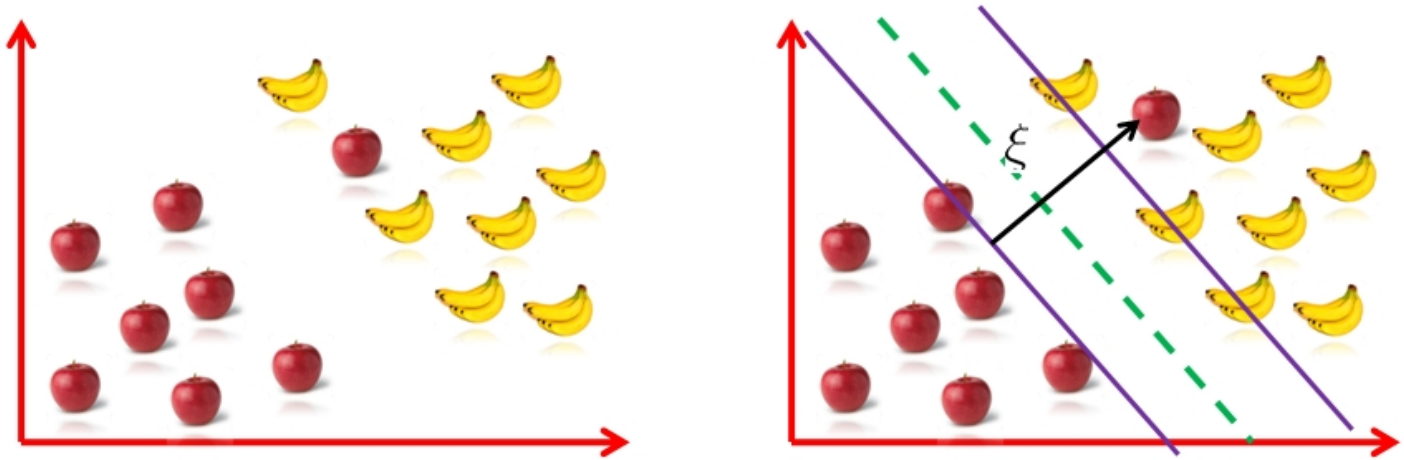
$$f(\mathbf{x}_k) = \text{sign}(\hat{h}^*(\mathbf{x}_k))$$

$$f(\mathbf{x}_k) = \text{sign}(\hat{w}_0^* + \mathbf{x}_k^\top \hat{\mathbf{w}}^*)$$



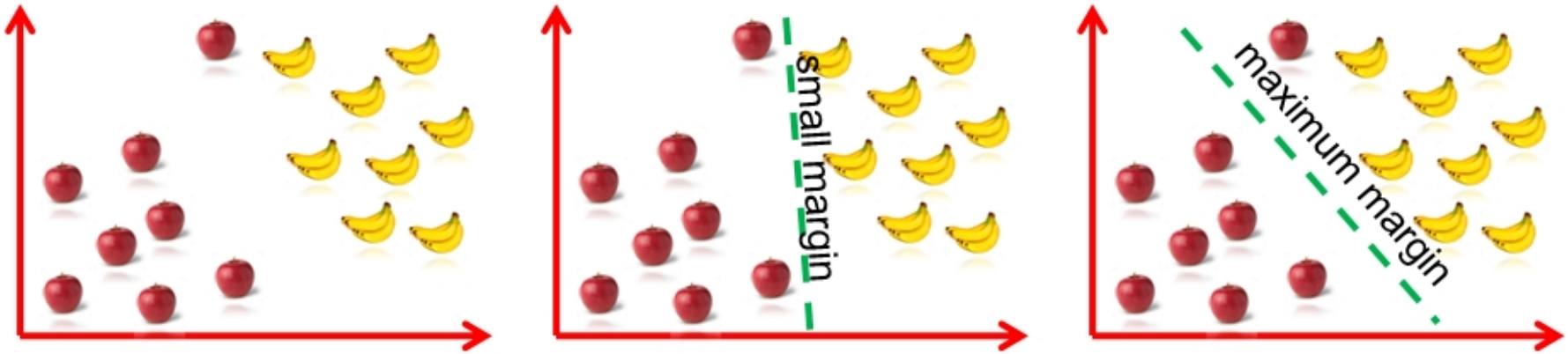
Outline

Support Vector Machines – Linear separability



- In real applications classes are rarely **linearly separable**. Usually, two classes **overlap** in feature space.
- In addition: Noise in the features? Mislabelled data? Outliers?
- **Soft margin**: allow **margin violations**, i.e. misclassification errors

Support Vector Machines – Soft margin



Soft margin:

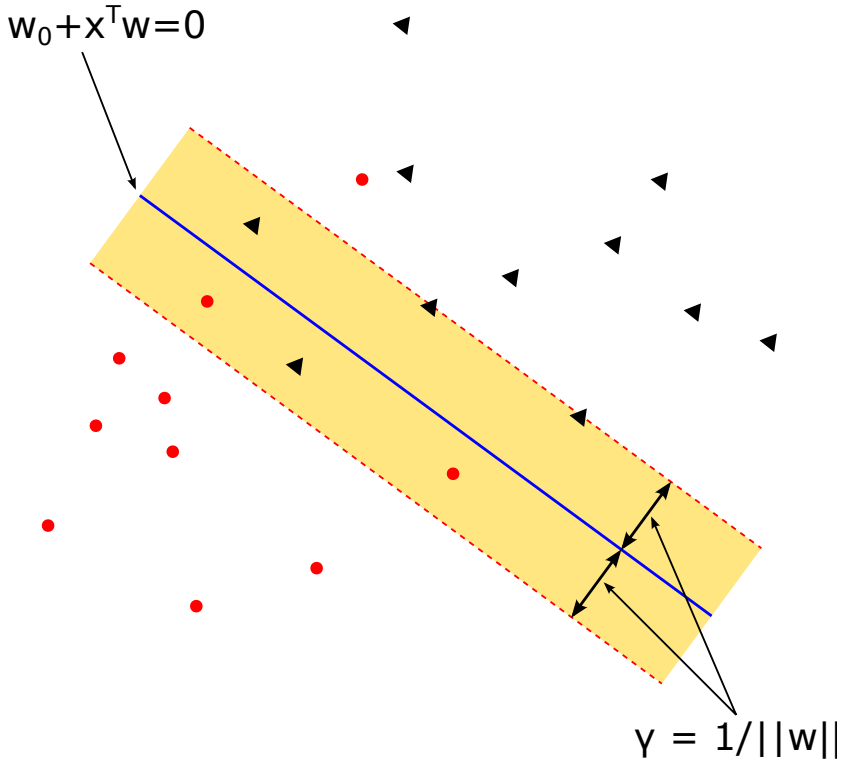
- Allows to find a solution in case of no-linear separability.
- Provides a better **generalization** ability:
  - Deals with noisy data and outliers.
  - Prevents from **overfitting** the data, i.e. prevents the model of describing random errors or noise.

Support Vector Machines – Soft margin

Soft Margin

Still maximise the margin but allow for some points to reside on the wrong side of the margin (**soft margin**).

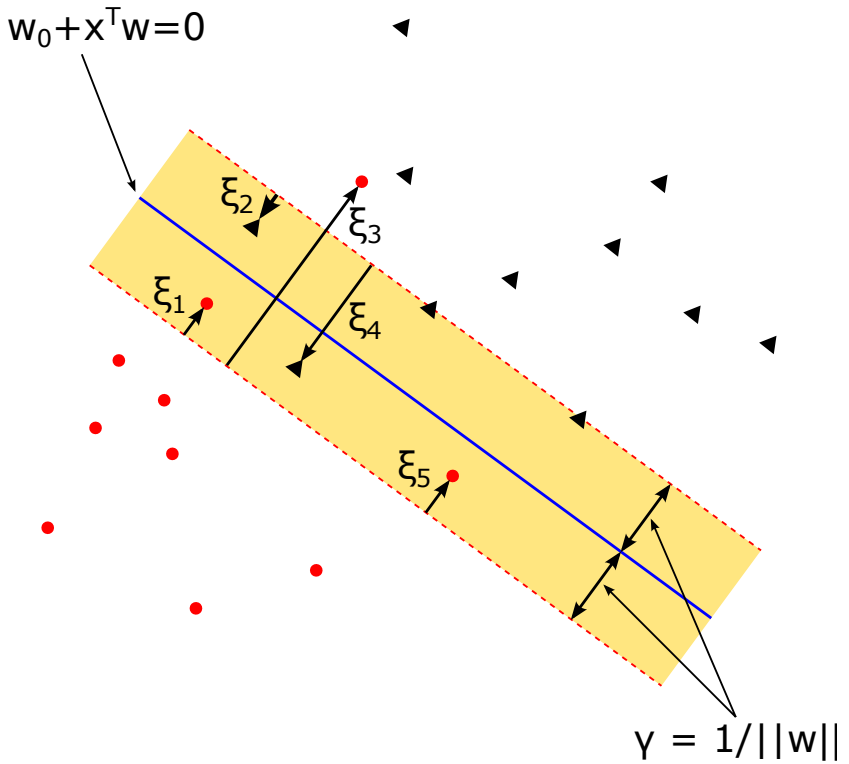
- Quels points?
- Comment viter une solution arbitraire?



Support Vector Machines – Slack variables

Introduce for each sample a **slack variable** $\xi_i \geq 0$:

- it is always positive $\xi_i \geq 0$
- if $\xi_i = 0$: **correct** classification
- if $\xi_i > 1$: **missclassification**
- if $0 < \xi_i \leq 1$: point lies between the margin and the correct side of the margin.



ξ_i gives the relative amount, with respect to the margin, by which the prediction falls on the wrong side of the hyperplane.

Support Vector Machines

Definition (SVM Optimization)

$$\min_{w_0, \mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \xi_i$$

subject to

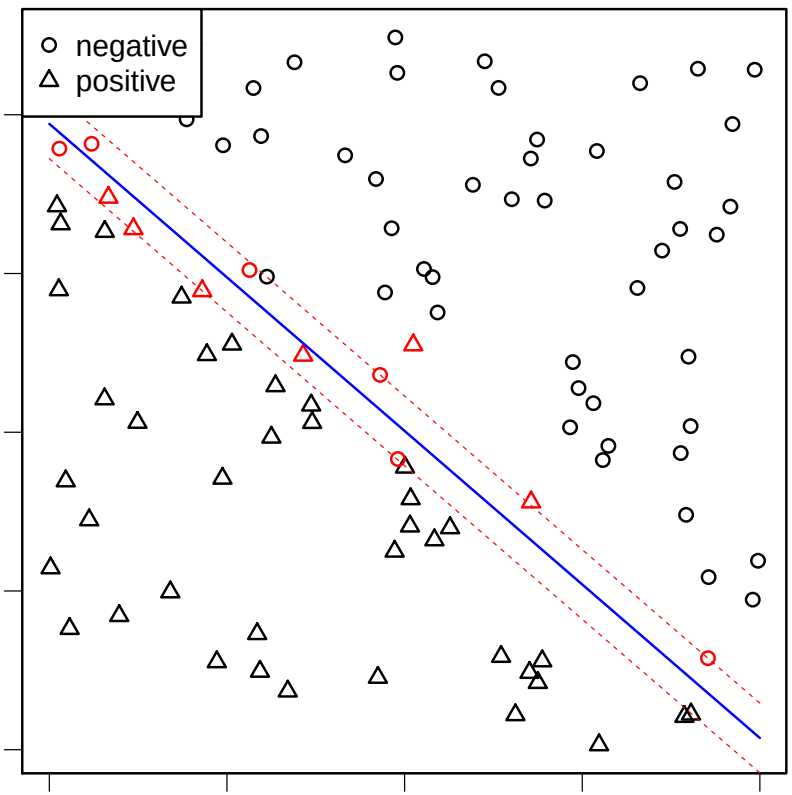
$$\xi_i \geq 0 \, \forall i$$

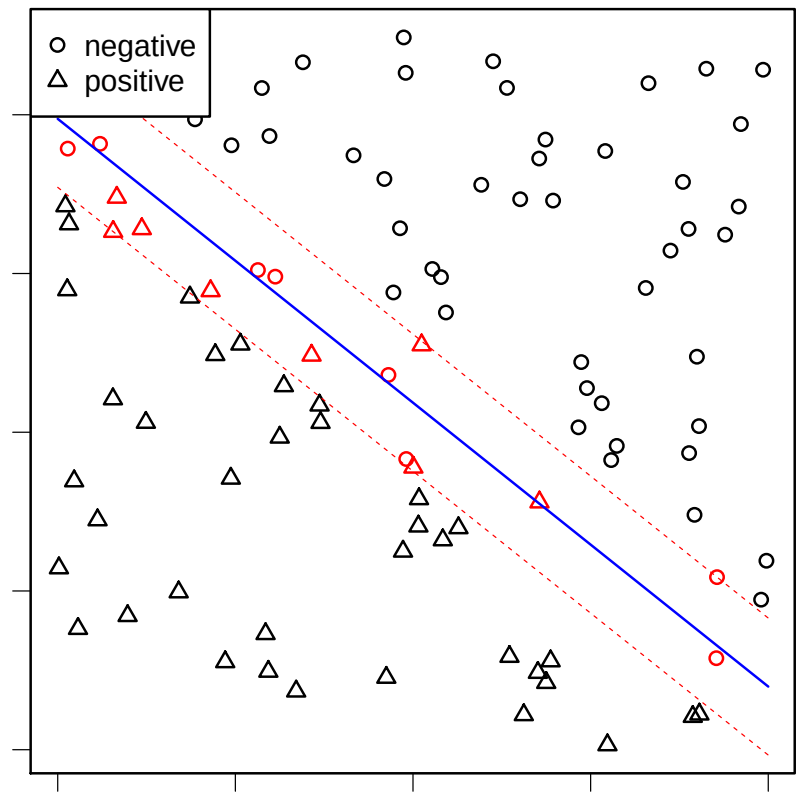
$$y_i(w_0 + \mathbf{x}_i^\top \mathbf{w}) \geq 1 - \xi_i \, \forall i$$

- The parameter $C > 0$ controls the trade-off between the slack variable penalty and the margin.
- If $C = \infty$, result equal to optimal separating hyperplanes.
- $\sum \xi_i$ is an upper bound on the number of misclassified points.

Support Vector Machines – Examples

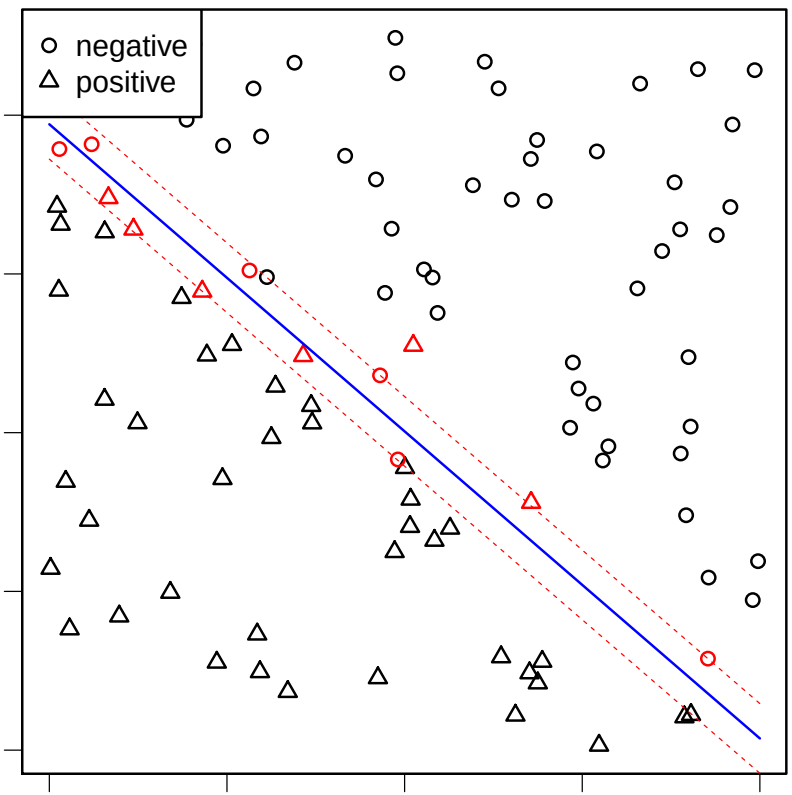
C ???



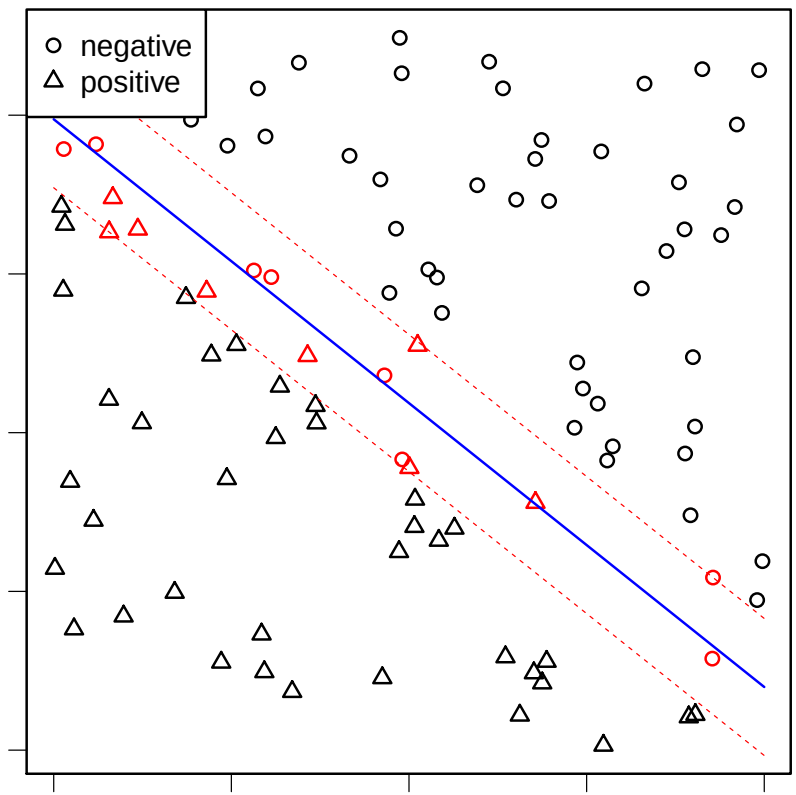


Support Vector Machines – Examples

$C = 10000$



$C = 1$



Support Vector Machines – Optimization

Regularized maximum margin

- **Minimize** $\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i$
- **subject to**: $y_i\left(\mathbf{w}^\top \mathbf{x}_i + w_0\right) \geq 1 - \xi_i$ for all i
  $\xi_i \geq 0$ for all i

How to solve this Convex Optimization Problem

- Use **Lagrange Multipliers** to convert a constrained optimization problem into an unconstrained one.
- Introduce new variables $\alpha_i \geq 0$ $r_i \geq 0$ to weight the constraints in a unique cost function.

SVM optimization

Lagrangian

$$\mathcal{L}(\mathbf{w}, w_0, \xi, \alpha, r) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i + \sum_{i=1}^{n}\alpha_i\left(1 - \xi_i - y_i(\mathbf{w}^\top \mathbf{x}_i + w_0)\right) - \sum_{i=1}^{n} r_i\xi_i$$

Use **Lagrange duality** ([Bishop 2006]) and an efficient method **Sequential Minimal Optimization** (SMO)

SVM optimization

Dual form of the problem

$$\max_{\alpha} \mathcal{D}(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,k=1}^{n} y_i y_k \alpha_i \alpha_k \langle \mathbf{x}_i, \mathbf{x}_k \rangle$$

subject to:

$$0 \leq \alpha_i \leq C, \ i = 1, \cdots, n$$
$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

Sequential Minimal Optimization (SMO):

- Do not solve the problem for all variables.
- Fix all $\alpha_1, \cdots, \alpha_n$ but one.
- Optimize the $\mathcal{D}$ according to this variable α_i.

Support Vector Machines– Prediction

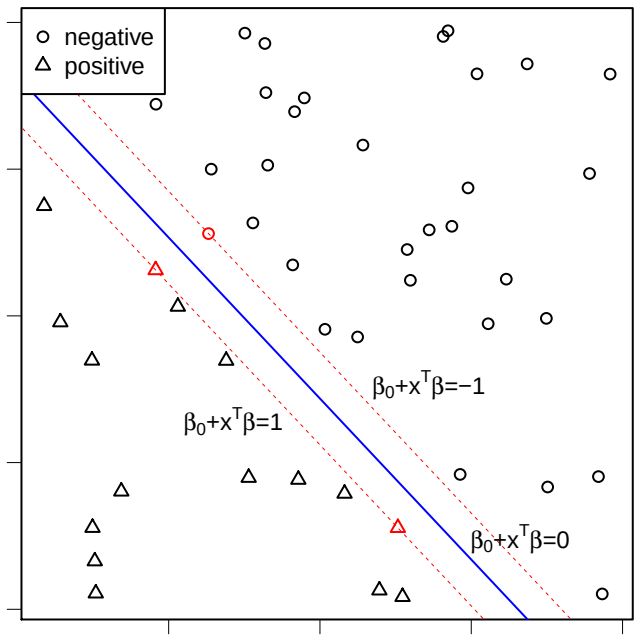
At optimum $\hat{\mathbf{w}}^*$ can be estimated as

$$\hat{\mathbf{w}}^* = \sum_{i=1}^{n} \alpha_i^* y_i \mathbf{x}_i$$

where $\boldsymbol{\alpha} \in \mathbb{R}^n$

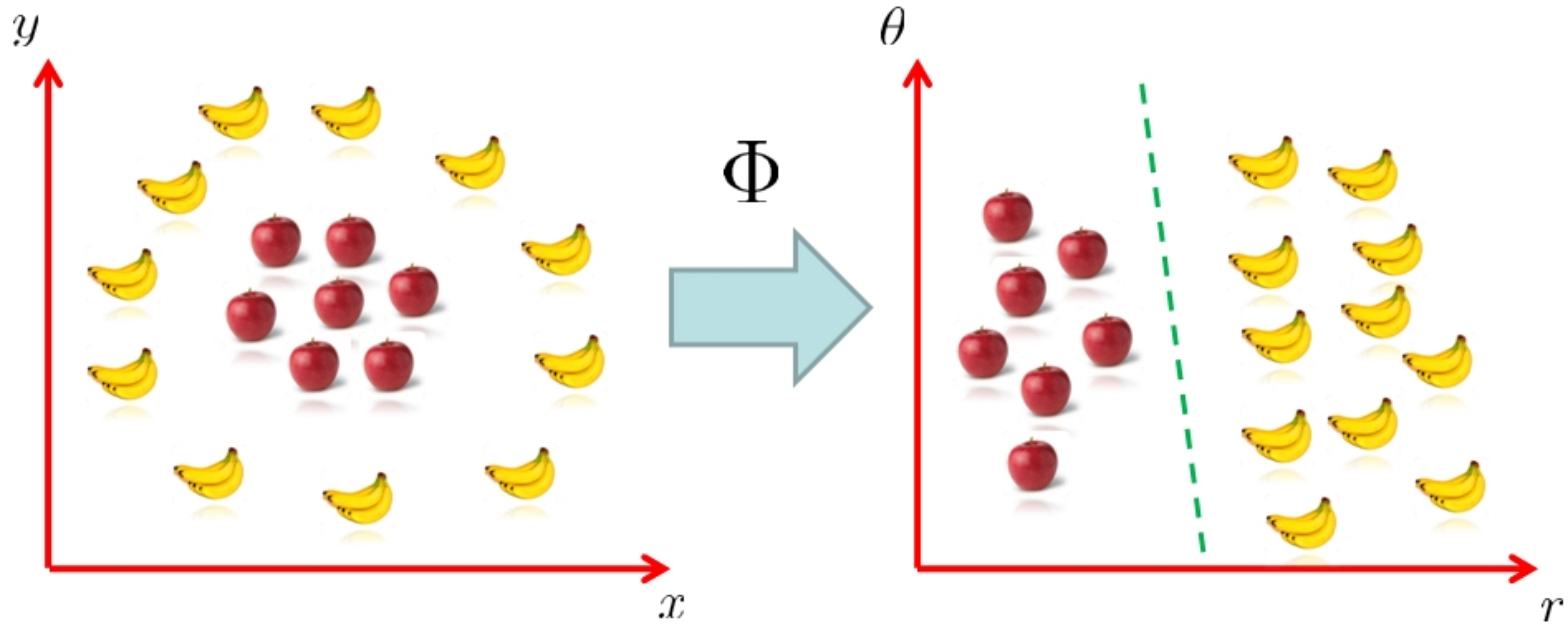
Prediction

$$\hat{f}^*(\mathbf{x}_k) = \text{sign}(\hat{h}^*(\mathbf{x}_k))$$
$$\hat{f}^*(\mathbf{x}_k) = \text{sign}(\hat{w}_0^* + \mathbf{x}_k^\top \hat{\mathbf{w}}^*)$$
$$\hat{f}^*(\mathbf{x}_k) = \text{sign}(\hat{w}_0^* + \sum_{i=1}^{n} \hat{\alpha}_i^* y_i \langle \mathbf{x}_k, \mathbf{x}_i \rangle)$$



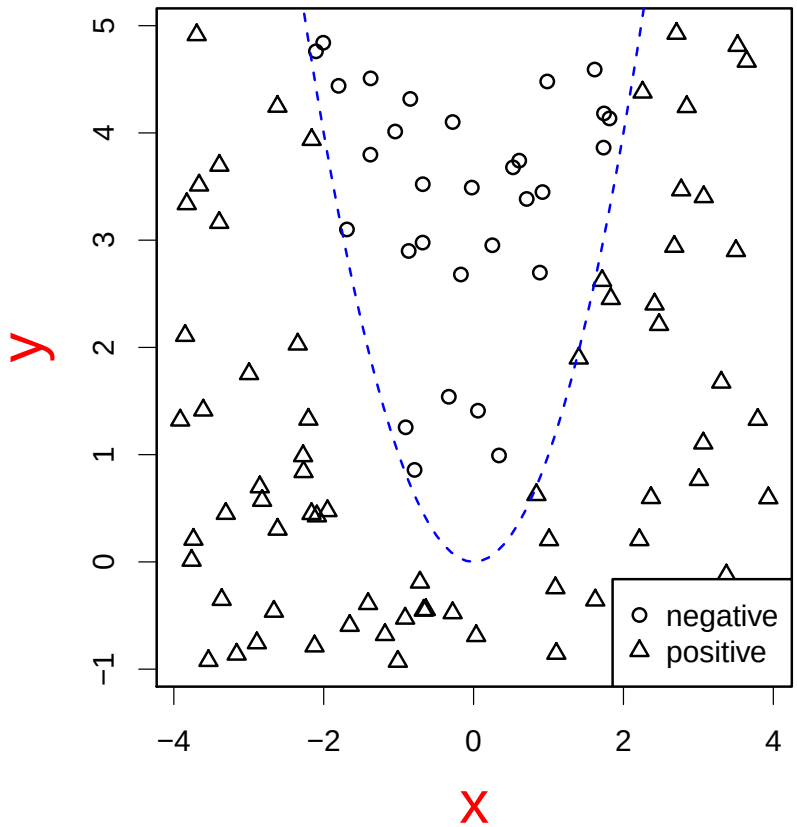
Outline

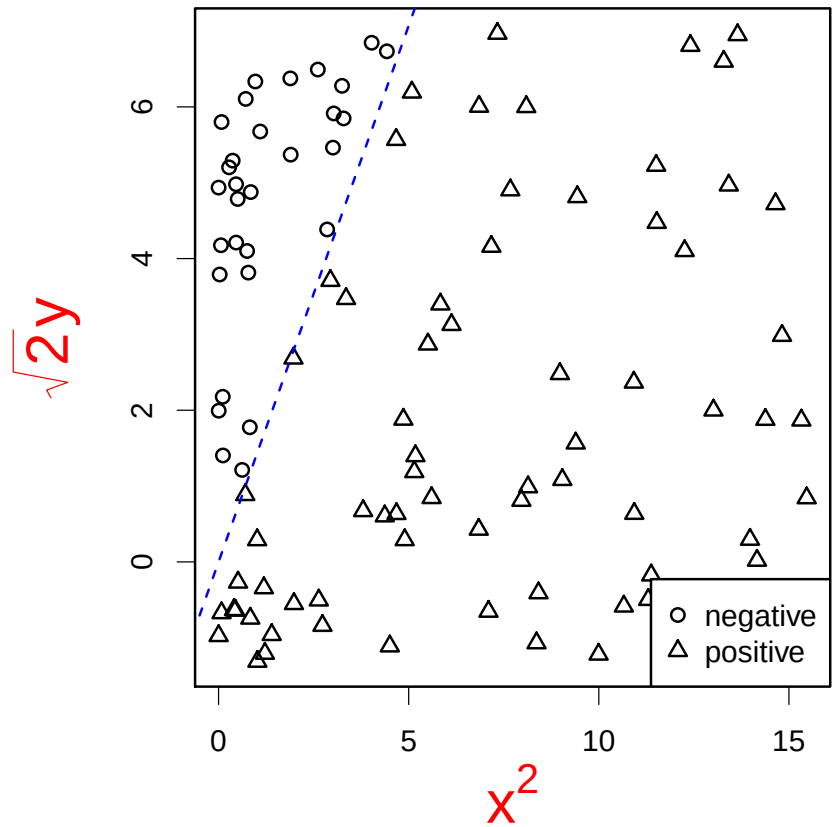
Non-linear separability



- Data are not always **linearly separable**, however...they may be separable in another (higher dimensional) space!
- **Idea**: find a **non-linear mapping** from the input space into a (higher dimensional) feature space in which data are separable.

Non-linear SVMs– Transformation





Example: Transform point (x, y) to $(x^2, \sqrt{2}y)$ where the data can be separated linearly.

Non-linear SVMs – Transformation

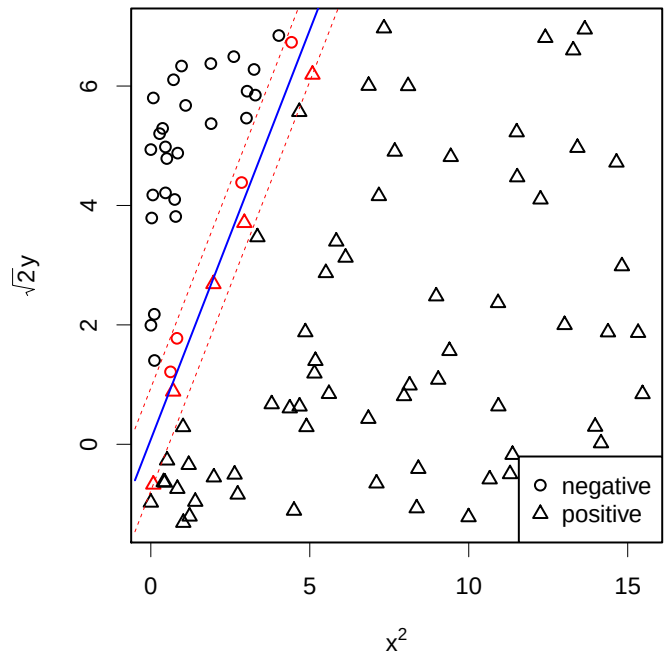
Map data:

- from the input space $\mathcal{X} \subseteq \mathbb{R}^d$
- to feature space $\mathcal{F} \subseteq \mathbb{R}^D$
- using a **non-linear function**

$$\phi : \mathcal{X} \to \mathcal{F}$$

The hyperplane function becomes

$$h(\mathbf{x}_i) = w_0 + \phi(\mathbf{x}_i)^\top \mathbf{w}$$



Example:

$$\phi(\mathbf{x}_i) = (x_{i1}^2, x_{i2}^2, \sqrt{2} \cdot x_{i1}, \sqrt{2} \cdot x_{i2}, \sqrt{2} \cdot x_{i1} \cdot x_{i2}, 1)^\top$$

Non-linear SVMs – Optimization

Given any (non-linear) mapping $\phi: \mathbb{R}^d \to \mathbb{R}^D$, with $d \leq D$:

Non-linear regularized maximum margin

- **Minimize** $\|\mathbf{w}\|^2 + C \sum_{i=1}^{n} \xi_i$
- **subject to**: $y_i \left(\mathbf{w}^\top \phi(\mathbf{x}_i) + w_0\right) \geq 1 - \xi_i$ for all i
  $\xi_i \geq 0$ for all i

Non-linear SVMs – Optimization

- In the linear SVMs

$$h(\mathbf{x}) = w_0 + \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i^\top \mathbf{x}$$

- Applying the transformation function ϕ we obtain:

$$h(\mathbf{x}) = w_0 + \sum_{i=1}^{n} \alpha_i y_i \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}).$$

- For both, the solution $\hat{\mathbf{w}}$ is a linear combination of the training data!

$$\hat{\mathbf{w}} = \sum_{i=1}^{n} \alpha_i y_i \phi(\mathbf{x}_i)$$

Non-linear SVMs – Kernel trick

Definition (Kernel Function)

$$K(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^\top \phi(\mathbf{x}')$$

Definition (Kernel SVM)

$$h(\mathbf{x}) = w_0 + \sum_{i=1}^{n} \alpha_i K(\mathbf{x}_i, \mathbf{x})$$

- Replace the internal dot product by the associated kernel function.
- If the Kernel function can be computed efficiently, we can avoid to explicitly transform the data into the feature space.
- No explicit representation of ϕ is required.

Kernel Functions

- Linear:

$$K(\mathbf{x}, \mathbf{x}') = \mathbf{x}^\top \mathbf{x}'$$

- d-th degree Polynomial:

$$K(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^\top \mathbf{x}' + c)^d$$

- Radial Basis Function (RBF):

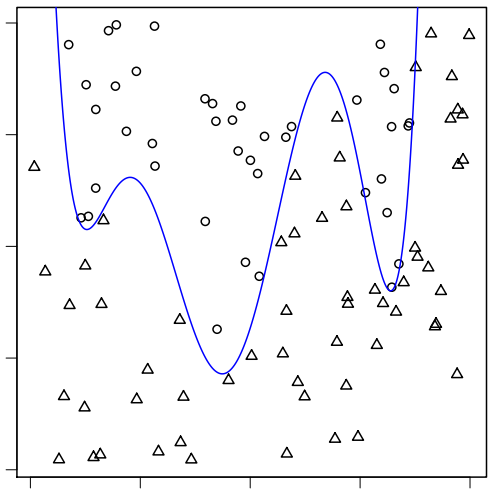
$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2\right)$$

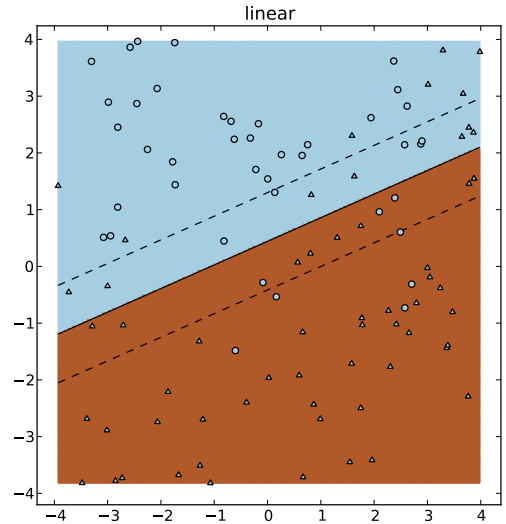
- Sigmoid:

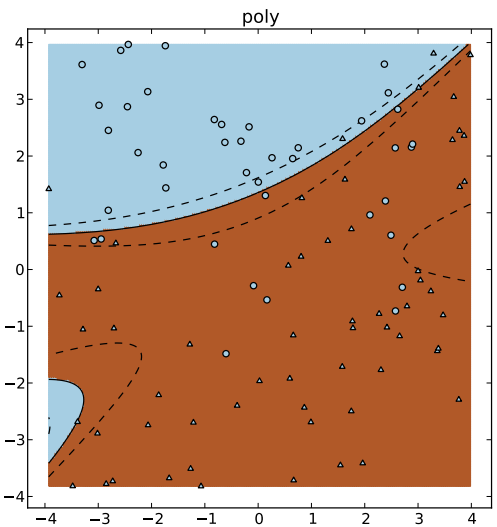
$$K(\mathbf{x}, \mathbf{x}') = \tanh(\gamma \cdot \mathbf{x}^\top \mathbf{x}' + c)$$

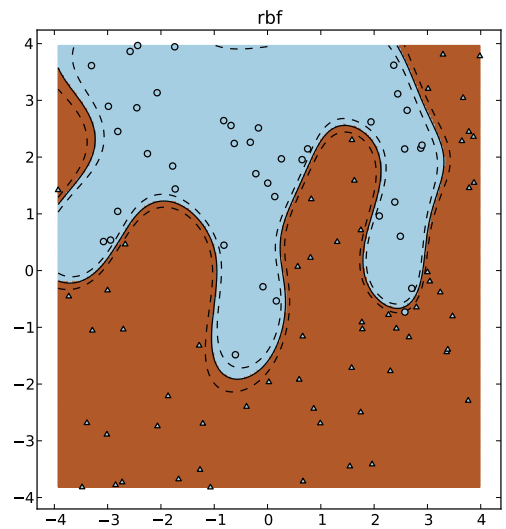
- Any valid kernel function: associated to a symmetric and positive semi-definite matrix. (Mercer's theorem)

Kernel FunctionsFonctions de noyau – Examples









Non-linear SVMs – Kernel Trick

For any learning algorithm that one can write in terms of only inner products between input attribute vectors, the dot product $\langle x, z \rangle$ may be replaced by a kernel function $K(x, z)$ where K is a kernel. This change "magically" allows a method to work efficiently in the high dimensional feature space corresponding to K.

Conclusion: Du classifieur linaire gnralis aux SVMs

- The **generalized linear classifiers** are not optimal (in the sense of the max margin).
- **Optimal separating hyperplanes** can be applied rarely.
- **Support Vector Machines**
  - SVMs find the **optimal** separating hyperplane which **maximizes** the margin between the classes.
  - In case of non separable data, using a **soft margin** allows misclassification errors. This provides also better generalization ability in the cases of noisy data or outliers.
  - Non-linear decision function can be modeled by using **Kernels**. They can project data in higher dimensional features space in which they become linearly separable.
  - However: no formulation for multi-class classification...

Support Vector Machines – Multiple classes

- SVMs as previously discussed are only applicable to binary classification problems.
- **Idea**: Construct multiple binary SVMs to distinguish $k > 2$ classes from each other.
- **One vs. all**: Train k classifiers where the i-th classifier is given the labels of the i-th class as positives and everything else as negative.
- **One vs. One**: Train $\sum_{i=1}^{k-1} i$ classifiers where each classifier is trained on samples from the i-th and j-th class, respectively.

Summary

- **Logistic regression** separates data linearly, yields true probabilities and the notion of log-odds makes it useful in numerous disciplines (e.g. medicine, social science). Can be extended to natively support multiple classes.
- **Support vector machines** can be used both for classification and regression and thanks to the Kernel trick in a wide range of applications. The best choice of Kernel and its parameters is not obvious and requires lots of testing.

References

- "Pattern Recognition and Machine Learning". Christopher M. Bishop. Springer.
- "Machine Learning: A Probabilistic Perspective" . Kevin P. Murphy, MIT Press, 2012.
- Standford CS229 lecture notes http://cs229.stanford.edu/notes/cs229-notes3.pdf