Hochschule Offenburg
offenburg.university

Machine Learning V

Non-Linear Models – Part II

Machine Learning IV

Hochschule Offenburg
offenburg.university

Outline

- **Part I : The Need for non-linear models**
- **Part I : Extending the simple linear classifier**
  - Adding non-linearity
  - Simple Neural Networks
- **Part II : Support Vector Machines**
  - Linear SVMs
  - The “Kernel-Trick”

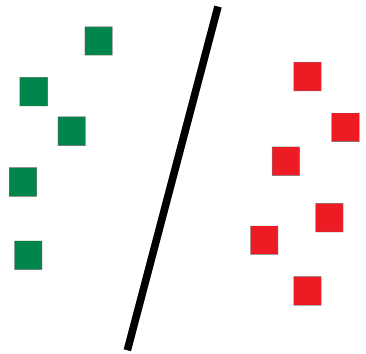
Support Vector Machines

- Invented in the mid 90s by Vapnik
- Classification and Regression
- State of the Art ML Algorithm of the pre Deep Learning era

Support Vector Machines

- Invented in the mid 90s by Vapnik
- Classification and Regression
- State of the Art ML Algorithm of the pre Deep Learning era

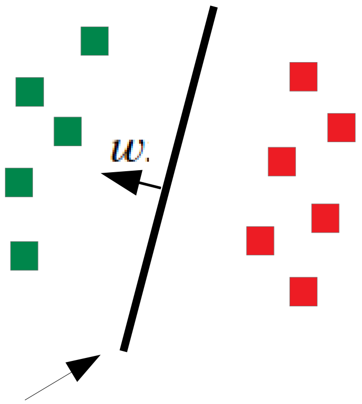
- **Basic model:**
  - Support only two classes {-1,1}
  - Linear classification



Support Vector Machines

- Invented in the mid 90s by Vapnik
- Classification and Regression
- State of the Art ML Algorithm of the pre Deep Learning era

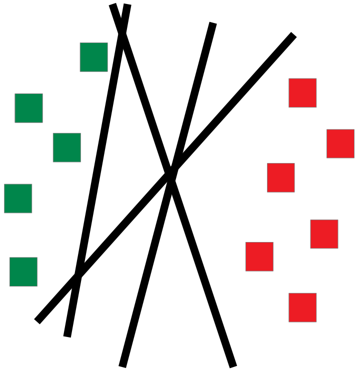
- **Basic model:**
  - Support only two classes {-1,1}
  - Linear classification



Parameterization: $wx - b = 0$

Support Vector Machines

What is the difference compared to previous formulations?

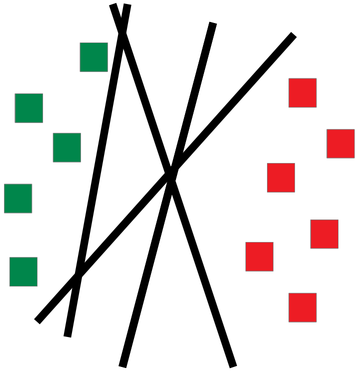


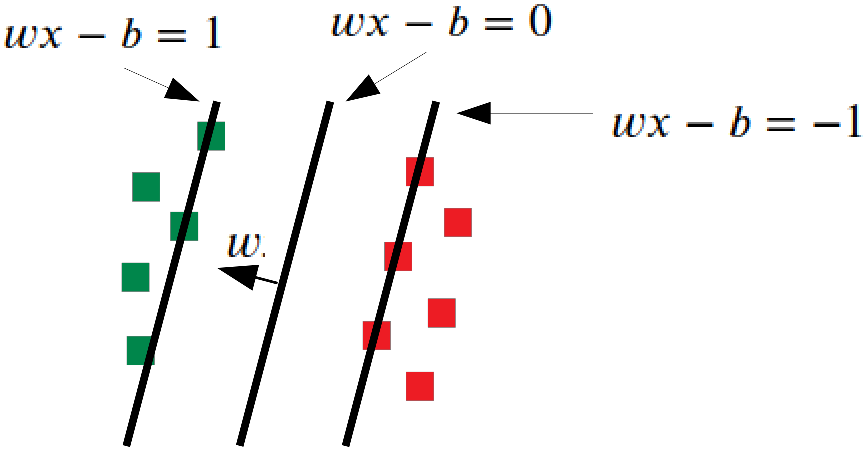
Standard linear model:
→ loss only on accuracy
→ many solutions

$$\arg\min_{w} \sum_{i=0}^{N} L(y_i, w^T x_i + b)$$

Support Vector Machines

What is the difference compared to previous formulations?





Standard linear model:
→ loss only on accuracy
→ many solutions

New optimization problem
→ “Max Margin”: $\frac{2}{\|w\|}$

→ only one solution, convex optimization problem

Support Vector Machines

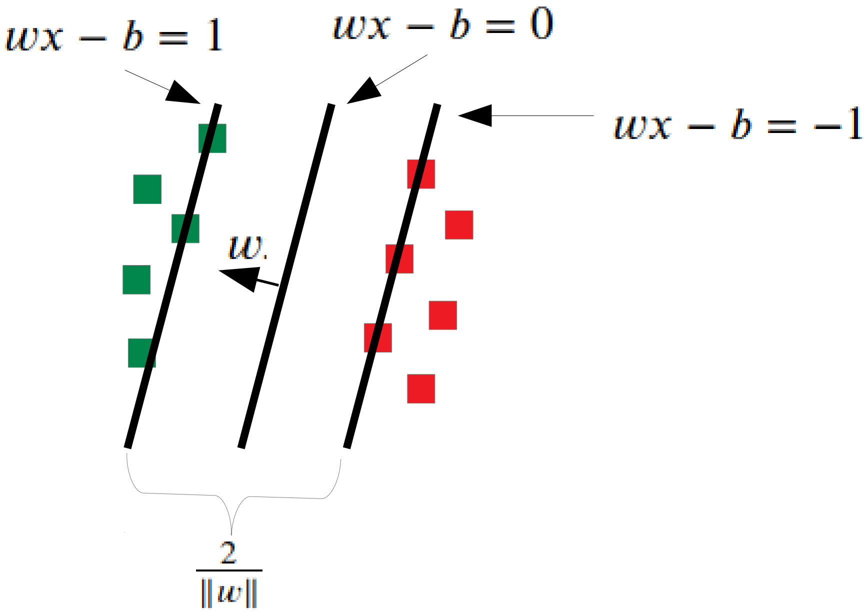
Basic formulation

New optimization problem
→ maximize “Margin”:
→ equals minimizing the uncertainty

$$\arg\min_{w} \frac{1}{2} \|w\|$$

subject to

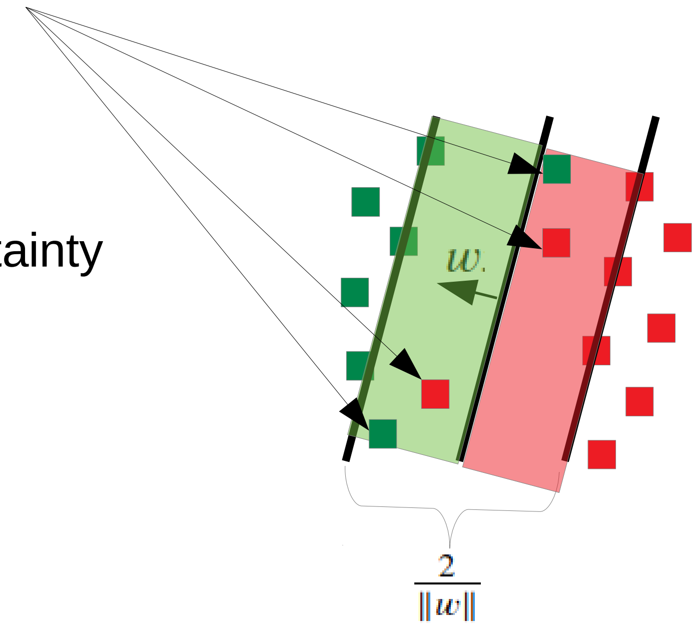
$$y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1$$



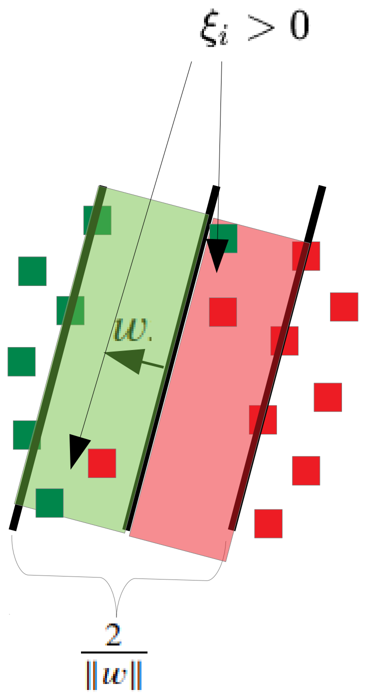
Support Vector Machines

Soft Margin: allow some outliers

Still
→ maximize “Margin”:
→ equals minimizing the uncertainty
→ enable linear separability



Soft Margin: allow some outliers



Still

→ maximize “Margin”:
→ equals minimizing the uncertainty
→ enable linear separability

$$\arg\min_{w} \frac{1}{2}\|w\| + C\sum_i \xi_i$$

subject to

$$y_i(\langle \mathbf{w}, \mathbf{x_i}\rangle + b) \geq 1 - \xi_i$$

Error

$$\zeta_i = \max\left(0, 1 - y_i(w \cdot x_i - b)\right)$$

Penalty factor

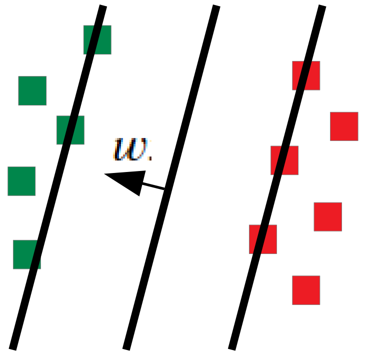
Support Vector Machines

Dual Formulation of the optimization Problem

Via ***Lagrange dual function,*** leads to quadratic optimization problem **(convex!)**

Re-write w as linear combination of samples:

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



There is exactly one solution and we will find it!

Support Vector Machines

Dual Formulation of the optimization Problem

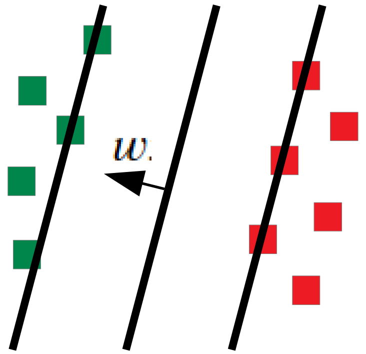
Via ***Lagrange dual function,*** leads to quadratic optimization problem (convex)

Re-write w as linear combination of samples:

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

$$\arg\max_{\alpha} \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle$$

Subject to $0 \leq \alpha_i \leq C$ and $\sum_{i=1}^{m} \alpha_i y_i = 0$



Normal as linear combination of samples

Support Vector Machines

Dual Formulation of the optimization Problem

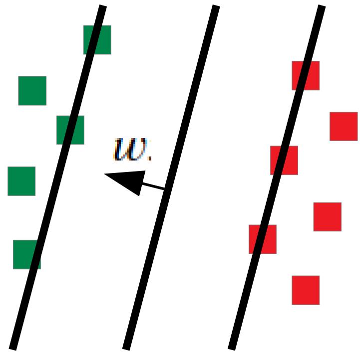
Via ***Lagrange dual function,*** leads to quadratic optimization problem (convex)

Re-write w as linear combination of samples:

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

$$\arg\max_{\alpha} \sum_{i=1}^{m} \alpha_i - \boxed{\frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle}$$

Subject to $0 \leq \alpha_i \leq C$ and $\sum_{i=1}^{m} \alpha_i y_i = 0$



Dot-product over all samples

Support Vector Machines

Dual Formulation of the optimization Problem

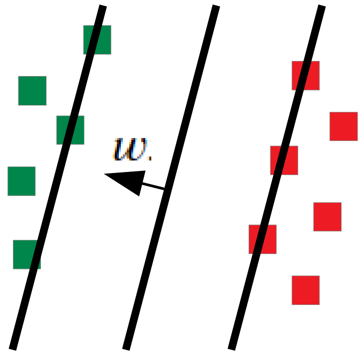
Via ***Lagrange dual function,*** leads to quadratic optimization problem (convex)

Re-write w as linear combination of samples:

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

$$\arg\max_{\alpha} \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle$$

Subject to $\boxed{0 \leq \alpha_i \leq C}$ and $\sum_{i=1}^{m} \alpha_i y_i = 0$



Soft Margin

Support Vector Machines

Dual Formulation of the optimization Problem

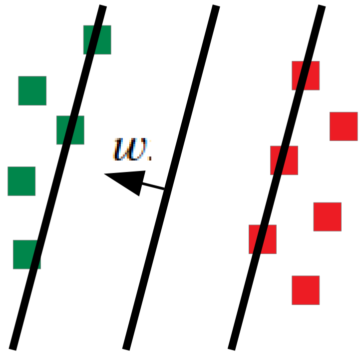
Via ***Lagrange dual function,*** leads to quadratic optimization problem (convex)

Re-write w as linear combination of samples:

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

$$\underset{\alpha}{\arg\max} \quad \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle$$

Subject to $0 \leq \alpha_i \leq C$ and $\boxed{\sum_{i=1}^{m} \alpha_i y_i = 0}$



Weight balance between classes

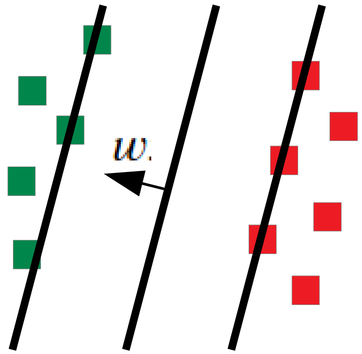
Support Vector Machines

Dual Formulation of the optimization Problem

Via ***Lagrange dual function,*** leads to quadratic optimization problem (convex)

Classification:

$$f(\mathbf{x}) = \operatorname{sgn}(\langle \mathbf{w}, \mathbf{x} \rangle + b) = \operatorname{sgn}\left(\sum_{i=1}^{m} \alpha_i y_i \langle \mathbf{x_i}, \mathbf{x} \rangle + b \right)$$



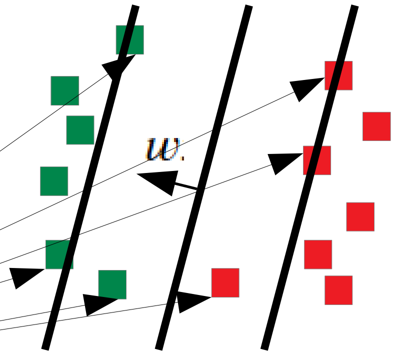
Support Vector Machines

Dual Formulation of the optimization Problem

Via ***Lagrange dual function,*** leads to quadratic optimization problem (convex)

Classification:

$$f(\mathbf{x}) = \operatorname{sgn}(\langle \mathbf{w}, \mathbf{x} \rangle + b) = \operatorname{sgn}\left(\sum_{i=1}^{m} \alpha_i y_i \langle \mathbf{x_i}, \mathbf{x} \rangle + b \right)$$



Only a few vectors have $\alpha_i > 0$

→ **“Support Vectors” form the actual model**

Hochschule Offenburg
offenburg.university

Non Linear SVMs:

→ follow same strategy as before and add simple non-linear function

At the formulation of the model normal: $\phi : \mathbb{R}^{d_1} \rightarrow \mathbb{R}^{d_2}, \mathbf{x} \mapsto \phi(\mathbf{x})$

$$f(\mathbf{x}) = \operatorname{sgn}(\langle \mathbf{w}, \mathbf{x} \rangle + b) = \operatorname{sgn}\left(\sum_{i=1}^{m} \alpha_i y_i \langle \mathbf{x_i}, \mathbf{x} \rangle + b \right)$$

⬇

$$f(\mathbf{x}) = \operatorname{sgn}(\langle \mathbf{w}, \phi(\mathbf{x}) \rangle + b) = \operatorname{sgn}\left(\sum_{i=1}^{m} \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}) + b \right)$$

Kernel function

$$k(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$$

Support Vector Machines

“Kernel Trick”: replace explicit non-linear function by *kernel*

Popular Kernels:

Polynomial (of degree d)

$$k(x, y) = \langle x, y \rangle$$

Gauss (or RBF)

$$k(x, y) = \exp\left(-\frac{||x-y||^2}{2\sigma^2}\right)$$

Linear

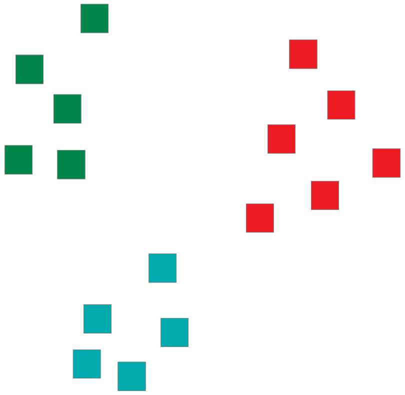
$$k(x, y) = \langle x, y \rangle$$

$$f(\mathbf{x}) = \mathrm{sgn}(\langle \mathbf{w}, \phi(\mathbf{x}) \rangle + b) = \mathrm{sgn}\left(\sum_{i=1}^{m} \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}) + b\right)$$

Training (optimization problem) and inference do not change!

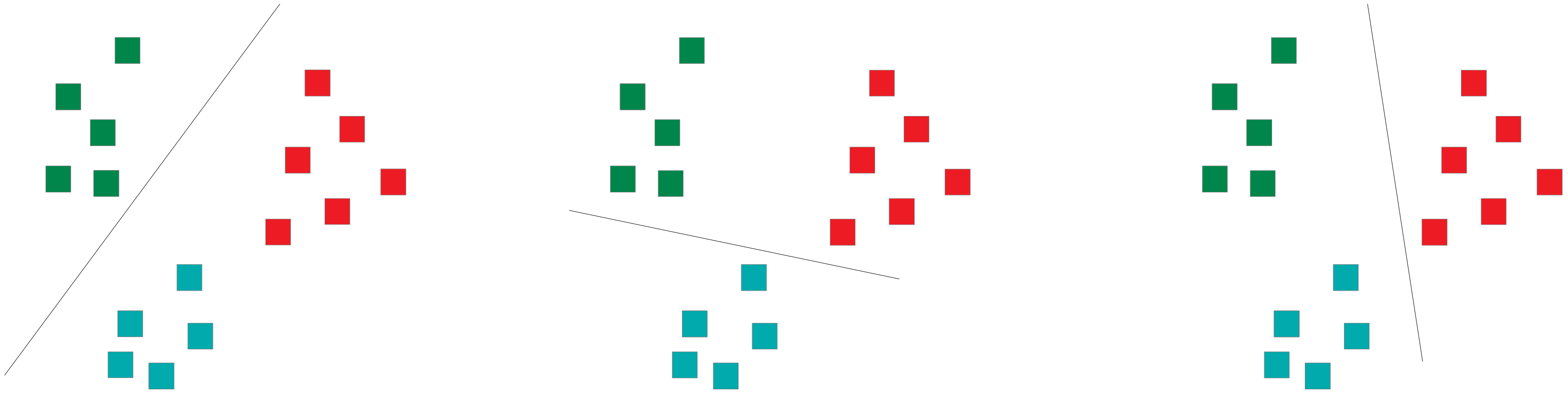
Support Vector Machines

Multi class problems:



Support Vector Machines

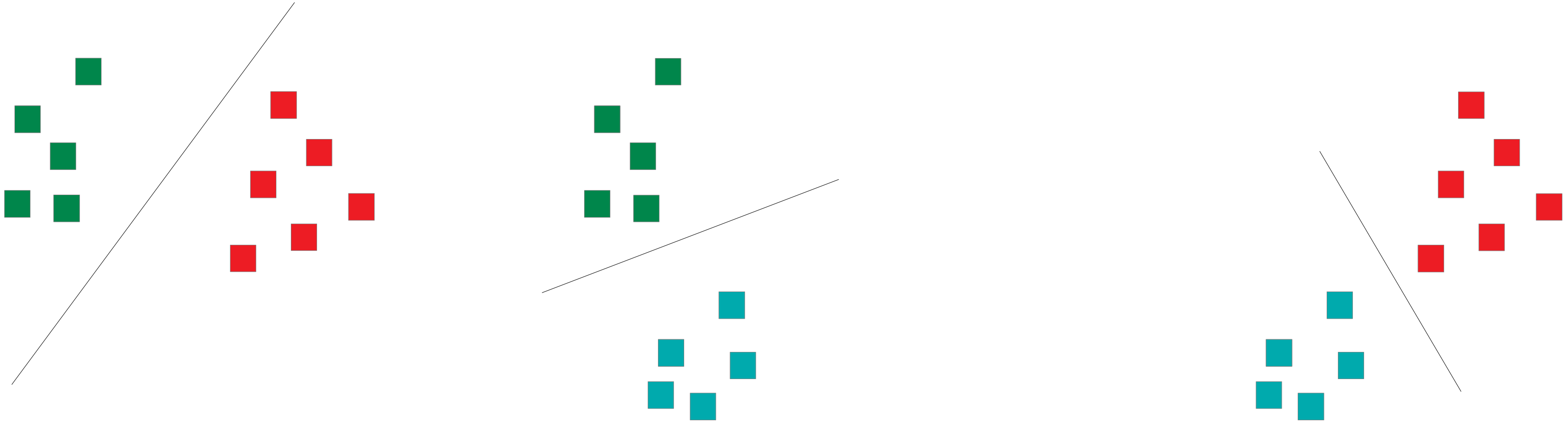
Multi class problems: 1-vs-Rest



N models, take best.

Support Vector Machines

Multi class problems: 1-vs-1



N(N-1)/2 models – tree execution, take best.

Discussion

Lab exercises coming up ...