Machine Intelligence 1

2.2 Support Vector Machines

Prof. Dr. Klaus Obermayer

Fachgebiet Neuronale Informationsverarbeitung (NI)

WS 2017/2018

2.2.1 Structural Risk Minimization

A bound on the generalization error

- finite samples: bound on the generalization error
  (c.f. SLT Result 3 on Slide 33 of Chapter 2.1.3)

$$P\left\{ \sup_{\underline{\mathbf{w}} \in \Lambda} \left| R_{(\underline{\mathbf{w}})} - R^{(p)}_{\text{emp}(\underline{\mathbf{w}})} \right| > \eta \right\} < \underbrace{4 \exp\left(G^{\Lambda}_{(2p)} - p\left(\eta - \tfrac{1}{p}\right)^2 \right)}_{\stackrel{!}{=} \epsilon}$$

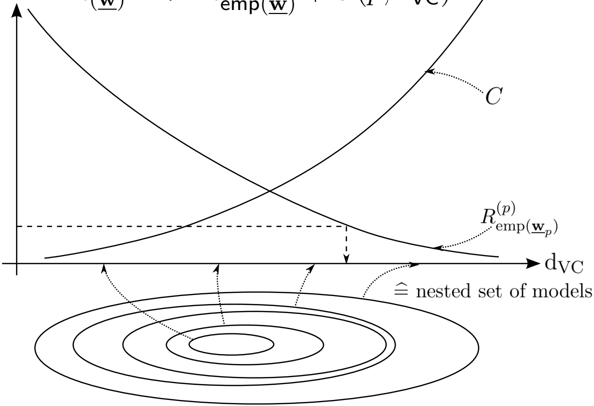
- with probability larger than $1 - \epsilon$ we obtain:

$$R_{(\underline{\mathbf{w}})} < \underbrace{R^{(p)}_{\text{emp}(\underline{\mathbf{w}})}}_{\text{empirical error}} + \underbrace{\left(\frac{G^{\Lambda}_{(2p)} - \ln \frac{\epsilon}{4}}{p} \right)^{\frac{1}{2}} + \frac{1}{p}}_{\text{complexity term } C}$$

- For a given ϵ, the complexity term C only depends on p and d_{VC}.

A bound on the generalization error

$$R_{(\underline{\mathbf{w}})} \quad < \quad R^{(p)}_{\mathsf{emp}(\underline{\mathbf{w}})} + C(p, d_{\mathsf{VC}})$$



underfitting ← ... appropriate model complexity ... → overfitting

Structural Risk Minimization (SRM)

$$R_{(\underline{\mathbf{w}})} \quad < \quad R^{(p)}_{\mathsf{emp}(\underline{\mathbf{w}})} + C(p, d_{\mathsf{VC}})$$

- Minimize complexity $C(p, d_{\mathsf{VC}})$ of the model class while keeping the empirical error $R^{(p)}_{\mathsf{emp}(\underline{\mathbf{w}})}$ bounded.
- SRM-learning is consistent *(cf. Vapnik 1998, chapter 6.3)*

2.2.2 Perceptrons Revisited

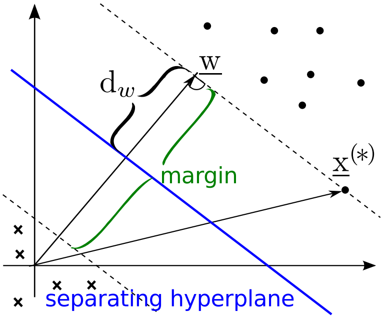
Canonical hyperplanes

- **data representation** binary classification: $\underline{\mathbf{x}} \in \mathbb{R}^N, \quad y_T \in \{-1, +1\}$
- **model class:** connectionist neurons $y = \operatorname{sign}\left(\underline{\mathbf{w}}^T \underline{\mathbf{x}} + b\right)$
- parameters of the seperating hyperplane $\underline{\mathbf{w}}^T \underline{\mathbf{x}} + b = 0$ are not unique
  - data dependent normalization

$$\min_{\alpha=1,\dots,p} \left| \underline{\mathbf{w}}^T \underline{\mathbf{x}}^{(\alpha)} + b \right| \stackrel{!}{=} 1$$

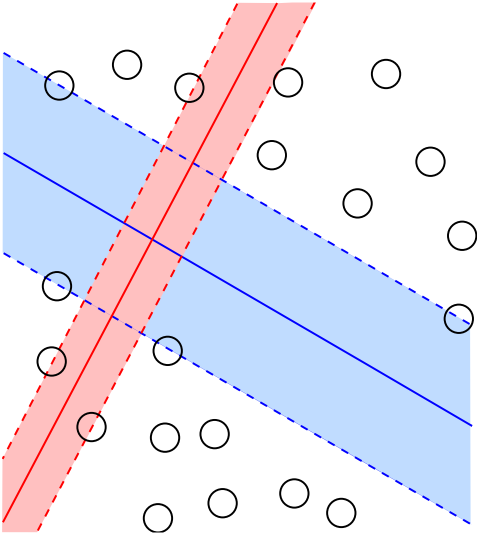
  - norm. distance to closest point $\underline{\mathbf{x}}^{(*)}$

$$\mathrm{d}_w = \frac{1}{\|\underline{\mathbf{w}}\|} \left| \underline{\mathbf{w}}^T \underline{\mathbf{x}}^{(*)} + b \right| = \frac{1}{\|\underline{\mathbf{w}}\|}$$

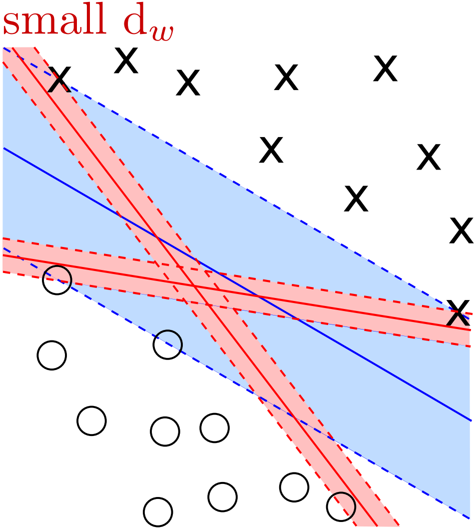


- The minimum normalized distance to the hyperplane is called **margin**.

Margins and capacity of the model class

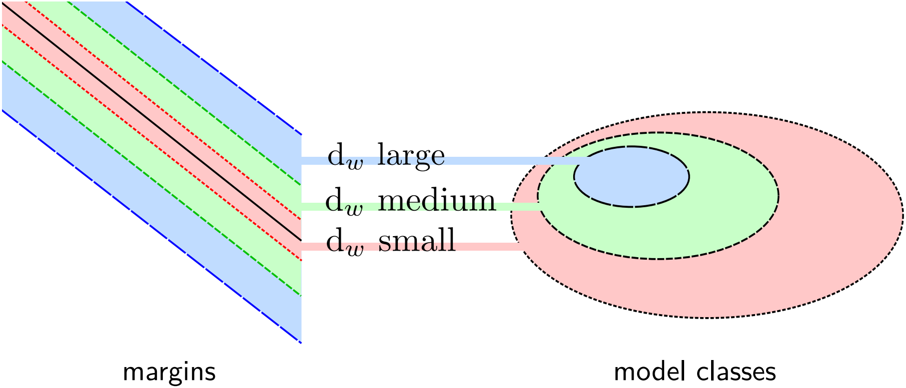


Margins and generalization of the model class



Nested set of models

nested set of models: $\frac{1}{\|\underline{\mathbf{w}}\|} \geq \mathrm{d}_w$



Margins and the VC dimension

Theorem (Vapnik, 1998)

$$\mathrm{d_{VC}} \quad \leq \quad \min\left(\left\lfloor \frac{\mathrm{d}_R^2}{\mathrm{d}_w^2} \right\rfloor, N \right) + 1$$

N : dimension of feature space

d_w : lower bound of the margin: $\frac{1}{\|\underline{\mathbf{w}}\|} \geq \mathrm{d}_w$

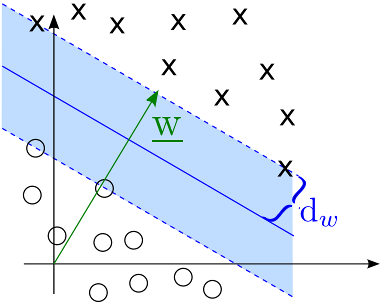
d_R : quantifies support of $\underline{\mathbf{x}}$: $P(\underline{\mathbf{x}}) \neq 0$ for $\|\underline{\mathbf{x}}\| \leq \mathrm{d}_R$

$\lfloor x \rfloor$: integer part of x

- $\frac{\mathrm{d}_R^2}{\mathrm{d}_w^2}$ is independent of the dimension N of feature space

2.2.3 Learning by Structural Risk Minimization

The primal optimization problem



$$y(\underline{\mathbf{x}}; \underline{\mathbf{w}}) \quad = \quad \operatorname{sign}(\underline{\mathbf{w}}^\top \underline{\mathbf{x}} + b)$$

$$\mathrm{d}_w \quad = \quad \frac{1}{\|\underline{\mathbf{w}}\|} \quad \stackrel{!}{=} \quad \max$$

$\frac{1}{2}\|\underline{\mathbf{w}}\|^2 \stackrel{!}{=} \min$ (minimize the capacity...)

s.t. $y_T^{(\alpha)}\left(\underline{\mathbf{w}}^\top \underline{\mathbf{x}}^{(\alpha)} + b\right) \geq 1\,, \quad \forall \alpha\,,$ (... for zero training error and normalized weight vectors)

The method of Lagrange multipliers

$$\underbrace{f_{0(\underline{\mathbf{x}})} \stackrel{!}{=} \min}_{\text{minimization}} \qquad \text{and} \qquad \underbrace{f_{k(\underline{\mathbf{x}})} \leq 0, \quad k = 1, \ldots, m}_{\text{constraints}}$$

$$L_{(\underline{\mathbf{x}}, \{\lambda_k\})} \stackrel{!}{=} f_{0(\underline{\mathbf{x}})} + \sum_{k=1}^{m} \lambda_k f_{k(\underline{\mathbf{x}})}\,, \qquad \lambda_k \geq 0\,, \quad \forall k \in \{1, \ldots, m\}$$

Theorem (Kuhn and Tucker)

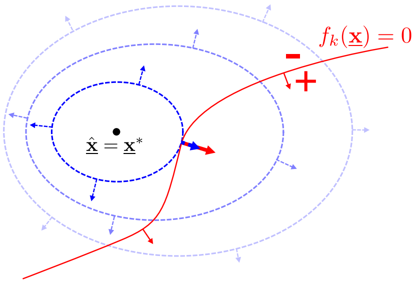
Let $A \subset \mathbb{R}^N$ be a convex subset and f_k be convex functions. If there *exists* at least one solution $\underline{\mathbf{x}} \in A$ that satisfies all constrains $f_k(\underline{\mathbf{x}}) \leq 0, \forall k$, then the solution $\underline{\mathbf{x}}^*$ of the constrained optimization problem is given by the saddle point of the Langrangian, i.e.

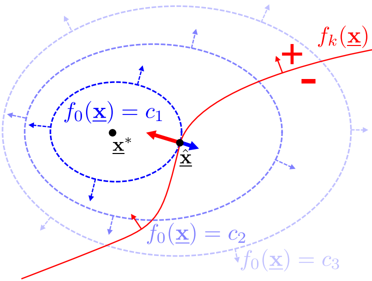
$$\min_{\underline{\mathbf{x}} \in A} L_{(\underline{\mathbf{x}}, \{\lambda_k^*\})} = L_{(\underline{\mathbf{x}}^*, \{\lambda_k^*\})} = \max_{\lambda_k \geq 0} L_{(\underline{\mathbf{x}}^*, \{\lambda_k\})}$$

The values of the Lagrange multipliers

$$f_{0(\underline{\mathbf{x}})} \overset{!}{=} \min \qquad \text{s.t.} \qquad f_{k(\underline{\mathbf{x}})} \leq 0\,, \quad \forall k$$

$$L_{(\underline{\mathbf{x}},\{\lambda_k\})} := f_{0(\underline{\mathbf{x}})} + \sum_{k=1}^{m} \lambda_k f_{k(\underline{\mathbf{x}})} \qquad \text{s.t.} \qquad \lambda_k \geq 0\,, \quad \forall k$$





- $f_{k(\hat{\underline{\mathbf{x}}})} < 0 \quad \Rightarrow \quad \lambda_k = 0$ (solution **behind** boundary, see left figure)
- $f_{k(\hat{\underline{\mathbf{x}}})} = 0 \quad \Rightarrow \quad \lambda_k > 0$ (solution **on** boundary, see right figure)
  - at minimum $\hat{\underline{\mathbf{x}}}$ of boundary $f_{k(\underline{\mathbf{x}})} = 0$: $\quad \frac{\partial f_0}{\partial \underline{\mathbf{x}}}\Big|_{\hat{\underline{\mathbf{x}}}} \propto -\frac{\partial f_k}{\partial \underline{\mathbf{x}}}\Big|_{\hat{\underline{\mathbf{x}}}}$

Application to the primal problem of SRM

- binary classification with linear connectionist neuron

$$\begin{aligned} f_{0(\underline{\mathbf{w}},b)} &= \frac{1}{2}\|\underline{\mathbf{w}}\|^2 \\ f_{\alpha(\underline{\mathbf{w}},b)} &= -\Big\{y_T^{(\alpha)}\Big(\underline{\mathbf{w}}^T\underline{\mathbf{x}}^{(\alpha)}+b\Big)-1\Big\} \leq 0\,, \quad \forall\alpha\in\{1,\ldots,p\} \end{aligned}$$

Lagrangian

$$L_{(\underline{\mathbf{w}},b,\{\lambda_\alpha\})} = \frac{1}{2}\|\underline{\mathbf{w}}\|^2 - \sum_{\alpha=1}^{p}\lambda_\alpha\Big\{y_T^{(\alpha)}\Big(\underline{\mathbf{w}}^T\underline{\mathbf{x}}^{(\alpha)}+b\Big)-1\Big\}$$

$$\min_{\underline{\mathbf{w}},b} L_{(\underline{\mathbf{w}},b,\{\lambda_\alpha^*\})} = L_{(\underline{\mathbf{w}}^*,b^*,\{\lambda_\alpha^*\})} = \max_{\lambda_\alpha\geq 0} L_{(\underline{\mathbf{w}}^*,b^*,\{\lambda_\alpha\})}$$

$\underline{\mathbf{w}}, b$: “primal” variables

λ_α : “dual” variables

(solution see blackboard)

The dual problem

$$\underline{\mathbf{w}}^* = \sum_{\alpha=1}^{p} \lambda_\alpha \, y_T^{(\alpha)} \, \underline{\mathbf{x}}^{(\alpha)}$$

$$L = -\frac{1}{2} \sum_{\alpha,\beta=1}^{p} \lambda_\alpha \lambda_\beta y_T^{(\alpha)} y_T^{(\beta)} \underbrace{\left(\underline{\mathbf{x}}^{(\alpha)}\right)^\top \underline{\mathbf{x}}^{(\beta)}}_{\circledast} + \sum_{\alpha=1}^{p} \lambda_\alpha \stackrel{!}{=} \max_{\{\lambda_\alpha\}}$$

$\lambda_\alpha \geq 0 \,, \quad \forall \alpha \in \{1, \ldots, p\} \,,$ and $\sum_{\alpha=1}^{p} \lambda_\alpha y_T^{(\alpha)} = 0$ (constraints)

- solved numerically using "sequential minimal optimization" (SMO)

The optimal classifier

- linear classifier

$$y(\underline{\mathbf{x}}) \quad = \quad \operatorname{sign}\left(\underline{\mathbf{w}}^\top \underline{\mathbf{x}} + b\right)$$

- When $\{\lambda_\alpha^*\}_{\alpha=1}^p$ are known, we can compute

$$\underline{\mathbf{w}}^* \quad = \quad \sum_{\alpha=1}^{p} \lambda_\alpha^* \, y_T^{(\alpha)} \, \underline{\mathbf{x}}^{(\alpha)} \,,$$

- and the classifier is

$$y(\underline{\mathbf{x}}) \quad = \quad \operatorname{sign}\left(\sum_{\alpha=1}^{p} \lambda_\alpha^* \, y_T^{(\alpha)} \underbrace{\left(\underline{\mathbf{x}}^{(\alpha)}\right)^\top \underline{\mathbf{x}}}_{\circledast} + b^*\right).$$

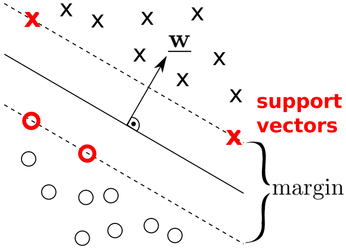
Support vectors

- Only constraints f_α correspond to finite Lagrange multipliers $\lambda_\alpha \neq 0$:

$$f_\alpha(\underline{\mathbf{w}}^*, b^*) \quad = \quad -\Big\{ y_T^{(\alpha)} \Big(\underline{\mathbf{w}}^{*T} \underline{\mathbf{x}}^{(\alpha)} + b^* \Big) - 1 \Big\} \quad \stackrel{!}{=} \quad 0$$

- This implies that the corresponding data points are located on the margin of the hyperplane

- These data points are called **support vectors**



Calculation of the bias

- for all support vectors $\underline{\mathbf{x}}^{\alpha}$:

$$f_{\alpha}(\underline{\mathbf{w}}^*, b^*) \;=\; -\Big\{ y_T^{(\alpha)} \Big(\underline{\mathbf{w}^*}^T \underline{\mathbf{x}}^{(\alpha)} + b^* \Big) - 1 \Big\} \;\stackrel{!}{=}\; 0$$

$$\Rightarrow \quad b^* \;=\; y_T^{(\alpha)} \;-\; \underline{\mathbf{w}}^{*\top} \underline{\mathbf{x}}^{(\alpha)}$$

- bias b^* is computed by averaging over the support vectors

$$b^* = \frac{1}{\#\mathrm{SV}} \sum_{\alpha \in \mathrm{SV}} \left(y_T^{(\alpha)} - \sum_{\beta \in \mathrm{SV}} \lambda_\beta y_T^{(\beta)} \underbrace{\left(\underline{\mathbf{x}}^{(\beta)}\right)^T \underline{\mathbf{x}}^{(\alpha)}}_{\circledast} \right)$$

Support Vector Machines (SVM)

- **perceptrons** $\hat{y}(\underline{\mathbf{x}}) = \operatorname{sign}\left(\underline{\mathbf{w}}^T \underline{\mathbf{x}} + b\right)$ trained by SRM are called SVM

- weights and threshold are calculated by solving the **dual optimization problem** for the Lagrange multipliers $\{\lambda_\alpha \geq 0\}_{\alpha=1}^p$, $\sum_{\alpha=1}^p \lambda_\alpha \, y_T^{(\alpha)} = 0$:

$$\max L(\{\lambda_\alpha\}) \;=\; -\frac{1}{2} \sum_{\alpha,\beta=1}^{p} \lambda_\alpha \lambda_\beta y_T^{(\alpha)} y_T^{(\beta)} \underbrace{\left(\underline{\mathbf{x}}^{(\alpha)}\right)^T \underline{\mathbf{x}}^{(\beta)}}_{\circledast} + \sum_{\alpha=1}^{p} \lambda_\alpha$$

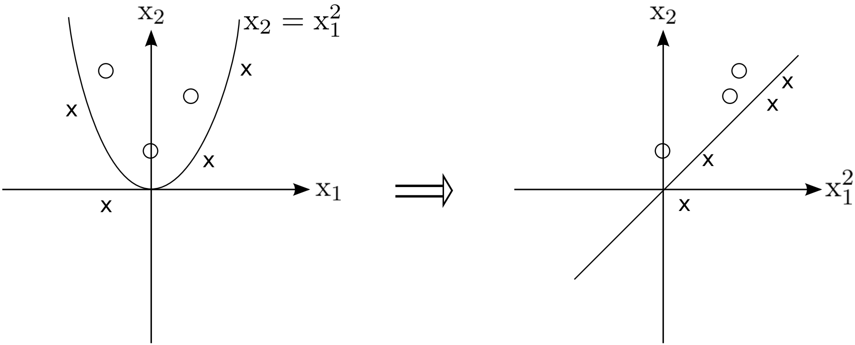
- the SVM classifier is given by :

$$\hat{y}(\underline{\mathbf{x}}) \;=\; \operatorname{sign}\left(\sum_{\alpha \in \mathrm{SV}} \lambda_\alpha \, y_T^{(\alpha)} \underbrace{\left(\underline{\mathbf{x}}^{(\alpha)}\right)^T \underline{\mathbf{x}}}_{\circledast} \;+\; b \right)$$

with $$b \;=\; \frac{1}{\#\mathrm{SV}} \sum_{\alpha \in \mathrm{SV}} \left(y_T^{(\alpha)} - \sum_{\beta \in \mathrm{SV}} \lambda_\beta y_T^{(\beta)} \underbrace{\left(\underline{\mathbf{x}}^{(\beta)}\right)^T \underline{\mathbf{x}}^{(\alpha)}}_{\circledast} \right)$$

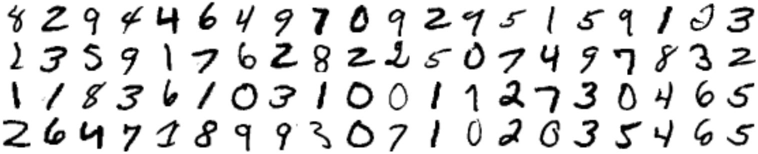
2.2.4 SRM Learning for Non-linear Classification Boundaries

Transformation of feature space



$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \underbrace{\underline{\mathbf{x}}}_{\text{elementary features}} \longrightarrow \underbrace{\underline{\phi}_{(\underline{\mathbf{x}})}}_{\text{new, nonlinear combinations}} = \begin{bmatrix} x_1 \\ x_1^2 \end{bmatrix}$$

Transformation of feature space



- feature space: monomials of degree n

$$\underbrace{\underline{\mathbf{x}}}_{\substack{\text{elementary}\\ \text{features}}} \longrightarrow \underbrace{\underline{\phi}_{(\underline{\mathbf{x}})}}_{\substack{\text{monomials of}\\ \text{degree } n}}$$

- $n = 10$ and N pixel values $x_i \Rightarrow N^{10}$ monomials

The kernel trick

$$\underbrace{\underline{\mathbf{x}}}_{\text{elementary features}} \longrightarrow \underbrace{\underline{\phi}_{(\underline{\mathbf{x}})}}_{\text{new, nonlinear features}}$$

- SVM requires only scalar products $\underline{\phi}_{(\underline{\mathbf{x}})}^{\top}\underline{\phi}_{(\underline{\mathbf{x}})}$ in feature space
- replace scalar products with **kernel function** $\underline{\phi}_{(\underline{\mathbf{x}})}^{\top}\underline{\phi}_{(\underline{\mathbf{x}}')} \rightarrow K_{(\underline{\mathbf{x}},\underline{\mathbf{x}}')}$
- e.g. the polynomial kernel $K_{(\underline{\mathbf{x}},\underline{\mathbf{x}}')} = (\underline{\mathbf{x}}^{\top}\underline{\mathbf{x}}' + 1)^n$ is a scalar product in the space of all monomials of degree n

Mercer's theorem

- let $\mathcal{X}$ be a *compact* subset of $\mathbb{R}^N$
- let $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}, K \in L_\infty$, be a symmetric function ("kernel")
- let $T_K : L_{2(\mathcal{X})} \rightarrow L_{2(\mathcal{X})}$ be the linear convolution operator

$$T_K[f]_{(\underline{\mathbf{x}})} \quad := \quad \int_{\mathcal{X}} K_{(\underline{\mathbf{x}},\underline{\mathbf{x}}')} f_{(\underline{\mathbf{x}}')} \, d\underline{\mathbf{x}}'$$

- let $\lambda_i \in \mathbb{R}$ be *eigenvalues* and $\psi_{i(\underline{\mathbf{x}})} \in L_{2(\mathcal{X})}$ *eigenfunctions* of T_K

Mercer's theorem

Every **positive semi-definite** kernel K can be written as infinite sum

$$K(\underline{\mathbf{x}}, \underline{\mathbf{x}}') \quad = \quad \sum_{i=1}^{\infty} \lambda_i \, \psi_{i(\underline{\mathbf{x}})} \, \psi_{i(\underline{\mathbf{x}}')} \, ,$$

where the convergence is absolute and uniform.

Kernel properties

symmetric kernels

orthonormal eigenfunctions:

$$\int_{\mathcal{X}} \psi_{i(\underline{\mathbf{x}})} \, \psi_{j(\underline{\mathbf{x}})} \, d\underline{\mathbf{x}} \quad = \quad \delta_{ij}, \qquad \forall i, j \in \mathbb{N}$$

positive semi-definite kernels

all eigenvalues λ_i are **non-negative**:

$$\iint_{\mathcal{X}\times\mathcal{X}} K_{(\underline{\mathbf{x}},\underline{\mathbf{x}}')} f_{(\underline{\mathbf{x}})} f_{(\underline{\mathbf{x}}')} \, d\underline{\mathbf{x}} \, d\underline{\mathbf{x}}' \geq 0 \,, \quad \forall f \in L_{2(\mathcal{X})} \,, \quad \text{(positive semi-definite)}$$

Induced feature space

$$K(\underline{\mathbf{x}}, \underline{\mathbf{x}}') \quad = \quad \sum_{i=1}^{\infty} \underbrace{\psi_{i(\underline{\mathbf{x}})} \sqrt{\lambda_i}}_{\phi_{i(\underline{\mathbf{x}})}} \underbrace{\sqrt{\lambda_i}\, \psi_{i(\underline{\mathbf{x}}')}}_{\phi_{i(\underline{\mathbf{x}}')}} \quad = \quad \underline{\phi}_{(\underline{\mathbf{x}})}^{\top} \underline{\phi}_{(\underline{\mathbf{x}}')}$$

- every positive semi-definite kernel K is an **inner product** in the induced space Φ, spanned by the features $\phi_{i(\underline{\mathbf{x}})} = \sqrt{\lambda_i}\, \psi_{i(\underline{\mathbf{x}})}$
- Φ is often high dimensional
- a linear classifier in Φ can solve non-linearly separable problems in $\mathcal{X}$

Typical kernel functions

$K_{(\underline{\mathbf{x}}, \underline{\mathbf{x}}')} = \left(\underline{\mathbf{x}}^T \underline{\mathbf{x}}' + 1\right)^d$ polynomial kernel of degree d

$\rightarrow$ image processing: pixel correlations

Typical kernel functions

$K_{(\underline{\mathbf{x}}, \underline{\mathbf{x}}')} = \left(\underline{\mathbf{x}}^T \underline{\mathbf{x}}' + 1\right)^d$ polynomial kernel of degree d
$\rightarrow$ image processing: pixel correlations

$K_{(\underline{\mathbf{x}}, \underline{\mathbf{x}}')} = \exp\left\{ - \frac{(\underline{\mathbf{x}} - \underline{\mathbf{x}}')^2}{2\sigma^2} \right\}$ RBF-kernel with range σ
$\rightarrow$ infinite dimensional feature space

Typical kernel functions

$K_{(\underline{\mathbf{x}},\underline{\mathbf{x}}')} = \left(\underline{\mathbf{x}}^T \underline{\mathbf{x}}' + 1\right)^d$ polynomial kernel of degree d
$\rightarrow$ image processing: pixel correlations

$K_{(\underline{\mathbf{x}},\underline{\mathbf{x}}')} = \exp\left\{ - \frac{(\underline{\mathbf{x}}-\underline{\mathbf{x}}')^2}{2\sigma^2} \right\}$ RBF-kernel with range σ
$\rightarrow$ infinite dimensional feature space

$K_{(\underline{\mathbf{x}},\underline{\mathbf{x}}')} = \tanh\left\{\kappa \underline{\mathbf{x}}^T \underline{\mathbf{x}}' + \theta\right\}$ neural network kernel with parameters κ and $\theta \rightarrow$ not positive definite

Typical kernel functions

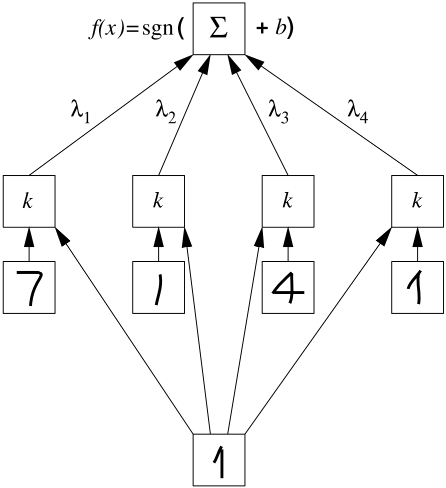
$K_{(\underline{\mathbf{x}},\underline{\mathbf{x}}')} = \left(\underline{\mathbf{x}}^T\underline{\mathbf{x}}' + 1\right)^d$ polynomial kernel of degree d
$\rightarrow$ image processing: pixel correlations

$K_{(\underline{\mathbf{x}},\underline{\mathbf{x}}')} = \exp\left\{-\frac{(\underline{\mathbf{x}}-\underline{\mathbf{x}}')^2}{2\sigma^2}\right\}$ RBF-kernel with range σ
$\rightarrow$ infinite dimensional feature space

$K_{(\underline{\mathbf{x}},\underline{\mathbf{x}}')} = \tanh\left\{\kappa\underline{\mathbf{x}}^T\underline{\mathbf{x}}' + \theta\right\}$ neural network kernel with parameters κ and $\theta \rightarrow$ not positive definite

$K_{(\underline{\mathbf{x}},\underline{\mathbf{x}}')} = \frac{1}{(\|\underline{\mathbf{x}}-\underline{\mathbf{x}}'\|^2+\epsilon^2)^{N/2}}$ Plummer kernel with parameter ϵ
$\rightarrow$ scale invariant kernel

SVM with kernels



classification $\boxed{f(x) = \mathrm{sgn}\left(\sum_i \lambda_i k(x, x_i) + b\right)}$

weights

comparison: e.g. $k(x, x_i) = (x \cdot x_i)^d$

$k(x, x_i) = \exp(-\|x - x_i\|^2 / c)$

support vectors $x_1 \ldots x_4$

$k(x, x_i) = \tanh(\kappa(x \cdot x_i) + \theta)$

input vector x

see Schölkopf & Smola (2001, p. 202)

Comments

- Mercer's theorem can be used to "kernelize" many different linear methods, both supervised or unsupervised.
  - Fisher discriminant analysis
  - principal component analysis *(see MI 2)*
  - k-means clustering & self-organizing maps
  - canonical correlation analysis

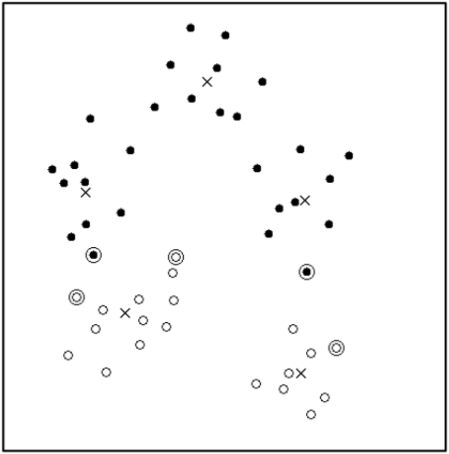
Comments

- SVM vs. RBF networks
- RBF network for classification
  - 5 Gaussian bases ($\times$)

$$y(\underline{\mathbf{x}}) = \text{sign}\left(\sum_{i=1}^{5} w_i \exp\left(\frac{1}{2\sigma_i^2}\|\underline{\mathbf{x}} - \underline{\mathbf{t}}_i\|^2\right)\right)$$

- SVM with Gaussian kernel
  - 5 support vectors $\underline{\mathbf{x}}_i$ ($\circ$)
  - $a_i = \lambda_\alpha y_T^{(\alpha)}$, for $\underline{\mathbf{x}}_i = \underline{\mathbf{x}}^{(\alpha)}$

$$y(\underline{\mathbf{x}}) = \text{sign}\left(\sum_{i=1}^{5} a_i \exp\left(\frac{1}{2\sigma^2}\|\underline{\mathbf{x}} - \underline{\mathbf{x}}_i\|^2\right) + b\right)$$

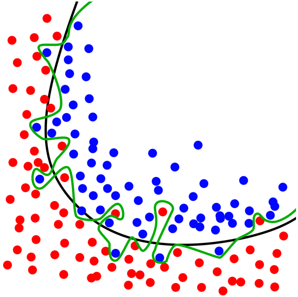


see Schölkopf et al. (1997), Schölkopf & Smola (2001, p. 204)

2.2.5 The C-Support Vector Machine

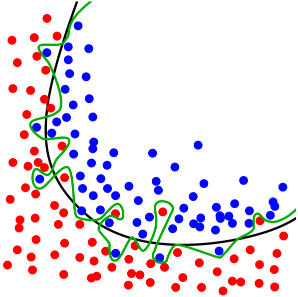
Classification of non-separable problems

- real-world problems are typically non-separable
- incomplete feature sets & noise
- perfect separation of the training set $\rightsquigarrow$ overfitting



Classification of non-separable problems

- real-world problems are typically non-separable
- incomplete feature sets & noise
- perfect separation of the training set $\rightsquigarrow$ overfitting



consequences

$$R_{(\underline{\mathbf{w}})} \quad \leq \quad R^{(p)}_{\mathrm{emp}(\underline{\mathbf{w}})} \;+\; C(p, d_{\mathsf{VC}})$$

- finite training error $R^{(p)}_{\mathrm{emp}} \neq 0$
- trade-off between minimization of the training error and the capacity of the model class

The primal problem

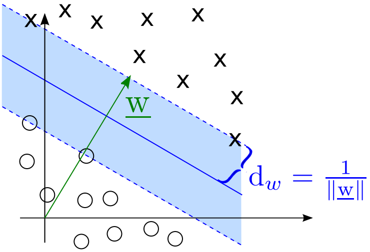
$$\frac{1}{2}\|\underline{\mathbf{w}}\|^2 \stackrel{!}{=} \min \quad \Big\{$$

minimize upper bound on VC dimension

constraints $(\forall\alpha)$:

$$y_T^{(\alpha)}\left(\underline{\mathbf{w}}^T\underline{\mathbf{x}}^{(\alpha)} + b\right) \geq 1$$

normalization & correct classification of all data points



The primal problem

$$\frac{1}{2}\|\underline{\mathbf{w}}\|^2 + \frac{C}{p}\sum_{\alpha=1}^{p}\varphi_\alpha \overset{!}{=} \min \quad \left\{ \begin{array}{l} \text{minimize upper bound on VC dimension} \\ + \text{ minimize (approx.) margin error} \end{array} \right.$$

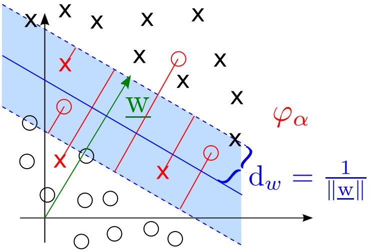
constraints ($\forall\alpha$): (C: regularization parameter)

$$y_T^{(\alpha)}\left(\underline{\mathbf{w}}^T\underline{\mathbf{x}}^{(\alpha)} + b\right) \geq 1 - \varphi_\alpha$$

normalization & correct classification of all data points for $\varphi_\alpha = 0$

$$\varphi_\alpha \geq 0$$

"margin errors" for $\varphi_\alpha \neq 0$



Dual problem of the C-SVM

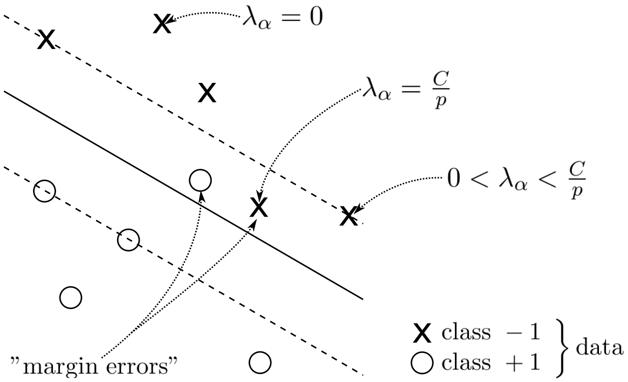
- Objective

$$-\frac{1}{2}\sum_{\alpha,\beta=1}^{p}\lambda_\alpha\lambda_\beta y_T^{(\alpha)}y_T^{(\beta)}\underbrace{\left(\underline{\mathbf{x}}^{(\alpha)}\right)^T\underline{\mathbf{x}}^{(\beta)}}_{\text{kernel function}}+\sum_{\alpha=1}^{p}\lambda_\alpha \quad \stackrel{!}{=} \quad \max_{\{\lambda_\alpha\}_{\alpha=1}^{p}}$$

- Constraints:

$$\sum_{\alpha=1}^{p}\lambda_\alpha y_T^{(\alpha)}=0 \qquad\qquad 0\le\underbrace{\lambda_\alpha \;\le\frac{C}{p}}_{\text{difference to separable case}}$$

Margin and support vectors



The C-SVM classifier

$$\underline{\mathbf{w}} \;=\; \sum_{\alpha=1}^{p} \lambda_\alpha y_T^{(\alpha)} \underline{\mathbf{x}}^{(\alpha)} \qquad \leadsto \lambda_\alpha \neq 0 \text{ only for support vectors } SV$$

The C-SVM classifier

$$\begin{aligned} \underline{\mathbf{w}} &= \sum_{\alpha=1}^{p} \lambda_\alpha y_T^{(\alpha)} \underline{\mathbf{x}}^{(\alpha)} \qquad \rightsquigarrow \lambda_\alpha \neq 0 \text{ only for support vectors } SV \\ b &= \frac{1}{\# SV_<} \sum_{\alpha \in SV_<} \left(y_T^{(\alpha)} - \sum_{\beta \in SV} \lambda_\beta y_T^{(\beta)} \underbrace{\left(\underline{\mathbf{x}}^{(\beta)}\right)^T \underline{\mathbf{x}}^{(\alpha)}}_{\text{kernel!}} \right) \end{aligned}$$

$SV_<$: SVs with $\lambda_\alpha < \frac{C}{p}$ (SVs on the margin)

The C-SVM classifier

$$\underline{\mathbf{w}} = \sum_{\alpha=1}^{p} \lambda_\alpha y_T^{(\alpha)} \underline{\mathbf{x}}^{(\alpha)} \qquad \rightsquigarrow \lambda_\alpha \neq 0 \text{ only for support vectors } SV$$

$$b = \frac{1}{\# SV_<} \sum_{\alpha \in SV_<} \left(y_T^{(\alpha)} - \sum_{\beta \in SV} \lambda_\beta y_T^{(\beta)} \underbrace{\left(\underline{\mathbf{x}}^{(\beta)}\right)^T \underline{\mathbf{x}}^{(\alpha)}}_{\text{kernel!}} \right)$$

$SV_<$: SVs with $\lambda_\alpha < \frac{C}{p}$ (SVs on the margin)

Classifier

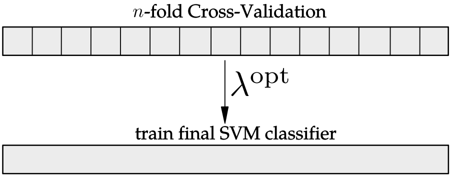
$$\hat{y}(\underline{\mathbf{x}}) = \text{sign}\left(\underline{\mathbf{w}}^T \underline{\mathbf{x}} + b\right) = \text{sign}\left(\sum_{\alpha \in SV} \lambda_\alpha y_\top^{(\alpha)} \underbrace{\left(\underline{\mathbf{x}}^{(\alpha)}\right)^\top \underline{\mathbf{x}}}_{\text{kernel function}} + b \right)$$

Validation & selection of hyperparameters

- validation and model selection w.r.t. **0-1 loss**

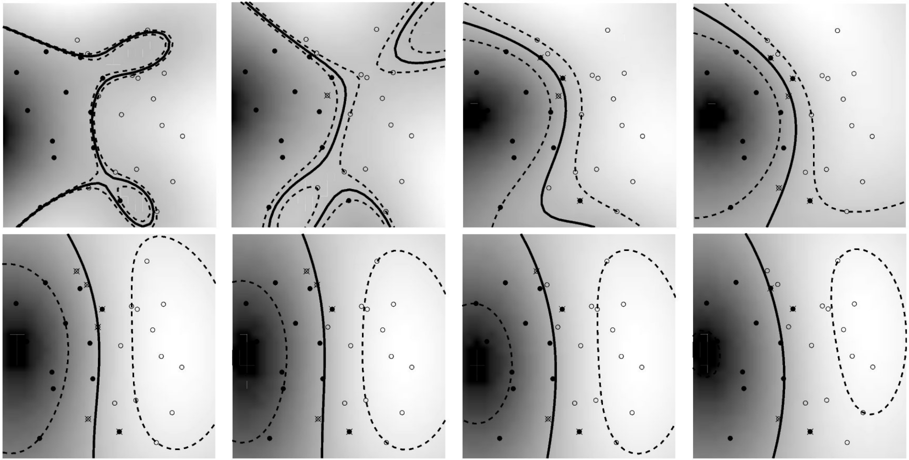
$$
e(\underline{\mathbf{x}}^{(\alpha)}, y_T^{(\alpha)}) \quad = \quad \begin{cases} 0 & \text{, if } \hat{y}(\underline{\mathbf{x}}^{(\alpha)}) = y_T^{(\alpha)} \\ 1 & \text{, otherwise} \end{cases}
$$

- hyper-parameter selection $(C, \sigma, \ldots)$ by n-fold **cross-validation**



- validation on hold-out **validation set**

SVM and overfitting



(related ν-SVM with $\nu \in \{0.1, 0.2, \ldots, 0.8\}$ and RBF kernel)

see Schölkopf & Smola (2001, p. 207)

2.2.6 Sequential Minimal Optimization

The dual problem

$$-\frac{1}{2}\sum_{\alpha,\beta=1}^{p} \lambda_\alpha \lambda_\beta y_T^{(\alpha)} y_T^{(\beta)} K_{(\underline{\mathbf{x}}^{(\alpha)},\underline{\mathbf{x}}^{(\beta)})} + \sum_{\alpha=1}^{p} \lambda_\alpha \quad \stackrel{!}{=} \quad \max_{\{\lambda_\alpha\}_{\alpha=1}^{p}}$$

$$\text{s.t.} \qquad \sum_{\alpha=1}^{p} \lambda_\alpha y_T^{(\alpha)} = 0\,, \qquad\qquad 0 \leq \ \lambda_\alpha \ \leq \frac{C}{p}\,.$$

The dual problem

$$-\frac{1}{2}\sum_{\alpha,\beta=1}^{p}\lambda_\alpha\lambda_\beta y_T^{(\alpha)} y_T^{(\beta)} K_{\alpha\beta} + \sum_{\alpha=1}^{p}\lambda_\alpha \quad \stackrel{!}{=} \quad \max_{\{\lambda_\alpha\}_{\alpha=1}^{p}}$$

$$\text{s.t.} \qquad \sum_{\alpha=1}^{p}\lambda_\alpha y_T^{(\alpha)} = 0\,, \qquad\qquad 0 \leq \quad \lambda_\alpha \quad \leq \frac{C}{p}\,.$$

The Gram matrix $\underline{\mathbf{K}}$

$$K_{\alpha\beta} = K_{(\underline{\mathbf{x}}^{(\alpha)},\underline{\mathbf{x}}^{(\beta)})}$$

	1	2	3	$\dots$	j
1	K_{11}	K_{12}	$\dots$	$\dots$	K_{1j}
2	$\vdots$	$\vdots$	K_{23}	$\dots$	K_{2j}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
i	K_{i1}	K_{i2}	$\dots$	$\dots$	K_{ij}

The dual problem

$$-\frac{1}{2}\sum_{\alpha,\beta=1}^{p} \lambda_\alpha \lambda_\beta y_T^{(\alpha)} y_T^{(\beta)} K_{\alpha\beta} + \sum_{\alpha=1}^{p} \lambda_\alpha \quad \stackrel{!}{=} \quad \max_{\{\lambda_\alpha\}_{\alpha=1}^{p}}$$

$$\text{s.t.} \qquad \sum_{\alpha=1}^{p} \lambda_\alpha y_T^{(\alpha)} = 0\,, \qquad\qquad 0 \leq \quad \lambda_\alpha \quad \leq \frac{C}{p}\,.$$

- SVMs operate on pairwise (similarity) data!
- positive definite kernel $\rightarrow$ positive definite Gram matrix $\underline{\mathbf{K}}$
  $\Rightarrow$ well defined optimization problem
- $\underline{\mathbf{K}}$ should be pre-computed to speed up subsequent computations.

The SMO procedure

$$-\frac{1}{2}\sum_{\alpha,\beta=1}^{p}\lambda_\alpha\lambda_\beta y_T^{(\alpha)}y_T^{(\beta)}K_{\alpha\beta}+\sum_{\alpha=1}^{p}\lambda_\alpha \quad \stackrel{!}{=} \quad \max_{\{\lambda_\alpha\}_{\alpha=1}^{p}}$$

$$\text{s.t.} \qquad \sum_{\alpha=1}^{p}\lambda_\alpha y_T^{(\alpha)} = 0\,, \qquad\qquad 0 \leq \quad \lambda_\alpha \quad \leq \frac{C}{p}\,.$$

while *not converged* **do**

  Choose two Lagrange multipliers $\lambda_\gamma, \lambda_\delta$.

  Optimize the constrained Lagrangian while changing only λ_γ and λ_δ.

end

Choosing λ_γ and λ_δ based on KKT

Karush-Kuhn-Tucker conditions (KKT conditions)

$$\underbrace{\left[y_T^{(\alpha)} \left(\underline{\mathbf{w}}^T \underline{\mathbf{x}}^{(\alpha)} + b \right) - 1 + \varphi_\alpha \right]}_{\substack{\text{constraint of the primal problem:} \\ =0 \text{ for all data points on and} \\ \text{within the margin}}} \underbrace{\lambda_\alpha}_{\substack{=0 \text{ for all data} \\ \text{points outside} \\ \text{the margin}}} = 0 \qquad \text{(KKT)}$$

1. loop over all λ_γ violating KKT-conditions
   (and additional "threshold"-conditions due to errors in b)
   pick λ_γ for which KKT $\neq 0$
2. for this λ_γ: select λ_δ yielding a "large step" towards optimum (general heuristics, difference in relative errors $f(x^{(\alpha)}) - y^{(\alpha)}$ vs. $f(x^{(\beta)}) - y^{(\beta)}$)

Reduced optimization problem

$$\min_{(\lambda_\alpha)} \quad \overset{!}{=} \quad \frac{1}{2} \sum_{\alpha\beta} \lambda_\alpha \lambda_\beta y_T^{(\alpha)} y_T^{(\beta)} K_{\alpha\beta} - \sum_{\alpha} \lambda_\alpha$$

Reduced optimization problem

$$\min_{(\lambda_\alpha)} \stackrel{!}{=} \frac{1}{2}\sum_{\alpha\beta}\lambda_\alpha\lambda_\beta y_T^{(\alpha)}y_T^{(\beta)}K_{\alpha\beta} - \sum_\alpha \lambda_\alpha$$

$$\min_{(\lambda_\delta,\lambda_\gamma)} \stackrel{!}{=} \frac{1}{2}\Bigg[\lambda_\gamma^2 \overbrace{\underbrace{\left(y_T^{(\gamma)}\right)^2}_{=1} K_{\gamma\gamma}}^{Q_{\gamma\gamma}} + \lambda_\delta^2 \overbrace{\underbrace{\left(y_T^{(\delta)}\right)^2}_{=1} K_{\delta\delta}}^{Q_{\delta\delta}} + 2\lambda_\gamma\lambda_\delta \overbrace{y_T^{(\gamma)}y_T^{(\delta)}K_{\gamma\delta}}^{Q_{\gamma\delta}}\Bigg]$$

$$+ \lambda_\gamma \underbrace{\left[\sum_{\beta\neq\delta,\gamma}\lambda_\beta y_T^{(\gamma)}y_T^{(\beta)}K_{\gamma\beta} - 1\right]}_{C_\gamma} + \lambda_\delta \underbrace{\left[\sum_{\beta\neq\gamma,\delta}\lambda_\beta y_T^{(\delta)}y_T^{(\beta)}K_{\delta\beta} - 1\right]}_{C_\delta} + \mathrm{const}_{(\lambda_\delta,\lambda_\gamma)}$$

$$\min_{(\lambda_\delta,\lambda_\gamma)} \stackrel{!}{=} \frac{1}{2}\Big[\lambda_\gamma^2 Q_{\gamma\gamma} + \lambda_\delta^2 Q_{\delta\delta} + 2\lambda_\gamma\lambda_\delta Q_{\gamma\delta}\Big] + C_\gamma\lambda_\gamma + C_\delta\lambda_\delta$$

Sequential Minimal Optimization (SMO)

- optimize

$$\min_{(\lambda_\delta, \lambda_\gamma)} \stackrel{!}{=} \frac{1}{2}\Big[\lambda_\gamma^2 Q_{\gamma\gamma} + \lambda_\delta^2 Q_{\delta\delta} + 2\lambda_\gamma \lambda_\delta Q_{\gamma\delta}\Big] + C_\gamma \lambda_\gamma + C_\delta \lambda_\delta$$

- under the following "box" and "equality" constraints

$$0 \leq \lambda_{\gamma,\delta} \leq \frac{C}{p}\,, \qquad \text{(i)}$$

$$\lambda_\gamma + \underbrace{\frac{y_T^{(\delta)}}{y_T^{(\gamma)}}}_{s} \lambda_\delta = \underbrace{- \frac{1}{y_T^{(\gamma)}} \sum_{\beta \neq \gamma, \delta} \lambda_\beta y_T^{(\beta)}}_{d} \quad \Rightarrow \quad \lambda_\gamma + s\lambda_\delta = -d \qquad \text{(ii)}$$

- Analytical solution: Schoelkopf & Smola, p. 308
- Pseudocode: Schoelkopf & Smola, p. 313
- Software: www.csie.ntu.edu.tw/~cjlin/libsvm/
  (also covers multiclass problems, support vector regression, one-class SVMs)

Remarks

Sequential Minimal Optimization (SMO) ...

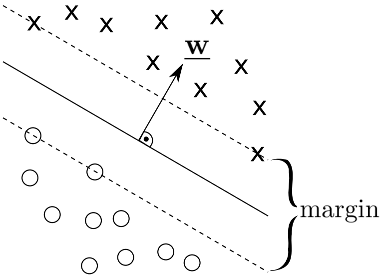
- ...exploits that for 2 constraints the optimization problem can be solved analytically
- ...needs little memory ($\approx$ number of datapoints)
- ...can be much faster than other algorithms
- ...convergence speed depends on rules to select the λ_i
  $\rightsquigarrow$ good heuristics are important

End of Section 2.2

the following slides contain

OPTIONAL MATERIAL

Classification margin



- **margin**: minimal (nomalized) distance to hyperplane $\mathrm{d}^{\mathsf{min}} = \frac{1}{\|\underline{\mathbf{w}}\|}$
- large margins have low *ambiguity* $\Rightarrow$ low VC-dimension

The solution of the primal problem

Lagrangian

$$L = \frac{1}{2}\|\underline{\mathbf{w}}\|^2 - \sum_{\alpha=1}^{p} \lambda_\alpha \left\{ y_T^{(\alpha)} \left(\underline{\mathbf{w}}^T \underline{\mathbf{x}}^{(\alpha)} + b \right) - 1 \right\}$$

- setting derivative w.r.t. weights w_l to zero: $\underline{\mathbf{w}} = \sum_{\alpha=1}^{p} \lambda_\alpha \, y_T^{(\alpha)} \underline{\mathbf{x}}^{(\alpha)}$

$$\frac{\partial L}{\partial \mathrm{w}_l} \quad = \quad \mathrm{w}_l - \sum_{\alpha=1}^{p} \lambda_\alpha y_T^{(\alpha)} \mathrm{x}_l^{(\alpha)} \quad \stackrel{!}{=} \quad 0$$

The solution of the primal problem

Lagrangian

$$L = \frac{1}{2}\|\underline{\mathbf{w}}\|^2 - \sum_{\alpha=1}^{p} \lambda_\alpha \left\{ y_T^{(\alpha)} \left(\underline{\mathbf{w}}^T \underline{\mathbf{x}}^{(\alpha)} + b \right) - 1 \right\}$$

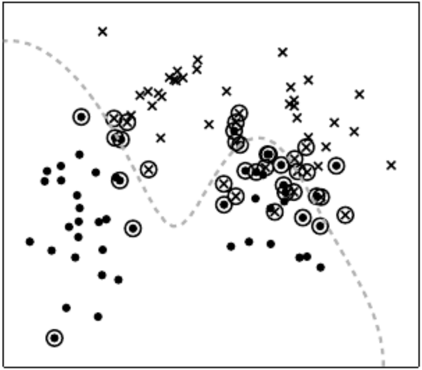
- setting derivative w.r.t. weights w_l to zero: $\underline{\mathbf{w}} = \sum_{\alpha=1}^{p} \lambda_\alpha \, y_T^{(\alpha)} \underline{\mathbf{x}}^{(\alpha)}$

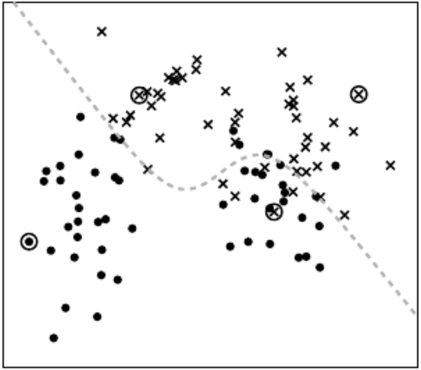
$$\frac{\partial L}{\partial \mathrm{w}_l} = \mathrm{w}_l - \sum_{\alpha=1}^{p} \lambda_\alpha y_T^{(\alpha)} \mathrm{x}_l^{(\alpha)} \stackrel{!}{=} 0$$

- setting derivative w.r.t. b to zero

$$\frac{\partial L}{\partial b} = -\sum_{\alpha=1}^{p} \lambda_\alpha y_T^{(\alpha)} \stackrel{!}{=} 0$$

Sparse Bayesian Regression: Relevance Vector Machines





see Tipping (2001)