extended version: Biehl–Part1.pdf

Support Vector Machine (streamlined)

Michael Biehl
Bernoulli Institute for Mathematics,
Computer Science and Artificial Intelligence
University of Groningen

www.cs.rug.nl/biehl

Solving the perceptron storage problem

re-write the problem ...

consider a given data set $I\!D = \{\boldsymbol{\xi}^\mu, S_R^\mu\}$

... find a vector **w** with $S_H^\mu = \mathrm{sign}(\mathbf{w} \cdot \boldsymbol{\xi}^\mu) = S_R^\mu$ for all μ

Note: $\mathrm{sign}(\mathbf{w} \cdot \boldsymbol{\xi}^\mu) = S_R^\mu \Leftrightarrow \mathrm{sign}(\mathbf{w} \cdot \boldsymbol{\xi}^\mu \, S_R^\mu) = 1 \Leftrightarrow E^\mu = \mathbf{w} \cdot \boldsymbol{\xi}^\mu \, S_R^\mu > 0$

(*local potentials* E^μ)

equivalent problem: solve a set of **linear inequalities** (in **w**)

... find a vector **w** with $E^\mu = \mathbf{w} \cdot \boldsymbol{\xi}^\mu \, S_R^\mu \geq c > 0$ for all μ

solving equations ?

Instead of *inequalities*, try to solve P *equations* for N unknowns:

$$E^\mu = \sum_{j=1}^{N} w_j \xi_j^\mu \, S^\mu \; = \; 1 \quad \text{for all} \;\; \mu = 1, 2, \ldots, P$$

(A) if no solution exists, find **approximation** by least square dev.:

$$\text{minimize}\, f \;=\; \frac{1}{2} \sum_{\mu=1}^{P} \left(1 - \; E^\mu\right)^2$$

minimization, e.g. by means of **gradient descent** with

$$\nabla_w f \;=\; -\sum_{\mu=1}^{P} \left(1 - E^\mu\right) \boldsymbol{\xi}^\mu \, S^\mu$$

solving equations ?

(B) if the system is under-determined → find a unique solution:

$$\text{minimize } \frac{1}{2} \mid \mathbf{w} \mid^2 \quad \text{under constraints} \quad \{E^\mu = 1\}_{\mu=1}^{P}$$

Lagrange function $$L = \frac{1}{2} \mid \mathbf{w} \mid^2 + \sum_{\mu=1}^{P} \lambda^\mu \left(1 - E^\mu\right)$$

necessary conditions for optimum: $$\frac{\partial L}{\partial \lambda^\mu} = \left(1 - E^\mu\right) \stackrel{!}{=} 0$$

$$\nabla_w L = \mathbf{w} - \sum_{\mu=1}^{P} \lambda^\mu \boldsymbol{\xi}^\mu S^\mu \stackrel{!}{=} 0 \qquad \Rightarrow \mathbf{w} = \sum_{\mu=1}^{P} \lambda^\mu \boldsymbol{\xi}^\mu S^\mu$$

Lagrange parameters ~ **embedding strengths** λ^μ (rescaled with N)

solution is a linear combination of the data

solving equations ?

eliminate weights:

$$E^\nu = \sum_{\mu=1}^{P} \underbrace{\frac{1}{N} \sum_{k=1}^{N} (\xi_k^\mu S^\mu)\ (\xi_k^\nu S^\nu)}_{\equiv C^{\nu\mu}}\ \lambda^\mu \qquad \sum_{j=1}^{N} w_j^2 \propto \sum_{\mu,\nu} \lambda^\nu\, C^{\nu\mu}\, \lambda^\mu$$

simplified problem: $$\max_\lambda L = -\frac{1}{2} \sum_{\mu,\nu} \lambda^\nu\, C^{\nu\mu}\, \lambda^\mu + \sum_\mu \lambda^\mu$$

gradient ascent with: $$\frac{\partial L}{\partial \lambda^\rho} = 1 - \sum_\mu C^{\rho\mu} \lambda^\mu = (1 - E^\rho)$$

---

in terms of weights:
the same as in (A) !!!

$$\Delta \mathbf{w} \propto \sum_\rho (1 - E^\rho)\, \boldsymbol{\xi}^\rho\, S^\rho$$

solving equations ?

rename the Lagrange parameters, re-writing the problem:

$$E^\nu = \sum_{\mu=1}^{P} \underbrace{\frac{1}{N} \sum_{k=1}^{N} (\xi_k^\mu S^\mu)\,(\xi_k^\nu S^\nu)}_{\equiv C^{\nu\mu}}\, x^\mu \qquad \sum_{j=1}^{N} w_j^2 \propto \sum_{\mu,\nu} x^\nu\, C^{\nu\mu}\, x^\mu$$

simplified problem: $$\max_x L = -\frac{1}{2} \sum_{\mu,\nu} x^\nu\, C^{\nu\mu}\, x^\mu + \sum_\mu x^\mu$$

gradient ascent with: $$\frac{\partial L}{\partial x^\rho} = 1 - \sum_\mu C^{\rho\mu} x^\mu = (1 - E^\rho)$$

---

in terms of weights:
the same as in (A) !!!

$$\Delta \mathbf{w} \propto \sum_\rho (1 - E^\rho)\, \boldsymbol{\xi}^\rho\, S^\rho$$

classical algorithm: ADALINE

Adaptive Linear Neuron (Widrow and Hoff, 1960)

Adaline algorithm:

$$\mathbf{w}(t) = \mathbf{w}(t-1) + \eta \left(1 - E^{\mu(t)}\right) \xi^{\mu(t)} S^{\mu(t)}$$

sequence $\mu(t)$ of examples

$$x^{\mu}(t) = x^{\mu}(t-1) + \eta \left(1 - E^{\mu(t)}\right)$$

iteration of weights / embedding strengths

---

more general: training of a linear unit with continuous output

$$\text{minimize}\, f = \frac{1}{2} \sum_{\mu=1}^{P} (h^{\mu} - E^{\mu})^2 \qquad \text{with} \quad h^{\mu} \in I\!R, \;\; \mu = 1, 2 \ldots, P$$

$$f = \frac{1}{2} \sum_{\mu=1}^{P} \left(y^{\mu} - \mathbf{w}^{\top} \boldsymbol{\xi}^{\mu}\right)^2 \qquad \text{with} \quad y^{\mu} = h^{\mu}\, S^{\mu}$$

gradient based learning for linear regression (MSE)
frequent strategy: regression as a proxy for classification

hardware realization “Science in action” ca. 1960

youtube video “science in action” with Bernard Widrow

http://www.youtube.com/watch?v=IEFRtz68m-8

Introduction:

- supervised learning, clasification, regression
- machine learning “vs.” statistical modeling

Early (important!) approaches:

- linear threshold classifier, Rosenblatt’s *Perceptron*
- adaptive linear neuron, Widrow and Hoff’s *Adaline*

From **Perceptron** to **Support Vector Machine**

- large margin classification
- beyond linear separability

Distance-based systems

- prototypes: K-means and Vector Quantization
- from K-Neares_Neighbors to Learning Vector Quantization
- adaptive distance measures and relevance learning

Optimal stability by quadratic optimization

$$\text{minimize} \;\; \frac{1}{2}\mathbf{w}^2 \;\; \text{subject to inequality constraints} \;\; \left\{ E^\mu = \mathbf{w}^\top \boldsymbol{\xi}^\mu \, S_R^\mu \; \geq 1 \right\}_{\mu=1}^{P}$$

Note: the solution $\mathbf{w}_{max}$ of the problem yields stability $\kappa_{max} = \dfrac{1}{|\,\mathbf{w}_{max}\,|}$

Notation:

correlation matrix $C \in \mathbb{R}^{P\times P}$ (outputs incorporated)

with elements $$C^{\mu\nu} = \frac{1}{N} S_R^{\mu} S_R^{\nu} \, \boldsymbol{\xi}^{\mu} \cdot \boldsymbol{\xi}^{\nu} = \frac{1}{N} S_R^{\mu} S_R^{\nu} \sum_{j=1}^{N} \xi_j^{\mu} \xi_j^{\nu}$$

***P*-vectors:** $\vec{x} = \left(x^1, x^2, \dots x^P\right)^\top \in \mathbb{R}^P, \quad \vec{E} = \left(E^1, E^2, \dots E^P\right)^\top \in \mathbb{R}^P$

inequalities $\vec{a} \geq \vec{b}$ iff $a^{\mu} \geq b^{\mu}$ for all $\mu = 1, 2, \dots P$

"one-vector": $\vec{1} = (1, 1, \dots, 1)^\top \in \mathbb{R}^P$

$$\vec{E} = C\,\vec{x} \qquad \text{with components} \quad E^{\mu} = \sum_{\nu=1}^{P} C^{\mu\nu} x^{\nu} = \left(\frac{1}{N} \sum_{\nu=1}^{P} x^{\nu} \boldsymbol{\xi}^{\nu} S_R^{\nu} \right) \cdot \boldsymbol{\xi}^{\mu} S_R^{\mu}$$

$$\mathbf{w}^2 = \frac{1}{N} \vec{x}^\top C\, \vec{x} \geq 0 \qquad \text{quadratic form} \quad \mathbf{w}^2 = \frac{1}{N} \sum_{\mu=1}^{P} x^{\mu} E^{\mu} = \frac{1}{N} \sum_{\mu,\nu=1}^{P} x^{\mu} C^{\mu\nu} x^{\nu}$$

(C is positive semi-definite)

Optimal stability by quadratic optimization

$$\text{minimize} \;\; \frac{1}{2}\mathbf{w}^2 \;\; \text{subject to inequality constraints} \;\; \left\{ E^\mu = \mathbf{w}^\top \boldsymbol{\xi}^\mu \, S_R^\mu \; \geq 1 \right\}_{\mu=1}^{P}$$

Note: the solution $\mathbf{w}_{max}$ of the problem yields stability $\kappa_{max} = \dfrac{1}{|\,\mathbf{w}_{max}\,|}$

We can formulate optimal stability completely in terms of embedding strengths:

$$\underset{\vec{x}}{\text{minimize}} \quad \frac{1}{2}\,\vec{x}^\top C\,\vec{x} \qquad \text{subject to linear constraints} \quad \vec{E} = C\,\vec{x} \geq \vec{1}$$

This is a special case of a standard problem in *Quadratic Programming*:
minimize a nonlinear function under linear inequality constraints

Optimization theory: **Kuhn–Tucker theorem**

see, e.g., R. Fletcher, Practical Methods of Optimization (Wiley, 1987)

or http://wikipedia.org "Karush-Kuhn-Tucker-conditions" for a quick start

necessary conditions for a local solution of a general

non-linear optimization problem with equality and inequality constraints

Max. stability: **Kuhn–Tucker theorem** for a special non-linear optimization problem

$$\text{minimize}_{\vec{x}} \ \frac{1}{2}\,\vec{x}^\top C\,\vec{x} \ \ \text{subject to} \ \ C\vec{x} \geq \vec{1}$$

Lagrange function: $\mathcal{L}(\vec{x},\vec{\lambda}) \ = \ \frac{1}{2}\,\vec{x}^\top C\,\vec{x} - \vec{\lambda}^\top (C\vec{x} - \vec{1})$

Any solution can be represented by a **Kuhn-Tucker (KT) point** $\vec{x}^*$ with:

$\vec{x}^* \geq \vec{0} \quad (\vec{x}^* \neq \vec{0})$	**non-negative** embedding strengths (←minover)
$C\,\vec{x}^* \geq \vec{1}$	**linear separability**
$x^{*\mu}\,(1 - [C\,\vec{x}^*]^\mu) = 0 \ \ \text{for all} \ \ \mu$	**complementarity**
	implies also: $\vec{x}^{*T}\,C\,\vec{x}^* = \vec{x}^{*T}\,\vec{E}^* = \vec{x}^{*T}\,\vec{1}$

straightforward to show:

→ all KT-points yield the same **unique** perceptron weight vector

→ any local solution is **globally optimal**

Duality, theory of Lagrange multipliers $\rightarrow$ equivalent formulation (*Wolfe dual*):

$$\underset{\vec{x}}{\text{maximize}} \;\; \widetilde{f} = -\frac{1}{2}\,\vec{x}^T\,C\,\vec{x} + \vec{x}^T\,\vec{1}$$

absent in the Adaline problem

Duality, theory of Lagrange multipliers $\rightarrow$ equivalent formulation (*Wolfe dual*):

$$\max_{\vec{x}}\text{imize } \widetilde{f} = -\frac{1}{2}\,\vec{x}^T C\,\vec{x} + \vec{x}^T\,\vec{1} \quad \text{subject to} \quad \vec{x} \geq 0$$

AdaTron algorithm: (**Ada**ptive Percep**Tron**) [Anlauf and Biehl, 1989]

– sequential presentation of examples $I\!D = \{\,\boldsymbol{\xi}^\mu, S^\mu\,\}$

– gradient ascent w.r.t. $\widetilde{f}$, projected onto $\vec{x} \geq 0$

$$x^\mu \;\rightarrow\; \max\left\{0,\, x^\mu + \overbrace{\eta\,(1 - [C\vec{x}]^\mu\,)}^{\eta\,\left[\nabla_{\vec{x}}\,\widetilde{f}\right]^\mu}\right\} \qquad (0 < \eta < 2)$$

Duality, theory of Lagrange multipliers $\rightarrow$ equivalent formulation (*Wolfe dual*):

$$\underset{\vec{x}}{\text{maximize}} \;\; \widetilde{f} = -\frac{1}{2}\,\vec{x}^T C\,\vec{x} + \vec{x}^T\,\vec{1} \quad \text{subject to} \quad \vec{x} \geq 0$$

AdaTron algorithm: (**Ada**ptive Percep**Tron**) [Anlauf and Biehl, 1989]

– sequential presentation of examples $I\!D = \{\, \boldsymbol{\xi}^\mu, S^\mu \,\}$

– gradient ascent w.r.t. $\widetilde{f}$, projected onto $\vec{x} \geq 0$

$$x^\mu \;\rightarrow\; \max\,\{\, 0,\, x^\mu + \eta\,(\, 1 - [C\vec{x}]^\mu \,)\,\} \qquad (0 < \eta < 2)$$

for the proof of convergence one can show:

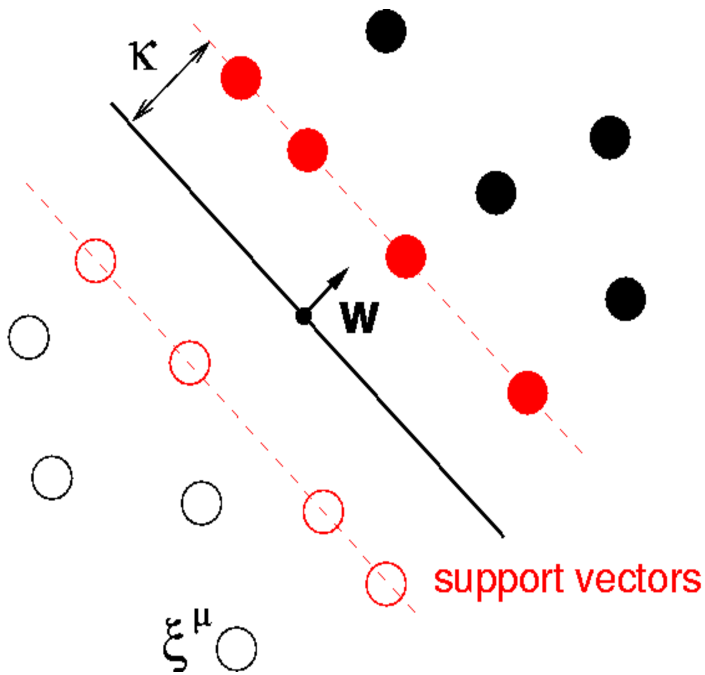
- for an arbitrary $\vec{x} \geq 0$ and a KT point $\vec{x}^*$: $\widetilde{f}(\vec{x}^*) \geq \widetilde{f}(\vec{x})$
- $\widetilde{f}(x)$ is bounded from above in $\vec{x} \geq 0$
- $\widetilde{f}(x)$ increases in every cycle through $I\!D$, unless a KT point has been reached

Support Vectors

complementarity condition: $x^\mu\,(1 - E^\mu) = 0$ for all μ

i.e. either $\left\{ \begin{array}{ccc} E^\mu & = & 1 \\ x^\mu & \geq & 0 \end{array} \right\}$ or $\left\{ \begin{array}{ccc} E^\mu & > & 1 \\ x^\mu & = & 0 \end{array} \right\}$

examples ... have to be embedded or ... are stabilized "automatically"



the weights $\mathbf{w} \propto \sum_\mu x^\mu \boldsymbol{\xi}^\mu S^\mu$

depend (explicitly) only on a subset of $I\!D$

if these support vectors were known

in advance, training could be restricted

to the subset

(unfortunately they are not...)

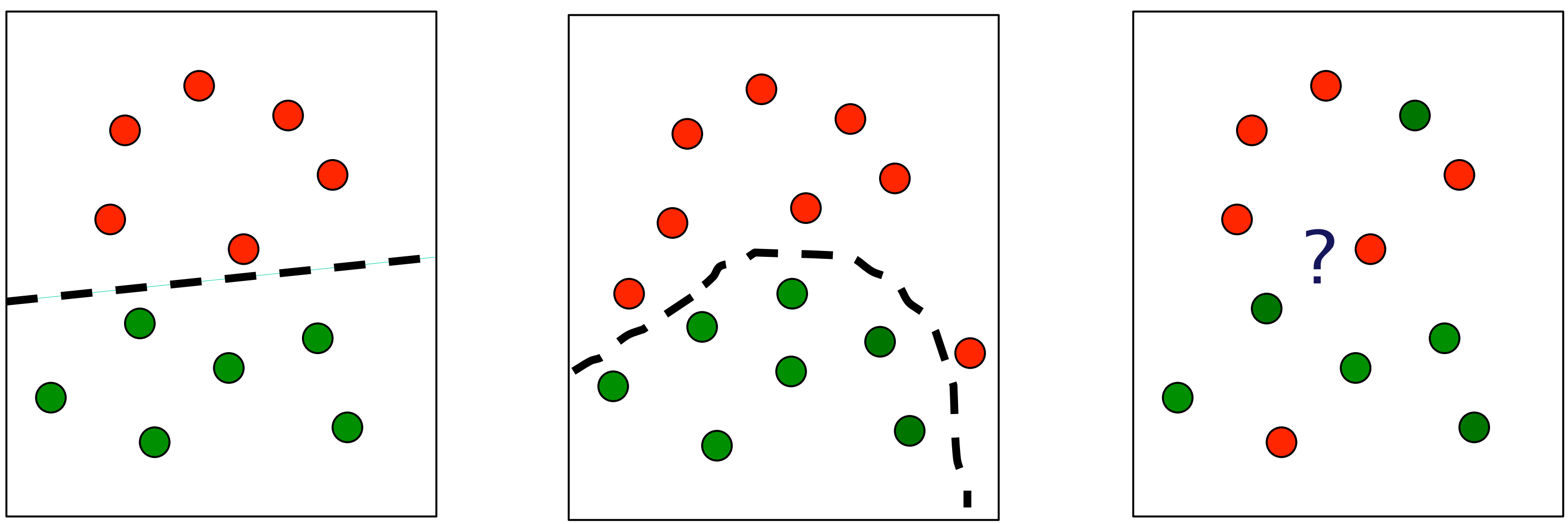
learning in version space ?

... (including max. stability) is only possible if

- the data set is linearly separable

... even then, it only makes sense if

- the unknown rule is a linearly separable function
- the data set is reliable (*noise-free*)



lin. separable — nonlin. boundary — noisy data (?)

Classification beyond linear separability

assume $\mathbb{D} = \{\boldsymbol{\xi}^\mu, S^\mu\}$ is **not linearly** separable - what can we do?

potential reasons: noisy data, more complex problem

- accept an approximation by a linearly separable function

large margins with errors

admit disagreements w.r.t. training data, but keep basic idea of optimal stability

$$\boxed{\text{minimize}_{\mathbf{w},\beta} \quad \frac{1}{2}\mathbf{w}^2 + \gamma \sum_{\mu=1}^{P} \beta^\mu \quad \text{subject to} \quad E^\mu \geq 1 - \beta^\mu \quad \text{and} \quad \beta^\mu \geq 0 \quad \text{for all} \quad \mu}$$

slack variables
$$\begin{cases} \beta^\mu = 0 \leftrightarrow E^\mu \geq 1 \\ \beta^\mu > 0 \leftrightarrow E^\mu < 1 \quad \text{includes errors with} \quad E^\mu < 0 \end{cases}$$

rewritten in terms of embedding strengths (see above for notation)

$$\text{minimize}_{\vec{x},\vec{\beta}} \quad \frac{1}{2}\,\vec{x}^{\top} C\,\vec{x} + \gamma\,\vec{\beta}\cdot\vec{1} \quad \text{subject to} \;\; C\,\vec{x} \geq \vec{1} - \vec{\beta}$$
$$\text{and} \;\; \vec{\beta} \geq 0$$

dual problem: (elimination of slack variables!)

$$\text{maximize}_{\vec{x}} \quad -\frac{1}{2}\,\vec{x}^{\top} C\,\vec{x} + \vec{1}\cdot\vec{x} \quad \text{subject to} \;\; 0 \leq \vec{x} \leq \gamma\,\vec{1}$$

positive and upper-bounded embedding strengths

parameter γ
- limits the growth of x^{μ} for misclassified data points
- controls a compromise between aims of *large margin* and *low error*
- has to be chosen appropriately, e.g. by validation methods (later chapter)
  note: even for lin. sep. data the optimum can include misclassifications!
- does not (in general) minimize the **number** of errors

example algorithm:

AdaTron with errors (projected gradient ascent)

$$\tilde{x^\mu} \leftarrow x^\mu + \eta \left(1 - [C\vec{x}]^\mu\right) \quad \text{gradient step}$$

$$\hat{x^\mu} \leftarrow \max\left\{0, \tilde{x^\mu}\right\} \quad \text{enforce non-negative embeddings}$$

$$x^\mu \leftarrow \min\left\{\gamma, \hat{x^\mu}\right\} \quad \text{limit embedding strenghts to } x^\mu \leq \gamma$$

Classification beyond linear separability

assume $\mathbb{D} = \{\boldsymbol{\xi}^\mu, S^\mu\}$ is **not linearly** separable - what can we do?

potential reasons: noisy data, more complex problem

- accept an approximation by a linearly separable function

- construct more complex architectures from perceptron-like units.
  e.g. multilayer networks (universal classificators, difficult training)

- consider *ensembles* of perceptrons
  train several student perceptrons $S^{(j)} = \text{sign}\left[\mathbf{w}^{(j)} \cdot \boldsymbol{\xi}\right]$
  combine the $S^{(j)}$ into an *ensemble classifier* , e.g. by majority vote $S_H = \text{sign}\left[\sum_j S^{(j)}\right]$

competing aims:
- each student should make a *small* number of errors
- the perceptrons should differ significantly

see also: Decision Trees and Forests (lectures by Dalya Baron)

- employ a linear decision boundary, but after a non-linear transformation of the data to an M-dim. feature space (M=N is possible, but not required)

$$S_H(\boldsymbol{\xi}) = \text{sign}\left[\underline{W} \cdot \underline{\Psi}(\boldsymbol{\xi})\right] \qquad \text{with} \quad \underline{W} \in \mathbb{R}^M$$

M−dim. weight vector

$$\underline{\Psi}(\boldsymbol{\xi}) \in \mathbb{R}^M$$

non-linear transformation $\mathbb{R}^N \to \mathbb{R}^M$

for a given, explicit transformation $\underline{\Psi}(\boldsymbol{\xi})$, perceptron training can be applied in $\mathbb{R}^M$

The Support Vector Machine

- Perceptron of optimal stability: **support vectors**
- SVM: non-linear transformation to high-dim. feature space
- implicit kernel formulation, Mercer's theorem

history: www.svms.org

- Vapnik and Lerner (1963) introduce the Generalized Portrait algorithm
- Aizerman, Braverman and Rozonoer (1964) introduced the geometrical interpretation of the kernels
- Vapnik and Chervonenkis (1964) further develop the Generalized Portrait algorithm.
- Vapnik (1982) wrote an English translation of his 1979 book.
- SVMs close to their current form were first introduced with a paper at the COLT 1992 conference (Boser, Guyon and Vapnik 1992).
- In 1995 the soft margin classifier was introduced by Cortes and Vapnik (1995)

basic idea:

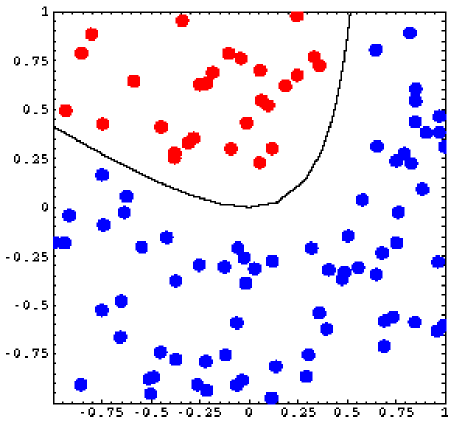
employ a linear decision boundary, but after a non-linear transformation of the data

$S_H^\mu = \text{sign}\,[\,\underline{W} \cdot \underline{\Psi}(\boldsymbol{\xi}^\mu)\,],\qquad \boldsymbol{\xi} \in I\!R^N \;\rightarrow\; \underline{\Psi}(\boldsymbol{\xi}) \in I\!R^M$ with weights $\underline{W} \in I\!R^M$

SVM: transformation with M>N to high-dim. feature space

An illustrative example (c/o R. Dietrich, PhD thesis)

consider original, two-dimensional data (x_1, x_2)



basic idea:

employ a linear decision boundary, but after a non-linear transformation of the data

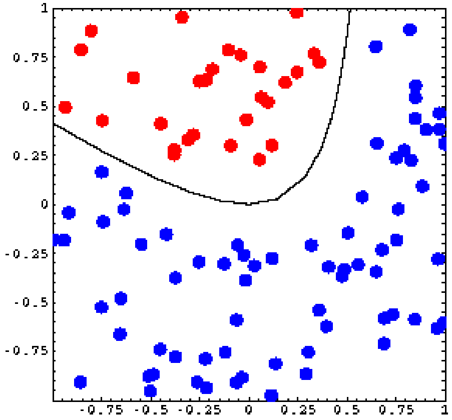
$S_H^\mu = \text{sign}\,[\,\underline{W} \cdot \underline{\Psi}(\boldsymbol{\xi}^\mu)\,]$, $\quad \boldsymbol{\xi} \in \mathbb{R}^N \rightarrow \underline{\Psi}(\boldsymbol{\xi}) \in \mathbb{R}^M$ with weights $\underline{W} \in \mathbb{R}^M$

SVM: transformation with M>N to high-dim. feature space

An illustrative example (c/o R. Dietrich, PhD thesis)

consider original, two-dimensional data (x_1, x_2)

and the non-linear transformed data $\underline{\Psi}(x_1, x_2) = \left(x_1^2,\, \sqrt{2}\, x_1\, x_2,\, x_2\right) \in \mathbb{R}^3$



basic idea:

employ a linear decision boundary, but after a non-linear transformation of the data

$$S_H^\mu = \text{sign}\,[\,\underline{W} \cdot \underline{\Psi}(\boldsymbol{\xi}^\mu)\,], \qquad \boldsymbol{\xi} \in I\!R^N \;\rightarrow\; \underline{\Psi}(\boldsymbol{\xi}) \in I\!R^M$$ with weights $\underline{W} \in I\!R^M$

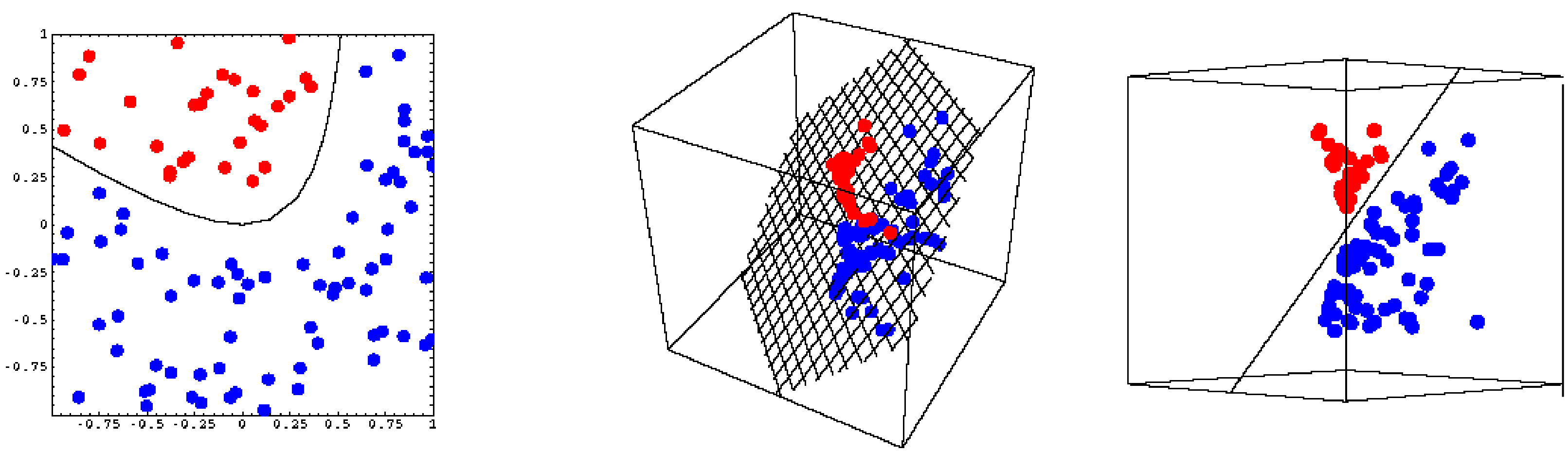
SVM: transformation with M>N to high-dim. feature space

An illustrative example (c/o R. Dietrich, PhD thesis)

consider original, two-dimensional data (x_1, x_2)

and the non-linear transformed data $\underline{\Psi}(x_1, x_2) = \left(x_1^2,\, \sqrt{2}\, x_1\, x_2,\, x_2 \right) \in I\!R^3$

$$S^\mu = \text{sign}\,(\,\underline{W} \cdot \underline{\Psi}(x_1, x_2)\,)$$ with $\vec{W} = (1, 1, -1)$



the non-separable classification in $I\!R^2$ becomes linearly separable in $I\!R^3$

assume: transformation guarantees linear separability of $\{ \underline{\Psi}(\boldsymbol{\xi}^\mu), S^\mu \}$

$\rightarrow$ a vector $\underline{W}$ exists with $S_H^\mu = \operatorname{sign}\left(\underline{W} \cdot \underline{\Psi}(\boldsymbol{\xi}^\mu) \right)$ for all μ.

optimal stability:

$$\underset{\underline{W}}{\text{maximize}} \; \kappa(\underline{W}) \quad \text{where} \quad \kappa(\underline{W}) = \min_{\mu} \left\{ \kappa^\mu = \frac{\underline{W} \cdot \underline{\Psi}(\boldsymbol{\xi}^\mu)\, S^\mu}{|\,\underline{W}\,|} \right\}$$

Exact same structure as the original perceptron problem – all above results from optimization theory apply accordingly assume: transformation guarantees linear separability of $\{\, \underline{\Psi}(\boldsymbol{\xi}^\mu), S^\mu \,\}$

$\longrightarrow$ a vector $\underline{W}$ exists with $S_H^\mu \,=\, \mathrm{sign}\,(\,\underline{W} \,\cdot\, \underline{\Psi}(\boldsymbol{\xi}^\mu)\,)$ for all μ.

optimal stability:

$$\underset{\underline{W}}{\mathrm{maximize}}\;\; \kappa(\underline{W}) \quad \text{where} \quad \kappa(\underline{W}) \,=\, \min_{\mu} \left\{ \kappa^\mu \,=\, \frac{\underline{W}\cdot\underline{\Psi}(\boldsymbol{\xi}^\mu)\, S^\mu}{|\,\underline{W}\,|} \right\}$$

Exact same structure as the original perceptron problem – all above results from optimization theory apply accordingly

re-formulate:

$$\underset{\vec{X}}{\mathrm{minimize}}\, \frac{1}{2}\, \vec{X}^T\, \Gamma\, \vec{X} \quad \text{subject to} \quad \Gamma\, \vec{X} \,\geq\, \vec{1}$$

---

here:

$$\underline{W} \,=\, \frac{1}{M} \sum_{\mu=1}^{P} X^\mu\, \underline{\Psi}(\boldsymbol{\xi}^\mu)\, S^\mu \qquad\qquad \Gamma^{\mu\nu} \,=\, \frac{1}{M}\, S^\mu\, \underline{\Psi}(\boldsymbol{\xi}^\mu) \,\cdot\, \underline{\Psi}(\boldsymbol{\xi}^\nu)\, S^\nu$$

$$\underline{W}^2 \,=\, \frac{1}{M}\, \vec{X}^T\, \Gamma\, \vec{X}$$

Kernel formulation

consider the function $K : I\!R^N \times I\!R^N \rightarrow I\!R$ with $K(\boldsymbol{\xi}^\mu, \boldsymbol{\xi}^\nu) = \frac{1}{M}\,\underline{\Psi}(\boldsymbol{\xi}^\mu) \cdot \underline{\Psi}(\boldsymbol{\xi}^\nu)$

re-write in terms of this *kernel function*

- the classification scheme: $S_H(\boldsymbol{\xi}) = \text{sign}\left(\underline{W} \cdot \underline{\Psi}(\boldsymbol{\xi})\right)$

$$= \text{sign}\left(\sum_{\mu=1}^{P} X^\mu\, S^\mu\, \underline{\Psi}(\boldsymbol{\xi}^\mu) \cdot \underline{\Psi}(\boldsymbol{\xi})\right) = \text{sign}\left(\sum_{\mu=1}^{P} X^\mu\, S^\mu\, K(\boldsymbol{\xi}^\mu, \boldsymbol{\xi})\right)$$

Kernel formulation

consider the function $K : I\!R^N \times I\!R^N \rightarrow I\!R$ with $K(\boldsymbol{\xi}^\mu, \boldsymbol{\xi}^\nu) = \frac{1}{M}\, \underline{\Psi}(\boldsymbol{\xi}^\mu) \cdot \underline{\Psi}(\boldsymbol{\xi}^\nu)$

re-write in terms of this *kernel function*

- the classification scheme: $S_H(\boldsymbol{\xi}) = \text{sign}\,\left(\underline{W} \cdot \underline{\Psi}(\boldsymbol{\xi})\right)$

$$= \text{sign}\left(\sum_{\mu=1}^{P} X^\mu\, S^\mu\, \underline{\Psi}(\boldsymbol{\xi}^\mu) \cdot \underline{\Psi}(\boldsymbol{\xi})\right) = \text{sign}\left(\sum_{\mu=1}^{P} X^\mu\, S^\mu\, K(\boldsymbol{\xi}^\mu, \boldsymbol{\xi})\right)$$

- training algorithms for the embedding strengths, just one example:

Kernel AdaTron $\quad X^\mu \rightarrow \max\left\{0, X^\mu + \eta\left(1 - S^\mu \sum_{\nu=1}^{P} S^\nu\, X^\nu\, K(\boldsymbol{\xi}^\mu, \boldsymbol{\xi}^\nu)\right)\right\}$

Kernel formulation

consider the function $K : I\!R^N \times I\!R^N \rightarrow I\!R$ with $K(\boldsymbol{\xi}^\mu, \boldsymbol{\xi}^\nu) = \frac{1}{M} \underline{\Psi}(\boldsymbol{\xi}^\mu) \cdot \underline{\Psi}(\boldsymbol{\xi}^\nu)$

re-write in terms of this *kernel function*

- the classification scheme: $S_H(\boldsymbol{\xi}) = \text{sign}\left(\underline{W} \cdot \underline{\Psi}(\boldsymbol{\xi}) \right)$

$$= \text{sign} \left(\sum_{\mu=1}^{P} X^\mu S^\mu \underline{\Psi}(\boldsymbol{\xi}^\mu) \cdot \underline{\Psi}(\boldsymbol{\xi}) \right) = \text{sign} \left(\sum_{\mu=1}^{P} X^\mu S^\mu K(\boldsymbol{\xi}^\mu, \boldsymbol{\xi}) \right)$$

- training algorithms for the embedding strengths, just one example:

Kernel AdaTron $\quad X^\mu \rightarrow \max \left\{ 0, X^\mu + \eta \left(1 - S^\mu \sum_{\nu=1}^{P} S^\nu X^\nu K(\boldsymbol{\xi}^\mu, \boldsymbol{\xi}^\nu) \right) \right\}$

– no explicit use of the transformed feature vectors $\underline{\Psi}(\boldsymbol{\xi})$

– only dot-products required, which can be expressed in terms of the *kernel* so far: define non-linear $\underline{\Psi}(\boldsymbol{\xi}) \in I\!R^M$, find corresponding kernel function $K(\boldsymbol{\xi}^\mu, \boldsymbol{\xi}^\nu)$

now: as we will never use $\underline{\Psi}(\boldsymbol{\xi})$ explicitly, why not start with defining a kernel function in the first place?

for practical purposes, we need not know $\underline{\Psi}$ nor its dimension M

Question: does a given kernel K correspond to some valid transformation $\underline{\Psi}$?

so far: define non-linear $\underline{\Psi}(\boldsymbol{\xi}) \in I\!R^M$, find corresponding kernel function $K(\boldsymbol{\xi}^\mu, \boldsymbol{\xi}^\nu)$

now: as we will never use $\underline{\Psi}(\boldsymbol{\xi})$ explicitly, why not start with defining a kernel function in the first place?

for practical purposes, we need not know $\underline{\Psi}$ nor its dimension M

Question: does a given kernel K correspond to some valid transformation $\underline{\Psi}$?

Mercer's Theorem (sufficient condition)

a given kernel function K can be written as $K(\boldsymbol{\xi}^\mu, \boldsymbol{\xi}^\nu) = \underline{\Psi}(\boldsymbol{\xi}^\mu) \cdot \underline{\Psi}(\boldsymbol{\xi}^\nu)$, if

$$\int\int g(\boldsymbol{\xi}^\mu)\, K(\boldsymbol{\xi}^\mu, \boldsymbol{\xi}^\nu)\, g(\boldsymbol{\xi}^\nu)\, d^N\xi^\mu\, d^N\xi^\nu \geq 0$$ holds true

for all functions g with finite norm $\int g(\boldsymbol{\xi})^2\, d^N\boldsymbol{\xi} < \infty$

popular classes of kernels (which satisfy Mercer's conditon)

- **polynomial kernels** of degree (up to) *q*, e.g.

$$K(\boldsymbol{\xi}^\mu, \boldsymbol{\xi}) = (1 + \boldsymbol{\xi}^\mu \cdot \boldsymbol{\xi})^q \quad \text{yields} \quad S_H(\boldsymbol{\xi}) = \text{sign}\left[\sum_{\mu=1}^{P} X^\mu \, S^\mu \, (1 + \boldsymbol{\xi}^\mu \cdot \boldsymbol{\xi})^q\right]$$

linear kernel $(q = 1)$

$$K(\boldsymbol{\xi}^\mu, \boldsymbol{\xi}) = (1 + \boldsymbol{\xi}^\mu \cdot \boldsymbol{\xi}) \quad \text{yields} \quad S_H(\boldsymbol{\xi}) = \text{sign}\left[\Theta + \sum_{\mu=1}^{P} X^\mu \, S^\mu \, \boldsymbol{\xi}^\mu \cdot \boldsymbol{\xi}\right]$$

= perceptron with threshold in original space

$\sum_\mu X^\mu S^\mu$

popular classes of kernels (which satisfy Mercer's conditon)

- **polynomial kernels** of degree (up to) *q*, e.g.

$$K(\boldsymbol{\xi}^\mu, \boldsymbol{\xi}) \;=(1+\,\boldsymbol{\xi}^\mu\cdot\boldsymbol{\xi})^q \qquad \text{yields} \;\; S_H(\boldsymbol{\xi}) \;=\; \text{sign}\left[\sum_{\mu=1}^{P} X^\mu\, S^\mu\,(1+\,\boldsymbol{\xi}^\mu\cdot\boldsymbol{\xi})^q\right]$$

linear kernel $(q=1)$

$$K(\boldsymbol{\xi}^\mu, \boldsymbol{\xi}) \;=(1+\,\boldsymbol{\xi}^\mu\cdot\boldsymbol{\xi}) \qquad \text{yields} \;\; S_H(\boldsymbol{\xi}) \;=\; \text{sign}\left[\Theta+\sum_{\mu=1}^{P} X^\mu\, S^\mu\,\boldsymbol{\xi}^\mu\cdot\boldsymbol{\xi}\right]$$

= perceptron with threshold in original space

$\sum_\mu X^\mu S^\mu$

quadratic kernel $(q=2)$

$$K(\boldsymbol{\xi}^\mu, \boldsymbol{\xi}) \;=\; (1+\boldsymbol{\xi}^\mu\cdot\boldsymbol{\xi})^2 \;==1+2\sum_j \xi_j^\mu \boxed{\xi_j} + \sum_{j,k} \xi_j^\mu \xi_k^\mu \;\boxed{\xi_j\xi_k}$$

-> perceptron with respect to feature vectors containing all single and products of 2 original features

$$\left(\xi_1,\, \xi_2,\, \ldots, \xi_N,\, \xi_1\xi_1,\, \xi_1\xi_2,\, \ldots,\, \ldots, \xi_{N=1}\xi_N,\, \xi_N\xi_N\right)^T \qquad \text{i.e.} \quad M = N + N(N-1)/2$$

- **Radial basis function (RBF) kernel**

$$K(\boldsymbol{\xi}^\mu, \boldsymbol{\xi}) = \exp\left[-\frac{|\boldsymbol{\xi}^\mu - \boldsymbol{\xi}|^2}{2\sigma}\right]$$

involves all powers of the features, "$M \to \infty$"

so much for the "curse of dimensionality" ☺

---

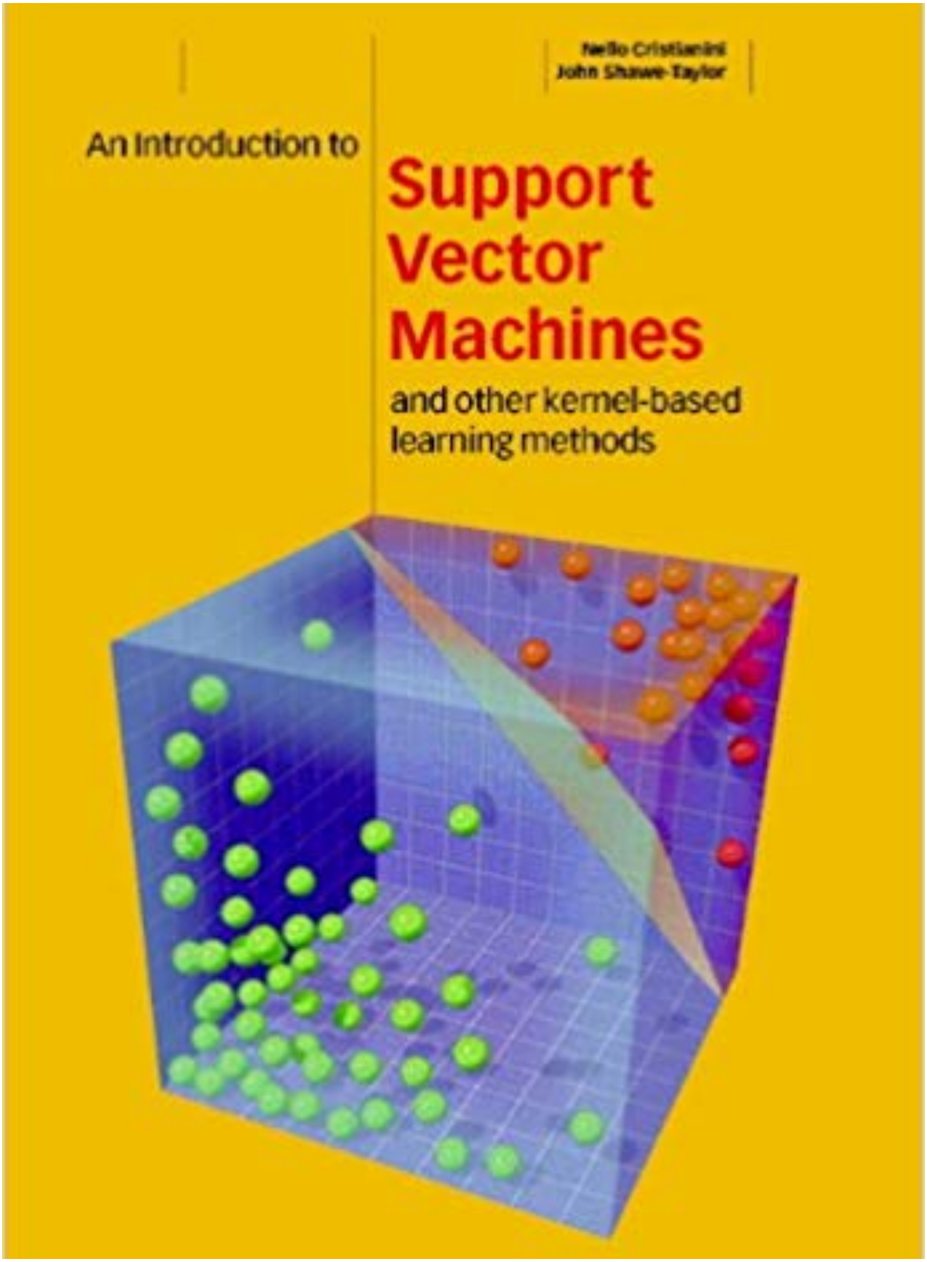
attractive aspects of the SVM approach:

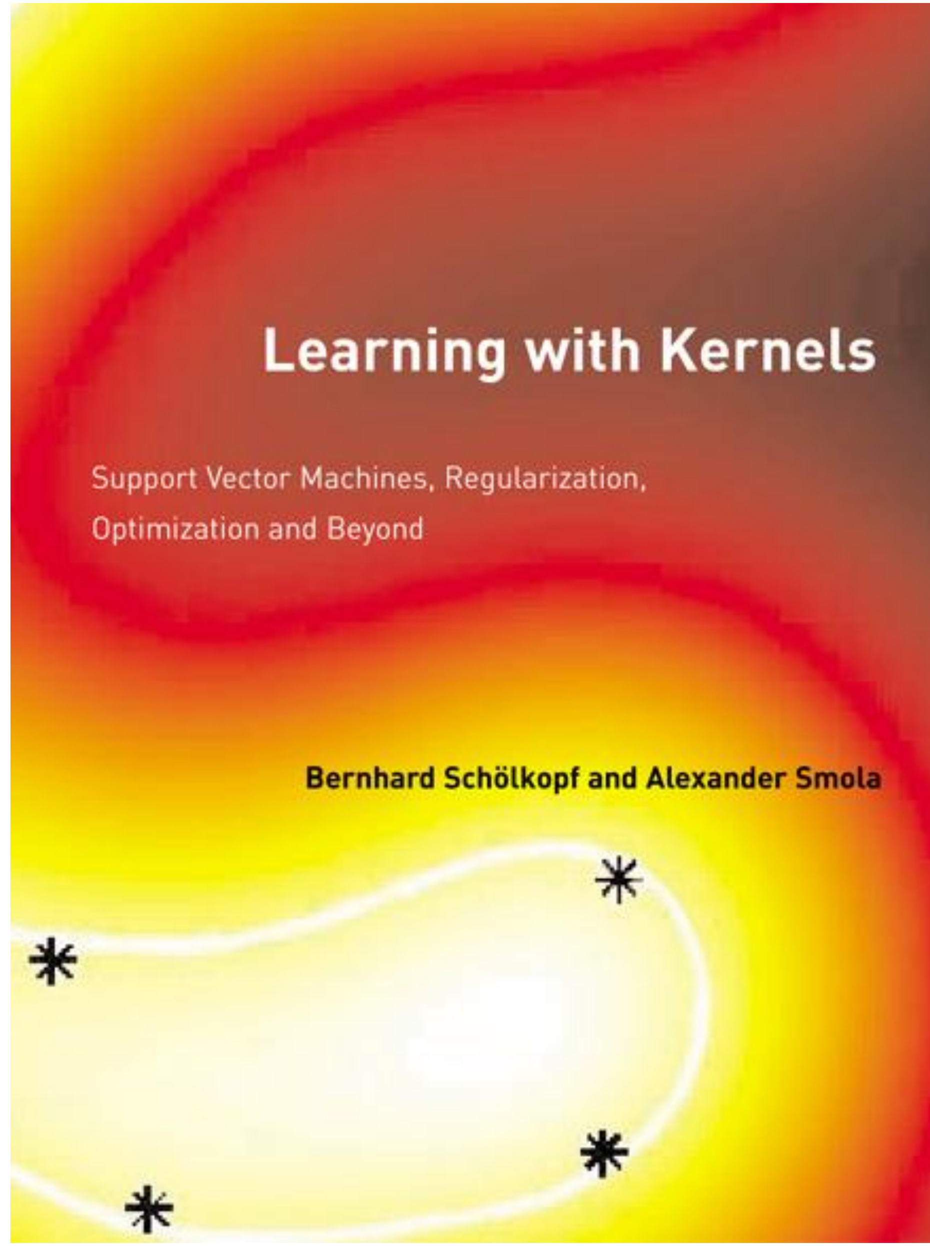
- optimization problem is uniquely solvable (no local minima)
- efficient training algorithms are known ("kernelized" max. stability algorithms)
- maximum stability facilitates good generalization ability

… if the kernel (its parameters) is (are) appropriately chosen

in practice:

- select simple kernels, allow for violations of some of the linear constraints by means of slack variables (e.g. kernel-version of Adatron with errors, see above)
- choose kernel (kernel parameters) by means of cross-validation procedures
- use approximate schemes for huge amounts of data (many support vectors)


Nello Cristianini
John Shawe-Taylor
An Introduction to
Support Vector Machines
and other kernel-based learning methods


Learning with Kernels
Support Vector Machines, Regularization, Optimization and Beyond
Bernhard Schölkopf and Alexander Smola