Support Vector Networks

Anastasios Giovanidis

Sorbonne-LIP6

Mars 18, 2020

Bibliography

A. Giovanidis 2020

Ref.1 Corinna Cortes and Vladimir Vapnik. "Support-Vector Networks". Machine Learning, 20, 273-297 (1995)

Ref.2 Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani. "An introduction to statistical learning: with applications in R". Springer Texts in Statistics.

☞ Chapter 09

ISBN 978-1-4614-7137-0 (DOI 10.1007/978-1-4614-7138-7)

Ref.3 John C. Platt. "Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods". Microsoft Research, March 26, 1999.

Ref.4 Alex J. Smola and Bernhard Schölkopf. "A tutorial on support vector regression". Statistics and Computing 14:199-222, (2004)

all figures in the slides taken by Ref.2 (except when stated)

SVM

A. Giovanidis 2020

Support Vector Networks (or Support Vector Machines aka SVMs) can be used for:

- Binary classification
- Regression

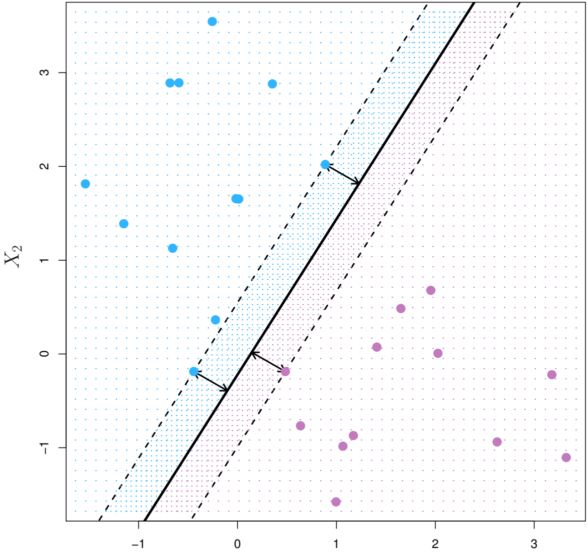
☞ As classifiers (most of this talk) they introduce boundaries with a maximal margin between the two classes.

- The margin can be hard or soft.

- The boundary can be linear (hyper-plane) or non-linear.

A. Giovanidis 2020

Linear Boundary



Linear Boundary

A. Giovanidis 2020

Train Data: $x_1 = (x_{1,1}, \ldots, x_{1,p})$, ..., $x_n = (x_{n,1}, \ldots, x_{n,p})$ – p features.
Labels: $y_1, \ldots, y_n \in \{-1, 1\}$.

Classification based on hyperplane
Given a hyperplane with $\mathbf{w}$ and b, we decide:

- If $f(\mathbf{x}_i) = \mathbf{w} \cdot \mathbf{x}_i + b > 0$, then $y_i = 1$,
- If $f(\mathbf{x}_i) = \mathbf{w} \cdot \mathbf{x}_i + b < 0$, then $y_i = -1$.

☞ If $f(\mathbf{x}_i)$ is close to zero, then we are less certain of the correctness of the decision. If it is far from zero, then the $\mathbf{x}_i$ lies far from the hyperplane and we are more certain of its class.

Maximal Hard Margin

A. Giovanidis 2020

The train data is linearly separable if there exists a **w** and b, such that

- $\mathbf{w} \cdot \mathbf{x}_i + b \geq 1$, if $y_i = 1$,
- $\mathbf{w} \cdot \mathbf{x}_i + b \leq -1$, if $y_i = -1$.

In compact form

$$y_i \cdot (\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, \;\; i = 1, \ldots, n$$

When the data can be optimally separated by a hyperplane (class 1 right, and class -1 left), then there exist many ways to choose a hyperplane.

The **optimal** is furthest away from the observations. It is unique.

$$f(\mathbf{x}) = \mathbf{w}_0 \cdot \mathbf{x} + b_0 = 0$$

Maximal distance

A. Giovanidis 2020

We need to find the direction $\mathbf{w}/|\mathbf{w}|$ where the distance between the projections of the training vectors of two different classes is maximal,

$$\rho(\mathbf{w}, b) = \min_{(\mathbf{x}:y=1)} \frac{\mathbf{x}\cdot\mathbf{w}}{|\mathbf{w}|} - \max_{(\mathbf{x}:y=-1)} \frac{\mathbf{x}\cdot\mathbf{w}}{|\mathbf{w}|}$$

Note that

- for $y_i = 1$ it holds $\mathbf{w}\cdot\mathbf{x}_i \geq 1 - b$, and
- for $y_i = -1$ it holds $\mathbf{w}\cdot\mathbf{x}_i \leq -1 - b$.

$$\rho(\mathbf{w}, b) = \frac{1-b}{|\mathbf{w}|} - \frac{-1-b}{|\mathbf{w}|} = \frac{2}{\sqrt{\mathbf{w}\cdot\mathbf{w}}}$$

Margin optimisation

A. Giovanidis 2020

We want to solve the problem of the optimal margin

$$\begin{array}{ll} \min & \Phi = \frac{1}{2}\mathbf{w}\cdot\mathbf{w} \\ \text{s.t.} & y_i \cdot (\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1,\ \ i = 1, \ldots, n \end{array}$$

The Lagrangian is

$$L(\mathbf{w}, b, \mathbf{\Lambda}) \ = \ \frac{1}{2}\mathbf{w}\cdot\mathbf{w} - \sum_{i=1}^{n} \lambda_i \left[y_i \cdot (\mathbf{w}\cdot\mathbf{x}_i + b) - 1 \right],$$

so we solve

$$\max_{\Lambda \geq 0} \left(\min_{\mathbf{w}, b} L(\mathbf{w}, b, \mathbf{\Lambda}) \right).$$

Solution primal

A. Giovanidis 2020

Taking partial derivatives on the primal variables

$$\begin{aligned} \frac{\partial L}{\partial \mathbf{w}}\Big|_{\mathbf{w}=\mathbf{w}_0} &= \mathbf{w}_0 - \sum_{i=1}^{n} \lambda_i y_i \mathbf{x}_i = 0 \\ \frac{\partial L}{\partial b}\Big|_{b=b_0} &= \sum_{i=1}^{n} \lambda_i y_i = 0 \end{aligned}$$

So we can write the optimal hyperplane direction as a linear combination of training vectors with $\lambda_i > 0$

$$\mathbf{w}_0 = \sum_{i=1}^{n} \lambda_i y_i \mathbf{x}_i$$

KKT and duals

A. Giovanidis 2020

From the Kuhn-Tucker theorem, at a saddle point $(\mathbf{w}_0, b_0, \mathbf{\Lambda}_0)$ any Lagrange multiplier and its corresponding constraint are connected by the equality

$$\lambda_i^0 \left[y_i \cdot (\mathbf{w}_0 \cdot \mathbf{x}_i + b_0) - 1\right] = 0, \;\; i = 1, \ldots, n$$

Hence,

- if $y_i \cdot (\mathbf{w}_0 \cdot \mathbf{x}_i + b_0) - 1 > 0$, then $\lambda_i^0 = 0$, and
- if $y_i \cdot (\mathbf{w}_0 \cdot \mathbf{x}_i + b_0) - 1 = 0$, then $\lambda_i^0 \geq 0$.

Support Vectors

A. Giovanidis 2020

☞ We call the vectors $\mathbf{x}_i$ for which $y_i \cdot (\mathbf{w}_0 \cdot \mathbf{x}_i + b_0) = 1$ support vectors. Only these vectors can be strictly positive $\lambda_i^0 > 0$,

$$\mathbf{w}_0 = \sum_{s: support \;\; vectors} \lambda_s^0 y_s \mathbf{x}_s .$$

The classification decision based on the optimal hyperplane is

$$I(\mathbf{x}) = sign\,(f\,(\mathbf{x})) = sign\left(\sum_{s: support \;\; vectors} \lambda_s^0 y_s \mathbf{x}_s \cdot \mathbf{x} + b_0\right)$$

A. Giovanidis 2020

Find the duals

To find the optimal dual vector $\mathbf{\Lambda}_0$ solve the **quadratic problem**

$$\begin{aligned} \max \quad & W(\mathbf{\Lambda}) = \mathbf{\Lambda}^T \mathbf{1} - \tfrac{1}{2}\mathbf{\Lambda}^T \mathbf{D} \mathbf{\Lambda} \\ \text{s.t.} \quad & \mathbf{\Lambda} \geq 0 \\ & \mathbf{\Lambda}^T \mathbf{Y} = 0 \end{aligned}$$

$\mathbf{Y}^T$ is the vector of labels, $\mathbf{D}$ is a symmetric $n \times n$ matrix with elements

$$D_{ij} = y_i y_j \mathbf{x}_i \mathbf{x}_j, \quad i, j = 1, \ldots, n.$$

Solving this, we can show that

$$W(\mathbf{\Lambda}_0) = \frac{2}{\rho_0^2},$$

relating the solution of the initial problem to the maximal margin.

Alternative formulation

A. Giovanidis 2020

To understand better what we did, we could just re-write the maximisation problem as

$$
\begin{aligned}
\max_{\mathbf{w},\rho} \quad & \rho \\
\text{s.t.} \quad & \mathbf{w}^T \cdot \mathbf{w} = 1 \\
& y_i \cdot (\mathbf{w} \cdot \mathbf{x}_i + b) \geq \rho, \quad i = 1, \ldots, n
\end{aligned}
$$

Soft Margin

A. Giovanidis 2020

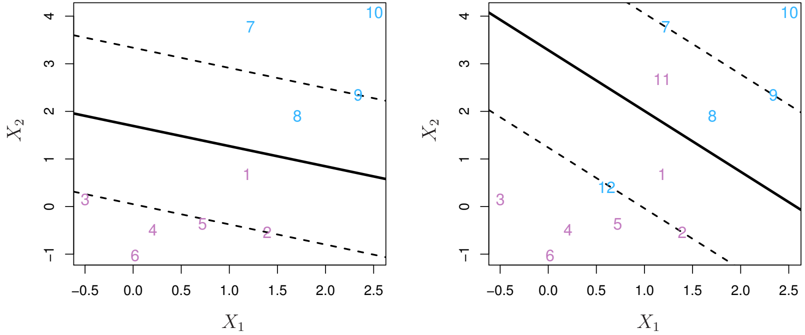
- Even when the data is linearly separable, the addition of just one new vector can change drastically the direction of the separating hyperplane (not robust).
- In many cases no separating hyperplane exists, and so there is no maximal margin classifier.

Solution: Misclassify a few training observations in order to do a better job in classifying the remaining observations.

We allow some observations to be on the incorrect side of the margin, or even the incorrect side of the hyperplane!

Soft Margin (example)

A. Giovanidis 2020



Soft Margin version 1

A. Giovanidis 2020

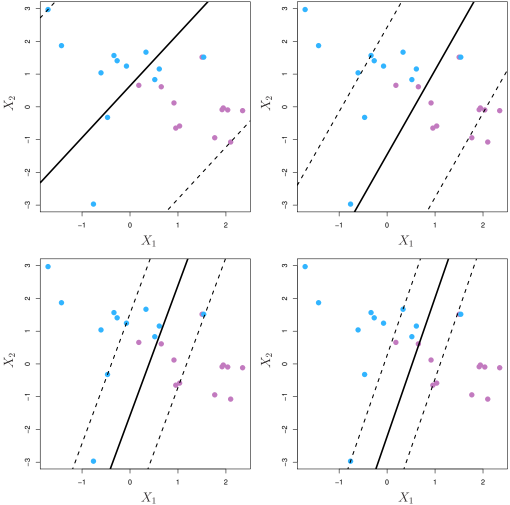
We can introduce slack variables in the optimisation problem

$$
\begin{array}{ll}
\max_{\mathbf{w},\rho,\xi} & \rho \\
 & \\
\text{s.t.} & \mathbf{w}^T \cdot \mathbf{w} = 1 \\
 & y_i \cdot (\mathbf{w} \cdot \mathbf{x}_i + b) \geq \rho(1 - \xi_i), \quad i = 1, \ldots, n \\
 & \sum_{i=1}^{n} \xi_i \leq C, \quad \xi_i \geq 0
\end{array}
$$

- If $\xi_i > 0$ then the *i*-th vector is on the wrong side of the margin.
- If $\xi_i > 1$ then it is on the wrong side of the hyperplane.
- C is the budget of violation. For $C > 0$ no more than C vectors can be on the wrong side of the hyperplane.
- It is a tuning parameter to be determined by cross-validation.

C-budget for Soft Margin (example)

A. Giovanidis 2020



Bias VS Variance tradeoff.

Soft Margin properties

A. Giovanidis 2020

- Only observations that either lie on the margin or that violate the margin will affect the hyperplane, and hence the classifier.
- An observation that lies strictly on the correct side of the margin does not affect the support vector classifier!
- Changing the position of that observation would not change the classifier at all, provided that its position remains on the correct side of the margin.

Soft Margin version 2

A. Giovanidis 2020

We can formulate the problem, with the slack constraint in the objective

$$\begin{aligned} \min \quad & \Phi = \tfrac{1}{2}\mathbf{w}\cdot\mathbf{w} + \mu F\left(\textstyle\sum_{i=1}^{n}\xi_i\right) \\ \text{s.t.} \quad & y_i\cdot(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1-\xi_i,\ \ i = 1,\dots,n \\ & \xi_i \geq 0 \end{aligned}$$

where $\mu > 0$ constant, and $F(z)$ is a monotonic convex function with $F(0) = 0$, e.g. $F(z) = z^k$, $k > 1$.

Using the Lagrangian, we find again that

$$\mathbf{w}_0 = \sum_{i=1}^{n}\lambda_i^0 y_i\mathbf{x}_i.$$

Find the (soft) duals

A. Giovanidis 2020

Special case: $F(z) = z^2$.

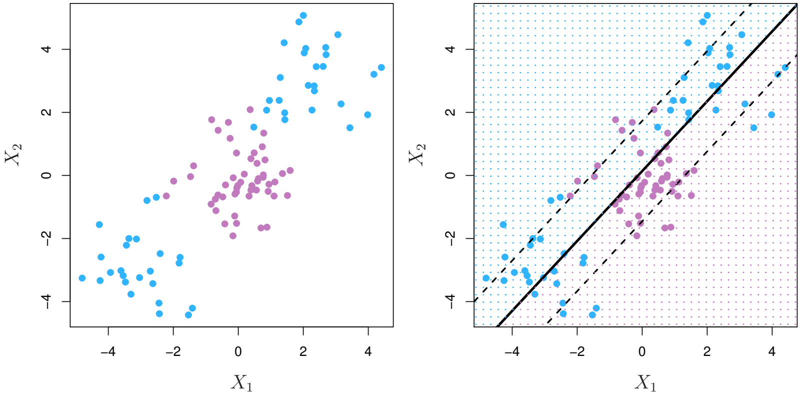
To find the optimal dual vector $\mathbf{\Lambda}_0$ solve the **quadratic problem**

$$\begin{array}{ll} \max_{\mathbf{\Lambda},\ \delta} & W(\mathbf{\Lambda}, \delta) = \mathbf{\Lambda}^T \mathbf{1} - \frac{1}{2}\left[\mathbf{\Lambda}^T \mathbf{D} \mathbf{\Lambda} + \frac{\delta^2}{\mu}\right] \\ & \\ \text{s.t.} & \mathbf{\Lambda}^T \mathbf{Y} = 0 \\ & \delta \geq 0 \\ & \mathbf{0} \leq \mathbf{\Lambda} \leq \delta \mathbf{1} \end{array}$$

$\mathbf{Y}^T$ is the vector of labels, $\mathbf{D}$ is the same symmetric $n \times n$ matrix as above, and δ is a slack variable.

When linear classifiers are not suitable

A. Giovanidis 2020



Non-linear features

A. Giovanidis 2020

☞ How can we introduce non-linearities in the feature space? (Remember we had p-features, i.e. $\mathbf{x}_i \in \mathbb{R}^p$)

- Consider new features: $z_1 = x_1^2, \ldots, z_p = x_p^2$
- Then each constraint i would be re-written as

$$y_i \cdot (\mathbf{w} \cdot \mathbf{x}_i + \mathbf{w}' \cdot \mathbf{z}_i + b) \geq 1 - \xi_i$$

The approach is problematic

- Now we need to estimate both $\mathbf{w}$ and $\mathbf{w}'$, i.e. for every new set of features, new vectors of coefficients.
- How many new coefficients will be introduced to include all pair products: $z'_{\ell,j} = x_\ell x_j$?

Separation in feature space

A. Giovanidis 2020

The idea is to construct hyperplanes in the feature space

$$\phi : \ \mathbb{R}^p \to \mathbb{R}^N$$

so that $\phi(\mathbf{x}_i) = (\phi_1(\mathbf{x}_i), \ldots, \phi_N(\mathbf{x}_i))$, $i = 1, \ldots, n$.

The new classification decision is

$$I_\phi(\mathbf{x}) \quad = \quad sign(f_\phi(\mathbf{x})) = sign\left(\mathbf{w} \cdot \phi(\mathbf{x}) + b\right).$$

We have seen from the linear case $\phi(\mathbf{x}) = \mathbf{x}$ that the optimal coefficients are written as linear combination of input vectors

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

Kernels

A. Giovanidis 2020

Replacing in the classification function we get

$$f_\phi(\mathbf{x}) \quad = \quad \sum_{i=1}^{n} \lambda_i y_i \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}) + b$$

Consider the general form of the dot-product:

$$\phi(\mathbf{u}) \cdot \phi(\mathbf{v}) = K(\mathbf{u}, \mathbf{v}).$$

where $K(\mathbf{u}, \mathbf{v})$ is a symmetric function.

Examples of potential functions

A. Giovanidis 2020

We can use a large number of potential functions:

- Linear machines

$$K(\mathbf{u},\mathbf{v}) = 1 + \mathbf{u}\cdot\mathbf{v}$$

- Polynomial machines

$$K(\mathbf{u},\mathbf{v}) = (1 + \mathbf{u}\cdot\mathbf{v})^d$$

- Radial basis function machines

$$K(\mathbf{u},\mathbf{v}) = \exp\left(-\frac{|\mathbf{u}-\mathbf{v}|^2}{\sigma^2}\right)$$

Classification function

A. Giovanidis 2020

The decision surface of these machines has the form

$$f_K(\mathbf{x}) = \sum_{i=1}^{n} \lambda_i y_i K(\mathbf{x}_i, \mathbf{x})$$

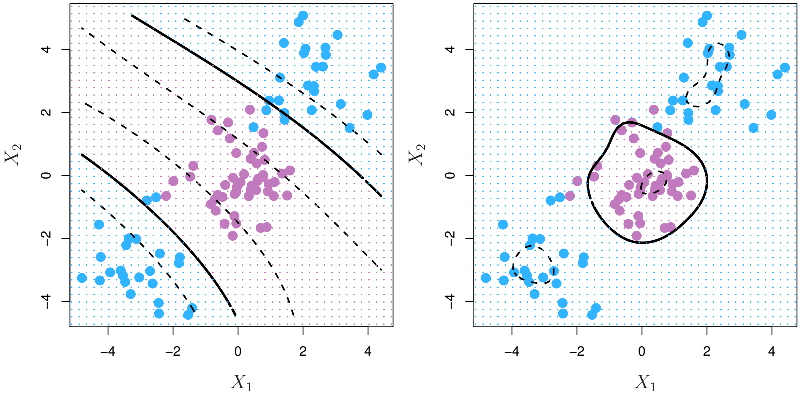
To find the optimal weights λ_i and the support vectors, one needs to solve again the dual quadratic program, as in the soft margin classifier.

The difference is that here

$$D_{ij} = y_i y_j K(\mathbf{x}_i \mathbf{x}_j), \quad i, j = 1, \ldots, n.$$

Non-linear Boundary

A. Giovanidis 2020



Loss+Penalty

A. Giovanidis 2020

In the most general case, we can write the SVM in standard Loss + Penalty form

$$\min_{\mathbf{w}} \qquad L\left(\mathbf{X}, \mathbf{y}, \mathbf{w}\right) + \mu P(\mathbf{w})$$

For the SVM these are:

- Hinge Loss

$$L\left(\mathbf{X}, \mathbf{y}, \mathbf{w}\right) \quad = \quad \max\left\{0, \sum_{i=1}^{n} 1 - y_i(\mathbf{w}\mathbf{x}_i + b)\right\}$$

- Ridge penalty

$$P(\mathbf{w}) \quad = \quad ||\mathbf{w}||_2^2$$

More than 2 Classes

A. Giovanidis 2020

- One-versus-One
  Consider the $\frac{K(K-1)}{2}$ pairs of classes, and build an SVM classifier per pair. Given a test vector, do all pair classifications, and give the vector to the majority class.
- One-versus-All
  Consider K classifiers. Each compares one class with all the rest $K-1$, and suppose the resulting coefficient vector is $\mathbf{w}_k$. We assign a test vector $\mathbf{x}^*$ to the class with maximum $\mathbf{w}_k\mathbf{x}^* + b_k$. This corresponds to the class with maximum confidence.

Probabilistic Outputs

A. Giovanidis 2020

Although the function $f_K(\mathbf{x}) = \sum_{i=1}^{n} \lambda_i y_i K(\mathbf{x}_i, \mathbf{x})$ is used to evaluate the class of a test vector $\mathbf{x}$ by its sign, it is not a probability!

It is useful in classification to have probabilistic outputs in order to evaluate certainty

$$Pr(class \mid input) = Pr(y = 1 \mid \mathbf{x}) = p(\mathbf{x}).$$

In the logistic regression, we used the logistic function

$$p(\mathbf{x}) = \frac{1}{1 + \exp(-f(\mathbf{x}))}.$$

John C. Platt (1999) proposed to fit a sigmoid after the SVM!

Fit a sigmoid after the SVM

A. Giovanidis 2020

☞ Use a parametric model to fit the parameters $Pr(y = 1 \mid f)$ directly

$$p(\mathbf{x}) \quad = \quad \frac{1}{1 + \exp(Af(\mathbf{x}) + B)},$$

with parameters A, B to train.

Use maximum likelihood estimation from the new training set $(f(\mathbf{x}_i), y_i)$.

First transform the output data as

$$t_i \quad = \quad \frac{y_i + 1}{2}$$

Then minimise the negative log-likelihood of the training data, which is a cross-entropy error function

$$\min \qquad -\sum_{i=1}^{n} t_i \log(p_i) + (1 - t_i) \log(1 - p_i).$$

The minimization is a two parameter estimation.

SV Regression

A. Giovanidis 2020

Until now we saw only classification problems.
☞ The concept can be extended to regression.

Training data: $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\} \subset \mathbb{R}^p \times \mathbb{R}$. Here, y_i is real.

Linear function: $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b$.

To get a function f as "flat" as possible, we solve the problem

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \frac{1}{2}||\mathbf{w}||_2^2 \\ \text{s.t.} \quad & y_i - \mathbf{w} \cdot \mathbf{x}_i - b \leq \epsilon, \quad i = 1, \ldots, n \\ & \mathbf{w} \cdot \mathbf{x}_i + b - y_i \leq \epsilon, \quad i = 1, \ldots, n. \end{aligned}$$

In the above ϵ is the precision to approximate all pairs $(\mathbf{x}_i, y_i)$ by the function f.

SV Regression (soft)

A. Giovanidis 2020

✍ In the case that such an ϵ-precision function is not feasible, we can again introduce slack variables ξ_i, ξ_i^*.

Linear function with soft margins:

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \frac{1}{2}||\mathbf{w}||_2^2 + \mu \sum_{i=1}^{\ell}(\xi_i + \xi_i^*) \\ \text{s.t.} \quad & y_i - \mathbf{w}\cdot\mathbf{x}_i - b \leq \epsilon + \xi_i, \\ & \mathbf{w}\cdot\mathbf{x}_i + b - y_i \leq \epsilon + \xi_i^*, \\ & \xi_i,\ \xi_i^* \geq 0. \end{aligned}$$

This corresponds to dealing with an $\epsilon-$insensitive loss function

$$|\xi|_\epsilon = \begin{cases} 0 & \text{if } |\xi| \leq \epsilon \\ |\xi| - \epsilon & \text{otherwise.} \end{cases}$$

SVR soft margin

A. Giovanidis 2020

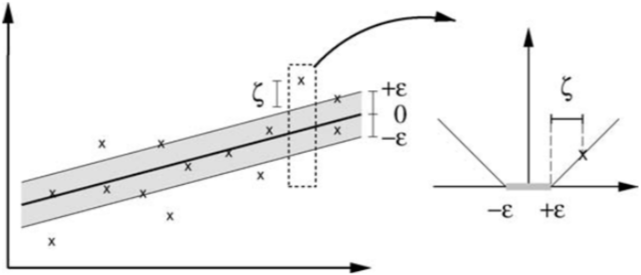


Fig. 1. *The soft margin loss setting for a linear SVM (from Schölkopf and Smola, 2002)*

SVR solution and support vectors

A. Giovanidis 2020

☞ The solution can be found again by Lagrange relaxation, and is:

$$\mathbf{w} = \sum_{i=1}^{n}(a_i - a_i^*)\mathbf{x}_i$$

and consequently

$$f(\mathbf{x}) = \sum_{i=1}^{n}(a_i - a_i^*)\mathbf{x}_i \cdot \mathbf{x} + b$$

In the above $a_i,\ a_i^*$ are the dual variables of the upper and lower constraint inequalities. The $a_i,\ a_i^*$ can never be simultaneously non-zero. The vectors with non-zero coefficients are called support vectors.

A. Giovanidis 2020

SVR solution and support vectors

Mapping again the input to feature space $\phi : \mathbb{R}^p \to \mathbb{R}^N$

$$\mathbf{w} = \sum_{i=1}^{n} (a_i - a_i^*) \phi(\mathbf{x}_i)$$

and consequently

$$f(\mathbf{x}) = \sum_{i=1}^{n} (a_i - a_i^*) K(\mathbf{x}_i, \mathbf{x}) + b$$

The used kernels are similar to the classification (Linear, Polynomial, Radial machines). We can include here the case of

- Hyperbolic Tangent machine

$$K(\mathbf{u}, \mathbf{v}) = tanh(\theta + \kappa \cdot \mathbf{u} \cdot \mathbf{v}).$$

Although this kernel does not satisfy **Mercer's condition** it is successful in practice.

A. Giovanidis 2020

END