Kernel-Based Learning & Multivariate Modeling

MIRI Master - DMKM Master

Lluís A. Belanche
belanche@cs.upc.edu

Soft Computing Research Group

Universitat Politècnica de Catalunya

2016-2017

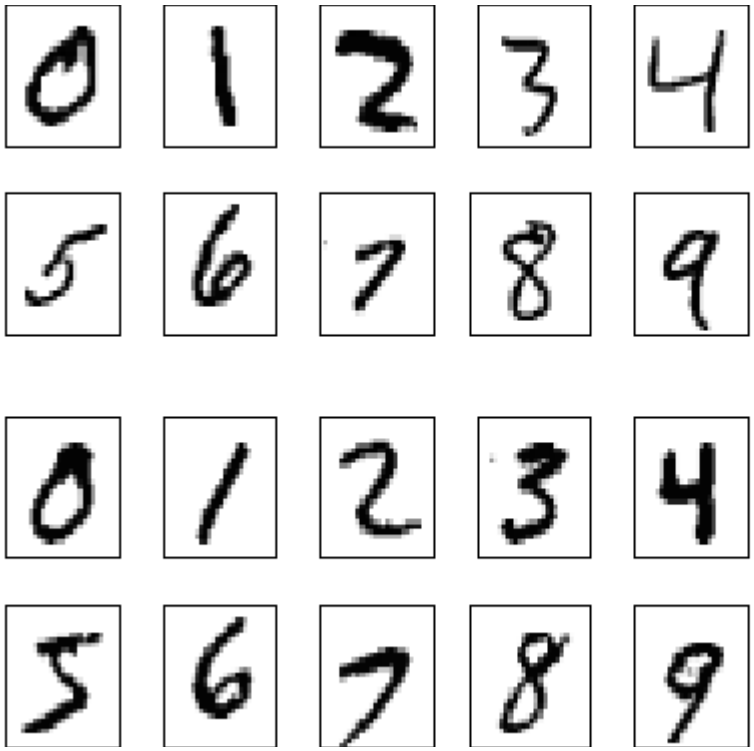
Kernel-Based Learning & Multivariate Modeling

Contents by lecture

Sep 14 Introduction to Kernel-Based Learning
Sep 21 The SVM for classification, regression & novelty detection (I)
Sep 28 The SVM for classification, regression & novelty detection (II
Oct 05 Kernel design (I): theoretical issues
Oct 19 Kernel design (II): practical issues
Oct 26 Kernelizing ML & stats algorithms
Nov 02 Advanced topics

Support Vector Machines

Application (digit recognition)



- Handwritten zip code recognition traces back to the 1960's

Support Vector Machines

Application (digit recognition)

- MNIST handwritten zip code recognition

- 60.000 training, 10.000 test examples (28×28 pixels)

Classifier	test error
Linear classifier	8.4 %
3-nearest-neighbor	2.4 %
SVM	1.4 %
Tangent distance	1.1 %
LeNet4	1.1 %
Boosted LeNet4	0.7 %
Translation invariant SVM	0.56 %

Support Vector Machines

The role of the C parameter

Increasing C ...

- penalizes margin errors more $\Rightarrow$ narrower margin (so worse theoretical generalization)
- allows the $\alpha_n \leq C$ to be larger (so more opportunities for outliers)
- increases training times (same reason)

Support Vector Machines

The role of the C parameter

We do not want C to be too large and specially too small:

1. very long training times are an indication of a too large C

2. no non-bound SVs are an indication of a too small C

- Large values of C (say, 100) should be used only when there is small noise in the data

- Small values of C (say, 0.5) should be used only when the learned function can be rather flat

Support Vector Machines

The role of the C parameter

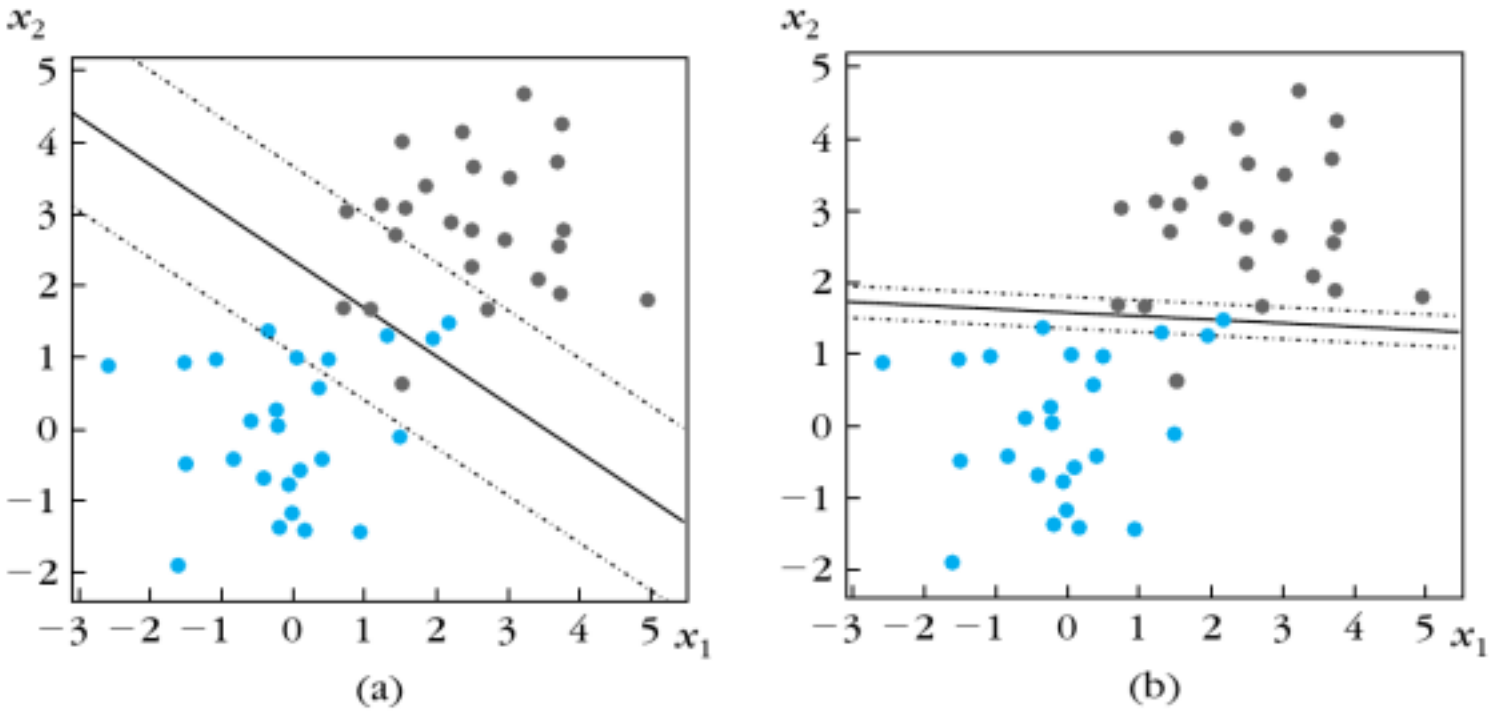


FIGURE 3.13

An example of two nonseparable classes and the resulting SVM linear classifier (full line) with the associated margin (dotted lines) for the values (a) $C = 0.2$ and (b) $C = 1000$. In the latter case, the location and direction of the classifier as well as the width of the margin have changed in order to include a smaller number of points inside the margin.

–from *Pattern Recognition (Fourth Edition)*, S. Theodoridis and K. Koutroumbas

Support Vector Machines

ν-SVMs

There are two commonly used versions of the SVM for classification:

'C-SVC': original SVM formulation, uses C parameter $C \in (0, \infty)$ to apply a penalty to the optimization for data points not correctly separated by the OSH

'nu-SVC': C is replaced by replaced by $\nu \in (0, 1)$:

- upper bound on the fraction of examples which are training errors (missclassified)
- lower bound on the fraction of points which are SVs.

Support Vector Machines

SVMs for regression

"The Support Vector method can also be applied to the case of regression, maintaining all the main features that characterise the maximal margin algorithm: a non-linear function is learned by a linear learning machine in a kernel-induced feature space while the capacity of the system is controlled by a parameter that does not depend on the dimensionality of the space."

–from N. Cristianini and J. Shawe-Taylor, *An introduction to Support Vector Machines* (2000)

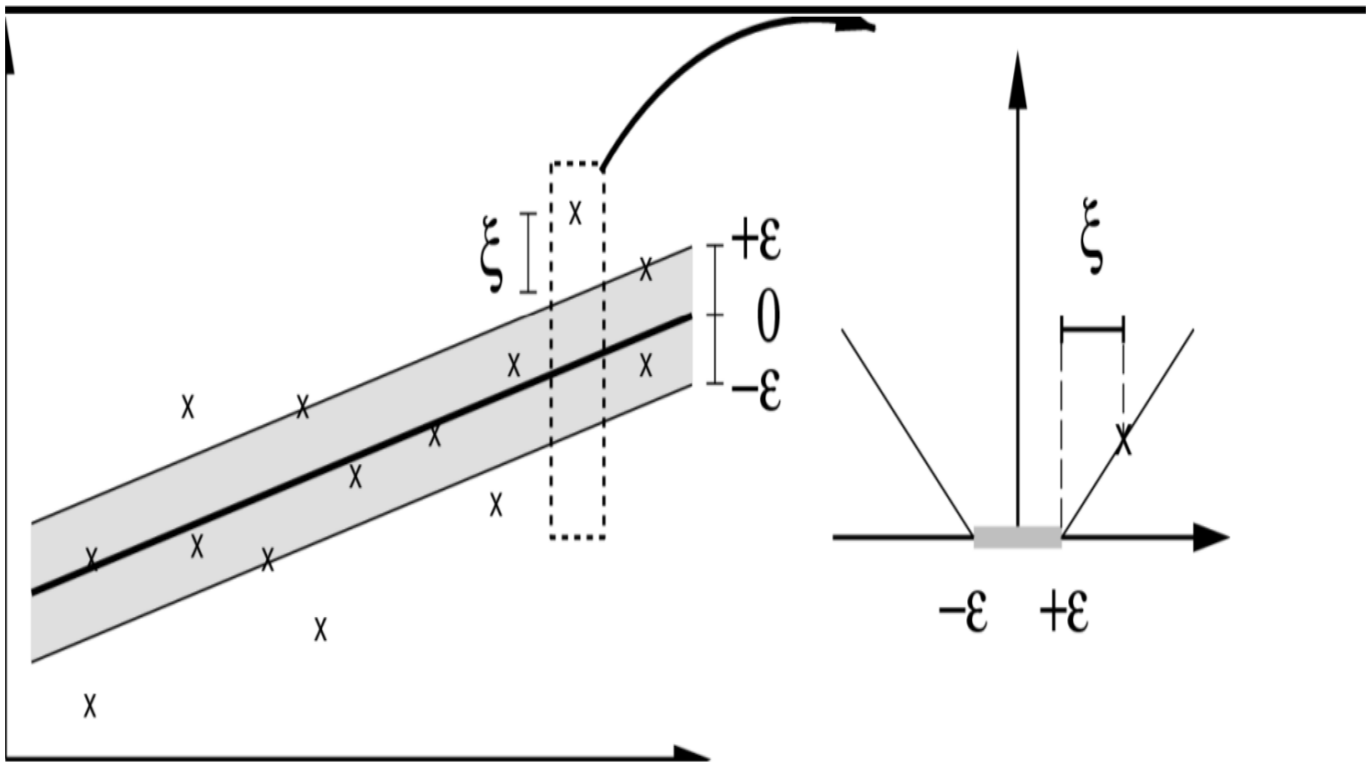
Support Vector Machines

SVMs for regression

Here we choose the ε-**insensitive** loss:

$$L(t_n, \langle \boldsymbol{w}, \boldsymbol{x}_n \rangle) = |t_n - g(\boldsymbol{x}_n)|_\varepsilon = \max(|t_n - g(\boldsymbol{x}_n)| - \varepsilon, 0)$$

where $g(\boldsymbol{x}) = \langle \boldsymbol{w}, \boldsymbol{x} \rangle + b$



Support Vector Machines

SVMs for regression

minimize $$\frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{n=1}^{N}(\xi_n + \xi_n^*)$$

subject to $$\begin{aligned} &\langle \boldsymbol{w}, \boldsymbol{x}_n\rangle + b - t_n \leq \varepsilon + \xi_n, \\ &t_n - \langle \boldsymbol{w}, \boldsymbol{x}_n\rangle + b \leq \varepsilon + \xi_n^*, \\ &\xi_n, \xi_n^* \geq 0 \end{aligned}$$

Support Vector Machines

SVMs for regression

The primal optimization problem can be transformed into the dual problem and its solution is given by:

$$y_{\text{SVM}}(\boldsymbol{x}) = \sum_{n=1}^{N} (\alpha_n - \alpha_n^*) k(\boldsymbol{x}, \boldsymbol{x}_n)$$

with $0 \leq \alpha_n, \alpha_n^* \leq C$

For convenience, we define $\beta_n = \alpha_n - \alpha_n^*$.

Support Vector Machines

SVMs for regression

A closer look at the structure of the solution:

- Data points that end up **within** the ε-tube have inactive slacks (*i.e.*, $\xi_n = \xi_n^* = 0$) and therefore $\beta_n = 0$ (**not SVs**)

- Data points that end up **not within** the ε-tube have exactly one active slack (*i.e.*, either $\xi_n > 0$ and $\xi_n^* = 0$, or vice versa) and therefore $\beta_n \neq 0$ (**non-bound SVs**)

- Data points that end up **outside** the ε-tube have exactly one bound slack (*i.e.*, $\xi_n = C$ and $\xi_n^* = 0$, or vice versa) and therefore $\beta_n \neq 0$ (**bound SVs**)

Support Vector Machines

SVMs for regression

In comparison to ridge regression, the only difference is in the choice of loss (since both are **regularized machines** and both are amenable to **kernelisation**) and its consequences:

- Deviations lower than ε are ignored
- The loss grows linearly (and not quadratically) in the residual, making it more robust against outliers
- The solution is **sparse** (the number of **basis functions** is automatically adapted)

Support Vector Machines

C versus ε

- C determines the trade off between model complexity (flatness) and tolerance to deviations larger than ε

- ε controls the width of the ε-insensitive tube

Larger ε or C implies less SVs (while smaller ε or C implies more SVs); but larger ε gives flatter models while larger C implies more complex models

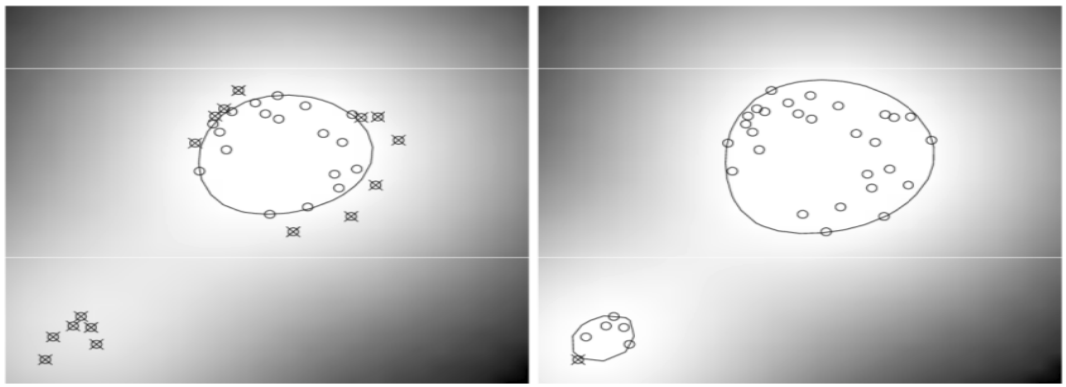
Hence, both parameters affect model complexity and number of SVs (but in a different way).

Support Vector Machines

SVMs for novelty detection (I)

- You are given a dataset drawn from a pdf $p(\boldsymbol{x})$; the $\boldsymbol{x}$ can be hand-written digits (recognizable/strange), process status (normal/faulty), credit card transactions (normal/fraudulent), ...

- The goal is to estimate a "simple" subset S of input space s.t. the probability that a test point drawn from p lies *outside* S equals some a priori specified $\nu \in (0, 1)$



–from Alex Smola: Hilbert Space Methods: Basics, Applications, Open Problems
`http://alex.smola.org/talks/rsisesvm.pdf`

Support Vector Machines

SVMs for novelty detection (II)

USPS dataset of handwritten digits: $9,298$ digit images of size 16×16

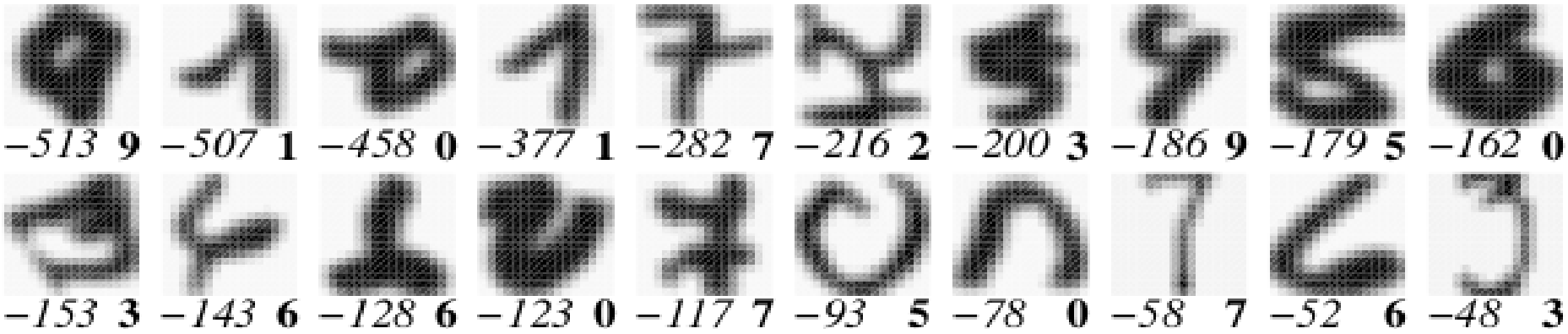


Figure 2: Outliers identified by the proposed algorithm, ranked by the negative output of the SVM (the argument of the sgn in the decision function). The outputs (for convenience in units of 10^{-5}) are written underneath each image in italics, the (alleged) class labels are given in bold face. Note that most of the examples are "difficult" in that they are either atypical or even mislabelled.

The 20 worst outliers for the USPS test set (here $\nu = 0.05$)

–from Schoelkopf et al, *Support Vector Method for Novelty Detection*, NIPS'2000

Support Vector Machines

Tricks of the trade for the kernel

1. **Standardizing** the variables is in general good (assumed numerical)

2. Kernel matrix values should not be very large or very small: if so, **normalize** the kernel

3. Kernel matrices close to the identity or close the the "all-ones" matrix are also an indication of bad kernel parameter