Aprenentatge Automàtic 2

GCED

Lluís A. Belanche

belanche@cs.upc.edu

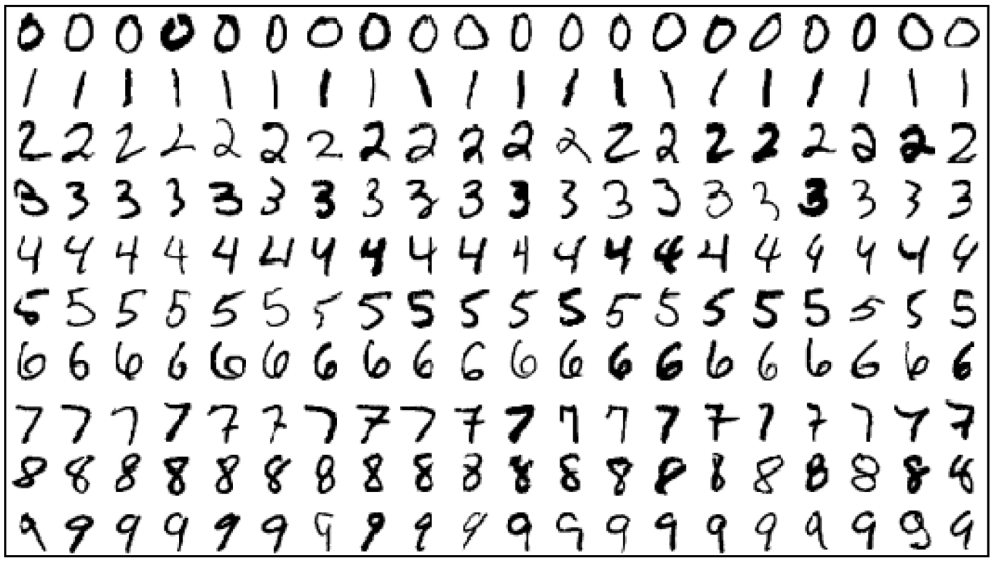
Soft Computing Research Group
Dept. de Ciències de la Computació (Computer Science)
Universitat Politècnica de Catalunya

2021-2022

LECTURE 4: Other SVMs: for regression, for novelty detection, and multiclass extensions

Support Vector Machines

Application (digit recognition)



- MNIST handwritten zip code recognition
- 60.000 training, 10.000 test examples (28×28 pixels)
- Handwritten zip code recognition traces back to the 1960's

Support Vector Machines

Application (digit recognition)

Method	Correct (%)	Error (%)	Reject (%)
LeNet-4 [1]	98.90	1.10	0
LeNet-5 [1]	99.05	0.95	0
*Boosted LeNet-4 [1]	99.30	0.70	0
SVC-poly [43]	98.60	1.40	0
Virtual SV [43]	99.00	1.00	0
Pairwise SVC [44]	98.48	1.52	0
Dong et al. [45]	99.01	0.99	0
Mayraz et al. [48]	98.30	1.70	0
Belongie et al. [47]	99.37	0.63	0
Teow et al. [46]	**99.41**	0.59	0

*Multiple classifiers.

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

Increasing the value of C ...

- penalizes margin errors more $\Rightarrow$ narrower margin $\Rightarrow$ larger VC-dimension

- allows the $\alpha_i \leq C$ to be larger (so more opportunities for outliers)

Support Vector Machines

The role of the C parameter

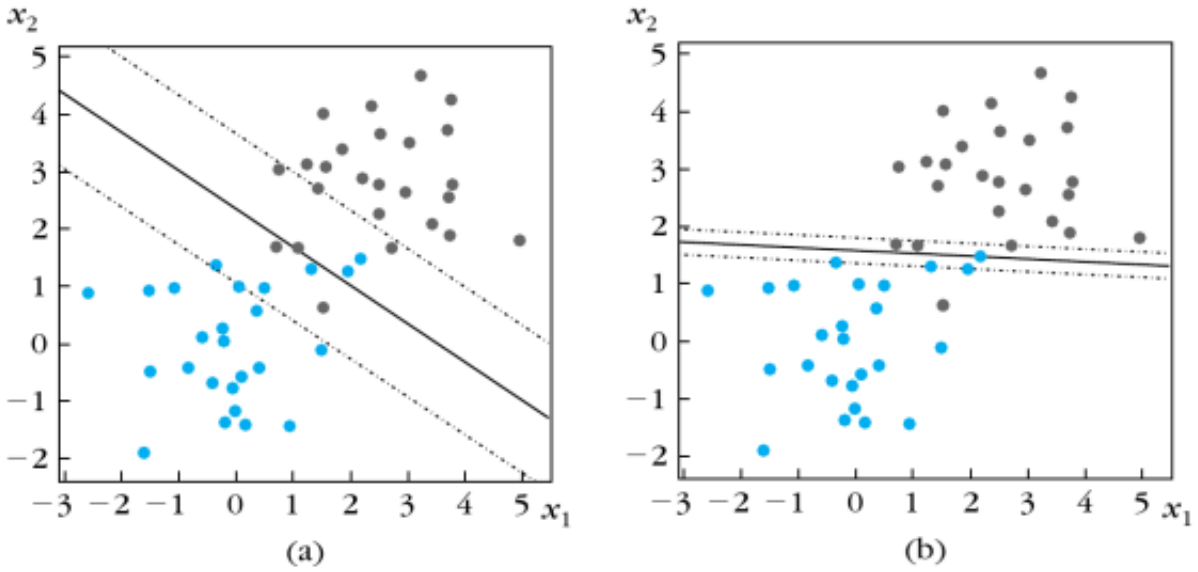


FIGURE 3.13

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 a parameter $C \in (0, \infty)$ to apply a penalty to the optimization for those data points not entirely separated by the OSH (violating the margins)

'nu-SVC': C is 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)

Exercise: list and acknowledge the common features of SVMs

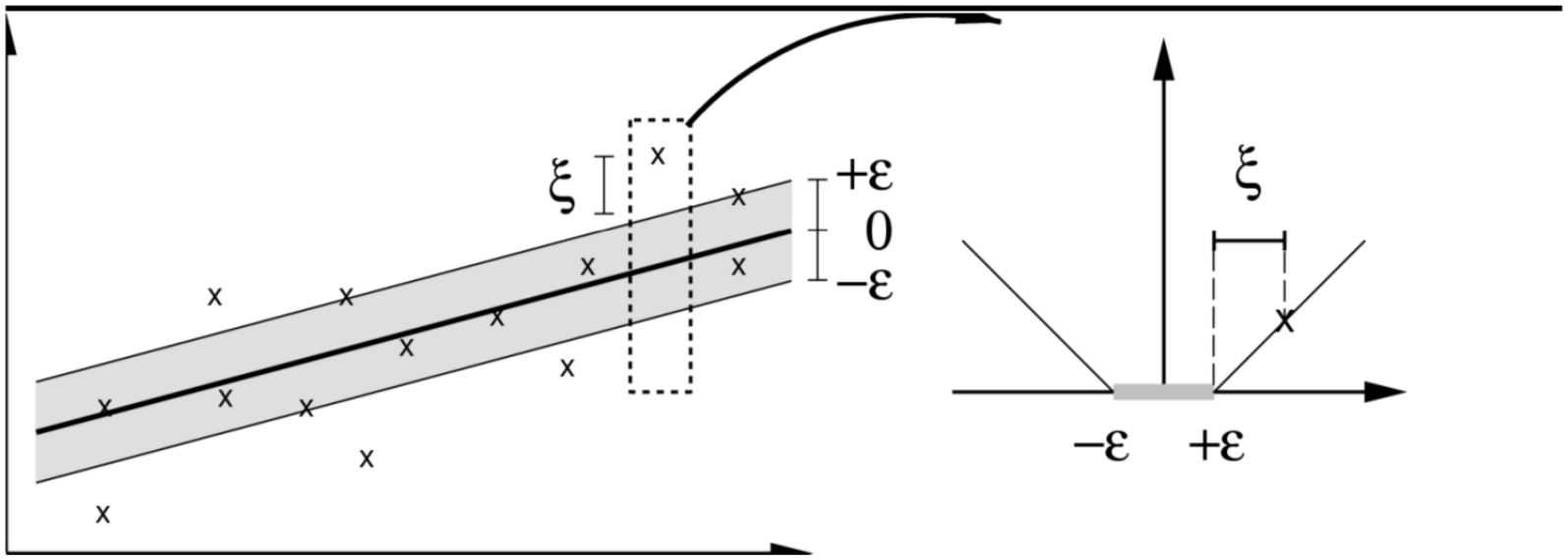
Support Vector Machines

SVMs for regression

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

$$L(y_i, \boldsymbol{w}^\top \boldsymbol{x}_i) = |y_i - g(\boldsymbol{x}_i)|_\varepsilon := \text{máx}(|y_i - g(\boldsymbol{x}_i)| - \varepsilon, 0)$$

where $g(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + b$



Support Vector Machines

SVMs for regression

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

subject to $$\begin{aligned} \boldsymbol{w}^\top \boldsymbol{x}_i + b - y_i &\leq \varepsilon + \xi_i, \\ y_i - \boldsymbol{w}^\top \boldsymbol{x}_i + b &\leq \varepsilon + \xi_i^*, \\ \xi_i, \xi_i^* &\geq 0 \end{aligned}$$

where the ξ_i, ξ_i^* are again slack variables controlling the "violations".

Support Vector Machines

SVMs for regression

Then feature maps $\phi(\cdot)$ are introduced, the primal optimization problem is transformed into the dual, and kernelised, to give:

$$y_{\mathsf{SVM}}(\boldsymbol{x}) = \sum_{i=1}^{n} \beta_i k(\boldsymbol{x}, \boldsymbol{x}_i)$$

with $0 \leq \alpha_i, \alpha_i^* \leq C$.

For convenience, we have defined $\beta_i := \alpha_i - \alpha_i^*$.

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.*, $\xi_i = \xi_i^* = 0$) and therefore $\beta_i = 0$ (**not SVs**)

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

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

Support Vector Machines

SVMs for regression

In comparison to ridge regression, the only difference is in the choice of the 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** $\phi(\boldsymbol{x}_i)$ 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

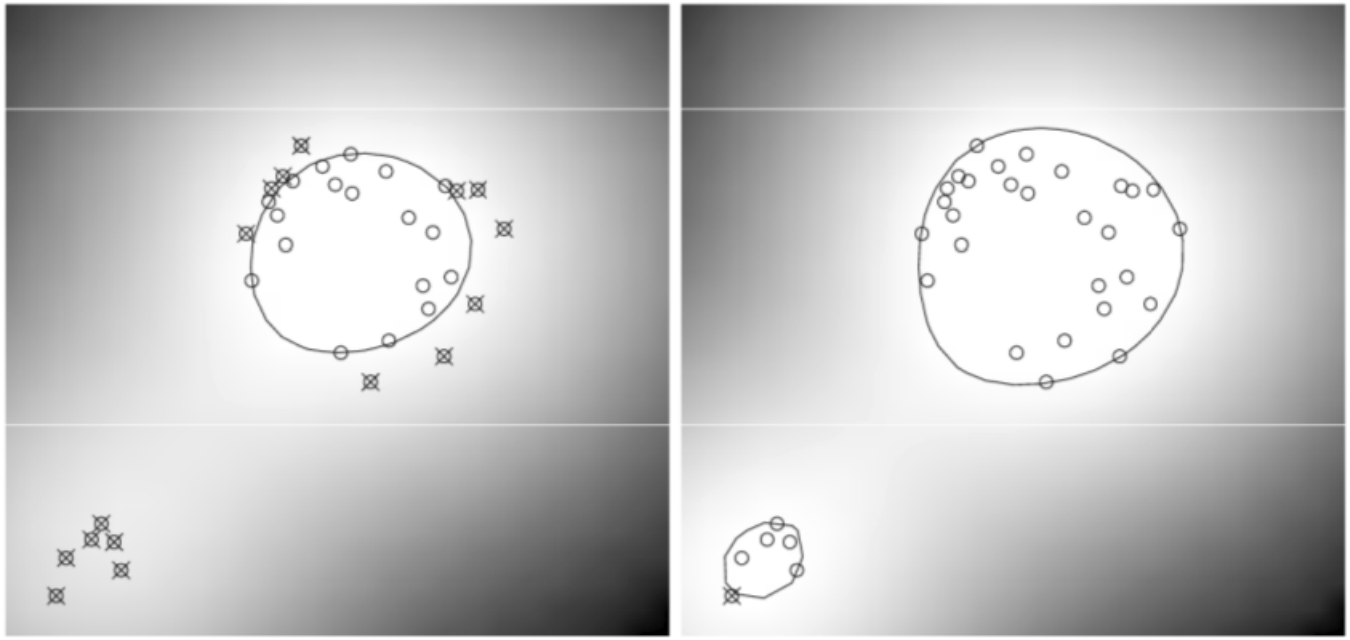
SVMs for novelty detection

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

- The SVM approach to this problem makes S a hypersurface to estimate a function which is positive on S and negative on S^c such that the *probability* that a test point drawn from p lies *outside* S equals some a priori specified $\rho \in (0, 1)$

Support Vector Machines

SVMs for novelty detection



- The SVs are points lying on the separating surface and the training errors are outliers ("novelties")

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

Support Vector Machines

SVMs for novelty detection

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

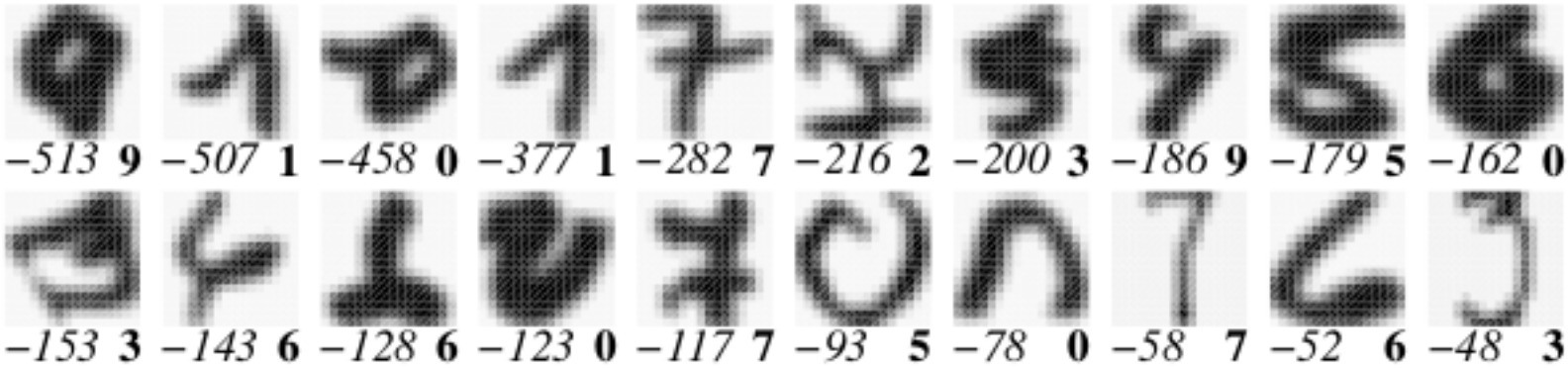


Figure 2: Outliers identified by the proposed algorithm, ranked by the negative output of the SVM (the argument of the sgn in the decision function). The outputs (for convenience in units of 10^{-5}) are written underneath each image in italics, the (alleged) class labels are given in bold face. Note 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 $\rho = 0{,}05$)

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

Support Vector Machines

SVMs for multiclass/multilabel problems

- A simple way of using SVMs to learn a K-class classification problem consists in choosing the maximum applied to the outputs of K SVMs solving a one-per-class decomposition of the general problem **[1-vs-all]**

- A more sophisticated way is to learn $K(K-1)/2$ SVMs, where the i,j SVM solves the problem of separating classes i and j **[all-vs-all]**

  If the sign of the i-vs-j classifier says that the data point is in the i-th class, then the vote for the i-th class is increased by 1. Otherwise, the vote for the j-th class is increased by 1. Finally, we conclude that the predicted class is the one with the highest vote

Support Vector Machines

SVMs for multiclass/multilabel problems

These "extension" approaches mainly fall into the two categories of **problem transformation** and **algorithm adaptation**:

problem transformation consists in using SVMs to learn a L-label classification problem consists in transforming it into a series of L binary classification problems (one for each label)

algorithm adaptation approaches, on the other hand, solve these problems directly by extending binary classification methods in such a way that all classes/labels are considered at once

-see *A Unified Framework for Multiclass and Multilabel Support Vector Machines*. 2003.11197 arxiv.org (2020)