Margin Knowledge Per Example Using Support Vector Machines

Marcin Orchel

AGH University of Science and Technology
in Poland

Support Vector Classification

- SVM hard margin classifier – classifies data with the hyperplane that has the largest distance to the closest training vectors
- SVM soft margin classifier – extended version able to classify nonseparable data

Support Vector Classification

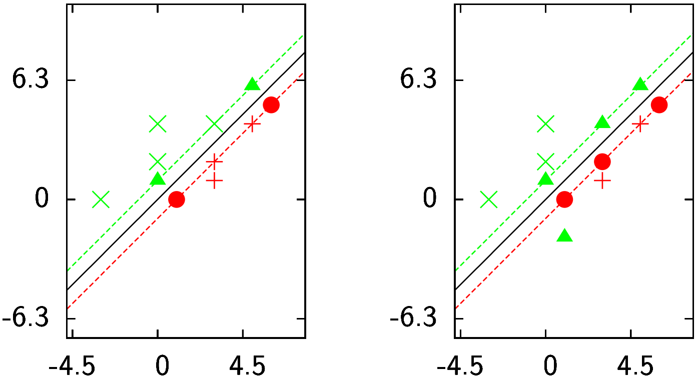

Figure: Two types of margin classifiers: hard on the left, and soft on the right. In the figures, there are example points, support vectors (triangles and circles), solutions (solid lines), margin lines (dashed lines). In the right figure, we can see a misclassified point (1, -2)

Support Vector Classification Formulation

- SVC primal problem:

$$\min f\left(\vec{w}, b, \vec{\xi}\right) = \frac{1}{2}\left\|\vec{w}\right\|^2 + \vec{C} \cdot \vec{\xi}$$

subject to $y_i h\left(\vec{x_i}\right) \geq 1 - \xi_i, \ \vec{\xi} \geq 0$ for $i \in \{1, \ldots, n\}$, where $\vec{C} \gg 0, \ h\left(\vec{x_i}\right) = \vec{w} \cdot \vec{x_i} + b$.

- SVC dual problem:

$$\max W\left(\vec{\alpha}\right) = 1 \cdot \vec{\alpha} - \frac{1}{2}\vec{\alpha}^T Q \vec{\alpha}$$

subject to

$$\vec{\alpha} \cdot \vec{y} = 0, \ 0 \leq \vec{\alpha} \leq C$$

where $Q_{ij} = y_i y_j \left(\vec{x_i} \cdot \vec{x_j}\right)$ for all $i, j \in \{1, \ldots, n\}$.

Support Vector Classification Solution

- decision boundary

$$h^*\left(\vec{x}\right) = \sum_{i=1}^{l} y_c^i \alpha_i^* K\left(\vec{x_i}, \vec{x}\right) + b_c^* = 0 \qquad (1)$$

- non support vectors have $\alpha_i^* = 0$
- they are ignored in the solution

ε-Support Vector Regression Idea

- find a function for which all examples fall between ε bounds

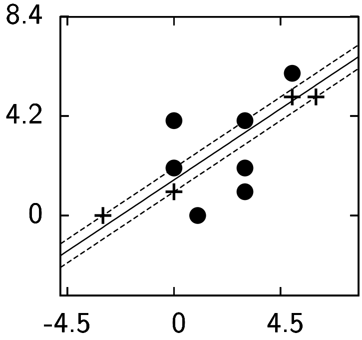

Figure: The idea of ε-SVR. In the figure, there are examples, support vectors (circles), a solution (solid line), and ε boundaries (dashed lines)

ε-Support Vector Regression Formulation

- ε-SVR primal problem:

$$\min f\left(\vec{w}, b, \vec{\xi}, \vec{\xi^*}\right) = \|\vec{w}\|^2 + C\sum_{i=1}^{l}\left(\xi_i + \xi_i^*\right)$$

subject to
$y_i - g\left(x_i\right) \le \varepsilon + \xi_i,\ g\left(x_i\right) - y_i \le \varepsilon + \xi_i^*,\ \vec{\xi} \ge 0,\ \vec{\xi^*} \ge 0$ for $i \in \{1, \ldots, n\}$,
where $C > 0$, $g\left(\vec{x_i}\right) = \vec{w} \cdot \vec{x_i} + b$.

SVC With Margin Weights Formulation

- SVC with margin weights formulation:

$$\min f\left(\vec{w}, b, \vec{\xi}\right) = \frac{1}{2}\left\|\vec{w}\right\|^2 + \vec{C} \cdot \vec{\xi}$$

  subject to $y_i h\left(\vec{x_i}\right) \geq 1 - \xi_i + \varphi_i$, $\vec{\xi} \geq 0$ for $i \in \{1, \ldots, n\}$,
  where $\vec{C} \gg 0$, $\varphi_i \in \mathbb{R}$, $h\left(\vec{x_i}\right) = \vec{w} \cdot \vec{x_i} + b$.

- points lying outside margins can be support vectors

SVC With Margin Weights

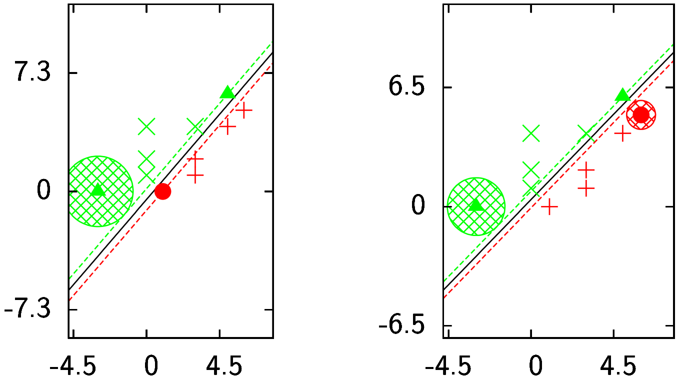

Figure: Interpretation of detractors as dynamic hyperspheres. In the figures, there are example points, solutions (solid lines), support vectors (triangles and circles), tractors (circles filled with grid pattern). In both figures, there is a detractor in (-3, 0) with $\varphi = 5.0$. A radius of a detractor differs in both cases (2.2 and 1.6 respectively)

ε-Support Vector Regression Reformulation

ε-SVR reformulation

regression points are duplicated, original points get 1 class, duplicated points get -1 class, and we solve φ-SVC with $\varphi_i = y_c^i y_r^i - \varepsilon - 1$

$h(\vec{x}) := w_c \cdot x + b_c = 0 \rightarrow g(\vec{x}) = w_c \cdot x + b_c$

Reduced Models

- Proposed reduced method:
  1. run the SVC or the ε-SVR on the original data
  2. compute values of the φ weights based on the found solution
  3. reduce the data by removing some of data vectors (including support vectors)
  4. run φ-SVC on a reduced data set with values of the φ weights
  5. for regression, transform back the solution into the regression form
- the proposed method preserves additional knowledge about the original solution in the φ weights

Properties of Reduced Models

- reduced models are models with increased sparsity,
- reduced models are faster for postprocessing (because of simpler solutions), e.g. testing new examples
- reduction ratio parameter is the trade-off between generalization performance and simplicity
- to some extent reducing data with the proposed method does not decrease generalization performance

Results

- we are able to generate reduced models with similar number of support vectors and better generalization performance than for the simple reduction
- increased generalization performance of reduced models ranging from 5% to 70% comparing to the simple reduction with similar number of support vectors

Conclusions

- with the φ-SVC we can create simpler SVM solutions – with less support vectors even without compromising generalization performance
- reduced models are faster in post-processing