Regression Based on Binary Classification Using Support Vector Machines

Marcin Orchel

Abstract

In this report, we describe the newest results regarding recently proposed by me a novel regression method, called δ support vector regression (δ-SVR). In this report, we propose a method, called δ-SVR, that replaces a regression problem with binary classification problems which are solved by support vector machines (SVM). We analyze statistical equivalence of a regression problem with a binary classification problem. We show potential possibility to improve generalization error bounds based on Vapnik-Chervonenkis (VC) dimension, compared to SVM. We conducted experiments comparing δ-SVR with ε-insensitive support vector regression (ε-SVR) on synthetic and real world data sets. The results indicate that δ-SVR achieves comparable generalization error, fewer support vectors, and smaller generalization error over different values of ε and δ. The δ-SVR method is faster for linear kernels while using sequential minimal optimization (SMO) solver, for nonlinear kernels speed results depend on the data set.

Contents

Chapter 1

Introduction

1.1 Support Vector Classification Basics

For a classification problem, we consider a set of n training vectors $\vec{x_i}$ for $i \in \{1, \ldots, n\}$, where $\vec{x_i} = \left(x_i^1, \ldots, x_i^m\right)$. The i-th training vector is mapped to $y_{\mathrm{c}}^i \in \{-1, 1\}$. The m is a dimension of the problem. The support vector classification (SVC) soft margin case optimization problem with $\|\cdot\|_1$ norm is

OP 1.

$$\min_{\vec{w_{\mathrm{c}}}, b_{\mathrm{c}}, \vec{\xi_{\mathrm{c}}}} \quad f\left(\vec{w_{\mathrm{c}}}, b_{\mathrm{c}}, \vec{\xi_{\mathrm{c}}}\right) = \frac{1}{2}\left\|\vec{w_{\mathrm{c}}}\right\|^2 + C_{\mathrm{c}} \sum_{i=1}^{n} \xi_{\mathrm{c}}^i \tag{1.1}$$

subject to

$$y_{\mathrm{c}}^i h\left(\vec{x_i}\right) \geq 1 - \xi_{\mathrm{c}}^i \ , \tag{1.2}$$

$$\vec{\xi_{\mathrm{c}}} \geq 0 \tag{1.3}$$

for $i \in \{1, \ldots, n\}$, where

$$h\left(\vec{x_i}\right) = \vec{w_{\mathrm{c}}} \cdot \vec{x_i} + b_{\mathrm{c}} \ , \tag{1.4}$$

$$C_c > 0 \ . \tag{1.5}$$

The $h^*\left(\vec{x}\right) = \vec{w_{\mathrm{c}}^*} \cdot \vec{x} + b_{\mathrm{c}}^* = 0$ is a decision curve of the classification problem. Some of training points can be incorrectly classified Fig. 1.1b.

The $\vec{w_c^*} \cdot \vec{x}$ can be computed as

$$\vec{w_c^*} \cdot \vec{x} = \sum_{i=1}^{n} y_{\mathrm{c}}^i \alpha_i^* K\left(\vec{x_i}, \vec{x}\right) \ . \tag{1.6}$$

Therefore, the decision curve is

$$h^*\left(\vec{x}\right) = \sum_{i=1}^{n} y_{\mathrm{c}}^i \alpha_i^* K\left(\vec{x_i}, \vec{x}\right) + b_{\mathrm{c}}^* = 0 \ , \tag{1.7}$$

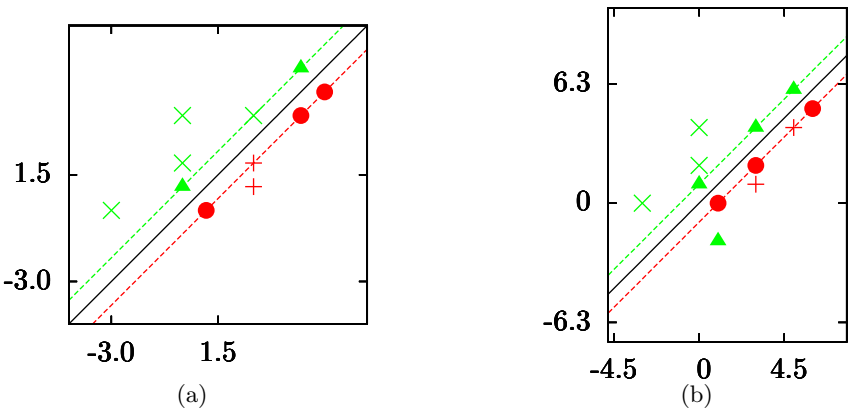


Figure 1.1: Two types of margin classifiers: hard and soft. Example points, support vectors (triangles and circles), solutions (solid lines), margin lines (dashed lines). (a) Hard. (b) Soft. A misclassified point is in (1, -2)

where α_i are Lagrange multipliers of the dual problem, $K(\cdot,\cdot)$ is a kernel function, which appears only in the dual problem. *Margin boundaries* are defined as the two hyperplanes $h(\vec{x}) = -1$ and $h(\vec{x}) = 1$. *Optimal margin boundaries* are defined as the two hyperplanes $h^*(\vec{x}) = -1$ and $h^*(\vec{x}) = 1$. *Geometric margin of the hyperplane* h is defined as $1/\|\vec{w_c}\|$. The i-th training example is *a support vector*, when $\alpha_i^* \neq 0$. A set of support vectors contains all training examples lying below optimal margin boundaries ($y_c^i h^*(\vec{x_i}) < 1$), and part of the examples lying exactly on the optimal margin boundaries ($y_c^i h^*(\vec{x_i}) = 1$), Fig. 1.1b.

1.2 Support Vector Regression Basics

In a regression problem, we consider a set of training vectors $\vec{x_i}$ for $i \in \{1,\ldots,n\}$, where $\vec{x_i} = \left(x_i^1,\ldots,x_i^m\right)$. The i-th training vector is mapped to $y_r^i \in \mathbb{R}$. The m is a dimension of the problem. The ε-SVR soft case optimization problem is

OP 2.

$$\min_{\vec{w_r},b_r,\vec{\xi_r},\vec{\xi_r^*}} \quad f\left(\vec{w_r},b_r,\vec{\xi_r},\vec{\xi_r^*}\right) = \frac{1}{2}\|\vec{w_r}\|^2 + C_r\sum_{i=1}^{n}\left(\xi_r^i + \xi_r^{*i}\right) \tag{1.8}$$

subject to

$$y_r^i - g(\vec{x_i}) \leq \varepsilon + \xi_r^i \ , \tag{1.9}$$

$$g(\vec{x_i}) - y_r^i \leq \varepsilon + \xi_r^{i*} \ , \tag{1.10}$$

$$\vec{\xi_r} \geq 0 \ , \tag{1.11}$$

$$\vec{\xi_r^*} \geq 0 \tag{1.12}$$

for $i \in \{1,\ldots,n\}$, where

$$g(\vec{x_i}) = \vec{w_r}\cdot\vec{x_i} + b_r \ , \tag{1.13}$$

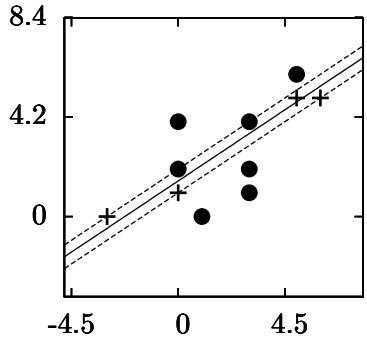


Figure 1.2: The idea of ε-SVR. Points - examples, circles - support vectors, a solid line - a solution, dashed lines - ε boundaries

$$\varepsilon \in \mathbb{R} \ . \tag{1.14}$$

The $g^* (\vec{x}) = \vec{w_{\mathrm{r}}^*} \cdot \vec{x} + b_{\mathrm{r}}^*$ is a regression function. Optimization problem 2 is transformed into an equivalent dual problem. The regression function becomes

$$g^* (\vec{x}) = \sum_{i=1}^{n} (\alpha_i^* - \beta_i^*) K (\vec{x_i}, \vec{x}) + b_{\mathrm{r}}^* \ , \tag{1.15}$$

where α_i, β_i are Lagrange multipliers, $K (\cdot, \cdot)$ is a kernel function. The ε boundaries are defined as $g (\vec{x}) - \varepsilon$ and $g (\vec{x}) + \varepsilon$. The i-th training example is *a support vector*, when $\alpha_i^* - \beta_i^* \neq 0$. For $\varepsilon \geq 0$, a set of support vectors contains all training examples lying outside ε boundaries, and part of the examples lying exactly on ε boundaries, Fig. 1.2. The number of support vectors can be controlled by ε parameter.

Chapter 2

Regression Based on Binary Classification

Recently, an alternative regression method was proposed by me in [11, 13], which is called δ-SVR. The idea of the new method is to duplicate and shift data in order to use SVC to solve regression problems. The δ-SVR has the same advantages as ε-SVR: one of the steps is to solve a convex optimization problem, it generates sparse solutions, kernel functions can be used for generating nonlinear solutions. The δ-SVR achieves similar or better for some settings generalization performance compared with ε-SVR and improves the number of support vectors, [11, 13]. Moreover, some types of prior knowledge already incorporated to SVC can be directly used for regression problems, [11]. The δ-SVR has a potential to use a much broader type of modifications and improvements of SVC directly for regression problems without the need of reformulating them for specific regression methods. In [13], we analyzed a general problem of transforming regression into classification and also generalization performance, structural risk minimization (SRM) and sparsity for δ-SVR.

Vapnik [15, 16] proposed generalization of capacity concepts introduced for classification to regression problems by describing regression functions as a complete set of indicators, see Appendix A.1. Based on this idea a method for solving regression problems as multiclass classification problems was proposed in [4, 5, 2]. The method uses discretization process to generate multiclass labels. Some attempts also were made to combine support vector regression (SVR) with SVC, [17]. In δ-SVR, we increase input dimension by 1 and create binary labels for duplicated and shifted points up and down, so we solve only one binary classification problem. The concept of duplicating and shifting data was first published in [7], it was investigated independently by me and submitted to [10] and published in [11]. The main problem of the realization of the concept in [7] is that an additional optimization problem must be solved every time a new example is tested in order to find a solution of the implicit equation; the authors used a golden section method. Moreover, two problems arise with this method. The solution might not exist or there could be more than one solution. In [11], we proposed a special type of kernels for which a unique solution is guaranteed and it is easily achievable by an explicit

formula without the need of solving an additional optimization problem. Furthermore, in [11], we proposed using the method to incorporate knowledge about the margin of an example, which was previously incorporated to SVC for classification problems in [9, 12], directly for regression problems. We noticed that the method has a potential to use a much broader type of extensions of SVC directly for regression problems, without the need of incorporating them additionally for specific regression methods, which is a practice nowadays. Lin and Guo [7] proposed an improvement to the method, for further increasing a sparseness of the solution by decreasing a value of a shifting parameter for examples with low and high values of the output, although it requires tuning an additional parameter during a training phase.

The goals of the research presented by me in [13] were to analyze a general concept of representing regression problems as classification ones by duplicating and shifting data, to analyze potential generalization improvements of δ-SVR over SVM and to extend experiments conducted in [11]. The outline is as follows. In the first section, we give an introduction to δ-SVR. In the second section, we give a theoretical analysis of the transformation. In the third section, we analyze generalization ability of δ-SVR. In the fourth section, we present experiments on synthetic and real world data sets.

2.1 Introduction to δ-SVR

We consider a set of training vectors $\vec{x_i}$ for $i \in \{1, \ldots, n\}$, where $\vec{x_i} = \left(x_i^1, \ldots, x_i^m\right)$. The i-th training vector is mapped to $y_{\mathrm{r}}^i \in \mathbb{R}$. The δ-SVR method finds a regression function by the following procedure.

1. Every training example $\vec{x_i}$ is duplicated, an output value y_{r}^i is increased by a value of a parameter $\delta \geq 0$ for original training examples, and decreased by δ for duplicated training examples.

2. Every training example $\vec{x_i}$ is converted to a classification example by incorporating the output to the input vector as an additional feature and setting class 1 for original training examples, class -1 for duplicated training examples.

3. The SVC method is launched for a classification problem.

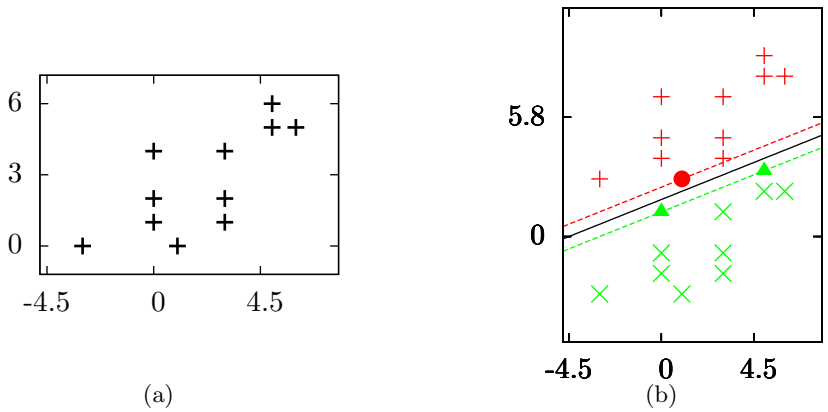
4. The solution of SVC method is converted back to function form.

The idea of the transformation is depicted in Fig. 2.1, Fig. 2.2. The result of the first step is a set of training mappings for $i \in \{1, \ldots, 2n\}$

$$\begin{cases} \vec{x_i} \to y_{\mathrm{r}}^i + \delta & \text{for } \; i \in \{1, \ldots, n\} \\ \vec{x_i} \to y_{\mathrm{r}}^i - \delta & \text{for } \; i \in \{n+1, \ldots, 2n\} \end{cases} , \tag{2.1}$$

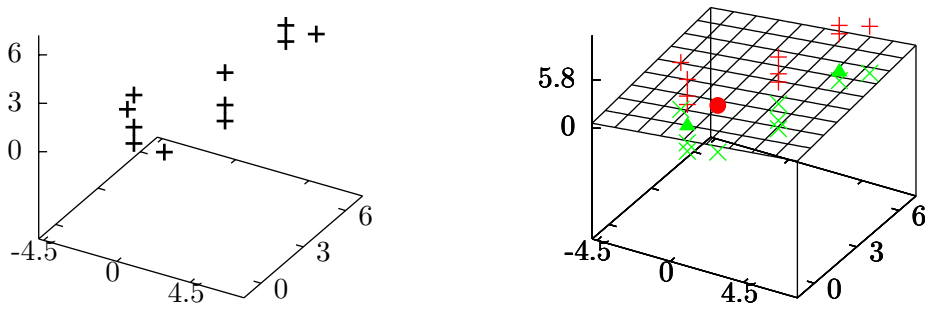
where $\vec{x_{n+i}} = \vec{x_i}$, $y_r^{n+i} = y_r^i$ for $i \in \{1, \ldots, n\}$, $\delta \in \mathbb{R}$. The δ is called *the translation parameter*. The result of the second step is a set of training mappings for $i \in \{1, \ldots, 2n\}$

$$\vec{c_i} = \left(x_i^1, \ldots, x_i^m, y_{\mathrm{r}}^i + y_c^i \delta\right) \to y_c^i \ , \tag{2.2}$$



(a) (b)

Figure 2.1: The idea of the transformation of the problem in δ-SVR for 2d. (a) Points - regression examples. (b) Points - classification examples after transformation, triangles and circles - support vectors, a solid line - a solution, dashed lines - optimal margin boundaries



(a) (b)

Figure 2.2: The idea of the transformation of the problem in δ-SVR for 3d. (a) Points - regression examples. (b) Points - classification examples after transformation, triangles and circles - support vectors, a plane - a solution

where $y_c^i = 1$ for $i \in \{1, \ldots, n\}$, and $y_c^i = -1$ for $i \in \{n+1, \ldots, 2n\}$. The dimension of the $\vec{c_i}$ vectors is equal to $m+1$. The set of $\vec{x_i}$ mappings before duplication is called *a regression data setting*, the set of $\vec{c_i}$ ones is called *a classification data setting*. In the third step, OP 1 is solved with $\vec{c_i}$ examples, so we can write OP 1 as

OP 3.

$$\min_{\vec{w_c},b_c,\vec{\xi_c}} \quad f\left(\vec{w_c},b_c,\vec{\xi_c}\right)=\frac{1}{2}\left\|\vec{w_c}\right\|^2+C_c\sum_{i=1}^{2n}\xi_c^i \tag{2.3}$$

subject to

$$y_c^i\left(w_{c,\vec{red}}\cdot\vec{x_i}+w_c^{m+1}\left(y_r^i+y_c^i\delta\right)+b_c\right)\geq 1-\xi_c^i \ , \tag{2.4}$$

$$\vec{\xi_c}\geq 0 \tag{2.5}$$

for $i\in\{1,\ldots,2n\}$.

The $w_{c,\vec{red}}$ is defined as $w_{c,\vec{red}}=(w_1,\ldots,w_m)$. The $h^*\left(\vec{x}\right)=\vec{w_c^*}\cdot\vec{x}+b_c^*=0$ is a decision curve of the classification problem. Note that $h^*\left(\vec{x}\right)$ is in the implicit form of the last coordinate of $\vec{x}$. In the fourth step, an explicit form of the last coordinate needs to be find. The explicit form is needed for example for testing new examples. The $\vec{w_c}$ variable of the primal problem for a simple linear kernel is found in the following way

$$\vec{w_c}=\sum_{i=1}^{2n}y_c^i\alpha_i\vec{c_i} \ . \tag{2.6}$$

For a simple linear kernel the explicit form of (1.7) is

$$x_{m+1}=\frac{-\sum_{j=1}^{m}w_c^jx_j-b_c}{w_c^{m+1}} \ . \tag{2.7}$$

The regression solution is $g^*\left(\vec{x}\right)=\vec{w_r}\cdot\vec{x}+b_r$, where $w_r^i=-w_c^i/w_c^{m+1}$, $b_r=-b_c/w_c^{m+1}$ for $i=1,\ldots,m$. For nonlinear kernels, a conversion to the explicit form has some limitations. First, a decision curve could have more than one value of the last coordinate for specific values of remaining coordinates of $\vec{x}$ and therefore it cannot be converted unambiguously to the function (for example a polynomial kernel with a dimension equals to 2). Second, even when the conversion to the function is possible, there is no explicit analytical formula (for example a polynomial kernel with a dimension greater than 4), hence a special method for finding the explicit formula of the coordinate should be used, for example a bisection method. The disadvantage of this solution is a longer time of testing new examples. To overcome these problems, we proposed to incorporate prior knowledge to the classification problem, that the solution will be always in the form of the function in the chosen direction. Thus, we proposed in [11] a new kernel type in which the last coordinate is placed only inside a linear term. The new kernel is constructed from an original kernel by removing the last coordinate, and adding the linear term with the last coordinate

$$K\left(\vec{x},\vec{y}\right)=K_o\left(\vec{x_{red}},\vec{y_{red}}\right)+x_{m+1}y_{m+1} \ , \tag{2.8}$$

where $\vec{x}$ and $\vec{y}$ are $m+1$ dimensional vectors, $\vec{x_{red}}=(x_1,\ldots,x_m)$, $\vec{y_{red}}=(y_1,\ldots,y_m)$, $K_o\left(\cdot,\cdot\right)$ is the original kernel from which the new one was constructed. For the most

popular kernels polynomial, radial basis function (RBF) and sigmoid, the conversions are respectively

$$(\vec{x}\cdot\vec{y})^d \quad\to\quad \left(\sum_{i=1}^{m} x_i y_i\right)^d + x_{m+1}y_{m+1} \ , \tag{2.9}$$

$$\exp -\frac{\|\vec{x}-\vec{y}\|^2}{2\sigma^2} \quad\to\quad \exp -\frac{\sum_{i=1}^{m}(x_i-y_i)^2}{2\sigma^2} + x_{m+1}y_{m+1} \ , \tag{2.10}$$

$$\tanh \vec{x}\vec{y} \quad\to\quad \tanh \sum_{i=1}^{m} x_i y_i + x_{m+1}y_{m+1} \ , \tag{2.11}$$

where $\vec{x}$ and $\vec{y}$ are $m+1$ dimensional vectors. The proposed method of constructing new kernels always generates a function satisfying the Mercer's condition, because it generates a function which is a sum of two kernels. For the new kernel type, the explicit form of (1.7) for δ-SVR is

$$x_{m+1} = \frac{-\sum_{i=1}^{2n} y_{\mathrm{c}}^i \alpha_i K_{\mathrm{o}}\left(\vec{x_i}, \vec{x_{\mathrm{red}}}\right) - b_{\mathrm{c}}}{\sum_{i=1}^{2n} y_{\mathrm{c}}^i \alpha_i c_i^{m+1}} \ . \tag{2.12}$$

2.1.1 Support Vectors

The SVC in δ-SVR is executed on duplicated number of examples and therefore the maximal number of support vectors of SVC is $2n$. We can reformulate (2.12) as

$$x_{m+1} = \frac{-\sum_{i=1}^{n} \left(\alpha_i - \alpha_{n+i}\right) K_{\mathrm{o}}\left(\vec{x_i}, \vec{x_{\mathrm{red}}}\right) - b_{\mathrm{c}}}{\sum_{i=1}^{2n} y_{\mathrm{c}}^i \alpha_i c_i^{m+1}} \ . \tag{2.13}$$

We call *support vectors for δ-SVR* vectors for which $\alpha_i - \alpha_{n+i} \neq 0$. The final number of support vectors for δ-SVR is maximally equal to n.

2.1.2 Basic Comparison With ε-SVR

The general idea of δ-SVR is that instead of finding the best model on the original data sample (like ε-SVR does), it finds the best model among multiple data transformations.

Both methods δ-SVR and ε-SVR have the same number of free parameters. For ε-SVR: C, kernel parameters, and ε. For δ-SVR: C, kernel parameters and δ. Each of them returns sparse solutions. Both parameters ε and δ control the number of support vectors.

There is an interesting relation between δ-SVR and ε-SVR for the proposed new kernels (2.8). We can write inequality constraints for δ-SVR (2.4) as

$$-\vec{w_{\mathrm{c,red}}}\cdot\vec{x_i} - w_c^{m+1}\left(y_r^i - \delta\right) - b_c \geq 1 - \xi_c^i \ , \tag{2.14}$$

$$\vec{w_{\mathrm{c,red}}}\cdot\vec{x_i} + w_c^{m+1}\left(y_r^i + \delta\right) + b_c \geq 1 - \xi_c^{*i} \ , \tag{2.15}$$

where $\xi_c^{*i} = \xi_c^{n+i}$. After reformulation:

$$-w_{\mathrm{c,red}}^{\vec{}} \cdot \vec{x_i} - b_c \geq w_c^{m+1} y_r^i - w_c^{m+1}\delta + 1 - \xi_c^i \ , \tag{2.16}$$

$$w_{\mathrm{c,red}}^{\vec{}} \cdot \vec{x_i} + b_c \geq -w_c^{m+1} y_r^i - w_c^{m+1}\delta + 1 - \xi_c^{*i} \ . \tag{2.17}$$

For $w_c^{m+1} > 0$, we get

$$\frac{-w_{\mathrm{c,red}}^{\vec{}} \cdot \vec{x_i} - b_c}{w_c^{m+1}} \geq y_r^i - \delta + \frac{1}{w_c^{m+1}} - \frac{\xi_c^i}{w_c^{m+1}} \ , \tag{2.18}$$

$$\frac{w_{\mathrm{c,red}}^{\vec{}} \cdot \vec{x_i} + b_c}{w_c^{m+1}} \geq -y_r^i - \delta + \frac{1}{w_c^{m+1}} - \frac{\xi_c^{*i}}{w_c^{m+1}} \ . \tag{2.19}$$

After transforming the above into regression convention we get

$$y_r^i - g\left(\vec{x_i}\right) \leq \delta - \frac{1}{w_c^{m+1}} + \frac{\xi_c^i}{w_c^{m+1}} \ , \tag{2.20}$$

$$g\left(\vec{x_i}\right) - y_r^i \leq \delta - \frac{1}{w_c^{m+1}} + \frac{\xi_c^{*i}}{w_c^{m+1}} \ , \tag{2.21}$$

where

$$g\left(\vec{x}\right) = \frac{-w_{\mathrm{c,red}}^{\vec{}} \cdot \vec{x} - b_c}{w_c^{m+1}} \ . \tag{2.22}$$

For $w_c^{m+1} < 0$, we get

$$y_r^i - g\left(\vec{x_i}\right) \geq \delta - \frac{1}{w_c^{m+1}} + \frac{\xi_c^i}{w_c^{m+1}} \ , \tag{2.23}$$

$$g\left(\vec{x_i}\right) - y_r^i \geq \delta - \frac{1}{w_c^{m+1}} + \frac{\xi_c^{*i}}{w_c^{m+1}} \ . \tag{2.24}$$

After changing notation from w_c^{m+1} to v, OP 3 can be formulated as

OP 4.

$$\min_{\vec{w_{\mathrm{r}}}, b_{\mathrm{r}}, \vec{\xi_{\mathrm{r}}}, \vec{\xi_{\mathrm{r}}^*}, v} \quad f\left(\vec{w_{\mathrm{r}}}, b_{\mathrm{r}}, \vec{\xi_{\mathrm{r}}}, \vec{\xi_{\mathrm{r}}^*}, v\right) = \|\vec{w_{\mathrm{r}}}, v\|^2 + C_{\mathrm{r}} \sum_{i=1}^{n} \left(\xi_{\mathrm{r}}^i + \xi_{\mathrm{r}}^{*i}\right) \tag{2.25}$$

subject to for $v > 0$

$$y_{\mathrm{r}}^i - g\left(\vec{x_i}\right) \leq \delta - \frac{1}{v} + \frac{\xi_r^i}{v} \ , \tag{2.26}$$

$$g\left(\vec{x_i}\right) - y_r^i \leq \delta - \frac{1}{v} + \frac{\xi_r^{*i}}{v} \ , \tag{2.27}$$

and for $v < 0$

$$-y_{\mathrm{r}}^i + g\left(\vec{x_i}\right) \leq -\delta - \frac{1}{-v} + \frac{\xi_r^i}{-v} \ , \tag{2.28}$$

$$-g\left(\vec{x_i}\right) + y_r^i \leq -\delta - \frac{1}{-v} + \frac{\xi_r^{*i}}{-v} \tag{2.29}$$

and

$$\vec{\xi}_{\mathrm{r}} \geq 0 \ , \tag{2.30}$$

$$\vec{\xi}_{\mathrm{r}}^{*} \geq 0 \tag{2.31}$$

for $i \in \{1, \ldots, n\}$, where

$$g\left(\vec{x_i}\right) = \vec{w}_{\mathrm{r}} \cdot \vec{x_i} + b_{\mathrm{r}} \ . \tag{2.32}$$

We can notice that when we have the solution of OP 4, the same solution can be found by running OP 2 with the parameters set to: when $v^* > 0$

$$\varepsilon = \delta - \frac{1}{v^*} \tag{2.33}$$

and

$$C_r^{\mathrm{new}} = C_r v^* \ , \tag{2.34}$$

and when $v^* < 0$, we have

$$\varepsilon = -\delta + \frac{1}{v^*} \tag{2.35}$$

and

$$C_r^{\mathrm{new}} = -C_r v^* \ . \tag{2.36}$$

Note that for $\varepsilon < 0$, we can replace (2.33) or (2.35) with

$$\varepsilon = 0 \tag{2.37}$$

due to the following proposition.

Proposition 2.1.1. *OP 2 for $\varepsilon < 0$ returns the same solution as for $\varepsilon = 0$.*

Proof. We will prove that the error difference between any two solution candidates after lowering ε from 0 to negative value remains unchanged. We have two solution candidates s_1 and s_2 with the following points: points with errors (p_e) and collinear points lying on the solution candidate, without errors. In the second group, we can further distinguish points that lie on both solution candidates (p_{csc}) and others (p_{css}). So we have two sets for both candidates

$$\left\{p_e^1, p_{css}^1, p_{csc}\right\} \ , \tag{2.38}$$

$$\left\{p_e^2, p_{css}^2, p_{csc}\right\} \ . \tag{2.39}$$

Because $p_{css}^2 \subset p_e^1$ and $p_{css}^1 \subset p_e^2$ so we can divide p_e to points p_{css} from other group and others (p_{er}). We can notice that $\left|p_{er}^1\right| = \left|p_{er}^2\right|$. So we have

$$\left\{p_{er}, p_{css}^2, p_{css}^1, p_{csc}\right\} \ , \tag{2.40}$$

$$\left\{p_{er}, p_{css}^1, p_{css}^2, p_{csc}\right\} \ . \tag{2.41}$$

For the first solution, the error difference is

$$\left(|p_{er}| + \left|p_{css}^2\right|\right) \Delta\varepsilon + \left(\left|p_{css}^1\right| + |p_{csc}|\right) \Delta\varepsilon \tag{2.42}$$

and for the second solution the error difference is

$$\left(|p_{er}| + \left|p^1_{css}\right|\right)\Delta\varepsilon + \left(\left|p^2_{css}\right| + |p_{csc}|\right)\Delta\varepsilon \ . \tag{2.43}$$

They are equal. □

We can see that when we fix the variable v to some value, we can get the same solution by using ε-SVR. But the additional variable is responsible for improving performance of ε-SVR. Let's consider the following example. For big value of ε, ε-SVR tends to return flat solutions, and in extreme it returns the solution $y = c$, where c is a constant. Such extreme solutions in most cases will be bad. We may decrease the value of ε to improve the solution. On the other hand, the δ-SVR has the additional variable v that eliminates tendency to return flat solutions. Consider two values of v, v_1 and $v_2 < v_1$, where $v_1, v_2 > 0$. Assume that v_1 value corresponds to the ε-SVR solution that is flat. The ability to decrease a value of v is related to decreasing the ε bounds, which is supported by the term v in the minimizer. It means that δ-SVR can automatically decrease the ε value.

2.1.3 Practical Realization

In practical realization, we find the best value of δ with a double grid search method by comparing some type of error measure. In the grid search method, we compare errors not on classification data, but on original regression data by using the regression function transformed from the classification decision boundary. We usually use mean squared error (MSE).

2.1.4 Weighting the Translation Parameter

We can consider incorporating prior knowledge by setting different values of the translation parameter for each example, so we can have δ_i parameters, for $i \in \{1, \ldots, n\}$ and the same parameters for $i \in \{n+1, \ldots, 2n\}$. We can also consider setting different values of δ for up and down translations, so we can have two parameters: δ_{u} and δ_{d}. And finally we can also consider the parameters δ^i_{u} and δ^i_{d} (additionally with the $\vec{C_c}$ weight)

OP 5.

$$\min_{\vec{w_{\mathrm{c}}}, b_{\mathrm{c}}, \vec{\xi_{\mathrm{c}}}} \quad f\left(\vec{w_{\mathrm{c}}}, b_{\mathrm{c}}, \vec{\xi_{\mathrm{c}}}\right) = \frac{1}{2}\left\|\vec{w_{\mathrm{c}}}\right\|^2 + \vec{C_{\mathrm{c}}} \sum_{i=1}^{2n} \xi^i_{\mathrm{c}} \tag{2.44}$$

subject to

$$y^i_{\mathrm{c}}\left(\vec{w_{\mathrm{c,red}}} \cdot \vec{x_i} + w^{m+1}_c\left(y^i_r + y^i_c\delta_i\right) + b_c\right) \geq 1 - \xi^i_{\mathrm{c}} \ , \tag{2.45}$$

$$\vec{\xi_{\mathrm{c}}} \geq 0 \tag{2.46}$$

for $i \in \{1, \ldots, 2n\}$, where $\delta_i = \delta^i_u$ for $i \in \{1, \ldots, n\}$ and $\delta_i = \delta^i_d$ for $i \in \{n+1, \ldots, 2n\}$.

2.2 Analysis of the Transformation

We analyze a general concept of representing regression problems as classification ones by duplicating and shifting data, introduced in δ-SVR. Intuitively, the transformed classification problem should lead to the similar results as the original regression one, Fig. 2.1, Fig. 2.2. We show the equivalence of Bayes solutions for regression and transformed classification problems for some special cases.

Random mapping $\vec{x_{\mathrm{r}}} \rightarrow y_{\mathrm{r}}$ is duplicated and original random mapping is translated up, and duplicated one is translated down. The random mapping is converted to random variable $\vec{x_{\mathrm{c}}}$, and original random variable gets 1 class, and duplicated one gets -1 class. We can notice that transformed data have a special distribution $\mathrm{F_c}\left(\vec{x_{\mathrm{c}}}\right)$ where random coordinate x_{m+1} is dependent on the remaining coordinates; for $\delta = 0$, $\mathrm{F_c}\left(\vec{x_{\mathrm{c}}}\right) \equiv \mathrm{F_r}\left(\vec{x_{\mathrm{r}}}, y_{\mathrm{r}}\right)$.

After transformation, Bayes optimal classification depends on a sign of

$$\Pr\left(1 \mid \vec{x_{\mathrm{c}}}\right) - \Pr\left(-1 \mid \vec{x_{\mathrm{c}}}\right) \quad . \tag{2.47}$$

A Bayes decision boundary is a group of points for which $\Pr\left(1 \mid \vec{x_{\mathrm{c}}}\right) \equiv \Pr\left(-1 \mid \vec{x_{\mathrm{c}}}\right)$. A regression function is defined as

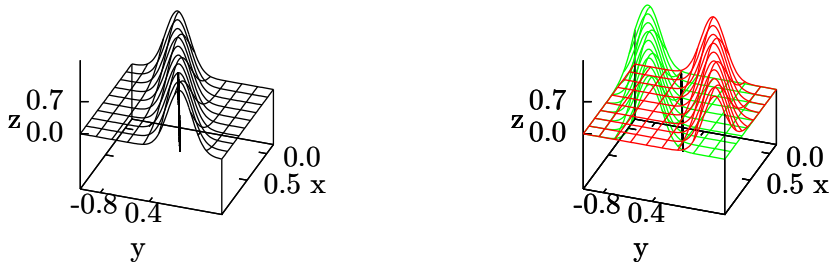
$$\mathrm{r}\left(\vec{x_{\mathrm{r}}}\right) = \mathbb{E}\left[y_{\mathrm{r}} \mid \vec{x_{\mathrm{r}}}\right] \quad . \tag{2.48}$$

Theorem 2.2.1. *For a unimodal, symmetrical probability distribution* $\mathrm{F_r}\left(y_{\mathrm{r}} \mid \vec{x_{\mathrm{r}}}\right)$ *of original examples, Bayes decision boundary of the transformed classification problem is equivalent geometrically to the regression function for every* $\delta \geq 0$.

(A proof is in Appendix A.2). The example to the theorem is depicted in Fig. 2.3. The theorem states that assuming symmetrical errors in the regression output, for any nonnegative value of δ the transformed classification problem is equivalent to the original one. The theorem can be extended to different unimodal, symmetrical distributions per point $\left(\mathrm{F}_{\mathrm{r}}^{x}\left(y_{\mathrm{r}} \mid \vec{x_{\mathrm{r}}}\right)\right)$, with the same effect. The question arises about asymmetric distributions.

For asymmetric (skewed), unimodal distributions the mean is different from the mode. For such distributions the mean lies on the side of a mode with a bigger variance. It can be noticed that after translating by δ, Bayes optimal decision boundary lies on the same side of the mode as the mean. Therefore it seems that δ-SVR could also handle asymmetric distribution of errors efficiently. We can reach the equivalence in two ways, either by choosing proper δ or by using δ_u and δ_d parameters instead of δ. First, we propose the following theorem

Theorem 2.2.2. *When* $F\left(m + \delta\right) - F\left(m - \delta\right) \geq 0$ *is satisfied for some* $\delta > m$, *then for a unimodal, probability distribution* $\mathrm{F_r}\left(y_{\mathrm{r}} \mid \vec{x_{\mathrm{r}}}\right)$ *of original examples, Bayes decision boundary of the transformed classification problem is equivalent geometrically to the regression function, where* $m > 0$ *is the expected value.*



(a) (b)

Figure 2.3: The idea of the transformation of the Bayes solution in δ-SVR. A solid line - a Bayes solution. (a) A density function before transformation. (b) Two density functions after transformation

(A proof is in Appendix A.3). The theorem states that by testing different values of δ we can reach the equivalence of the problems for the asymmetric case, when the distribution satisfies some general assumptions. And finally we can use δ_u and δ_d parameters

Theorem 2.2.3. *For a unimodal, asymmetric probability distribution* $\mathrm{F_r}\left(y_\mathrm{r} \mid \vec{x_\mathrm{r}}\right)$ *of original examples, Bayes classification of the transformed classification problem is equivalent geometrically to the regression function for some ratio* δ_u/δ_d *for every* $\delta_u \geq 0$.

When $\delta_u/\delta_d = 1$, then we have a symmetrical distribution. When $\delta_u/\delta_d > 1$ the distribution has a bigger variance on the upper side of the optimal regression function. The above theorem indicates that it would be possible to improve the results for asymmetric regression errors by introducing the new parameter to δ-SVR. The disadvantage of this improvement is that a value of the ratio must be found either by experiments (this is an additional parameter that must be tuned) or by testing the skewness of the distribution. This extension of δ-SVR will be evaluated practically in the future.

2.3 Generalization Performance of δ-SVR

The key point in statistical learning theory is the analysis of generalization capabilities of machine learning methods without assuming any particular data distribution, [15, 16]. The δ-SVR for particular values of δ uses SVC for solving classification problems, therefore all analysis of generalization capabilities of SVC are applicable for δ-SVR for any δ. The δ-SVR provides the known dependency to the distribution of the classification data without reducing the possible universe of the problems. Therefore we will first analyze how this distribution constraint influences generalization capabilities of machine learning methods.

In this section we compare empirical risk minimization (ERM) principle for the original regression problem and transformed classification problems. Then, we compare realization of SRM by ε-SVR and δ-SVR. And finally we consider generalization bounds for SVC for shifted data without assuming any data distribution.

2.3.1 Empirical Risk Minimization for δ-SVR

The ERM principle states that we should minimize empirical risk. It means that for classification problems we should minimize the number of training errors, and for regression problems we should minimize the sum of training errors. So empirical risk for regression is a real number measure, for classification it is a discrete measure. For transformed classification problems when increasing value of δ starting from zero, a minimum of the classification empirical risk decreases and tends to zero

$$\mathrm{R}_{\mathrm{emp}}\left(\alpha_l\right) \underset{\delta\to\infty}{\to} 0 \ , \tag{2.49}$$

where α_l is a curve for which $\mathrm{R}_{\mathrm{emp}}$ is minimal. We can notice that there exists δ_p for which all training examples are correctly classified. Hence for all $\delta \geq \delta_p$ transformed data are correctly classified. Moreover for $\delta \geq \delta_p$ there may exist multiple solutions with no training errors at all. It means that for some values of δ ERM for classification might hardly give a valuable solution. So for such cases better results could be obtained by using ERM for regression. The ERM for regression has an advantage that the output is nonzero (except some degenerate cases, for example for a linear function goinng through collinear examples). This suggests that the grid search method used for choosing the best value of δ might compare empirical risk for original regression data instead of empirical risk for classification data.

The ε-SVR and SVC realize a trade-off between ERM and minimizing a VC dimension which describes capacity of a learning machine. The ERM for ε-SVR is realized in a standard way by minimizing a sum of training errors. In SVC, ERM is realized by minimizing a sum of slack variables (1.1). Therefore for a particular value of δ, δ-SVR, which uses SVC, also minimizes a sum of slack variables. It does not minimize ERM for the regression. In the following subsection, we compare in details similarities and differences between ERM for classification and regression.

2.3.2 Comparison of ERM for ε-SVR and δ-SVR

Comparing ERM for ε-SVR and δ-SVR leads to a comparison of the second terms in cost functions (1.8) and (1.1). Let's analyze all hypotheses where $\|\vec{w_{\mathrm{c}}}\| = p$ and $\|\vec{w_{\mathrm{r}}}\| = q$, where p and q are some constants such as $p, q \geq 0$. First, we will define examples involved in realization of ERM: for δ-SVR, let's call a vector lying on margin boundaries or inside margin boundaries, *an essential margin vector* and a set of such vectors for a particular hypothesis, EMV. For ε-SVR, let's call a vector lying on ε boundaries or outside ε boundaries, *an essential margin vector* and a set of such vectors for a particular hypothesis, EMV. By a configuration of essential margin vectors, called $CEMV$, we

mean a list of essential margin vectors for a particular hypothesis, each with the distance to a margin boundary (or ε boundary).

Let's imagine all hypotheses for some p and q. The ε-SVR realizes ERM by finding the hypothesis that has a minimal value of a sum of differences in distances in an output direction from EMV to the hypothesis function. The δ-SVR realizes ERM by finding the hypothesis that has a minimal value of a sum of differences in perpendicular distances in transformed space between EMV and the hypothesis curve.

Theorem 2.3.1. *For $\|\vec{w}_c\| = p$, SVC minimizes a sum of perpendicular distances from the decision curve to EMV.*

Proof. For different hypotheses with $\|\vec{w}_c\| = p$ a first term in the cost function (1.1) is constant, so we minimize only the second term. The distance from the i-th example with nonzero ξ_i to a margin boundary is $\xi_i / \|\vec{w}_c\|$. Because the denominator is constant, minimizing distances to examples lying outside margin boundaries means minimizing a sum of ξ_i. □

This theorem leads to the potential relation of SVC to the *total least squares regression* method (*orthogonal regression*), which is used mainly for errors-in-variable data. We can notice that for completely flat curves sum of perpendicular distances is equal to a sum of distances in x_c^{m+1} direction and the difference grows for less flat functions. So this might be the reason of expecting better performance of ERM for δ-SVR for data with only output errors, for flat functions. Now when we know how ERM is computed for ε-SVR and δ-SVR for specific values of δ and ε, we analyze which examples are involved in computing ERM.

First, we propose

Proposition 2.3.2. *For two values of δ, $\delta_1 > 0$ and $\delta_2 > 0$, where $\delta_2 > \delta_1$, for every $CEMV$ for δ_1, there exists the same $CEMV$ for δ_2.*

When we consider $CEMV$ for δ_2, $h(\vec{x}) = 0$ and increasing a value of δ by $\Delta\delta = \delta_2 - \delta_1$ we get the same $CEMV$ for $ph(\vec{x}) = 0$, where $p = 1/\left(1 + w_c^{m+1}\Delta\delta\right)$. This proposition states that the same $CEMV$ could be present for multiple values of δ. This is a difference from ε-SVR where every $CEMV$ is present only once for one value of ε. We can also notice that the distance to the margin boundaries from the solution for δ_2 is $1/\|p\vec{w}_c\|$.

Now let's investigate a closer relation between ε-SVR and δ-SVR.

Proposition 2.3.3. *Every $CEMV$ for ε-SVR for a particular value ε_s is present in classification setting in δ-SVR for every $\delta > \delta_p$, where $\delta_p = \varepsilon_s$.*

We choose margin boundaries in a way that the distance to them in an output direction is equal to $\delta - \varepsilon$. We can extend this proposition to the following.

Proposition 2.3.4. *Every $CEMV$ for ε-SVR for every $\varepsilon < \varepsilon_s$ is present in classification setting in δ-SVR for every $\delta > \delta_p$, where $\delta_p = \varepsilon_s$.*

The above proposition means that for a single value of δ, δ-SVR is able to take into account a bunch of $CEMV$ from ε-SVR for multiple values of ε. Note that δ-SVR can have $CEMV$ that are absent from ε-SVR.

Proposition 2.3.5. *When $|EMV| \leq n$ for δ-SVR then the same $CEMV$ exists for ε-SVR.*

The $CEMV$ of δ-SVR where $|EMV| > n$ are absent from ε-SVR. It can be noticed that when $|EMV| > n$, there exists an equivalent EMV for regression when taking into account optimization for support vectors stated in (2.13). It is a consequence of the fact that all support vectors for SVC lying below margin boundaries have $\alpha_i = C$, which is a conclusion from Karush-Kuhn-Tucker complementary condition for SVC; therefore they disappear in (2.13) and are not support vectors for δ-SVR. For example when $|EMV|$ is close to $2n$ we get the small number of support vectors for δ-SVR.

Summarizing, it is most likely that comparing ERM for particular values of ε and δ, δ-SVR would perform better. Next we will investigate a trade-off between ERM and capacity minimization (CM).

In order to compare realization of the trade-off between ERM and CM first we rewrite δ-SVR cost function by incorporating perpendicular distances from EMV to the curve

$$d_{\mathrm{c}}^{i} = \frac{\xi_{\mathrm{c}}^{i}}{\|\vec{w}_{\mathrm{c}}\|} \ . \tag{2.50}$$

The δ-SVR minimization function (1.1) can be rewritten as

$$f\left(\vec{w}_{\mathrm{c}}, b_{\mathrm{c}}, \vec{\xi_{\mathrm{c}}}\right) = \|\vec{w}_{\mathrm{c}}\|^2 + C_{\mathrm{c}} \|\vec{w}_{\mathrm{c}}\| \sum_{i=1}^{n} d_{\mathrm{c}}^{i} \ . \tag{2.51}$$

When we treat the differences between distances in the last coordinate direction and perpendicular distances negligible, then we can see that the difference between the cost function for ε-SVR (1.8) and the above for δ-SVR is $\|\vec{w_{\mathrm{c}}}\|$. For ε-SVR a trade-off between the empirical risk and the capacity is controlled by C_{r}, for δ-SVR we can also control the trade-off with C_{c}, but additionally it depends on $\|\vec{w_{\mathrm{c}}}\|$. It means that when the capacity decreases (a geometric margin increases), a value of the trade-off also decreases.

Let's analyze the trade-off while changing a value of δ. For a particular value of $CEMV$ for δ_1 increasing δ to δ_2, where $\delta_2 > \delta_1$ and preserving the same $CEMV$ leads to

$$f\left(\vec{w}_{\mathrm{c}}, b_{\mathrm{c}}, \vec{\xi_{\mathrm{c}}}\right) = p^2 \|\vec{w}_{\mathrm{c}}\|^2 + C_{\mathrm{c}} p \|w_{\mathrm{c}}\| \sum_{i=1}^{n} d_{\mathrm{c}}^{i} \ , \tag{2.52}$$

where

$$p = \frac{1}{1 + w_{\mathrm{c}}^{m+1} (\delta_2 - \delta_1)} \ , \tag{2.53}$$

so

$$f\left(\vec{w}_{\mathrm{c}}, b_{\mathrm{c}}, \vec{\xi_{\mathrm{c}}}\right) = p^2 \left(\|\vec{w}_{\mathrm{c}}\|^2 + \frac{C_{\mathrm{c}}}{p} \|w_{\mathrm{c}}\| \sum_{i=1}^{n} d_{\mathrm{c}}^{i} \right) \ , \tag{2.54}$$

$$f\left(\vec{w_{\mathrm{c}}}, b_{\mathrm{c}}, \vec{\xi_{\mathrm{c}}}\right) = p^2\left(\left\|\vec{w_{\mathrm{c}}}\right\|^2 + \left(1 + w_{\mathrm{c}}^{m+1}\left(\delta_2 - \delta_1\right)\right) C_{\mathrm{c}} \left\|w_{\mathrm{c}}\right\| \sum_{i=1}^{n} d_{\mathrm{c}}^{i}\right) \quad . \tag{2.55}$$

While increasing a value of δ, the trade-off between ERM and CM changes even for the curve with the same $CEMV$. The change depends on the last coefficient. For bigger values of w_{c}^{m+1}, the importance of ERM increases more while increasing δ.

2.3.3 VC Bounds for δ-SVR

In this subsection, we consider the translation for δ-SVR independently of the data distribution. The generalization bounds based on a VC dimension are as following, [16]. With the probability at least $1 - \eta$ the inequality holds true

$$\mathrm{R}\left(\alpha\right) \leq \mathrm{R}_{\mathrm{emp}}\left(\alpha\right) + \frac{\varepsilon\left(n\right)}{2}\left(1 + \sqrt{1 + \frac{4\mathrm{R}_{\mathrm{emp}}\left(\alpha\right)}{\varepsilon\left(n\right)}}\right) \quad , \tag{2.56}$$

where

$$\varepsilon\left(n\right) = 4\frac{\ln 2\tau + 1}{\tau} - \frac{\ln \eta/4}{n} \quad , \tag{2.57}$$

$$\tau = \frac{n}{h} \quad , \tag{2.58}$$

h is a VC dimension. For real valued functions when the admissible set of functions is a set of totally bounded functions ($0 \leq Q\left(z, \alpha\right) \leq B$), with the probability at least $1 - \eta$ the inequality holds true

$$\mathrm{R}\left(\alpha\right) \leq \mathrm{R}_{\mathrm{emp}}\left(\alpha\right) + \frac{B\varepsilon\left(n\right)}{2}\left(1 + \sqrt{1 + \frac{4\mathrm{R}_{\mathrm{emp}}\left(\alpha\right)}{B\varepsilon\left(n\right)}}\right) \quad . \tag{2.59}$$

Therefore the bounds for classification and regression are pretty much the same. They are independent of data distribution. The key to minimize the right hand side is to control h. For this purpose Vapnik [16] proposed SRM. For SVC, it is realized by controlling the trade-off between ERM and CM. Let's see the relation of CM to h.

Consider hyperplanes $\vec{w_{\mathrm{c}}} \cdot \vec{x} = 0$, where $\vec{w_{\mathrm{c}}}$ is normalized such that they are in a canonical form, that is for a set of points $\mathrm{A} = \{\vec{x_1}, \ldots, \vec{x_n}\}$

$$\min_i \left|\vec{w_{\mathrm{c}}} \cdot \vec{x_i}\right| = 1 \quad . \tag{2.60}$$

The set of decision functions $f_w\left(\vec{x}\right) = \operatorname{sgn} \vec{x} \cdot \vec{w_{\mathrm{c}}}$ defined on A, satisfying the constraint $\left\|\vec{w_c}\right\| \leq \Lambda$ has a VC dimension satisfying

$$h \leq \min\left(R^2\Lambda^2, m + 1\right) \quad , \tag{2.61}$$

where R is the radius of the smallest sphere centered at the origin and containing A. This theorem could be generalized for any hyperplanes, not necessarily crossing the 0

point. The proof can be found in [14]. So minimization of $\|\vec{w}_c\|$ is a minimization of the upper bound on h.

There are two factors that have influence on a VC bound for SVC, Λ and R. The SVC realizes CM by minimizing the first one. The second factor is rather constant for standard classification and regression methods. But for δ-SVR, R is a variable, hence it leads to the opportunity to improve VC bounds.

For δ-SVR, R depends on a value of δ. Let's consider changing δ from δ_1 to δ_2, $\Delta\delta = \delta_2 - \delta_1$, $\Delta\delta > 0$. After this change a VC bound takes a form

$$h \leq p^2\Lambda^2 \left(R + \Delta\delta\right)^2 \quad , \tag{2.62}$$

where

$$p = \frac{1}{1 + w_c^{m+1}\Delta\delta} \quad . \tag{2.63}$$

When increasing δ, R is increasing, and Λ is decreasing. Therefore δ is a trade-off between R and Λ. We can see that it is possible to improve the bound by increasing a value of δ. Consider the inequality describing the improvement

$$p^2 \left(R + \Delta\delta\right)^2 < R^2 \quad . \tag{2.64}$$

The solution for $p > 0$ (see Appendix A.4) is

$$w_c^{m+1} > \frac{1}{R} \quad . \tag{2.65}$$

For $p < 0$

$$w_c^{m+1} < \frac{-2}{\Delta\delta} - \frac{1}{R} \quad . \tag{2.66}$$

Let's have a look on the example for $p > 0$, $m = 1$. Consider a bunch of hypotheses with $\|\vec{w}_c\| = c$, we can rewrite the decision curve as a function of the last coordinate

$$x_c^{m+1} = -w_c^m x_c^m / w_c^{m+1} \quad . \tag{2.67}$$

For $w_c^m < 0$ increasing slope is done by decreasing w_c^{m+1} and increasing w_c^m. It means that for less positive slope we expect better VC bound.

Therefore δ-SVR has a potential to improve a VC bound by shifting without worsening empirical risk (2.49).

Let's consider VC bounds for δ-SVR and ε-SVR. We start the analysis from the classification problem introduced by δ-SVR for some value of δ. The δ-SVR uses SVC to solve it, so we realize CM by using the term $\|w_c\|^2$. The ε-SVR can be interpreted as δ-SVR with lack of the last variable, see OP 4. So ε-SVR does not minimize the whole term $\|w_c\|^2$, but the term without the last coefficient. So we think that δ-SVR better realizes SRM principle than ε-SVR.

2.4 Solving Total Least Squares Regression with δ-SVR

Orthogonal regression is a variant of total least squares (TLS) regression. The TLS method, [8], is a technique for solving overdetermined system of equations $AX \approx B$, where $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{m\times d}$. The variable $X \in \mathbb{R}^{n\times d}$ is unknown. The TLS regression is a natural generalization of the least squares (LS) approximation method when the data in both A and B are perturbed. The LS approximation $\hat{X}$ is obtained as a solution of the following optimization problem:

$$LS := \arg\min \|\Delta B\|_F \tag{2.68}$$

subject to

$$AX = B + \Delta B \ . \tag{2.69}$$

The idea behind this approximation method is to correct the right-hand side B as little as possible in the Frobenius norm sense, so that the corrected system of equations $AX = B + \Delta B$ has an exact solution. The definition of the TLS method is motivated by the asymmetry of the LS method: B is corrected, while A is not. The classical TLS problem looks for the minimal (in the Frobenius norm sense) correction ΔA and ΔB on the given data A and B that make the corrected system of equations solvable, that is

$$TLS := \arg\min \|[\Delta A \ \Delta B]\|_F \tag{2.70}$$

subject to

$$(A + \Delta A)\, X = B + \Delta B \ . \tag{2.71}$$

The LS approximation is statistically motivated as a maximum likelihood estimator in a linear regression model under standard assumptions (zero mean, normally distributed residual with a covariance matrix that is a multiple of the identity). Similarly, the TLS approximation is a maximum likelihood estimator in the errors-in-variables model:

$$A = \overline{A} + \tilde{A}, \quad B = \overline{B} + \tilde{B} \ . \tag{2.72}$$

There exists an $\overline{X} \in \mathbb{R}^{n\times d}$ such that $\overline{A}\ \overline{X} = \overline{B}$, under the assumptions that $\left[\tilde{A}\ \tilde{B}\right]$ is a zero mean, normally distributed random vector with a covariance matrix that is a multiple of the identity.

The LS and TLS methods assess the fitting accuracy in different ways: the LS method minimizes the sum of squared vertical distances from the data points to the fitting line, while the TLS method with the same distribution of errors among input features minimizes the sum of the squared orthogonal distances from the data points to the fitting line, [3], and therefore it is also called *orthogonal regression*. It is reported that difference between the performance of LS and TLS is small, and is significant especially for ill-posed problems.

δ-SVR realizes ERM by minimizing for fixed $\|\vec{w}_c\|$ a sum of squared distances, Thm. 2.3.1. We can notice that for completely flat curves sum of perpendicular distances is equal to a sum of distances in x_c^{m+1} direction and the difference grows for less flat functions. This is the reason of expecting better performance of ERM for δ-SVR for data with only output errors, for flat functions.

2.5 Using Extensions of SVC for Regression Problems

For solving δ-SVR, we use SVC, thus any type of methods and optimizations for solving SVC can be applied to regression problems. Moreover, we can use extensions of SVC in regression problems. One of applications of the extensions is incorporation of prior knowledge. For example, the φ support vector classification (φ-SVC) method was proposed by me in [9] to incorporate knowledge about hyperspheres that cannot cross the solution from a particular side. Such prior knowledge can be directly converted to regression problems with the similar meaning, that hyperspheres cannot cross the regression function from a particular side.

2.6 Experiments

For solving ε-SVR and SVC for particular parameters we use LibSVM, [1], ported to Java. For all data sets, every feature is scaled linearly to $[0, 1]$ including an output. For variable parameters like C, σ for the RBF kernel, we use a double grid search method for finding the best values. The number of values searched by the grid method is a trade-off between an accuracy and a speed of simulations. Note that for particular data sets, it is possible to use more accurate grid searches than for massive tests with multiple number of simulations. All tests are performed either on synthetic or real world data sets. Synthetic data sets are generated from particular functions with added Gaussian noise for output values, Table 2.2. We performed tests with a linear kernel on linear functions, with a polynomial kernel on the polynomial function, with the RBF kernel on the sine function.

The real world data sets were taken from the LibSVM site, [6], except stock price data, Table 2.3. They originally come from UCI Machine Learning Repository and StatLib DataSets Archive. The stock price data consist of monthly prices of the Dow Jones Industrial Average (DJIA) index from 1898 up to 2010. We generated the stock data as follows: for every month the output value is a growth/fall comparing to the next month. Every feature i is a percent price change between the month and the i-th previous month.

For all tests we choose a size of training sets satisfying $n/h < 20$. Recently, Yang et al. [18] used double cross-validation for SVM. We use double cross-validation for comparing performance of δ-SVR with ε-SVR, 5 fold inner cross-validation is used. Outer cross-validation is slightly modified it order to allow using a small training set size: if a training set size is less than a half of all known mappings, then we use cross-validation but for training data, otherwise we use standard cross-validation. When it is greater then the number of possible steps for cross-validation additional data shuffles are performed.

In the first experiment, we check the theoretical result from Prop. 2.3.4 by comparing the number of support vectors and generalization performance for some values of δ and ε. In the second experiment, we compare the generalization performance for variable δ and ε.

Table 2.1: Relation between ε, δ and RMSE. Column descriptions: *id* – an id of a test (for synthetic s prefix, for real world data sets t prefix), ε,δ – a value of a parameter ε or δ, *ti* – a percentage difference in RMSE between ε-SVR and δ-SVR for the i-th test, positive means that δ-SVR has smaller RMSE

id	ε,δ	s0	s3	s4	t0	t1	t2	t3	t4	t5	t9
1	0.01	−1.24	−2.5	**1.0**	−4.1	−14.8	−5.54	−14.4	−1.24	−6.22	−9.24
2	0.04	−0.05	−0.4	**1.8**	**0.1**	**3.64**	−0.07	**1.03**	−0.05	**1.01**	−1.03
3	0.16	**53.8**	**49.4**	−0.47	**18.3**	**21.3**	**20.1**	**3.1**	**53.8**	**8.5**	**8.74**
4	0.32	−58.5	**72**	−0.54	**39.5**	**39.3**	**37.5**	**14.5**	**75.9**	**24.7**	**37.67**
5	0.64	**46.3**	**6.0**	**2.97**	**36.2**	**42.1**	**44.6**	**25.9**	**46.3**	**23**	**45.7**

2.6.1 First Experiment

In the first experiment, we check the theoretical result from Prop. 2.3.4 that $|EMV|$ is much broader for δ-SVR than for ε-SVR for particular values of δ and ε, so we check how $|EMV|$ depends on a value of ε and δ. For this purpose we compare the number of support vectors for the same values of δ and ε. We expect greater number of support vectors especially when values of the parameters increases and the number of support vectors for ε-SVR is close to 0. Results are depicted in Fig. 2.4, Fig. 2.5.

We can see that for ε-SVR the number of support vectors decreases while increasing ε. We can see that while the number of support vectors is close to zero for ε-SVR, δ-SVR can return a solution with more support vectors, therefore δ-SVR can return better solutions than ε-SVR while comparing a broad range of values of δ and ε.

In the second part of the first experiment, we compare generalization performance for δ-SVR and ε-SVR for various values of δ and ε. Performance for δ-SVR is better than ε-SVR for various δ and ε, except a value closest to zero, for which a performance is worse (Table 2.1, Fig. 2.6, Fig. 2.7). We can also notice greater difference in the performance when the difference in the number of support vectors is bigger. We can see that the performance of δ-SVR is similar and close to the best value for any δ in a checked range of values, while performance of ε-SVR decreases while increasing ε.

The results may be valuable in practice when we lack of enough time to find the best values of δ and ε. Then we expect better generalization performance of δ-SVR than ε-SVR. In the next experiment we will compare results for variable δ and ε.

2.6.2 Second Experiment

In the second experiment, we compare generalization performance of ε-SVR and δ-SVR for variable ε and δ. We use a double grid search method for finding the best values of ε and δ. We limit the number of iterations to 10000 in these tests for both methods, for results without this limit and additional description of parameters of the tests see [13].

Test results on synthetic data sets are presented in Table 2.2. We can notice similar generalization performance for both δ-SVR and ε-SVR, with one statistically better

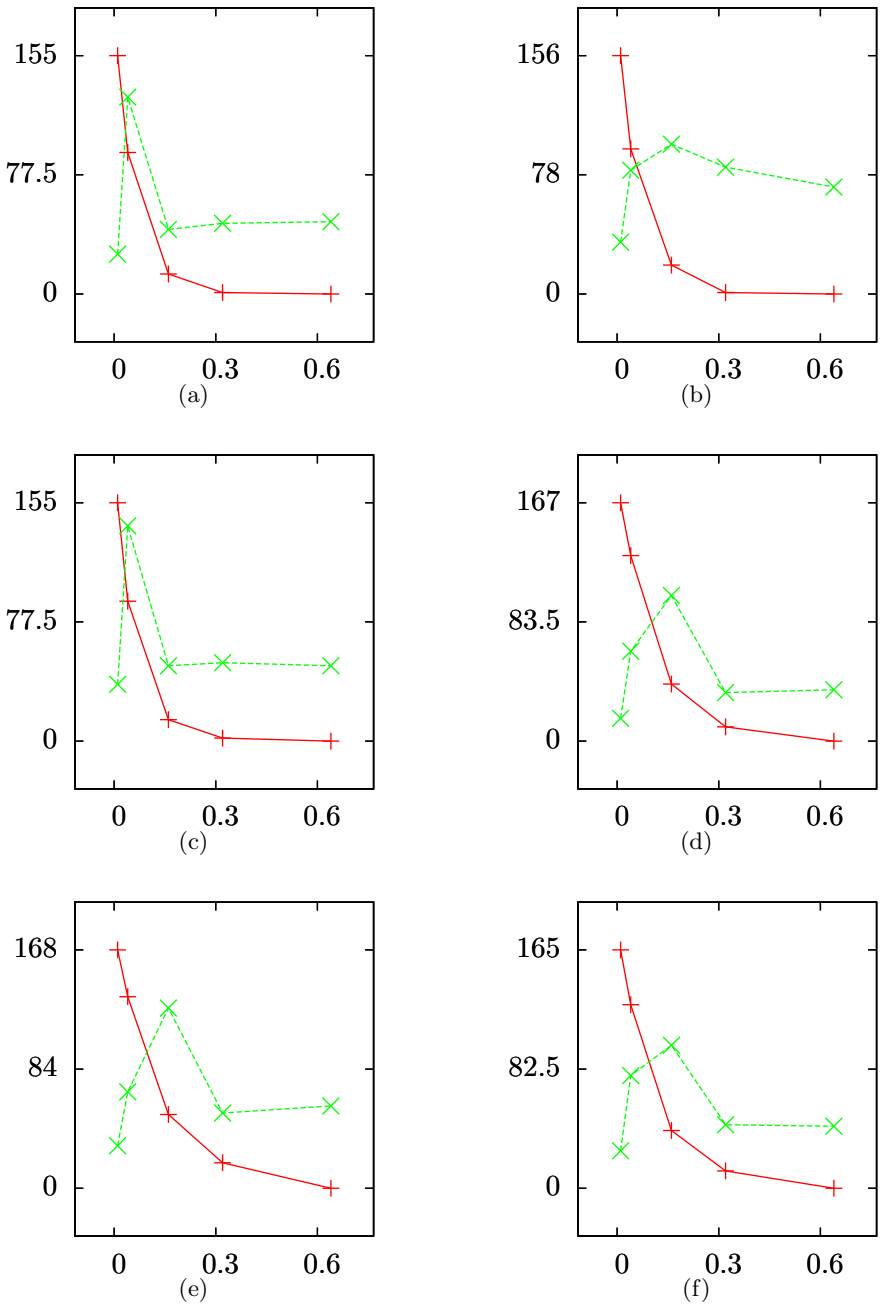


Figure 2.4: Relation between ε, δ and the number of support vectors for the test cases with ids 0-5 from Table 2.3a. A function with '+' points represents the relationship between value of ε and the number of support vectors, a function with 'x' points represents the relationship between value of δ and the number of support vectors

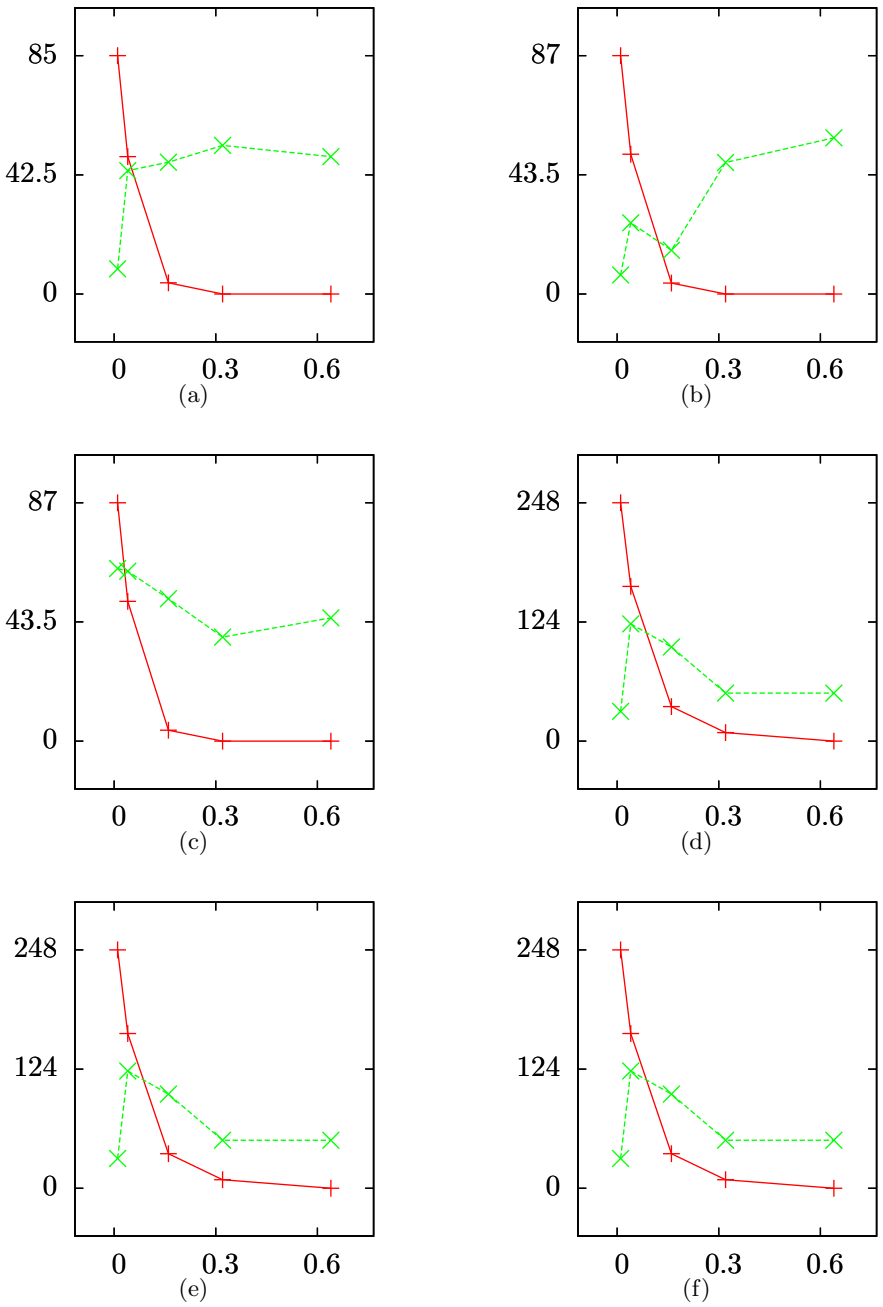


Figure 2.5: Relation between ε, δ and the number of support vectors for the test cases with ids 6-11 from Table 2.3a, cont. A function with '+' points represents the relationship between value of ε and the number of support vectors, a function with 'x' points represents the relationship between value of δ and the number of support vectors

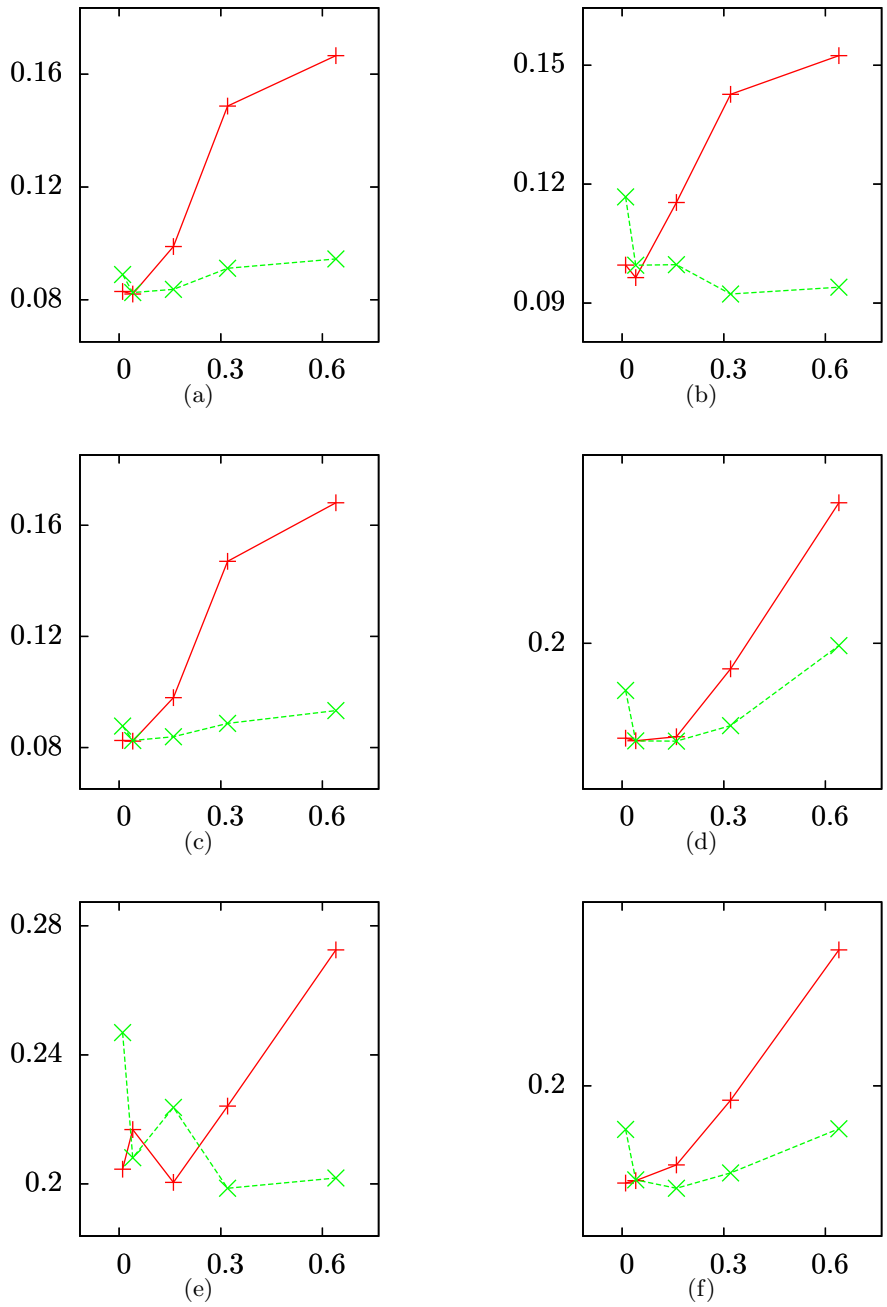


Figure 2.6: Relation between ε, δ and RMSE for the test cases with ids 0-5 from Table 2.3a. A function with '+' points represents the relationship between value of ε and RMSE, a function with 'x' points represents the relationship between value of δ and RMSE.

result for δ-SVR, and one for ε-SVR. However, we can notice an improved number of support vectors for δ-SVR for all tests, which is also statistically significant for all of

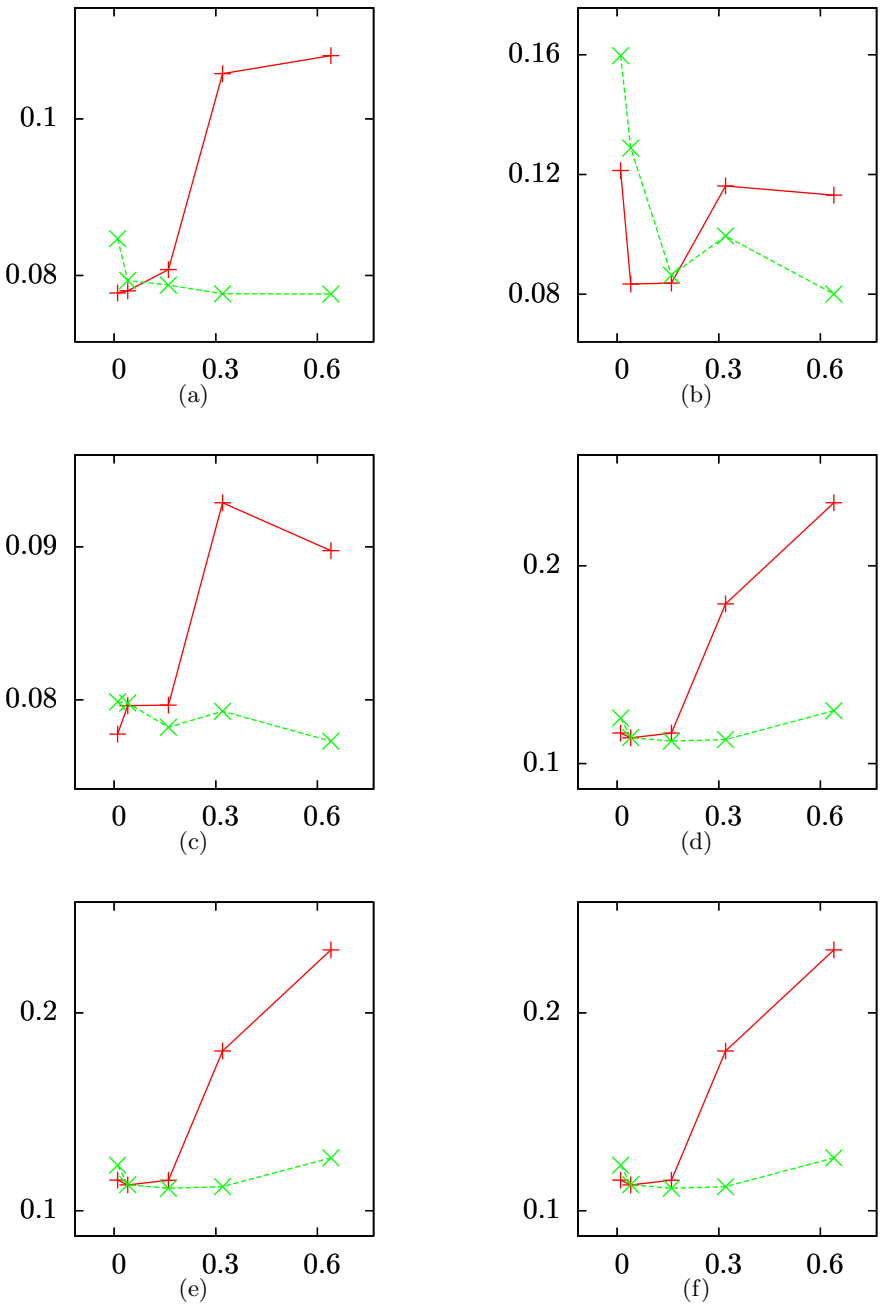


Figure 2.7: Relation between ε, δ and RMSE for the test cases with ids 6-11 from Table 2.3a, cont. A function with '+' points represents the relationship between value of ε and RMSE, a function with 'x' points represents the relationship between value of δ and RMSE.

them.

For real world data sets, results are presented in Table 2.3. We can notice similar

Table 2.2: Performance of δ-SVR for synthetic data. Column descriptions: *id* – id of the test, *a function* – a function used for generating data $y_1 = \sum_{i=1}^{\dim} x_i$, $y_4 = \left(\sum_{i=1}^{\dim} x_i\right)^{\text{kerP}}$, $y_5 = 0.5 \sum_{i=1}^{\dim} \sin 10x_i + 0.5$, *ker* – a kernel with a kernel parameter (for a polynomial kernel it is a dimension, for the RBF kernel it is σ), *idRef* – a reference to the test, *te12M* – a percent average difference in MSE for testing data, if greater than 0 than δ-SVR is better, *teT* – t value for the t-test for comparing testing error, *s1* – the average number of support vectors for ε-SVR, *s2* – the average number of support vectors for δ-SVR, *sT* – t value for the t-test for comparing the number of support vectors

(a)

id	function	ker
0	y_1	denseLinear 0.0
1	$y_2 = 3y_1$	denseLinear 0.0
2	$1/3y_1$	denseLinear 0.0
3	y_4	densePolynomial 5.0
4	y_5	denseRBF 0.25

(b)

idRef	te12M	teT	s1	s2	sT
0	3.92	2.4	77	63	2.82
1	1.02	0.87	74	60	3.24
2	0.33	0.25	73	62	2.27
3	−3.44	−1.18	76	69	1.81
4	0.32	0.38	56	49	2.07

generalization performance for both δ-SVR and ε-SVR without any statistical difference, except two tests for a polynomial kernel when ε-SVR is better. However, we can notice the improved number of support vectors for δ-SVR for 10 tests out of 12 tests, for one of them ε-SVR is better with statistical significance. The statistical significance for the number of support vectors is achieved for δ-SVR without the limitation of 10000 iterations and for variable σ parameter for RBF kernel for about half of the tests, see the results in [13].

2.7 Summary

In this report, we analyzed a novel regression method, called δ-SVR. We conducted experiments comparing δ-SVR with ε-SVR on synthetic and real world data sets. The results indicate that δ-SVR achieves comparable generalization error. The first advantage of δ-SVR is fewer support vectors. Thus we get simpler predictive models. Therefore, computational time of testing new examples is decreased. The next advantage is smaller generalization error over different values of ε and δ. Therefore, there exists possibility to decrease time of training, but with accepting suboptimal solutions. The next advantage is faster time of training for linear kernels while using SMO solver. The last advantage of δ-SVR, but not least, is the possibility of replacing the standard SVC classification method by any other classification method based on kernel functions. In particular, any improvements for classification methods in respect of generalization error, speed of training and testing, ability to incorporate prior knowledge, can be used directly for regression problems.

The disadvantage of δ-SVR are ambiguous results comparing time of training for

Table 2.3: Performance of δ-SVR for real world data. Column descriptions: *id* – id of the test, *dn* – a data set, *ker* – a kernel with a kernel parameter (for a polynomial kernel it is a dimension, for the RBF kernel it is σ), *idRef* – a reference to the test, *te12M* – a percentage average difference in MSE for testing data, if greater than 0 than δ-SVR is better, *teT* – t value for the t-test for comparing testing error, *s1* – the average number of support vectors for ε-SVR, *s2* – the average number of support vectors for δ-SVR, *sT* – t value for the t-test for comparing the number of support vectors

(a)

id	dn	ker
0	abalone	denseLinear 0.0
1	abalone	densePolynomial 5.0
2	abalone	denseRBF 0.125
3	cadata	denseLinear 0.0
4	cadata	densePolynomial 5.0
5	cadata	denseRBF 0.125
6	djia	denseLinear 0.0
7	djia	densePolynomial 5.0
8	djia	denseRBF 0.1
9	housing	denseLinear 0.0
10	housing	densePolynomial 5.0
11	housing	denseRBF 0.077

(b)

idRef	te12M	teT	s1	s2	sT
0	−0.48	−0.74	93	89	0.66
1	−0.57	−0.22	103	94	1.54
2	0.09	0.11	123	120	0.73
3	0.26	0.25	102	96	0.95
4	−2.14	−0.21	105	99	1.05
5	0.03	0.03	148	147	0.3
6	0.18	0.04	62	50	0.72
7	−10.72	−1.19	70	58	0.95
8	0.56	0.31	49	60	−0.9
9	0.4	0.2	92	87	0.67
10	−0.46	−0.11	124	122	0.46
11	0.13	0.11	175	179	−19.45

nonlinear kernels. For some of data sets δ-SVR is slower than ε-SVR.

For future work, we plan to test δ-SVR with different parameters for shifting the data up and down. We plan also to test δ-SVR for data sets with errors not only in

output, but also in input vectors.

Appendix A

A.1 The idea of a Set of Indicator Functions

The classification problem could be defined in terms of minimizing the risk function, [16]

$$R(\alpha) = \int L(c, \phi(x, \alpha))\, dF(c, x) \quad , \tag{A.1}$$

where L is *a loss function* defined as

$$L(c, \phi) = \begin{cases} 0 \; if \; c = \phi \\ 1 \; if \; c \neq \phi \end{cases} . \tag{A.2}$$

The regression problem could be defined as minimizing the risk function

$$R(\alpha) = \int (y - f(x, \alpha))^2\, dF(y, x) \quad . \tag{A.3}$$

Vapnik estimated the rate of uniform convergence for the set of bounded functions $A \leq Q(z, \alpha) \leq B$ as following

$$P\left\{ \sup_{\alpha \in A} \left(\int Q(z, \alpha)\, dF(z) - \frac{1}{n} \sum_{i=1}^{n} Q(z_i, \alpha) \right) > \varepsilon \right\} \tag{A.4}$$

$$\leq P\left\{ \sup_{\alpha \in A, \beta \in B} \left(\int \Phi(Q(z, \alpha) - \beta)\, dF(z) - \frac{1}{n} \sum_{i=1}^{n} \Phi(Q(z_i, \alpha) - \beta) \right) > \frac{\varepsilon}{B - A} \right\} . \tag{A.5}$$

He proposed capacity concepts for regression estimation by introducing the set of indicator functions for a real-valued function in the following way. Let $Q(z, \alpha^*)$ be a real-valued function. *The set of indicators* for this function is defined as

$$\phi(Q(z, \alpha^*) - \beta) \quad , \tag{A.6}$$

where

$$\beta \in \left(\inf_z Q(z, \alpha^*), \sup_z Q(z, \alpha^*) \right) \quad . \tag{A.7}$$

The ϕ is 1 when $Q(z, \alpha^*) - \beta$ is greater than 0, otherwise it is 0. *The complete set of indicators* for a set of real-valued functions $Q(z, \alpha^*)$, where $\alpha \in A$ is defined as

$$\phi(Q(z, \alpha) - \beta) \quad , \tag{A.8}$$

where $\alpha \in A$ and

$$\beta \in B = \left(\inf_{z,\alpha} Q(z, \alpha), \sup_{z,\alpha} Q(z, \alpha) \right) \quad . \tag{A.9}$$

The concept of a VC dimension for a set of real-valued functions is defined as the maximal number h of vectors $z_1, \ldots, z_h$ that can be shattered by the complete set of indicators $\phi(Q(z, \alpha^*) - \beta)$, $a \in A$, $\beta \in B$. For example, a VC dimension of a set of linear functions is the same for classification and regression, that is for a set of functions

$$f(z, \alpha) = \sum_{i=1}^{n} \alpha_i \phi_i(z) + \alpha_0 \quad , \tag{A.10}$$

a VC dimension is equal to $n + 1$, the same as for a set of indicator functions

$$f(z, \alpha) = \phi \left(\sum_{i=1}^{n} \alpha_i \phi_i(z) + \alpha_0 \right) \quad , \tag{A.11}$$

because the complete set of indicators coincides with the set of linear indicator functions. For bounded functions with bounds $A = 0, B = 1$, the bounds on the risk for bounded real-valued functions coincide with the bounds on the risk for indicator functions. From the conceptual point of view, the problem of minimizing the risk for indicator functions is equivalent to the problem of minimizing a risk for real-valued bounded functions.

A.2 A Proof of Thm. 2.2.1

Proof. Original data are distributed according to the probability distribution $\mathrm{F_r}(\vec{x_\mathrm{r}} \mid y_\mathrm{r})$. The expected value is equal to the mode for unimodal and symmetrical distributions

$$\mathbb{E}[y_\mathrm{r} \mid \vec{x_\mathrm{r}}] \equiv \mathrm{M}(y_\mathrm{r} \mid \vec{x_\mathrm{r}}) \quad . \tag{A.12}$$

First, we create joint random variable $(\vec{x_\mathrm{r}}, y_\mathrm{r})$ and we have

$$\mathrm{F_r}(\vec{x_\mathrm{r}}, y_\mathrm{r}) \equiv \mathrm{F_r}(y_\mathrm{r} \mid \vec{x_\mathrm{r}}) F(\vec{x_\mathrm{r}}) \quad . \tag{A.13}$$

After the transformation we define two new random variables $(\vec{x_\mathrm{c}} \mid 1)$ and $(\vec{x_\mathrm{c}} \mid -1)$. The optimal classification decision boundary contains points for which

$$\Pr(1 \mid \vec{x_\mathrm{c}}) \equiv \Pr(-1 \mid \vec{x_\mathrm{c}}) \quad . \tag{A.14}$$

We can rewrite it as

$$\mathrm{F}(\vec{x_\mathrm{c}} \mid 1) \Pr(1) = \mathrm{F}(\vec{x_\mathrm{c}} \mid -1) \Pr(-1) \quad . \tag{A.15}$$

Both classes have the same number of examples so

$$\Pr(1) = \Pr(-1) \quad , \tag{A.16}$$

so

$$\mathrm{F}(\vec{x_{\mathrm{c}}} \mid 1) = \mathrm{F}(\vec{x_{\mathrm{c}}} \mid -1) \quad . \tag{A.17}$$

Because both distributions are symmetrical and unimodal the above holds for

$$\frac{\mathrm{M}(\vec{x_{\mathrm{c}}} \mid 1) + \mathrm{M}(\vec{x_{\mathrm{c}}} \mid -1)}{2.0} \tag{A.18}$$

and because of symmetrical translation we get

$$\frac{\mathrm{M}(\vec{x_{\mathrm{c}}} \mid 1) + \mathrm{M}(\vec{x_{\mathrm{c}}} \mid -1)}{2.0} \equiv \mathrm{M}(y_{\mathrm{r}} \mid \vec{x_{\mathrm{r}}}) \equiv \mathbb{E}[y_{\mathrm{r}} \mid \vec{x_{\mathrm{r}}}] \quad . \tag{A.19}$$

□

A.3 A Proof of Thm. 2.2.2

Proof. Consider the distribution $\mathrm{F}_{\mathrm{r}}(y_{\mathrm{r}} \mid \vec{x_{\mathrm{r}}})$. For asymmetric distributions the mean could be different from the mode. Let's assume that the mode is equal to 0, and assume that the mean is equal to some value $m \geq 0$. So we need to prove that there exists δ such that

$$f(x + \delta) - f(x - \delta) = 0 \quad , \tag{A.20}$$

where

$$-\delta \leq x \leq \delta \tag{A.21}$$

for

$$x = m \quad . \tag{A.22}$$

So

$$f(m + \delta) - f(m - \delta) = 0 \quad , \tag{A.23}$$

where

$$0 \leq m \leq \delta \quad . \tag{A.24}$$

When $\delta = m$ then

$$f(2m) - f(0) \leq 0 \quad , \tag{A.25}$$

if for some value $\delta > m$,

$$f(m + \delta) - f(m - \delta) \geq 0 \quad , \tag{A.26}$$

then from intermediate value theorem there exists the δ.

□

A.4 Solution for (2.64)

The following holds

$$p^2 \left(R + \Delta\delta\right)^2 < R^2 \ , \tag{A.27}$$

$$|p| \left(R + \Delta\delta\right) < R \ , \tag{A.28}$$

$$p \left(R + \Delta\delta\right) < R \ \text{ and } \ p \left(R + \Delta\delta\right) > -R \ . \tag{A.29}$$

For $p > 0$, first inequality from (A.29) becomes

$$\frac{R + \Delta\delta}{1 + w_{\mathrm{c}}^{m+1} \Delta\delta} < R \ , \tag{A.30}$$

$$w_{\mathrm{c}}^{m+1} > \frac{1}{R} \ . \tag{A.31}$$

For $p < 0$, second inequality from (A.29) becomes

$$\frac{R + \Delta\delta}{1 + w_{\mathrm{c}}^{m+1} \Delta\delta} > -R \ , \tag{A.32}$$

$$R + \Delta\delta < - \left(1 + w_{\mathrm{c}}^{m+1} \Delta\delta\right) R \ , \tag{A.33}$$

$$2R + \Delta\delta < -w_{\mathrm{c}}^{m+1} \Delta\delta R \ , \tag{A.34}$$

$$w_{\mathrm{c}}^{m+1} < \frac{-2R - \Delta\delta}{\Delta\delta R} \ , \tag{A.35}$$

$$w_{\mathrm{c}}^{m+1} < \frac{-2}{\Delta\delta} - \frac{1}{R} \ . \tag{A.36}$$

References

[1] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm. 21

[2] Sami M. Halawani, Ibrahim A.Albidewi, and Amir Ahmad. A novel ensemble method for regression via classification problems. *Journal of Computer Science*, 7:387–393, 2011. 5

[3] Sabine Van Huffel and Joos Vandewalle. *The Total Least Squares Problem: Computational Aspects and Analysis*. Society for Industrial and Applied Mathematics, 1991. 20

[4] Nitin Indurkhya and Sholom M. Weiss. Solving regression problems with rule-based ensemble classifiers. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '01, pages 287–292, New York, NY, USA, 2001. ACM. ISBN 1-58113-391-X. 5

[5] Vojislav Kecman and Tao Yang. Adaptive local hyperplane for regression tasks. In *Proceedings of the 2009 international joint conference on Neural Networks*, IJCNN'09, pages 2371–2375, Piscataway, NJ, USA, 2009. IEEE Press. ISBN 978-1-4244-3549-4. 5

[6] LibSVM data sets. Libsvm data sets. www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/, 06 2011. 21

[7] Fuming Lin and Jun Guo. A novel support vector machine algorithm for solving nonlinear regression problems based on symmetrical points. In *Proceedings of the 2010 2nd International Conference on Computer Engineering and Technology (ICCET)*, pages 176–180, 2010. 5, 6

[8] Ivan Markovsky and Sabine Van Huffel. Overview of total least-squares methods. *Signal Processing*, 87(10):2283 – 2302, 2007. ISSN 0165-1684. Special Section: Total Least Squares and Errors-in-Variables Modeling. 20

[9] Marcin Orchel. Incorporating detractors into svm classification. In Krzysztof Cyran, Stanislaw Kozielski, James Peters, Urszula Stańczyk, and Alicja Wakulicz-Deja,

editors, *Man-Machine Interactions*, volume 59 of *Advances in Intelligent and Soft Computing*, pages 361–369. Springer Berlin / Heidelberg, 2009. ISBN 978-3-642-00562-6. doi: 10.1007/978-3-642-00563-3_38. URL http://dx.doi.org/10.1007/978-3-642-00563-3_38. 6, 21

[10] Marcin Orchel. Paper id 55. In *Submitted to European Conference of Machine Learning, ECML 2010*, April 2010. 5

[11] Marcin Orchel. Regression based on support vector classification. In Andrej Dobnikar, Uroš Lotric, and Branko Šter, editors, *Adaptive and Natural Computing Algorithms*, volume 6594 of *Lecture Notes in Computer Science*, pages 353–362. Springer Berlin / Heidelberg, 2011. ISBN 978-3-642-20266-7. doi: 10.1007/978-3-642-20267-4_37. URL http://dx.doi.org/10.1007/978-3-642-20267-4_37. 5, 6, 8

[12] Marcin Orchel. Support vector regression as a classification problem with a priori knowledge in the form of detractors. In Tadeusz Czachorski, Stanislaw Kozielski, and Urszula Stańczyk, editors, *Man-Machine Interactions 2*, volume 103 of *Advances in Intelligent and Soft Computing*, pages 353–362. Springer Berlin / Heidelberg, 2011. ISBN 978-3-642-23168-1. doi: 10.1007/978-3-642-23169-8_38. URL http://dx.doi.org/10.1007/978-3-642-23169-8_38. 6

[13] Marcin Orchel. Support vector regression based on data shifting. *Neurocomputing*, 96:2–11, 2012. 5, 6, 22, 27

[14] Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001. ISBN 0262194759. 19

[15] Vladimir N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995. ISBN 0-387-94559-8. 5, 14

[16] Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, September 1998. ISBN 0471030031. 5, 14, 18, 30

[17] Chang-An Wu and Hong-Bing Liu. An improved support vector regression based on classification. In *Proceedings of the 2007 International Conference on Multimedia and Ubiquitous Engineering*, MUE '07, pages 999–1003, Washington, DC, USA, 2007. IEEE Computer Society. ISBN 0-7695-2777-9. 5

[18] Tao Yang, Vojislav Kecman, Longbing Cao, and Chengqi Zhang. Testing adaptive local hyperplane for multi-class classification by double cross-validation. In *IJCNN*, pages 1–5, 2010. 21

Notation and Symbols

Miscellaneous

$|A|$ the cardinality of a finite set A, i.e., the number of elements in the set A.

$\cdot$ Dot product of two vectors, sometimes it is written with additional parentheses, for example for two vectors: $\vec{u}$ and $\vec{v}$, the dot product is $\vec{u} \cdot \vec{v}$ or $(\vec{u} \cdot \vec{v})$.

$\vec{v} \geq \vec{w}$ For two n dimensional vectors $\vec{v}$ and $\vec{w}$, it means that for all $i = 1...n$ $v_i \geq w_i$.

$\vec{v} \gg \vec{w}$ For two n dimensional vectors $\vec{v}$ and $\vec{w}$, it means that for all $i = 1...n$ $v_i > w_i$.

$\rho(A)$ the rank of a matrix A.

$\vec{w}^i_{\mathbf{r}}$ When a vector has an index in the subscript, the coefficient index is placed in the superscript, the example means the i-th coefficient of the $\vec{w}_{\mathrm{r}}$.

Optimization theory

* an asterisk as a superscript in optimization theory denotes a solution of the optimization problem.

Regression Based on Binary Classification

$CEMV$ a configuration of essential margin vectors.

EMV a set of essential margin vectors.

Abbreviations

δ-SVR δ support vector regression.

ε-SVR ε-insensitive support vector regression.

CM capacity minimization.

DJIA Dow Jones Industrial Average.

ERM empirical risk minimization.

LS least squares.

MSE mean squared error.

RBF radial basis function.

RMSE root mean squared error.

SMO sequential minimal optimization.

SRM structural risk minimization.

SVC support vector classification.

SVM support vector machines.

SVR support vector regression.

TLS total least squares.

VC Vapnik-Chervonenkis.