AGH University of Science and Technology

Faculty of Electrical Engineering, Automatics, Computer Science and Electronics

Department of Computer Science

Doctoral Thesis

Marcin Orchel

Incorporating Prior Knowledge into SVM Algorithms in Analysis of Multidimensional Data

Supervisor:
prof. dr hab. Witold Dzwinel

Kraków 2012

Abstract

In this thesis, we present results for research conducted by the author regarding the regression method, called δ support vector regression (δ-SVR), the method of incorporating knowledge about margin per example, called φ support vector classification (φ-SVC), implementation of support vector machines (SVM) and application of SVM to executing stock orders. In this thesis, we propose a method, called δ-SVR which replaces a regression problem with binary classification problems which are solved by 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, statistically significant fewer number of 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. In this thesis, we propose a concept of margin knowledge per example and a method called φ-SVC for incorporating this knowledge for classification and regression problems. We use φ-SVC for lowering generalization error for reduced models. In consequence, we get models with the decreased number of support vectors, with smaller generalization error compared to models without the margin knowledge. The method was tested for SVM classifier, ε-SVR and δ-SVR. Experiments on real world data sets show smaller generalization error for reduced models with margin knowledge. In this thesis, we propose two implementation improvements, the first one for speed of training of SVM, the second one for simplifying implementation of SVM solver. The first improvement, called heuristic of alternatives (HoA), regards a new heuristic for choosing parameters to an active set. It checks not only fulfillment of Karush-Kuhn-Tucker (KKT) conditions, but also growth of an objective function. Tests on real world data sets show, that HoA leads to decreased time of training of SVM, compared to the standard heuristic. The second improvement, called Sequential Multidimensional Subsolver (SMS), regards a new way of solving subproblems with more than two parameters, instead of using complicated quadratic programming solvers, we use heuristic approach solving subproblems with two parameters. We achieve simpler implementation with similar speed performance. In this thesis, we propose an application of support vector regression (SVR) for executing orders on stock markets. We use SVR for predicting a function of volume participation. We propose improvement of predicting participation function by incorporating additional constraints to SVM optimization problem using modified kernels and φ-SVC. We show that quality of the prediction has influence on stability of the method which achieves theoretically the price of execution called volume-weighted average price (VWAP). Moreover, we show how we can include results from predictive models for stock prices, achieving the price of execution better than VWAP. We compared ε-SVR and δ-SVR with simple predictors such as the average price of execution from previous days. The tests were performed on data for stocks from NASDAQ-100 index. For both methods we achieved smaller variance of execution costs. Moreover, we decreased costs of order execution by using prediction of stock prices.

Streszczenie

W tej pracy przedstawiamy wyniki badań przeprowadzonych przez autora dotyczące metody regresji, zwanej δ-SVR, metody włączania wiedzy o marginesie per przykład, zwanej φ-SVC, implementacji SVM oraz zastosowania SVM do składania zleceń giełdowych. Poniżej zamieszczamy streszczenie badań. W tej pracy zamieszczamy opis niedawno zaproponowanej przez autora nowoczesnej metody regresji, zwanej δ-SVR. W tym raporcie, proponujemy metodę, zwaną δ-SVR, która polega na zamianie problemu regresji w problemy binarnej klasyfikacji, które są rozwiązywane za pomocą SVM. Analizujemy statystyczną równoważność problemu regresji z problemem binarnej klasyfikacji. Pokazujemy potencjalną możliwość ulepszenia ograniczeń błędu generalizacji opartych na wymiarze VC, w porównaniu do SVM. Wykonaliśmy eksperymenty porównujące δ-SVR z ε-SVR na rzeczywistych zbiorach danych. Rezultaty wskazują, ze δ-SVR osiąga podobny błąd generalizacji, statystycznie istotnie mniejszą liczbę wektorów wspierających oraz mniejszy błąd generalizacji dla różnych wartości ε i δ. Metoda δ-SVR jest szybsza dla liniowych jąder używając metody SMO, dla nieliniowych jąder rezultaty szybkości zależą od zbioru danych. W tej pracy zamieszczamy opis niedawno zaproponowanej przez autora koncepcji wiedzy marginesowej per przykład włączonej do SVM dla problemów klasyfikacji i regresji. W tym raporcie proponujemy koncepcję wiedzy marginesowej per przykład oraz metodę zwaną φ-SVC do włączania tej wiedzy do problemów klasyfikacji i regresji. Używamy φ-SVC do zmniejszenia błędu generalizacji dla modeli zredukowanych. W konsekwencji, otrzymujemy modele o mniejszej liczbie wektorów wspierających, z mniejszym błędem generalizacji w stosunku do modeli nie stosujących wiedzy marginesowej. Metoda została przetestowana dla klasyfikatora SVM, oraz dla metod regresji ε-SVR oraz δ-SVR. Eksperymenty na danych rzeczywistych pokazują mniejszy błąd generalizacji dla modeli zredukowanych z wiedzą marginesową. W tej pracy zamieszczamy opis niedawno zaproponowanych dwóch usprawnień, pierwsze w szybkości trenowania SVM, drugie w uproszczeniu implementacji SVM. W tym raporcie proponujemy dwa usprawnienia w implementacji SVM, pierwsze w szybkości trenowania SVM, drugie w uproszczeniu implementacji SVM. Pierwsze usprawnienie, zwane HoA, dotyczy nowej heurystyki wyboru parametrów do zbioru aktywnego. Bierze ona pod uwagę nie tylko spełnienie warunków KKT, ale rowniez zmianę wartości funkcji celu. Testy na rzeczywistych zbiorach danych pokazują, ze HoA prowadzi do zmniejszenia czasu trenowania SVM, porównując do heurystki bazowej. Drugie usprawnienie, zwane SMS, dotyczy nowego sposobu rozwiązywania podproblemów o więcej niż dwóch parametrach, zamiast stosowania skomplikowanych metod rozwiązujących problemy z zakresu programowania kwadratowego, używamy do tego celu podejście heurystyczne rozwiązujące podproblemy dwuparametrowe. Otrzymujemy prostszą implementację, z podobnymi wynikami szybkościowymi. W tej pracy zamieszczamy opis zastosowania SVR do wykonywania zleceń na rynkach akcyjnych. W tym raporcie proponujemy zastosowanie SVR do wykonywania zleceń na rynkach akcyjnych. Używamy SVR do predykcji funkcji partycypacji w wolumenie. Proponujemy ulepszenie przewidywania funkcji partycypacji za pomocą włączenia dodatkowych warunków do SVM używając w tym celu zmodyfikowanych jąder oraz metody φ-SVC. Pokazujemy, że jakość przewidywania wpływa na stabilność metody osiągającej teoretycznie cenę wykonania VWAP. Ponadto pokazujemy jak za pomocą φ-SVC możemy uwzględnić wyniki z modelu przewidującego ceny akcji osiągając cenę wykonania lepszą niz VWAP. Porównaliśmy ε-SVR i δ-SVR z prostymi predyktorami takimi jak średnia cena wykonania z poprzednich dni. Testy zostaly przeprowadzone na danych dla spółek z indeksu NASDAQ-100. Dla obu metod otrzymaliśmy mniejszą wariancję kosztów egzekucji zleceń. Ponadto, zmniejszyliśmy koszty wykonania

zleceń wykorzystując dodatkowo predykcję cen giełdowych.

To Isa

Preface

The thesis is devoted to SVM, the novel method of machine learning that effectively finds dependencies in data. In recent times, SVM have become the new discipline that covers not only the machine learning theory, but also the optimization theory, the multi-dimensional geometry and the theory of linear functions. The basic problems of machine learning are classification and regression, for both problems effective variants of SVM were invented. Many extensions were also created, in particular allowing incorporation of prior knowledge.

The research covered by the thesis led to the development of the new regression method, which solves regression problems by binary classification. The method and the concept standing behind it shed some light on the relation between the two most important problems in machine learning, classification and regression. The second important part of the research was the analysis of the prior knowledge in the form of margin width per example. It is a natural extension of SVM, because the margin is the basic concept of SVM. The third element of the research was the analysis of implementation of SVM and application to finance engineering.

This thesis became possible due to the support of prof. Witold Dzwinel from Department of Computer Science on AGH University of Science and Technology. It was inspired by collaboration with my colleagues Marcin Kurdziel, Tomasz Arodź, Witold Dzwinel.

I discussed the ideas presented in the thesis with prof. Vojislav Kecman.

I would like to thank Isa, my parents and my sister. Thanks to Ryszard Zięba for showing me a beauty of math at the secondary school.

I would like to express my deep gratitude to all of them.

The research was financed by

- the Ministry of Science and Higher Education (in Poland), project No. NN519579338,
- internal Department of Computer Science on AGH University of Science and Technology grant and the project co-financed by EU and the Ministry of Science and Higher Education (in Poland), nr UDA-POKL.04.01.01-00-367/08-00 entitled "Doskonalenie i Rozwój Potencjału Dydaktycznego Kierunku Informatyka w AGH",
- the Ministry of Education and Science (in Poland), Project No. 3 T11F 010 30.

Marcin Orchel

Contents

Chapter I

Introduction

In recent years, SVM have become popular due to excellent both theoretical and practical results [44, 45]. They are successfully used for solving classification, regression and many others machine learning problems. The SVM were widely used in various domains, such as: text classification [13, 40], biotechnology [10, 21], economy [53], chemistry [4, 21], physics [38] and many others. They are popular learning methods due to mainly good generalization and fast training. Moreover, due to: solid basis in statistical learning theory, returning sparse, nonlinear solutions, resistance to outliers, geometric interpretation, formulation as convex quadratic optimization problems.

Two most popular problems in machine learning are classification and regression. Regression methods can be easily used for classification problems. This implies possibility to use power of regression methods for classification problems. Is it possible to use classification methods for regression problems? We will try to answer this question in this thesis.

The second main topic of this thesis is incorporating prior knowledge to SVM. A survey on research in this topic is in [19, 20]. Prior knowledge can lead to decrease of generalization error. It was incorporated to SVM for many real world problems, such as image retrieval [46], DNA promoter recognition [6], breast cancer prognosis [6].

I.1 Overview

The SVM are machine learning methods used mainly for solving classification and regression problems. They were developed by Vapnik [44, 45] in 1990s. They become popular up to now due to excellent both theoretical and practical results. Performance of machine learning methods is evaluated mainly based on the following criteria: generalization, speed, sparsity of the solution, and ability to incorporate prior knowledge.

Exploiting the relation between classification and regression is important for improving performance of existing methods. Moreover, the relation between classification and regression is an important element of machine learning theory. Vapnik in [44] derived generalization bounds for regression problems by using the concept of replacing a regression function with a set of indicator functions. Based on this idea, the methods which solve regression problems as multi-class classification were developed, e.g. [16], Fig. I.1(c), Fig. I.2(c). Another concept of transforming regression problems into classification were proposed in [24] and independently by the author in [31, 33]. The idea is to duplicate examples and move the original examples up, and the duplicated down, Fig. I.1(b), Fig. I.2(b). Based on this idea, we developed a novel regression method, called δ-SVR which possesses all advantages of SVM, and for some aspects it is better. Additionally, one of the practical advantages of δ-SVR is the ability to use it with any classifier based on kernel functions. This implicates broad possibilities of immediate application of any modifications and improvements of classification methods, directly for regression problems.

One of the possibilities to improve generalization performance is to incorporate additional knowledge to the problem, sometimes called *prior knowledge*. Various types of prior knowledge have been already incorporated to SVM. In [19], authors distinguish two types of prior knowl-

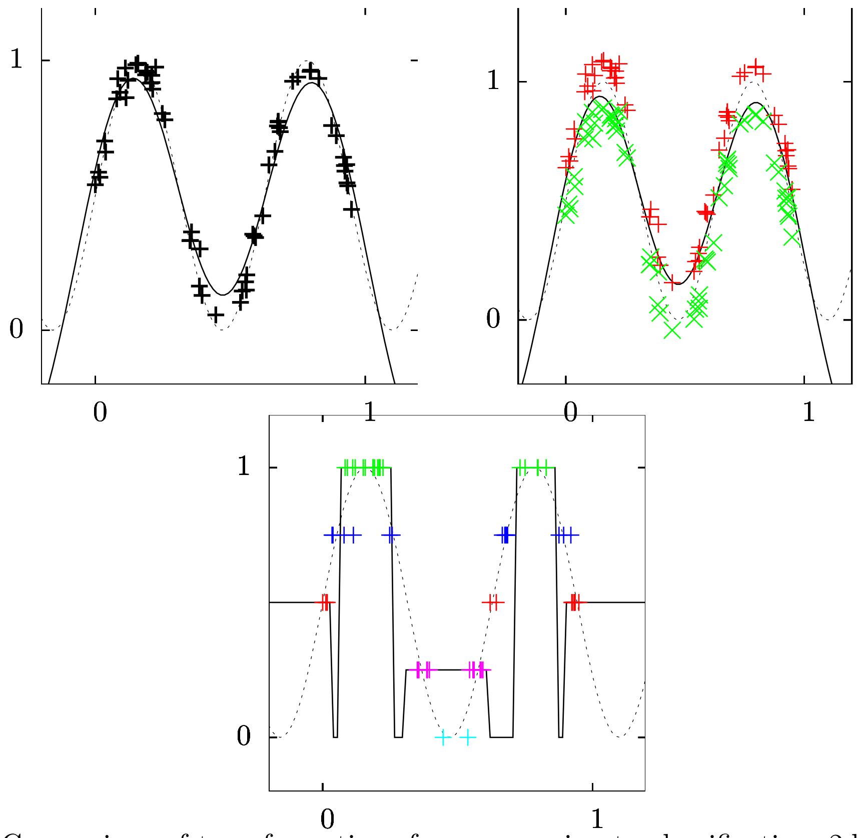

Figure I.1: Comparison of transformations from regression to classification, 2d. In the figures, there are solutions (solid lines), original functions from which the points were generated (dotted lines). In the left figure, there are a solution of ε-SVR and regression data points, in the center figure, there are a solution of δ-SVR and classification data points after transformation, in the right figure, there are a solution of regression by multiclass classification and classification data points after transformation

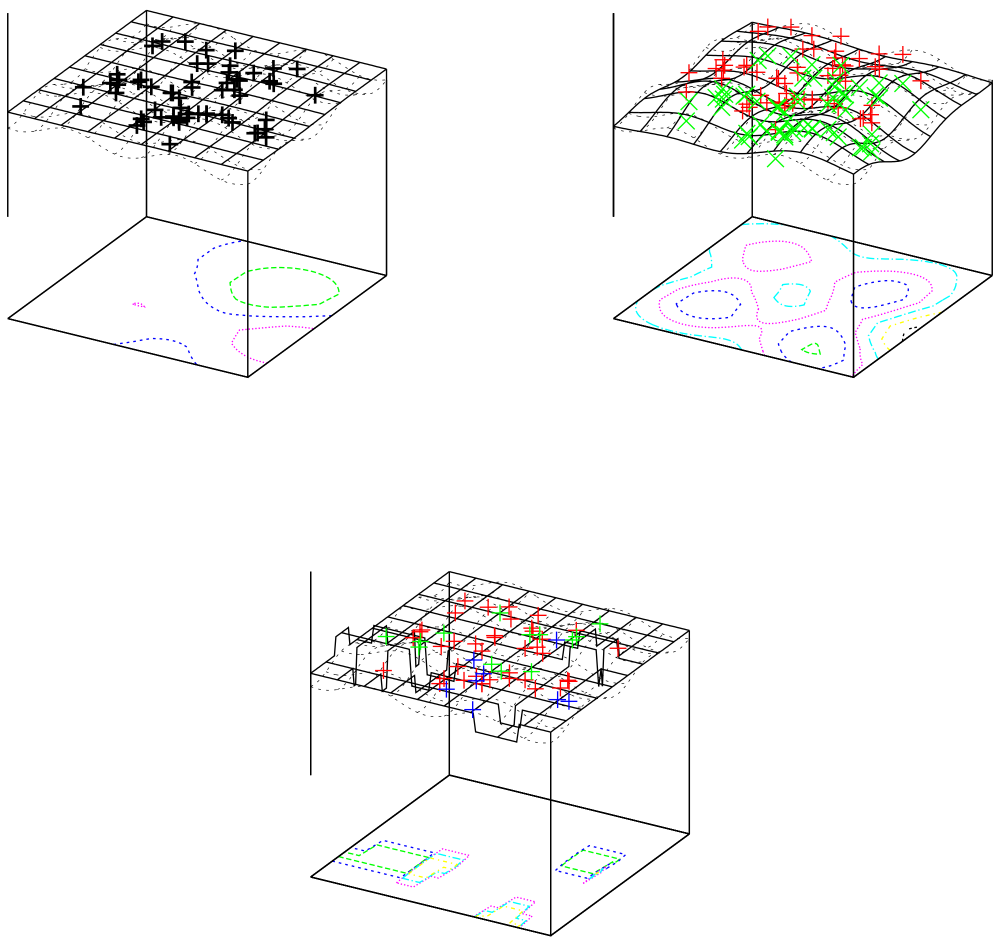

Figure I.2: Comparison of transformations from regression to classification 3d. In the figures, there are solutions (solid lines), original functions from which the points were generated (dotted lines). In the left figure, there are a solution of ε-SVR and regression data points, in the center figure, there are a solution of δ-SVR and classification data points after transformation, in the right figure, there are a solution of regression by multiclass classification and classification data points after transformation

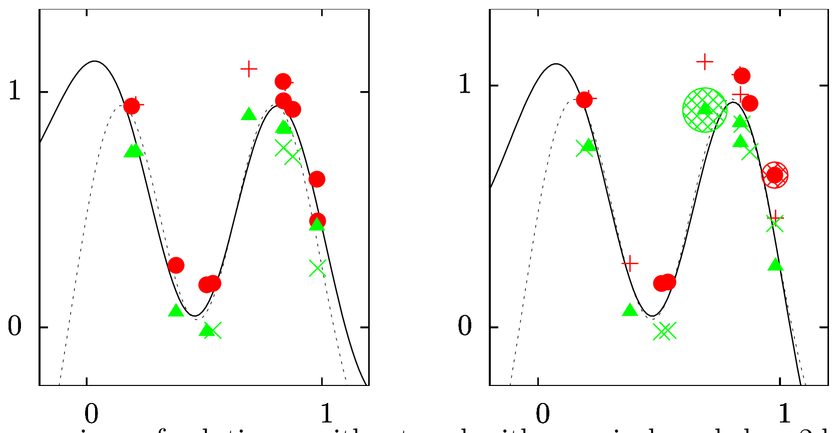

Figure I.3: Comparison of solutions: without and with margin knowledge, 2d. In the figures, there are example points, support vectors (triangles and circles), solutions (solid lines), original functions from which the points were generated (dotted lines). In the right figure, there is margin knowledge (circles filled with grid pattern)

edge: knowledge about class invariance, and knowledge about the data. The first type includes e.g. knowledge about classification in regions of the input space [6, 5, 26], knowledge about class invariance during transformations of the input. The second type includes e.g. knowledge about unlabeled examples, imbalance of classes, quality of the data. There exists different ways of incorporating prior knowledge to SVM, depended on the type of prior knowledge. We can distinguish three basic ways:

1. modification of input data such as a set of features, values of input parameters,
2. modification of the SVM method,
3. modification of output.

The second method leads to modification of the SVM optimization problem, particularly modification of the cost function, changing feasible region or modification of the kernel function.

In this thesis, we analyze recently proposed by the author margin knowledge per example [30, 32, 34]. It is closely related to formulation of SVM optimization problem. This knowledge can be interpreted in simplification as regions in the form of hypersphere's neighborhoods of examples with variable radii dependent on the margin width, mapped to some class, Fig. I.3, Fig. I.4. Regions that have been already incorporated to SVM are polyhedral regions [6, 5, 18, 48], ellipsoidal regions including spheroidal regions [37] and nonlinear regions [26]. Margin knowledge is incorporated to SVM by using generalization of standard SVM optimization problem. The similar idea was used for defining γ-shattering in statistical learning theory. We show that ε-SVR can be treated as a classification problem with margin knowledge, [34]. The main application of margin knowledge proposed in this thesis is to decrease the complication of solutions, what implicates the decreased number of support vectors. Simpler solution means simpler interpretation of the problem, and decreased time of testing new examples. The second important application of margin knowledge proposed in this thesis, is possibility to formulate other significant types of prior knowledge in this form. We show that knowledge in the form of sum of function values for some data can be effectively incorporated to the SVM problem by an additional constraint, and finally as margin knowledge.

It is important to incorporate knowledge in a soft way, so we can handle *imperfect prior knowledge* [28]. The incorporation of margin knowledge as additional parameters to the inequality constraints of SVM optimization problem makes possible to return solutions with a trade-off between performance of classifying data and fulfilling margin knowledge constraints.

In recent years, algorithmic trading becomes popular due to the progress in computer industry. Trades are automatically generated by the trading system and sent to order management system (OMS) which routes the orders to exchanges. One of the task of OMS is to efficiently

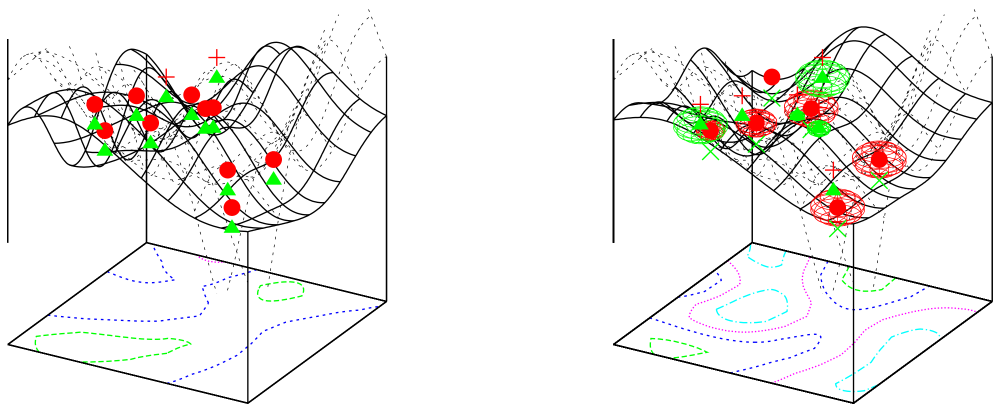

Figure I.4: Comparison of solutions: without and with margin knowledge, 3d. In the figures, there are example points, support vectors (triangles and circles), solutions (solid lines), original functions from which the points were generated (dotted lines). In the right figure, there is margin knowledge (circles filled with grid pattern)

divide the order into smaller parts and sent them during some time period. One of the method of assessing performance of order execution is to compare the price of execution with the other market participants. For this purpose we use the measure called VWAP. Recently, a simple theoretical model which achieves the ratio of VWAP equals to 1 has been proposed, [2]. In practice, achieving such results depends on quality of prediction of volume participation function. This prediction leads to a regression problem with additional constraints on the solution. For models which can achieve even better ratio, prediction of prices is required, Fig. I.5. Recently, we proposed using SVM with margin knowledge for the system predicting volume participation and additionally allowing to incorporate price prediction, [35].

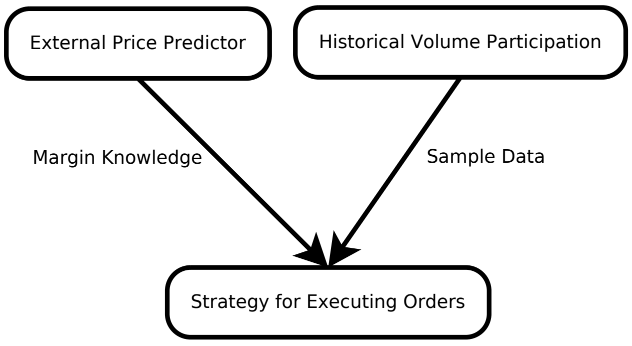
External Price Predictor
Historical Volume Participation
Margin Knowledge
Sample Data
Strategy for Executing Orders

Figure I.5: Data model for strategy of executing orders on exchanges. Strategy of executing orders uses predicted prices in the form of margin knowledge and historical volume participation in the form of examples

The hypotheses of the thesis are

1. The proposed regression method δ-SVR leads to the decreased number of support vectors and improved flexibility of SVM.

2. Margin knowledge per example leads to decreased generalization error while creating reduced models for classification and regression problems.

3. Proposed heuristic of alternatives HoA leads to increased speed of heuristic part of SMO algorithm.

4. Proposed sequential multidimensional subproblem solver SMS is a replacement for general libraries for quadratic programming for SVM algorithm.

5. SVM with prior knowledge leads to decreased generalization error for predicting function of volume participation in methods optimizing cost of executing orders on exchanges.

Roadmap. The detail description of the mentioned methods is in separate chapters. In the first chapter, we introduce shortly support vector classification (SVC), SVR, and prior knowledge incorporation to SVM. In the second chapter, we present δ-SVR. In the third chapter, we present φ-SVC. In the fourth chapter, we present improvements for implementation of SVM. In the fifth chapter, we present application of SVM to executing orders on exchanges.

I.2 SVC Optimization Problems

For a classification problem, we consider a set of n training vectors $\vec{x_i}$ for $i \in \{1, \ldots, n\}$, where $\vec{x_i} = (x_i^1, \ldots, x_i^m)$. The i-th training vector is mapped to $y_{\mathrm{c}}^i \in \{-1, 1\}$. The m is a dimension of the problem.

The SVC optimization problem for hard margin case with $\|\cdot\|_1$ norm Fig. I.6(a) is

OP 1.

$$\min_{\vec{w_{\mathrm{c}}}, b_{\mathrm{c}}} \quad f\left(\vec{w_{\mathrm{c}}}, b_{\mathrm{c}}\right) = \left\|\vec{w_{\mathrm{c}}}\right\|^2 \tag{I.1}$$

subject to

$$y_{\mathrm{c}}^i h\left(\vec{x_i}\right) \geq 1 \tag{I.2}$$

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

$$h\left(\vec{x_i}\right) = \vec{w_{\mathrm{c}}} \cdot \vec{x_i} + b_{\mathrm{c}} \quad . \tag{I.3}$$

The SVC soft margin case optimization problem with $\|\cdot\|_1$ norm Fig. I.6(b) is

OP 2.

$$\min_{\vec{w_{\mathrm{c}}}, b_{\mathrm{c}}, \vec{\xi_{\mathrm{c}}}} \quad f\left(\vec{w_{\mathrm{c}}}, b_{\mathrm{c}}, \vec{\xi_{\mathrm{c}}}\right) = \frac{1}{2}\left\|\vec{w_{\mathrm{c}}}\right\|^2 + C_{\mathrm{c}} \sum_{i=1}^{n} \xi_{\mathrm{c}}^i \tag{I.4}$$

subject to

$$y_{\mathrm{c}}^i h\left(\vec{x_i}\right) \geq 1 - \xi_{\mathrm{c}}^i \tag{I.5}$$

$$\vec{\xi_{\mathrm{c}}} \geq 0 \tag{I.6}$$

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

$$h\left(\vec{x_i}\right) = \vec{w_{\mathrm{c}}} \cdot \vec{x_i} + b_{\mathrm{c}} \quad . \tag{I.7}$$

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.

I.2.1 SVC Dual Optimization Problem

The OP 2 optimization problem after transformation to an equivalent dual optimization problem becomes

OP 3.

$$\max_{\vec{\alpha}} \quad f\left(\vec{\alpha}\right) = 1 \cdot \vec{\alpha} - \frac{1}{2}\vec{\alpha^T}\mathbf{Q}\vec{\alpha} \tag{I.8}$$

subject to

$$\vec{\alpha} \cdot \vec{y} = 0 \tag{I.9}$$

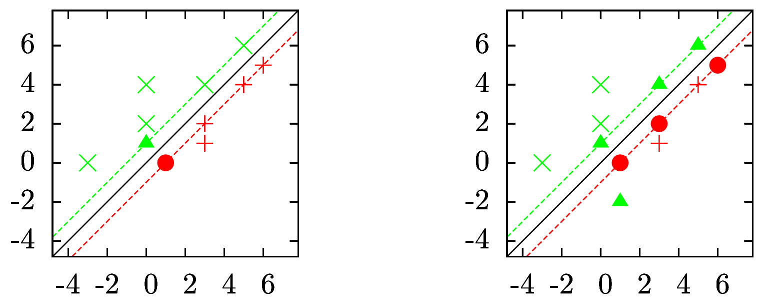

Figure I.6: Two types of margin classifiers: hard on the left, and soft on the right. In the figures, there are example points, support vectors (triangles and circles), solutions (solid lines), margin lines (dashed lines). In the right figure, we can see a misclassified point (1, -2)

$$0 \le \alpha_i \le C_c \tag{I.10}$$

where

$$Q_{ij} = y_i y_j K\left(\vec{x_i}, \vec{x_j}\right) \tag{I.11}$$

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

The decision curve is

$$h^*\left(\vec{x}\right) = \sum_{i=1}^{n} y_{\mathrm{c}}^i \alpha_i^* K\left(\vec{x_i}, \vec{x}\right) + b_{\mathrm{c}}^* = 0 \ , \tag{I.12}$$

where α_i are Lagrange multipliers of the dual problem, $K(\cdot, \cdot)$ is a kernel function which appears only in the dual problem. The most popular kernel functions are linear, polynomial, radial basis function (RBF) and sigmoid. A kernel function which is a dot product of its arguments we call *a simple linear kernel*. *Margin boundaries* are defined as the two hyperplanes $h\left(\vec{x}\right) = -1$ and $h\left(\vec{x}\right) = 1$. *Optimal margin boundaries* are defined as the two hyperplanes $h^*\left(\vec{x}\right) = -1$ and $h^*\left(\vec{x}\right) = 1$. The i-th training example is *a support vector*, when $\alpha_i^* \neq 0$. It can be proved that a set of support vectors contains all training examples lying below optimal margin boundaries ($y_{\mathrm{c}}^i h^*\left(\vec{x_i}\right) < 1$), and most of the examples lying exactly on the optimal margin boundaries ($y_{\mathrm{c}}^i h^*\left(\vec{x_i}\right) = 1$).

The KKT complementary condition for OP 2 is

$$\alpha_i \left(y_c^i h\left(\vec{x_i}\right) - 1 + \xi_c^i\right) = 0 \tag{I.13}$$

$$\left(C_c - \alpha_i\right) \xi_c^i = 0 \ . \tag{I.14}$$

We can find values of ξ_i parameters from the solution of the dual form as following. When

$$y_c^i h^*\left(\vec{x_i}\right) \ge 1 \ , \tag{I.15}$$

then $\xi_i = 0$, else

$$\xi_i = 1 - y_c^i h\left(\vec{x_i}\right) \ . \tag{I.16}$$

I.2.2 SVC Without the Offset

Another variant of SVC is the SVC without the offset b_c, analyzed recently in [41]. The optimization problem is the same except missing b_c term, for the soft case it is

OP 4.

$$\min_{\vec{w_{\mathrm{c}}}, \vec{\xi_c}} \ f\left(\vec{w_{\mathrm{c}}}, \vec{\xi_c}\right) = \frac{1}{2} \left\|\vec{w_{\mathrm{c}}}\right\|^2 + \vec{C_{\mathrm{c}}} \cdot \vec{\xi_{\mathrm{c}}} \tag{I.17}$$

subject to

$$y_{\mathrm{c}}^{i} h\left(\vec{x_i}\right) \geq 1 - \xi_{\mathrm{c}}^{i} + \varphi_i \tag{I.18}$$

$$\vec{\xi} \geq 0 \tag{I.19}$$

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

$$\vec{C}_{\mathrm{c}} \gg 0 \tag{I.20}$$

$$\varphi_{\mathrm{c}}^{i} \in \mathbb{R} \tag{I.21}$$

$$h\left(\vec{x_i}\right) = \vec{w}_{\mathrm{c}} \cdot \vec{x_i} \ . \tag{I.22}$$

The dual problem is

OP 5.

$$\max_{\vec{\alpha}} \quad d\left(\vec{\alpha}\right) = \vec{\alpha} \cdot \left(1 + \vec{\varphi}\right) - \frac{1}{2} \vec{\alpha}^{T} Q \vec{\alpha} \tag{I.23}$$

subject to

$$0 \leq \vec{\alpha} \leq \vec{C} \tag{I.24}$$

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

We can notice missing linear constraint. The decision curve is

$$h^{*}\left(\vec{x}\right) = \sum_{i=1}^{n} y_{\mathrm{c}}^{i} \alpha_i^{*} K\left(\vec{x_i}, \vec{x}\right) = 0 \ , \tag{I.25}$$

I.2.3 ν-SVC

Another variant of SVC is ν support vector classification (ν-SVC) where we replace C by $\nu \in [0, 1]$. The modified optimization problem is

OP 6.

$$\min_{\vec{w}, b, \vec{\xi}, p} \quad f\left(\vec{w}, b, \vec{\xi}, p\right) = \frac{1}{2} \left\|\vec{w}\right\|^2 - \nu p + \frac{1}{n} \sum_{i=1}^{n} \xi_c^i \tag{I.26}$$

subject to

$$y_i h\left(\vec{x_i}\right) \geq p - \xi_i \tag{I.27}$$

$$\vec{\xi} \geq 0 \tag{I.28}$$

$$p \geq 0 \tag{I.29}$$

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

$$\vec{C} \gg 0 \tag{I.30}$$

$$h\left(\vec{x_i}\right) = \vec{w} \cdot \vec{x_i} + b \ . \tag{I.31}$$

We can notice different cost function, the additional variable p and the additional constraint.

I.3 SVR Optimization Problems

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 $y_{\mathrm{r}}^{i} \in \mathbb{R}$. The m is a dimension of the problem. The ε-SVR soft case optimization problem is

OP 7.

$$\min_{\vec{w_{\mathrm{r}}}, b_{\mathrm{r}}, \vec{\xi_{\mathrm{r}}}, \vec{\xi_{\mathrm{r}}^{*}}} \quad f\left(\vec{w_{\mathrm{r}}}, b_{\mathrm{r}}, \vec{\xi_{\mathrm{r}}}, \vec{\xi_{\mathrm{r}}^{*}}\right) = \frac{1}{2} \left\|\vec{w_{\mathrm{r}}}\right\|^2 + C_{\mathrm{r}} \sum_{i=1}^{n} \left(\xi_{\mathrm{r}}^{i} + \xi_{\mathrm{r}}^{*i}\right) \tag{I.32}$$

subject to

$$y_{\mathrm{r}}^{i} - g\left(\vec{x_i}\right) \leq \varepsilon + \xi_{\mathrm{r}}^{i} \tag{I.33}$$

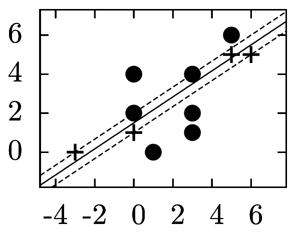

Figure I.7: The idea of ε-SVR. In the figure, there are examples, support vectors (circles), a solution (solid line), and ε boundaries (dashed lines)

$$g\left(\vec{x}_i\right) - y_{\mathrm{r}}^i \le \varepsilon + \xi_{\mathrm{r}}^{i*} \tag{I.34}$$

$$\vec{\xi}_{\mathrm{r}} \ge 0 \tag{I.35}$$

$$\vec{\xi}_{\mathrm{r}}^{*} \ge 0 \tag{I.36}$$

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

$$g\left(\vec{x}_i\right) = \vec{w}_{\mathrm{r}} \cdot \vec{x}_i + b_{\mathrm{r}} \ . \tag{I.37}$$

The $g^*\left(\vec{x}\right) = \vec{w}_{\mathrm{r}}^* \cdot \vec{x} + b_{\mathrm{r}}^*$ is a regression function. Optimization problem 7 is transformed to an equivalent dual problem. The regression function becomes

$$g^*\left(\vec{x}\right) = \sum_{i=1}^{n} \left(\alpha_i^* - \beta_i^*\right) K\left(\vec{x}_i, \vec{x}\right) + b_{\mathrm{r}}^* \ , \tag{I.38}$$

where α_i, β_i are Lagrange multipliers, $K(\cdot,\cdot)$ is a kernel function. The i-th training example is *a support vector*, when $\alpha_i^* - \beta_i^* \ne 0$. It can be proved that a set of support vectors contains all training examples lying outside ε boundaries, and most of the examples which lie exactly on ε boundaries. The number of support vectors can be controlled by ε parameter.

I.4 Incorporating Prior Knowledge to SVM

Various schemes of incorporating prior knowledge to SVM optimization problem have been already proposed. They belong to the following categories:

1. a modified cost function (either primal or dual),
2. modified constraints (either primal or dual),
3. new constraints (either primal or dual),

Besides listed categories, modification of input data, solution or a kernel is possible. Some incorporations requires many changes of an optimization problem, new variables or new parameters. A survey on incorporating prior knowledge to SVM is in [19, 20]. In this thesis, we are concentrated on knowledge per example, especially formulated in the form of additional parameters to an optimization problem for every example.

One of the type of weights per example are weights meaning different misclassification costs C_i per example. The special case is a different cost of wrong classification for negative and positive training examples C_+ and C_- used for incorporating knowledge about unbalanced data for C support vector machines (C-SVM) [13], for ν support vector machines (ν-SVM) [47]. The C_i weights were also used for fuzzy support vector machines [23], for denoting confidence of pseudo labels [46] and for denoting confidence dependent on quality of the data [51]. A 1-norm soft margin SVC optimization problem for training examples $\vec{x}_i$ with weights C_i is

OP 8.

$$\min_{\vec{w},b,\vec{\xi}} \; f\left(\vec{w}, b, \vec{\xi}\right) = \frac{1}{2}\left\|\vec{w}\right\|^2 + \vec{C}_c \cdot \vec{\xi} \tag{I.39}$$

subject to

$$y_i h\left(\vec{x}_i\right) \geq 1 - \xi_i \tag{I.40}$$

$$\vec{\xi} \geq 0 \tag{I.41}$$

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

$$\vec{C}_c \gg 0 \tag{I.42}$$

$$h\left(\vec{x}_i\right) = \vec{w} \cdot \vec{x}_i + b \ \ . \tag{I.43}$$

The C_i weights were also used with ε-SVR for predicting time series data [42]. The formulation is as follows:

OP 9.

$$\min_{\vec{w}_{\mathrm{r}},b_{\mathrm{r}},\vec{\xi}_{\mathrm{r}},\vec{\xi}^*_{\mathrm{r}}} \; f\left(\vec{w}_{\mathrm{r}}, b_{\mathrm{r}}, \vec{\xi}_{\mathrm{r}}, \vec{\xi}^*_{\mathrm{r}}\right) = \frac{1}{2}\left\|\vec{w}_{\mathrm{r}}\right\|^2 + \vec{C}_{\mathrm{r}} \sum_{i=1}^{n}\left(\xi^i_{\mathrm{r}} + \xi^{*i}_{\mathrm{r}}\right) \tag{I.44}$$

subject to

$$y^i_{\mathrm{r}} - g\left(\vec{x}_i\right) \leq \varepsilon + \xi^i_{\mathrm{r}} \tag{I.45}$$

$$g\left(\vec{x}_i\right) - y^i_{\mathrm{r}} \leq \varepsilon + \xi^{i*}_{\mathrm{r}} \tag{I.46}$$

$$\vec{\xi}_{\mathrm{r}} \geq 0 \tag{I.47}$$

$$\vec{\xi}^*_{\mathrm{r}} \geq 0 \tag{I.48}$$

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

$$\vec{C}_r \gg 0 \tag{I.49}$$

$$g\left(\vec{x}_i\right) = \vec{w}_{\mathrm{r}} \cdot \vec{x}_i + b_{\mathrm{r}} \ \ . \tag{I.50}$$

The other type of weights, which were used with ε-SVR, are ε_i weights per example replacing the parameter ε. They were used for density estimation [45]. The optimization problem with additional weights ε_i is

OP 10.

$$\min_{\vec{w}_{\mathrm{r}},b_{\mathrm{r}},\vec{\xi}_{\mathrm{r}},\vec{\xi}^*_{\mathrm{r}}} \; f\left(\vec{w}_{\mathrm{r}}, b_{\mathrm{r}}, \vec{\xi}_{\mathrm{r}}, \vec{\xi}^*_{\mathrm{r}}\right) = \frac{1}{2}\left\|\vec{w}_{\mathrm{r}}\right\|^2 + C_{\mathrm{r}} \sum_{i=1}^{n}\left(\xi^i_{\mathrm{r}} + \xi^{*i}_{\mathrm{r}}\right) \tag{I.51}$$

subject to

$$y^i_{\mathrm{r}} - g\left(\vec{x}_i\right) \leq \varepsilon_i + \xi^i_{\mathrm{r}} \tag{I.52}$$

$$g\left(\vec{x}_i\right) - y^i_{\mathrm{r}} \leq \varepsilon_i + \xi^{i*}_{\mathrm{r}} \tag{I.53}$$

$$\vec{\xi}_{\mathrm{r}} \geq 0 \tag{I.54}$$

$$\vec{\xi}^*_{\mathrm{r}} \geq 0 \tag{I.55}$$

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

$$g\left(\vec{x}_i\right) = \vec{w}_{\mathrm{r}} \cdot \vec{x}_i + b_{\mathrm{r}} \ \ . \tag{I.56}$$

Sometimes we use different ε weights for inequalities (I.52), (I.53), thus instead of ε_i weights we have only two weights ε_u and ε_d

OP 11.

$$\min_{\vec{w}_{\mathrm{r}},b_{\mathrm{r}},\vec{\xi}_{\mathrm{r}},\vec{\xi}^*_{\mathrm{r}}} \; f\left(\vec{w}_{\mathrm{r}}, b_{\mathrm{r}}, \vec{\xi}_{\mathrm{r}}, \vec{\xi}^*_{\mathrm{r}}\right) = \frac{1}{2}\left\|\vec{w}_{\mathrm{r}}\right\|^2 + C_{\mathrm{r}} \sum_{i=1}^{n}\left(\xi^i_{\mathrm{r}} + \xi^{*i}_{\mathrm{r}}\right) \tag{I.57}$$

subject to

$$y^i_{\mathrm{r}} - g\left(\vec{x}_i\right) \leq \varepsilon_u + \xi^i_{\mathrm{r}} \tag{I.58}$$

$$g\left(\vec{x}_i\right) - y^i_{\mathrm{r}} \leq \varepsilon_d + \xi^{i*}_{\mathrm{r}} \tag{I.59}$$

$$\vec{\xi}_{\mathrm{r}} \geq 0 \tag{I.60}$$

$$\vec{\xi}^*_{\mathrm{r}} \geq 0 \tag{I.61}$$

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

$$g\left(\vec{x_i}\right) = \vec{w}_{\mathrm{r}} \cdot \vec{x_i} + b_{\mathrm{r}} \quad . \tag{I.62}$$

And finally we can also use ε^i_u and ε^i_d weights. We can notice that changing y^i_r value by Δy^i_r is equivalent to changing ε^i_u by $-\Delta y^i_r$ and changing ε^i_d by Δy^i_r.

Chapter II

Regression Based on Binary Classification

Recently, an alternative regression method was proposed [33, 24], 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 possesses the same important advantages as ε-SVR: it leads to convex optimization problems, it generates sparse solutions, kernel functions can be used for generating nonlinear solutions. It was shown experimentally, that δ-SVR can achieve comparable or better generalization performance compared to ε-SVR [33]. It was also reported in [33] that some type of prior knowledge already incorporated to SVC can be directly used for regression problems. The δ-SVR has a potential to use a much broader type of modifications and improvements of SVC directly for regression problems without need of reinventing them for specialized regression methods like ε-SVR. In this thesis, we focus on analyzing some important properties of δ-SVR regarding its ability to generalize, realize structural risk minimization (SRM), and generate sparse solutions.

The topic of connection between classification and regression problems was investigated by Vapnik [44, 45]. Vapnik proposed generalization of capacity concepts introduced for classification to regression problems by describing regression functions as a complete set of indicators, see Appendix C.1. Based on this idea a method for solving regression problems as multiclass classification problems was proposed, [11, 16, 7]. The method uses discretization process to generate multiclass labels. Some attempts also were made to combine SVR with SVC [49]. The concept used in δ-SVR is different, 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 [24], it was investigated independently by the author and submitted to [31] and published in [33]. The main problem of the realization of the concept in [24] 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 use a golden section method. Moreover, two problems arise with this workaround: the solution might not exist, there could be more than one solution. In [33], authors 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 [33], the author proposed using the method to incorporate margin knowledge which was previously incorporated to SVC for classification problems in [30, 34], directly for regression problems. The author 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 specialized regression methods like ε-SVR which is a practice nowadays. In [24], authors 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 in this thesis were to analyze a general concept of representing regression problems as classification ones by duplicating and shifting data, to analyze

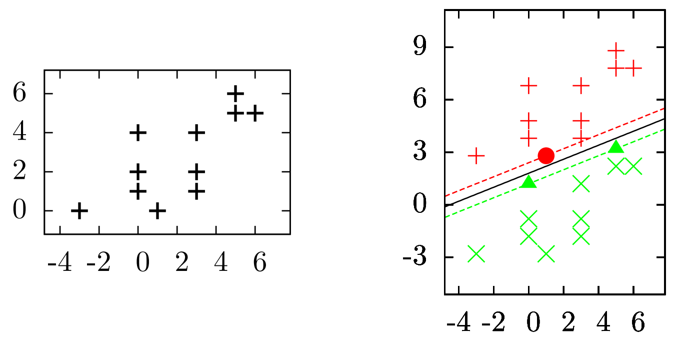

Figure II.1: The idea of the problem transformation in δ-SVR for 2d. In the left figure, there are regression example points. In the right figure, there are classification example points after transformation, support vectors (triangles and circles), solution (a solid line), margin lines (dashed lines)

potential generalization improvements of δ-SVR over SVM and to improve experiments conducted in [33]. The outline of the thesis is as follows. In the first section, we give an introduction to the idea of a set of indicator functions. In the second section, we give a theoretical analysis of the transformation. In the third section, we analyze generalization abilities of δ-SVR. In the fourth section, we present experiments on synthetic and real world data sets.

II.1 Introduction to δ-SVR

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

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. II.1, Fig. II.2.

The result of the first step is a set of training mappings for $i \in \{1, \ldots, 2n\}$

$$\begin{cases} \vec{b_i} = (x_i^1, \ldots, x_i^m) \to y_{\mathrm{r}}^i + \delta & \text{for } i \in \{1, \ldots, n\} \\ \vec{b_i} = (x_{i-n}^1, \ldots, x_{i-n}^m) \to y_{\mathrm{r}}^{i-n} - \delta & \text{for } i \in \{n+1, \ldots, 2n\} \end{cases} \tag{II.1}$$

for $\delta \geq 0$. The δ is called *the translation parameter*. The result of the second step is a set of training mappings for $i \in \{1, \ldots, 2n\}$

$$\begin{cases} \vec{c_i} = (b_i^1, \ldots, b_i^m, y_{\mathrm{r}}^i + \delta) \to 1 & \text{for } i \in \{1, \ldots, n\} \\ \vec{c_i} = (b_i^1, \ldots, b_i^m, y_{\mathrm{r}}^{i-n} - \delta) \to -1 & \text{for } i \in \{n+1, \ldots, 2n\} \end{cases} \tag{II.2}$$

for $\delta \geq 0$. The dimension of the $\vec{c_i}$ vectors is equal to $m+1$. The set of $\vec{x_i}$ mappings is called *a regression data setting*, the set of $\vec{c_i}$ ones is called *a classification data setting*. In the third step, OP 2 is solved with $\vec{c_i}$ examples. Note that $h^*(\vec{x})$ is in the implicit form of the last coordinate

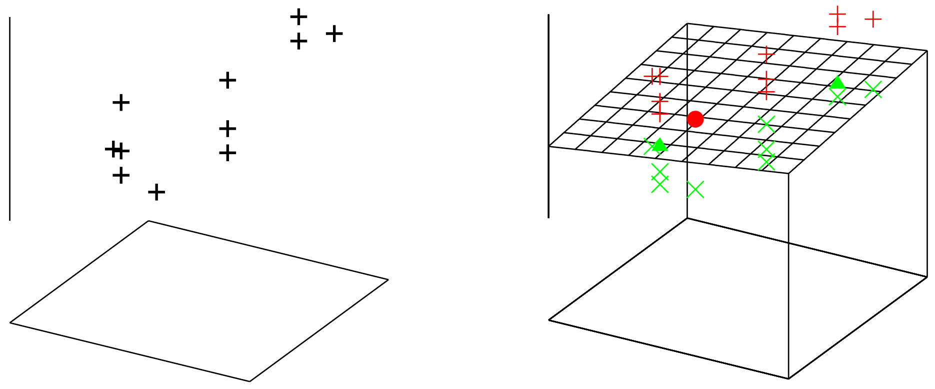

Figure II.2: The idea of the problem transformation in δ-SVR for 3d. In the left figure, there are regression example points. In the right figure, there are classification example points after transformation, support vectors (triangles and circles), solution (a plane)

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 $\vec{w_{\mathrm{c}}}$ variable of the primal problem for a simple linear kernel is found based on the solution of the dual problem in the following way

$$\vec{w_{\mathrm{c}}} = \sum_{i=1}^{2n} y_{\mathrm{c}}^i \alpha_i \vec{c_i} \ , \tag{II.3}$$

where $y_{\mathrm{c}}^i = 1$ for $i \in \{1, \ldots, n\}$ and $y_{\mathrm{c}}^i = -1$ for $i \in \{n+1, \ldots, 2n\}$. For a simple linear kernel the explicit form of (I.12) is

$$x_{m+1} = \frac{-\sum_{j=1}^{m} w_{\mathrm{c}}^j x_j - b_{\mathrm{c}}}{w_{\mathrm{c}}^{m+1}} \ . \tag{II.4}$$

The regression solution is $g^*(\vec{x}) = \vec{w_{\mathrm{r}}} \cdot \vec{x} + b_{\mathrm{r}}$, where $w_{\mathrm{r}}^i = -w_{\mathrm{c}}^i / w_{\mathrm{c}}^{m+1}$, $b_{\mathrm{r}} = -b_{\mathrm{c}} / w_{\mathrm{c}}^{m+1}$ 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 (e.g. 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 (e.g. a polynomial kernel with a dimension greater than 4), or it is not easy to find it and hence a special method for finding the explicit formula of the coordinate should be used, e.g. a bisection method. The disadvantage of this solution is a longer time of testing new examples. To overcome these problems, we propose 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 propose a new kernel type in which the last coordinate is placed only inside a linear term [33]. The new kernel is constructed from an original kernel by removing the last coordinate, and adding the linear term with the last coordinate. For the most popular kernels polynomial, RBF and sigmoid, the conversions are respectively

$$\left(\vec{x}\cdot\vec{y}\right)^d \quad\rightarrow\quad \left(\sum_{i=1}^{m} x_i y_i\right)^d + x_{m+1}y_{m+1} \quad , \tag{II.5}$$

$$\exp -\frac{\|\vec{x}-\vec{y}\|^2}{2\sigma^2} \quad\rightarrow\quad \exp -\frac{\sum_{i=1}^{m}\left(x_i-y_i\right)^2}{2\sigma^2} + x_{m+1}y_{m+1} \quad , \tag{II.6}$$

$$\tanh \vec{x}\vec{y} \quad\rightarrow\quad \tanh \sum_{i=1}^{m} x_i y_i + x_{m+1}y_{m+1} \quad , \tag{II.7}$$

where $\vec{x}$ and $\vec{y}$ are here $m+1$ dimensional vectors. The proposed method of constructing new kernels always generates a function fulfilling Mercer's condition, because it generates a function which is a sum of two kernels. For the new kernel type, the explicit form of (I.12) for δ-SVR is

$$x_{m+1} = \frac{-\sum_{i=1}^{2n} y_c^i \alpha_i K_o\left(\vec{b_i}, \vec{x_r}\right) - b_c}{\sum_{i=1}^{2n} y_c^i \alpha_i c_i^{m+1}} \quad , \tag{II.8}$$

where $\vec{x_r} = (x_1, \ldots, x_m)$, $K_o(\cdot,\cdot)$ is the original kernel from which the new one was constructed ((II.5),(II.6),(II.7)).

II.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 (II.8) as

$$x_{m+1} = \frac{-\sum_{i=1}^{n}\left(\alpha_i - \alpha_{n+i}\right) K_o\left(\vec{b_i}, \vec{x_r}\right) - b_c}{\sum_{i=1}^{2n} y_c^i \alpha_i c_i^{m+1}} \quad . \tag{II.9}$$

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.

II.1.2 Basic Comparison With ε-SVR

The general idea of δ-SVR is that instead of finding the best model on 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. We can write inequality constraints for δ-SVR as:

$$-\vec{w_c^r}\cdot\vec{x_i^r} - w_c^{m+1}\left(x_i^{m+1} - \delta\right) - b_c \geq 1 - \xi_c^i \tag{II.10}$$

$$\vec{w_c^r}\cdot\vec{x_i^r} + w_c^{m+1}\left(x_i^{m+1} + \delta\right) + b_c \geq 1 - \xi_c^{*i} \tag{II.11}$$

After reformulation:

$$-\vec{w_c^r}\cdot\vec{x_i^r} - b_c \geq w_c^{m+1}x_i^{m+1} - w_c^{m+1}\delta + 1 - \xi_c^i \tag{II.12}$$

$$\vec{w_c^r}\cdot\vec{x_i^r} + b_c \geq -w_c^{m+1}x_i^{m+1} - w_c^{m+1}\delta + 1 - \xi_c^{*i} \tag{II.13}$$

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

$$\frac{-\vec{w_c^r}\cdot\vec{x_i^r} - b_c}{w_c^{m+1}} \geq x_i^{m+1} - \delta + \frac{1}{w_c^{m+1}} - \frac{\xi_c^i}{w_c^{m+1}} \tag{II.14}$$

$$\frac{\vec{w_c^r} \cdot \vec{x_i^r} + b_c}{w_c^{m+1}} \geq -x_i^{m+1} - \delta + \frac{1}{w_c^{m+1}} - \frac{\xi_c^{*i}}{w_c^{m+1}} \tag{II.15}$$

After transforming to regression convention we get:

$$y_r^i - g\left(\vec{x_i}\right) \leq \delta - \frac{1}{w_c^{m+1}} + \frac{\xi_c^i}{w_c^{m+1}} \tag{II.16}$$

$$g\left(\vec{x_i}\right) - y_r^i \leq \delta - \frac{1}{w_c^{m+1}} + \frac{\xi_c^{*i}}{w_c^{m+1}} \tag{II.17}$$

After changing notation from w_c^{m+1} to v, we get the following optimization problem:

OP 12.

$$\min_{\vec{w_\mathrm{r}}, b_\mathrm{r}, \vec{\xi_\mathrm{r}}, \vec{\xi_\mathrm{r}^*}, v} \quad f\left(\vec{w_\mathrm{r}}, b_\mathrm{r}, \vec{\xi_\mathrm{r}}, \vec{\xi_\mathrm{r}^*}, v\right) = \left\|\vec{w_\mathrm{r}}, v\right\|^2 + C_\mathrm{r} \sum_{i=1}^{n} \left(\xi_\mathrm{r}^i + \xi_\mathrm{r}^{*i}\right) \tag{II.18}$$

subject to

$$\vec{\xi_\mathrm{r}} \geq 0 \tag{II.19}$$

$$\vec{\xi_\mathrm{r}^*} \geq 0 \tag{II.20}$$

and for $v > 0$

$$y_\mathrm{r}^i - g\left(\vec{x_i}\right) \leq \delta - \frac{1}{v} + \frac{\xi_r^i}{v} \tag{II.21}$$

$$g\left(\vec{x_i}\right) - y_r^i \leq \delta - \frac{1}{v} + \frac{\xi_r^{*i}}{v} \tag{II.22}$$

and for $v < 0$

$$y_\mathrm{r}^i - g\left(\vec{x_i}\right) \geq \delta - \frac{1}{v} + \frac{\xi_r^i}{v} \tag{II.23}$$

$$g\left(\vec{x_i}\right) - y_r^i \geq \delta - \frac{1}{v} + \frac{\xi_r^{*i}}{v} \tag{II.24}$$

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

$$g\left(\vec{x_i}\right) = \vec{w_\mathrm{r}} \cdot \vec{x_i} + b_\mathrm{r} \quad . \tag{II.25}$$

We can notice that when we have the solution of OP 12 and $v^* > 0$, the same solution can be found by OP 7 with the parameters set to

$$\varepsilon = \delta - \frac{1}{v^*} \tag{II.26}$$

and

$$C_r^{\mathrm{new}} = C_r v^* \quad , \tag{II.27}$$

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

$$\varepsilon = -\delta + \frac{1}{v^*} \tag{II.28}$$

and

$$C_r^{\mathrm{new}} = -C_r v^* \quad . \tag{II.29}$$

We can replace the first constraint for $v^* < 0$ with

$$\varepsilon = 0 \tag{II.30}$$

due to the following proposition.

Proposition II.1.1. *OP 7 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 colinear points lying on the solution candidate, without errors. In the second group, we can further distinguish points which lie on both solution candidates (p_{csc}) and others (p_{css}). So we have two sets for both candidates

$$\left\{p_e^1, p_{css}^1, p_{csc}\right\} \tag{II.31}$$

$$\left\{p_e^2, p_{css}^2, p_{csc}\right\} \tag{II.32}$$

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 $|p_{er}^1| = |p_{er}^2|$. So we have

$$\left\{p_{er}, p_{css}^2, p_{css}^1, p_{csc}\right\} \tag{II.33}$$

$$\left\{p_{er}, p_{css}^1, p_{css}^2, p_{csc}\right\} \tag{II.34}$$

For solution 1, the error difference is

$$\left(|p_{er}| + \left|p_{css}^2\right|\right) \Delta\varepsilon + \left(\left|p_{css}^1\right| + |p_{csc}|\right) \Delta\varepsilon \tag{II.35}$$

adn for the second solution the error difference is

$$\left(|p_{er}| + \left|p_{css}^1\right|\right) \Delta\varepsilon + \left(\left|p_{css}^2\right| + |p_{csc}|\right) \Delta\varepsilon \tag{II.36}$$

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 plays important role in improving 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 some constant. It is obvious that such extreme solutions in most cases will not be good. We may decrease the value of ε to improve the solution. The δ-SVR, on the other hand, has the additional variable v which 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 which 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.

II.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 grid search method, we compare errors not on classification data, but on original regression data by using the regression function transformed from the classification boundary. We usually use mean squared error (MSE).

II.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_{u}^i and δ_{d}^i.

II.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. II.1, Fig. II.2. We show the equivalency of Bayes solutions for regression and transformed classification problems for some special cases.

Random mapping $\vec{x_{\mathrm{r}}} \to y_{\mathrm{r}}$ 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 has 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

$$\Pr\left(1 \mid \vec{x_{\mathrm{c}}}\right) - \Pr\left(-1 \mid \vec{x_{\mathrm{c}}}\right) \quad . \tag{II.37}$$

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

$$\mathrm{r}\left(\vec{x_{\mathrm{r}}}\right) = \mathbb{E}\left[y_{\mathrm{r}} \mid \vec{x_{\mathrm{r}}}\right] \quad . \tag{II.38}$$

Theorem II.2.1. *For a unimodal, symmetric 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* $\delta \geq 0$*.*

(Proof is in Appendix C.2). 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, symmetric distributions per point ($\mathrm{F}_{\mathrm{r}}^{x}\left(y_{\mathrm{r}} \mid \vec{x_{\mathrm{r}}}\right)$), with the same effect. The question arises about nonsymmetric distributions.

For nonsymmetric (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 nonsymmetric distribution of errors efficiently. We can reach the equivalency 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 II.2.2. *When* $F\left(m+\delta\right) - F\left(m-\delta\right) \geq 0$ *is fulfilled 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.*

(Proof is in Appendix C.3). The theorem states that by testing different values of δ we can reach the equivalency of the problems for the nonsymmetrical case, when the distribution fulfills some general assumptions. And finally we can use δ_u and δ_d parameters

Theorem II.2.3. *For a unimodal, nonsymmetric 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 do not have skewness in data at all. When $\delta_u/\delta_d > 1$ the distribution has a bigger variance on the upper side of the optimal regression function. The above theorem implicates that it would be possible to improve the results for nonsymmetrical 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.

II.3 Generalization Ability of δ-SVR

Vapnik developed statistical learning theory and SVM based on it [44, 45]. The key point in statistical learning theory is the analysis of generalization capabilities of machine learning methods without assuming any particular data distribution. 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.

II.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

$$\mathrm{R}_{\mathrm{emp}}\left(\alpha_l\right) \underset{\delta \to \infty}{\rightarrow} 0 \ , \tag{II.39}$$

where α_l is a curve for which $\mathrm{R}_{\mathrm{emp}}$ is minimal. In practice, for particular sample data 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 generally nonzero (e.g. for a linear set of functions where examples are not collinear). 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 comparing 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 (I.4). Therefore for particular 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.

II.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 (I.32) and (I.4). 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 define examples involved in realization of ERM, following [33]: for δ-SVR let's call a margin boundaries vector or an inside margin boundaries vector as *an essential margin vector* and a set of such vectors for particular hypothesis, EMV. For ε-SVR let's call ε-tube vector or outside ε-tube one *an essential margin vector* and a set of such vectors for particular hypothesis, EMV. By a configuration of essential margin vectors, labeled $CEMV$, we call a list of essential margin vectors for particular hypothesis, each with a distance to the margin boundary.

Let's imagine all hypotheses for particular p and q. The ε-SVR realizes ERM by finding the hypothesis which 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 which has a minimal value of a sum of differences in perpendicular distances in transformed space between EMV and the hypothesis curve.

Theorem II.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 (I.4) is constant, so we minimize only the second term. The distance from the i-th example with nonzero ξ_i to margin is $\xi_i / \|\vec{w_c}\|$. Because the denominator is constant, minimizing distances to examples lying outside a margin means minimizing 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 (errors-in-output data) and the difference grows for less flat functions. So this might be the reason of expecting better performance of ERM for δ-SVR for errors-in-output data, for flat functions. Now when we know how ERM is computed for ε-SVR and δ-SVR for particular δ and ε, we analyze which examples are involved in computing ERM.

First recall the proposition from [33]

Proposition II.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 the one value of ε. We can also notice that the distance to the margin 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 II.3.3. *Every CEMV for ε-SVR for particular ε_s is present in classification setting in δ-SVR for every $\delta > \delta_p$, where $\delta_p = \varepsilon_s$.*

If we consider $CEMV$ for ε-SVR, after transformation by δ, the margin distance is equal to $\delta - \varepsilon$. We can extend this proposition to the following:

Proposition II.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 do not exist in ε-SVR.

Proposition II.3.5. *When $|EMV| \le n$ for δ-SVR then the same CEMV exists for ε-SVR.*

It is quite obvious that $CEMV$ of δ-SVR where $|EMV| > n$ does not exist for 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 (II.9). 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 (II.9) and are not support vectors for δ-SVR. E.g. when $|EMV|$ is close to $2n$ we get the very small number of support vectors for δ-SVR.

Summarizing, based on the above propositions 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

$$d_c^i = \frac{\xi_c^i}{\|\vec{w_c}\|} \quad . \tag{II.40}$$

δ-SVR minimization function (I.4) can be rewritten as

$$f\left(\vec{w_c}, b_c, \vec{\xi_c}\right) = \|\vec{w_c}\|^2 + C_c \|\vec{w_c}\| \sum_{i=1}^{n} d_c^i \ . \tag{II.41}$$

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 (I.32) and the above for δ-SVR is $\|\vec{w_c}\|$. For ε-SVR a trade-off between ERM and CM is controlled by C_r, for δ-SVR we can also control the trade-off with C_c, but additionally it is dependent on $\|\vec{w_c}\|$. So when the method is looking for the best hypothesis, improving CM (increasing the margin distance) implicates lowering ERM importance.

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

$$f\left(\vec{w_c}, b_c, \vec{\xi_c}\right) = p^2 \|\vec{w_c}\|^2 + C_c p \|w_c\| \sum_{i=1}^{n} d_c^i \ , \tag{II.42}$$

where

$$p = \frac{1}{1 + w_c^{m+1}\left(\delta_2 - \delta_1\right)} \ , \tag{II.43}$$

so

$$f\left(\vec{w_c}, b_c, \vec{\xi_c}\right) = p^2 \left(\|\vec{w_c}\|^2 + \frac{C_c}{p} \|w_c\| \sum_{i=1}^{n} d_c^i\right) \ , \tag{II.44}$$

$$f\left(\vec{w_c}, b_c, \vec{\xi_c}\right) = p^2 \left(\|\vec{w_c}\|^2 + \left(1 + w_c^{m+1}\left(\delta_2 - \delta_1\right)\right) C_c \|w_c\| \sum_{i=1}^{n} d_c^i\right) \ . \tag{II.45}$$

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 greater while increasing δ.

II.3.3 VC Bounds for δ-SVR

In this subsection we consider the translation for δ-SVR independently of the data distribution. Vapnik proposed generalization bounds which are based on a VC dimension h [45]. With the probability at least $1 - \eta$ the inequality holds true

$$\mathrm{R}\left(\alpha\right) \leq \mathrm{R}_{\mathrm{emp}}\left(\alpha\right) + \frac{\varepsilon\left(n\right)}{2}\left(1 + \sqrt{1 + \frac{4\mathrm{R}_{\mathrm{emp}}\left(\alpha\right)}{\varepsilon\left(n\right)}}\right) \ , \tag{II.46}$$

where

$$\varepsilon\left(n\right) = 4\frac{\ln 2\tau + 1}{\tau} - \frac{\ln \eta/4}{n} \tag{II.47}$$

$$\tau = \frac{n}{h} \ , \tag{II.48}$$

h is a VC dimension. Vapnik derived also the bounds 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

$$\mathrm{R}\left(\alpha\right) \leq \mathrm{R}_{\mathrm{emp}}\left(\alpha\right) + \frac{B\varepsilon\left(n\right)}{2}\left(1 + \sqrt{1 + \frac{4\mathrm{R}_{\mathrm{emp}}\left(\alpha\right)}{B\varepsilon\left(n\right)}}\right) \ . \tag{II.49}$$

Therefore the bounds for classification and regression are pretty much the same. They are independent on data distribution. The key to minimize the right hand side is to control h. For this purpose Vapnik proposed Structural Risk Minimization. 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 $\vec{w_c} \cdot \vec{x} = 0$, where $\vec{w_c}$ is normalized such that they are in a canonical form, i.e. for a set of points $A = \{\vec{x_1}, \ldots, \vec{x_n}\}$

$$\min_i |\vec{w_c} \cdot \vec{x_i}| = 1 \ . \tag{II.50}$$

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

$$h \le \min\left(R^2\Lambda^2, m+1\right) \ , \tag{II.51}$$

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 [39]. 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 not constant, 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

$$h \le p^2\Lambda^2 (R + \Delta\delta)^2 \ , \tag{II.52}$$

where

$$p = \frac{1}{1 + w_c^{m+1}\Delta\delta} \ . \tag{II.53}$$

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

$$p^2 (R + \Delta\delta)^2 < R^2 \ . \tag{II.54}$$

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

$$w_c^{m+1} > \frac{1}{R} \ . \tag{II.55}$$

For $p < 0$

$$w_c^{m+1} < \frac{-2}{\Delta\delta} - \frac{1}{R} \ . \tag{II.56}$$

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

$$x_c^{m+1} = -w_c^m x_c^m / w_c^{m+1} \ . \tag{II.57}$$

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 generally δ-SVR has a potential to improve a VC bound by shifting without worsening empirical risk (II.39).

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 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 12. 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.

II.4 Experiments

For solving ε-SVR and SVC for particular parameters we use LibSVM [3] 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 II.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 [22] except stock price data, Table II.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 fulfilling $n/h < 20$. Recently, double-cross validation were used for SVM [52]. 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. The number of steps for outer cross validation is shown in *simC* column. 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. II.3.4 by comparing the number of support vectors and generalization performance for some particular values of δ and ε. In the second experiment we compare the generalization performance for variable δ and ε.

II.4.1 First Experiment

In the first experiment, we check the theoretical result from Prop. II.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. II.3, Fig. II.4.

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. We expect based on the results from the previous experiment that δ-SVR will achieve better generalization performance especially for some values of parameters for which we notice a differences in the number of support vectors. We can see in Table II.1, Fig. II.5, Fig. II.6 that generally performance for δ-SVR is better than ε-SVR for various δ and ε. We can see that for abalone, some caData and housing, data performance of δ-SVR is similar to the best performance for any δ in checked range, which is the difference from ε-SVR when performance decreases sharply while increasing ε.

The results are valuable in practice when we do not have enough time resources to find the best values of δ and ε, and we have to stick with some values chosen a priori. Then we expect better generalization performance of δ-SVR than ε-SVR for some value of δ and ε. In the next experiment we will compare results for variable δ and ε.

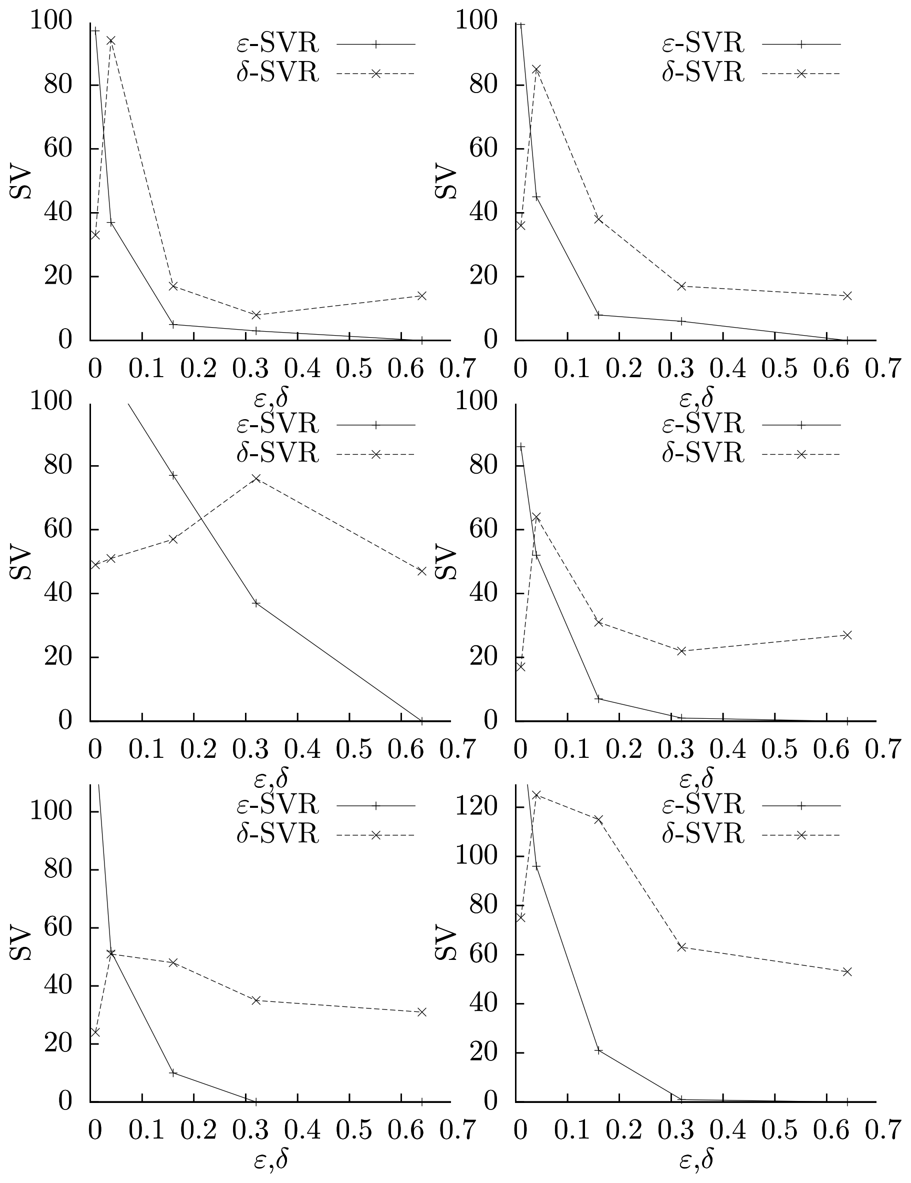

Figure II.3: Relation between ε, δ and the number of support vectors. In the figure, there is a relationship between value of ε for a line with '+' and the number of support vectors, and a relationship between a value of δ for a line with 'x' and the number of support vectors for the tests with ids: 1,4,5,10,11,12 from Table II.2 and Table II.3 respectively

Table II.1: Relation between ε, δ and RMSE. Column descriptions: *id* – an id of a test, 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	t1	t4	t5	t10	t11	t12	t13	t14	t15	t16
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**

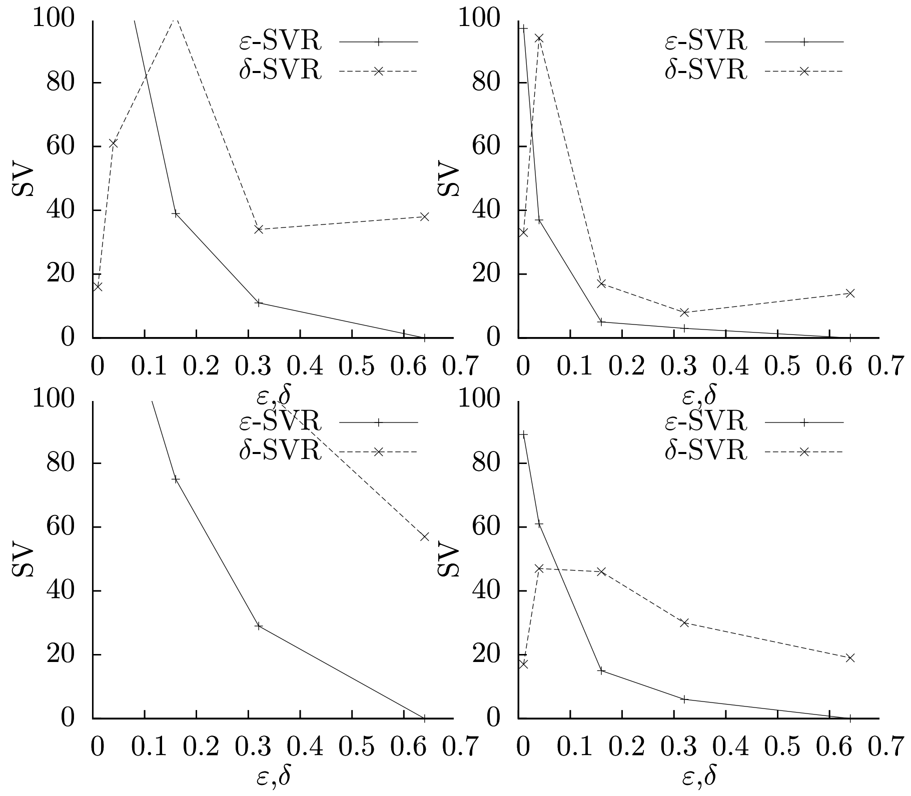

Figure II.4: Relation between ε, δ and the number of support vectors, cont. In the figure, there is a relationship between value of ε for a line with '+' and the number of support vectors, and a relationship between a value of δ for a line with 'x' and the number of support vectors for the tests with ids: 13,14,15,16 from Table II.2 and Table II.3 respectively

II.4.2 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.

Test results on synthetic data sets are presented in Table II.2. We can notice similar generalization performance for both δ-SVR and ε-SVR without any statistical difference based on t-test. However, we can notice an improved number of support vectors for δ-SVR for all tests which is also statistically significant for most of the tests.

For real world data sets, results are presented in Table II.3. We can notice similar generalization performance for both δ-SVR and ε-SVR without any statistical difference based on t-test. However, we can notice the improved number of support vectors for δ-SVR for half of the tests which is also statistically significant (see Table II.3).

II.5 Summary

In this thesis, 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 number of 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 standard SVC classification method by any other classification method based on kernel functions, implicating potential extension of applicability of a broad group of methods of classification for regression problems. In particular, any improvements for classification methods

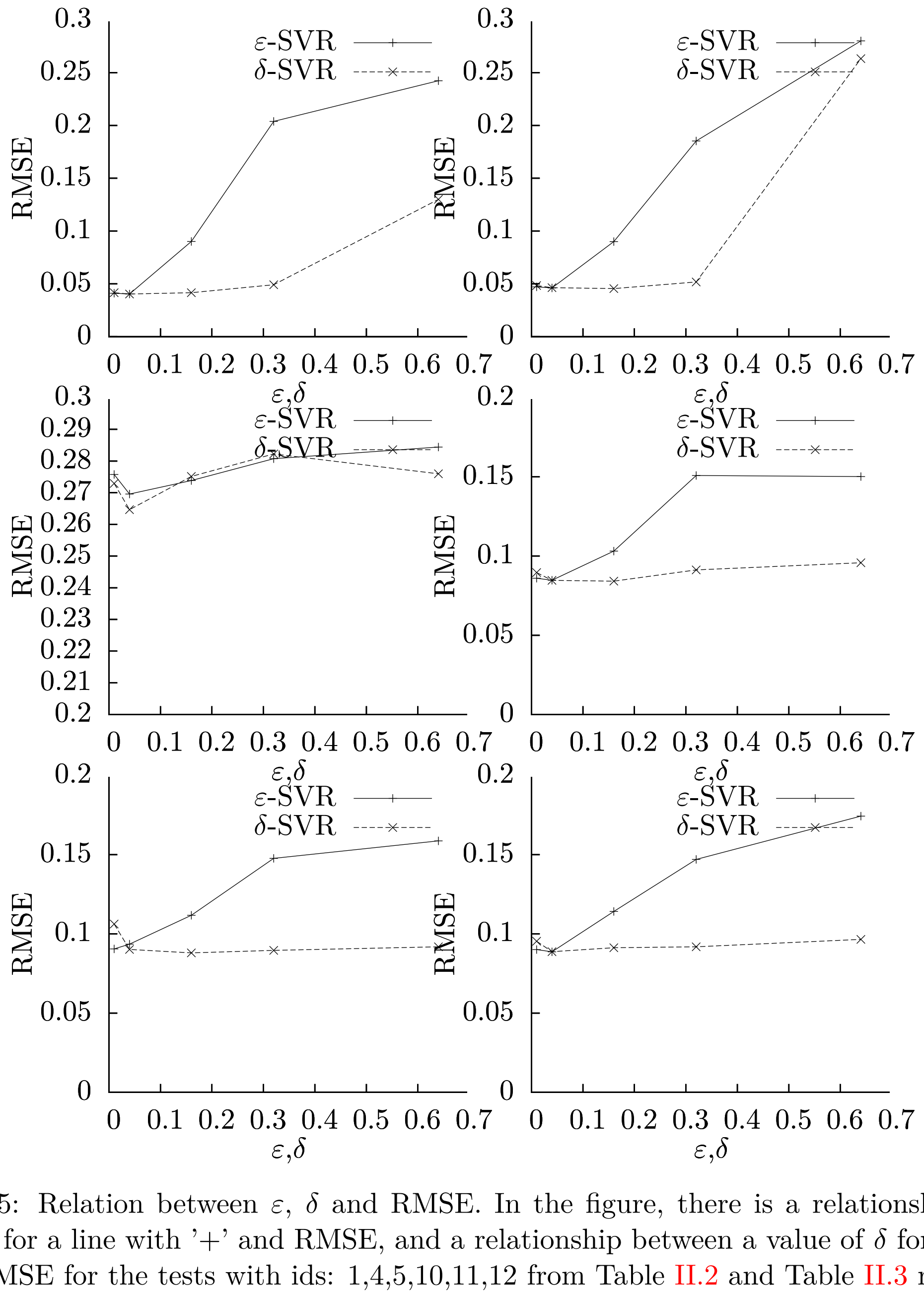

Figure II.5: Relation between ε, δ and RMSE. In the figure, there is a relationship between value of ε for a line with '+' and RMSE, and a relationship between a value of δ for a line with 'x' and RMSE for the tests with ids: 1,4,5,10,11,12 from Table II.2 and Table II.3 respectively

Table II.2: The δ-SVR performance for synthetic data. Column descriptions: *id* – an id of a test, *a function* – a function used for generating data $y_1 = \sum_{i=1}^{\dim} x_i$, $y_4 = \left(\sum_{i=1}^{\dim} x_i\right)^{\text{kerP}}$, $y_5 = 0.5 \sum_{i=1}^{\dim} \sin 10x_i + 0.5$, *simC* – the number of simulations, results are averaged, σ – a standard deviation used for generating noise in output, *ker* – a kernel (*pol* – a polynomial kernel), *kerP* – a kernel parameter (for a polynomial kernel it is a dimension, for the RBF kernel it is σ), *trs* – a training set size, *tes* – a testing set size, *dm* – a dimension of the problem, $dm = m$, *idRef* – a reference to the first table, *tr12M* – a percent average difference in MSE for training data, if greater than 0 than δ-SVR is better, *te12M* – the same as tr12M, but for testing data, *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. The value 'var' means that we search for the best value

id	function	simC	σ	ker	kerP	trs	tes	dm
1	y_1	100	0.04	lin	—	120	1000	4
2	$y_2 = 3y_1$	100	0.04	lin	—	120	1000	4
3	$y_3 = 1/3y_1$	100	0.04	lin	—	120	1000	4
4	y_4	100	0.04	pol	3	120	1000	4
5	y_5	10	0.04	rbf	var	100	1000	4

idRef	tr12M	te12M	teT	s1	s2	sT
1	−0.3%	−0.4%	−0.4	72	57	**3.6**
2	−0.3%	−0.3%	−0.3	70	61	**1.8**
3	−0.2%	−0.2%	−0.3	73	58	**3.2**
4	−2.3%	−0.4%	−0.2	73	68	**1.2**
5	−88%	−2.4%	−0.9	79	54	**1.6**

Table II.3: The δ-SVR performance for real world data. Column descriptions: *id* – an id of a test, *a name* – a name of the data set, *simC* – the number of random simulations, where training data are randomly selected, results are averaged, *ker* – a kernel (*pol* – a polynomial kernel), *kerP* – a kernel parameter (for a polynomial kernel it is a dimension, for the RBF kernel it is σ), *trs* – a training set size, *all* – the number of all data, it is a sum of training and testing data, *dm* – a dimension of the problem, $dm = m$, *idRef* – a reference to the first table, *tr12M* – a percent average difference in MSE for training data, if greater than 0 than δ-SVR is better, *te12M* – the same as tr12M, but for testing data, *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. The value 'var' means that we search for the best value

id	name	simC	ker	kerP	trs	all	dm
10	abalone	100	lin	—	180	4177	8
11	abalone	10	pol	5	180	4177	8
12	abalone	100	rbf	var	180	4177	8
13	caData	100	lin	—	180	20640	8
14	caData	100	pol	5	180	20640	8
15	caData	10	rbf	var	180	20640	8
16	housing	10	lin	—	280	506	13
19	stock	100	lin	—	100	1351	10
20	stock	10	pol	5	100	1351	10
21	stock	10	rbf	var	100	1351	10

idRef	tr12M	te12M	teT	s1	s2	sT
10	−0.2%	**0.8%**	1.3	90	91	−0.2
11	2.1%	−16.3%	−0.65	118	120	−0.1
12	−1.4%	**1.4%**	1.1	103	92	**1.74**
13	−0.6%	−1.3%	−1.9	91	80	**2.1**
14	0.5%	**2.7%**	0.5	99	99	0.0
15	−5.7%	**1%**	0.3	119	131	−0.7
16	−0.4%	**0.8%**	0.11	124	112	**0.5**
19	−0.2%	−0.3%	−0.3	62	50	**2.7**
20	7%	−29%	−1.2	46	59	−1.2
21	−5.5%	**2.1%**	0.13	84	62	**2.4**

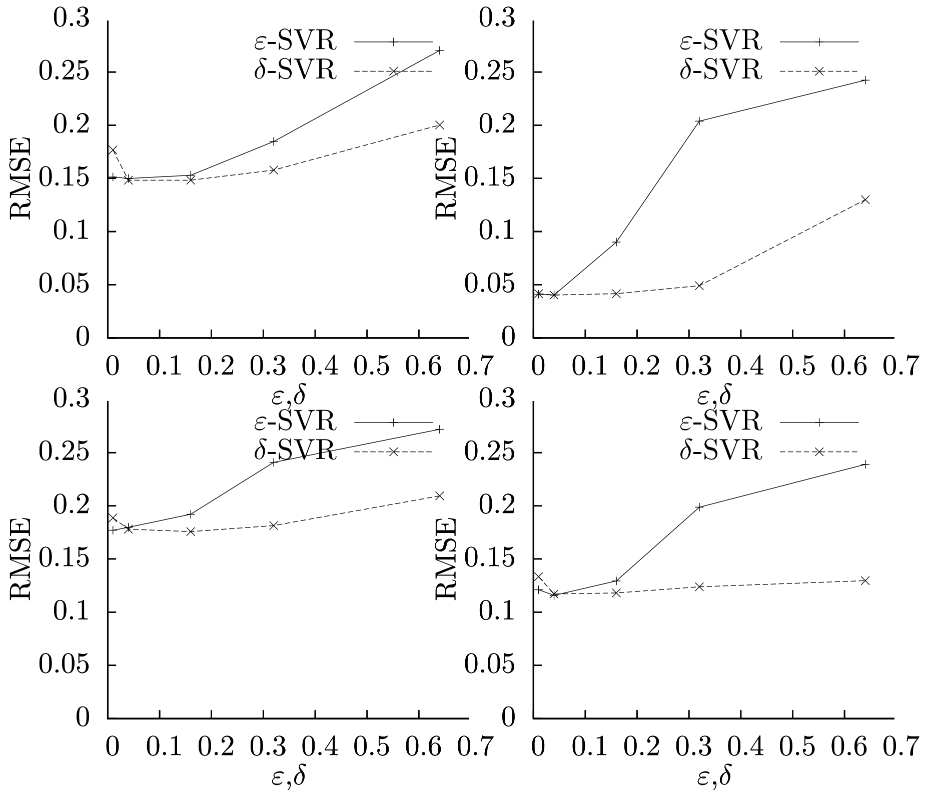

Figure II.6: Relation between ε, δ and RMSE, cont. In the figure, there is a relationship between value of ε for a line with '+' and RMSE, and a relationship between a value of δ for a line with 'x' and RMSE for the tests with ids: 13,14,15,16 from Table II.2 and Table II.3 respectively

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 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.

Chapter III

Margin Knowledge Per Example

III.1 Introduction to φ-SVC

The φ-SVC method is a recently proposed method of incorporating knowledge about margin to SVC [30, 32, 34]. The φ-SVC optimization problem includes an additional parameter per example in inequality constraints, on the right side of inequalities, (2). Another modification of inequality constraints were proposed in [51]. The authors modify the margin by multiplying the left side of the inequalities by some monotonically decreasing function of additional example weights. The φ-SVC is a more general concept of weights per example with any values possible and with different interpretation.

The φ-SVC method is used for incorporating margin knowledge represented by additional parameters per training example.

Definition III.1.1 (A tractor)**.** *A tractor is defined as an example with a classification value (1 or -1), and with the additional parameter φ, called a tractor parameter.*

The incorporation of tractors into SVC contains two steps: a tractor example is added to a training set with a proper classification value. Next, the modified SVC optimization problem is used. If a training set already contain a tractor example, the first step is skipped. Now, we will closely look at φ-SVC optimization problem. We are based on SVC with cost weights per example, OP 8

OP 13.

$$\min_{\vec{w_c}, b_c, \vec{\xi_c}} f\left(\vec{w_c}, b_c, \vec{\xi_c}\right) = \frac{1}{2}\left\|\vec{w_c}\right\|^2 + \vec{C_c} \cdot \vec{\xi_c} \tag{III.1}$$

subject to

$$y_c^i h\left(\vec{x_i}\right) \geq 1 + \varphi_i - \xi_c^i \tag{III.2}$$

$$\vec{\xi} \geq 0 \tag{III.3}$$

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

$$\vec{C_c} \gg 0 \tag{III.4}$$

$$\varphi_c^i \in \mathbb{R} \tag{III.5}$$

$$h\left(\vec{x_i}\right) = \vec{w_c} \cdot \vec{x_i} + b_c \quad . \tag{III.6}$$

The new weights $\vec{\varphi}$ are only present in constraints. When $\vec{\varphi} = 0$, the OP 13 is equivalent to OP 8. When all φ_i are equal to some constant $\varphi > -1$, we will get the same decision boundary as for $\varphi = 0$, when we change $\vec{C_c}$ to $\vec{C_c}/\left(1+\varphi\right)$.

Proof. We will prove that we get the same decision boundary when we replace 1 with $1/d$, for some $d > 0$, $\varphi_i = 0$. Let's replace ξ_i with ξ_i/d and we get the inequalities $y_c^i h\left(\vec{x_i}\right) \geq 1/d - \xi_c^i/d$. After multiplying by d we get $y_c^i dh\left(\vec{x_i}\right) \geq 1 - \xi_c^i$. The objective function can be multiplied by d^2 and we get $\frac{1}{2}\left\|d\vec{w_c}\right\|^2 + d\vec{C_c} \cdot \vec{\xi_c}$, the inequalities $\xi_i/d \geq 0$ can be replaced by $\xi_i \geq 0$. So we get the same optimization problem as the original one with the new $\vec{C}_c = d\vec{C}_c$ and with the new decision curve $dh(\vec{x}) = 0$. □

A functional margin for a point $\vec{p}$ is defined as a value $y_{\vec{p}}h(\vec{p})$. A value v in functional margin units is equal to $v/\|\vec{w}\|$. We can easily verify that a tractor parameter for $\varphi_i \geq 0$ is a lower bound on a distance from the tractor example to the margin measured in functional margin units: when we omit ξ_i in constraints for simplicity, we can see that $y_i h^*(\vec{x}_i) \geq 1 + \varphi_i$, when we divide both sides by $\|\vec{w}\|$, we get $y_i h^*(\vec{x}_i)/\|\vec{w}\| \geq 1/\|\vec{w}\| + \varphi_i/\|\vec{w}\|$.

Definition III.1.2 (A detractor)**.** *A detractor* is a tractor with $\varphi_i > 0$.

For $\varphi_i < 0$ and when the example is lying below the margin, the absolute value of a tractor parameter is an upper bound on a distance from the tractor example to the margin measured in functional margin units: when we omit ξ_i in constraints for simplicity, we get $-y_i h^*(\vec{x}_i)/\|\vec{w}\| + 1/\|\vec{w}\| \leq -\varphi_i/\|\vec{w}\|$. Both sides of the inequality are positives for this case.

Definition III.1.3 (An attractor)**.** *An attractor* is a tractor with $\varphi_i < 0$.

Note that when we take into account ξ_i, tractors can incorporate imperfect prior knowledge which is controlled by C_i parameters. Comparing loosely attractors to slack variables: attractors are constants, they are absent in objective function, whereas a sum of slack variables is minimized.

We can also derive the equivalent optimization problem to OP 13, where φ_i weights are present in the constraints with slack variables

OP 14.

$$\min_{\vec{w}_c, b_c, \vec{\xi}_c} f\left(\vec{w}_c, b_c, \vec{\xi}_c\right) = \frac{1}{2}\|\vec{w}_c\|^2 + \vec{C}_c \cdot \vec{\xi}_c \tag{III.7}$$

subject to

$$y_c^i h(\vec{x}_i) \geq 1 - \xi_c^i \tag{III.8}$$

$$\vec{\xi} \geq \varphi_i \tag{III.9}$$

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

In order to construct an efficient algorithm for the OP 13 its dual form was derived (derivation in D.1). The final form of the dual problem is

OP 15.

$$\max_{\vec{\alpha}} d(\vec{\alpha}) = \vec{\alpha} \cdot (1 + \vec{\varphi}) - \frac{1}{2}\vec{\alpha}^T Q \vec{\alpha} \tag{III.10}$$

subject to

$$\vec{\alpha} \cdot \vec{y} = 0 \tag{III.11}$$

$$0 \leq \vec{\alpha} \leq \vec{C} \ , \tag{III.12}$$

where

$$Q_{ij} = y_i y_j (\vec{x}_i \cdot \vec{x}_j) \tag{III.13}$$

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

It differs from the original SVC dual form by only $\vec{\alpha} \cdot \vec{\varphi}$ term. In the above formulation, similarly as for the original SVC, it is possible to introduce nonlinear decision functions by using a kernel function instead of a scalar product. The final decision boundary has a form

$$h^*(\vec{x}) = \sum_{i=1}^{n} y_i \alpha_i^* K(\vec{x}_i, \vec{x}) + b^* = 0 \ , \tag{III.14}$$

where $K(\cdot, \cdot)$ is a kernel function. The KKT complementary condition is

$$\alpha_i (y_i h(\vec{x}_i) - 1 - \varphi_i + \xi_i) = 0 \tag{III.15}$$

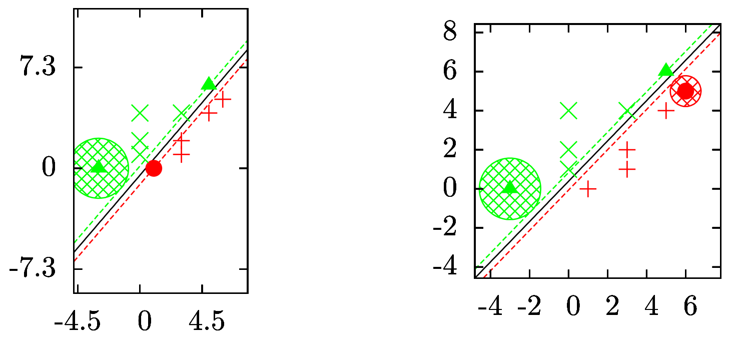

Figure III.1: Interpretation of detractors as dynamic hyperspheres. In the figures, there are example points, solutions (solid lines), support vectors (triangles and circles), tractors (circles filled with grid pattern). In both figures, there is a detractor in (-3, 0) with $\varphi = 5.0$. Radii of the detractors differ in both cases (2.2 and 1.6 respectively)

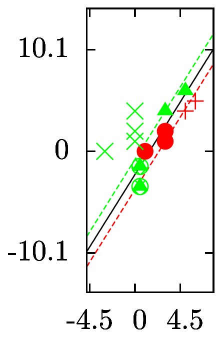

Figure III.2: Interpretation of an attractor as a hypersphere. In the figure, there are example points, a solution (solid line), support vectors (triangles and circles), tractors (circles filled with grid pattern). One of the attractor is (0.5, -1.5) with $\varphi = -1.0$.

$$(C_i - \alpha_i)\, \xi_i = 0 \quad . \tag{III.16}$$

We can find ξ_i parameters from the solution of the dual form as following. When

$$y_c^i h^* (\vec{x}_i) \geq 1 + \varphi_i \quad , \tag{III.17}$$

then $\xi_c^i = 0$, else

$$\xi_c^i = 1 + \varphi_i - y_c^i h (\vec{x}_i) \quad . \tag{III.18}$$

III.2 Interpretation of Tractors as Dynamic Hyperspheres

A detractor example $\vec{p}$ can be interpreted as a hypersphere with a radius equals to φ_p in functional margin units and therefore this is a dynamic hypersphere with a variable radius which depends on a decision function. The hypersphere must not intersect the margin boundary (in more than one point) $y_p h (\vec{x}) = 1$. A value of the radius is represented in functional margin units and hence its absolute value varies among solution candidates. For the two solution candidates $h_1 (\vec{x}) = 0$ and $h_2 (\vec{x}) = 0$, where $h_2 (\vec{x}) = a h_1 (\vec{x})$ and $a \neq 0$ (both hyperplanes have the same geometric locations), the hyperspheres are respectively $S_1 (\vec{p}, r)$, and $S_2 (\vec{p}, r/a)$ (Fig. III.1).

An attractor example $\vec{p}$ can be interpreted as a hypersphere with a radius equals to $|\varphi_p|$ in functional margin units with a center in the p-th point that must intersect a margin boundary $y_i h (\vec{x}) = 1$, Fig. III.2.

III.3 An Efficient Solution of φ-SVC

In order to solve OP 15, a decomposition method similar to SMO [36] which solves the original SVC dual optimization problem was derived. For two chosen parameters i_1 and i_2 the solution without clipping is

$$\alpha_{i_2}^{\text{new}} = \alpha_{i_2} + \frac{y_{i_2}\left(E_{i_1} - E_{i_2}\right)}{\kappa} \quad , \tag{III.19}$$

where $\kappa = K_{i_1 i_1} + K_{i_2 i_2} - 2K_{i_1 i_2}$ and

$$E_i = \sum_{j=1}^{n} y_j \alpha_j K_{ij} - y_i - y_i \varphi_i \quad . \tag{III.20}$$

After that, α_{i_2} is clipped in the same way as for SMO, but with variable weights C_i (derivation in D.2)

$$U \le \alpha_{i_2}^{\text{clipped}} \le V \quad , \tag{III.21}$$

where for $y_1 \neq y_2$:

$$U = \max\left(0, \alpha_{i_2}^{\text{old}} - \alpha_{i_1}^{\text{old}}\right) \tag{III.22}$$

$$V = \min\left(C_{i_2}, C_{i_1} - \alpha_{i_1}^{\text{old}} + \alpha_{i_2}^{\text{old}}\right) \tag{III.23}$$

for $y_1 = y_2$:

$$U = \max\left(0, \alpha_{i_1}^{\text{old}} + \alpha_{i_2}^{\text{old}} - C_{i_1}\right) \tag{III.24}$$

$$V = \min\left(C_{i_2}, \alpha_{i_1}^{\text{old}} + \alpha_{i_2}^{\text{old}}\right) \quad . \tag{III.25}$$

The parameter α_{i_1} is

$$\alpha_{i_1}^{\text{new}} = \gamma - y_{i_1} y_{i_2} \alpha_{i_2}^{\text{clipped}} \quad , \tag{III.26}$$

where

$$\gamma = \alpha_{i_1}^{\text{old}} + y_{i_1} y_{i_2} \alpha_{i_2}^{\text{old}} \quad . \tag{III.27}$$

Based on the KKT complementary condition, it is possible to derive equations for the SVC heuristic and the SVC stopping criteria. After incorporating weights $\vec{\varphi}$, a heuristic and stopping criteria are almost the same, with the one difference, that values of E_i are computed as stated in (III.20).

III.4 New Types of Support Vectors

The i-th example is a support vector, when $\alpha_i^* \neq 0$. From KKT (III.15),(III.16) we can conclude which examples could be support vectors. In the original SVC, only the example which lies on the optimal margin boundaries ($y_i h^*\left(\vec{x_i}\right) = 1$) or below optimal margin boundaries ($y_i h^*\left(\vec{x_i}\right) < 1$) could be a support vector. In φ-SVC, also the example fulfilling $\varphi_i > 0$ and lying above margin boundaries ($y_i h^*\left(\vec{x_i}\right) > 1$) could be a support vector. Such example is called *a detractor support vector*, Fig. III.1. An output model is defined based on support vectors. Introducing the new type of support vectors leads to richer models, where additional examples lying above optimal margin boundaries could participate in defining a decision function.

Attractors could lead to the examples which lie below the margin and are not support vectors. Moreover, attractors could lead to the new type of support vectors – which have slack variables equal to zero and do not lie on the margin.

The better flexibility of φ-SVC in choosing support vectors suggests that we can build models with fewer support vectors for the similar generalization performance.

III.5 ε-SVR Reformulation as φ-SVC

It was shown recently that a reformulation of ε-SVR is a special case of φ-SVC [34]. The similar reformulation was implemented in LibSVM [3] for solving ordinal classification problems – without prior knowledge, and ε-SVR. The consequence of this reformulation is that we can apply all the applications for generalized SVC also for ε-SVR. These are manipulating of the regression function proposed as a manipulation of the decision curve for classification problems in [30] and creating improved reduced models by removing support vectors proposed for classification problems in [32]. The former was also investigated for δ-SVR recently proposed [33].

Here we present a reformulation of the OP 7. Every regression training example is duplicated, Fig. III.3. Every original training example gets 1 class, and the duplicated training example gets -1 class and therefore we get

OP 16.

$$\min_{\vec{w_c}, b_c, \vec{\xi_c}} f\left(\vec{w_c}, b_c, \vec{\xi_c}\right) = \frac{1}{2}\left\|\vec{w_c}\right\|^2 + C_c \sum_{i=1}^{2n} \xi_c^i \tag{III.28}$$

subject to

$$y_c^i h\left(\vec{x_i}\right) \geq 1 - \xi_c^i + \varphi_i \tag{III.29}$$

$$\vec{\xi_c} \geq 0 \tag{III.30}$$

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

$$h\left(\vec{x_i}\right) = \vec{w_c} \cdot \vec{x_i} + b_c \tag{III.31}$$

and

$$\varphi_i = y_c^i y_r^i - \varepsilon - 1 \ . \tag{III.32}$$

The OP 16 is a special case of OP 13. Instead of using a decision curve of OP 16 we use a regression function

$$\sum_{i=1}^{2n} y_c^i \alpha_i^* K\left(\vec{x_i}, \vec{x}\right) + b_c^* = 0 \quad \rightarrow \quad g^*\left(\vec{x}\right) = \sum_{i=1}^{2n} y_c^i \alpha_i^* K\left(\vec{x_i}, \vec{x}\right) + b_c^* \ . \tag{III.33}$$

In a typical scenario $\varphi_i < 0$, because ε is close to 0 and y_r^i is less than 1. We can notice the following property of the OP 16. Because every training example is duplicated, for every possible solution, n training examples will be always incorrectly classified except those lying on a classification decision boundary, Fig. III.3. An efficient solution of the OP 16 based on Sequential Minimal Optimization [36] is the same as described in [30, 32]. The KKT complementary condition for ε-SVR is the same as for φ-SVC after reformulation.

III.6 Using Margin Knowledge with ε-SVR

The ε-SVR can be reformulated as φ-SVC. We can additionally modify margin weights for incorporating margin knowledge. There are two options of incorporation available, either we can modify φ_p and φ_{p+n} for some vector p after transforming the problem to φ-SVC or we can modify ε_u^p or ε_d^p before the transformation. In the first option, the φ_p and φ_{p+n} weights are already set according to (III.32). So adding margin knowledge means the manipulation of these weights. In the second option, ε_u^p or ε_d^p weights are set to some ε value for standard ε-SVR. Both approached are equivalent, in the sense that modification of ε_u^p and ε_d^p is equivalent to the modification

$$\Delta\varphi_p = -\Delta\varepsilon_d^p \ , \tag{III.34}$$

$$\Delta\varphi_{p+n} = -\Delta\varepsilon_u^p \ . \tag{III.35}$$

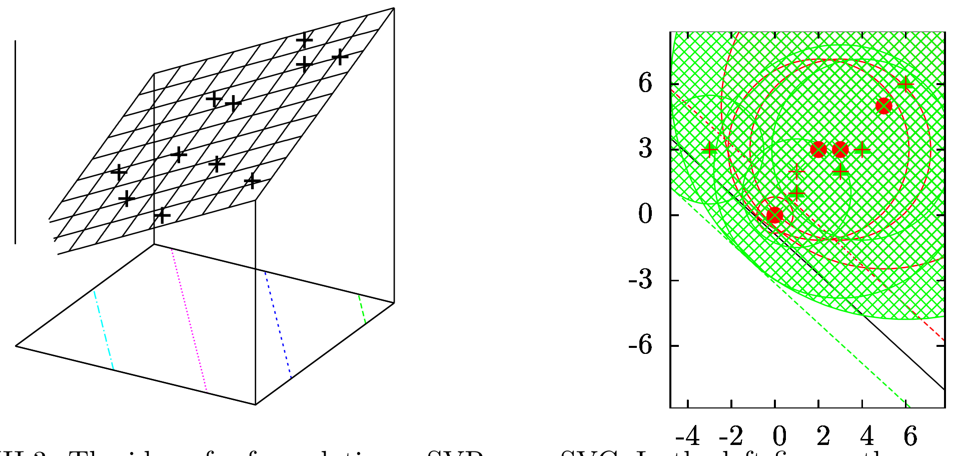

Figure III.3: The idea of reformulating ε-SVR as φ-SVC. In the left figure, there are regression data points and a solution (a plane). In the right figure, there is a transformed problem, classification data points, support vectors (triangles and circles), tractors (circles filled with grid pattern), a solution (a solid line), margins (dashed lines)

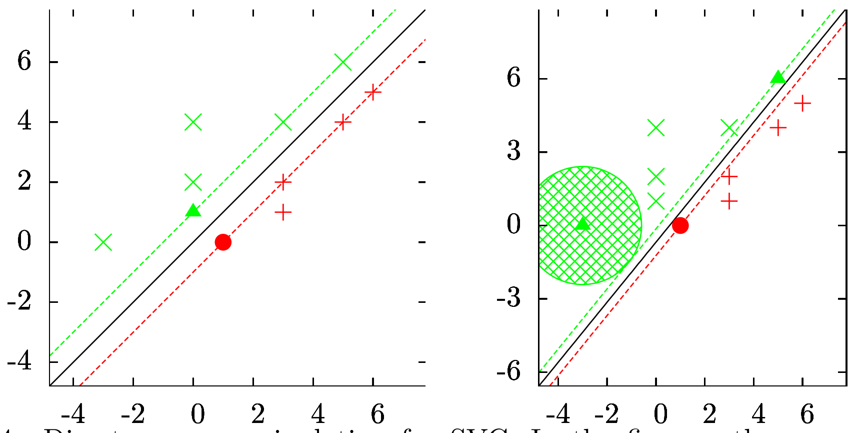

Figure III.4: Direct curve manipulation for SVC. In the figures, there are example points, support vectors (triangles and circles), tractors (circles filled with grid pattern), solutions (solid lines), margins (dashed lines). In the right figure, there is a tractor which causes lowering the solution from the left figure

III.7 Using Margin Knowledge with δ-SVR

The δ-SVR is transformed to the classification problem, so we can use φ-SVC instead of standard SVC for incorporating margin knowledge. From some point p, we set φ_p for the original example, and φ_{p+n} for the duplicated one.

III.8 Changing the Output Curve

After modification of φ_i weights, either the output curve stays the same, or it is changed. For example, let's assume that we modify only a one example $\vec{p}$ and $\varphi_{\vec{p}}$ is equal to zero before the modification. When $y_{\vec{p}} h^* (\vec{p}) > 1$, then setting $0 < \varphi_{\vec{p}} \leq y_{\vec{p}} h^* (\vec{p}) - 1$ will not affect a solution. When we set $\varphi_{\vec{p}} > y_{\vec{p}} h^* (\vec{p}) - 1$, the solution will be different, but not necessarily a decision boundary. Particularly, setting $\varphi_{\vec{p}} > 0$ can increase a slack variable, when a value of $C_{\vec{p}}$ is small, or it can cause decrease of the margin size (decreased $\|\vec{w_c}\|$), and in both cases the solution can stay the same. We can see the example of changed output curve for φ-SVC, Fig. III.4, for δ-SVR, Fig. III.5, for ε-SVR, Fig. III.6.

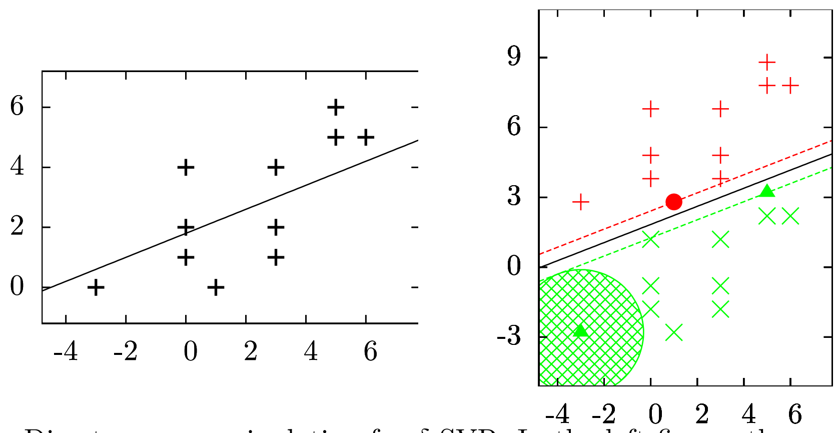

Figure III.5: Direct curve manipulation for δ-SVR. In the left figure, there are regression data points, a solution (a solid line). In the right figure, there is a transformed problem: classification data points, support vectors (triangles and circles), tractors (circles filled with grid pattern), a solution (a solid line), margins (dashed lines). In the right figure, there is also a tractor which causes changing the solution from the left figure

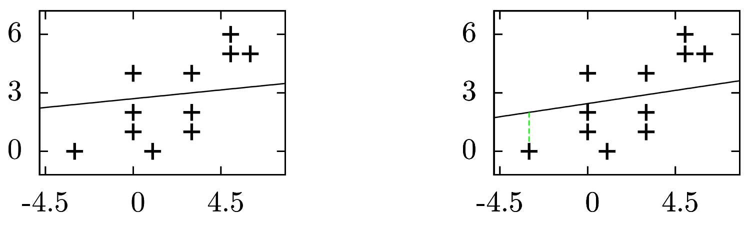

Figure III.6: Direct curve manipulation for ε-SVR. In both figures, there are regression data points, solutions (solid lines). In the right figure, there is a tractor which causes changing the solution from the left figure (a dotted line)

Let's now analyze in details changing the output curve. First consider changing the C parameter. We can see that for OP 8, if we increase value of C_c^p for some point p for which $\xi_c^p = 0$, then the solution remains the same. Consider now OP 13 and the solution with all $\varphi_i = 0$. If we want to add a detractor, we have to set $\varphi_p > 0$ for some example p. After adding a detractor for which

$$\varphi_p \leq y_c^p h\left(x_c^p\right) - 1 + \xi_c^p \ , \tag{III.36}$$

the solution remains the same, in the other case variables $\vec{w}, \vec{\xi}, b$ will be adjusted, but the decision curve can stay the same. It will stay the same in the following cases:

1. The ξ_c^p is only adjusted, to

   $$\xi_{cnew}^p = \varphi_p - y_c^p h\left(\vec{x_p}\right) + 1 \ . \tag{III.37}$$

   The objective function will raise by

   $$\Delta f = C_p \Delta \xi_c^p \ . \tag{III.38}$$

2. The $h\left(\vec{x_c}\right)$ is modified

   $$h_{cnew}\left(\vec{x_c}\right) = c h\left(\vec{x_c}\right) \ , \tag{III.39}$$

   where $c > 1$. The objective function will raise by

   $$\Delta f = \frac{1}{2}\left\|\vec{w}\right\|^2\left(c^2 - 1\right) \ . \tag{III.40}$$

   The modification parameter c is set in a way that the following equation is fulfilled

   $$c y_c^p h\left(\vec{x_p}\right) = 1 - \xi_c^p + \varphi_p \ . \tag{III.41}$$

   Note that in this case other non-zero ξ_c^i values have to be modified

   $$\xi_c^{iold} = 1 - y_c^i h\left(\vec{x_i}\right) \tag{III.42}$$

   $$\xi_c^{inew} = 1 - c y_c^i h\left(\vec{x_i}\right) \tag{III.43}$$

   $$\Delta \xi_c^i = y_c^i h\left(\vec{x_i}\right)(1 - c) \ . \tag{III.44}$$

   For misclassified examples the change of an objective function will be positive, otherwise negative

   $$\Delta f = \frac{1}{2}\left\|\vec{w}\right\|^2\left(c^2 - 1\right) + \sum_{i=1, i \neq p}^{n} C_c^i \Delta \xi_c^i \ . \tag{III.45}$$

3. Both above changes are present simultaneously. The objective function will be changed by

   $$\Delta f = \frac{1}{2}\left\|\vec{w}\right\|^2\left(c^2 - 1\right) + C_c^p \Delta \xi_c^p + \sum_{i=1, i \neq p}^{n} C_c^i \Delta \xi_c^i \ . \tag{III.46}$$

Changing the decision curve is possible, especially when Δf is high, because other solutions (which are able to change the decision curve) are more likely to have a better value of f. We can see that we can increase a chance for changing the decision boundary by increasing the parameter C_c^p.

III.9 Incorporating Linear Dependency of Function Values

Another example of application for tractors is the incorporation of the additional constraint in the form of a constant value of linear dependency of function values, that the solution must fulfill, for regression:

$$\sum_{i=1}^{s} s_i g\left(\vec{d_i}\right) = e \quad , \tag{III.47}$$

where s_i are some parameters, for which $\sum_{i=1}^{s} s_i \neq 0$, d_i are some points, s is the number of points d_i, e is a parameter, g is defined in (I.37), and for classification:

$$\sum_{i=1}^{s} s_i h\left(\vec{d_i}\right) = e \quad . \tag{III.48}$$

where s_i are some parameters, for which $\sum_{i=1}^{s} s_i \neq 0$, d_i are some points, s is the number of d_i points, e is a parameter.

We will show how to incorporate this constraint to φ-SVC, δ-SVR, and ε-SVR. Tractors are used in incorporation of this constraint to φ-SVC and ε-SVR.

III.9.1 Incorporating Linear Dependency of Function Values to φ-SVC

We will incorporate (III.48) to OP 13. After reformulation

$$\sum_{i=1}^{s} s_i \vec{w_c} \cdot \vec{d_i} + b \sum_{i=1}^{s} s_i = e \tag{III.49}$$

$$b = \frac{1}{\sum_{i=1}^{s} s_i} \left(e - \sum_{i=1}^{s} s_i \vec{w_c} \cdot \vec{d_i} \right) \quad . \tag{III.50}$$

Now we can substitute b to $h\left(\vec{x}\right)$ and we get

$$h\left(\vec{x}\right) = \vec{w_c} \cdot \vec{x} + \frac{1}{\sum_{i=1}^{s} s_i} \left(e - \sum_{i=1}^{s} s_i \vec{w_c} \cdot \vec{d_i} \right) \tag{III.51}$$

after reformulation

$$h\left(\vec{x}\right) = \vec{w_c} \cdot \vec{x} - \frac{1}{\sum_{i=1}^{s} s_i} \vec{w_c} \cdot \sum_{i=1}^{s} s_i \vec{d_i} + \frac{e}{\sum_{i=1}^{s} s_i} \quad . \tag{III.52}$$

After substituting above to OP 13, we get the φ-SVC problem without the offset with the new kernel in the form of transformation of the input vectors

$$\vec{x} \to \vec{x} - \frac{1}{\sum_{i=1}^{s} s_i} \sum_{i=1}^{s} s_i \vec{d_i} \tag{III.53}$$

$$K\left(\vec{x}, \vec{y}\right) = \left(\vec{x} - \frac{1}{\sum_{i=1}^{s} s_i} \sum_{i=1}^{s} s_i \vec{d_i} \right) \left(\vec{y} - \frac{1}{\sum_{i=1}^{s} s_i} \sum_{i=1}^{s} s_i \vec{d_i} \right) \tag{III.54}$$

and tractors set as

$$\varphi_i = \varphi_{old} - y_i \frac{e}{\sum_{i=1}^{s} s_i} \quad . \tag{III.55}$$

We propose also nonlinear solutions by using the following way of further kernelization

$$K\left(\vec{x}, \vec{y}\right) = K_o\left(\vec{x}, \vec{y}\right) - \frac{1}{\sum_{i=1}^{s} s_i} \sum_{i=1}^{s} s_i K_o\left(\vec{x}, \vec{d_i}\right) - \frac{1}{\sum_{i=1}^{s} s_i} \sum_{i=1}^{s} s_i K_o\left(\vec{y}, \vec{d_i}\right) \tag{III.56}$$

$$+ \frac{1}{\left(\sum_{i=1}^{s} s_i\right)^2} \sum_{i=1}^{s} \sum_{j=1}^{s} s_i s_j K_o\left(\vec{d_i}, \vec{d_j}\right) \tag{III.57}$$

This kernel is used only for solving the optimization problem. Now we derive the solution, because

$$\vec{w_c} = \sum_{j=1}^{n} \alpha_j y_c^j \left(\vec{x_j} - \frac{1}{\sum_{i=1}^{s} s_i} \sum_{i=1}^{s} s_i \vec{d_i} \right) \tag{III.58}$$

we get

$$h(\vec{x}) = \vec{w_c} \cdot \vec{x} + b = \sum_{j=1}^{n} \alpha_j y_c^j \left(\vec{x_j} - \frac{1}{\sum_{i=1}^{s} s_i} \sum_{i=1}^{s} s_i \vec{d_i} \right) \cdot \vec{x} + b \tag{III.59}$$

$$= \sum_{j=1}^{n} \alpha_j y_c^j \vec{x_j} \cdot \vec{x} - \frac{1}{\sum_{i=1}^{s} s_i} \sum_{j=1}^{n} \sum_{i=1}^{s} \alpha_j y_c^j s_i \vec{d_i} \cdot \vec{x} + b \tag{III.60}$$

and after adding internal kernels we get

$$h(\vec{x}) = \sum_{j=1}^{n} \alpha_j y_c^j K_o(\vec{x_j}, \vec{x}) - \frac{1}{\sum_{i=1}^{s} s_i} \sum_{j=1}^{n} \sum_{i=1}^{s} \alpha_j y_c^j s_i K_o\left(\vec{d_i}, \vec{x}\right) + b \tag{III.61}$$

when b is computed as

$$b = \frac{1}{\sum_{i=1}^{s} s_i} \left(e - \sum_{i=1}^{s} s_i \sum_{j=1}^{n} \alpha_j y_c^j \left(\vec{x_j} - \frac{1}{\sum_{i=1}^{s} s_i} \sum_{k=1}^{s} s_k \vec{d_k} \right) \cdot \vec{d_i} \right) \tag{III.62}$$

$$= \frac{1}{\sum_{i=1}^{s} s_i} \left(e - \sum_{i=1}^{s} s_i \sum_{j=1}^{n} \alpha_j y_c^j K_o\left(\vec{x_j}, \vec{d_i}\right) + \frac{1}{\sum_{i=1}^{s} s_i} \sum_{i=1}^{s} s_i \sum_{j=1}^{n} \sum_{k=1}^{s} \alpha_j y_c^j s_k K_o\left(\vec{d_k}, \vec{d_i}\right) \right) \tag{III.63}$$

So we can interpret a solution as a standard SVM solution with a non-symmetrical translated kernel, where only training examples are translated, and with b computed as above.

III.9.2 Incorporating Linear Dependency of Function Values to δ-SVR

We will incorporate (III.47) to δ-SVR. We assume that we use special kernels developed for δ-SVR. For δ-SVR the condition becomes

$$\sum_{i=1}^{s} s_i \frac{-\vec{w_{\mathrm{red}}} \cdot \vec{d_{\mathrm{i,red}}} - b_{\mathrm{c}}}{w_{\mathrm{c}}^{m+1}} = e \quad , \tag{III.64}$$

After transformation

$$e w_{\mathrm{c}}^{m+1} + \sum_{i=1}^{s} s_i \vec{w_{\mathrm{red}}} \cdot \vec{d_{\mathrm{i,red}}} + \sum_{i=1}^{s} s_i b_c = 0 \tag{III.65}$$

$$b_c = \frac{1}{\sum_{i=1}^{s} s_i} \left(-e w_{\mathrm{c}}^{m+1} - \sum_{i=1}^{s} s_i \vec{w_{\mathrm{red}}} \cdot \vec{d_{\mathrm{i,red}}} \right) \tag{III.66}$$

Substituting it to $h(\vec{x})$ we get

$$h(\vec{x}) = \vec{w_c} \cdot \vec{x} + \frac{1}{\sum_{i=1}^{s} s_i} \left(-e w_{\mathrm{c}}^{m+1} - \sum_{i=1}^{s} s_i \vec{w_{\mathrm{red}}} \cdot \vec{d_{\mathrm{i,red}}} \right) \tag{III.67}$$

$$h(\vec{x}) = \vec{w_{\mathrm{red}}} \cdot \vec{x_{\mathrm{red}}} + w_{\mathrm{c}}^{m+1} x_{m+1} + \frac{1}{\sum_{i=1}^{s} s_i} \left(-e w_{\mathrm{c}}^{m+1} - \sum_{i=1}^{s} s_i \vec{w_{\mathrm{red}}} \cdot \vec{d_{\mathrm{i,red}}} \right) \tag{III.68}$$

$$h(\vec{x}) = \vec{w_{\mathrm{red}}} \cdot \left(\vec{x_{\mathrm{red}}} - \frac{1}{\sum_{i=1}^{s} s_i} \sum_{i=1}^{s} s_i \vec{d_{\mathrm{i,red}}} \right) + w_c^{m+1} \left(x_{m+1} - \frac{e}{\sum_{i=1}^{s} s_i} \right) \tag{III.69}$$

After substituting above to OP 13, we get the φ-SVC problem without the offset with the new kernel in the form

$$K(\vec{x}, \vec{y}) = \left(\vec{x_{\text{red}}} - \frac{1}{\sum_{i=1}^{s} s_i} \sum_{i=1}^{s} s_i \vec{d_{\text{i,red}}}\right) \left(\vec{y_{\text{red}}} - \frac{1}{\sum_{i=1}^{s} s_i} \sum_{i=1}^{s} s_i \vec{d_{\text{i,red}}}\right) \tag{III.70}$$

$$+ \left(x_{m+1} - \frac{e}{\sum_{i=1}^{s} s_i}\right) \left(y_{m+1} - \frac{e}{\sum_{i=1}^{s} s_i}\right) \tag{III.71}$$

Then

$$K(\vec{x}, \vec{y}) = \vec{x_{\text{red}}} \cdot \vec{y_{\text{red}}} - \frac{1}{\sum_{i=1}^{s} s_i} \sum_{i=1}^{s} s_i \vec{d_{\text{i,red}}} \cdot \vec{x_{\text{red}}} - \frac{1}{\sum_{i=1}^{s} s_i} \sum_{i=1}^{s} s_i \vec{d_{\text{i,red}}} \cdot \vec{y_{\text{red}}} \tag{III.72}$$

$$+ \frac{1}{\left(\sum_{i=1}^{s} s_i\right)^2} \sum_{i=1}^{s} \sum_{j=1}^{s} s_i s_j \vec{d_{\text{i,red}}} \cdot \vec{d_{\text{j,red}}} + x_{m+1} y_{m+1} - x_{m+1} \frac{e}{\sum_{i=1}^{s} s_i} - y_{m+1} \frac{e}{\sum_{i=1}^{s} s_i} \tag{III.73}$$

$$+ \frac{e^2}{\left(\sum_{i=1}^{s} s_i\right)^2} \tag{III.74}$$

We propose also nonlinear solutions by using the following way of further kernelization

$$K(\vec{x}, \vec{y}) = K_o\left(\vec{x_{\text{red}}}, \vec{y_{\text{red}}}\right) - \frac{1}{\sum_{i=1}^{s} s_i} \sum_{i=1}^{s} s_i K_o\left(\vec{d_{\text{i,red}}}, \vec{x_{\text{red}}}\right) \tag{III.75}$$

$$- \frac{1}{\sum_{i=1}^{s} s_i} \sum_{i=1}^{s} s_i K_o\left(\vec{d_{\text{i,red}}}, \vec{y_{\text{red}}}\right) + \frac{1}{\left(\sum_{i=1}^{s} s_i\right)^2} \sum_{i=1}^{s} \sum_{j=1}^{s} s_i s_j K_o\left(\vec{d_{\text{i,red}}}, \vec{d_{\text{j,red}}}\right) \tag{III.76}$$

$$+ x_{m+1} y_{m+1} - x_{m+1} \frac{e}{\sum_{i=1}^{s} s_i} - y_{m+1} \frac{e}{\sum_{i=1}^{s} s_i} + \frac{e^2}{\left(\sum_{i=1}^{s} s_i\right)^2} \; . \tag{III.77}$$

This kernel is used only for solving the optimization problem. So solving δ-SVR with the additional constraint leads to SVC optimization problem without the offset and with a special kernel presented above. Now we derive the solution

$$h(\vec{x}) = \vec{w_c} \cdot \vec{x} + b_c = \sum_{j=1}^{2n} \alpha_j y_c^j \left(\vec{x_{\text{j,red}}} - \frac{1}{\sum_{i=1}^{s} s_i} \sum_{i=1}^{s} s_i \vec{d_{\text{i,red}}}\right) \cdot \vec{x_{\text{red}}} \tag{III.78}$$

$$+ \sum_{j=1}^{2n} \alpha_j y_c^j \left(x_j^{m+1} - \frac{e}{\sum_{i=1}^{s} s_i}\right) x_{m+1} + b_c \tag{III.79}$$

with internal kernels

$$h(\vec{x}) = \sum_{j=1}^{2n} \alpha_j y_c^j \left(K_o\left(\vec{x_{\text{j,red}}}, \vec{x_{\text{red}}}\right) - \frac{1}{\sum_{i=1}^{s} s_i} \sum_{i=1}^{s} s_i K_o\left(\vec{d_{\text{i,red}}}, \vec{x_{\text{red}}}\right)\right) \tag{III.80}$$

$$+ \sum_{j=1}^{2n} \alpha_j y_c^j \left(x_j^{m+1} - \frac{e}{\sum_{i=1}^{s} s_i}\right) x_{m+1} + b_c \tag{III.81}$$

where b_c is computed as

$$b_c = \frac{1}{\sum_{i=1}^{s} s_i} \left(-e \sum_{j=1}^{2n} \alpha_j y_c^j \left(x_j^{m+1} - \frac{e}{\sum_{i=1}^{s} s_i}\right)\right) \tag{III.82}$$

$$- \frac{1}{\sum_{i=1}^{s} s_i} \left(\sum_{i=1}^{s} s_i \sum_{j=1}^{2n} \alpha_j y_c^j \left(\vec{x_{\text{j,red}}} - \frac{1}{\sum_{i=1}^{s} s_i} \sum_{k=1}^{s} s_k \vec{d_{\text{k,red}}}\right) \cdot \vec{d_{\text{i,red}}}\right) \tag{III.83}$$

$$= -\frac{1}{\sum_{i=1}^{s} s_i} e \sum_{j=1}^{2n} \alpha_j y_c^j \left(x_j^{m+1} - \frac{e}{\sum_{i=1}^{s} s_i} \right) - \frac{1}{\sum_{i=1}^{s} s_i} \sum_{i=1}^{s} s_i \sum_{j=1}^{2n} \alpha_j y_c^j K_o \left(\vec{x_{\mathrm{j,red}}}, \vec{d_{\mathrm{i,red}}} \right) \tag{III.84}$$

$$+ \frac{1}{\sum_{i=1}^{s} s_i} \sum_{i=1}^{s} s_i \sum_{j=1}^{2n} \alpha_j y_c^j \frac{1}{\sum_{i=1}^{s} s_i} \sum_{k=1}^{s} s_k K_o \left(\vec{d_{\mathrm{k,red}}}, \vec{d_{\mathrm{i,red}}} \right) \tag{III.85}$$

where $\vec{x_{\mathrm{red}}} = (x_1, \ldots, x_m)$, $\vec{y_{\mathrm{red}}} = (y_1, \ldots, y_m)$. So we can interpret a solution as a standard δ-SVR solution with a non-symmetrical kernel, where only training examples are translated, and with b computed as above.

III.9.3 Incorporating the Linear Dependency of Function Values to ε-SVR

We will incorporate to OP 7 (III.47). Because ε-SVR is a special case of φ-SVC, and we have derived already the incorporation for φ-SVC, for ε-SVR we set the following weights

$$\varphi_i = y_c^i y_r^i - \varepsilon - 1 - y_c^i \frac{e}{\sum_{i=1}^{s} s_i} \tag{III.86}$$

for $i \in \{1, \ldots, 2n\}$. We also use input space transformation. We use φ-SVC without b.

III.10 Incorporating Inequalities with Function Values

Another example of application for tractors is the incorporation of the additional constraints in the form of inequalities with function values for training points for regression case:

$$g(\vec{x_i}) \geq a_i \tag{III.87}$$

for $i = 1..n$, where a_i are some parameters, g is defined in (I.37). In order not to introduce new optimization problems, we propose soft incorporation by changing tractor values. For ε-SVR, it leads to the modification of ε_u^i and ε_d^i values:

$$\varepsilon_{\mathrm{d,new}}^i = \max\left(0, \min\left(\varepsilon_{\mathrm{d,old}}^i, y_r^i - a_i\right)\right) \tag{III.88}$$

$$\varepsilon_{\mathrm{u,new}}^i = \min\left(\varepsilon_{\mathrm{u,old}}^i, \varepsilon_{\mathrm{d,new}}^i\right) \tag{III.89}$$

For δ-SVR we modify δ_u^i and δ_d^i parameters:

$$\delta_{\mathrm{d,new}}^i = \max\left(0, \min\left(\delta_{\mathrm{d,old}}^i, y_r^i - a_i\right)\right) \tag{III.90}$$

$$\delta_{\mathrm{u,new}}^i = \min\left(\delta_{\mathrm{u,old}}^i, \delta_{\mathrm{d,new}}^i\right) \tag{III.91}$$

Although we do not modify directly tractor parameters, they are modified indirectly for ε-SVR, because changing ε_i leads to changing φ_i. Changing δ_i can be interpreted as changing ε_i. Similar incorporation scheme exists for

$$g(\vec{x_i}) \leq a_i \ . \tag{III.92}$$

III.11 Reduce a Model with φ-SVC

Various methods for reducing the complexity of the output model were widely investigated [17]. In particular, the reduction by removing support vectors was also analyzed in [15] for regression problems.

Sparse models have an advantage of faster post-processing such as testing new examples. For some data, SVC return many support vectors. Consider highly linear nonseparable distribution of data. Many training vectors would have $\xi_i > 0$. A general idea of constructing even more sparse solutions than SVC is to find a solution spanned on the given number of support vectors

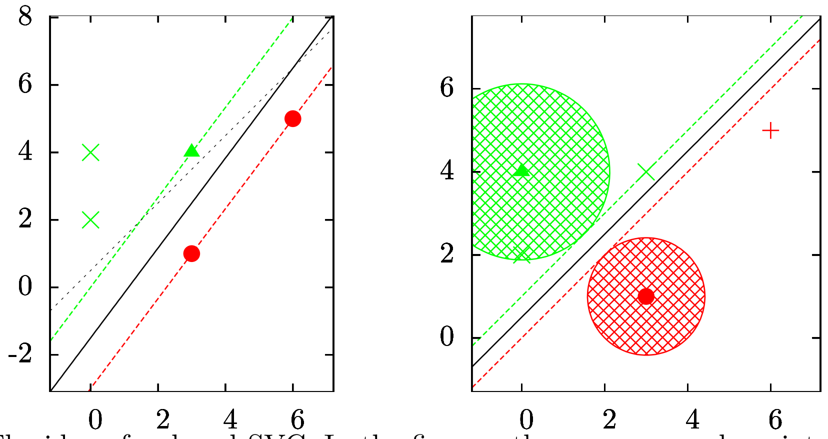

Figure III.7: The idea of reduced SVC. In the figures, there are example points after reducing, support vectors (triangles and circles), solutions (solid lines), original solutions before reduction (dotted lines), margins (dashed lines). In the right figure, there are tractors (circles filled with grid pattern)

as close to the original SVC solution as possible. The most representative method for this idea is presented in [50]. The another example for this group are reduced support vector machines (RSVM) which randomly selects support vectors to a reduced set [14, 27]. The alternative approach is to replace SVM with another training method which is designed for returning sparse solutions. The most representative method for this group is a greedy approach which adds basis functions to the solution until no progress in optimizing a cost function is made [17].

Our proposed approach [32, 34] is conceptually closer to the first idea. After SVM is trained, we remove randomly selected support vectors, and run again SVM on a reduced training set with incorporated prior knowledge about the original solution, Fig. III.7, Fig. III.8, Fig. III.9. A concept of randomly removing support vectors was presented in [15]. For highly small number of support vectors removing some of them would definitely lead to decreasing generalization performance. The proposed method generates reduced models from the original full model with incorporated prior knowledge in the form of tractors. The procedure of generating knowledge in the form of tractors is as following. First, tractors are automatically generated from an existing solution by setting

$$\varphi_i = y_i h^* (\vec{x}_i) - 1 \tag{III.93}$$

for all training examples. For ε-SVR, instead of modifying φ_i weights, we can alternatively modify ε_u^i and ε_d^i weights:

$$\varepsilon_u^i = y_r^i - g^* (\vec{x}_i) \quad ,$$

$$\varepsilon_d^i = g^* (\vec{x}_i) - y_r^i \quad .$$

After that, a reduced model is generated by removing a bunch of data vectors – randomly selected data vectors, with maximal removal ratio of $p\%$ off all training vectors, where p is a configurable parameter. Finally, we run φ-SVC with reduced data.

III.12 Generation of Prior Knowledge

When prior knowledge comes from the external source, we can directly use it for comparison of accuracy of the models. Otherwise, we have to generate prior knowledge from the available data. Authors in [25] divide a data set to two parts (about 20% and 80%), and they generate manually prior knowledge from the first, and use it in the second part. We proposed a slightly modified procedure. In order to generate meaningful prior knowledge, we generate the knowledge from the whole training set, and then we create data for the models to compare, by removing some of support vectors. The procedure is as follows:

1. Find a solution of SVM problem without prior knowledge.

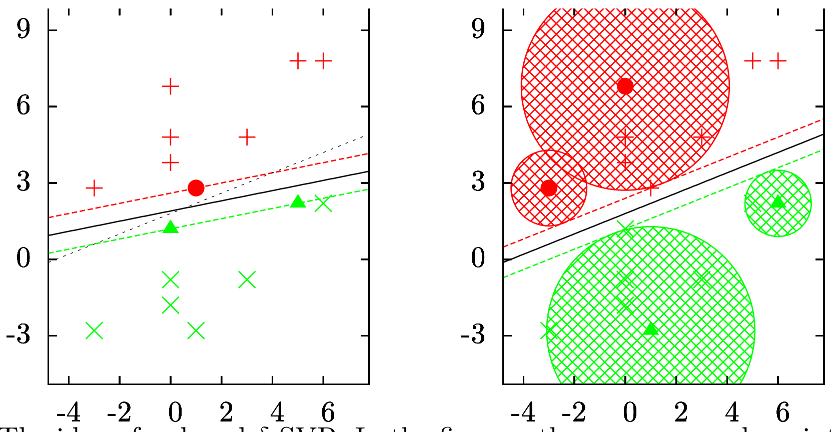

Figure III.8: The idea of reduced δ-SVR. In the figures, there are example points after reducing, support vectors (triangles and circles), solutions (solid lines), original solutions before reduction (dotted lines), margins (dashed lines). In the right figure, there are tractors (circles filled with grid pattern)

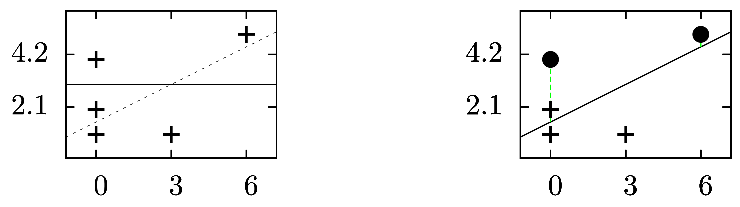

Figure III.9: The idea of reduced ε-SVR. In the figures, there are example points after reducing (pluses), support vectors (filled circles), solutions (solid lines), original solutions before reduction (dotted lines). In the right figure, there are tractors (points with dashed lines to the solution)

Table III.1: Performance of φ-SVC for reduced models for synthetic data. Column descriptions: *id* – id of the test, *a function* – a function used for generating data $y_1 = \sum_{i=1}^{\mathrm{dim}-1} x_i$, $y_4, y_5 = \left(\sum_{i=1}^{\mathrm{dim}-1} x_i\right)^{\mathrm{kerP}}$, $y_6 = 0.5 \sum_{i=1}^{\mathrm{dim}-1} \sin 10x_i + 0.5$, *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 correctly classified examples for testing data, if greater than 0 than a method with margin knowledge is better, *s1* – the average number of support vectors for a method without margin knowledge, *s2* – the average number of support vectors for a method with margin knowledge

id	function	ker	idRef	te12M	s1	s2
0	y_1	denseLinear 0.0	0	2.4	23	19
1	$y_2 = 3y_1$	denseLinear 0.0	1	3.22	23	19
2	$y_3 = 1/3y_1$	denseLinear 0.0	2	-0.63	24	21
3	y_4	densePolynomial 5.0	3	1.19	23	16
4	y_5	denseRBF 0.25	4	0.92	25	20
5	y_6	denseRBF 0.25	5	0.43	26	25

2. Extract prior knowledge from the solution.
3. Remove randomly $p\%$ of input data.
4. Find the solution with and without prior knowledge on reduced data set.

III.13 Experiments

In experiments, we show that the reduced models with margin knowledge have better performance than without the additional knowledge. The first method does not use margin knowledge in reduced models, the second one use the additional knowledge. In the first experiment, we set arbitrarily $p = 70$. Note that for comparison purposes a reduced model is the same for both methods. We use the author implementation of SVC for both methods. In the second experiment, we show that the proposed method has better performance for variable p.

For all data sets, every feature is scaled linearly to $[0, 1]$ including an output. For variable parameters like the C, σ for the RBF kernel, φ for δ-SVR, and ε for ε-SVR we use a grid search method for finding 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 set it is possible to use more accurate grid searches than for massive tests with the multiple number of simulations.

III.13.1 Synthetic Data Tests

We compare both methods on data generated from particular functions with added Gaussian noise for output values. We perform 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 tests with results are presented in Table III.1. The method with margin knowledge has better performance for every kernel, the number of support vectors is comparable. A testing performance gain varies from 0% to 51%.

III.13.2 Real World Data Tests

The real world data sets were taken from the LibSVM site [3] [22] except stock price data. The stock price data consist of monthly prices of the DJIA index from 1898 up to 2010. We generated the training set as follows: for every month the output value is a growth/fall comparing to the

Table III.2: Performance of φ-SVC for reduced models 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 percent average difference in correctly classified examples for testing data, if greater than 0 than a method with margin knowledge is better, *s1* – the average number of support vectors for a method without margin knowledge, *s2* – the average number of support vectors for a method with margin knowledge

id	dn	ker	idRef	te12M	s1	s2
0	a1aAll	denseLinear 0.0	0	15.47	16	16
1	a1aAll	densePolynomial 5.0	1	3.79	11	25
2	a1aAll	denseRBF 0.00813	2	0.0	27	27
3	breast-cancer	denseLinear 0.0	3	14.26	7	8
4	breast-cancer	densePolynomial 3.0	4	24.71	5	6
5	breast-cancer	denseRBF 0.1	5	3.61	24	25
6	diabetes	denseLinear 0.0	6	9.41	20	17
7	diabetes	densePolynomial 3.0	7	7.16	16	15
8	diabetes	denseRBF 0.125	8	0.42	26	26
9	djia	denseLinear 0.0	9	0.67	23	16
10	djia	densePolynomial 5.0	10	1.26	22	16
11	djia	denseRBF 0.08333	11	1.52	29	28

next month. Every feature i is a percent price change between the month and the i-th previous month. In every simulation, training data are randomly chosen, the remaining examples become test data. The tests with results are presented in Table III.2. The method with margin knowledge has better performance for all data sets, for all kernels with similar number of support vectors. The testing performance varies from 0% to 27%. For the DJIA data set, results are comparable.

III.13.3 Variable p

In the second experiment, we show that for variable p the second method has better performance. The example results for the first test case from synthetic tests are depicted in Fig. III.10, Fig. III.11.

III.13.4 Experiments for Regression

In the first experiment, we show that the reduced models with margin knowledge have better generalization performance than without that knowledge for various p. The first method does not use margin knowledge in reduced models, the second one use it. In the third experiment, we show that the reduced models have much better time of testing new examples which mainly depends on the number of support vectors. Note that for purposes of fair comparison training data of a reduced model is the same for both methods. We use the author implementation of reformulation of the ε-SVR for both methods.

For all data sets, every feature is scaled linearly to $[0, 1]$ including an output. For ε we use a grid search method for finding the best values.

Comparing generalization performance of reduced model.

The synthetic samples were generated from particular functions with added Gaussian noise for output values. The real world data sets were taken from the LibSVM site [3][22] except stock price data. They originally come from UCI Machine Learning Repository and StatLib DataSets Archive. The stock price data consist of monthly prices of the DJIA index from 1898 up to 2010.

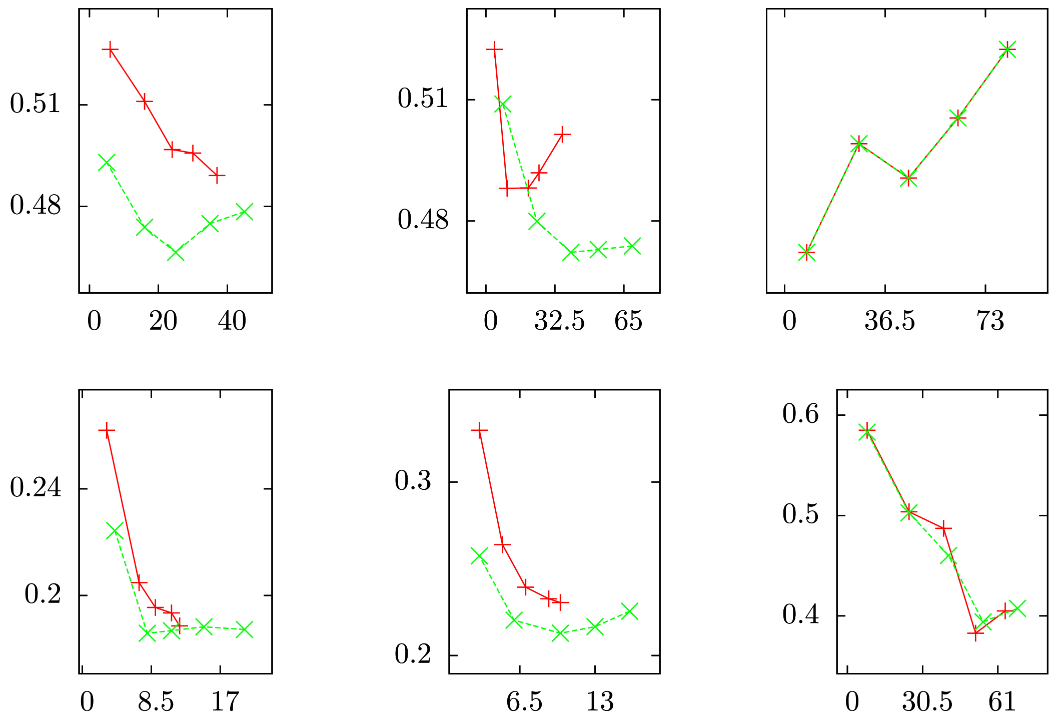

Figure III.10: Comparison of two methods of removing support vectors for the test cases from Table III.2. On x axis there is a number of support vectors, on y axis there is a percent difference in misclassified testing examples. The line with '+' points represents a random removing method, while the line with 'x' points represents proposed removing method with margin knowledge

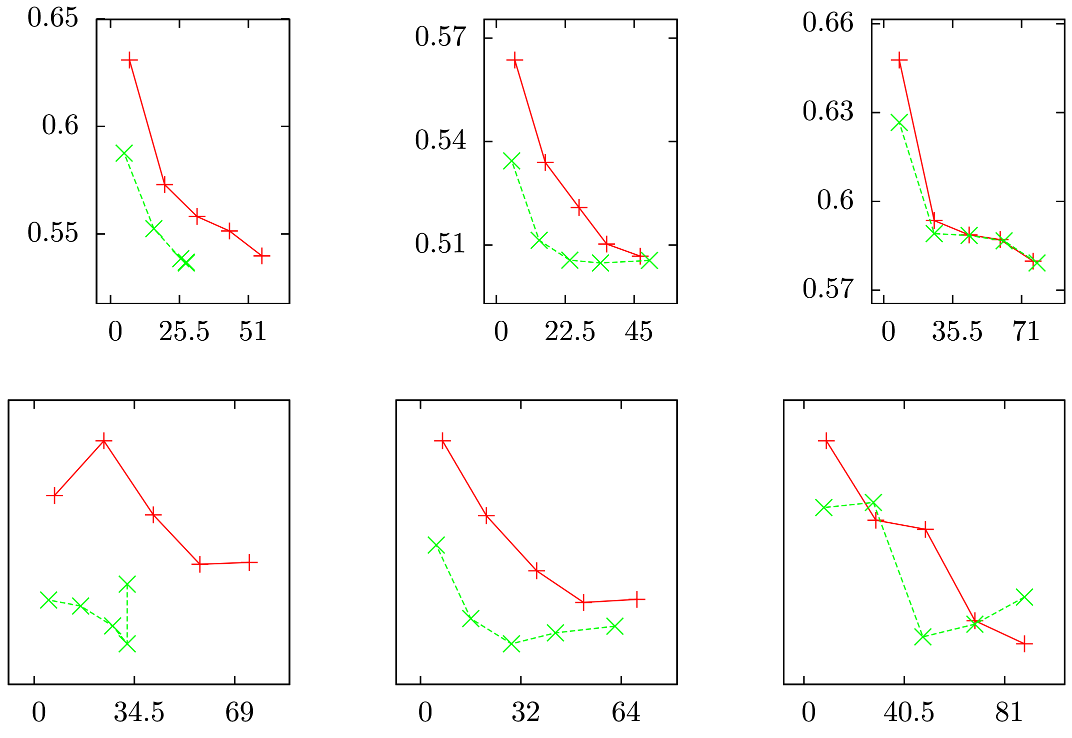

Figure III.11: Comparison of two methods of removing support vectors for the test cases from from Table III.2, cont. On x axis there is a number of support vectors, on y axis there is a percent difference in misclassified testing examples. The line with '+' points represents a random removing method, while the line with 'x' points represents proposed removing method with margin knowledge

Table III.3: Performance of φ-SVC for reduced models for synthetic data, for regression. Column descriptions: *id* – id of the test, *a function* – a function used for generating data $y_1 = \sum_{i=1}^{\dim-1} x_i$, $y_4, y_5 = \left(\sum_{i=1}^{\dim-1} x_i\right)^{\text{kerP}}$, $y_6 = 0.5 \sum_{i=1}^{\dim-1} \sin 10x_i + 0.5$, *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 a method with margin knowledge is better, *s1* – the average number of support vectors for a method without margin knowledge, *s2* – the average number of support vectors for a method with margin knowledge

id	function	ker	idRef	te12M	s1	s2
0	y_1	denseLinear 0.0	0	50.05	11	20
1	y_2	densePolynomial 5.0	1	0.98	42	52
2	y_3	denseRBF 0.25	2	4.6	41	54

We generated the sample data set 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. In every simulation, training data are randomly chosen, the remaining examples become test data. For $p = 70$, $C = 0.1$ the method with margin knowledge has better performance in all tests with similar number of support vectors (columns *s1* and *s2*), Table III.3, Table III.4. The testing performance improvement varies from 0% to 68%. For variable p the proposed method is also superior, example results for the cadata1 test are depicted in Fig. III.12, Fig. III.13.

Comparing testing speed performance of reduced model.

Testing speed of new examples depends mainly on the number of support vectors. With fewer support vectors we achieve testing time reduction.

III.14 Summary

In this thesis, we analyzed applicability of margin knowledge per example incorporated to SVM for lowering generalization error of reduced models. The method was tested for SVM classifier, ε-SVR and δ-SVR. Experiments on real world data sets show smaller generalization error for reduced models with margin knowledge. A potential list of applications for margin knowledge is much broader. In future research, we plan to investigate possibility to create margin knowledge by experts and to apply margin knowledge for time series data. Moreover, we plan to use margin knowledge for combining SVM classifiers with each other in order to decrease a generalization error.

Table III.4: Performance of φ-SVC for reduced models for real world data, for regression. Column descriptions: *id* – id of the test, *a function* – a function used for generating data $y_1 = \sum_{i=1}^{\dim-1} x_i$, $y_4, y_5 = \left(\sum_{i=1}^{\dim-1} x_i\right)^{\text{kerP}}$, $y_6 = 0.5 \sum_{i=1}^{\dim-1} \sin 10x_i + 0.5$, *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 a method with margin knowledge is better, *s1* – the average number of support vectors for a method without margin knowledge, *s2* – the average number of support vectors for a method with margin knowledge

id	dn	ker	idRef	te12M	s1	s2
0	abalone	denseLinear 0.0	0	11.39	12	30
1	abalone	densePolynomial 5.0	1	34.86	14	48
2	abalone	denseRBF 0.125	2	11.11	18	53
3	cadata	denseLinear 0.0	3	14.08	25	47
4	cadata	densePolynomial 5.0	4	16.38	26	52
5	cadata	denseRBF 0.125	5	11.41	34	53
6	djia	denseLinear 0.0	6	5.81	9	26
7	djia	densePolynomial 5.0	7	34.16	13	48
8	djia	denseRBF 0.1	8	0.2	8	50
9	housing	denseLinear 0.0	9	19.47	17	29
10	housing	densePolynomial 5.0	10	18.39	19	53
11	housing	denseRBF 0.077	11	1.33	30	53

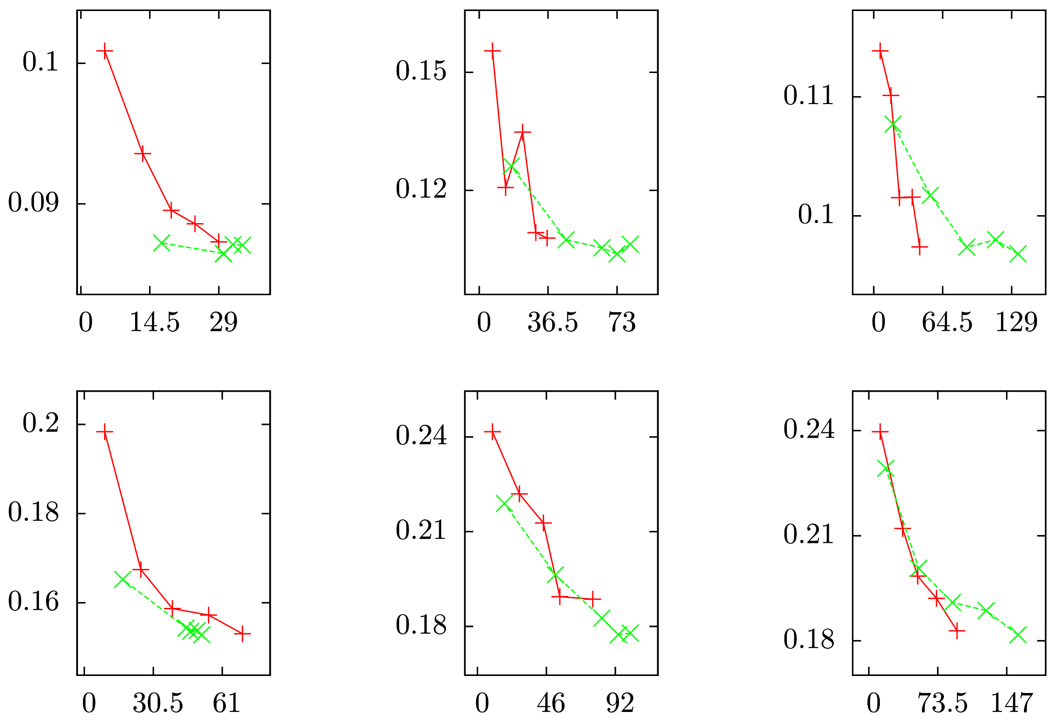

Figure III.12: Comparison of two methods of removing support vectors for the test cases from Table III.4. On x axis there is a number of support vectors, on y axis there is a percent difference in misclassified testing examples. The line with '+' points represents a random removing method, while the line with 'x' points represents proposed removing method with margin knowledge

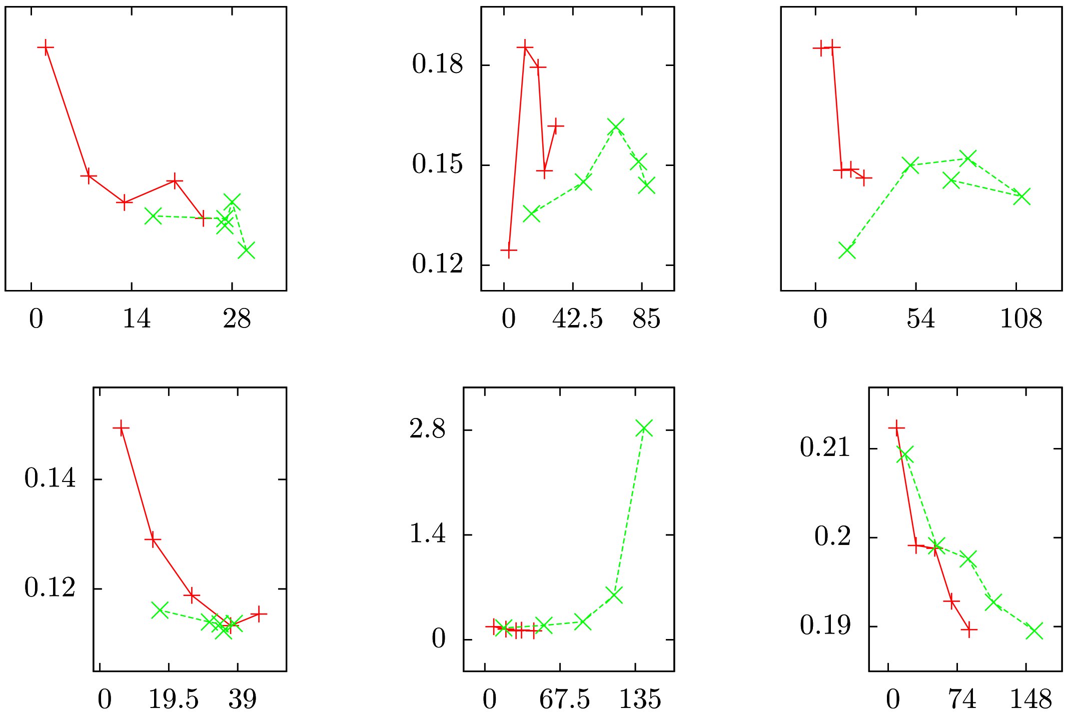

Figure III.13: Comparison of two methods of removing support vectors for the test cases from from Table III.4, cont. On x axis there is a number of support vectors, on y axis there is a percent difference in misclassified testing examples. The line with '+' points represents a random removing method, while the line with 'x' points represents proposed removing method with margin knowledge

Chapter IV

Implementation Techniques Based on KKT Conditions

One of the category of methods used for solving OP 3 are subset selection methods. In every iteration only a few number of Lagrange multipliers are optimized. The special case is SMO method proposed in [36] which solves 2-parameter subproblems analytically in every iteration. For subproblems with more than 2 parameters, general quadratic programming solvers are used. We propose using SMO for solving subproblems with more than 2 parameters. The advantage of such solver is a simpler method without external quadratic programming solvers.

The important part of subset selection methods is strategy for choosing parameters in every iteration. The most popular strategy is based on KKT criterion. We propose a strategy which computes a value of cost function in every iteration for a view alternative pairs of parameters. We choose a pair for which we achieve the biggest decrease of the cost function. The advantage of this strategy is the decreased number of iterations and possibility to effective parallelization of the algorithm.

We can use all proposed method with SVC, and therefore also with δ-SVR. They also work with φ-SVC, so we can use them with ε-SVR as well.

IV.1 Introduction to SMO

SMO is a well established method described in [39, 8]. In a subset selection method in every iteration we solve the following optimization problem

OP 17.

$$\begin{aligned}\max_{\vec{\beta}} \quad f_2\left(\vec{\beta}\right) = \sum_{i=1}^{p} \beta_i + \varphi_{c_i} + \sum_{\substack{i=1\\ i\notin P}}^{n} \alpha_i + \varphi_i - \tfrac{1}{2} \sum_{i=1}^{p} y_{c_i}\beta_i \sum_{j=1}^{p} y_{c_j}\beta_j K_{c_i c_j} \\ - \sum_{i=1}^{p} y_{c_i}\beta_i \sum_{\substack{j=1\\ j\notin P}}^{n} y_j\alpha_j K_{c_i j} - \tfrac{1}{2} \sum_{\substack{i=1\\ i\notin P}}^{n} \sum_{\substack{j=1\\ j\notin P}}^{n} y_{ij}\alpha_i\alpha_j K_{ij}\end{aligned} \tag{IV.1}$$

subject to

$$\sum_{i=1}^{p} y_{c_i}\beta_i + \sum_{\substack{i=1\\ i\notin P}}^{n} y_i\alpha_i = 0 \tag{IV.2}$$

$$0 \le \beta_i \le C \tag{IV.3}$$

for $i \in \{1, 2, \ldots, p\}$, where $P = \{c_1, \ldots, c_p\}$ is a set of indices of parameters chosen to the working set, $c_i \in I$, $c_i \neq c_j$ for $i \neq j$, $\vec{\beta}$ is a subproblem variable vector, β_i is a searched value of c_i parameter. The vector α is a previous solution of OP 15. It must fulfill the linear constraint.

SMO solves 2 parameter subproblems in every iteration step. So SMO solves a special case of OP 17 when $p = 2$. The solution is

$$\beta_2^{unc} = \alpha_{c_2} + \frac{y_{c_2}\left(E_{c_1} - E_{c_2}\right)}{\kappa} \tag{IV.4}$$

$$\beta_2 = \begin{cases} V,\ if\ \beta_2^{unc} > V \\ \beta_2^{unc},\ if\ U \le \beta_2^{unc} \le V \\ U,\ if\ \beta_2^{unc} < U \end{cases} \tag{IV.5}$$

$$\beta_1 = \alpha_{c_1} + y_{c_1} y_{c_2}\left(\alpha_{c_2} - \beta_2\right) \quad , \tag{IV.6}$$

where

$$E_i = \sum_{j=1}^{n} y_j \alpha_j K\left(\vec{x}_i, \vec{x}_j\right) - y_i - y_i \varphi_i \tag{IV.7}$$

for $i \in \{1, \dots, n\}$

$$\kappa = K_{c_1 c_1} + K_{c_2 c_2} - 2K_{c_1 c_2} \tag{IV.8}$$

when $y_{c_1} \neq y_{c_2}$

$$U = \max\left(0, \alpha_{c_2} - \alpha_{c_1}\right) \tag{IV.9}$$

$$V = \min\left(C_2, C_1 - \alpha_{c_1} + \alpha_{c_2}\right) \tag{IV.10}$$

when $y_{c_1} = y_{c_2}$

$$U = \max\left(0, \alpha_{c_1} + \alpha_{c_2} - C_1\right) \tag{IV.11}$$

$$V = \min\left(C_2, \alpha_{c_1} + \alpha_{c_2}\right) \quad . \tag{IV.12}$$

IV.1.1 SMO for SVM without the offset

The SMO can be defined for SVC without the offset. The difference is that the minimal number of parameters that can be optimized in every step is just one parameter:

$$\alpha_1^{\text{new}} = \alpha_1 - \frac{y_1 E_1}{K_{11}} \quad . \tag{IV.13}$$

Then we have to bound α_1^{new}:

$$0 \le \alpha_1^{\text{new}} \le C_1 \quad . \tag{IV.14}$$

See Appendix E.2 and Appendix E.3.

IV.2 Multidimensional Heuristics

The most popular heuristic for a subset selection method is based on choosing the parameters based on KKT conditions. We present two possible derivations of the heuristic directly from KKT and by more insight analysis with incorporating the linear constraint to the cost function. Multidimensional heuristic was proposed in [12]. In every step we have to choose p parameters.

IV.2.1 Optimization Conditions

. We can transform the linear constraint (IV.2) to a form

$$\beta_d = -y_{c_d} \sum_{\substack{i=1 \\ i \neq d}}^{p} y_{c_i} \beta_i - y_{c_d} \sum_{\substack{i=1 \\ i \notin C}}^{n} y_i \alpha_i \quad , \tag{IV.15}$$

where $d \in \{1, 2, \dots, p\}$ is an arbitrarily chosen parameter. After substituting β_d to the (IV.1) we get the following optimization subproblem

OP 18.

$$
\begin{aligned}
\max_{\vec{\gamma}} \quad & f_3\left(\vec{\gamma}\right) = -y_{c_d} \sum_{\substack{i=1\\i\neq d}}^{p} y_{c_i}\gamma_{e_i} - y_{c_d} \sum_{\substack{i=1\\i\notin C}}^{n} y_i\alpha_i + \sum_{\substack{i=1\\i\neq d}}^{p} \gamma_{e_i} \\
& + \sum_{\substack{i=1\\i\notin C}}^{n} \alpha_i - \frac{1}{2}\left(\sum_{\substack{i=1\\i\neq d}}^{p} y_{c_i}\gamma_{e_i} + \sum_{\substack{i=1\\i\notin C}}^{n} y_i\alpha_i\right)^2 K_{c_d c_d} \\
& + \left(\sum_{\substack{i=1\\i\neq d}}^{p} y_{c_i}\gamma_{e_i} + \sum_{\substack{i=1\\i\notin C}}^{n} y_i\alpha_i\right) \sum_{\substack{i=1\\i\neq d}}^{p} y_{c_i}\gamma_{e_i} K_{c_d c_i} \\
& - \frac{1}{2} \sum_{\substack{i=1\\i\neq d}}^{p} y_{c_i}\gamma_{e_i} \sum_{\substack{j=1\\j\neq d}}^{p} y_{c_j}\gamma_{e_j} K_{c_i c_j} \\
& + \left(\sum_{\substack{i=1\\i\neq d}}^{p} y_{c_i}\gamma_{e_i} + \sum_{\substack{i=1\\i\notin C}}^{n} y_i\alpha_i\right) \sum_{\substack{i=1\\i\notin C}}^{n} y_i\alpha_i K_{c_d i} \\
& - \sum_{\substack{i=1\\i\neq d}}^{p} y_{c_i}\gamma_{e_i} \sum_{\substack{j=1\\j\notin C}}^{n} y_j\alpha_j K_{c_i j} - \frac{1}{2} \sum_{\substack{i=1\\i\notin C}}^{n} \sum_{\substack{j=1\\j\notin C}}^{n} y_{ij}\alpha_i\alpha_j K_{ij}
\end{aligned}
\tag{IV.16}
$$

subject to

$$0 \leq \gamma_{e_i} \leq C, \text{ for } i \in \{1, 2, \ldots, p\} \setminus \{d\}\,,\ C > 0 \tag{IV.17}$$

$$0 \leq c = -y_{c_d} \sum_{\substack{i=1\\i\neq d}}^{p} y_{c_i}\gamma_{e_i} - y_{c_d} \sum_{\substack{i=1\\i\notin C}}^{n} y_i\alpha_i \leq C \quad , \tag{IV.18}$$

where
$\vec{\gamma}$ is a $p-1$ elements variable vector,
$e_i = i$ for $i < d$,
$e_i = i - 1$ for $i > d$,
γ_{e_i} is a searched value of c_i parameter,
c is a searched value of c_d parameter.
The vector α is a previous solution. It must fulfill the constraints from O_1 problem.

The partial derivative of $f_3\left(\vec{\gamma}\right)$ in the point for which $\gamma_{e_i} = \alpha_{c_i}$ has a value

$$\frac{\partial}{\partial \gamma_{e_k}} f_3\left(\vec{\gamma}_{\text{old}}\right) = y_{c_k}\left(E_{c_d} - E_{c_k}\right) \quad , \tag{IV.19}$$

where

$$E_i = \sum_{j=1}^{n} y_j\alpha_j K_{ij} - y_i \quad . \tag{IV.20}$$

Let's analyze conditions for optimization possibility. The first obvious necessary condition is that one of all parameters must change its value. The remaining optimization conditions consist of two parts. The first part consists of conditions based on fulfilling (IV.18), the second part consists of conditions based on partial derivatives. Merging all conditions leads to the overall optimization conditions.

(IV.18) must be fulfilled after changes, hence we can write

$$
\begin{aligned}
& -\alpha_{c_d}^{\text{old}} \leq \Delta\alpha_{c_d} = -\alpha_{c_d}^{\text{old}} - y_{c_d} \sum_{\substack{i=1\\i\neq d}}^{p} y_{c_i}\alpha_{c_i}^{\text{new}} \\
& -y_{c_d} \sum_{\substack{i=1\\i\notin C}}^{n} y_i\alpha_i \leq C - \alpha_{c_d}^{\text{old}} \quad .
\end{aligned}
\tag{IV.21}
$$

After substituting

$$\alpha_{c_d}^{\text{old}} = -y_{c_d} \sum_{\substack{i=1\\i\neq d}}^{p} y_{c_i}\alpha_{c_i}^{\text{old}} - y_{c_d} \sum_{\substack{i=1\\i\notin C}}^{n} y_i\alpha_i \tag{IV.22}$$

we get the following condition

$$-\alpha_{c_d}^{\text{old}} \le \Delta\alpha_{c_d} = -y_{c_d} \sum_{\substack{i=1\\ i\neq d}}^{p} y_{c_i} \Delta\alpha_{c_i} \le C - \alpha_{c_d}^{\text{old}} \ . \tag{IV.23}$$

Theorem IV.2.1 (Necessary optimization conditions based on fulfilling (IV.18)). *If the condition* (IV.23) *is fulfilled, then there exist two parameters c_i, where $i \in \{1, \ldots, p\}$, which belong to the opposite groups G_1 and G_2 defined as*

$$\begin{aligned} &G_1 := \{i \in \{1, 2, \ldots, n\} : (y_i = 1 \wedge \alpha_i = 0) \\ &\vee (y_i = -1 \wedge \alpha_i = C) \vee (0 < \alpha_i < C)\} \\ &G_2 := \{i \in \{1, 2, \ldots, n\} : (y_i = -1 \wedge \alpha_i = 0) \\ &\vee (y_i = 1 \wedge \alpha_i = C) \vee (0 < \alpha_i < C)\} \ . \end{aligned} \tag{IV.24}$$

Note that nonbound parameters are included in both groups.

Proof. We prove that, if all parameters belong to only one group G_1 or G_2, then the condition (IV.23) will not be fulfilled. We choose parameters belong to the G_1 group. The proof for G_2 group is similar. The set of chosen parameters does not contain any nonbound parameters, because they belong to the both groups. If all c_i parameters for $i \in \{1, 2, \ldots, p\} \setminus \{d\}$ does not change, then $\sum_{\substack{i=1\\ i\neq d}}^{p} y_{c_i}\Delta\alpha_{c_i} = 0$ and therefore $\Delta\alpha_{c_d} = 0$; so the set of all parameters c_i does not change what cannot be true. Otherwise the following holds: $\sum_{\substack{i=1\\ i\neq d}}^{p} y_{c_i}\Delta\alpha_{c_i} > 0$. If $y_{c_d} = 1$, then $\Delta\alpha_{c_d} < 0$ and $\alpha_{c_d}^{\text{old}} = 0$. The condition (IV.23) becomes $0 \le \Delta\alpha_{c_d} \le C$, what cannot be true. If $y_{c_d} = -1$, then $\Delta\alpha_{c_d} > 0$ and $\alpha_{c_d}^{\text{old}} = C$. The condition (IV.23) becomes $-C \le \Delta\alpha_{c_d} \le 0$ what cannot be true. □

Theorem IV.2.2 (Sufficient optimization conditions based on fulfilling (IV.18)). *If there exist two parameters c_i, where $i \in \{1, \ldots, p\}$, which belong to the opposite groups G_1 and G_2, then condition* (IV.23) *is fulfilled for some parameter changes.*

Proof. If none of chosen two parameters (c_a from G_1 group and c_b from G_2 group) is c_d parameter, then we can set $\Delta\alpha_{c_a}$ and $\Delta\alpha_{c_b}$ to the same values or with inverse signs, in the way that $\Delta\alpha_{c_d} = 0$ so (IV.23) is fulfilled. If the chosen parameters are c_d parameter from G_1 group and c_b parameter from G_2 group, then when we set all remaining parameter changes to zero the following can hold: $\sum_{\substack{i=1\\ i\neq d}}^{p} y_{c_i}\Delta\alpha_{c_i} < 0$. If $y_{c_d} = 1$, then $\Delta\alpha_{c_d} > 0$. If $\alpha_{c_d}^{\text{old}} = 0$, then condition (IV.23) is obviously fulfilled. If $0 < \alpha_{c_d}^{\text{old}} < C$, then condition (IV.23) is fulfilled, when $\Delta\alpha_{c_b}$ is set to close enough to zero value. If $y_{c_d} = -1$, then $\Delta\alpha_{c_d} < 0$. If $\alpha_{c_d}^{\text{old}} = C$, then condition (IV.23) is obviously fulfilled. If $0 < \alpha_{c_d}^{\text{old}} < C$, then condition (IV.23) is fulfilled, when $\Delta\alpha_{c_b}$ is set to close enough to zero value. □

Theorem IV.2.3 (Necessary optimization conditions based on partial derivatives). *If optimization is possible based on partial derivatives, then one of the partial derivatives of the function f_3 must fulfill the following condition*

$$\begin{aligned} &\frac{f_3(\vec{\gamma})}{\gamma_{e_k}} > 0 \text{ when } \alpha_{c_k} = 0 \\ &\frac{f_3(\vec{\gamma})}{\gamma_{e_k}} < 0 \text{ when } \alpha_{c_k} = C \\ &\frac{f_3(\vec{\gamma})}{\gamma_{e_k}} \neq 0 \text{ when } 0 < \alpha_{c_k} < C \ . \end{aligned} \tag{IV.25}$$

Proof. We prove that if all partial derivatives of the function f_3 do not fulfill the condition (IV.25), then optimization will not be possible. If (IV.25) is not fulfilled, then objective function

f_3 can't increase its value in any direction and therefore the function f_3 can't increase its value at all. □

Corollary IV.2.1. *After substitution* (IV.19) *to* (IV.25) *we get*
$y_{c_k}\left(E_{c_d} - E_{c_k}\right) > 0$ *when* $\alpha_{c_k} = 0$
$y_{c_k}\left(E_{c_d} - E_{c_k}\right) < 0$ *when* $\alpha_{c_k} = C$
$y_{c_k}\left(E_{c_d} - E_{c_k}\right) \neq 0$ *when* $0 < \alpha_{c_k} < C$
After simplification:
When $y_{c_k} = 1$*:*
$E_{c_k} < E_{c_d}$ *when* $\alpha_{c_k} = 0$
$E_{c_k} > E_{c_d}$ *when* $\alpha_{c_k} = C$
$E_{c_k} \neq E_{c_d}$ *when* $0 < \alpha_{c_k} < C$
When $y_{c_k} = -1$*:*
$E_{c_k} > E_{c_d}$ *when* $\alpha_{c_k} = 0$
$E_{c_k} < E_{c_d}$ *when* $\alpha_{c_k} = C$
$E_{c_k} \neq E_{c_d}$ *when* $0 < \alpha_{c_k} < C$

Theorem IV.2.4 (Sufficient optimization conditions based on partial derivatives)**.** *If one of the partial derivatives of the function f_3 fulfills the condition* (IV.25)*, then optimization is possible based on partial derivatives for some parameter changes.*

Proof. We can change the parameter which fulfills the condition (IV.25). The remaining parameters which are attributed to f_3 variables can stay unchanged, and then f_3 value will grow. □

Theorem IV.2.5 (Overall optimization conditions)**.** *Optimization is possible for some parameter changes, if and only if there exist two parameters c_i, where $i \in \{1, 2, \ldots, p\}$, which belong to the opposite groups G_1 and G_2 and one of the partial derivatives of the function f_3 fulfills the condition* (IV.25)*.*

Proof. Because of Thm. IV.2.1, Thm. IV.2.2, Thm. IV.2.3, Thm. IV.2.4 we only have to prove, that overall optimization is a multiplication of optimization based on (IV.18) and based on partial derivatives. This can be shown in the terms of multidimensional functions with set of linear constraints and one nonlinear constraint. Multidimensional function f_3 can be optimized when conditions with derivatives are fulfilled with respect to the linear conditions. There is additionaly only one nonlinear constraint. When it is also fulfilled, then optimization is possible. □

IV.2.2 Choosing Two Parameters to a Working Set

In the standard multidimensional heuristic, we do not graduate the condition based on (IV.18) and therefore we only want to fulfill this condition.

Using the optimization conditions based on partial derivatives, we choose those two parameters which maximize $m_{c_d c_k}$ defined as
when parameter c_k is bound and belongs to the G_1 group, then

$$m_{c_d c_k} := E_{c_d} - E_{c_k} \quad , \tag{IV.26}$$

when parameter c_k is bound and belongs to the G_2 group, then

$$m_{c_d c_k} := E_{c_k} - E_{c_d} \quad , \tag{IV.27}$$

when parameter c_k is non bound, then

$$m_{c_d c_k} := \left|E_{c_k} - E_{c_d}\right| \quad . \tag{IV.28}$$

The conclusion from above is that the best two parameters to optimize will be with minimal E from G_1 group and with maximal E from G_2 group, if the chosen parameters are different.

IV.2.3 Choosing Remaining Parameters

One of the already chosen parameters is c_d parameter. In order to maximize $m_{c_d c_k}$ for the remaining parameters, that have to be chosen, we choose them from the opposite group to the group with c_d parameter: when α_{c_d} belongs to the G_1 group, then the remaining parameters are chosen from the G_2 group, when α_{c_d} belongs to the G_2 group, then the remaining parameters are chosen from the G_1 group.

After choosing two parameters, c_d parameter can be one of them. So we choose remaining parameters either from G_1 group or from G_2 group, and then compare both cases by summing m_i values for all chosen parameters. The case with maximal sum of m_i is finally chosen.

IV.2.4 Alternative Derivation from KKT Conditions

We can derive the presented above heuristic directly from KKT conditions (III.15), (III.16). We have the following cases:

- When $\alpha_i = 0$, then from (III.16), $\xi_i = 0$. From (III.15)

$$y_i \left(w x_i + b \right) \geq 1 \tag{IV.29}$$

when $y_i = 1$

$$b \geq 1 - w x_i \tag{IV.30}$$

when $y_i = -1$

$$b \leq -w x_i - 1 \ . \tag{IV.31}$$

- When $\alpha_i = C$

$$y_i \left(w x_i + b \right) - 1 + \xi_i = 0 \tag{IV.32}$$

$$\xi_i = -y_i \left(w x_i + b \right) + 1 \ . \tag{IV.33}$$

Because $\xi_i \geq 0$, so

$$- y_i \left(w x_i + b \right) + 1 \geq 0 \tag{IV.34}$$

$$y_i \left(w x_i + b \right) \leq 1 \tag{IV.35}$$

when $y_i = 1$

$$b \leq 1 - w x_i \tag{IV.36}$$

when $y_i = -1$

$$b \geq -w x_i - 1 \ . \tag{IV.37}$$

- When $0 < \alpha_i < C$, then $\xi_i = 0$ and

$$y_i \left(w x_i + b \right) - 1 = 0 \tag{IV.38}$$

$$b = -w x_i + y_i \ . \tag{IV.39}$$

We can transform above equations to a dual form because

$$w_i = \sum_k \alpha_k y_k x_{ik} \tag{IV.40}$$

$$w x_i = \sum_j x_{ij} \sum_k \alpha_k y_k x_{kj} = \sum_k \sum_j \alpha_k y_k x_{kj} x_{ij} = \sum_k \alpha_k y_k K_{ki} \ . \tag{IV.41}$$

With the notation

$$E_i = \sum_j y_j \alpha_j K_{ij} - y_i \tag{IV.42}$$

we get

$$w x_i = E_i + y_i \ . \tag{IV.43}$$

Inequality constraints with symbol E_i are

- when $\alpha_i = 0$ and y = 1
$$b \geq -E_i \tag{IV.44}$$
when y = -1
$$b \leq -E_i \ , \tag{IV.45}$$
- when $\alpha_i = C$ and $y = 1$
$$b \leq -E_i \tag{IV.46}$$
when y = -1
$$b \geq -E_i \ , \tag{IV.47}$$
- when $0 < \alpha_i < C$
$$b = -E_i \ . \tag{IV.48}$$

From the above we can conclude the same optimization conditions as derived in the previous part.

IV.3 Sequential Multidimensional Subsolver

There were two basic methods for solving SVM subproblems. A new, third method was recently proposed [29].

1. Solve 2 parameter subproblems analytically (SMO algorithm).
2. Solve more than 2 parameter subproblems with a general quadratic programming solver (Multiparameter General Solver).
3. Solve more than 2 parameter subproblems with SMO algorithm (SMS).

The third option will be analyzed here.

IV.3.1 Comparison of SMS with Multiparameter General Solvers

In the second method, subproblems are solved by quadratic programming solvers (for example *interior point method* solver, see [43]) which have computation times independent from the SVM problem length, but dependent on a subproblem length: $O(p)$. Also the third solution solves subproblems with computation time $O(p)$. However, the proposed solver is less complicated and easier to implement.

IV.3.2 Comparison of SMS with SMO

The second and third solvers solve more than 2 parameter subproblems. In practice, generally SVM problems are computed faster with the second and third solvers than with the first one.

We can deduce primal optimization problem for OP 17

OP 19.

$$\min_{\vec{w},b,\vec{\xi}} \ f\left(\vec{w},b,\vec{\xi}\right) = \|\vec{w}\|^2 + C\sum_{i=1}^{n}\xi_i - \sum_{\substack{i=1\\ i\notin P}}^{n}\alpha_i\left(y_i h\left(\vec{x_i}\right) - 1 + \xi_i\right) - \sum_{\substack{i=1\\ i\notin P}}^{n}\left(C-\alpha_i\right)\xi_i \tag{IV.49}$$

subject to

$$y_{c_i} h\left(\vec{x_{c_i}}\right) \geq 1 - \xi_{c_i} + \varphi_i \tag{IV.50}$$
$$\xi_{c_i} \geq 0 \tag{IV.51}$$

for $i \in \{1,\ldots,p\}$, where $h\left(\vec{x_i}\right) = \vec{w}\cdot\vec{x_i} + b$ for $i \in \{1,\ldots,n\}$.

We can notice differences that the set of constraints is limited to subproblem variables, and additional terms are present in a cost function. We can notice that KKT conditions for OP 19 are similar to (III.15), (III.16)

$$\begin{cases} \beta_i \left(y_{c_i} h\left(\vec{x_{c_i}}\right) - 1 - \varphi_{c_i} + \xi_{c_i}\right) = 0 \\ \left(C - \beta_i\right) \xi_{c_i} = 0 \end{cases} \tag{IV.52}$$

for $i \in \{1, \dots, p\}$,

$$w_i = \sum_{j=1}^{n} \alpha_j y_c^j x_{ij} \tag{IV.53}$$

for $i \in \{1, \dots, m\}$,

$$\sum_{i=1}^{p} \beta_i y_{c_i} + \sum_{\substack{i=1 \\ i \notin P}}^{n} y_i \alpha_i = 0 \ . \tag{IV.54}$$

KKT complementary conditions are the same as for OP 13, but only for variable parameters. The linear condition is also slightly different, because it contains an information that the linear constraint for OP 13 should be fulfilled. Nonetheless, it is possible to use slightly modified SMO method for solving OP 17. We have to two variables γ_1 and γ_2 for e_1 and e_2 indexes respectively where $e_1, e_2 \in \{1, \dots, n\}$ and d_1, d_2 indexes in P respectively where $d_1, d_2 \in \{1, \dots, p\}$. All values of β before running SMO are set that $\beta_i = \alpha_{c_i}$ for $i = \{1, \dots, p\}$. A solution for 2 parameter subproblems for OP 17 is almost the same as for SMO

$$\gamma_2^{unc} = \beta_{d_2} + \frac{y_{c_2}\left(E_{e_1} - E_{e_2}\right)}{\kappa} \tag{IV.55}$$

$$\gamma_2 = \begin{cases} V, \ if \ \gamma_2^{unc} > V \\ \gamma_2^{unc}, \ if \ U \leq \gamma_2^{unc} \leq V \\ U, \ if \ \gamma_2^{unc} < U \end{cases} \tag{IV.56}$$

$$\gamma_1 = \beta_{d_1} + y_{e_1} y_{e_2} \left(\beta_{d_2} - \gamma_2\right) \ , \tag{IV.57}$$

where

$$E_i = \sum_{j=1}^{p} y_{c_j} \beta_j K\left(\vec{x_{c_i}}, \vec{x_{c_j}}\right) + \sum_{\substack{j=1 \\ j \notin P}}^{n} y_j \alpha_j K\left(\vec{x_i}, \vec{x_j}\right) - y_i - y_i \varphi_i \tag{IV.58}$$

for $i \in \{1, \dots, n\}$

$$\kappa = K_{e_1 e_1} + K_{e_2 e_2} - 2K_{e_1 e_2} \tag{IV.59}$$

When $y_{e_1} \neq y_{e_2}$, then

$$U = \max\left(0, \beta_{d_2} - \beta_{d_1}\right) \tag{IV.60}$$

$$V = \min\left(C_{c_2}, C_{c_1} - \beta_{c_1} + \beta_{c_2}\right) \ . \tag{IV.61}$$

When $y_{e_1} = y_{e_2}$, then

$$U = \max\left(0, \beta_{d_1} + \beta_{d_2} - C_{e_1}\right) \tag{IV.62}$$

$$V = \min\left(C_{e_2}, \beta_{d_1} + \beta_{d_2}\right) \ . \tag{IV.63}$$

So in order to solve OP 17 we use SMO method, with the difference that E_{c_i} for $i = \{1, \dots, p\}$ are used while running SMO, and updated globally after SMO solves OP 17. From the practical point of view it is enough to use existing code for SMO for solving subproblems.

IV.3.3 Experiments

We compared SMS with SMO and found that in deed SMS is generally faster than SMO. The SVM optimization with SMS algorithm was tested with the subproblem size of 5. The size was experimentally chosen as the best size among the others.

IV.4 Heuristic of Alternatives

SVM standard heuristic choose parameters in every iteration based on KKT conditions. We propose additionally taking into account objective function value growth from OP 15.

The HoA for the selected pairs of parameters compute objective function growth and choose the pair maximizing this growth. Both heuristic try to near to the solution the most in every iteration. Sometimes they choose the same parameters, sometimes not.

In HoA the strategy of generating pairs to check is to create pairs from parameters which fulfill SVM optimization conditions the best or near the best. In the set of pairs there is always a pair, that would be chosen by SVM standard heuristic. So the heuristic of alternatives has two strategies incorporated, one to check optimization conditions and the second to check objective function value growth.

The pairs that will be chosen for checking might look like this

$$\left(s_{11}, s_{21}\right), \left(s_{12}, s_{21}\right), \left(s_{11}, s_{22}\right), \left(s_{13}, s_{21}\right), \ldots \quad . \tag{IV.64}$$

The pair which has the maximal objective function value growth will be chosen. In practice, we choose among 4, 9 or 16 pairs, e.g.

$$\left(s_{11}, s_{21}\right), \left(s_{12}, s_{21}\right), \left(s_{11}, s_{22}\right), \left(s_{12}, s_{21}\right) \quad . \tag{IV.65}$$

Note that we excluded pairs with both parameters the same.

IV.4.1 Comparison of Time Complexity

In SMO standard heuristic in every iteration optimization conditions must be computed. For every parameter, we have to compute E value. The complexity of computing E value is $O\left(n\right)$. For all parameters and all iterations the complexity is $O\left(kn^2\right)$, where k is the number of iterations.

In HoA objective function value growth of OP 13 needs to be computed in every iteration for every alternative pair. From the (IV.1) we get the formula for objective function value growth

$$\begin{aligned}\Delta f_2\left(\vec{\beta}\right) = \sum_{i=1}^{2} \Delta\beta_i - \sum_{j=1}^{2} y_{c_j} \Delta\beta_j \sum_{\substack{i=1 \\ i \notin C}}^{n} y_i \alpha_i K_{c_j i} - \frac{1}{2} \sum_{i=1}^{2} \left(\beta_{i\text{new}}^2 - \beta_{i\text{old}}^2\right) K_{c_i c_i} \\ - y_{c_1 c_2} \left(\beta_{1\text{new}} \beta_{2\text{new}} - \beta_{1\text{old}} \beta_{2\text{old}}\right) K_{c_1 c_2} \quad .\end{aligned} \tag{IV.66}$$

This step needs computing solution for all alternative pairs. Computing solution for single alternative pair has constant time. The complexity of computing objective function growth for all iterations is $O\left(kmn\right)$, where m is the number of alternative pairs in every iteration. Overall complexity of heuristic of alternatives is $O\left(kn^2 + kmn\right)$. The complexity of HoA differs from SMO standard heuristic with the kmn part, which does not influence on overall time, when the number of parameters is big enough.

Both heuristics can be speed up by incorporating actualization of E values for all parameters. After this modification computing optimization conditions for single parameter becomes constant. Complexity of SMO standard heuristic falls to $O\left(kn\right)$. Computing objective function value growth also becomes constant for every parameter, so for HoA the complexity is: $O\left(kn + km\right)$. The difference is the km part which doesn't influence on overall time, when the number of parameters is big enough.

IV.4.2 Experiments

HoA will be compared with SMO standard heuristic. We can see the comparison of a number of iterations and computation time with HoA heuristic in Table IV.1. The method was tested with classification and regression problems. For regression problems, we used the ε-SVR method. We can see the improvement in the number of iterations in all tests with a linear and polynomial kernels. For RBF kernel, results depend on a value of the sigma parameter. A strong improve-

Table IV.1: The HoA performance for real world data sets. Column descriptions: *id* – an id of a test, *dn* – data set name, *ker* – a kernel with a parameter, *m1it* – the number of iterations of SVM, *m21it* – the number of iterations of SVM with HoA, *m1ctt* – cumulative training time of SVM (in *s*), *m2ctt* – cumulative training time of SVM with HoA (in *s*)

id	dn	ker	idRef	m1it	m2it	m1ctt	m2ctt
0	a1aAll	denseLinear 0.0	0	10291	8004	9.93725	9.6975
1	a1aAll	denseRBF 0.00813	1	50775	50775	13582.618	13677.475
2	breast-cancer	denseRBF 0.1	2	905	969	0.8405	1.097
3	diabetes	denseRBF 0.125	3	1045	1066	0.669	0.89
4	djia	denseRBF 0.08333	4	1736	1586	1.883	2.035
5	abalone	denseLinear 0.0	5	6033	5565	18.487	19.681
6	abalone	denseRBF 0.125	6	15382	15879	99.377	122.48
7	abalone	denseRBF 0.5	7	10399	10314	67.824	76.006
8	cadata	denseRBF 0.125	8	100126	99571	3291.658	3311.83
9	djia	denseLinear 0.0	9	1833	1251	6.202	4.506
10	djia	denseRBF 0.1	10	1154	1073	2.283	2.339
11	djia	denseRBF 0.5	11	2542	2307	4.27	4.243
12	housing	denseLinear 0.0	12	4195	2857	4.555	3.438
13	housing	densePolynomial 5.0	13	414420	92011	108.594	31.754
14	housing	denseRBF 0.077	14	318	313	0.901	0.939
15	housing	denseRBF 0.5	15	1209	1048	2.393	2.338

ment in the number of iterations leads to the improvement in training time. Tests have shown, that heuristic of alternatives can be better than SMO standard heuristic.

IV.5 Summary

In this thesis, we analyzed two implementation improvements for SVM, the first one for speed of training of SVM, the second one for simplifying implementation of SVM solver. Tests on real world data sets show, that HoA can lead to a decrease of time of training of SVM, compared to the standard heuristic. Using the SMS method, we get simpler implementation of SVM solver with similar speed performance. Both methods can be used with δ-SVR and ε-SVR for solving regression problems.

Chapter V

Applications: Order Execution Strategies

Big orders cannot be executed on exchanges at once because of the limited number of offers on the opposite side. They must be split into smaller orders and execute in a longer time period. There are various possible measures of the quality of order execution. The most popular are market VWAP, pre-trade price, and post-trade price, all values compared to VWAP for the order. In this thesis, we investigate the first one. The model of the strategy achieving market VWAP was presented recently in [1, 2]. In [1], authors found that improving quality of the volume prediction leads to better execution performance, however they had found contradicting results in [9].

The goal of the conducted work was to extend the theoretical results for the execution strategy achieving VWAP and to show on which factors the final execution error depends on. Furthermore, we wanted to implement a part of the strategy by using a general purpose machine learning method such as SVR.

In [1], authors predict a volume function by decomposing volume into two parts and using the average method and autoregressive models for prediction. In [2], authors predict a volume participation function by decomposing it to some parts and using a generalized method of moments for predicting parameters of a statistical model. We propose to use a different approach for prediction, namely, use general machine learning methods which do not assume any particular distribution and statistical properties of the model. We compared SVR with some proposed null hypotheses such as predicting volume participation while assuming constant volume profile, prediction based on average from historical data for the same time slice and prediction from the previous time slice.

The final execution performance depends not only on volume but also on stock prices during order execution. One of the way of improving the strategy is to incorporate information about prices to the model. The presented strategy splits the order to smaller chunks based on volume participation function. The possible way of incorporating information about prices to the model is to adjust volume participation function. We propose modeling the final solution by incorporating prior knowledge about prices by using margin knowledge, recently proposed for SVC [30], for δ-SVR [32] and for ε-SVR [34]. It was used for manipulating a decision curve for classification problems, and manipulating a regression function for regression ones.

A test scenario which is investigated in this article is to split execution of the order during a one exchange session. Note that the size of the order has a direct influence on the possibility of achieving VWAP. It is easier to achieve VWAP for bigger orders relative to the daily volume, because the order is also a part of the market VWAP. In the extreme situation where the order is the only one executed during the session we achieve VWAP (neglecting transaction costs of executing the order).

The outline of the thesis is as follows. In the first section, we define VWAP ratio, in the second section, we present an introduction to Volume Participation Strategy. In the third section, we present predicting volume participation, in the fourth section, we show how to incorporate prior knowledge about prices, and finally in the fifth section, we describe conducted experiments.

V.1 VWAP Ratio

In this section, we present the definition of the VWAP ratio, preceded by some definitions and propositions regarding VWAP measure which we will use later. First, we will introduce some notation: T is the time period for executing the order (e.g. one session), n is the number of trades during T, $v\left(i\right)$ is a volume of the i-th trade, v is a market volume in T, $p\left(i\right)$ is a price of the i-th trade. We have

$$v = \sum_{i=1}^{n} v\left(i\right) \ . \tag{V.1}$$

Definition V.1.1 (Market VWAP)**.** Market VWAP is

$$VWAP = \frac{\sum_{i=1}^{n} p\left(i\right) v\left(i\right)}{v} \ . \tag{V.2}$$

For a volume of the order in T, labeled v_0, we have

$$v_0 = \sum_{i=1}^{n} v_0\left(i\right) \ , \tag{V.3}$$

where $v_0\left(i\right)$ is a part of the order volume belongs to the i-th trade.

Definition V.1.2 (Order VWAP)**.** Order VWAP is

$$VWAP_0 = \frac{\sum_{i=1}^{n} p\left(i\right) v_0\left(i\right)}{v_0} \ . \tag{V.4}$$

In the presented strategy, we divide T to some time slices. Below, we list some propositions regarding time slices.

Proposition V.1.1. *Assuming that the volume is divided to two parts with known VWAP for these parts ($VWAP_1$ and $VWAP_2$) and known volumes (v_1 and v_2 respectively), overall VWAP is*

$$VWAP = \frac{VWAP_1 v_1 + VWAP_2 v_2}{v_1 + v_2} \ . \tag{V.5}$$

We can generalize this proposition to multiple parts, e.g. multiple time slices, we can divide T to m parts, aggregated volume from all trades in the i-th part is noted as $v\left(T_i\right)$, $VWAP$ for all trades in the i-th part is noted as $VWAP\left(T_i\right)$, aggregated volume of the order in the i-th part is noted as $v_0\left(T_i\right)$. Then a market volume in T is

$$v = \sum_{i=1}^{m} v\left(T_i\right) \ . \tag{V.6}$$

Market VWAP in T is

$$VWAP = \frac{\sum_{i=1}^{m} VWAP\left(T_i\right) v\left(T_i\right)}{v} \ . \tag{V.7}$$

A volume of the order in T is

$$v_0 = \sum_{i=1}^{m} v_0\left(T_i\right) \tag{V.8}$$

and order VWAP in T is

$$VWAP_0 = \frac{\sum_{i=1}^{m} VWAP_0\left(T_i\right) v_0\left(T_i\right)}{v_0} \ . \tag{V.9}$$

In this thesis, we investigate a problem of developing a strategy which optimizes the ratio of order VWAP to market VWAP for future trades.

Definition V.1.3 (VWAP ratio)**.** A VWAP ratio is defined as

$$\frac{VWAP_0}{VWAP} = \frac{\sum_{i=1}^{n} p\left(i\right) v_0\left(i\right)}{v_0} \frac{v}{\sum_{i=1}^{n} p\left(i\right) v\left(i\right)} \quad . \tag{V.10}$$

We can reformulate (V.10) by substituting

$$v_0 = V_1 v \tag{V.11}$$

and we get

$$\frac{VWAP_0}{VWAP} = \frac{\sum_{i=1}^{n} p\left(i\right) v_0\left(i\right)}{V_1 \sum_{i=1}^{n} p\left(i\right) v\left(i\right)} \quad , \tag{V.12}$$

where V_1 is a ratio of order volume to market volume

$$V_1 = \frac{v_0}{v} \quad . \tag{V.13}$$

For m time slices we get

$$\frac{VWAP_0}{VWAP} = \frac{\sum_{i=1}^{m} VWAP\left(T_i\right) v_0\left(T_i\right)}{V_1 \sum_{i=1}^{m} VWAP\left(T_i\right) v\left(T_i\right)} \quad . \tag{V.14}$$

For buy orders we would like to minimize this ratio, for sell orders maximize. Particularly, the goal is to achieve the ratio equal or less than 1 for buy orders and equal or greater than 1 for sell orders. Note that a challenge in optimizing this ratio is that future volume and/or future prices have to be predicted. First, we will present a strategy which achieves the ratio equal to 1 by predicting volume participation. Second, we will present an extension of this strategy which allows to incorporate information about prices. Such separation is desirable, because we can compute the error for prediction based on volume, and the error for price prediction.

V.2 Volume Participation Strategy

Here, we describe a model of the strategy which achieves VWAP ratio equal to 1 without assuming any price information. The strategy is to trade with a predicted volume. It means that for every time slice T_i we have

$$v_0\left(T_i\right) = V_1 v\left(T_i\right) = \frac{v_0}{v} v\left(T_i\right) \quad . \tag{V.15}$$

We can see that the strategy fulfills (V.8). We can reformulate it

$$v_0\left(T_i\right) = \frac{v\left(T_i\right)}{v} v_0 = r\left(T_i\right) v_0 \quad , \tag{V.16}$$

where

$$r\left(T_i\right) = \frac{v\left(T_i\right)}{v} \quad . \tag{V.17}$$

The r is called *volume participation*, Fig. V.1. We can easily check that for this strategy (V.11) is fulfilled. After substituting (V.15) to (V.14) we get the ratio equals to 1.

In order to use this strategy in practice we have to predict volume participation $r\left(T_i\right)$ (V.16) for every time slice and try to trade at $VWAP\left(T_i\right)$ inside every time slice. Note that it would be possible to use (V.15) instead of (V.16), but then we would need to predict volume v. Predicting separately volume v and $v\left(T_i\right)$ is more richer prediction than just only ratios $r\left(T_i\right)$. For the same ratios, we could have multiple possible values of v. In other words, when we have only ratios $r\left(T_i\right)$ we are not able to conclude about a value of v. There exist multiple different volume shapes with the same ratios $r\left(T_i\right)$.

Note that for different values of a free term a of a volume function we can get different values

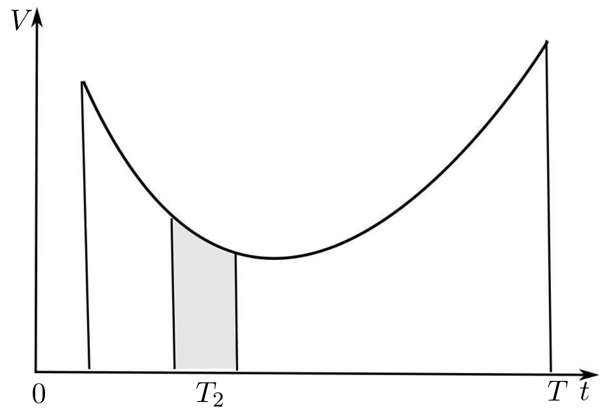

Figure V.1: Idea of volume participation. Volume participation for T_2 is interpreted as a ratio of gray area to the whole area below volume from 0 to T

of $r(T_p)$ for some p, in other words translating a volume function would change $v_0(T_p)$

$$v_0(T_p) = \frac{v_0(v(T_p)+a)}{\sum_{i=1}^m v(T_p)+a} \quad . \tag{V.18}$$

The $v_0(T_p)$ can have different values for different values of the free term a. So it is not enough to predict only volume shape (without a free term).

Let's consider an improvement to the model that our orders are taken into account in global volume. We will redefine v as a volume of other orders. Then we have

$$VWAP = \frac{\sum_{i=1}^m VWAP(T_i)(v(T_i)+v_0(T_i))}{v+v_0} \quad . \tag{V.19}$$

For m time slices the ratio is

$$\frac{VWAP_0}{VWAP} = \frac{(v+v_0)\sum_{i=1}^m VWAP(T_i)v_0(T_i)}{v_0\sum_{i=1}^m VWAP(T_i)(v(T_i)+v_0(T_i))} \quad . \tag{V.20}$$

Let's analyze the similar strategy of trading as before, that is

$$v_0(T_i) = \frac{v_0}{v}v(T_i) \quad . \tag{V.21}$$

We can see that (V.8) is fulfilled. Let's derive the ratio

$$\frac{VWAP_0}{VWAP} = \frac{(v+v_0)\frac{v_0}{v}\sum_{i=1}^m VWAP(T_i)v(T_i)}{v_0(1+\frac{v_0}{v})\sum_{i=1}^m VWAP(T_i)v(T_i)} \tag{V.22}$$

$$\frac{VWAP_0}{VWAP} = \frac{v_0+\frac{v_0^2}{v}}{v_0+\frac{v_0^2}{v}} = 1 \quad . \tag{V.23}$$

We can see that again a VWAP ratio is equal to 1.

V.2.1 Errors for Volume Participation Strategy

There are two possible sources of errors in realizing this strategy. The first error ε_1 is related to trading with $VWAP(T_i)$, the second error ε_2 is related to predicting volume participation in T_i, after substituting (V.16) to (V.9) and considering the errors

$$VWAP_0 = \sum_{i=1}^m (VWAP(T_i)+\varepsilon_1(T_i))(r(T_i)+\varepsilon_2(T_i)) \quad . \tag{V.24}$$

While comparing $VWAP$ to $VWAP_0$ we get the following error

Theorem V.2.1.

$$\varepsilon = \frac{VWAP_0}{VWAP} - 1 = \frac{\sum_{i=1}^{m} \varepsilon_1 (T_i) \, r(T_i)}{\sum_{i=1}^{m} VWAP(T_i) \, r(T_i)} + \frac{\sum_{i=1}^{m} \varepsilon_2 (T_i) \, VWAP(T_i)}{\sum_{i=1}^{m} VWAP(T_i) \, r(T_i)} \tag{V.25}$$

$$+ \frac{\sum_{i=1}^{m} \varepsilon_1 (T_i) \, \varepsilon_2 (T_i)}{\sum_{i=1}^{m} VWAP(T_i) \, r(T_i)} \ . \tag{V.26}$$

In this thesis, we are interested mainly in optimizing ε_2. So we either generate prior values of E_1 where $\varepsilon_1 (T_i) = E_1 (T_i) \, VWAP (T_i)$, or substitute $\varepsilon_1 (T_i) = 0$. Lowering ε_2 leads to a lower variance of ε.

Comparison to time-weighted average price (TWAP) strategy. The TWAP strategy trades the same quantity in every time slice. The TWAP can be interpreted as the volume participation strategy with predicted volume as a constant function. We expect worser prediction of volume participation for TWAP, therefore greater value of ε_1 compared to the VWAP strategy, so we expect greater variance of ε for TWAP method.

V.3 Predicting Volume Participation

In order to use Volume Participation Strategy we need to predict volume participation $r(T_i)$ for all time slices. In this thesis, we investigate four methods of prediction, the first one arbitrarily assumes that a volume is a constant function, so a volume participation function is also a constant one (it is used in TWAP strategy), the second one predicts $r(T_i)$ as an average value from previous days, it is kind of a local strategy. The third one predicts $r(T_i)$ as $r(T_{i-1})$ from the previous time slice and the last one predicts volume participation $r(T_i)$ from historical data by assuming that $r(\cdot)$ is a continuous function. There is only one feature that is the id of the time slice, so the feature space is a discrete one. For the last prediction, we use SVR methods. Volume participation prediction has two additional constraints that should be fulfilled:

$$\sum_{i=1}^{m} r(T_i) = 1 \ , \tag{V.27}$$

$$r(T_i) > 0 \ . \tag{V.28}$$

For the TWAP predictor, they are fulfilled out of hand. For the remaining predictors we need special consideration. For the second predictor, we propose the following procedure: we equally decrease values of all $r(T_i)$ in order to fulfill (V.27), and when some values are below zero, we adjust them to zero. We repeat these two steps until both constraints are fulfilled. For the last predictor, we propose the direct incorporation of (V.27) and (V.28) to the optimization problem.

V.4 Incorporating Prior Knowledge About Prices

Volume Participation Strategy achieves the ratio equal to 1 in the presented model. It is possible to achieve better execution performance by taking into account price prediction. The general idea of an improvement is to increase order volume when the predicted price is relatively low during the session for buy orders (relatively high for sell orders).

There are two problems concerning manipulating a participation function based on price prediction. First is in achieving enough price prediction performance for improving the error ε. Second, that increased order volume for some time slices could change noticeably the prices during the next sessions (it is called *market impact*) and additionally decrease price prediction performance.

Because price prediction is a challenging task, we propose to incorporate simple price prediction rules, such as *in the second part of the session prices will be generally higher than in the* *first one (or vice versa).* For this rule we might want to increase participation in the first half of the session, and decrease in the second one (for buy orders). The simple way of incorporating such knowledge is to increase participation by some value e.g. $p = 0.1$ for the first part of the session and decrease by the same value in the second part of the session (assuming the even number of time slices). The problem with this solution is that participation rate is not smooth in the half of the session. The second issue is that participation change by the same value in the first part and the second. We cannot improve participation changes by using price information, because we have just only simple prediction rules. So we propose to set participation changes based on volume participation prediction performance. We want to increase value and chance of changing p for time slices with worser volume participation prediction performance, and decrease value of p for the rest. For this purpose, we use SVM with margin knowledge introduced for SVC in [30, 32], for ε-SVR in [34] and for δ-SVR in [33]. The technique was used for manipulating classification boundaries [30] and regression functions [33]. It posses a desired property of adjusting the output function depending on the prediction performance.

V.4.1 Defining Knowledge About Prices

We divide the period T to 2 periods, first half of the session and the second. We propose setting a tractor parameter $\varphi_i = r$ for all training examples, where r is a configurable parameter. When we expect that prices will be higher in the second part of the session, for every tractor from the first part of the session we set -1 class, and for the second part we set 1 class (in reverse for opposite prediction).

V.5 Experiments

We divide experiments into three parts: in the first part we compare prediction performance of SVM with null hypotheses. In the second experiment, we compare ε for SVM and null hypotheses. We compare prediction performance of SVM with the following null hypotheses: prediction based on constant function, prediction based on average participation from historical data for the same time slice and prediction from the previous time slice. In the third experiment, we compare ε for δ-SVR and δ-SVR with incorporated margin knowledge.

For solving ε-SVR and SVC for particular parameters we use LibSVM [3] ported to Java. Data which are used for experiments are tick data for securities from National Association of Securities Dealers Automated Quotations (NASDAQ)-100 index for about a half year period (from 01.01.2011 to 20.05.2011) which were compressed to a desired size of time slices. Data includes trades from opening and closing crosses. For all data sets, every feature is scaled linearly to $[0, 1]$. The results are averaged for all tested instruments. For variable parameters like the C, σ for the RBF kernel, δ for δ-SVR, and ε for ε-SVR, we use a double grid search method for finding the best values. We use modified double cross validation with shifting data. Inner cross validation is used for finding the best values of the variable parameters. Instead of standard outer cross validation, we shift data. Hence, the validation set is always after the training set. We use a fixed size for the training set, that is 2 weeks, and for the validation set 1 week.

V.5.1 Prediction Performance and ε Comparison

We compare δ-SVR and ε-SVR with null hypotheses. Results are presented in Table V.1. For fair comparison purposes we choose $\varepsilon_1 = 0$. We performed tests for half hour slices.

We achieve better generalization performance for ε-SVR and δ-SVR for almost all null hypotheses with better results for δ-SVR. The ε-SVR had problems with achieving significant improvements for a linear kernel. The average null hypothesis is the most competitive comparing to SVR, we achieve slightly better generalization performance for SVR, but without significant difference based on t-test for ε-SVR, with significant difference for δ-SVR. Comparing additional measure of variance of ε we achieve slightly better results for ε-SVR than for

Table V.1: Performance of δ-SVR for order execution. Column descriptions: *id* – an id of a test, *a name* – a name of the test, δ-SVR compared with hypotheses 1 or 2 or 3, *ts* – a size of time slice (in minutes), *simT* – the number of shifts, results are averaged, *ker* – a kernel (*pol* – a polynomial kernel), *kerP* – a kernel parameter (for a polynomial kernel it is a dimension, for the RBF kernel it is σ), *trs* – a training set size for every stock, *all* – the number of all data for every stock, *dm* – a dimension of the problem, *tr12M* – a percent average difference in mean error for training data, if greater than 0 than SVM is better, *te12M* – the same as tr12M, but for testing data, *teT* – t value for the t-test for comparing testing error, *e12M* – comparison of a variance of ε. The value 'var' means that we search for the best value

id	name	ts	simT	ker	kerP	trs	all	dm	tr12M	te12M	teT	e12M
1	δ-SVRvsH1	30m	5	lin	—	130	1075	1	12.7%	11.7%	15.2	−92.6%
2	ε-SVRvsH1	30m	5	lin	—	130	1075	1	1.11%	0.7%	0.8	0.23%
5	δ-SVRvsH1	30m	5	rbf	0.1	130	1075	1	51.2%	46.9%	62.6	-72.1%
6	ε-SVRvsH1	30m	5	rbf	0.1	130	1075	1	49.5%	45.6%	59.9	0.28%
11	δ-SVRvsH2	30m	5	rbf	0.1	130	1075	1	2.75%	3.4%	3.0	−72%
12	ε-SVRvsH2	30m	5	rbf	0.1	130	1075	1	−0.5%	1.17%	1.0	−0.02%
13	δ-SVRvsH3	30m	5	lin	—	130	1075	1	10.83%	9.1%	9.58	−92.5%
14	ε-SVRvsH3	30m	5	lin	—	130	1075	1	−1.05%	−2.13%	−2.1	0.96%
17	δ-SVRvsH3	30m	5	rbf	0.1	130	1075	1	50.1%	45.3%	48.6	-71.9%
18	ε-SVRvsH3	30m	5	rbf	0.1	130	1075	1	48.4%	44.06%	46.7	1.02%

the first and the third hypotheses, and similar results to the second hypothesis. For δ-SVR we achieve much worser variance of ε then for all hypotheses.

V.5.2 Execution Performance with Knowledge About Prices

We compare ε for δ-SVR with incorporated prior knowledge about prices, and without. The scope of this thesis does not include testing the effectiveness of the price prediction. Therefore we propose the following procedure for generating prior nowledge about prices, we check in advance on historical data whether market VWAP will be higher in the first part of the session, or in the second one. According to this prediction we set tractors, r value is chosen arbitrarily to 0.5. Results are presented in Table V.2.

The results show that volume participation prediction performance could be worser after adjusting the function, but we can see significant improvement in ε for the modified solution. The δ-SVR with prior knowledge about prices achieves better execution performance than without prior knowledge.

V.6 Summary

In this thesis, we analyzed application of SVR for executing orders on stock markets. We compared ε-SVR and δ-SVR with simple predictors such as the average execution price from previous days. Tests were performed for stocks from NASDAQ-100 index. For both methods we achieved smaller variance of execution costs. Moreover, we decreased costs of order execution by using prediction of stock prices.

In future research, we plan to perform tests on a broader list of stocks and exchanges.

Table V.2: Performance of δ-SVR with prior knowledge about prices for order execution. Column descriptions: *id* – an id of a test, *ts* – a size of time slice (in hours), *simT* – the number of shifts, results are averaged, *ker* – a kernel (*pol* – a polynomial kernel), *kerP* – a kernel parameter (for a polynomial kernel it is a dimension, for the RBF kernel it is σ), *trs* – a training set size for every stock, *all* – the number of all data for every stock, *dm* – a dimension of the problem, *r* – a detractor value, *tr12M* – a percent average difference in mean error for training data, if greater than 0 than SVM is better, *te12M* – the same as tr12M, but for testing data, *teT* – t value for the t-test for comparing testing error, *e12M* – comparison of ε, *eT* – t-value for comparing ε. The value 'var' means that we search for the best value

id	ts	simT	ker	kerP	trs	all	dm	r	tr12M	te12M	teT	e12M	eT
22	30m	5	rbf	0.1	130	1075	1	1	−5%	−6%	−1.7	19%	2.4

Chapter VI

Summary

The main contributions of the author within the research presented in this thesis are

1. proposed a novel regression method, able to effectively used also for nonlinear problems, theoretical and practical analysis,
2. proposed margin knowledge per example for classification and regression, theoretical and practical analysis,
3. using margin knowledge per example for decreasing the number of support vectors,
4. application of SVM for executing orders.

In future research we plan to extend theoretical analysis of the proposed methods.

Appendix A

Introduction to Optimization Theory

An optimization problem in R^n is one where values of a given function $f : R^n \to R$ are to be maximized or minimized over a given set $D \subset R^n$. The function f is called the *objective function*, and the set D the *constraint set*. Notationally, we will represent these problems by

$$\text{maximize } f(x)$$

subject to

$$\vec{x} \in D$$

Alternatively

$$\max \{ f(\vec{x}) | x \in D \}$$

Such problems are called *maximization problems*. A *solution* to the problem $\max \{ f(\vec{x}) | \vec{x} \in D)$ is a point $\vec{x} \in D$ such as

$$f(\vec{x}) \geq f(\vec{y})$$

for all $\vec{y} \in D$. We will say in this case that f attains a maximum on D at x, and also refer to x as a *maximizer* of f on D.

We are especially interested in *constrained optimization problems*, the constraint set D has a form

$$D = U \cap \{ \vec{x} \in R^n | g(\vec{x}) = 0, h(\vec{x}) \geq 0 \}$$

where $U \subset R^n$ is open, $g : R^n \to R^k$, and $h : R^n \to R^n$. We will refer to the functions $g = (g_1, \ldots, g_k)$ as *equality constraints*, and to the functions $h = (h_1, \ldots, h_n)$ as *inequality constraints*.

First we will investigate the case where all the constraints are equality constraints, i.e. where the constraint set D can be represented as

$$D = U \cap \{ \vec{x} | g(\vec{x}) = 0 \}$$

where $U \subset R^n$ is open, $g : R^n \to R^k$. We will call this case as *equality-constrained optimization problems*. Second we will investigate the case where all the constraints are inequality constraints, i.e. where the constraint set has the form

$$D = U \cap \{ \vec{x} | h(\vec{x}) \geq 0 \}$$

where $U \subset R^n$ is open, $h : R^n \to R^n$. We label these *inequality-constrained optimization problems*. Finally, we will combine both type of constraints into a general case of *mixed constraints*.

A.1 Equality Constraints

First we provide a characterization of local optima of equality-constrained optimization problems.

Theorem A.1.1 (The Theorem of Lagrange)**.** *Let $f : R^n \to R$ and $g_i : R^n \to R^k$ be C^1 functions, $i = 1, \ldots, k$. Suppose $\vec{x}^*$ is a local maximum or minimum of f on the set*

$$D = U \cap \{\vec{x} | g_i(\vec{x}) = 0, i = 1, \ldots, k\} \ ,$$

where $U \subset R^n$ is open. Suppose also that $\rho(Dg(x^*)) = k$. Then, there exists a vector $\lambda^* = (\lambda_1^*, \ldots, \lambda_k^*) \in R^k$ such that

$$Df(x^*) + \sum_{j=1}^{k} \lambda_i^* Dg_i(x^*) = 0$$

There are also called first-order necessary conditions. The vector $\vec{\lambda}^*$ is called the vector of *Lagrangian multipliers.* A function $L : D \times R^k \to R$ is called the *Lagrangian* and is defined by:

$$L\left(\vec{x}, \vec{\lambda}\right) = f(\vec{x}) + \sum_{i=1}^{k} \lambda_i g_i(\vec{x}) \ .$$

Now we will present second-order conditions for these problems. We will assume that f and g are both C^2 functions.

Theorem A.1.2. *Suppose there exist points $\vec{x}^* \in D$ and $\lambda^* \in R^k$ such that $\rho(Dg(x^*)) = k$, and $Df(x^*) + \sum_{i=1}^{k} \lambda_i^* Dg_i(x^*) = 0$. Define*

$$Z(x^*) = \{z \in R^n | Dg(x^*) z = 0\}$$

and let D^2L^* denote the $n \times n$ matrix

$$D^2L(x^*, \lambda^*) = D^2f(x^*) + \sum_{i=1}^{k} \lambda_i^* D^2 g_i(x^*)$$

1. *If f has a local maximum on D at x^*, then $z' D^2 L^* z \leq 0$ for all $z \in Z(x^*)$*

2. *If f has a local minimum on D at x^*, then $z' D^2 L^* z \geq 0$ for all $z \in Z(x^*)$*

3. *If $z' D^2 L^* z < 0$ for all $z \in Z(x^*)$ with $z \neq 0$, then x^* is a strict local maximum of f on D*

4. *If $z' D^2 L^* z > 0$ for all $z \in Z(x^*)$ with $z \neq 0$, then x^* is a strict local minimum of f on D*

A.2 Inequality Constraints

We say that an inequality constraint $h_i(\vec{x}) \geq 0$ is *effective* at a point x^* if the constraint holds with equality at x^*, that is, we have $h_i(x^*) = 0$.

Theorem A.2.1 (Theorem of Kuhn and Tucker)**.** *Let $f : R^n \to R$ and $h_i : R^n \to R$ be C^1 functions, $i = 1, \ldots, n$. Suppose x^* is a local maximum of f on*

$$D = U \cap \{x \in R^n | h_i(x) \geq 0, i = 1, \ldots, n\} \ ,$$

where U is an open set in R^n. Let $E \subset \{1, \ldots, n\}$ denote the set of effective constraints at x^*, and let $h_E = (h_i)_{i \in E}$. Suppose $\rho(Dh_E(x^*)) = |E|$. Then, there exists a vector $\lambda^* = (\lambda_1^*, \ldots, \lambda_n^*) \in R^n$ such that the following conditions are met:

1. *$\lambda_i^* \geq 0$ and $\lambda_i^* h_i(x^*) = 0$ for $i = 1, \ldots, n$*

2. *$Df(x^*) + \sum_{i=1}^{n} \lambda_i^* Dh_i(x^*) = 0$*

The first condition is called *complementary slackness.*

A.3 Mixed Constraints

For notational ease, define

$$c_i = \begin{cases} g_i, \ if \ i \in \{1, \ldots, k\} \\ h_{i-k}, \ if \ i \in \{k+1, \ldots, k+n\} \end{cases}$$

Theorem A.3.1. *Let $f : R^n \to R$ and $c_i : R^n \to R$, $i = 1, \ldots, l + k$ be C^1 functions. Suppose x^* maximizes f on*

$$D = U \cap \{\vec{x} \in R^n | c_i(x) = 0, i = 1, \ldots, k, c_j(x) \geq 0, j = k+1, \ldots, k+n\}$$

where $U \subset R^n$ is open. Let $E \subset \{1, \ldots, k+n\}$ denote the set of effective constraints at x^*, and let $c_E = (c_i)_{i \in E}$. Suppose $\rho(Dc_E(x^*)) = |E|$. Then, there exists $\lambda \in R^{l+k}$ such that

1. *$\lambda_j \geq 0$ and $\lambda_j c_i(x^*) = 0$ for $j \in \{k+1, \ldots, k+n\}$*
2. *$Df(x^*) + \sum_{i=1}^{k+n} \lambda_i Dc_i(x^*) = 0$*

A.4 Optimization under Convexity

In convex optimization problems, all local optima must also be global optima.

Theorem A.4.1 (Theorem of Kuhn and Tucker)**.** *Let f be a concave C^1 function mapping U into R, where $U \subset R^n$ is open and convex. For $i = 1, \ldots, n$, let $h_i : U \to R$ also be concave C^1 functions. Suppose there is some $\hat{x} \in U$ such that*

$$h_i(\hat{x}) > 0$$

where $i = 1, \ldots, n$. Then x^* maximizes f over

$$D = \{x \in U | h_i(x) \geq 0, i = 1, \ldots, n\}$$

if and only if there is $\lambda^* \in R^k$ such that the Kuhn-Tucker first-order conditions hold:

1. *$Df(x^*) + \sum_{i=1}^{n} \lambda_i^* Dh_i(x^*) = 0$*
2. *$\lambda^* \geq 0$ and $\sum_{i=1}^{n} \lambda_i^* h_i(x^*) = 0$*

The condition that there exist a point $\hat{x}$ at which $h_i(\hat{x}) \geq 0$ for all i is called *Slater's condition.*

A.5 Duality

Theorem A.5.1. *If the point $\left(\vec{x}^*, \vec{\lambda}^*\right)$, with $\vec{\lambda}^* \geq 0$ is a saddle point of the Lagrangian associated with the primal problem then $\vec{x}^*$ is a solution to the primal problem.*

Define the *dual function*

$$h\left(\vec{\lambda}\right) = \min_{\vec{x}} L\left(\vec{x}, \vec{\lambda}\right)$$

Defining the set

$$D = \left\{\vec{\lambda} | h\left(\vec{\lambda}\right) \exists \ and \ \vec{\lambda} \geq 0\right\}$$

allows for the formulation of the dual problem

$$\underset{\vec{\lambda} \in D}{\text{maximize}} \, h\left(\vec{\lambda}\right)$$

which is equivalent to

$$\max_{\vec{\lambda} \in D} \left(\min_{\vec{x}} L \left(\vec{x}, \vec{\lambda} \right) \right)$$

Theorem A.5.2. *The point* $\left(\vec{x}^*, \vec{\lambda}^* \right)$, *with* $\vec{\lambda}^* \geq 0$ *is a saddle point of the Lagrangian function of the primal problem, if and only if:*

1. $\vec{x}^*$ *is a solution to the primal problem*
2. $\vec{\lambda}^*$ *is a solution to the dual problem*
3. $f\left(\vec{x}^*\right) = h\left(\vec{\lambda}^*\right)$

Appendix B

Introduction

B.1 Derivation of the Dual Form of OP 4

OP 20.

$$\max_{\vec{\alpha},\vec{r}} \quad d\left(\vec{\alpha},\vec{r}\right) \tag{B.1}$$

where

$$\begin{aligned} d\left(\vec{\alpha},\vec{r}\right) &= \min_{\vec{w}} t\left(\vec{w},\vec{\alpha},\vec{\xi},\vec{r}\right) \\ t\left(\vec{w},\vec{\alpha},\vec{\xi},\vec{r}\right) &= \frac{1}{2}\left\|\vec{w}\right\|^2 + \sum_{i=1}^{n} C_i\xi_i - \\ &\quad -\sum_{i=1}^{n}\alpha_i\left(y_i h\left(\vec{x_i}\right) - 1 + \xi_i - \varphi_i\right) - \sum_{i=1}^{n} r_i\xi_i \end{aligned}$$

subject to

$$\begin{aligned} \alpha_i &\geq 0 \\ r_i &\geq 0 \end{aligned}$$

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

A partial derivative with respect to w_i is

$$\frac{\partial h\left(\vec{w},\vec{\alpha},\vec{\xi},\vec{r}\right)}{\partial w_i} = w_i - \sum_{j=1}^{n}\alpha_j y_j x_{ji} = 0 \tag{B.2}$$

for $i \in \{1,\ldots,m\}$. A partial derivative with respect to ξ_i is

$$\frac{\partial h\left(\vec{w},\vec{\alpha},\vec{\xi},\vec{r}\right)}{\partial \xi_i} = C_i - r_i - \alpha_i = 0 \ . \tag{B.3}$$

After substitution of above equations to $d\left(\vec{\alpha},\vec{r}\right)$ we get

$$\begin{gathered} d\left(\vec{\alpha},\vec{r}\right) = \tfrac{1}{2}\sum_{i=1}^{m}\left(\sum_{j=1}^{n}\alpha_j y_j x_{ji}\right)\left(\sum_{k=1}^{n}\alpha_k y_k x_{ki}\right) \\ -\sum_{i=1}^{n}\alpha_i y_i\left(\sum_{j=1}^{m} w_j x_{ij}\right) + \sum_{i=1}^{n}\alpha_i\left(1+\varphi_i\right) + C_i\sum_{i=1}^{n}\xi_i \\ -\sum_{i=1}^{n}\alpha_i\xi_i - \sum_{i=1}^{n} r_i\xi_i \end{gathered} \tag{B.4}$$

$$\begin{aligned} d\left(\vec{\alpha},\vec{r}\right) = \tfrac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{n}\alpha_k\alpha_j y_k y_j x_{ki}x_{ji} - \sum_{i=1}^{n}\alpha_i y_i \sum_{j=1}^{m} w_j x_{ij} \\ + \sum_{i=1}^{n}\alpha_i\left(1+\varphi_i\right) \end{aligned} \tag{B.5}$$

$$\begin{aligned} d\left(\vec{\alpha},\vec{r}\right) = \tfrac{1}{2}\sum_{j=1}^{n}\sum_{k=1}^{n}\alpha_k\alpha_j y_k y_j \sum_{i=1}^{m} x_{ji}x_{ki} \\ - \sum_{i=1}^{n}\alpha_i y_i \sum_{j=1}^{m} x_{ij}\sum_{k=1}^{n}\alpha_k y_k x_{kj} + \sum_{i=1}^{n}\alpha_i\left(1+\varphi_i\right) \end{aligned} \tag{B.6}$$

$$\begin{aligned} d\left(\vec{\alpha},\vec{r}\right) = \tfrac{1}{2}\sum_{j=1}^{n}\sum_{k=1}^{n}\alpha_k\alpha_j y_k y_j \sum_{i=1}^{m} x_{ji}x_{ki} \\ - \sum_{i=1}^{n}\sum_{k=1}^{n}\alpha_k\alpha_i y_k y_i \sum_{j=1}^{m} x_{ij}x_{kj} + \sum_{i=1}^{n}\alpha_i\left(1+\varphi_i\right) \end{aligned} \tag{B.7}$$

$$d\left(\vec{\alpha},\vec{r}\right) = -\frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{n}\alpha_k\alpha_i y_k y_i \sum_{j=1}^{m} x_{ij}x_{kj} + \sum_{i=1}^{n}\alpha_i\left(1+\varphi_i\right) \ . \tag{B.8}$$

The dual form is

OP 21.

$$\max_{\vec{\alpha},\vec{r}} \quad d\left(\vec{\alpha},\vec{r}\right) = \sum_{i=1}^{n}\alpha_i\left(1+\varphi_i\right) - \frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{n}\alpha_k\alpha_i y_k y_i \sum_{j=1}^{m} x_{ij}x_{kj} \tag{B.9}$$

subject to

$$C_i = r_i + \alpha_i \tag{B.10}$$

$$\alpha_i \geq 0 \tag{B.11}$$

$$r_i \geq 0 \tag{B.12}$$

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

Appendix C

Regression Based on Binary Classification

C.1 An Idea of a Set of Indicator Functions

The classification problem could be defined in terms of minimizing the risk function [45]

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

where L is *a loss function* defined as

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

The regression problem could be defined as minimizing the risk function

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

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

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

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

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

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

where

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

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-values functions $Q(z, \alpha^*)$, where $\alpha \in A$ is defined as

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

where $\alpha \in A$ and

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

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 $\phi\left(Q\left(z,\alpha^*\right) - \beta\right)$, $a \in A$, $\beta \in B$. For example, a VC dimension of a set of linear functions is the same for classification and regression, i.e. for a set of functions

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

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

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

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. Based on this idea various methods directly replacing regression problem with a multiclass classification problem were proposed, [11, 16, 7].

C.2 A Proof of Thm. II.2.1

Proof. Original data are distributed according to the probability distribution $\mathrm{F_r}\left(\vec{x_r} \mid y_r\right)$. The expected value is equal to the mode for unimodal and symmetric distributions

$$\mathbb{E}\left[y_r \mid \vec{x_r}\right] \equiv \mathrm{M}\left(y_r \mid \vec{x_r}\right) \quad . \tag{C.12}$$

First, we create joint random variable $\left(\vec{x_r}, y_r\right)$ and we have

$$\mathrm{F_r}\left(\vec{x_r}, y_r\right) \equiv \mathrm{F_r}\left(y_r \mid \vec{x_r}\right) F\left(\vec{x_r}\right) \quad . \tag{C.13}$$

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

$$\Pr\left(1 \mid \vec{x_c}\right) \equiv \Pr\left(-1 \mid \vec{x_c}\right) \quad . \tag{C.14}$$

We can rewrite it as

$$\mathrm{F}\left(\vec{x_c} \mid 1\right) \Pr\left(1\right) = \mathrm{F}\left(\vec{x_c} \mid -1\right) \Pr\left(-1\right) \quad . \tag{C.15}$$

Creating classes by duplicating points implicates that

$$\Pr\left(1\right) = \Pr\left(-1\right) \quad , \tag{C.16}$$

so

$$\mathrm{F}\left(\vec{x_c} \mid 1\right) = \mathrm{F}\left(\vec{x_c} \mid -1\right) \quad . \tag{C.17}$$

Because both distributions are symmetrical and unimodal the above holds for

$$\frac{\mathrm{M}\left(\vec{x_c} \mid 1\right) + \mathrm{M}\left(\vec{x_c} \mid -1\right)}{2.0} \tag{C.18}$$

and because of symmetric translation we get

$$\frac{\mathrm{M}\left(\vec{x_c} \mid 1\right) + \mathrm{M}\left(\vec{x_c} \mid -1\right)}{2.0} \equiv \mathrm{M}\left(y_r \mid \vec{x_r}\right) \equiv \mathbb{E}\left[y_r \mid \vec{x_r}\right] \quad . \tag{C.19}$$

□

C.3 A Proof of Thm. II.2.2

Proof. Consider the distribution $\mathrm{F_r}\left(y_r \mid \vec{x_r}\right)$. For nonsymmetrical 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 $m \geq 0$. So we need to prove that there exists δ such that

$$f\left(x+\delta\right) - f\left(x-\delta\right) = 0 \tag{C.20}$$

where

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

for

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

So

$$f\left(m+\delta\right) - f\left(m-\delta\right) = 0 \tag{C.23}$$

where

$$0 \leq m \leq \delta \tag{C.24}$$

When $\delta = m$ then

$$f\left(2m\right) - f\left(0\right) \leq 0 \tag{C.25}$$

if for some $\delta > m$

$$f\left(m+\delta\right) - f\left(m-\delta\right) \geq 0 \tag{C.26}$$

then from intermediate value theorem there exists the δ.

□

C.4 Solution for (II.54)

The following holds

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

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

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

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

$$\frac{R+\Delta\delta}{1+w_c^{m+1}\Delta\delta} < R \tag{C.30}$$

$$w_c^{m+1} > \frac{1}{R} \quad . \tag{C.31}$$

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

$$\frac{R+\Delta\delta}{1+w_c^{m+1}\Delta\delta} > -R \tag{C.32}$$

$$R+\Delta\delta < -\left(1+w_c^{m+1}\Delta\delta\right)R \tag{C.33}$$

$$2R+\Delta\delta < -w_c^{m+1}\Delta\delta R \tag{C.34}$$

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

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

Appendix D

Margin Knowledge Per Example

D.1 Derivation of the Dual Form of OP 13

OP 22.

$$\max_{\vec{\alpha},\vec{r}} \quad d\left(\vec{\alpha},\vec{r}\right) \tag{D.1}$$

where

$$\begin{aligned} d\left(\vec{\alpha},\vec{r}\right) &= \min_{\vec{w},b} t\left(\vec{w},b,\vec{\alpha},\vec{\xi},\vec{r}\right) \\ t\left(\vec{w},b,\vec{\alpha},\vec{\xi},\vec{r}\right) &= \frac{1}{2}\left|\vec{w}\right|^2 + \sum_{i=1}^{n} C_i\xi_i - \\ &\quad -\sum_{i=1}^{n}\alpha_i\left(y_i h\left(\vec{x}_i\right) - 1 + \xi_i - \varphi_i\right) - \sum_{i=1}^{n} r_i\xi_i \end{aligned}$$

subject to

$$\begin{aligned} \alpha_i &\geq 0 \\ r_i &\geq 0 \end{aligned}$$

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

A partial derivative with respect to w_i is

$$\frac{\partial h\left(\vec{w},b,\vec{\alpha},\vec{\xi},\vec{r}\right)}{\partial w_i} = w_i - \sum_{j=1}^{n}\alpha_j y_j x_{ji} = 0 \tag{D.2}$$

for $i \in \{1, \ldots, m\}$. A partial derivative with respect to b is

$$\frac{\partial h\left(\vec{w},b,\vec{\alpha},\vec{\xi},\vec{r}\right)}{\partial b} = \sum_{i=1}^{n}\alpha_i y_i = 0 \ . \tag{D.3}$$

A partial derivative with respect to ξ_i is

$$\frac{\partial h\left(\vec{w},b,\vec{\alpha},\vec{\xi},\vec{r}\right)}{\partial \xi_i} = C_i - r_i - \alpha_i = 0 \ . \tag{D.4}$$

After substitution of above equations to $d\left(\vec{\alpha},\vec{r}\right)$ we get

$$\begin{aligned} d\left(\vec{\alpha},\vec{r}\right) = \tfrac{1}{2}\sum_{i=1}^{m}\left(\sum_{j=1}^{n}\alpha_j y_j x_{ji}\right)\left(\sum_{k=1}^{n}\alpha_k y_k x_{ki}\right) \\ -\sum_{i=1}^{n}\alpha_i y_i\left(\sum_{j=1}^{m} w_j x_{ij} + b\right) + \sum_{i=1}^{n}\alpha_i\left(1+\varphi_i\right) + C_i\sum_{i=1}^{n}\xi_i \\ -\sum_{i=1}^{n}\alpha_i\xi_i - \sum_{i=1}^{n} r_i\xi_i \end{aligned} \tag{D.5}$$

$$\begin{aligned} d\left(\vec{\alpha},\vec{r}\right) = \tfrac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{n}\alpha_k\alpha_j y_k y_j x_{ki} x_{ji} - \sum_{i=1}^{n}\alpha_i y_i\sum_{j=1}^{m} w_j x_{ij} \\ -b\sum_{i=1}^{n}\alpha_i y_i + \sum_{i=1}^{n}\alpha_i\left(1+\varphi_i\right) \end{aligned} \tag{D.6}$$

$$\begin{aligned} d\left(\vec{\alpha},\vec{r}\right) = \tfrac{1}{2}\sum_{j=1}^{n}\sum_{k=1}^{n}\alpha_k\alpha_j y_k y_j\sum_{i=1}^{m} x_{ji} x_{ki} \\ -\sum_{i=1}^{n}\alpha_i y_i\sum_{j=1}^{m} x_{ij}\sum_{k=1}^{n}\alpha_k y_k x_{kj} + \sum_{i=1}^{n}\alpha_i\left(1+\varphi_i\right) \end{aligned} \tag{D.7}$$

$$\begin{aligned} d\left(\vec{\alpha},\vec{r}\right) = \tfrac{1}{2}\sum_{j=1}^{n}\sum_{k=1}^{n}\alpha_k\alpha_j y_k y_j\sum_{i=1}^{m} x_{ji} x_{ki} \\ -\sum_{i=1}^{n}\sum_{k=1}^{n}\alpha_k\alpha_i y_k y_i\sum_{j=1}^{m} x_{ij} x_{kj} + \sum_{i=1}^{n}\alpha_i\left(1+\varphi_i\right) \end{aligned} \tag{D.8}$$

$$d\left(\vec{\alpha},\vec{r}\right) = -\frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{n}\alpha_k\alpha_i y_k y_i\sum_{j=1}^{m} x_{ij} x_{kj} + \sum_{i=1}^{n}\alpha_i\left(1+\varphi_i\right) \ . \tag{D.9}$$

The dual form is

OP 23.

$$\max_{\vec{\alpha},\vec{r}} \quad d\left(\vec{\alpha},\vec{r}\right) = \sum_{i=1}^{n}\alpha_i\left(1+\varphi_i\right) - \frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{n}\alpha_k\alpha_i y_k y_i\sum_{j=1}^{m} x_{ij} x_{kj} \tag{D.10}$$

subject to

$$\sum_{i=1}^{n}\alpha_i y_i = 0 \tag{D.11}$$

$$C_i = r_i + \alpha_i \tag{D.12}$$

$$\alpha_i \geq 0 \tag{D.13}$$

$$r_i \geq 0 \tag{D.14}$$

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

D.2 Derivation of SMO α_2 Bounds for φ-SVC

We will derive bounds for α_2, (III.22), (III.23), (III.24), (III.25):

$$U \leq \ \alpha_2 \leq V \ , \tag{D.15}$$

where for $y_1 \neq y_2$

$$U = \max\left(0,\ \alpha_2^{old} - \alpha_1^{old}\right), \tag{D.16}$$

$$V = \min\left(C_2,\ C_1 - \alpha_1^{old} + \alpha_2^{old}\right) \ , \tag{D.17}$$

for $y_1 = y_2$

$$U = \max\left(0,\ \alpha_1^{old} + \alpha_2^{old} - C_1\right), \tag{D.18}$$

$$V = \min\left(C_2,\ \alpha_1^{old} + \alpha_2^{old}\right) \ . \tag{D.19}$$

We present two derivations: geometrical and analytical one.

Geometrical Proof. The equality equation of SVM is

$$\alpha_2 = \alpha_1^{\text{old}} y_1 y_2 + \alpha_2^{\text{old}} - \alpha_1 y_1 y_2 \ . \tag{D.20}$$

The line crosses left side of the square, where $\alpha_1 = 0$. The line crosses the right side of the square, where $\alpha_1 = C_1$. When $y_1 = y_2$, then $y_1 y_2 = 1$, after substituting it to (D.20) we get

$$p : \alpha_2 = \alpha_1^{old} + \alpha_2^{old} - \alpha_1 \ . \tag{D.21}$$

The line p has a negative slope equals to -1, Fig. D.1(a).

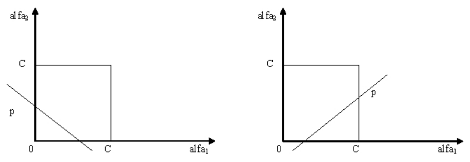

Figure D.1: Visualization of the constraints. We can see a line p with the negative slope in a) and the positive one in b)

After substituting $\alpha_1 = 0$ and $\alpha_1 = C_1$, we get values of points of crossings of p line with lines $\alpha_1 = 0$ and $\alpha_1 = C_1$:

$$\alpha_2 = \alpha_1^{old} + \alpha_2^{old} \tag{D.22}$$

and

$$\alpha_2 = \alpha_1^{old} + \alpha_2^{old} - C_1 \ . \tag{D.23}$$

Because crossing points have to lie in the square we get the following bounds for α_2

$$U = \max\left(0,\ \alpha_1^{old} + \alpha_2^{old} - C_1\right), \tag{D.24}$$

$$V = \min\left(C_2,\ \alpha_1^{old} + \alpha_2^{old}\right) \ . \tag{D.25}$$

When $y_1 \neq y_2$, then $y_1 y_2 = -1$, after substituting it to (D.20)

$$p : \alpha_2 = -\alpha_1^{old} + \alpha_2^{old} + \alpha_1 \ . \tag{D.26}$$

The line p has a positive slope equals to 1, Fig. D.1(b). After substituting $\alpha_1 = 0$ and $\alpha_1 = C_1$, we get values of points of crossings of p line with lines $\alpha_1 = 0$ and $\alpha_1 = C_1$:

$$\alpha_2 = -\alpha_1^{old} + \alpha_2^{old} \tag{D.27}$$

and

$$\alpha_2 = -\alpha_1^{old} + \alpha_2^{old} + C_1 \ . \tag{D.28}$$

Because crossing points have to lie in the square we get the following bounds for α_2

$$U = \max\left(0,\ \alpha_2^{old} - \alpha_1^{old}\right), \tag{D.29}$$

$$V = \min\left(C_2,\ C_1 - \alpha_1^{old} + \alpha_2^{old}\right) \ . \tag{D.30}$$

□

Analytical Proof. We have an inequality for α_1

$$0 \leq \alpha_1 \leq C_1 \tag{D.31}$$

and a line p

$$\alpha_1 y_1 + \alpha_2 y_2 = \alpha_1^{old} y_1 + \alpha_2^{old} y_2 \ , \tag{D.32}$$

after transformation

$$\alpha_1 = \alpha_1^{old} + y_1 y_2 \alpha_2^{old} - y_1 y_2 \alpha_2 \ , \tag{D.33}$$

after substituting it to the inequality we get

$$0 \le \alpha_1^{old} + y_1 y_2 \alpha_2^{old} - y_1 y_2 \alpha_2 \le C_1 \ . \tag{D.34}$$

When $y_1 = y_2$, then $y_1 y_2 = 1$, so

$$0 \le \alpha_1^{old} + \alpha_2^{old} - \alpha_2 \le C_1 \ . \tag{D.35}$$

Consider now the first part of the inequality,

$$\alpha_1^{old} + \alpha_2^{old} - \alpha_2 \ge 0 \tag{D.36}$$

$$\alpha_2 \le \alpha_1^{old} + \alpha_2^{old} \ . \tag{D.37}$$

Because the parameter α_2 has to fulfill the inequality $\alpha_2 \le C_2$, so the upper bound for α_2 is

$$V = \min\left(C_2,\ \alpha_1^{old} + \alpha_2^{old}\right) \ . \tag{D.38}$$

Considering the second part of the inequality,

$$\alpha_1^{old} + \alpha_2^{old} - \alpha_2 \le C_1 \tag{D.39}$$

$$\alpha_2 \ge \alpha_1^{old} + \alpha_2^{old} - C_1 \ . \tag{D.40}$$

Because the parameter α_2 has to fulfill the inequality $\alpha_2 \ge 0$, so the lower bound for α_2 is

$$U = \max\left(0,\ \alpha_1^{old} + \alpha_2^{old} - C_1\right) \ . \tag{D.41}$$

When $y_1 \ne y_2$, then $y_1 y_2 = -1$, so

$$0 \le \alpha_1^{old} - \alpha_2^{old} + \alpha_2 \le C_1 \ . \tag{D.42}$$

Consider now the first part of the inequality,

$$\alpha_1^{old} - \alpha_2^{old} + \alpha_2 \ge 0 \tag{D.43}$$

$$\alpha_2 \ge \alpha_2^{old} - \alpha_1^{old} \ . \tag{D.44}$$

Because the parameter α_2 has to fulfill the inequality $\alpha_2 \ge 0$, so the lower bound for α_2 is

$$U = \max\left(0,\ \alpha_2^{old} - \alpha_1^{old}\right) \ . \tag{D.45}$$

Considering the second part of the inequality,

$$\alpha_1^{old} - \alpha_2^{old} + \alpha_2 \le C_1 \tag{D.46}$$

$$\alpha_2 \le C_1 + \alpha_2^{old} - \alpha_1^{old} \ . \tag{D.47}$$

Because the parameter α_2 has to fulfill the inequality $\alpha_2 \le C_2$, so the upper bound for α_2 is

$$V = \min\left(C_2,\ C_1 - \alpha_1^{old} + \alpha_2^{old}\right) \ . \tag{D.48}$$

□

D.3 Derivation of the SMO Solution

We have to find new values of parameters in SMO step for SVC. First we compute α_2^{unc}

$$\alpha_2^{\text{unc}} = \alpha_2^{\text{old}} + \frac{y_2 \left(E_1 - E_2\right)}{\kappa} \tag{D.49}$$

and then

$$\alpha_2 = \begin{cases} V, \quad if\ \alpha_2^{\text{new,unc}} > V, \\ \alpha_2^{\text{new,unc}}, \quad if\ U \leq \alpha_2^{\text{new,unc}} \leq V, \\ U, \quad if\ \alpha_2^{\text{new,unc}} < U \end{cases} \tag{D.50}$$

$$\alpha_1^{\text{new}} = \alpha_1^{\text{old}} + y_1 y_2 \left(\alpha_2^{\text{old}} - \alpha_2^{\text{new}}\right) \quad . \tag{D.51}$$

For simplicity of the proof we use a notation

$$K\left(\vec{x_i}, \vec{x_j}\right) \equiv K_{ij} \quad , \tag{D.52}$$

where $i, j = 1, 2$, and we define

$$f_i = \sum_{j=1}^{n} y_j \alpha_j K_{ij} \tag{D.53}$$

$$E_i = f_i - y_i = \sum_{j=1}^{n} y_j \alpha_j K_{ij} - y_i \tag{D.54}$$

$$v_i = \sum_{j=3}^{n} y_j \alpha_j K_{ij} = f_i - \sum_{j=1}^{2} y_j \alpha_j K_{ij} \tag{D.55}$$

for $i = 1$ or $i = 2$, where n is the number of all vectors.

The objective function has a form

$$\begin{aligned} f\left(\alpha_1, \alpha_2\right) = \alpha_1 + \alpha_2 - \tfrac{1}{2} K_{11} \alpha_1^2 - \tfrac{1}{2} K_{22} \alpha_2^2 - \\ y_1 y_2 K_{12} \alpha_1 \alpha_2 - y_1 \alpha_1 v_1 - y_2 \alpha_2 v_2 + \text{const} \ . \end{aligned} \tag{D.56}$$

After substituting $s_{ij} = y_i y_j$ for $i, j = 1, 2$ for simplification of notation we get

$$\begin{aligned} f\left(\alpha_1, \alpha_2\right) = \alpha_1 + \alpha_2 - \tfrac{1}{2} K_{11} \alpha_1^2 - \tfrac{1}{2} K_{22} \alpha_2^2 - \\ s_{12} K_{12} \alpha_1 \alpha_2 - y_1 \alpha_1 v_1 - y_2 \alpha_2 v_2 + \text{const} \ . \end{aligned} \tag{D.57}$$

The linear constraint has a form: $\sum\limits_{i=1}^{n} y_i \alpha_i = 0$. It must be fulfilled with new values of α_1 and α_2

$$y_1 \alpha_1 + y_2 \alpha_2 = y_1 \alpha_1^{\text{old}} + y_2 \alpha_2^{\text{old}} = \text{const} \ . \tag{D.58}$$

Dividing above by y_1 and noticing that $y_1 y_2 = y_1 / y_2$ we get

$$\alpha_1 + y_1 y_2 \alpha_2 = \alpha_1^{\text{old}} + y_1 y_2 \alpha_2^{\text{old}} \quad , \tag{D.59}$$

after simplification

$$\alpha_1 + s_{12} \alpha_2 = \alpha_1^{\text{old}} + s_{12} \alpha_2^{\text{old}} \quad . \tag{D.60}$$

Introducing notation

$$\gamma = \alpha_1^{\text{old}} + s_{12} \alpha_2^{\text{old}} \tag{D.61}$$

we have

$$\alpha_1 + s_{12} \alpha_2 = \gamma \tag{D.62}$$

$$\alpha_1 = \gamma - s_{12} \alpha_2 \ . \tag{D.63}$$

The above equation shows how to get α_1 from α_2. After substituting above to the objective function we get

$$\begin{aligned}f\left(\alpha_1,\alpha_2\right)=\gamma-s_{12}\alpha_2+\alpha_2-\tfrac{1}{2}K_{11}\left(\gamma-s_{12}\alpha_2\right)^2-\tfrac{1}{2}K_{22}\alpha_2^2-\\ s_{12}K_{12}\left(\gamma-s_{12}\alpha_2\right)\alpha_2-y_1\left(\gamma-s_{12}\alpha_2\right)v_1-y_2\alpha_2v_2+\text{const}\ \ .\end{aligned} \tag{D.64}$$

After transformations

$$\begin{aligned}f\left(\alpha_1,\alpha_2\right)=\gamma-s_{12}\alpha_2+\alpha_2-\tfrac{1}{2}K_{11}\left(\gamma^2-2\gamma s_{12}\alpha_2+s_{12}^2\alpha_2^2\right)-\tfrac{1}{2}K_{22}\alpha_2^2-\\ s_{12}K_{12}\left(\gamma-s_{12}\alpha_2\right)\alpha_2-y_1\left(\gamma-s_{12}\alpha_2\right)v_1-y_2\alpha_2v_2+\text{const}\end{aligned} \tag{D.65}$$

$$\begin{aligned}f\left(\alpha_1,\alpha_2\right)=\gamma-s_{12}\alpha_2+\alpha_2-\tfrac{1}{2}K_{11}\gamma^2+K_{11}\gamma s_{12}\alpha_2-\tfrac{1}{2}K_{11}\alpha_2^2-\tfrac{1}{2}K_{22}\alpha_2^2\\ -s_{12}K_{12}\left(\gamma-s_{12}\alpha_2\right)\alpha_2-y_1\left(\gamma-s_{12}\alpha_2\right)v_1-y_2\alpha_2v_2+\text{const}\end{aligned} \tag{D.66}$$

$$\begin{aligned}f\left(\alpha_1,\alpha_2\right)=\gamma-s_{12}\alpha_2+\alpha_2-\tfrac{1}{2}K_{11}\gamma^2+K_{11}\gamma s_{12}\alpha_2-\tfrac{1}{2}K_{11}\alpha_2^2-\tfrac{1}{2}K_{22}\alpha_2^2\\ -s_{12}K_{12}\gamma\alpha_2+K_{12}\alpha_2^2-y_1\gamma v_1+y_2\alpha_2v_1-y_2\alpha_2v_2+\text{const}\ \ .\end{aligned} \tag{D.67}$$

Now we compute a partial derivate with respect to α_2

$$\begin{aligned}\tfrac{\partial f(\alpha_2)}{\partial\alpha_2}=1-s_{12}+K_{11}\gamma s_{12}-K_{11}\alpha_2-K_{22}\alpha_2\\ -s_{12}K_{12}\gamma+2K_{12}\alpha_2+y_2v_1-y_2v_2\ \ .\end{aligned} \tag{D.68}$$

We are looking for stationary points by equating the derivative to zero

$$\begin{aligned}\tfrac{\partial f(\alpha_2)}{\partial\alpha_2}=1-s_{12}+K_{11}\gamma s_{12}-K_{11}\alpha_2-K_{22}\alpha_2-\\ s_{12}K_{12}\gamma+2K_{12}\alpha_2+y_2v_1-y_2v_2=0\end{aligned} \tag{D.69}$$

$$\begin{aligned}1-s_{12}+K_{11}\gamma s_{12}-K_{11}\alpha_2-K_{22}\alpha_2\\ -s_{12}K_{12}\gamma+2K_{12}\alpha_2+y_2v_1-y_2v_2=0\ \ .\end{aligned} \tag{D.70}$$

Dividing both sides by y_2 we get

$$\begin{aligned}y_2-y_1+K_{11}\gamma y_1-K_{11}y_2\alpha_2-K_{22}y_2\alpha_2-\\ y_1K_{12}\gamma+2K_{12}y_2\alpha_2+v_1-v_2=0\ \ .\end{aligned} \tag{D.71}$$

Adding the new superscript for α_2 we get

$$\begin{aligned}y_2-y_1+K_{11}\gamma y_1-K_{11}y_2\alpha_2^{\text{new}}-K_{22}y_2\alpha_2^{\text{new}}-\\ y_1K_{12}\gamma+2K_{12}y_2\alpha_2^{\text{new}}+v_1-v_2=0\ \ .\end{aligned} \tag{D.72}$$

Substituting for γ, v_1 i v_2

$$\begin{aligned}y_2-y_1+K_{11}\left(\alpha_1+s_{12}\alpha_2\right)y_1-K_{11}y_2\alpha_2^{\text{new}}-K_{22}y_2\alpha_2^{\text{new}}\\ -y_1K_{12}\left(\alpha_1+s_{12}\alpha_2\right)+2K_{12}y_2\alpha_2^{\text{new}}+f_1-y_1\alpha_1K_{11}-y_2\alpha_2K_{12}\\ -f_2+y_1\alpha_1K_{12}+y_2\alpha_2K_{22}=0\end{aligned} \tag{D.73}$$

$$\begin{aligned}y_2-y_1+K_{11}\alpha_1y_1+K_{11}y_2\alpha_2-K_{11}y_2\alpha_2^{\text{new}}-K_{22}y_2\alpha_2^{\text{new}}\\ -y_1K_{12}\alpha_1-K_{12}y_2\alpha_2+2K_{12}y_2\alpha_2^{\text{new}}+f_1-y_1\alpha_1K_{11}-y_2\alpha_2K_{12}\\ -f_2+y_1\alpha_1K_{12}+y_2\alpha_2K_{22}=0\end{aligned} \tag{D.74}$$

$$\begin{aligned}y_2-y_1+K_{11}y_2\alpha_2-K_{11}y_2\alpha_2^{\text{new}}-K_{22}y_2\alpha_2^{\text{new}}\\ -K_{12}y_2\alpha_2+2K_{12}y_2\alpha_2^{\text{new}}+f_1-y_2\alpha_2K_{12}\\ -f_2+y_2\alpha_2K_{22}=0\end{aligned} \tag{D.75}$$

$$\begin{aligned}y_2-y_1-y_2\alpha_2^{\text{new}}\left(K_{11}+K_{22}-2K_{12}\right)+y_2\alpha_2(K_{11}+K_{22}-2K_{12})\\ +f_1-f_2=0\ \ .\end{aligned} \tag{D.76}$$

Introducing notation $\kappa=K_{11}+K_{22}-2K_{12}$ we get

$$y_2-y_1-y_2\alpha_2^{\text{new}}\kappa+y_2\alpha_2\kappa+f_1-f_2=0\ \ . \tag{D.77}$$

Dividing both sides by y_2 and κ

$$\alpha_2^{\text{new}} = \alpha_2 + \frac{y_2\left(E_1 - E_2\right)}{\kappa} \quad . \tag{D.78}$$

After all we have to limit α_2^{new}, so it will lie in $[U, V]$.

D.4 Derivation of the SMO solution for φ-SVC

Compare it with D.3. We have a new objective function

$$f\left(\alpha_1, \alpha_2\right) = \alpha_1\varphi_1 + \alpha_2\varphi_2 + f_{\text{smo}}\left(\alpha_1, \alpha_2\right) \quad , \tag{D.79}$$

where f_{smo} is the f function for SMO from D.3. After substituting

$$\alpha_1 = \gamma - y_1 y_2 \alpha_2 \quad , \tag{D.80}$$

where

$$\gamma = \alpha_1^{\text{old}} + y_1 y_2 \alpha_2^{\text{old}} \tag{D.81}$$

we get

$$f\left(\alpha_1, \alpha_2\right) = \varphi_1\gamma - \varphi_1 y_1 y_2 \alpha_2 + \alpha_2\varphi_2 + f_{\text{smo}}\left(\alpha_1, \alpha_2\right) \quad . \tag{D.82}$$

After differentiating we get

$$\frac{\partial f\left(\alpha_1, \alpha_2\right)}{\partial \alpha_2} = \varphi_2 - \varphi_1 y_1 y_2 + \frac{\partial f_{\text{smo}}\left(\alpha_1, \alpha_2\right)}{\partial \alpha_2} \quad . \tag{D.83}$$

And a solution is

$$\alpha_2^{\text{new}} = \alpha_2 + \frac{y_2\left(E_1 - E_2\right)}{\kappa} \quad , \tag{D.84}$$

where

$$\begin{aligned} E_i &= \sum_{j=1}^{n} y_j \alpha_j K_{ij} - y_i - y_i\varphi_i \\ \kappa &= K_{11} + K_{22} - 2K_{12} \quad . \end{aligned} \tag{D.85}$$

D.5 Derivation of ε-SVR Reformulation as φ-SVC

OP 24.

$$\min_{\vec{w_r}, b_r, \vec{\xi_r}, \vec{\xi_r^*}} \quad f\left(\vec{w_r}, b_r, \vec{\xi_r}, \vec{\xi_r^*}\right) = \left\|\vec{w_r}\right\|^2 + C_r \sum_{i=1}^{n}\left(\xi_r^i + \xi_r^{*i}\right) \tag{D.86}$$

subject to

$$y_r^i - g\left(\vec{a_i}\right) \leq \varepsilon + \xi_r^i \tag{D.87}$$

$$g\left(\vec{a_i}\right) - y_r^i \leq \varepsilon + \xi_r^{i*} \tag{D.88}$$

$$\vec{\xi_r} \geq 0 \tag{D.89}$$

$$\vec{\xi_r^*} \geq 0 \tag{D.90}$$

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

$$g\left(\vec{a_i}\right) = \vec{w_r} \cdot \vec{a_i} + b_r \quad . \tag{D.91}$$

OP 25.

$$\min_{\vec{w_r}, b_r, \vec{\xi_r}, \vec{\xi_r^*}} \quad f\left(\vec{w_r}, b_r, \vec{\xi_r}, \vec{\xi_r^*}\right) = \left\|\vec{w_r}\right\|^2 + C_r \sum_{i=1}^{2n}\left(\xi_r^i\right) \tag{D.92}$$

subject to

$$g\left(\vec{a_i}\right) \geq y_r^i - \varepsilon - \xi_r^i \tag{D.93}$$

$$g\left(\vec{a_i}\right) \leq \varepsilon + y_r^i + \xi_r^{i*} \tag{D.94}$$

$$\vec{\xi_r} \geq 0 \tag{D.95}$$

$$\vec{\xi_r^*} \geq 0 \tag{D.96}$$

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

$$g\left(\vec{\vec{a_i}}\right) = \vec{w_r} \cdot \vec{a_i} + b_r \quad . \tag{D.97}$$

OP 26.

$$\min_{\vec{w_r}, b_r, \vec{\xi_r}, \vec{\xi_r^*}} \quad f\left(\vec{w_r}, b_r, \vec{\xi_r}, \vec{\xi_r^*}\right) = \|\vec{w_r}\|^2 + C_r \sum_{i=1}^{2n} \left(\xi_r^i\right) \tag{D.98}$$

subject to

$$y_i g\left(\vec{a_i}\right) \geq y_r^i - \varepsilon - \xi_r^i \tag{D.99}$$

$$y_i g\left(\vec{a_i}\right) \geq -\varepsilon - y_r^i - \xi_r^{i*} \tag{D.100}$$

$$\vec{\xi_r} \geq 0 \tag{D.101}$$

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

$$g\left(\vec{\vec{a_i}}\right) = \vec{w_r} \cdot \vec{a_i} + b_r \quad . \tag{D.102}$$

OP 27.

$$\min_{\vec{w_r}, b_r, \vec{\xi_r}, \vec{\xi_r^*}} \quad f\left(\vec{w_r}, b_r, \vec{\xi_r}, \vec{\xi_r^*}\right) = \|\vec{w_r}\|^2 + C_r \sum_{i=1}^{2n} \left(\xi_r^i\right) \tag{D.103}$$

subject to

$$y_i g\left(\vec{a_i}\right) \geq y_c^i y_r^i - \varepsilon - \xi_r^i \tag{D.104}$$

$$\vec{\xi_r} \geq 0 \tag{D.105}$$

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

$$g\left(\vec{\vec{a_i}}\right) = \vec{w_r} \cdot \vec{a_i} + b_r \quad . \tag{D.106}$$

OP 28.

$$\min_{\vec{w_r}, b_r, \vec{\xi_r}, \vec{\xi_r^*}} \quad f\left(\vec{w_r}, b_r, \vec{\xi_r}, \vec{\xi_r^*}\right) = \|\vec{w_r}\|^2 + C_r \sum_{i=1}^{2n} \left(\xi_r^i\right) \tag{D.107}$$

subject to

$$y_i g\left(\vec{a_i}\right) \geq 1 + y_c^i y_r^i - \varepsilon - \xi_r^i - 1 \tag{D.108}$$

$$\vec{\xi_r} \geq 0 \tag{D.109}$$

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

$$g\left(\vec{\vec{a_i}}\right) = \vec{w_r} \cdot \vec{a_i} + b_r \quad . \tag{D.110}$$

And we have

$$w_{c1i} = \sum_{j=1}^{2n} y_j \alpha_j a_{ij} = \sum_{j=1}^{n} \alpha_j a_{ij} - \sum_{j=n}^{2n} \alpha_j^* a_{ij} \tag{D.111}$$

$$w_{c1i} = \sum_{j=1}^{2n} y_j \alpha_j a_{ij} = \sum_{j=1}^{n} \alpha_j a_{ij} - \sum_{j=n}^{2n} \alpha_j^* a_{ij} = \sum_{j=1}^{n} \left(\alpha_j - \alpha_j^*\right) a_{ij} \quad . \tag{D.112}$$

Appendix E

Implementation Techniques Based on KKT Conditions

E.1 The Proof of OP 19

Proof. Maximization of

$$f\left(\vec{w}, b, \vec{\xi}\right) = \|\vec{w}\|^2 + C\sum_{i=1}^{n}\xi_i - \sum_{\substack{i=1\\ i\notin P}}^{n}\alpha_i\left(y_i h\left(\vec{x_i}\right) - 1 + \xi_i\right) - \sum_{\substack{i=1\\ i\notin P}}^{n}\left(C - \alpha_i\right)\xi_i \tag{E.1}$$

with constraints $y_{c_i} h\left(\vec{x_{c_i}}\right) \geq 1 - \xi_{c_i} + \varphi_i, \quad \xi_{c_i} \geq 0$
for $i \in \{1, \ldots, p\}$, where $h\left(\vec{x_i}\right) = \vec{w} \cdot \vec{x_i} + b$ for $i \in \{1, \ldots, n\}$.

$$L = \|\vec{w}\|^2 + C\sum_{i=1}^{n}\xi_i - \sum_{\substack{i=1\\ i\notin P}}^{n}\alpha_i\left(y_i h\left(\vec{x_i}\right) - 1 + \xi_i\right) - \sum_{\substack{i=1\\ i\notin P}}^{n}\left(C - \alpha_i\right)\xi^i \tag{E.2}$$

$$-\sum_{i=1}^{p}\alpha_i\left(y_i h\left(\vec{x_i}\right) - 1 + \xi_i\right) - \sum_{i=1}^{p} r_i\xi_i \tag{E.3}$$

with constraints

$$\alpha_i \geq 0 \tag{E.4}$$

$$r_i \geq 0 \tag{E.5}$$

for $i \in \{1, \ldots, p\}$. Partial derivatives are

$$\frac{\partial L}{\partial b} = \sum_{i=1}^{p} y_i\alpha_i + \sum_{\substack{i=1\\ i\notin C}}^{n} y_i\alpha_i = 0 \tag{E.6}$$

$$\frac{\partial L}{\partial w_i} = w_i - \sum_{j=1}^{n}\alpha_j y_j x_{ij} = 0 \tag{E.7}$$

$$w_i = \sum_{j=1}^{n}\alpha_j y_j x_{ij} \quad . \tag{E.8}$$

□

E.2 Derivation of SMO Without Offset

We have to find new values of parameters in SMO step for SVC. We will optimize one parameter per step. For simplicity of the proof we use a notation

$$K\left(\vec{x_i}, \vec{x_j}\right) \equiv K_{ij} \quad , \tag{E.9}$$

where $i, j = 1, 2$, and we define

$$E_i = \sum_{j=1}^{n} y_j \alpha_j K_{ij} - y_i \tag{E.10}$$

$$v_1 = \sum_{j=2}^{n} y_j \alpha_j K_{ij} = \sum_{j=1}^{n} y_j \alpha_j K_{1j} - y_1 \alpha_1 K_{11} \quad . \tag{E.11}$$

The objective function has a form

$$W\left(\alpha_1\right) = \alpha_1 - \tfrac{1}{2} K_{11} \alpha_1^2 - y_1 \alpha_1 v_1 + \text{const} \quad . \quad . \tag{E.12}$$

Now we compute a partial derivate with respect to α_1

$$\tfrac{\partial W(\alpha_1)}{\partial \alpha_1} = 1 - K_{11} \alpha_1 - y_1 v_1 \quad . \tag{E.13}$$

We are looking for stationary points by equating the derivative to zero

$$\frac{\partial W\left(\alpha_1\right)}{\partial \alpha_1} = 1 - K_{11} \alpha_1^{\text{new}} - y_1 v_1 = 0 \tag{E.14}$$

$$1 - K_{11} \alpha_1^{\text{new}} - y_1 \sum_{j=1}^{n} y_j \alpha_j K_{1j} + \alpha_1 K_{11} = 0 \tag{E.15}$$

$$\alpha_1^{\text{new}} = \alpha_1 - \frac{y_1 E_1}{K_{11}} \quad . \tag{E.16}$$

Then we have to bound α_1^{new}:

$$0 \le \alpha_1^{\text{new}} \le C_1 \quad . \tag{E.17}$$

E.3 Derivation of SMO Without Offset for φ-SVC

Compare it with E.2. We have a new objective function

$$f\left(\alpha_1\right) = \alpha_1 \varphi_1 + f_{\text{smo}}\left(\alpha_1, \alpha_2\right) \quad , \tag{E.18}$$

where f_{smo} is the f function for SMO from E.2. After differentiating we get

$$\frac{\partial f\left(\alpha_1\right)}{\partial \alpha_1} = \varphi_1 + \frac{\partial f_{\text{smo}}\left(\alpha_1\right)}{\partial \alpha_1} \quad . \tag{E.19}$$

And a solution is

$$\alpha_1^{\text{new}} = \alpha_1 - \frac{y_1 E_1}{K_{11}} \quad , \tag{E.20}$$

where

$$E_i = \sum_{j=1}^{n} y_j \alpha_j K_{ij} - y_i - y_i \varphi_i \quad . \tag{E.21}$$

Appendix F

Applications: Order Execution Strategies

F.1 Proof of Thm. V.2.1

Proof. The proof is

$$VWAP_0 = \frac{\sum_{i=1}^{m} \left(VWAP\left(\Delta t_i\right) + \varepsilon_1\left(\Delta t_i\right)\right)\left(v_0\left(r\left(\Delta t_i\right) + \varepsilon_2\left(\Delta t_i\right)\right)\right)}{v_0} \tag{F.1}$$

$$VWAP_0 = \sum_{i=1}^{m} \left(VWAP\left(\Delta t_i\right) + \varepsilon_1\left(\Delta t_i\right)\right)\left(r\left(\Delta t_i\right) + \varepsilon_2\left(\Delta t_i\right)\right) \tag{F.2}$$

$$\frac{VWAP_0}{VWAP} - 1 = \frac{v\sum_{i=1}^{m} \left(VWAP\left(\Delta t_i\right) + \varepsilon_1\left(\Delta t_i\right)\right)\left(r\left(\Delta t_i\right) + \varepsilon_2\left(\Delta t_i\right)\right)}{\sum_{i=1}^{m} VWAP\left(\Delta t_i\right) v\left(\Delta t_i\right)} - 1 = \tag{F.3}$$

$$= \frac{\sum_{i=1}^{m} \left(VWAP\left(\Delta t_i\right) + \varepsilon_1\left(\Delta t_i\right)\right)\left(r\left(\Delta t_i\right) + \varepsilon_2\left(\Delta t_i\right)\right)}{\sum_{i=1}^{m} VWAP\left(\Delta t_i\right) r\left(\Delta t_i\right)} - 1 = \tag{F.4}$$

$$= \frac{\sum_{i=1}^{m} \varepsilon_1\left(\Delta t_i\right) r\left(\Delta t_i\right)}{\sum_{i=1}^{m} VWAP\left(\Delta t_i\right) r\left(\Delta t_i\right)} + \frac{\sum_{i=1}^{m} \varepsilon_2\left(\Delta t_i\right) VWAP\left(\Delta t_i\right)}{\sum_{i=1}^{m} VWAP\left(\Delta t_i\right) r\left(\Delta t_i\right)} \tag{F.5}$$

$$+ \frac{\sum_{i=1}^{m} \varepsilon_1\left(\Delta t_i\right) \varepsilon_2\left(\Delta t_i\right)}{\sum_{i=1}^{m} VWAP\left(\Delta t_i\right) r\left(\Delta t_i\right)} \quad . \tag{F.6}$$

□

References

[1] Jedrzej Bialkowski, Serge Darolles, and Gaelle Le Fol. Improving vwap strategies: A dynamic volume approach. *Journal of Banking & Finance*, 32(9):1709–1722, September 2008. 61

[2] Christian T. Brownlees, Fabrizio Cipollini, and Giampiero M. Gallo. Intra-daily volume modeling and prediction for algorithmic trading. Econometrics working papers archive, Universita' degli Studi di Firenze, Dipartimento di Statistica "G. Parenti", February 2009. 5, 61

[3] 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`. 24, 35, 45, 46, 66

[4] Nianyi Chen, Wencong Lu, Jie Yang, and Guozheng Li. *Support Vector Machine in Chemistry*. World Scientific Publishing Co., 2004. 1

[5] Glenn M. Fung, Olvi L. Mangasarian, and Jude W. Shavlik. Knowledge-based nonlinear kernel classifiers. In *Learning Theory and Kernel Machines*, Lecture Notes in Computer Science, pages 102–113. Springer Berlin / Heidelberg, 2003. 4

[6] Glenn M. Fung, Olvi L. Mangasarian, and Jude W. Shavlik. Knowledge-based support vector machine classifiers. In S. Thrun S. Becker and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 521–528. MIT Press, Cambridge, MA, 2003. 1, 4

[7] 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. 13, 78

[8] L. Hamel. *Knowledge discovery with support vector machines*. Wiley series on methods and applications in data mining. John Wiley & Sons, 2009. 51

[9] D. Hobson. Vwap and volume profiles. *Journal of Trading*, 1(2):38–42, 2006. 61

[10] Te-Ming Huang, Vojislav Kecman, and Ivica Kopriva. *Kernel Based Algorithms for Mining Huge Data Sets*. Springer, 2006. 1

[11] 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. 13, 78

[12] T. Joachims. Making large-scale support vector machine learning practical, 1998. 52

[13] Thorsten Joachims. *Learning to Classify Text Using Support Vector Machines*. Kluwer Academic Publishers, 2002. 1, 9

[14] Yuh jye Lee and Olvi L. Mangasarian. Rsvm: Reduced support vector machines. In *Data Mining Institute, Computer Sciences Department, University of Wisconsin*, pages 00–07, 2001. 43

[15] Masayuki Karasuyama, Ichiro Takeuchi, and Ryohei Nakano. Reducing svr support vectors by using backward deletion. In *KES '08: Proceedings of the 12th international conference on Knowledge-Based Intelligent Information and Engineering Systems, Part III*, pages 76–83, Berlin, Heidelberg, 2008. Springer-Verlag. 42, 43

[16] 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. 1, 13, 78

[17] S. Sathiya Keerthi, Olivier Chapelle, and Dennis DeCoste. Building support vector machines with reduced classifier complexity. *J. Mach. Learn. Res.*, 7:1493–1515, December 2006. 42, 43

[18] Gautam Kunapuli, Kristin P. Bennett, Amina Shabbeer, Richard Maclin, and Jude W. Shavlik. Online knowledge-based support vector machines. In *ECML/PKDD (2)*, pages 145–161, 2010. 4

[19] Fabien Lauer and Gérard Bloch. Incorporating prior knowledge in support vector machines for classification: A review. *Neurocomputing*, 71(7-9):1578–1594, 2008. 1, 9

[20] Fabien Lauer and Gérard Bloch. Incorporating prior knowledge in support vector regression. *Mach. Learn.*, 70:89–118, January 2008. 1, 9

[21] Yizeng Liang, Qing-Song Xu, Hong-Dong Li, and Dong-Sheng Cao. *Support Vector Machines and Their Application in Chemistry and Biotechnology.* Taylor and Francis Group, LLC, 2011. 1

[22] Libsvm data sets. http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/, 06 2011. 24, 45, 46

[23] Chun-Fu Lin and Sheng-De Wang. Fuzzy support vector machines. *IEEE Transaction on Neural Networks*, 13(2):464–471, 2002. 9

[24] 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. 1, 13

[25] Olvi L. Mangasarian, Jude W. Shavlik, and Edward W. Wild. Knowledge-based kernel approximation. *Journal of Machine Learning Research*, 5:1127–1141, 2004. 43

[26] Olvi L. Mangasarian and Edward W. Wild. Nonlinear knowledge-based classification. *IEEE Transactions on Neural Networks*, 19(10):1826–1832, 2008. 4

[27] Kuan ming Lin and Chih jen Lin. A study on reduced support vector machines. *IEEE TRANSACTIONS ON NEURAL NETWORKS*, 14:1449–1459, 2003. 43

[28] Tom M. Mitchell. *Machine Learning.* McGraw-Hill, New York, 1997. 4

[29] Marcin Orchel. Support vector machines: Sequential multidimensional subsolver (sms). In Adam Dabrowski, editor, *Signal Processing: Algorithms, Architectures, Arrangements, and Applications, 2007*, pages 135–140. IEEE - The Institute of Electrical and Electronics Engineers Inc. Region 8 - Europe, Middle East and Africa. Chapter Circuits and Systems. Poland Section. Poznan University of Technology. Faculty of Computing Science and Management. Division of Signal Processing and Electronic Systems., September 2007. 57

[30] 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. 4, 13, 31, 35, 61, 66

[31] Marcin Orchel. Paper id 55. In *Submitted to European Conference of Machine Learning, ECML 2010*, April 2010. 1, 13

[32] Marcin Orchel. Incorporating a priori knowledge from detractor points into 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 332–341. Springer Berlin / Heidelberg, 2011. 4, 31, 35, 43, 61, 66

[33] 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. 1, 13, 14, 15, 20, 21, 35, 66

[34] 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. 4, 13, 31, 35, 43, 61, 66

[35] Marcin Orchel. Support vector regression with a priori knowledge used in order execution strategies based on vwap. In Jie Tang, Irwin King, Ling Chen, and Jianyong Wang, editors, *Advanced Data Mining and Applications*, volume 7121 of *Lecture Notes in Computer Science*, pages 318–331. Springer Berlin / Heidelberg, 2011. 5

[36] John C. Platt. *Fast training of support vector machines using sequential minimal optimization*, pages 185–208. MIT Press, Cambridge, MA, USA, 1999. 34, 35, 51

[37] Jean-Baptiste Pothin and Cedric Richard. Incorporating prior information into support vector machines in the form of ellipsoidal knowledge sets, 2006. 4

[38] Manel Martinez Ramon and Christos Christodoulou. *Support Vector Machines for Antenna Array Processing and Electromagnetics*. Margan & Claypool Publishers, 2006. 1

[39] Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001. 23, 51

[40] Catarina Silva and Bernardete Ribeiro. *Inductive Inference for Large Scale Text Classification*. Springer, 2010. 1

[41] Ingo Steinwart, Don R. Hush, and Clint Scovel. Training svms without offset. *Journal of Machine Learning Research*, 12:141–202, 2011. 7

[42] Francis Eng Hock Tay and Lijuan Cao. Modified support vector machines in financial time series forecasting. *Neurocomputing*, 48(1-4):847–861, 2002. 10

[43] R. J. Vanderbei. LOQO: An interior point code for quadratic programming. *Optimization Methods and Software*, 11:451–484, 1999. 57

[44] Vladimir N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995. 1, 13, 19

[45] Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, September 1998. 1, 10, 13, 19, 22, 77

[46] Lei Wang, Ping Xue, and Kap Luk Chan. Incorporating prior knowledge into svm for image retrieval. In *ICPR '04: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 2*, pages 981–984, Washington, DC, USA, 2004. IEEE Computer Society. 1, 9

[47] Meng Wang, Jie Yang, Guo-Ping Liu, Zhi-Jie Xu, and Kuo-Chen Chou. Weighted-support vector machines for predicting membrane protein types based on pseudo amino acid composition. *Protein engineering, design & selection*, 17(6):509–516, 2004. 9

[48] Edward W. Wild. *Optimization-based Machine Learning and Data Mining*. PhD thesis, University of Wisconsin-Madison, 2008. 4

[49] 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. 13

[50] Mingrui Wu, Bernhard Schölkopf, and Gökhan Bakir. A direct method for building sparse kernel learning algorithms. *JOURNAL OF MACHINE LEARNING RESEARCH*, 7:603–624, 2006. 43

[51] Xiaoyun Wu and Rohini Srihari. Incorporating prior knowledge with weighted margin support vector machines. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 326–333, New York, NY, USA, 2004. ACM. 9, 31

[52] 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. 24

[53] L. Yu, S. Wang, K.K. Lai, and L. Zhou. *Bio-inspired credit risk analysis: computational intelligence with support vector machines*. Springer Verlag, 2008. 1

List of Figures

List of Tables

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,

$\vec{w}_{\mathbf{r}}^{i}$ 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,

Order Execution Strategies

$VWAP$ it is a symbol for a volume weighted average price,

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,

ν-SVC ν support vector classification,

ν-SVM ν support vector machines,

ε-SVR ε-insensitive support vector regression,

φ-SVC φ support vector classification,

C-SVC C support vector classification,

C-SVM C support vector machines,

CM capacity minimization,

DJIA Dow Jones Industrial Average,

ERM empirical risk minimization,

HoA heuristic of alternatives,

KKT Karush-Kuhn-Tucker,

LS least squares,

MSE mean squared error,

NASDAQ National Association of Securities Dealers Automated Quotations,

OMS order management system,

RBF radial basis function,

RMSE root mean squared error,

RSVM reduced support vector machines,

SMO sequential minimal optimization,

SMS Sequential Multidimensional Subsolver,

SRM structural risk minimization,

SVC support vector classification,

SVM support vector machines,

SVR	support vector regression,
TLS	total least squares,
TWAP	time-weighted average price,
VC	Vapnik-Chervonenkis,
VWAP	volume-weighted average price,

Glossary

Machine Learning

a binary classification problem A learning problem with binary outputs,

A set of hypotheses or a hypothesis space, a set or class of candidate functions,

batch learning All the data are given to the learner at the start of learning,

decision function A solution for the classification problem,

generalization The ability of a hypothesis to correctly classify data not in the training set,

learning algorithm The algorithm which takes the training data as input and selects a hypothesis from the hypothesis space,

multi-class classification A learning problem with a finite number of categories,

overfit Hypotheses that become too complex in order to become consistent are said to overfit,

regression A learning problem with real-valued outputs,

solution of the learning problem The estimate of the target function which is learned or output by the learning algorithm,

supervised learning Learning when examples are input/output pairs,

target function When a function from inputs to outputs exists it is referred to as the target function,

training data A set of examples of input/output functionality,

Order execution strategies

broker An individual or firm that charges a fee or commission for executing buy and sell orders submitted by an investor,

DJIA DJIA, known as the "Dow". One of the main US share indices which monitors the movement of 30 blue chip companies traded on the New York Stock Exchange. The index is a simple average of the share prices, not allowing for market capitalization,

index In the stock market, an index is a device that measures changes in the prices of a basket of shares, and represents the changes using a single figure. The purpose is to give investors an easy way to see the general direction of the market or shares in the index,

NASDAQ National Association of Securities Dealers' Automated Quotations System. The first electronic stock market, which uses computers and telecommunications to trade shares rather than a traditional trading floor,

NASDAQ 100 index An index composed of the 100 largest, most actively traded U.S. companies listed on the Nasdaq stock exchange. This index includes companies from a broad range of industries with the exception of those that operate in the financial industry, such as banks and investment companies,

order The instruction, by a customer to a brokerage, for the purchase or sale of a security with specific conditions,

Order Management System (OMS) An electronic system developed to execute securities orders in an efficient and cost-effective manner. Brokers and dealers use OMSs when filling orders for various types of securities and are able to track the progress of each order throughout the system,

security A financial asset such as a share or bond of a company, government body or other organization,

stock exchange An electronic screen-based or trading floor-based market where securities are bought and sold,

symbol An identity code, or ticker, allocated to a company by the exchange on which its stock is traded. Usually the code is an abbreviation of the company's name,

tick The minimum upward or downward movement in the price of a security or a futures or options contract,

trade A transaction involving buying or selling a security or commodity,

trading session The period from when a market or exchange opens until it closes,

trading volume The total number of securities or contracts traded in a given period,

Volume Weighted Average Price A measure of the price at which most of trading took place during some period. It is calculated as the value of trades divided by the volume over a given period,