AGH University of Science and Technology

Faculty of Computer Science, Electronics and Telecommunications

Department of Computer Science

Doctoral Thesis

mgr inż. Marcin Orchel

Incorporating Prior Knowledge into SVM Algorithms in Analysis of Multidimensional Data

Supervisor:
prof. dr hab. inż. Witold Dzwinel

Kraków 2012

Abstract

In this thesis, we present results for research conducted by us regarding the regression method, called δ support vector regression (δ-SVR), the method of incorporating knowledge about margin of an 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, that 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, fewer support vectors, and smaller generalization error over different values of ε and δ. The δ-SVR method is faster for linear kernels while using sequential minimal optimization (SMO) solver, for nonlinear kernels speed results depend on the data set. In this thesis, we propose a method called φ-SVC for incorporating knowledge about margin of an example for classification and regression problems. We propose two applications for φ-SVC: decreasing the generalization error of reduced models while preserving the similar number of support vectors, and incorporating the nonlinear constraint of a special type to the problem. The method was tested for SVM classifier and ε-SVR. Experiments on real world data sets show decreased generalization error of reduced models for linear and polynomial kernels. 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 the working set. It checks not only satisfaction 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 method of solving subproblems with more than two parameters, instead of using complicated quadratic programming solvers, we use SMO method. 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 the improvement of predicting participation function by using SVM with incorporated additional nonlinear constraint to the problem. We show that quality of the prediction influences execution costs. Moreover, we show how we can incorporate knowledge about stock prices. 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, 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 syntetycznych i rzeczywistych zbiorach danych. Rezultaty wskazują, że δ-SVR osiąga porównywalny błąd generalizacji, mniej 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 proponujemy metodę zwaną φ-SVC do włączania wiedzy marginesowej per przykład do problemów klasyfikacji i regresji. Proponujemy dwie aplikacje dla φ-SVC: zmniejszenie błędu generalizacji modeli zredukowanych przy zachowaniu podobnej liczby wektorów wspierających, oraz włączenie nieliniowego warunku specjalnego typu do problemu. Metoda była przetesowana dla klasyfikatora SVM, oraz dla metody regresji ε-SVR. Eksperymenty na danych rzeczywistych pokazują zmniejszony błąd generalizacji modeli zredukowanych dla liniowych i wielomianowych jąder. W tej pracy 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 roboczego. Bierze ona pod uwagę nie tylko spełnienie warunków KKT, ale również zmianę wartości funkcji celu. Testy na rzeczywistych zbiorach danych pokazują, ze HoA prowadzi do zmniejszenia czasu trenowania SVM, porównując do heurystyki standardowej. 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 metodę SMO. Otrzymujemy prostszą implementację, z podobnymi wynikami szybkościowymi. W tej pracy proponujemy zastosowanie SVR do wykonywania zleceń na rynkach akcyjnych. Używamy SVR do predykcji funkcji partycypacji w wolumenie. Proponujemy ulepszenie przewidywania funkcji partycypacji używając SVM z włączonym do problemu dodatkowym warunkiem nieliniowym. Pokazujemy, że jakość przewidywania wpływa na koszty egzekucji. Ponadto pokazujemy jak możemy włączyć wiedzę o cenach akcji. Porównaliśmy ε-SVR i δ-SVR z prostymi predyktorami takimi jak średnia cena wykonania z poprzednich dni. Testy zostały 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 finds dependencies in data. In recent times, SVM have become the new discipline that covers not only machine learning theory, but also optimization theory, multi-dimensional geometry and the theory of linear functions. The basic problems of machine learning are classification and regression. For both problems 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 that solves regression problems by binary classification. The method and the concept standing behind it shed some light on the relation between the two most popular problems in machine learning, classification and regression. The second part of the research was the analysis of knowledge about margin of an example. It is a natural extension of SVM, because 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 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 are due 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 from the following projects

- 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, [49, 50]. They are successfully used for solving classification, regression and other machine learning problems. The SVM were widely used in various domains, such as: text classification, [14, 44], biotechnology, [11, 23], economy [58], chemistry, [4, 23], physics, [42] 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 take advantage 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 [21, 22]. Prior knowledge can lead to decrease of generalization error. It was incorporated to SVM for many real world problems, such as image retrieval, [51], DNA promoter recognition, [5], breast cancer prognosis, [5].

I.1 Overview

The SVM are machine learning methods used mainly for solving classification and regression problems. They were developed by Vapnik [49, 50] 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.

A concept of replacing regression problems with classification was investigated in [49]. Vapnik derived generalization bounds for regression problems by replacing a regression function with a set of indicator functions. Based on this idea, the methods that solve regression problems as multi-class classification were developed, e.g. in [17], Fig. I.1, Fig. I.2.

Another concept of transforming regression problems into classification were proposed in [26] and independently by me in [34, 36]. The idea is to duplicate examples and move the original examples up, and the duplicated down, Fig. I.3b, Fig. I.4b. Based on this idea, we developed a novel regression method, called δ-SVR that has all advantages of SVM, and for some aspects it is better. Additionally, one of practical advantages of δ-SVR is that it can be used with any classifier based on kernel functions. Thus, any modifications and improvements of classification methods can be directly applied to regression problems.

One of 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 [21], the authors distinguish two types of prior knowledge: knowledge about class invariance, and knowledge about the data. The first type includes e.g. knowledge about classification in regions of the input space, [5, 6, 27], knowledge about

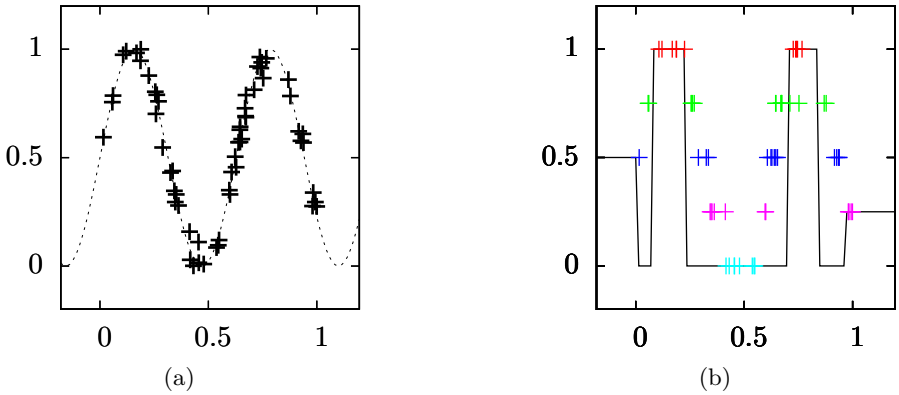
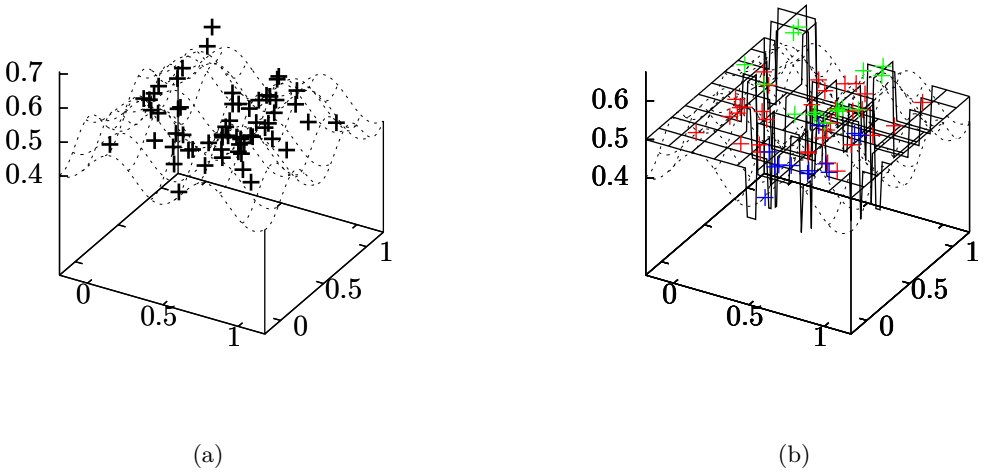


Figure I.1: Transformation from regression into multiclass classification, 2d. A dotted line - an original function from which the points were generated. (a) Points - regression examples before transformation. (b) Points - classification examples after tranformation, a solid line - a solution of multiclass classifier



(a) (b)

Figure I.2: Transformation from regression into multiclass classification, 3d. A dotted line - an original function from which the points were generated. (a) Points - regression examples before transformation. (b) Points - classification examples after tranformation, a solid line - a solution of multiclass classifier

class invariance during transformation of the input. The second type includes e.g. knowledge about unlabeled examples, imbalance of classes, quality of the data. There exist 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 proposed recently by us knowledge about margin of an example and the incorporation method, called φ-SVC, [33, 35, 37]. It is closely related to formulation of SVM optimization problem. Simplifying, knowledge about margin of an example can be interpreted as a region in the form of a hypersphere around the example, mapped to some class, Fig. I.5, Fig. I.6. Regions that have been already incorporated to SVM are polyhedral regions,

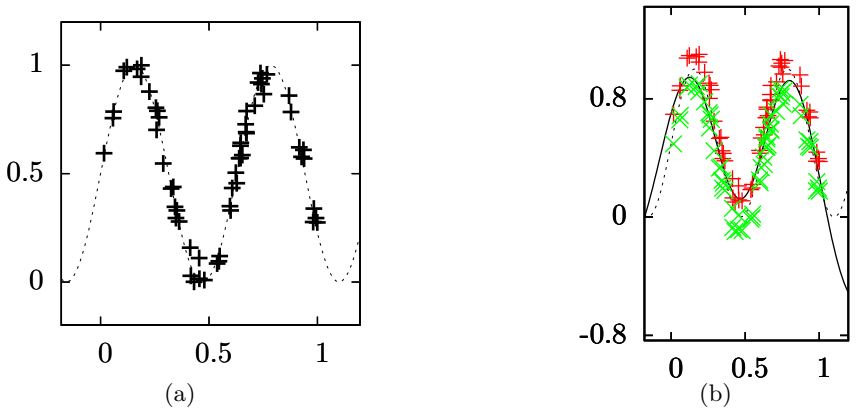


Figure I.3: Transformation from regression into binary classification, 2d. A dotted line - an original function from which the points were generated. (a) Points - regression examples before transformation. (b) Points - classification examples after tranformation, a solid line - a solution of δ-SVR

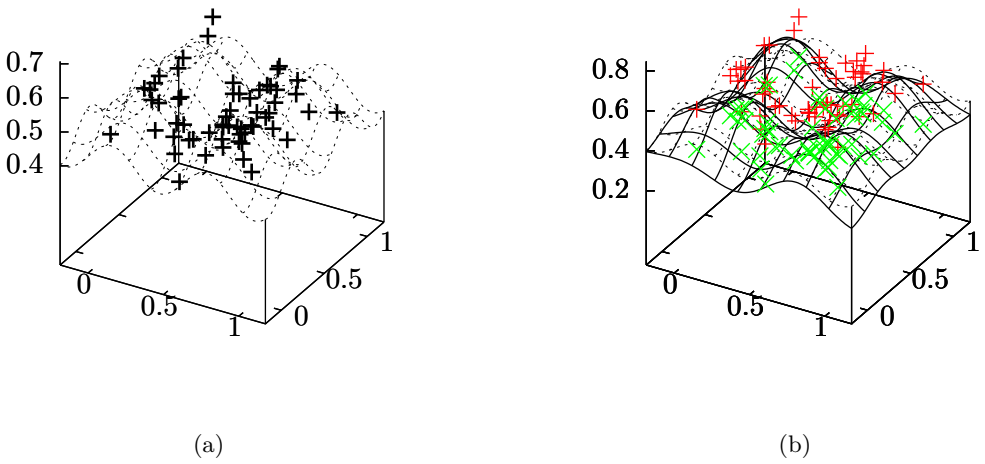
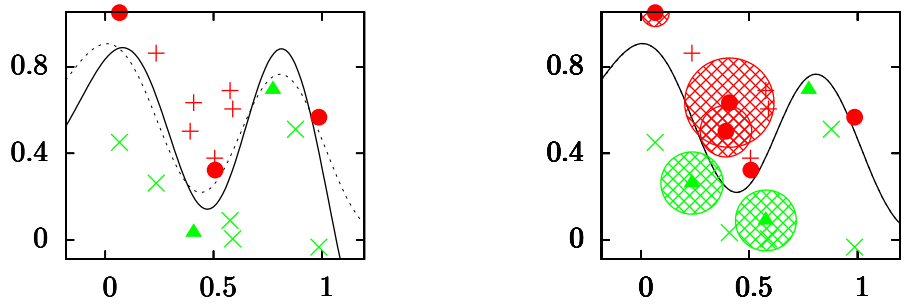


Figure I.4: Transformation from regression into binary classification, 3d. A dotted line - an original function from which the points were generated. (a) Points - regression examples before transformation. (b) Points - classification examples after tranformation, a solid line - a solution of δ-SVR

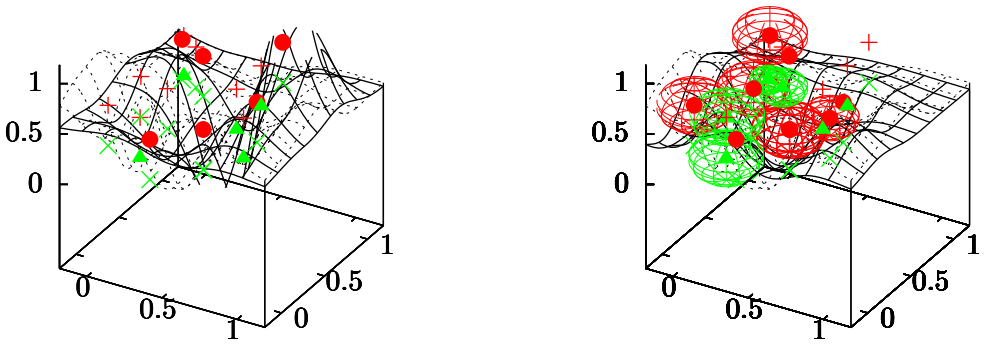
[5, 6, 20, 53], ellipsoidal regions including spheroidal regions, [41] and nonlinear regions, [27]. Knowledge about margin of an example is incorporated to SVM by using generalization of a standard SVM optimization problem. The similar idea was used for defining γ-shattering in statistical learning theory, [9]. We showed that ε-SVR can be treated as a classification problem with knowledge about margin of an example, [37]. The main application of knowledge about margin of an example presented in this thesis is decrease in complexity of solutions (decrease in the number of support vectors). A simpler solution means simpler interpretation of the problem, and decreased time of testing new examples. The second application of knowledge about margin of an example presented in this thesis is incorporation of the other types of prior knowledge. We will show that the nonlinear equality constraint in the form of a sum of function values for some data can be incorporated to the SVM optimization problem by using knowledge about margin of an example.

One of the types of prior knowledge is *imperfect prior knowledge*, [30]. Knowledge about margin of an example could be imperfect. The method of incorporation allows partial satisfaction of the knowledge conditions. There is a trade-off between standard SVM classification and the knowledge.



(a) (b)

Figure I.5: Comparison of solutions: without and with knowledge about margin of an example, 2d. Points - examples, triangles and circles - support vectors, solid lines - solutions, dotted lines - original functions from which the points were generated. (b) Circles filled with grid pattern - knowledge about margin of an example



(a) (b)

Figure I.6: Comparison of solutions: without and with knowledge about margin of an example, 3d. Points - examples, triangles and circles - support vectors, solid lines - solutions, dotted lines - original functions from which the points were generated. (b) Circles filled with grid pattern - knowledge about margin of an example

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 execution management system (EMS), which sends the orders to exchanges. One of tasks of EMS is to efficiently divide the order into smaller parts and sent them during some time period. One of methods 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 volume-weighted average price (VWAP). Recently, a simple theoretical model that 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 that can achieve even better ratio, prediction of prices is required, Fig. I.7. Recently, we proposed using SVM with knowledge about margin of an example for combining prediction of volume participation and prices, [38].

The hypotheses of the thesis are

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

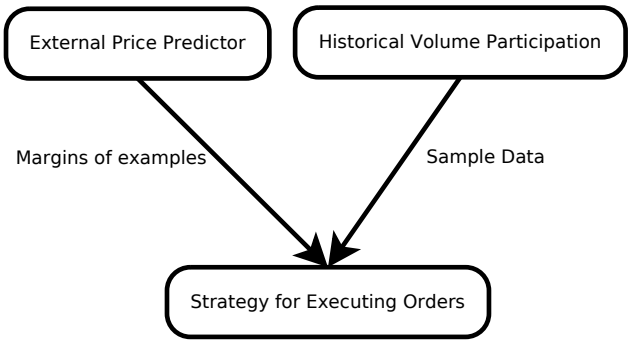


Figure I.7: A data model for the strategy of executing orders on exchanges. The strategy uses predicted prices in the form of knowledge about margin of an example and historical volume participation in the form of examples

2. Knowledge about margin of an example leads to decreased number of support vectors for reduced models for classification and regression problems.

3. Knowledge about margin of an example can be used for incorporating the nonlinear equality constraint to the solution.

4. Proposed HoA leads to increased speed of heuristic part of the SMO method.

5. Proposed SMS is a replacement for general libraries for quadratic programming for solving SVM with working set methods.

6. Knowledge about margin of an example can be used in finance for predicting a volume participation function.

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 $y_c^i \in \{-1, 1\}$. The m is a dimension of the problem. The SVC optimization problem for hard margin case with $\|\cdot\|_1$ norm is

OP 1.

$$\min_{\vec{w_c}, b_c} \quad f(\vec{w_c}, b_c) = \|\vec{w_c}\|^2 \tag{I.1}$$

subject to

$$y_c^i h(\vec{x_i}) \geq 1 \tag{I.2}$$

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

$$h(\vec{x_i}) = \vec{w_c} \cdot \vec{x_i} + b_c \ . \tag{I.3}$$

All points must be correctly classified Fig. I.8a. The SVC soft margin case optimization problem with $\|\cdot\|_1$ norm is

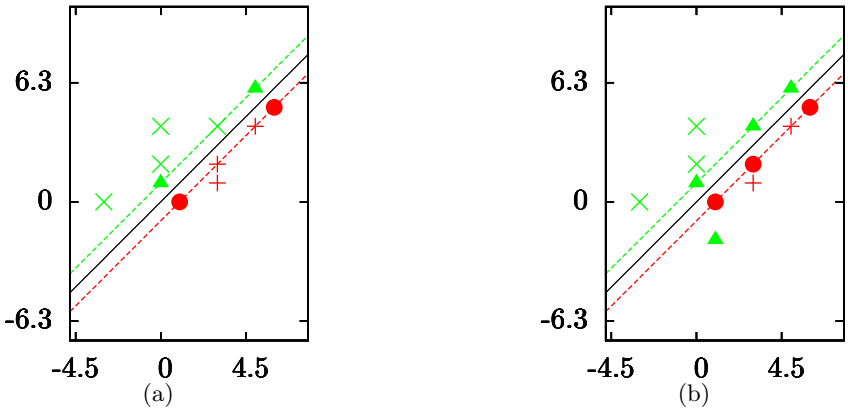


Figure I.8: Two types of margin classifiers: hard and soft. Example points, support vectors (triangles and circles), solutions (solid lines), margin lines (dashed lines). (a) Hard. (b) Soft. A misclassified point is in (1, -2)

OP 2.

$$\min_{\vec{w}_c, b_c, \vec{\xi}_c} \quad f\left(\vec{w}_c, b_c, \vec{\xi}_c\right) = \frac{1}{2}\left\|\vec{w}_c\right\|^2 + C_c \sum_{i=1}^{n} \xi_c^i \tag{I.4}$$

subject to

$$y_c^i h\left(\vec{x_i}\right) \geq 1 - \xi_c^i \tag{I.5}$$

$$\vec{\xi_c} \geq 0 \tag{I.6}$$

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

$$h\left(\vec{x_i}\right) = \vec{w_c} \cdot \vec{x_i} + b_c \ , \tag{I.7}$$

$$C_c > 0 \ . \tag{I.8}$$

The $h^*\left(\vec{x}\right) = \vec{w_c^*} \cdot \vec{x} + b_c^* = 0$ is a decision curve of the classification problem. Some of training points can be incorrectly classified Fig. I.8b.

I.2.1 SVC Dual Optimization Problem

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

OP 3.

$$\max_{\vec{\alpha}} \quad f\left(\vec{\alpha}\right) = 1 \cdot \vec{\alpha} - \frac{1}{2}\vec{\alpha^T}\mathbf{Q}\vec{\alpha} \tag{I.9}$$

subject to

$$\vec{\alpha} \cdot \vec{y} = 0 \tag{I.10}$$

$$0 \leq \alpha_i \leq C_c \tag{I.11}$$

where

$$Q_{ij} = y_i y_j K\left(\vec{x_i}, \vec{x_j}\right) \tag{I.12}$$

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

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

$$\vec{w_c^*} \cdot \vec{x} = \sum_{i=1}^{n} y_c^i \alpha_i^* K\left(\vec{x_i}, \vec{x}\right) \ . \tag{I.13}$$

Therefore, the decision curve is

$$h^*\left(\vec{x}\right) = \sum_{i=1}^{n} y_c^i \alpha_i^* K\left(\vec{x_i}, \vec{x}\right) + b_c^* = 0 \ , \tag{I.14}$$

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 that is a dot product of its arguments we call *a simple linear kernel*. *Margin boundaries* are defined as the two hyperplanes $h(\vec{x}) = -1$ and $h(\vec{x}) = 1$. *Optimal margin boundaries* are defined as the two hyperplanes $h^*(\vec{x}) = -1$ and $h^*(\vec{x}) = 1$. *Geometric margin of the hyperplane* h is defined as $1/\|\vec{w_c}\|$. The i-th training example is *a support vector*, when $\alpha_i^* \neq 0$. A set of support vectors contains all training examples lying below optimal margin boundaries ($y_c^i h^*(\vec{x_i}) < 1$), and part of the examples lying exactly on the optimal margin boundaries ($y_c^i h^*(\vec{x_i}) = 1$), Fig. I.8b.

The KKT complementary condition for OP 2 is

$$\alpha_i \left(y_c^i h(\vec{x_i}) - 1 + \xi_c^i \right) = 0 \ , \tag{I.15}$$

$$(C_c - \alpha_i)\, \xi_c^i = 0 \ . \tag{I.16}$$

The conclusions from (I.15) and (I.16) are: when $\alpha_i = 0$, then $\xi_c^i = 0$, when $0 < \alpha_i < C_c$, then $\xi_c^i = 0$ and $y_c^i h(\vec{x_i}) = 1$, when $\alpha_i = C_c$, then $y_c^i h(\vec{x_i}) = 1 - \xi_c^i$. Moreover, when $\xi_c^i > 0$, then $\alpha_i = C_c$ and $y_c^i h(\vec{x_i}) = 1 - \xi_c^i$, when $y_c^i h(\vec{x_i}) > 1 - \xi_c^i$, then $\alpha_i = 0$ and $\xi_c^i = 0$ and $y_c^i h(\vec{x_i}) > 1$. We can find values of ξ_i parameters from the solution of the dual form as following. When

$$y_c^i h^*(\vec{x_i}) \geq 1 \ , \tag{I.17}$$

then $\xi_i = 0$, else

$$\xi_i = 1 - y_c^i h(\vec{x_i}) \ . \tag{I.18}$$

I.2.2 SVC Without the Offset

Another variant of SVC is the SVC without the offset b_c, analyzed recently in [45]. The optimization problem is the same except missing b_c term, for the soft case it is

OP 4.

$$\min_{\vec{w_c}, \vec{\xi_c}} \quad f\left(\vec{w_c}, \vec{\xi_c}\right) = \frac{1}{2}\|\vec{w_c}\|^2 + C_c \cdot \vec{\xi_c} \tag{I.19}$$

subject to

$$y_c^i h(\vec{x_i}) \geq 1 - \xi_c^i \tag{I.20}$$

$$\vec{\xi} \geq 0 \tag{I.21}$$

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

$$h(\vec{x_i}) = \vec{w_c} \cdot \vec{x_i} \ , \tag{I.22}$$

$$C_c > 0 \ . \tag{I.23}$$

The dual problem is

OP 5.

$$\max_{\vec{\alpha}} \quad d(\vec{\alpha}) = \vec{\alpha} - \frac{1}{2}\vec{\alpha^T} Q \vec{\alpha} \tag{I.24}$$

subject to

$$0 \leq \vec{\alpha} \leq C_c \tag{I.25}$$

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

We can notice missing linear constraint. The decision curve is

$$h^*(\vec{x}) = \sum_{i=1}^{n} y_c^i \alpha_i^* K(\vec{x_i}, \vec{x}) = 0 \ . \tag{I.26}$$

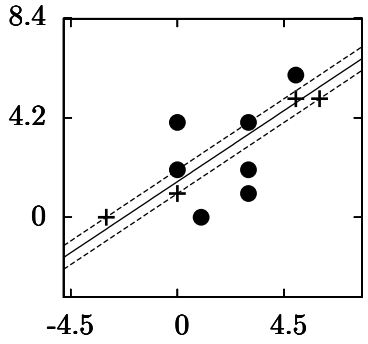


Figure I.9: The idea of ε-SVR. Points - examples, circles - support vectors, a solid line - a solution, dashed lines - ε boundaries

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 $\vec{x_i} = (x_i^1, \ldots, x_i^m)$. 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 6.

$$\min_{\vec{w}_{\mathrm{r}}, b_{\mathrm{r}}, \vec{\xi}_{\mathrm{r}}, \vec{\xi}_{\mathrm{r}}^*} \quad f\left(\vec{w}_{\mathrm{r}}, b_{\mathrm{r}}, \vec{\xi}_{\mathrm{r}}, \vec{\xi}_{\mathrm{r}}^*\right) = \frac{1}{2}\left\|\vec{w}_{\mathrm{r}}\right\|^2 + C_{\mathrm{r}} \sum_{i=1}^{n} \left(\xi_{\mathrm{r}}^i + \xi_{\mathrm{r}}^{*i}\right) \tag{I.27}$$

subject to

$$y_{\mathrm{r}}^i - g\left(\vec{x_i}\right) \leq \varepsilon + \xi_{\mathrm{r}}^i \tag{I.28}$$

$$g\left(\vec{x_i}\right) - y_{\mathrm{r}}^i \leq \varepsilon + \xi_{\mathrm{r}}^{i*} \tag{I.29}$$

$$\vec{\xi}_{\mathrm{r}} \geq 0 \tag{I.30}$$

$$\vec{\xi}_{\mathrm{r}}^* \geq 0 \tag{I.31}$$

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

$$g\left(\vec{x_i}\right) = \vec{w_{\mathrm{r}}} \cdot \vec{x_i} + b_{\mathrm{r}} \ , \tag{I.32}$$

$$\varepsilon \in \mathbb{R} \ . \tag{I.33}$$

The $g^*\left(\vec{x}\right) = \vec{w}_{\mathrm{r}}^* \cdot \vec{x} + b_{\mathrm{r}}^*$ is a regression function. Optimization problem 6 is transformed into an equivalent dual problem. The regression function becomes

$$g^*\left(\vec{x}\right) = \sum_{i=1}^{n} \left(\alpha_i^* - \beta_i^*\right) K\left(\vec{x_i}, \vec{x}\right) + b_{\mathrm{r}}^* \ , \tag{I.34}$$

where α_i, β_i are Lagrange multipliers, $K\left(\cdot, \cdot\right)$ is a kernel function. The ε boundaries are defined as $g\left(\vec{x}\right) - \varepsilon$ and $g\left(\vec{x}\right) + \varepsilon$. The i-th training example is *a support vector*, when $\alpha_i^* - \beta_i^* \neq 0$. For $\varepsilon \geq 0$, a set of support vectors contains all training examples lying outside ε boundaries, and part of the examples lying exactly on ε boundaries, Fig. I.9. 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 incorporation schemes require many changes of an optimization problem, new variables or new parameters. A survey on incorporating prior knowledge to SVM is in [21, 22]. In this thesis, we are concentrated on *knowledge per example*, that is knowledge associated with particular examples. We analyze mainly knowledge per example in the form of additional parameters to an optimization problem per each example.

One of types 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), [14], for ν support vector machines (ν-SVM), [52]. The C_i weights were also used for fuzzy support vector machines, [25], for denoting confidence of pseudo labels, [51] and for denoting confidence depended on quality of the data, [56]. A 1-norm soft margin SVC optimization problem for training examples $\vec{x_i}$ with weights C_i is

OP 7.

$$\min_{\vec{w},b,\vec{\xi}} \quad f\left(\vec{w},b,\vec{\xi}\right) = \frac{1}{2}\left\|\vec{w}\right\|^2 + \vec{C_c}\cdot\vec{\xi} \tag{I.35}$$

subject to

$$y_i h\left(\vec{x_i}\right) \geq 1 - \xi_i \tag{I.36}$$

$$\vec{\xi} \geq 0 \tag{I.37}$$

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

$$\vec{C_c} \gg 0 \tag{I.38}$$

$$h\left(\vec{x_i}\right) = \vec{w}\cdot\vec{x_i} + b \ . \tag{I.39}$$

The C_i weights were also used with ε-SVR for predicting time series data, [47]. The other types of weights that were used with ε-SVR are ε_i weights per example replacing the parameter ε. They were used for density estimation, [50]. Sometimes we use different ε weights for inequalities (I.28), (I.29), ε_u and ε_d respectively. And finally we can combine above changes and use ε_u^i and ε_d^i weights

OP 8.

$$\min_{\vec{w_\mathrm{r}},b_\mathrm{r},\vec{\xi_\mathrm{r}},\vec{\xi_\mathrm{r}^*}} \quad f\left(\vec{w_\mathrm{r}},b_\mathrm{r},\vec{\xi_\mathrm{r}},\vec{\xi_\mathrm{r}^*}\right) = \frac{1}{2}\left\|\vec{w_\mathrm{r}}\right\|^2 + \vec{C_\mathrm{r}}\sum_{i=1}^{n}\left(\xi_\mathrm{r}^i + \xi_\mathrm{r}^{*i}\right) \tag{I.40}$$

subject to

$$y_\mathrm{r}^i - g\left(\vec{x_i}\right) \leq \varepsilon_u^i + \xi_\mathrm{r}^i \tag{I.41}$$

$$g\left(\vec{x_i}\right) - y_\mathrm{r}^i \leq \varepsilon_d^i + \xi_\mathrm{r}^{i*} \tag{I.42}$$

$$\vec{\xi_\mathrm{r}} \geq 0 \tag{I.43}$$

$$\vec{\xi_\mathrm{r}^*} \geq 0 \tag{I.44}$$

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

$$g\left(\vec{x_i}\right) = \vec{w_\mathrm{r}}\cdot\vec{x_i} + b_\mathrm{r} \ . \tag{I.45}$$

We can notice that changing y_r^i value by Δy_r^i is equivalent to changing ε_u^i by $-\Delta y_r^i$ and changing ε_d^i by Δy_r^i.

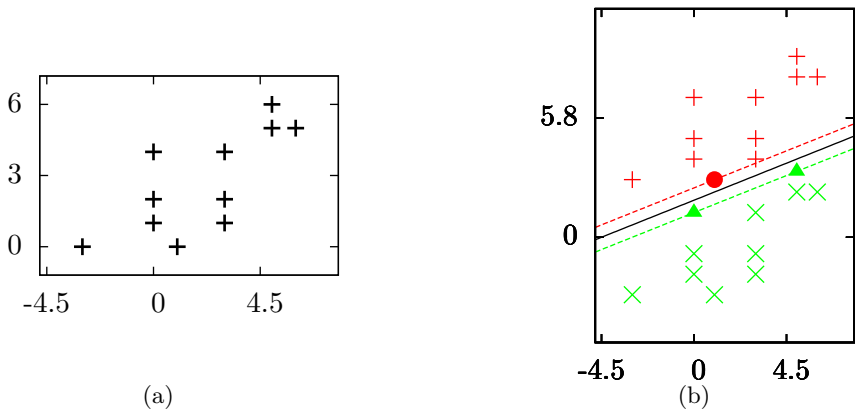
Chapter II

Regression Based on Binary Classification

Recently, an alternative regression method was proposed by me in [36, 39], which is called δ-SVR. The idea of the new method is to duplicate and shift data in order to use SVC to solve regression problems. The δ-SVR has the same advantages as ε-SVR: one of the steps is to solve a convex optimization problem, it generates sparse solutions, kernel functions can be used for generating nonlinear solutions. The δ-SVR achieves similar or better for some settings generalization performance compared with ε-SVR and improves the number of support vectors, [36, 39]. Moreover, some types of prior knowledge already incorporated to SVC can be directly used for regression problems, [36]. The δ-SVR has a potential to use a much broader type of modifications and improvements of SVC directly for regression problems without the need of reformulating them for specific regression methods. In [39], we analyzed a general problem of transforming regression into classification and also generalization performance, structural risk minimization (SRM) and sparsity for δ-SVR.

Vapnik [49, 50] proposed generalization of capacity concepts introduced for classification to regression problems by describing regression functions as a complete set of indicators, see Appendix B.1. Based on this idea a method for solving regression problems as multiclass classification problems was proposed in [12, 17, 7]. The method uses discretization process to generate multiclass labels. Some attempts also were made to combine SVR with SVC, [54]. In δ-SVR, we increase input dimension by 1 and create binary labels for duplicated and shifted points up and down, so we solve only one binary classification problem. The concept of duplicating and shifting data was first published in [26], it was investigated independently by me and submitted to [34] and published in [36]. The main problem of the realization of the concept in [26] is that an additional optimization problem must be solved every time a new example is tested in order to find a solution of the implicit equation; the authors used a golden section method. Moreover, two problems arise with this method. The solution might not exist or there could be more than one solution. In [36], we proposed a special type of kernels for which a unique solution is guaranteed and it is easily achievable by an explicit formula without the need of solving an additional optimization problem. Furthermore, in [36], we proposed using the method to incorporate knowledge about the margin of an example, which was previously incorporated to SVC for classification problems in [33, 37], directly for regression problems. We noticed that the method has a potential to use a much broader type of extensions of SVC directly for regression problems, without the need of incorporating them additionally for specific regression methods, which is a practice nowadays. Lin and Guo [26] proposed an improvement to the method, for further increasing a sparseness of the solution by decreasing a value of a shifting parameter for examples with low and high values of the output, although it requires tuning an additional parameter during a training phase.

The goals of the research presented by me in [39] were to analyze a general concept of representing regression problems as classification ones by duplicating and shifting data, to analyze potential generalization improvements of δ-SVR over SVM and to extend experiments conducted



(a) (b)

Figure II.1: The idea of the transformation of the problem in δ-SVR for 2d. (a) Points - regression examples. (b) Points - classification examples after transformation, triangles and circles - support vectors, a solid line - a solution, dashed lines - optimal margin boundaries

in [36]. The outline is as follows. In the first section, we give an introduction to δ-SVR. In the second section, we give a theoretical analysis of the transformation. In the third section, we analyze generalization ability of δ-SVR. In the fourth section, we present experiments on synthetic and real world data sets.

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 finds a regression function by the following procedure.

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

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

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

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

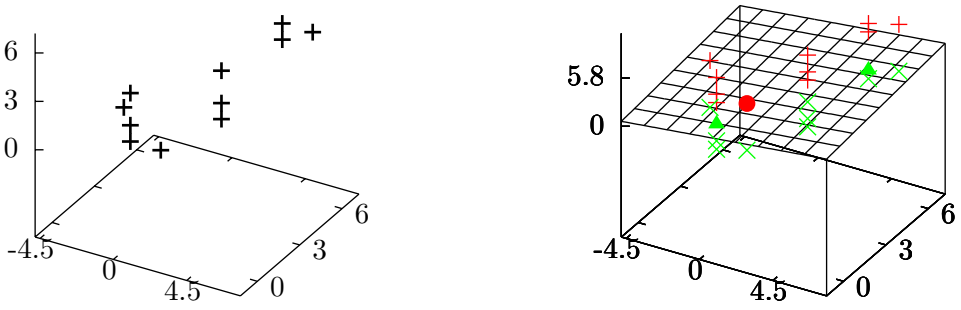
The idea of the transformation is depicted in Fig. II.1, Fig. II.2. The result of the first step is a set of training mappings for $i \in \{1, \ldots, 2n\}$

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

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

$$\vec{c_i} = \left(x_i^1, \ldots, x_i^m, y_{\mathrm{r}}^i + y_c^i \delta\right) \to y_c^i \tag{II.2}$$

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



(a) (b)

Figure II.2: The idea of the transformation of the problem in δ-SVR for 3d. (a) Points - regression examples. (b) Points - classification examples after transformation, triangles and circles - support vectors, a plane - a solution

OP 9.

$$\min_{\vec{w}_\mathrm{c},b_\mathrm{c},\vec{\xi}_\mathrm{c}} \quad f\left(\vec{w}_\mathrm{c},b_\mathrm{c},\vec{\xi}_\mathrm{c}\right)=\frac{1}{2}\left\|\vec{w}_\mathrm{c}\right\|^2+C_\mathrm{c}\sum_{i=1}^{2n}\xi_\mathrm{c}^i \tag{II.3}$$

subject to

$$y_\mathrm{c}^i\left(\vec{w_{\mathrm{c,red}}}\cdot\vec{x_i}+w_c^{m+1}\left(y_r^i+y_c^i\delta\right)+b_c\right)\geq 1-\xi_\mathrm{c}^i \tag{II.4}$$

$$\vec{\xi}_\mathrm{c}\geq 0 \tag{II.5}$$

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

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

$$\vec{w}_\mathrm{c}=\sum_{i=1}^{2n}y_\mathrm{c}^i\alpha_i\vec{c_i}\ . \tag{II.6}$$

For a simple linear kernel the explicit form of (I.14) is

$$x_{m+1}=\frac{-\sum_{j=1}^{m}w_\mathrm{c}^j x_j-b_\mathrm{c}}{w_\mathrm{c}^{m+1}}\ . \tag{II.7}$$

The regression solution is $g^*\left(\vec{x}\right)=\vec{w}_\mathrm{r}\cdot\vec{x}+b_\mathrm{r}$, 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), 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 proposed to incorporate prior knowledge to the classification problem, that the solution will be always in the form of the

function in the chosen direction. Thus, we proposed in [36] a new kernel type in which the last coordinate is placed only inside a linear term. The new kernel is constructed from an original kernel by removing the last coordinate, and adding the linear term with the last coordinate

$$K\left(\vec{x},\vec{y}\right)=K_o\left(\vec{x_{\text{red}}},\vec{y_{\text{red}}}\right)+x_{m+1}y_{m+1}\ , \tag{II.8}$$

where $\vec{x}$ and $\vec{y}$ are $m+1$ dimensional vectors, $\vec{x_{\text{red}}}=(x_1,\ldots,x_m)$, $\vec{y_{\text{red}}}=(y_1,\ldots,y_m)$, $K_{\text{o}}\left(\cdot,\cdot\right)$ is the original kernel from which the new one was constructed. For the most popular kernels polynomial, RBF and sigmoid, the conversions are respectively

$$\left(\vec{x}\cdot\vec{y}\right)^d \ \rightarrow\ \left(\sum_{i=1}^{m}x_iy_i\right)^d+x_{m+1}y_{m+1}\ , \tag{II.9}$$

$$\exp-\frac{\left\|\vec{x}-\vec{y}\right\|^2}{2\sigma^2}\ \rightarrow\ \exp-\frac{\sum_{i=1}^{m}\left(x_i-y_i\right)^2}{2\sigma^2}+x_{m+1}y_{m+1}\ , \tag{II.10}$$

$$\tanh\vec{x}\vec{y}\ \rightarrow\ \tanh\sum_{i=1}^{m}x_iy_i+x_{m+1}y_{m+1}\ , \tag{II.11}$$

where $\vec{x}$ and $\vec{y}$ are $m+1$ dimensional vectors. The proposed method of constructing new kernels always generates a function satisfying the Mercer's condition, because it generates a function which is a sum of two kernels. For the new kernel type, the explicit form of (I.14) for δ-SVR is

$$x_{m+1}=\frac{-\sum_{i=1}^{2n}y_{\text{c}}^i\alpha_iK_{\text{o}}\left(\vec{x_i},\vec{x_{\text{red}}}\right)-b_{\text{c}}}{\sum_{i=1}^{2n}y_{\text{c}}^i\alpha_ic_i^{m+1}}\ . \tag{II.12}$$

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.12) as

$$x_{m+1}=\frac{-\sum_{i=1}^{n}\left(\alpha_i-\alpha_{n+i}\right)K_{\text{o}}\left(\vec{x_i},\vec{x_{\text{red}}}\right)-b_{\text{c}}}{\sum_{i=1}^{2n}y_{\text{c}}^i\alpha_ic_i^{m+1}}\ . \tag{II.13}$$

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

II.1.2 Basic Comparison With ε-SVR

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

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

There is an interesting relation between δ-SVR and ε-SVR for the proposed new kernels (II.8). We can write inequality constraints for δ-SVR (II.4) as

$$-w_{\text{c},\vec{\text{red}}}\cdot\vec{x_i}-w_c^{m+1}\left(y_r^i-\delta\right)-b_c\geq 1-\xi_c^i \tag{II.14}$$

$$w_{\text{c},\vec{\text{red}}}\cdot\vec{x_i}+w_c^{m+1}\left(y_r^i+\delta\right)+b_c\geq 1-\xi_c^{*i} \tag{II.15}$$

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

$$-w_{\text{c},\vec{\text{red}}}\cdot\vec{x_i}-b_c\geq w_c^{m+1}y_r^i-w_c^{m+1}\delta+1-\xi_c^i \tag{II.16}$$

$$w_{\text{c},\vec{\text{red}}}\cdot\vec{x_i}+b_c\geq -w_c^{m+1}y_r^i-w_c^{m+1}\delta+1-\xi_c^{*i}\ . \tag{II.17}$$

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

$$\frac{-w_{\mathrm{c,red}}^{\vec{}} \cdot \vec{x_i} - b_c}{w_c^{m+1}} \geq y_r^i - \delta + \frac{1}{w_c^{m+1}} - \frac{\xi_c^i}{w_c^{m+1}} \tag{II.18}$$

$$\frac{w_{\mathrm{c,red}}^{\vec{}} \cdot \vec{x_i} + b_c}{w_c^{m+1}} \geq -y_r^i - \delta + \frac{1}{w_c^{m+1}} - \frac{\xi_c^{*i}}{w_c^{m+1}} \ . \tag{II.19}$$

After transforming the above into regression convention we get

$$y_r^i - g\left(\vec{x_i}\right) \leq \delta - \frac{1}{w_c^{m+1}} + \frac{\xi_c^i}{w_c^{m+1}} \ , \tag{II.20}$$

$$g\left(\vec{x_i}\right) - y_r^i \leq \delta - \frac{1}{w_c^{m+1}} + \frac{\xi_c^{*i}}{w_c^{m+1}} \ , \tag{II.21}$$

where

$$g\left(\vec{x}\right) = \frac{-w_{\mathrm{c,red}}^{\vec{}} \cdot \vec{x} - b_c}{w_c^{m+1}} \ . \tag{II.22}$$

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

$$y_r^i - g\left(\vec{x_i}\right) \geq \delta - \frac{1}{w_c^{m+1}} + \frac{\xi_c^i}{w_c^{m+1}} \ , \tag{II.23}$$

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

After changing notation from w_c^{m+1} to v, OP 9 can be formulated as

OP 10.

$$\min_{\vec{w_{\mathrm{r}}}, b_{\mathrm{r}}, \vec{\xi_{\mathrm{r}}}, \vec{\xi_{\mathrm{r}}^*}, v} \quad f\left(\vec{w_{\mathrm{r}}}, b_{\mathrm{r}}, \vec{\xi_{\mathrm{r}}}, \vec{\xi_{\mathrm{r}}^*}, v\right) = \left\|\vec{w_{\mathrm{r}}}, v\right\|^2 + C_{\mathrm{r}} \sum_{i=1}^{n} \left(\xi_{\mathrm{r}}^i + \xi_{\mathrm{r}}^{*i}\right) \tag{II.25}$$

subject to for $v > 0$

$$y_{\mathrm{r}}^i - g\left(\vec{x_i}\right) \leq \delta - \frac{1}{v} + \frac{\xi_r^i}{v} \tag{II.26}$$

$$g\left(\vec{x_i}\right) - y_r^i \leq \delta - \frac{1}{v} + \frac{\xi_r^{*i}}{v} \tag{II.27}$$

and for $v < 0$

$$-y_{\mathrm{r}}^i + g\left(\vec{x_i}\right) \leq -\delta - \frac{1}{-v} + \frac{\xi_r^i}{-v} \tag{II.28}$$

$$-g\left(\vec{x_i}\right) + y_r^i \leq -\delta - \frac{1}{-v} + \frac{\xi_r^{*i}}{-v} \tag{II.29}$$

and

$$\vec{\xi_{\mathrm{r}}} \geq 0 \tag{II.30}$$

$$\vec{\xi_{\mathrm{r}}^*} \geq 0 \tag{II.31}$$

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

$$g\left(\vec{x_i}\right) = \vec{w_{\mathrm{r}}} \cdot \vec{x_i} + b_{\mathrm{r}} \ . \tag{II.32}$$

We can notice that when we have the solution of OP 10, the same solution can be found by running OP 6 with the parameters set to: when $v^* > 0$

$$\varepsilon = \delta - \frac{1}{v^*} \tag{II.33}$$

and

$$C_r^{\mathrm{new}} = C_r v^* \ , \tag{II.34}$$

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

$$\varepsilon = -\delta + \frac{1}{v^*} \tag{II.35}$$

and

$$C_r^{\text{new}} = -C_r v^* \ . \tag{II.36}$$

Note that for $\varepsilon < 0$, we can replace (II.33) or (II.35) with

$$\varepsilon = 0 \tag{II.37}$$

due to the following proposition.

Proposition II.1.1. *OP 6 for $\varepsilon < 0$ returns the same solution as for $\varepsilon = 0$.*

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

$$\left\{p_e^1, p_{css}^1, p_{csc}\right\} \tag{II.38}$$

$$\left\{p_e^2, p_{css}^2, p_{csc}\right\} \ . \tag{II.39}$$

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

$$\left\{p_{er}, p_{css}^2, p_{css}^1, p_{csc}\right\} \tag{II.40}$$

$$\left\{p_{er}, p_{css}^1, p_{css}^2, p_{csc}\right\} \ . \tag{II.41}$$

For the first solution, the error difference is

$$\left(|p_{er}| + \left|p_{css}^2\right|\right)\Delta\varepsilon + \left(\left|p_{css}^1\right| + |p_{csc}|\right)\Delta\varepsilon \tag{II.42}$$

and for the second solution the error difference is

$$\left(|p_{er}| + \left|p_{css}^1\right|\right)\Delta\varepsilon + \left(\left|p_{css}^2\right| + |p_{csc}|\right)\Delta\varepsilon \ . \tag{II.43}$$

They are equal. □

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

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

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 (additionally with the $\vec{C_c}$ weight)

OP 11.

$$\min_{\vec{w_{\mathrm{c}}}, b_{\mathrm{c}}, \vec{\xi_{\mathrm{c}}}} \quad f\left(\vec{w_{\mathrm{c}}}, b_{\mathrm{c}}, \vec{\xi_{\mathrm{c}}}\right) = \frac{1}{2}\left\|\vec{w_{\mathrm{c}}}\right\|^2 + \vec{C_{\mathrm{c}}} \sum_{i=1}^{2n} \xi_{\mathrm{c}}^i \tag{II.44}$$

subject to

$$y_{\mathrm{c}}^i \left(\vec{w_{\mathrm{c,red}}} \cdot \vec{x_i} + w_c^{m+1}\left(y_r^i + y_c^i \delta_i\right) + b_c\right) \geq 1 - \xi_{\mathrm{c}}^i \tag{II.45}$$

$$\vec{\xi_{\mathrm{c}}} \geq 0 \tag{II.46}$$

for $i \in \{1, \ldots, 2n\}$, where $\delta_i = \delta_u^i$ for $i \in \{1, \ldots, n\}$ and $\delta_i = \delta_d^i$ for $i \in \{n+1, \ldots, 2n\}$.

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 equivalence 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 have a special distribution $\mathrm{F_c}\left(\vec{x_{\mathrm{c}}}\right)$ where random coordinate x_{m+1} is dependent on the remaining coordinates; for $\delta = 0$, $\mathrm{F_c}\left(\vec{x_{\mathrm{c}}}\right) \equiv \mathrm{F_r}\left(\vec{x_{\mathrm{r}}}, y_{\mathrm{r}}\right)$.

After transformation, Bayes optimal classification depends on a sign of

$$\Pr\left(1 \mid \vec{x_{\mathrm{c}}}\right) - \Pr\left(-1 \mid \vec{x_{\mathrm{c}}}\right) \quad . \tag{II.47}$$

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

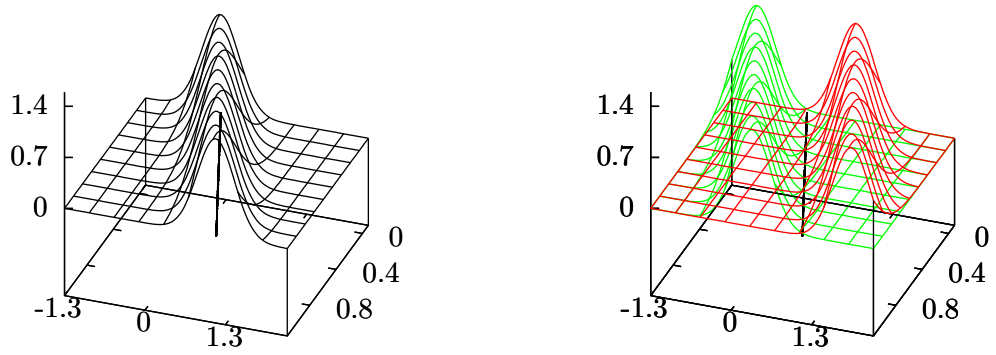
$$\mathrm{r}\left(\vec{x_{\mathrm{r}}}\right) = \mathbb{E}\left[y_{\mathrm{r}} \mid \vec{x_{\mathrm{r}}}\right] \quad . \tag{II.48}$$

Theorem II.2.1. *For a unimodal, symmetrical probability distribution* $\mathrm{F_r}\left(y_{\mathrm{r}} \mid \vec{x_{\mathrm{r}}}\right)$ *of original examples, Bayes decision boundary of the transformed classification problem is equivalent geometrically to the regression function for every* $\delta \geq 0$.

(A proof is in Appendix B.2). The example to the theorem is depicted in Fig. II.3. The theorem states that assuming symmetrical errors in the regression output, for any nonnegative value of δ the transformed classification problem is equivalent to the original one. The theorem can be extended to different unimodal, symmetrical distributions per point $\left(\mathrm{F_r^x}\left(y_{\mathrm{r}} \mid \vec{x_{\mathrm{r}}}\right)\right)$, with the same effect. The question arises about asymmetric distributions.

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

Theorem II.2.2. *When* $F\left(m+\delta\right) - F\left(m-\delta\right) \geq 0$ *is satisfied for some* $\delta > m$, *then for a unimodal, probability distribution* $\mathrm{F_r}\left(y_{\mathrm{r}} \mid \vec{x_{\mathrm{r}}}\right)$ *of original examples, Bayes decision boundary*



(a) (b)

Figure II.3: The idea of the transformation of the Bayes solution in δ-SVR. A solid line - a Bayes solution. (a) A density function before transformation. (b) Two density functions after transformation

of the transformed classification problem is equivalent geometrically to the regression function, where $m > 0$ is the expected value.

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

Theorem II.2.3. *For a unimodal, asymmetric probability distribution $\mathrm{F_r}\left(y_\mathrm{r} \mid \vec{x_\mathrm{r}}\right)$ of original examples, Bayes classification of the transformed classification problem is equivalent geometrically to the regression function for some ratio δ_u/δ_d for every $\delta_u \geq 0$.*

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

II.3 Generalization Performance of δ-SVR

The key point in statistical learning theory is the analysis of generalization capabilities of machine learning methods without assuming any particular data distribution, [49, 50]. 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

$$\mathrm{R}_{\mathrm{emp}}\left(\alpha_l\right) \underset{\delta\to\infty}{\to} 0 \ , \tag{II.49}$$

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

The ε-SVR and SVC realize a trade-off between ERM and minimizing a VC dimension which describes capacity of a learning machine. The ERM for ε-SVR is realized in a standard way by minimizing a sum of training errors. In SVC, ERM is realized by minimizing a sum of slack variables (I.4). Therefore for a particular value of δ, δ-SVR, which uses SVC, also minimizes a sum of slack variables. It does not minimize ERM for the regression. In the following subsection, we compare in details similarities and differences between ERM for classification and regression.

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.27) 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 will define examples involved in realization of ERM: for δ-SVR, let's call a vector lying on margin boundaries or inside margin boundaries, *an essential margin vector* and a set of such vectors for a particular hypothesis, EMV. For ε-SVR, let's call a vector lying on ε boundaries or outside ε boundaries, *an essential margin vector* and a set of such vectors for a particular hypothesis, EMV. By a configuration of essential margin vectors, called $CEMV$, we mean a list of essential margin vectors for a particular hypothesis, each with the distance to a margin boundary (or ε boundary).

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

Theorem II.3.1. *For $\|\vec{w}_{\mathrm{c}}\| = p$, 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 a margin boundary is $\xi_i / \|\vec{w}_c\|$. Because the denominator is constant, minimizing distances to examples lying outside margin boundaries means minimizing a sum of ξ_i. □

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

functions. Now when we know how ERM is computed for ε-SVR and δ-SVR for specific values of δ and ε, we analyze which examples are involved in computing ERM.

First, we propose

Proposition 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 $p = 1/(1 + w_c^{m+1}\Delta\delta)$. This proposition states that the same $CEMV$ could be present for multiple values of δ. This is a difference from ε-SVR where every $CEMV$ is present only once for one value of ε. We can also notice that the distance to the margin boundaries from the solution for δ_2 is $1/\|p\vec{w_c}\|$.

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

Proposition II.3.3. *Every CEMV for ε-SVR for a particular value ε_s is present in classification setting in δ-SVR for every $\delta > \delta_p$, where $\delta_p = \varepsilon_s$.*

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

Proposition 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 are absent from ε-SVR.

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

The $CEMV$ of δ-SVR where $|EMV| > n$ are absent from ε-SVR. It can be noticed that when $|EMV| > n$, there exists an equivalent EMV for regression when taking into account optimization for support vectors stated in (II.13). It is a consequence of the fact that all support vectors for SVC lying below margin boundaries have $\alpha_i = C$, which is a conclusion from Karush-Kuhn-Tucker complementary condition for SVC; therefore they disappear in (II.13) and are not support vectors for δ-SVR. E.g. when $|EMV|$ is close to $2n$ we get the small number of support vectors for δ-SVR.

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

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

$$d_c^i = \frac{\xi_c^i}{\|\vec{w_c}\|} \ . \tag{II.50}$$

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

$$f\left(\vec{w_c}, b_c, \vec{\xi_c}\right) = \|\vec{w_c}\|^2 + C_c \|\vec{w_c}\| \sum_{i=1}^{n} d_c^i \ . \tag{II.51}$$

When we treat the differences between distances in the last coordinate direction and perpendicular distances negligible, then we can see that the difference between the cost function for ε-SVR (I.27) and the above for δ-SVR is $\|\vec{w_c}\|$. For ε-SVR a trade-off between the empirical risk and the capacity is controlled by C_r, for δ-SVR we can also control the trade-off with C_c, but additionally it depends on $\|\vec{w_c}\|$. It means that when the capacity decreases (a geometric margin increases), a value of the trade-off also decreases.

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

$$f\left(\vec{w_c}, b_c, \vec{\xi_c}\right) = p^2 \left\|\vec{w_c}\right\|^2 + C_c p \left\|w_c\right\| \sum_{i=1}^{n} d_c^i \ , \tag{II.52}$$

where

$$p = \frac{1}{1 + w_c^{m+1} \left(\delta_2 - \delta_1\right)} \ , \tag{II.53}$$

so

$$f\left(\vec{w_c}, b_c, \vec{\xi_c}\right) = p^2 \left(\left\|\vec{w_c}\right\|^2 + \frac{C_c}{p} \left\|w_c\right\| \sum_{i=1}^{n} d_c^i\right) \ , \tag{II.54}$$

$$f\left(\vec{w_c}, b_c, \vec{\xi_c}\right) = p^2 \left(\left\|\vec{w_c}\right\|^2 + \left(1 + w_c^{m+1} \left(\delta_2 - \delta_1\right)\right) C_c \left\|w_c\right\| \sum_{i=1}^{n} d_c^i\right) \ . \tag{II.55}$$

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

II.3.3 VC Bounds for δ-SVR

In this subsection, we consider the translation for δ-SVR independently of the data distribution. The generalization bounds based on a VC dimension are as following, [50]. With the probability at least $1 - \eta$ the inequality holds true

$$\mathrm{R}\left(\alpha\right) \leq \mathrm{R_{emp}}\left(\alpha\right) + \frac{\varepsilon\left(n\right)}{2} \left(1 + \sqrt{1 + \frac{4\mathrm{R_{emp}}\left(\alpha\right)}{\varepsilon\left(n\right)}}\right) \ , \tag{II.56}$$

where

$$\varepsilon\left(n\right) = 4\frac{\ln 2\tau + 1}{\tau} - \frac{\ln \eta/4}{n} \tag{II.57}$$

$$\tau = \frac{n}{h} \ , \tag{II.58}$$

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

$$\mathrm{R}\left(\alpha\right) \leq \mathrm{R_{emp}}\left(\alpha\right) + \frac{B\varepsilon\left(n\right)}{2} \left(1 + \sqrt{1 + \frac{4\mathrm{R_{emp}}\left(\alpha\right)}{B\varepsilon\left(n\right)}}\right) \ . \tag{II.59}$$

Therefore the bounds for classification and regression are pretty much the same. They are independent of data distribution. The key to minimize the right hand side is to control h. For this purpose Vapnik [50] proposed SRM. For SVC, it is realized by controlling the trade-off between ERM and CM. Let's see the relation of CM to h.

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

$$\min_i \left|\vec{w_c} \cdot \vec{x_i}\right| = 1 \ . \tag{II.60}$$

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

$$h \leq \min\left(R^2\Lambda^2, m + 1\right) \ , \tag{II.61}$$

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

proof can be found in [43]. So minimization of $\|\vec{w_c}\|$ is a minimization of the upper bound on h.

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

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

$$h \leq p^2\Lambda^2 \left(R + \Delta\delta\right)^2 \quad , \tag{II.62}$$

where

$$p = \frac{1}{1 + w_c^{m+1}\Delta\delta} \quad . \tag{II.63}$$

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

$$p^2 \left(R + \Delta\delta\right)^2 < R^2 \quad . \tag{II.64}$$

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

$$w_c^{m+1} > \frac{1}{R} \quad . \tag{II.65}$$

For $p < 0$

$$w_c^{m+1} < \frac{-2}{\Delta\delta} - \frac{1}{R} \quad . \tag{II.66}$$

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

$$x_c^{m+1} = -w_c^m x_c^m / w_c^{m+1} \quad . \tag{II.67}$$

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

Therefore δ-SVR has a potential to improve a VC bound by shifting without worsening empirical risk (II.49).

Let's consider VC bounds for δ-SVR and ε-SVR. We start the analysis from the classification problem introduced by δ-SVR for some value of δ. The δ-SVR uses SVC to solve it, so we realize CM by using the term $\|w_c\|^2$. The ε-SVR can be interpreted as δ-SVR with lack of the last variable, see OP 10. 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, [24], except stock price data,

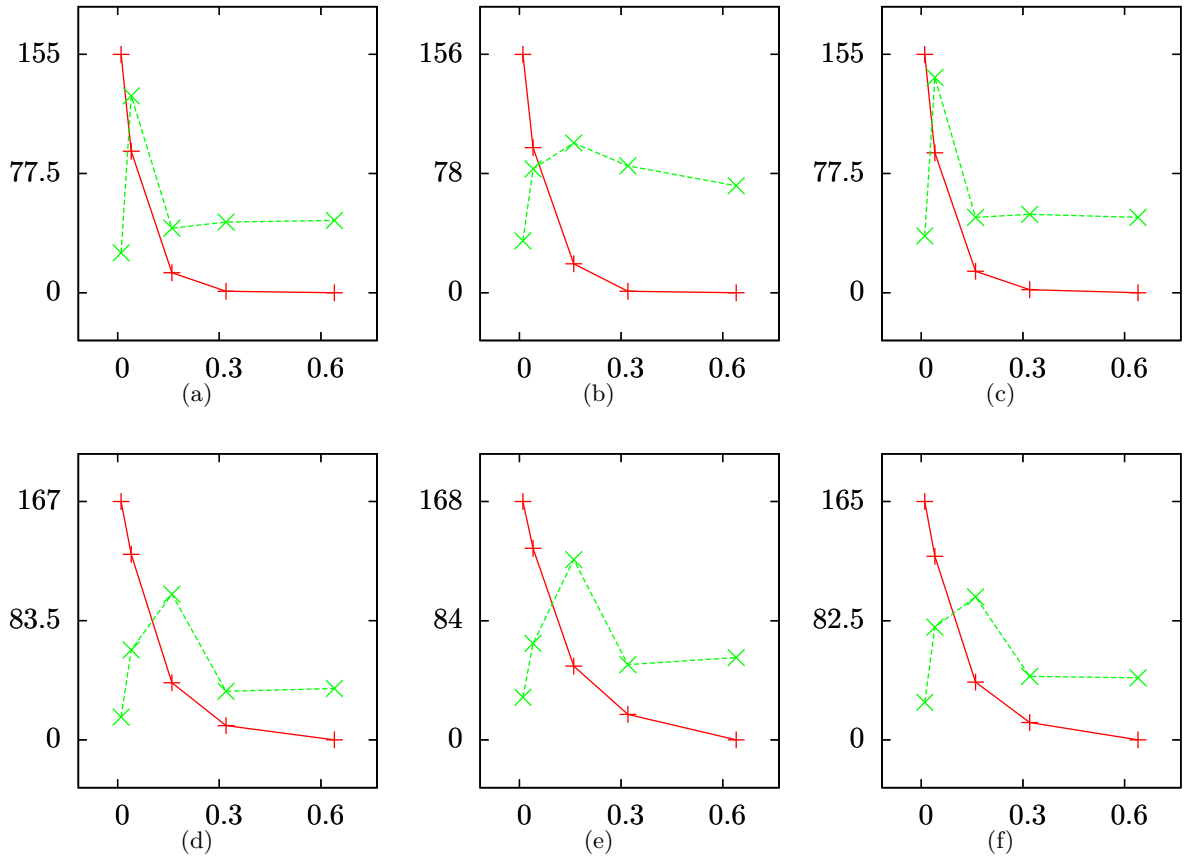


Figure II.4: Relation between ε, δ and the number of support vectors for the test cases with ids 0-5 from Table II.3a. A function with '+' points represents the relationship between value of ε and the number of support vectors, a function with 'x' points represents the relationship between value of δ and the number of support vectors

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 satisfying $n/h < 20$. Recently, Yang et al. [57] used double cross-validation for SVM. We use double cross-validation for comparing performance of δ-SVR with ε-SVR, 5 fold inner cross-validation is used. Outer cross-validation is slightly modified it order to allow using a small training set size: if a training set size is less than a half of all known mappings, then we use cross-validation but for training data, otherwise we use standard cross-validation. When it is greater then the number of possible steps for cross-validation additional data shuffles are performed.

In the first experiment, we check the theoretical result from Prop. II.3.4 by comparing the number of support vectors and generalization performance for some values of δ and ε. In the second experiment, we compare the generalization performance for variable δ and ε.

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.4, Fig. II.5.

We can see that for ε-SVR the number of support vectors decreases while increasing ε. We

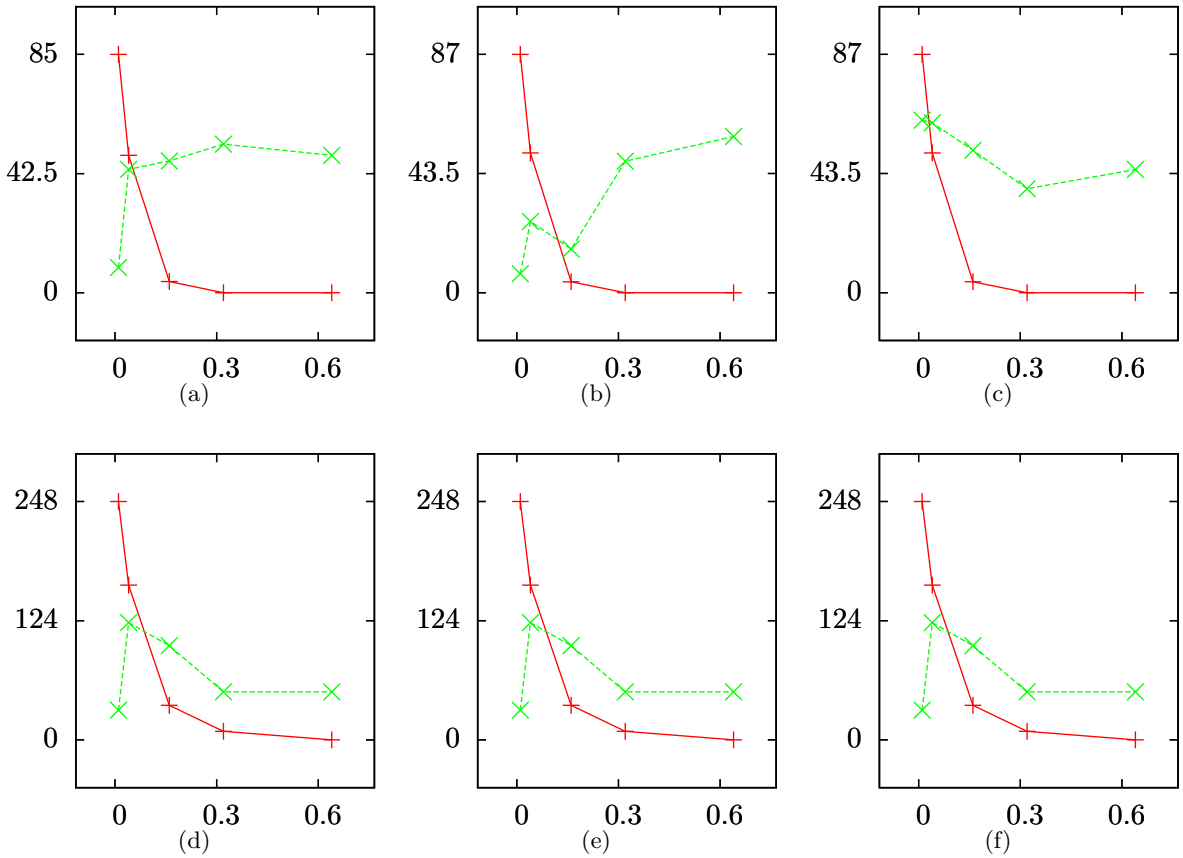


Figure II.5: Relation between ε, δ and the number of support vectors for the test cases with ids 6-11 from Table II.3a, cont. A function with '+' points represents the relationship between value of ε and the number of support vectors, a function with 'x' points represents the relationship between value of δ and the number of support vectors

Table II.1: Relation between ε, δ and RMSE. Column descriptions: *id* – an id of a test (for synthetic s prefix, for real world data sets t prefix), ε,δ – a value of a parameter ε or δ, *ti* – a percentage difference in RMSE between ε-SVR and δ-SVR for the *i*-th test, positive means that δ-SVR has smaller RMSE

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

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

In the second part of the first experiment, we compare generalization performance for δ-SVR and ε-SVR for various values of δ and ε. Performance for δ-SVR is better than ε-SVR for various δ and ε, except a value closest to zero, for which a performance is worse (Table II.1, Fig. II.6, Fig. II.7). We can also notice greater difference in the performance when the difference in the number of support vectors is bigger. We can see that the performance of δ-SVR is similar and close to the best value for any δ in a checked range of values, while performance of ε-SVR decreases while increasing ε.

The results may be valuable in practice when we lack of enough time to find the best values

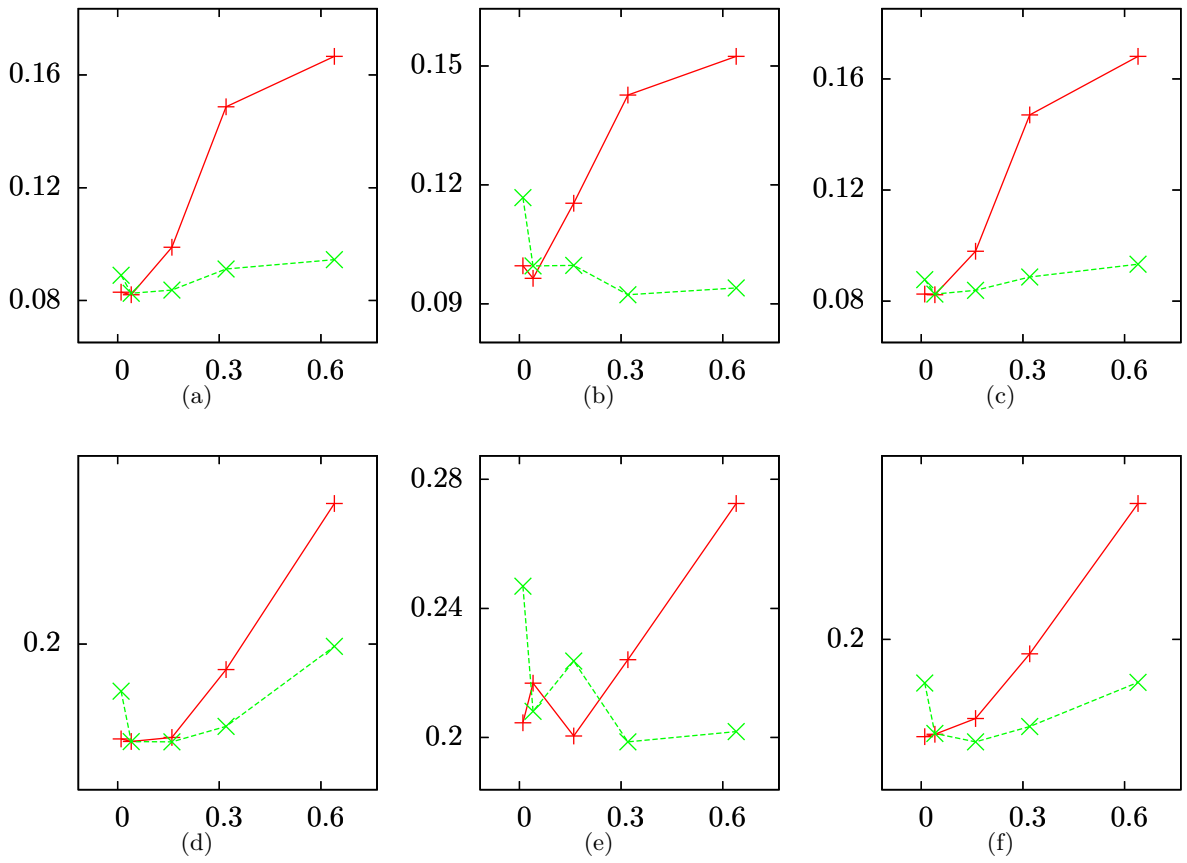


Figure II.6: Relation between ε, δ and RMSE for the test cases with ids 0-5 from Table II.3a. A function with '+' points represents the relationship between value of ε and RMSE, a function with 'x' points represents the relationship between value of δ and RMSE.

of δ and ε. Then we expect better generalization performance of δ-SVR than ε-SVR. In the next experiment we will compare results for variable δ and ε.

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. We limit the number of iterations to 10000 in these tests for both methods, for results without this limit and additional description of parameters of the tests see [39].

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

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, except two tests for a polynomial kernel when ε-SVR is better. However, we can notice the improved number of support vectors for δ-SVR for 10 tests out of 12 tests, for one of them ε-SVR is better with statistical significance. The statistical significance for the number of support vectors is achieved for δ-SVR without the limitation of 10000 iterations and for variable σ parameter for RBF kernel for about half of the tests, see the results in [39].

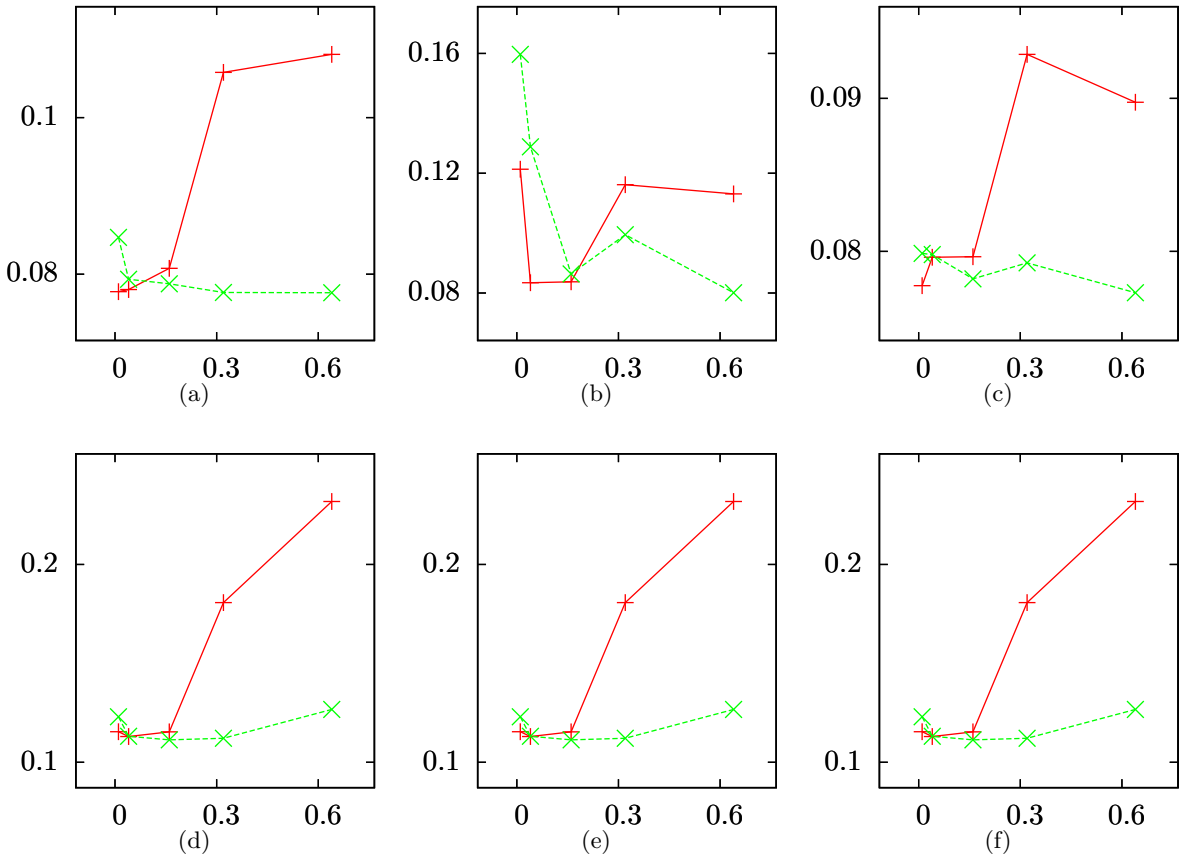


Figure II.7: Relation between ε, δ and RMSE for the test cases with ids 6-11 from Table II.3a, cont. A function with '+' points represents the relationship between value of ε and RMSE, a function with 'x' points represents the relationship between value of δ and RMSE.

Table II.2: Performance of δ-SVR for synthetic data. Column descriptions: *id* – id of the test, *a function* – a function used for generating data $y_1 = \sum_{i=1}^{\dim} x_i$, $y_4 = \left(\sum_{i=1}^{\dim} x_i\right)^{\text{kerP}}$, $y_5 = 0.5 \sum_{i=1}^{\dim} \sin 10x_i + 0.5$, *ker* – a kernel with a kernel parameter (for a polynomial kernel it is a dimension, for the RBF kernel it is σ), *idRef* – a reference to the test, *te12M* – a percent average difference in MSE for testing data, if greater than 0 than δ-SVR is better, *teT* – t value for the t-test for comparing testing error, *s1* – the average number of support vectors for ε-SVR, *s2* – the average number of support vectors for δ-SVR, *sT* – t value for the t-test for comparing the number of support vectors

(a)

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

(b)

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

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 support vectors. Thus we get simpler predictive models. Therefore, computational time of

Table II.3: Performance of δ-SVR for real world data. Column descriptions: *id* – id of the test, *dn* – a data set, *ker* – a kernel with a kernel parameter (for a polynomial kernel it is a dimension, for the RBF kernel it is σ), *idRef* – a reference to the test, *te12M* – a percentage average difference in MSE for testing data, if greater than 0 than δ-SVR is better, $teT - t$ value for the t-test for comparing testing error, *s1* – the average number of support vectors for ε-SVR, *s2* – the average number of support vectors for δ-SVR, $sT - t$ value for the t-test for comparing the number of support vectors

(a)

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

(b)

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

testing new examples is decreased. The next advantage is smaller generalization error over different values of ε and δ. Therefore, there exists possibility to decrease time of training, but with accepting suboptimal solutions. The next advantage is faster time of training for linear kernels while using SMO solver. The last advantage of δ-SVR, but not least, is the possibility of replacing the standard SVC classification method by any other classification method based on kernel functions. In particular, any improvements for classification methods in respect of generalization error, speed of training and testing, ability to incorporate prior knowledge, can be used directly for regression problems.

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

Knowledge About a Margin

We define the margin of an example (sometimes called just a margin) in the following way:

Definition III.0.1. Given some curve $h(\vec{x}) = 0$, *the margin of the* $\vec{x_p}$ *example* is defined as a value $|h(\vec{x_p})|$.

For hard margin SVC, OP 1, the margin of the closest examples is equal to 1. *The knowledge about the margin of an example* (sometimes called the knowledge about a margin) is defined as prior information about the margins of particular examples.

III.1 Introduction to φ-SVC

The φ-SVC method is a recently proposed method of incorporating knowledge about the margin of an example to SVC, [33, 35, 37]. The φ-SVC optimization problem is defined with an additional parameter per example added in the right side of the inequality (I.5). Another modification of inequality constraints was proposed in [56]. The authors modify the inequalities 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.

Now, we will closely look at φ-SVC optimization problem. We define φ-SVC optimization problem based on SVC with cost weights per example OP 7 as

OP 12.

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

subject to

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

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

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

$$\vec{C}_c \gg 0 \tag{III.4}$$

$$\varphi_i \in \mathbb{R} \tag{III.5}$$

$$h(\vec{x_i}) = \vec{w}_c \cdot \vec{x_i} + b_c \ . \tag{III.6}$$

The new weights φ_i are present only in (III.2). When $\vec{\varphi} = 0$, the OP 12 is equivalent to OP 7. 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 $\vec{C}_c/(1+\varphi)$.

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 $y_c^i h(\vec{x_i}) \geq 1/d - \xi_c^i/d$. After multiplying by d we get $y_c^i d h(\vec{x_i}) \geq 1 - \xi_c^i$. 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 $\vec{C}_{\mathrm{c}} = d\vec{C}_{\mathrm{c}}$ and with the new decision curve $dh(\vec{x}) = 0$. □

We can notice that when neglecting ξ_{c}^i, we get different bounds for the margin: when $y_{\mathrm{c}}^i = 1$ and $g(\vec{x_i}) \geq 0$, then we get a lower bound $1 + \varphi_i$, when $y_{\mathrm{c}}^i = -1$ and $g(\vec{x_i}) \geq 0$, then we get an upper bound $-(1 + \varphi_i)$, when $y_{\mathrm{c}}^i = 1$ and $g(\vec{x_i}) < 0$, then we get an upper bound $-(1 + \varphi_i)$, when $y_{\mathrm{c}}^i = -1$ and $g(\vec{x_i}) < 0$, then we get a lower bound $1 + \varphi_i$. We can distinguish three cases: $1 + \varphi_i > 0$, $1 + \varphi_i < 0$ and $1 + \varphi_i = 0$. For the first one, we get a lower bound on the margin equal to $1 + \varphi_i$. For the second, we get an upper bound on the margin equal to $-(1 + \varphi_i)$, for the third, we get an upper bound on the margin equal to 0. Therefore, by using φ_i weights, we can incorporate knowledge about the margin of an example.

The next property of φ_i weights is that they have impact on the distance between the curve $h(\vec{x}) = 0$ and the i-th example. We can conduct the same analysis as above for distances by dividing both sides of (III.2) by $\|\vec{w}_{\mathrm{c}}\|$, so we get $y_{\mathrm{c}}^i h(\vec{x_i}) / \|\vec{w}_{\mathrm{c}}\| \geq 1/\|\vec{w}_{\mathrm{c}}\| + \varphi_i/\|\vec{w}_{\mathrm{c}}\|$. The conclusion is similar: for the case when $1 + \varphi_i > 0$, we get a lower bound on the distance equal to $(1 + \varphi_i)/\|\vec{w}_{\mathrm{c}}\|$. For the case when $1 + \varphi_i < 0$, we get an upper bound on the distance equal to $-(1 + \varphi_i)/\|\vec{w}_{\mathrm{c}}\|$. For the case when $1 + \varphi_i = 0$, we get an upper bound on the distance equal to 0.

Note that when we take into account ξ_{c}^i, we can see that knowledge about the margin of an example is incorporated as imperfect prior knowledge. The violation of knowledge about the margin of an example is controlled by the C_{c}^i parameters. Comparing loosely φ_i weights with slack variables: φ_i weights are constant, they are absent in the objective function, whereas a sum of slack variables is minimized as part of the objective function.

We can also derive the equivalent optimization problem to OP 12, where φ_i weights are present in the constraints with slack variables

OP 13.

$$\min_{\vec{w}_{\mathrm{c}}, b_{\mathrm{c}}, \vec{\xi_{\mathrm{c}}}} \quad f\left(\vec{w}_{\mathrm{c}}, b_{\mathrm{c}}, \vec{\xi_{\mathrm{c}}}\right) = \frac{1}{2}\|\vec{w}_{\mathrm{c}}\|^2 + \vec{C}_{\mathrm{c}} \cdot \vec{\xi}_{\mathrm{c}} \tag{III.7}$$

subject to

$$y_{\mathrm{c}}^i h(\vec{x_i}) \geq 1 - \xi_{\mathrm{c}}^i \tag{III.8}$$

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

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

In order to construct an efficient algorithm for the OP 12 its dual form was derived (derivation in C.1). The final form of the dual problem is

OP 14.

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

subject to

$$\vec{\alpha} \cdot \vec{y_{\mathrm{c}}} = 0 \tag{III.11}$$

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

where

$$Q_{ij} = y_{\mathrm{c}}^i y_{\mathrm{c}}^j (\vec{x_i} \cdot \vec{x_j}) \tag{III.13}$$

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

It differs from the original SVC dual form OP 3 by only $\vec{\alpha} \cdot \vec{\varphi}$ term (also here we use C_{c}^i weights instead of C_{c}). 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

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

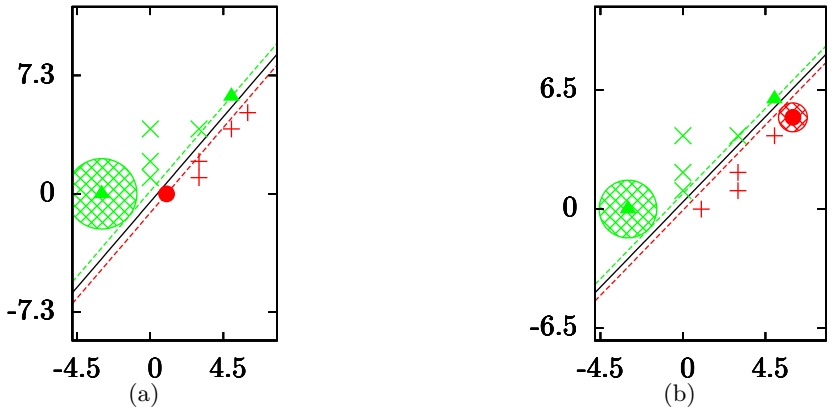


Figure III.1: Interpretation of knowledge about a margin as dynamic hyperspheres. Points - examples, solid lines - solutions, triangles and circles - support vectors, circles filled with grid pattern - dynamic hyperspheres. The example in (-3, 0) has $\varphi_i = 5.0$ and it is a marginal support vector. The hyperspheres must not cross the appropriate margin boundaries. (a) $C = 100$. (b) $C = 10$ and $\varphi_i = 2.5$ in (6,5). A radius of a dynamic hypersphere in (-3, 0) differs in both cases, in (a) it is 2.2, and in (b) it is 1.6

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

$$\alpha_i \left(y_{\mathrm{c}}^i h(\vec{x_i}) - 1 - \varphi_i + \xi_{\mathrm{c}}^i \right) = 0 \tag{III.15}$$

$$(C_i - \alpha_i)\, \xi_{\mathrm{c}}^i = 0 \ . \tag{III.16}$$

The conclusions from (III.15) and (III.16) are: when $\alpha_i = 0$, then $\xi_{\mathrm{c}}^i = 0$, when $0 < \alpha_i < C_{\mathrm{c}}$, then $\xi_{\mathrm{c}}^i = 0$ and $y_{\mathrm{c}}^i h(\vec{x_i}) = 1 + \varphi_i$, when $\alpha_i = C_{\mathrm{c}}$, then $y_{\mathrm{c}}^i h(\vec{x_i}) = 1 + \varphi_i - \xi_{\mathrm{c}}^i$. Moreover, when $\xi_{\mathrm{c}}^i > 0$, then $\alpha_i = C_{\mathrm{c}}$ and $y_{\mathrm{c}}^i h(\vec{x_i}) = 1 + \varphi_i - \xi_{\mathrm{c}}^i$, when $y_{\mathrm{c}}^i h(\vec{x_i}) > 1 + \varphi_i - \xi_{\mathrm{c}}^i$, then $\alpha_i = 0$, $\xi_{\mathrm{c}}^i = 0$ and $y_{\mathrm{c}}^i h(\vec{x_i}) > 1 + \varphi_i$. We can find ξ_{c}^i parameters from the solution of the dual form as following. When

$$y_{\mathrm{c}}^i h^*(\vec{x_i}) \geq 1 + \varphi_i \ , \tag{III.17}$$

then $\xi_{\mathrm{c}}^i = 0$, else

$$\xi_{\mathrm{c}}^i = 1 + \varphi_i - y_{\mathrm{c}}^i h(\vec{x_i}) \ . \tag{III.18}$$

III.2 Knowledge About the Margin as Dynamic Hyperspheres

Knowledge about the margin of the p-th example can be visualized as a hypersphere with the center in the p point and a radius equal to $\varphi_p / \|\vec{w_{\mathrm{c}}}\|$. We call it *a dynamic hypersphere*, because a radius depends on $\|\vec{w_{\mathrm{c}}}\|$. 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 location), the hyperspheres are respectively $S_1(\vec{x_p}, r)$, and $S_2(\vec{x_p}, r/a)$, Fig. III.1. Knowledge about the margin of an example in the form of a lower bound on the margin can be interpreted as a constraint that the hypersphere must not cross the appropriate margin boundary, Fig. III.1. Knowledge about the margin of an example in the form of an upper bound on the margin can be interpreted as a constraint that the hypersphere must cross the appropriate margin boundary, Fig. III.2.

III.3 Solving φ-SVC Optimization Problem

For solving OP 14, a decomposition method similar to SMO, [40] was derived. In every step, two parameter subproblems are solved. Heuristic and stopping criterion are based on KKT conditions.

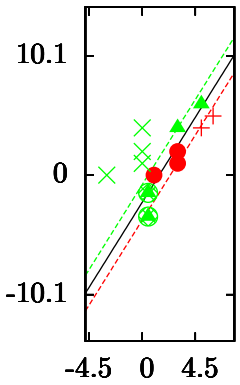


Figure III.2: Interpretation of knowledge about a margin as dynamic hyperspheres. Points - examples, a solid line - a solution, triangles and circles - support vectors, circles filled with grid pattern - dynamic hyperspheres. The example in (0.5, -1.5) has $\varphi_i = -1.0$. Its hypersphere must cross the appropriate margin boundary

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 that lies on the optimal margin boundaries ($y_i h^* (\vec{x_i}) = 1$) or below optimal margin boundaries ($y_i h^* (\vec{x_i}) < 1$) could be a support vector. In φ-SVC, also the example satisfying $\varphi_i > 0$ and lying above margin boundaries ($y_i h^* (\vec{x_i}) > 1$) could be a support vector. Such example is called *marginal support vector*, Fig. III.1. Because an output model is formulated based on support vectors, introducing the new type of support vectors leads to richer models with additional examples lying above optimal margin boundaries.

Examples with upper bounded margins could be the examples that lie below the margin boundary and are not support vectors. Moreover, such examples could lead to the new type of support vectors – that have slack variables equal to zero and lie below the margin boundaries.

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

III.5 Reformulation of ε-SVR as φ-SVC

We have found that the ε-SVR is a special case of φ-SVC, [37]. The similar reformulation was previously 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 φ-SVC also for ε-SVR.

Here we present a reformulation of the OP 6 (derivation in C.2). 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 15.

$$\min_{\vec{w_c}, b_c, \vec{\xi_c}} \quad f\left(\vec{w_c}, b_c, \vec{\xi_c}\right) = \frac{1}{2}\left\|\vec{w_c}\right\|^2 + C_c \sum_{i=1}^{2n} \xi_c^i \tag{III.19}$$

subject to

$$y_c^i h\left(\vec{x_i}\right) \geq 1 - \xi_c^i + \varphi_i \tag{III.20}$$

$$\vec{\xi_c} \geq 0 \tag{III.21}$$

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

$$h\left(\vec{x_i}\right) = \vec{w_c} \cdot \vec{x_i} + b_c \tag{III.22}$$

and

$$\varphi_i = y_c^i y_r^i - \varepsilon - 1 \ . \tag{III.23}$$

The OP 15 is a special case of OP 12. Instead of using a decision curve of OP 15 we use a regression function

$$\sum_{i=1}^{2n} y_{\mathrm{c}}^{i}\alpha_{c,i}^{*}K\left(\vec{x_i},\vec{x}\right)+b_{\mathrm{c}}^{*}=0 \quad\rightarrow\quad g^{*}\left(\vec{x}\right)=\sum_{i=1}^{n} y_{\mathrm{c}}^{i}\alpha_{r,i}^{*}K\left(\vec{x_i},\vec{x}\right)+\sum_{i=n+1}^{2n} y_{\mathrm{c}}^{i}\beta_{r,i-n}^{*}K\left(\vec{x_i},\vec{x}\right)+b_{r}^{*}\ , \tag{III.24}$$

where $\alpha_{r,i}^{*}=\alpha_{c,i}^{*}$, $\beta_{r,i-n}^{*}=\alpha_{c,i}^{*}$, $b_{r}^{*}=b_{\mathrm{c}}^{*}$. Because data are duplicated we can merge the sums in the final regression and we get

$$g^{*}\left(\vec{x}\right)=\sum_{i=1}^{n}\left(\alpha_{r,i}^{*}-\beta_{r,i}^{*}\right)K\left(\vec{x_i},\vec{x}\right)+b_{r}^{*}\ . \tag{III.25}$$

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 15. 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. The KKT complementary condition for ε-SVR is the same as for φ-SVC after reformulation. From (III.15) and (III.16), we get

$$\alpha_i\left(h\left(\vec{x_i}\right)-y_r^i+\varepsilon+\xi_i\right)=0 \tag{III.26}$$

$$\beta_i\left(-h\left(\vec{x_i}\right)+y_r^i+\varepsilon+\xi_i^*\right)=0 \tag{III.27}$$

$$\left(C_{\mathrm{c}}-\alpha_i\right)\xi_i=0\ . \tag{III.28}$$

$$\left(C_{\mathrm{c}}-\beta_i\right)\xi_i^*=0\ . \tag{III.29}$$

We can also analyze some properties of the variable $\gamma_r^i=\alpha_{r,i}-\beta_{r,i}$. When $\gamma_r^i=-C_{\mathrm{c}}$, then $\alpha_i=0$ and $\beta_i=C_{\mathrm{c}}$, so $g\left(\vec{x_i}\right)-y_r^i\geq\varepsilon$. When $\gamma_r^i=C_{\mathrm{c}}$, then $\alpha_i=C_{\mathrm{c}}$ and $\beta_i=0$, so $g\left(\vec{x_i}\right)-y_r^i\leq\varepsilon$. When $\gamma_r^i=0$, then $\alpha_i=\beta_i$. When $\alpha_i=\beta_i<C_{\mathrm{c}}$, then $\xi_i=0$ and $\xi_i^*=0$. Because $-\varepsilon-\xi_r^*\leq g\left(\vec{x_i}\right)-y_r^i\leq\varepsilon+\xi_r^*$, so $-\varepsilon\leq g\left(\vec{x_i}\right)-y_r^i\leq\varepsilon$. The case when $\alpha_i=\beta_i=C_{\mathrm{c}}$ is possible only when $\xi_i+\xi_i^*=-2\varepsilon$, then $g\left(\vec{x_i}\right)-y_r^i=-\varepsilon-\xi_i$. When $-C_{\mathrm{c}}<\gamma_r^i<0$, then $\alpha_i<\beta_i$, so $\alpha_i<C_{\mathrm{c}}$ and $\beta_i>0$. From KKT we have

$$-g\left(\vec{x_i}\right)+y_r^i+\varepsilon+\xi_i^*=0$$

and

$$\alpha_i\left(g\left(\vec{x_i}\right)-y_r^i+\varepsilon\right)=0$$

Merging both we get

$$\alpha_i\left(2\varepsilon+\xi_i^*\right)=0$$

So $\xi_i^*=-2\varepsilon$ or $\alpha_i=0$. For the second case, $\beta<C_{\mathrm{c}}$, because $\gamma_r^i>-C_{\mathrm{c}}$, so

$$g\left(\vec{x_i}\right)-y_r^i=\varepsilon\ .$$

Finally for the case when $0<\gamma_r^i<C_{\mathrm{c}}$, then $\alpha_i>\beta_i$, so $\alpha_i>0$ and $\beta_i<C_{\mathrm{c}}$. From KKT we have

$$g\left(\vec{x_i}\right)-y_r^i+\varepsilon+\xi_i=0$$

and

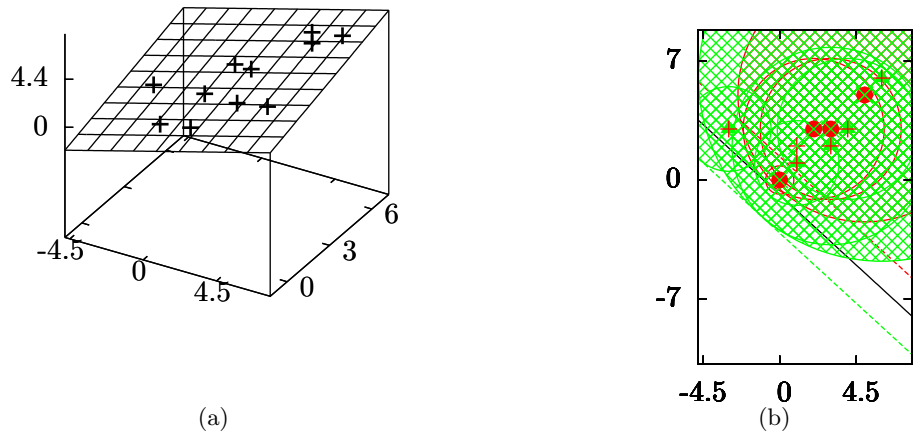
$$\beta_i\left(-g\left(\vec{x_i}\right)+y_r^i+\varepsilon\right)=0$$

Merging both we get

$$\beta_i\left(2\varepsilon+\xi_i\right)=0$$

So $\xi_i=-2\varepsilon$ or $\beta_i=0$. For the second case, $\alpha<C_{\mathrm{c}}$, because $\gamma_r^i<C_{\mathrm{c}}$, so

$$g\left(\vec{x_i}\right)-y_r^i=-\varepsilon\ .$$



(a) (b)

Figure III.3: The idea of reformulating ε-SVR as φ-SVC. (a) Points - regression examples, a plane - a solution. (b) The transformed problem. Points - classification examples, triangles and circles - support vectors, circles filled with grid pattern - dynamic hyperspheres, a solid line - a solution, dashed lines - margin boundaries

III.6 Using Knowledge About a Margin with ε-SVR

The ε-SVR can be reformulated as φ-SVC. We can additionally modify margin weights for incorporating knowledge about a margin. 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 into φ-SVC or we can modify ε_u^p or ε_d^p in OP 8 before the transformation. In the first option, the φ_p and φ_{p+n} weights are already set according to (III.23). So adding knowledge about a margin 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

$$\Delta\varphi_p = -\Delta\varepsilon_d^p \ , \tag{III.30}$$

$$\Delta\varphi_{p+n} = -\Delta\varepsilon_u^p \ . \tag{III.31}$$

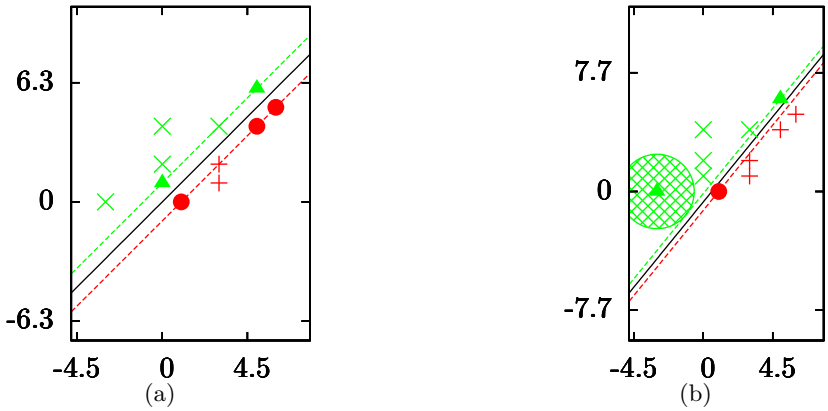
III.7 Using Knowledge About a Margin with δ-SVR

The δ-SVR is transformed into the classification problem, so we can use φ-SVC instead of standard SVC for incorporating knowledge about a margin. 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

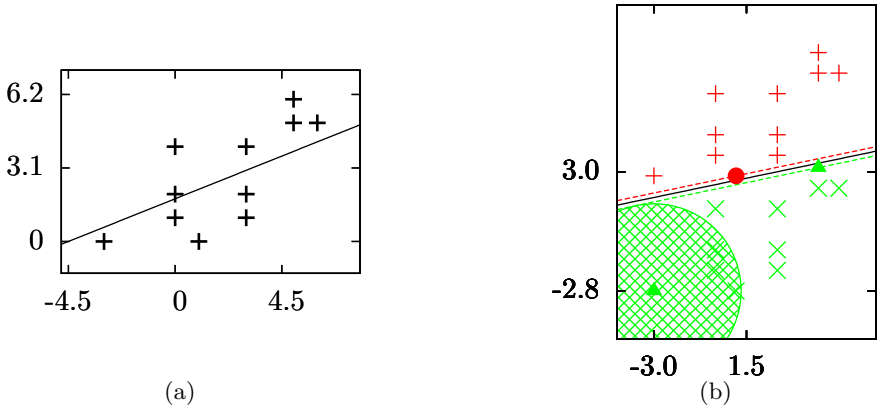
After modification of φ_i weights, either the output curve stays the same, or it is changed. For example, let's assume that we modify only a one example $\vec{p}$ and $\varphi_{\vec{p}}$ is equal to zero before the modification. When $y_{\vec{p}}h^*(\vec{p}) > 1$, then after setting $0 < \varphi_{\vec{p}} \leq y_{\vec{p}}h^*(\vec{p}) - 1$ the solution remains the same. 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 geometric margin (increase of $\|\vec{w}_c\|$), 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.

Let's now analyze in details changing the output curve. First consider changing the $\vec{C}_c$ parameter. We can see that for OP 7, 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 12 and the solution with all



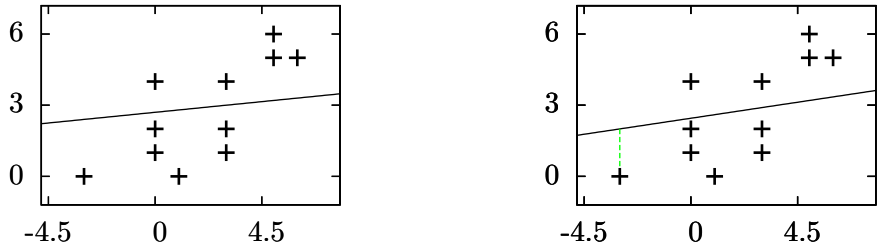
(a) (b)

Figure III.4: Direct curve manipulation for SVC. Points - examples, triangles and circles - support vectors, circles filled with grid pattern - dynamic hyperspheres, solid lines - solutions, dashed lines - margin boundaries. (b) The example with φ_i weight caused lowering the solution from (a)



(a) (b)

Figure III.5: Direct curve manipulation for δ-SVR. (a) Points - regression examples, a solid line - a solution. (b) The transformed problem. Points - classification examples, triangles and circles - support vectors, circles filled with grid pattern - dynamic hyperspheres, a solid line - a solution, dashed lines - margin boundaries. The example with φ_i weight caused changing the solution from (a)



(a) (b)

Figure III.6: Direct curve manipulation for ε-SVR. Points - regression examples, solid lines - solutions. (b) A dotted line must cross the solution. The example with φ_i weight caused changing the solution from (a)

$\varphi_i = 0$. When we set $\varphi_p > 0$ for some example p, such as

$$\varphi_p \leq y_c^p h\left(x_c^p\right) - 1 + \xi_c^p \ , \tag{III.32}$$

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

$$\xi_{cnew}^p = \varphi_p - y_{\mathrm{c}}^p h\left(\vec{x_p}\right) + 1 \ . \tag{III.33}$$

   The objective function will raise by

$$\Delta f = C_p \Delta \xi_{\mathrm{c}}^p \ . \tag{III.34}$$

2. The $h\left(\vec{x_{\mathrm{c}}}\right)$ is modified

$$h_{cnew}\left(\vec{x_{\mathrm{c}}}\right) = ch\left(\vec{x_{\mathrm{c}}}\right) \ , \tag{III.35}$$

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

$$\Delta f = \frac{1}{2}\left\|\vec{w}\right\|^2 \left(c^2 - 1\right) \ . \tag{III.36}$$

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

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

   Note that in this case other non-zero ξ_{c}^i values have to be modified

$$\xi_{\mathrm{c}}^{iold} = 1 - y_{\mathrm{c}}^i h\left(\vec{x_i}\right) \tag{III.38}$$

$$\xi_{\mathrm{c}}^{inew} = 1 - c y_{\mathrm{c}}^i h\left(\vec{x_i}\right) \tag{III.39}$$

$$\Delta \xi_{\mathrm{c}}^i = y_{\mathrm{c}}^i h\left(\vec{x_i}\right)(1 - c) \ . \tag{III.40}$$

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

$$\Delta f = \frac{1}{2}\left\|\vec{w}\right\|^2 \left(c^2 - 1\right) + \sum_{i=1, i\neq p}^{n} C_{\mathrm{c}}^i \Delta \xi_{\mathrm{c}}^i \ . \tag{III.41}$$

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

$$\Delta f = \frac{1}{2}\left\|\vec{w}\right\|^2 \left(c^2 - 1\right) + C_{\mathrm{c}}^p \Delta \xi_{\mathrm{c}}^p + \sum_{i=1, i\neq p}^{n} C_{\mathrm{c}}^i \Delta \xi_{\mathrm{c}}^i \ . \tag{III.42}$$

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 knowledge about the margin of an example is the incorporation of the additional constraint in the form of a constant value of linear dependency on function values, that the solution must satisfy, for regression it is

$$\sum_{i=1}^{s} s_i g\left(\vec{d_i}\right) = e \ , \tag{III.43}$$

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, g is defined in (I.32), and for classification it is

$$\sum_{i=1}^{s} s_i h\left(\vec{d}_i\right) = e \ . \tag{III.44}$$

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. Knowledge about the margin of an example is used in incorporation of this constraint to φ-SVC and ε-SVR.

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

Incorporating (III.44) to OP 12 leads to the φ-SVC optimization problem without the offset, which is the SVC optimization problem without the offset OP 4 with additional margin weights (and $\vec{C}_\mathrm{c}$ weights for completeness)

OP 16.

$$\min_{\vec{w}_\mathrm{c}, \vec{\xi}_\mathrm{c}} \quad f\left(\vec{w}_\mathrm{c}, \vec{\xi}_\mathrm{c}\right) = \frac{1}{2}\left\|\vec{w}_\mathrm{c}\right\|^2 + \vec{C}_\mathrm{c} \cdot \vec{\xi}_\mathrm{c} \tag{III.45}$$

subject to

$$y_\mathrm{c}^i h\left(\vec{x}_i\right) \geq 1 - \xi_\mathrm{c}^i + \varphi_i \tag{III.46}$$

$$\vec{\xi} \geq 0 \tag{III.47}$$

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

$$\vec{C}_\mathrm{c} \gg 0 \tag{III.48}$$

$$\varphi_\mathrm{c}^i \in \mathbb{R} \tag{III.49}$$

$$h\left(\vec{x}_i\right) = \vec{w}_\mathrm{c} \cdot \vec{x}_i \ . \tag{III.50}$$

The dual problem compared to OP 5 contains additionally margin weights in an objective function (derivation in Appendix C.3)

OP 17.

$$\max_{\vec{\alpha}} \quad d\left(\vec{\alpha}\right) = \vec{\alpha} \cdot \left(1 + \vec{\varphi}\right) - \frac{1}{2}\vec{\alpha}^T Q \vec{\alpha} \tag{III.51}$$

subject to

$$0 \leq \vec{\alpha} \leq \vec{C}_\mathrm{c} \tag{III.52}$$

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

Incorporating (III.44) to OP 12 also leads to the new kernel in the form of transformation of the input vectors (derivation in Appendix C.4)

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

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

This is a symmetrical kernel. We set the margin weights as

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

We propose also a nonlinear kernel by further kernelization of (III.54), we get

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

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

This is also a symmetrical kernel. In the final solution, we use the transformation of the input vectors (III.53) only for the training vectors present in the term $\vec{w_c}$, therefore for the solution, we get the following kernel

$$K\left(\vec{x_j},\vec{x}\right) = K_o\left(\vec{x_j},\vec{x}\right) - \frac{1}{\sum_{i=1}^{s} s_i}\sum_{i=1}^{s} s_i K_o\left(\vec{d}_i,\vec{x}\right) \quad , \tag{III.58}$$

where $j \in \{1,\ldots,n\}$. Note that this is a kernel between a training vector and $\vec{x}$ used only for defining the solution. For the final solution, we compute the offset as

$$b = \frac{1}{\sum_{i=1}^{s} s_i}\left(e - \sum_{i=1}^{s} s_i \sum_{j=1}^{n} \alpha_j y_c^j K_o\left(\vec{x_j},\vec{d}_i\right)\right) \tag{III.59}$$

$$+\frac{1}{\left(\sum_{i=1}^{s} s_i\right)^2}\sum_{i=1}^{s} s_i \sum_{j=1}^{n}\sum_{k=1}^{s} \alpha_j y_c^j s_k K_o\left(\vec{d}_k,\vec{d}_i\right) \quad . \tag{III.60}$$

III.9.2 Incorporating the Linear Dependency on Function Values to ε-SVR

We incorporated (III.43) to OP 6. 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

$$\varphi_i = y_c^i y_r^i - \varepsilon - 1 - y_c^i \frac{e}{\sum_{i=1}^{s} s_i} \tag{III.61}$$

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

III.9.3 Incorporating Linear Dependency on Function Values to δ-SVR

For completeness we show here the incorporation of linear dependency on function values to δ-SVR. We do not use margin weights in this case. We use kernels developed for δ-SVR, (II.8). Incorporating (III.43) to δ-SVR leads to the SVC optimization problem without the offset OP 4 with the new kernel (derivation in Appendix C.5)

$$K\left(\vec{x},\vec{y}\right) = \left(\vec{x_{\text{red}}} - \frac{1}{\sum_{i=1}^{s} s_i}\sum_{i=1}^{s} s_i \vec{d_{\text{i,red}}}\right)\left(\vec{y_{\text{red}}} - \frac{1}{\sum_{i=1}^{s} s_i}\sum_{i=1}^{s} s_i \vec{d_{\text{i,red}}}\right) \tag{III.62}$$

$$+\left(x_{m+1} - \frac{e}{\sum_{i=1}^{s} s_i}\right)\left(y_{m+1} - \frac{e}{\sum_{i=1}^{s} s_i}\right) \quad , \tag{III.63}$$

where $\vec{d_{i,\text{red}}} = (d_i^1,\ldots,d_i^m)$. This is a symmetrical kernel. We propose also a nonlinear kernel by further kernelization of (III.62), we get

$$K\left(\vec{x},\vec{y}\right) = K_o\left(\vec{x_{\text{red}}},\vec{y_{\text{red}}}\right) - \frac{1}{\sum_{i=1}^{s} s_i}\sum_{i=1}^{s} s_i K_o\left(\vec{d_{\text{i,red}}},\vec{x_{\text{red}}}\right) \tag{III.64}$$

$$-\frac{1}{\sum_{i=1}^{s} s_i}\sum_{i=1}^{s} s_i K_o\left(\vec{d_{\text{i,red}}},\vec{y_{\text{red}}}\right) + \frac{1}{\left(\sum_{i=1}^{s} s_i\right)^2}\sum_{i=1}^{s}\sum_{j=1}^{s} s_i s_j K_o\left(\vec{d_{\text{i,red}}},\vec{d_{\text{j,red}}}\right) \tag{III.65}$$

$$+ x_{m+1}y_{m+1} - x_{m+1}\frac{e}{\sum_{i=1}^{s} s_i} - y_{m+1}\frac{e}{\sum_{i=1}^{s} s_i} + \frac{e^2}{\left(\sum_{i=1}^{s} s_i\right)^2} \quad . \tag{III.66}$$

This is also a symmetrical kernel. In the final solution, we use the kernelization process only for the training vectors present in the term $\vec{w_c}$, therefore for the solution, we get the following

kernel

$$K\left(\vec{x_j}, \vec{x}\right) = K_o\left(x_{\mathrm{j,red}}^{\rightarrow}, x_{\mathrm{red}}^{\rightarrow}\right) - \frac{1}{\sum_{i=1}^{s} s_i} \sum_{i=1}^{s} s_i K_o\left(d_{\mathrm{i,red}}^{\rightarrow}, x_{\mathrm{red}}^{\rightarrow}\right) + \left(x_j^{m+1} - \frac{e}{\sum_{i=1}^{s} s_i}\right) x_{m+1} \quad , \tag{III.67}$$

where $j \in \{1, \ldots, n\}$. Note that this is a kernel between a training vector and $\vec{x}$ used only for defining the solution. For the final solution, we compute the offset as

$$b_{\mathrm{c}} = -\frac{1}{\sum_{i=1}^{s} s_i} e \sum_{j=1}^{2n} \alpha_j y_{\mathrm{c}}^j \left(x_j^{m+1} - \frac{e}{\sum_{i=1}^{s} s_i}\right) - \frac{1}{\sum_{i=1}^{s} s_i} \sum_{i=1}^{s} s_i \sum_{j=1}^{2n} \alpha_j y_{\mathrm{c}}^j K_o\left(x_{\mathrm{j,red}}^{\rightarrow}, d_{\mathrm{i,red}}^{\rightarrow}\right) \tag{III.68}$$

$$+ \frac{1}{\sum_{i=1}^{s} s_i} \sum_{i=1}^{s} s_i \sum_{j=1}^{2n} \alpha_j y_{\mathrm{c}}^j \frac{1}{\sum_{i=1}^{s} s_i} \sum_{k=1}^{s} s_k K_o\left(d_{\mathrm{k,red}}^{\rightarrow}, d_{\mathrm{i,red}}^{\rightarrow}\right) \quad . \tag{III.69}$$

III.10 Incorporating Inequalities with Function Values

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

$$g\left(\vec{x_i}\right) \geq a_i \tag{III.70}$$

for $i = 1..n$, where a_i are some parameters, g is defined in (I.32). We propose soft incorporation by changing φ_i values. For ε-SVR, it leads to the modification of the ε_u^i value in OP 8:

$$\varepsilon_{\mathrm{u,new}}^i = \min\left(\varepsilon_{\mathrm{u,old}}^i, y_r^i - a_i\right) \tag{III.71}$$

Proof. From (I.41) we have

$$g\left(\vec{x_i}\right) \geq y_r^i - \varepsilon_u^i - \xi_r^i \tag{III.72}$$

It is a soft incorporation so

$$g\left(\vec{x_i}\right) \geq y_r^i - \varepsilon_u^i \tag{III.73}$$

So we should set ε_u^i such as

$$y_r^i - \varepsilon_u^i \geq a_i \tag{III.74}$$

$$\varepsilon_u^i \leq y_r^i - a_i \tag{III.75}$$

so

$$\varepsilon_{\mathrm{u,new}}^i = \min\left(\varepsilon_{\mathrm{u,old}}^i, y_r^i - a_i\right) \tag{III.76}$$

□

For δ-SVR, we set a value of the δ_d^i parameter in OP 11:

$$\delta_{\mathrm{d,new}}^i = \min\left(\delta_{\mathrm{d,old}}^i, y_r^i - a_i\right) \quad . \tag{III.77}$$

Proof. After shifting data the barrier for the function is the shifted point. It will be more restrict barrier, because it is a barrier for the margin boundary. It implies the barrier for the function. We have

$$y_r^i - \delta_d^i \geq a_i \tag{III.78}$$

$$\delta_d^i \leq y_r^i - a_i \quad . \tag{III.79}$$

So we modify the shifting parameter δ_d^i as

$$\delta_{\mathrm{d,new}}^i = \min\left(\delta_{\mathrm{d,old}}^i, y_r^i - a_i\right) \quad . \tag{III.80}$$

□

Although we do not modify directly φ_i weights, 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

$$g\left(\vec{x_i}\right) \leq a_i \quad . \tag{III.81}$$

For ε-SVR, it leads to the modification of the ε_d^i value in OP 8:

$$\varepsilon_{\mathrm{d,new}}^i = \min\left(\varepsilon_{\mathrm{d,old}}^i, a_i - y_r^i\right) \tag{III.82}$$

Proof. From (I.42) we have

$$g\left(\vec{x_i}\right) \leq \varepsilon_d^i + y_r^i + \xi_r^i \tag{III.83}$$

It is a soft incorporation so

$$g\left(\vec{x_i}\right) \leq \varepsilon_d^i + y_r^i \tag{III.84}$$

So we should set ε_d^i such as

$$\varepsilon_d^i + y_r^i \leq a_i \tag{III.85}$$

$$\varepsilon_d^i \leq a_i - y_r^i \tag{III.86}$$

so

$$\varepsilon_{\mathrm{d,new}}^i = \min\left(\varepsilon_{\mathrm{d,old}}^i, a_i - y_r^i\right) \tag{III.87}$$

□

For δ-SVR, we set a value of the δ_u^i parameter in OP 11:

$$\delta_{\mathrm{u,new}}^i = \min\left(\delta_{\mathrm{u,old}}^i, a_i - y_r^i\right) \tag{III.88}$$

Proof. We have

$$y_r^i + \delta_u^i \leq a_i \tag{III.89}$$

$$\delta_u^i \leq a_i - y_r^i \tag{III.90}$$

So we modify the shifting parameter δ_u^i as

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

□

We can also increase values of proper C_i parameters to improve fulfillment of the constraints.

Another incorporation type is of the same form as (III.70) but defined for a new point $\vec{x_{n+1}}$ without defined y_r^{n+1}

$$g\left(\vec{x_{n+1}}\right) \geq a_{n+1} \quad , \tag{III.92}$$

where a_{n+1} is a parameter, g is defined in (I.32). We propose soft incorporation. We will use a special version of ε-SVR, where we allow presence of only one constraint either (I.41) or (I.42). So for (III.92) we will have only one constraint that is (I.41). We set

$$y_r^{n+1} = a_{n+1} \tag{III.93}$$

After substituting above to (III.71) we get

$$\varepsilon_{\mathrm{u,new}}^{n+1} = \min\left(\varepsilon_{\mathrm{u,old}}^{n+1}, 0\right) \tag{III.94}$$

One constraint in the formulation means that while transforming to the φ-SVC only one classification point will be created, in this case with $y_c^{n+1} = 1$.

For δ-SVR, we propose also soft incorporation. We will use a special version of δ-SVR, where we allow presence of points which are not duplicated but shifted either up or down. We define

such point with $y_c^{n+1} = -1$ and we set

$$y_r^{n+1} = a_{n+1} \tag{III.95}$$

and we shift it down by

$$\delta_{\text{d,new}}^{n+1} = \min\left(\delta_{\text{d,old}}^{n+1}, 0\right) \quad . \tag{III.96}$$

For

$$g\left(\vec{x_{n+1}}\right) \leq a_{n+1} \quad , \tag{III.97}$$

For (III.97) we will have only one constraint that is (I.42). We set

$$y_r^{n+1} = a_{n+1} \tag{III.98}$$

After substituting above to (III.82), we get

$$\varepsilon_{\text{d,new}}^{n+1} = \min\left(\varepsilon_{\text{d,old}}^{n+1}, 0\right) \tag{III.99}$$

One constraint in the formulation means that while transforming to the φ-SVC only one classification point will be created, in this case with $y_c^{n+1} = -1$.

For δ-SVR, we define a point with $y_c^{n+1} = 1$ and we set

$$y_r^{n+1} = a_{n+1} \tag{III.100}$$

and we shift it up by

$$\delta_{\text{u,new}}^{n+1} = \min\left(\delta_{\text{u,old}}^{n+1}, 0\right) \quad . \tag{III.101}$$

III.11 Reduce a Model with φ-SVC

Various methods for reducing the complexity of the output model were widely investigated, [19]. In particular, the reduction by removing support vectors was also analyzed in [16] for regression problems.

Sparse models have an advantage of faster post-processing such as testing new examples. For highly nonseparable data with a linear decision boundary, SVC has multiple support vectors. The general idea of constructing even more sparse solutions than SVC is to find a solution spanned on the given number of support vectors as close to the original SVC solution as possible. The most representative method for this idea is presented in [55]. Another example for this group are reduced support vector machines (RSVM), which randomly selects support vectors to a reduced set, [15, 29]. The alternative approach is to replace SVM with another training method that is designed for returning sparse solutions. The most representative method for this group is a greedy approach that adds basis functions to the solution until no progress in optimizing a cost function is made, [19].

Our proposed approach, [35, 37], is conceptually closer to the first idea. After SVM is trained, we remove randomly selected data 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 [16]. 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 knowledge about the margin of an example. The procedure of generating knowledge about the margin of an example is as following. First, φ_i weights are automatically generated from an existing solution as

$$\varphi_i = y_i h^*\left(\vec{x_i}\right) - 1 \tag{III.102}$$

for all training examples. For ε-SVR, instead of modifying φ_i weights, we can alternatively

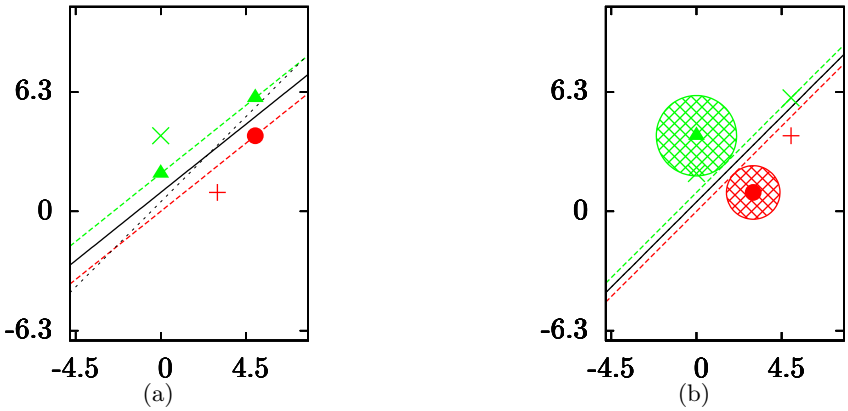


Figure III.7: The idea of reduced SVC. Points - examples after reducing, triangles and circles - support vectors, solid lines - solutions, dotted lines - original solutions before reduction, dashed lines - margin boundaries. (b) Circles filled with grid pattern - dynamic hyperspheres

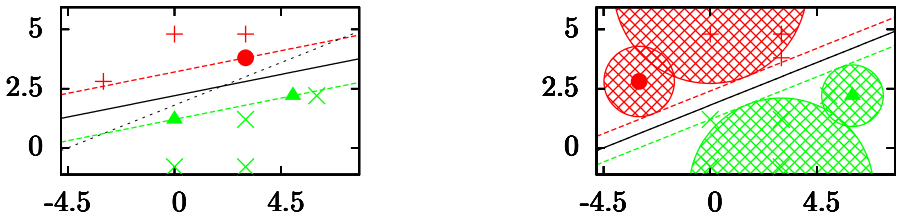


Figure III.8: The idea of reduced δ-SVR. Points - examples after reducing, triangles and circles - support vectors, solid lines - solutions, dotted lines - original solutions before reduction, dashed lines - margin boundaries. (b) Circles filled with grid pattern - dynamic hyperspheres

modify ε_u^i and ε_d^i weights in OP 8:

$$\varepsilon_u^i = y_r^i - g^*\left(\vec{x_i}\right) \quad ,$$

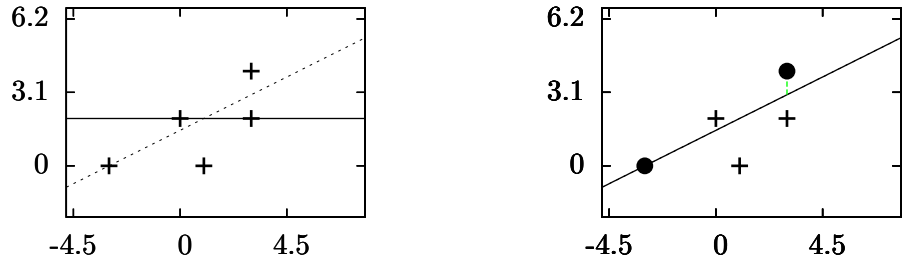
$$\varepsilon_d^i = g^*\left(\vec{x_i}\right) - y_r^i \quad .$$

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. Not that in this set, there are also support vectors that will be removed. Finally, we run φ-SVC with a reduced data set.

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. In [28], the authors divide a data set to two sets (about 20% and 80%), and they generate manually prior knowledge from the first set, and use it in the other. We proposed a slightly modified procedure. We generate knowledge from the whole training data set, and then we remove some of data vectors. The procedure is as follows:

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



(a) (b)

Figure III.9: The idea of reduced ε-SVR. Points - examples after reducing, filled circles - support vectors, solid lines - solutions, dotted lines - original solutions before reduction. (b) The dashed lines must cross the solution

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

III.13 Experiments

In the first experiment, we compare the performance of the SVM without and with knowledge about the margin of an example depends on the removal ratio p. For comparison purposes, a reduced model is the same for both methods. We use our implementation of all methods. In the second experiment, we compare the performance of both methods related to various number of support vectors.

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, δ 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 a particular data set it is possible to use more accurate grid searches than for massive tests with multiple number of simulations.

III.13.1 First Experiment

For synthetic data, we compare both methods on data generated from particular functions with added Gaussian noise for output values. For $p = 70$, the SVC with knowledge about a margin has smaller generalization error for every test except one, Table III.1. The testing performance gain varies from 0% to 3%. The number of support vectors is slightly smaller for the method with knowledge about a margin. For the regression case, for $p = 70$, the ε-SVR with knowledge about a margin has smaller generalization error for every test, Table III.2. The testing performance gain varies from 0% to 50%. The number of support vectors is bigger for the method with knowledge about a margin.

The real world data sets were taken from the LibSVM site, [3, 24], 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 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. We can see that for variable p, the SVM with knowledge about a margin has generally smaller generalization error than without, Fig. III.10, Fig. III.11, Fig. III.12, Fig. III.13. For $p = 70$, the SVC with knowledge about a margin has smaller generalization error for all data sets, with smaller or equal number of support vectors in 8 out of 12 tests, Table III.3. The

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 $y_1 = \sum_{i=1}^{\dim-1} x_i$, $y_4, y_5 = \left(\sum_{i=1}^{\dim-1} x_i\right)^{\text{kerP}}$, $y_6 = 0.5 \sum_{i=1}^{\dim-1} \sin 10x_i + 0.5$, *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 correctly classified examples for testing data, if greater than 0 than a method with knowledge about a margin is better, *s1* – the average number of support vectors for a method without knowledge about a margin, *s2* – the average number of support vectors for a method with knowledge about a margin

(a)

id	function	ker
0	y_1	denseLinear 0.0
1	$y_2 = 3y_1$	denseLinear 0.0
2	$y_3 = 1/3y_1$	denseLinear 0.0
3	y_4	densePolynomial 5.0
4	y_5	denseRBF 0.25
5	y_6	denseRBF 0.25

(b)

idRef	te12M	s1	s2
0	2.4	23	19
1	3.22	23	19
2	−0.63	24	21
3	1.19	23	16
4	0.92	25	20
5	0.43	26	25

Table III.2: 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 $y_1 = \sum_{i=1}^{\dim-1} x_i$, $y_2 = \left(\sum_{i=1}^{\dim-1} x_i\right)^{\text{kerP}}$, $y_3 = 0.5 \sum_{i=1}^{\dim-1} \sin 10x_i + 0.5$, *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 knowledge about a margin is better, *s1* – the average number of support vectors for a method without knowledge about a margin, *s2* – the average number of support vectors for a method with knowledge about a margin

(a)

id	function	ker
0	y_1	denseLinear 0.0
1	y_2	densePolynomial 5.0
2	y_3	denseRBF 0.25

(b)

idRef	te12M	s1	s2
0	50.05	11	20
1	0.98	42	52
2	4.6	41	54

testing performance gain varies from 0% to 25%. For a regression case, for $p = 70$, the ε-SVR with knowledge about a margin has smaller generalization error for every test, Table III.4. The testing performance gain varies from 0% to 34%. The number of support vectors is bigger for the method with knowledge about a margin, for all regression tests.

We achieve smaller generalization error for the method with knowledge about a margin. We confirmed that knowledge about a margin in deed is able to preserve information about the original solution.

III.13.2 Second experiment

In the second experiment, we compare performance of the method without and with knowledge about a margin, but related to the number of support vectors. We performed tests for classification and regression. We can notice that the method with knowledge about a margin tends to achieve smaller generalization error for the similar number of support vectors, although for the RBF kernel the results are similar (Fig. III.14, Fig. III.15, Fig. III.16, Fig. III.17).

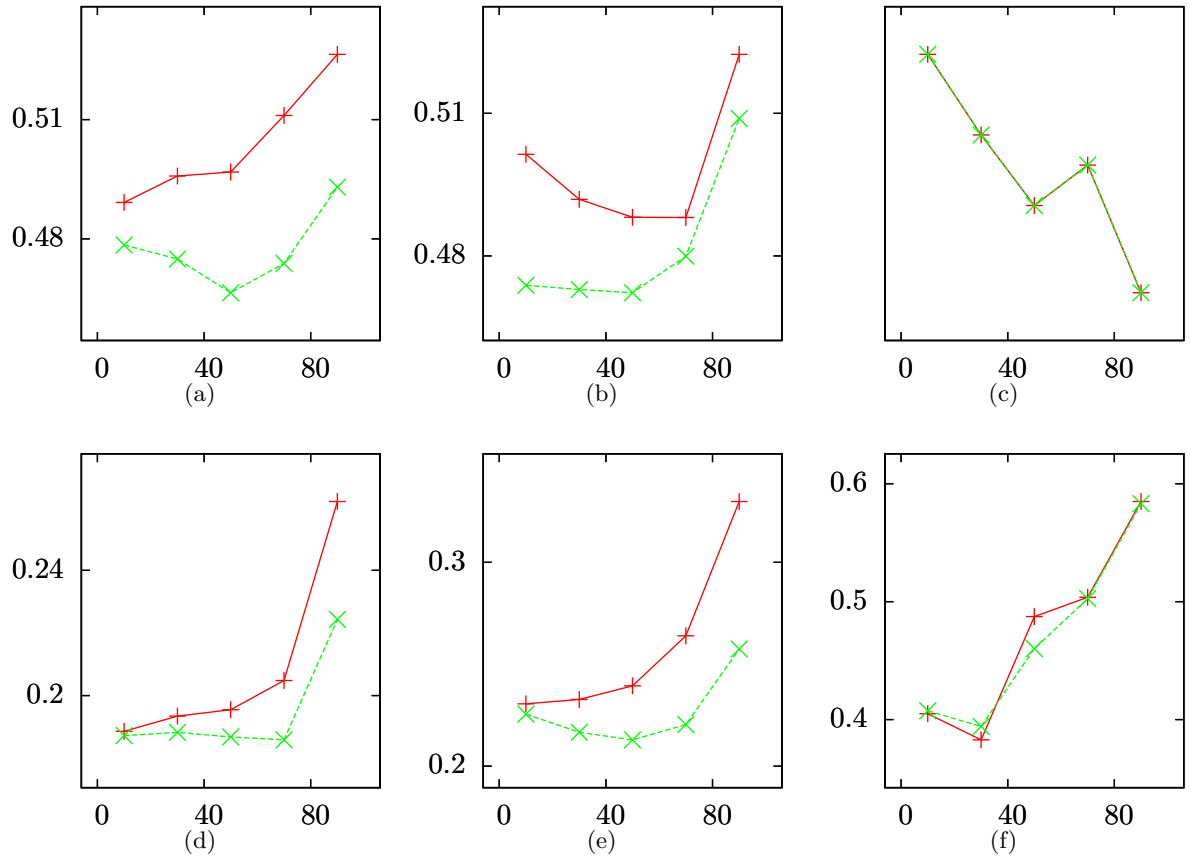


Figure III.10: Comparison of two methods of removing support vectors for the test cases with ids 0-5 from Table III.3. The x axis - the percent of removed data, y axis - a percent difference in misclassified testing examples, the line with '+' points - a random removing method, the line with 'x' points - proposed removing method with knowledge about a margin

III.14 Summary

In this thesis, we analyzed the possibility to preserve knowledge about the original solution by using margin weights while creating reduced models. We also used knowledge about the margin of an example for incorporating the additional nonlinear constraint to the problem. We also showed that the method with knowledge about the margin of an example can achieve smaller generalization error for similar number of support vectors, for reduced models. A potential list of applications for margin weights is much broader. In future research, we plan to investigate possibility to create knowledge about a margin by experts and to apply it for time series data. Moreover, we plan to use it for combining SVM classifiers with each other in order to decrease generalization error.

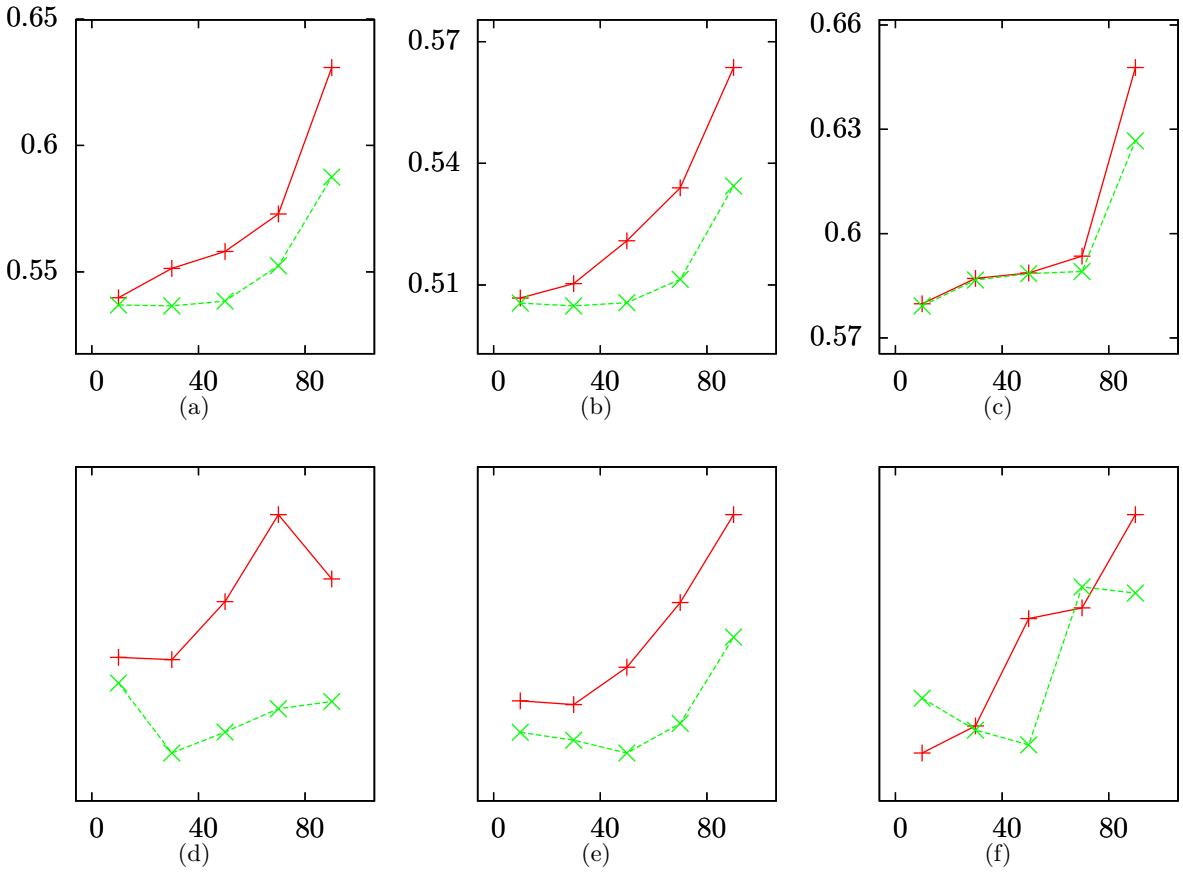


Figure III.11: Comparison of two methods of removing support vectors for the test cases with ids 6-11 from Table III.3, cont. The x axis - the percent of removed data, y axis - a percent difference in misclassified testing examples, the line with '+' points - a random removing method, the line with 'x' points - proposed removing method with knowledge about a margin

Table III.3: 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 knowledge about a margin is better, *s1* – the average number of support vectors for a method without knowledge about a margin, *s2* – the average number of support vectors for a method with knowledge about a margin

(a)

id	dn	ker
0	a1aAll	denseLinear 0.0
1	a1aAll	densePolynomial 5.0
2	a1aAll	denseRBF 0.00813
3	breast-cancer	denseLinear 0.0
4	breast-cancer	densePolynomial 3.0
5	breast-cancer	denseRBF 0.1
6	diabetes	denseLinear 0.0
7	diabetes	densePolynomial 3.0
8	diabetes	denseRBF 0.125
9	djia	denseLinear 0.0
10	djia	densePolynomial 5.0
11	djia	denseRBF 0.08333

(b)

idRef	te12M	s1	s2
0	15.47	16	16
1	3.79	11	25
2	0.0	27	27
3	14.26	7	8
4	24.71	5	6
5	3.61	24	25
6	9.41	20	17
7	7.16	16	15
8	0.42	26	26
9	0.67	23	16
10	1.26	22	16
11	1.52	29	28

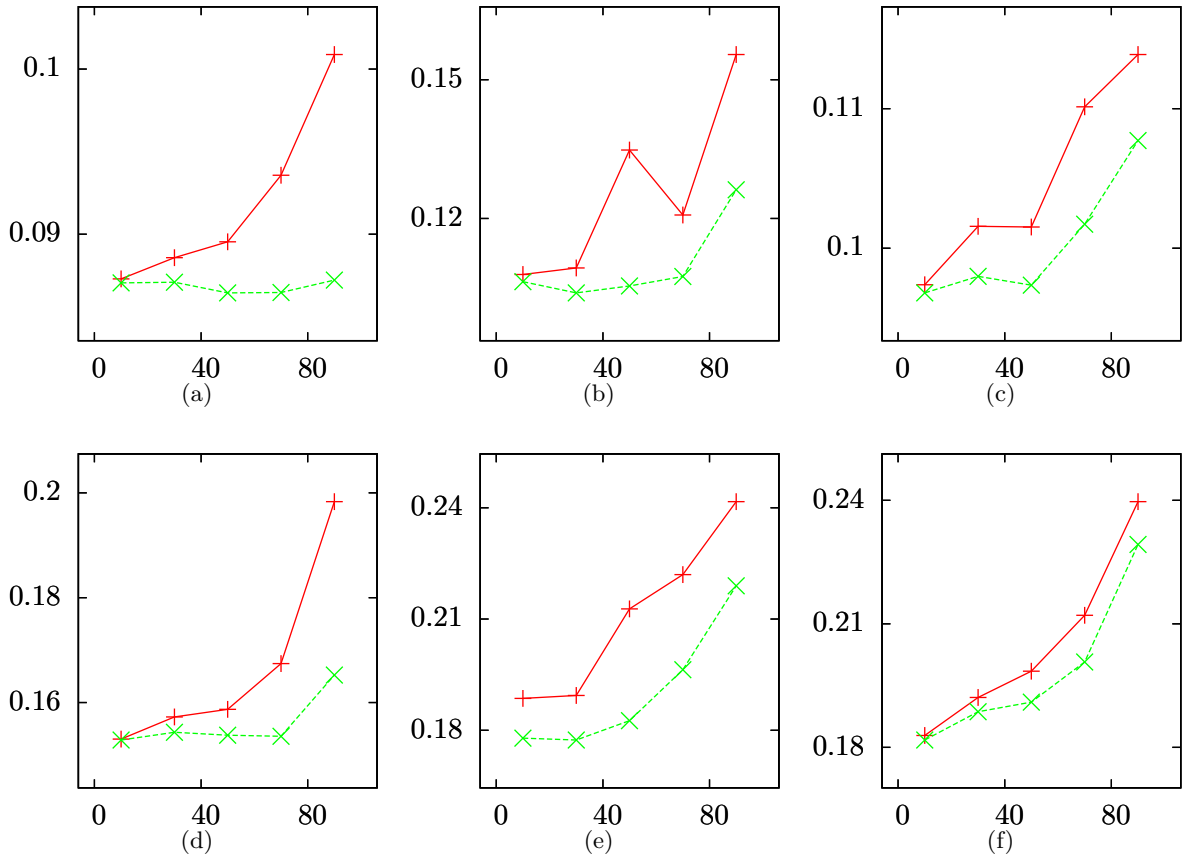


Figure III.12: Comparison of two methods of removing support vectors for the test cases with ids 0-5 from Table III.4. The x axis - the percent of removed data, y axis - a percent difference in MSE for testing data, the line with '+' points - a random removing method, the line with 'x' points - proposed removing method with knowledge about a margin

Table III.4: Performance of φ-SVC for reduced models for real world data, for regression. 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 MSE for testing data, if greater than 0 than a method with knowledge about a margin is better, *s1* – the average number of support vectors for a method without knowledge about a margin, *s2* – the average number of support vectors for a method with knowledge about a margin

(a)

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

(b)

idRef	te12M	s1	s2
0	11.39	12	30
1	34.86	14	48
2	11.11	18	53
3	14.08	25	47
4	16.38	26	52
5	11.41	34	53
6	5.81	9	26
7	34.16	13	48
8	0.2	8	50
9	19.47	17	29
10	18.39	19	53
11	1.33	30	53

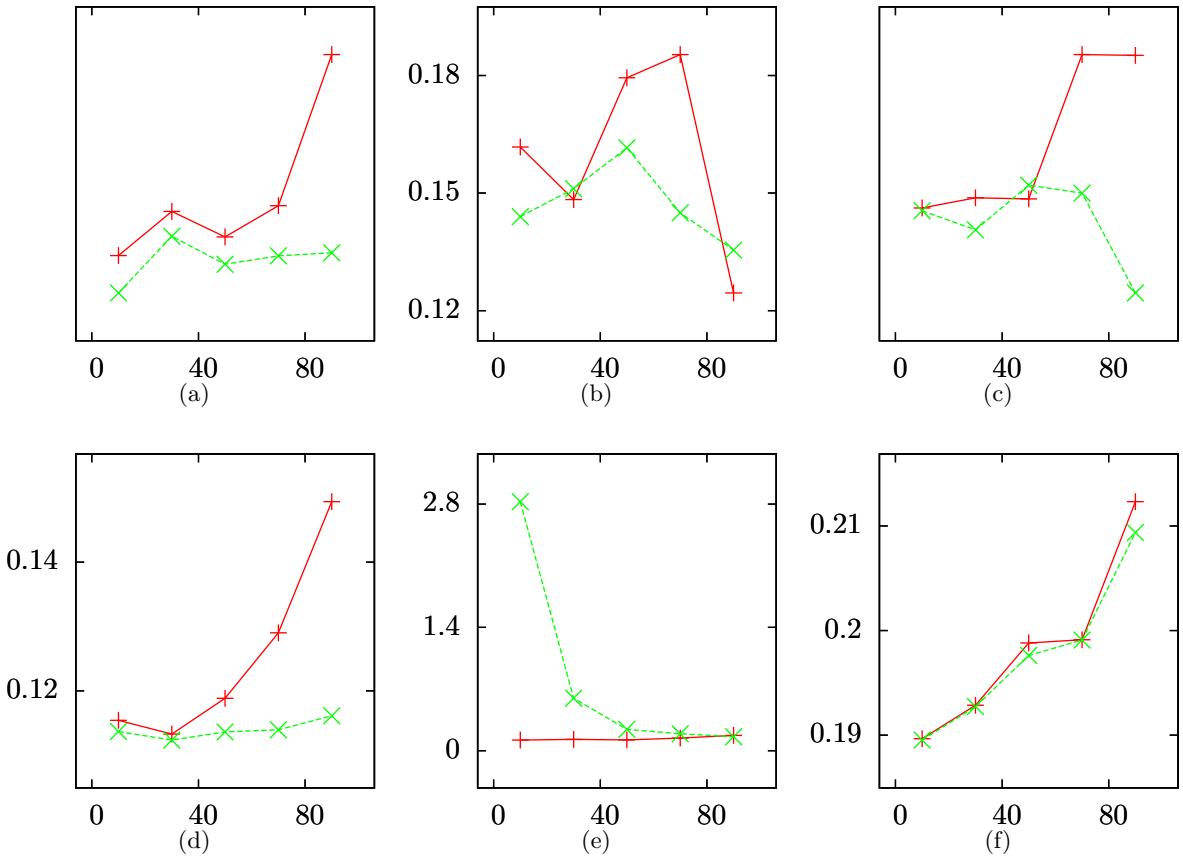


Figure III.13: Comparison of two methods of removing support vectors for the test cases with ids 6-11 from Table III.4, cont. The x axis - the percent of removed data, y axis - a percent difference in MSE for testing data, the line with '+' points - a random removing method, the line with 'x' points - proposed removing method with knowledge about a margin

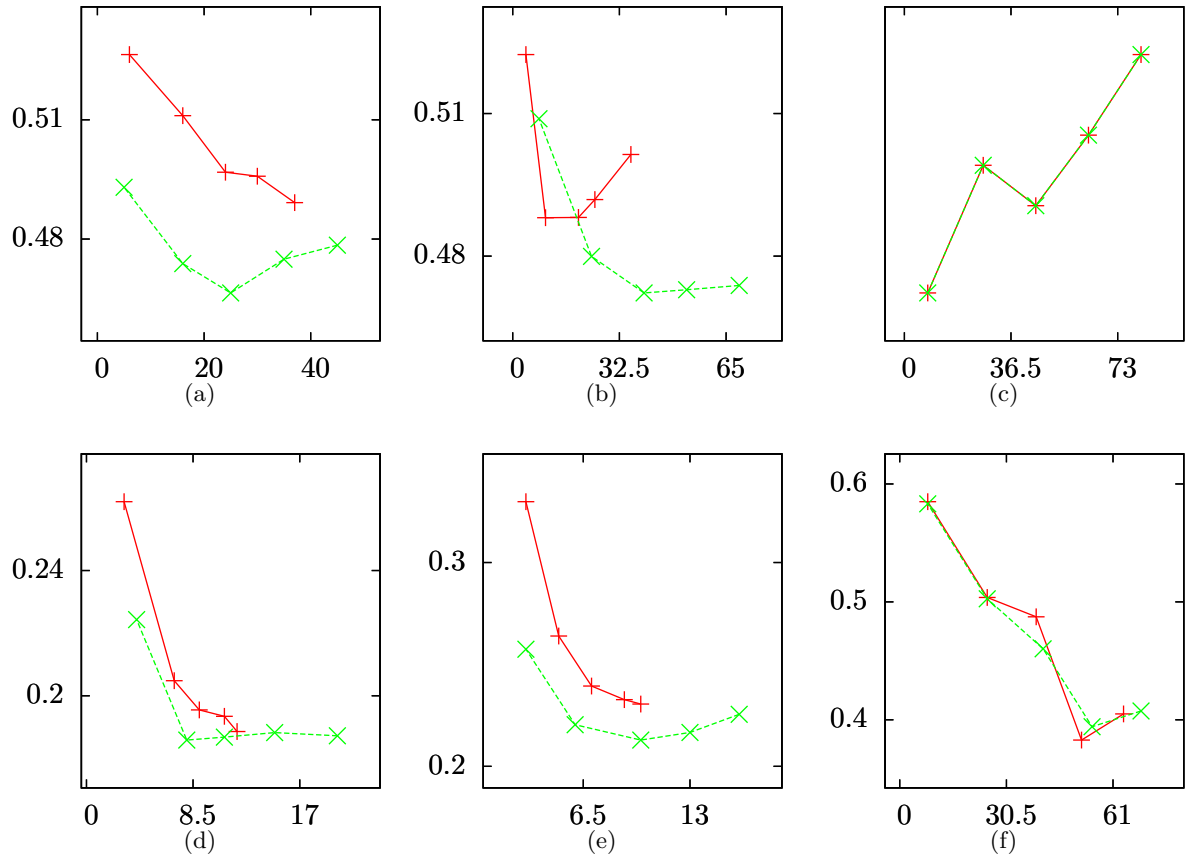


Figure III.14: Comparison of two methods of removing support vectors for the test cases with ids 0-5 from Table III.3. The x axis - the number of support vectors, y axis - a percent difference in misclassified testing examples, the line with '+' points - a random removing method, the line with 'x' points - proposed removing method with knowledge about a margin

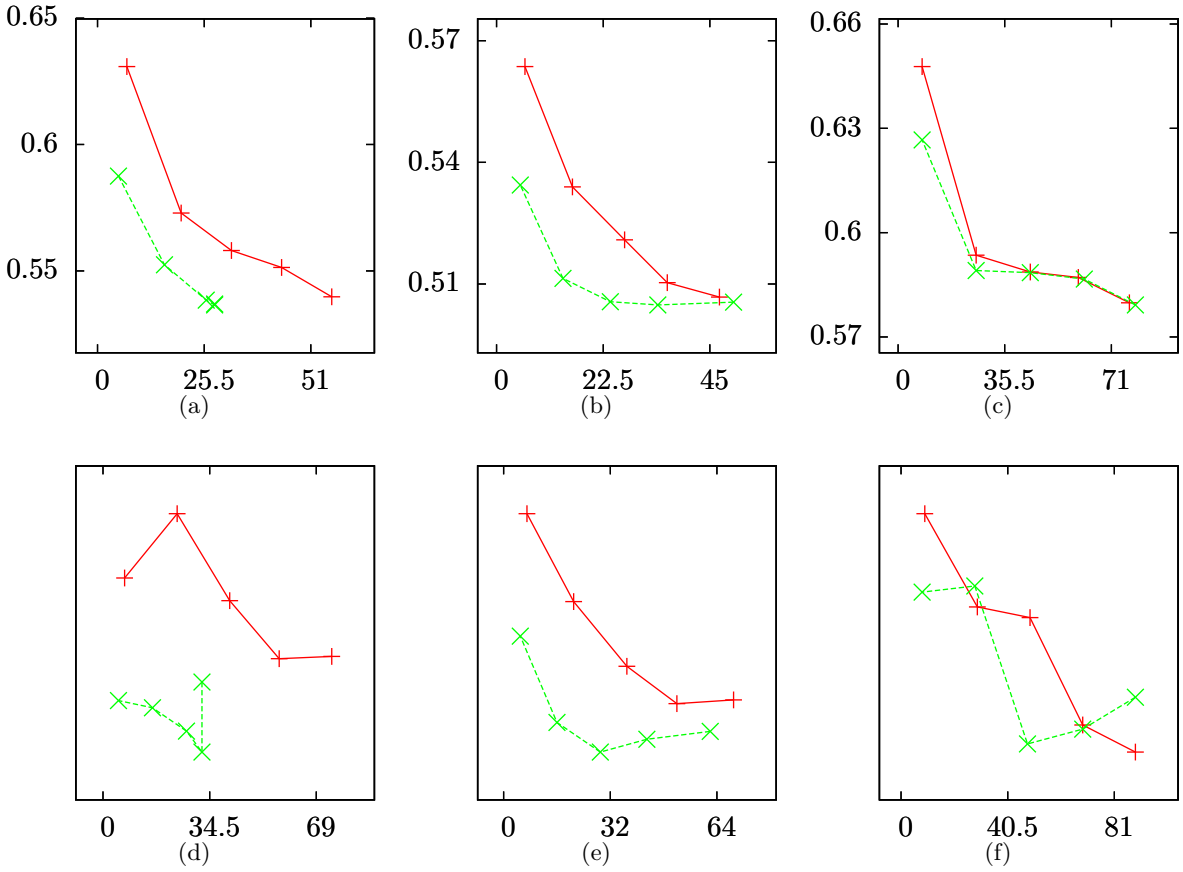


Figure III.15: Comparison of two methods of removing support vectors for the test cases with ids 6-11 from Table III.3, cont. The x axis - the number of support vectors, y axis - a percent difference in misclassified testing examples, the line with '+' points - a random removing method, the line with 'x' points - proposed removing method with knowledge about a margin

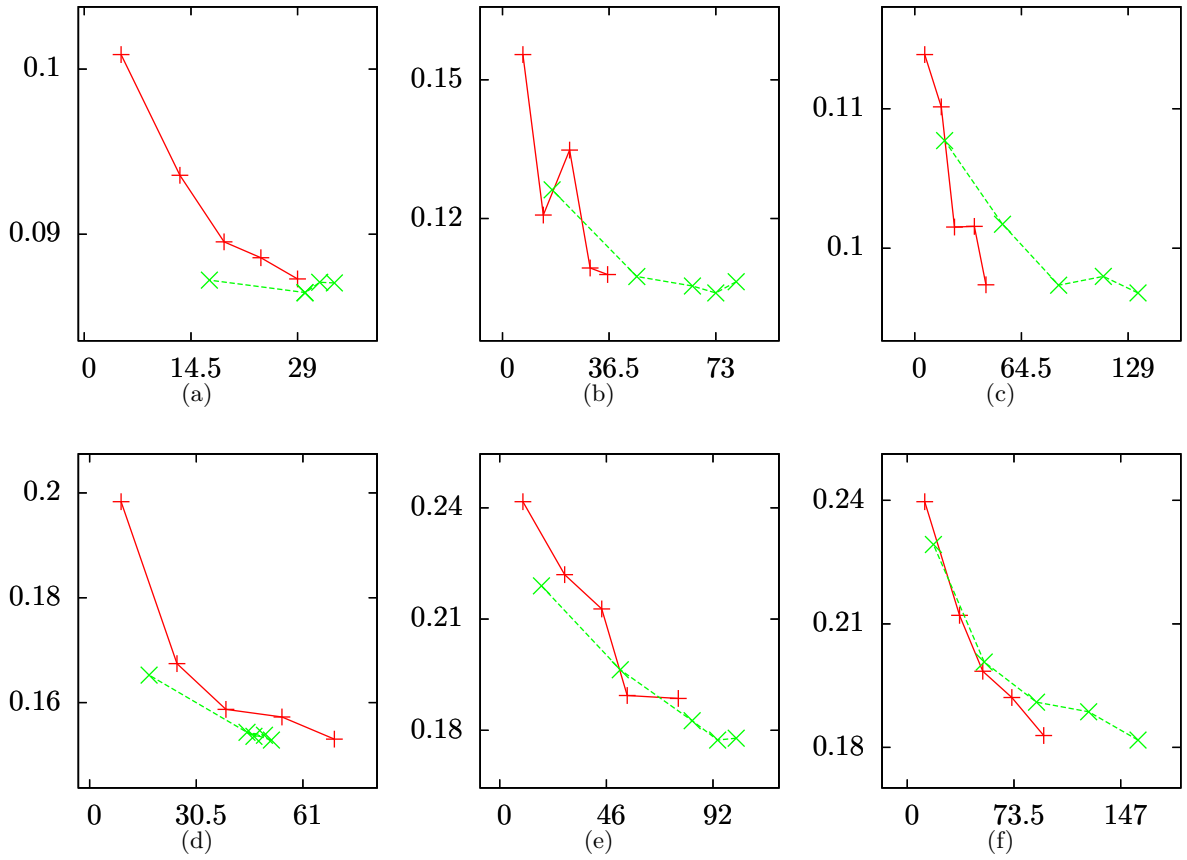


Figure III.16: Comparison of two methods of removing support vectors for the test cases with ids 0-5 from Table III.4. The x axis - the number of support vectors, y axis - a percent difference in MSE for testing data, the line with '+' points - a random removing method, the line with 'x' points - proposed removing method with knowledge about a margin

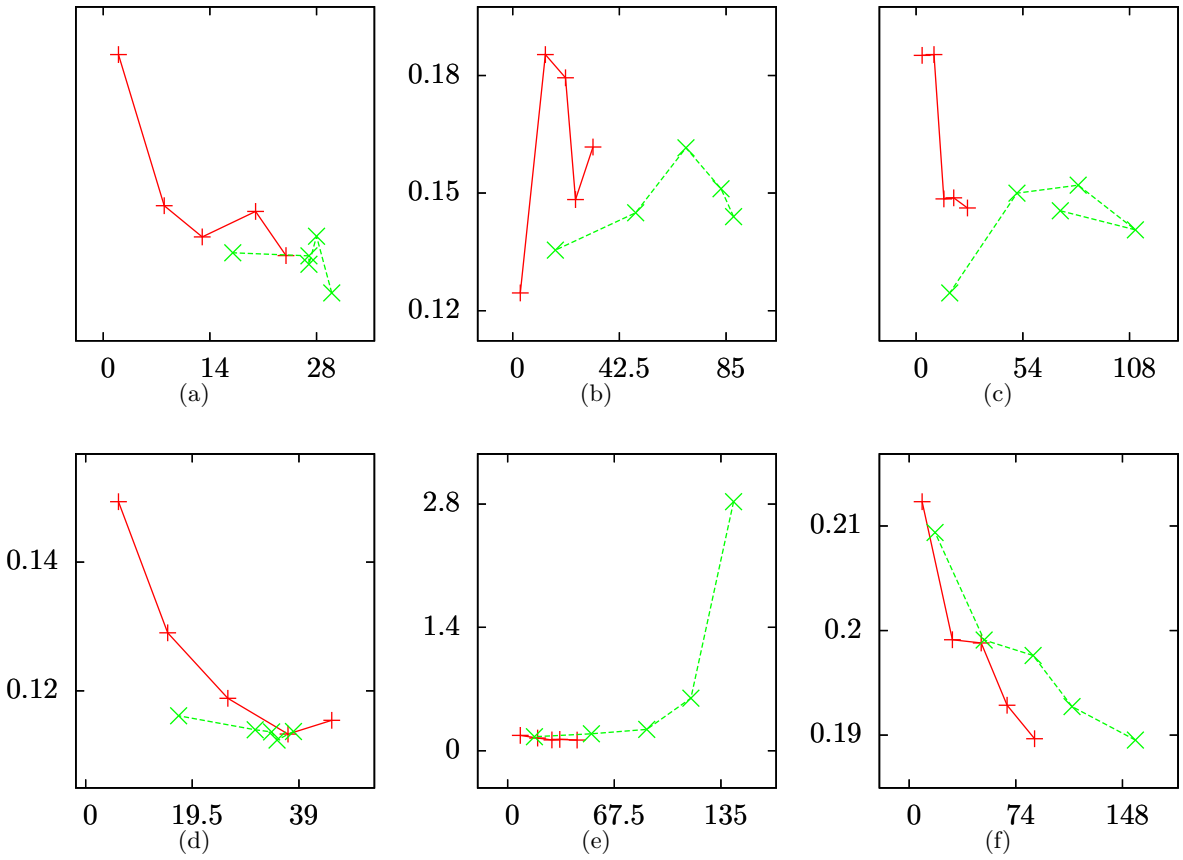


Figure III.17: Comparison of two methods of removing support vectors for the test cases with ids 6-11 from Table III.4, cont. The x axis - the number of support vectors, y axis - a percent difference in MSE for testing data, the line with '+' points - a random removing method, the line with 'x' points - proposed removing method with knowledge about a margin

Chapter IV

Solving SVM by Decomposition

One of categories of methods used for solving OP 3 are decomposition methods (working set methods). In every iteration only a few Lagrange multipliers are optimized. The special case is SMO method proposed in [40], which solves 2-parameter subproblems analytically in every iteration. For subproblems with more than 2 parameters, general quadratic programming solvers are used, [13]. We proposed using SMO for solving subproblems with more than 2 parameters, [31]. The advantage of such solver is a simpler method without external quadratic programming solvers.

One of the parts of working set methods is a strategy for choosing parameters in every iteration. The most popular strategy is based on KKT criterion. We proposed a strategy that in every iteration from a few best alternative pairs of parameters based on KKT criterion chooses a pair which caused the biggest increase of a value of the objective function, [32]. The advantage of this strategy is the decreased number of iterations.

We can use all proposed methods 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 Working Set Methods for φ-SVC

In a working set method applied to φ-SVC, in every iteration the following reduced optimization problem is solved

OP 18.

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

subject to

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

$$0 \le \beta_i \le C_{c_i} \tag{IV.3}$$

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, $c_i \in \{1,\ldots,n\}$, $c_i \neq c_j$ for $i \neq j$, $\vec{\beta}$ is a subproblem variable vector, β_i corresponds to the c_i-th parameter. The α vector is a previous solution of OP 14. It must satisfy the linear constraint (III.11).

After solving OP 18, we replace values of α_{c_i} parameters with β_i values for $i \in \{1,2,\ldots,p\}$. The new solution will always fulfill the linear constraint (III.11).

IV.2 Introduction to SMO for φ-SVC

The SMO is a well established method for solving SVC described in [43, 8]. The SMO is a working set method with a fixed size of a working set equal to 2. So the reduced optimization problem for SMO used for φ-SVC is a special case of OP 18 when $p = 2$. We can find directly the solution for this case, before clipping it is

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

where

$$E_i = \sum_{j=1}^{n} y_j \alpha_j K\left(\vec{x_i}, \vec{x_j}\right) - y_i - y_i \varphi_i \ , \tag{IV.5}$$

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

$$\kappa = K\left(\vec{x_{c_1}}, \vec{x_{c_1}}\right) + K\left(\vec{x_{c_2}}, \vec{x_{c_2}}\right) - 2K\left(\vec{x_{c_1}}, \vec{x_{c_2}}\right) \ . \tag{IV.6}$$

After clipping (derivation in Appendix D.1, Appendix D.2, Appendix D.3)

$$\beta_2 = \begin{cases} V, \ if \ \beta_2^{\text{unc}} > V \\ \beta_2^{\text{unc}}, \ if \ U \leq \beta_2^{\text{unc}} \leq V \\ U, \ if \ \beta_2^{\text{unc}} < U \end{cases} \tag{IV.7}$$

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

$$U = \max\left(0, \alpha_{c_2} - \alpha_{c_1}\right) \ , \tag{IV.8}$$

$$V = \min\left(C_{c_2}, C_{c_1} - \alpha_{c_1} + \alpha_{c_2}\right) \ , \tag{IV.9}$$

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

$$U = \max\left(0, \alpha_{c_1} + \alpha_{c_2} - C_{c_1}\right) \ , \tag{IV.10}$$

$$V = \min\left(C_2, \alpha_{c_1} + \alpha_{c_2}\right) \ . \tag{IV.11}$$

A value of the first variable is

$$\beta_1 = \alpha_{c_1} + y_{c_1} y_{c_2}\left(\alpha_{c_2} - \beta_2\right) \ . \tag{IV.12}$$

IV.2.1 SMO for SVM without the offset

The SMO can be defined for SVC and φ-SVC without the offset (derivation in Appendix D.4 and Appendix D.5 respectively). The difference is that the minimal number of parameters that can be optimized in every step is just one:

$$\beta_1^{\text{unc}} = \alpha_{c_1} - \frac{y_{c_1} E_{c_1}}{K\left(\vec{x_{c_1}}, \vec{x_{c_1}}\right)} \ . \tag{IV.13}$$

Then we have to bound β_1^{unc} to

$$0 \leq \beta_1^{\text{unc}} \leq C_{c_1} \ . \tag{IV.14}$$

IV.3 Introduction To Multivariable Heuristics

The most popular heuristic for a working set method for 2 parameters is based on choosing the parameters most violating the KKT conditions, [18]. The multivariable heuristic based on Zoutendijk's method was proposed in [13]. We proposed a simple strategy for multivariable heuristic based on violation of KKT conditions, [31]. We will present here the proposed heuristic extended to φ-SVC. First, we will show the optimization possibility conditions, that can be derived from KKT conditions.

Theorem IV.3.1. *Optimization is possible for reduced optimization problem OP 18 when there exist two parameters with indices c_d and c_k, where $c_d, c_k \in P$, such as they belong to different groups G_1 and G_2 defined as*

$$\begin{aligned} G_1 &:= \{i \in \{1,2,\ldots,n\} : (y_i = 1 \wedge \alpha_i = 0) \\ &\vee (y_i = -1 \wedge \alpha_i = C_i) \vee (0 < \alpha_i < C_i)\} \\ G_2 &:= \{i \in \{1,2,\ldots,n\} : (y_i = -1 \wedge \alpha_i = 0) \\ &\vee (y_i = 1 \wedge \alpha_i = C_i) \vee (0 < \alpha_i < C_i)\} \quad . \end{aligned} \tag{IV.15}$$

and the following holds:

1. *when c_d is from G_1 group, $\alpha_{c_d} = 0 \vee \alpha_{c_d} = C_{c_d}$, c_k is from G_2, $\alpha_{c_k} = 0 \vee \alpha_{c_k} = C_{c_k}$, then $E_{c_k} > E_{c_d}$*
2. *when $0 < \alpha_{c_d} < C_{c_d}$, c_k is from G_2, $\alpha_{c_k} = 0 \vee \alpha_{c_k} = C_{c_k}$, then $E_{c_k} > E_{c_d}$*
3. *when $0 < \alpha_{c_d} < C_{c_d}$, $0 < \alpha_{c_k} < C_{c_k}$, then $E_{c_d} \neq E_{c_k}$*

The proof is in Appendix D.6. In every step of a working set method, we have to choose p parameters for optimization. First, we will discuss how we choose the first two parameters, and then the rest.

IV.3.1 Choosing Two Parameters to a Working Set

The goal is choose two parameters that violates the KKT conditions the most, so we are looking for parameters that fulfill the optimization possibility conditions from Thm. IV.3.1 with the biggest differences in inequalities from the theorem. Therefore, we choose those two parameters c_d and c_k that maximize $m_{c_d c_k}$ defined as: when $\alpha_{c_k} = 0 \vee \alpha_{c_k} = C_{c_k}$ and α_{c_k} belongs to the G_1 group, then

$$m_{c_d c_k} := E_{c_d} - E_{c_k} \quad , \tag{IV.16}$$

when $\alpha_{c_k} = 0 \vee \alpha_{c_k} = C_{c_k}$ and α_{c_k} belongs to the G_2 group, then

$$m_{c_d c_k} := E_{c_k} - E_{c_d} \quad , \tag{IV.17}$$

when $0 < \alpha_{c_k} < C_{c_k}$, then

$$m_{c_d c_k} := |E_{c_k} - E_{c_d}| \quad . \tag{IV.18}$$

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

IV.3.2 Choosing Remaining Parameters

In [31], we proposed the simple strategy for choosing remaining parameters in which all of them are chosen from either G_1 or G_2 in a way that when chosen from group G_1 the first with minimal E_i are picked, and when chosen from G_2 the first with maximal E_i are picked. The two alternatives are considered and we choose the one for which the sum of values of $m_{c_d c_k}$ for each pair is greater. In the future, we plan to test the alternative strategy that chooses the similar number of parameters from both groups.

IV.4 Subproblem Solver Based on SMO

There were two basic methods for solving SVM subproblems. A new, third method was proposed by me in [31].

1. Solve 2 parameter subproblems analytically (SMO algorithm).
2. Solve more than 2 parameter subproblems with a general quadratic programming solver.

3. Solve more than 2 parameter subproblems with SMO algorithm (SMS).

The third option will be analyzed here. First, we need to introduce a novel SVC modification, which we will call free term support vector classification (bSVC).

IV.4.1 Free Term Support Vector Machines

We will introduce free term support vector machines (bSVM) for φ-SVC, the optimization problem is

OP 19.

$$\min_{\vec{w}_c, b_c, \vec{\xi}_c} \quad f\left(\vec{w}_c, b_c, \vec{\xi}_c\right) = \frac{1}{2}\left\|\vec{w}_c\right\|^2 + C_c \sum_{i=1}^{n} \xi_c^i + Db \tag{IV.19}$$

subject to

$$y_c^i h\left(\vec{x}_i\right) \geq 1 + \varphi_i - \xi_c^i \tag{IV.20}$$

$$\vec{\xi}_c \geq 0 \tag{IV.21}$$

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

$$h\left(\vec{x}_i\right) = \vec{w}_c \cdot \vec{x}_i + b_c \ , \tag{IV.22}$$

$$C_c > 0 \ . \tag{IV.23}$$

We modified (IV.19) by adding the last term. We propose to derive a dual optimization problem (derivation in Appendix D.8) which is

OP 20.

$$\max_{\vec{\alpha}} \quad f\left(\vec{\alpha}\right) = \vec{\alpha} \cdot \left(1 + \vec{\varphi}\right) - \frac{1}{2}\vec{\alpha}^T \mathbf{Q} \vec{\alpha} \tag{IV.24}$$

subject to

$$\vec{\alpha} \cdot \vec{y} = D \tag{IV.25}$$

$$0 \leq \alpha_i \leq C_c \tag{IV.26}$$

where

$$Q_{ij} = y_i y_j K\left(\vec{x}_i, \vec{x}_j\right) \tag{IV.27}$$

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

We can notice that the only difference compared to OP 14 is in (IV.25). We can use SMO for solving bSVM as well. First note that clipping formulas for bSVM are the same as (IV.7), because in derivation (Appendix D.1) we do need to use D parameter. For the similar reason the solution (IV.7) is also the same. The only difference is that the initial solution must fulfill the new condition (IV.25). We can see that the bounds for D are

$$\sum_{i=1}^{n_2} C_{d_i} \leq D \leq \sum_{i=1}^{n_1} C_{c_i} \ , \tag{IV.28}$$

where c_i is the i-th point for which $y_{c_i} = 1$, d_i is the i-th point for which $y_{d_i} = -1$. So for all points with $y_i = 1$, we can set initial values to $\alpha_{c_i} = C_{c_i}$. For remaining points we try to set

$$\alpha_{d_i} = \frac{\sum_{i=1}^{n_1} C_{c_i} - D}{n_2} \ . \tag{IV.29}$$

If any of constraints (IV.26) are violated then we have to use an additional procedure for changing α_{d_i}. For example all violated parts distribute equally to nonviolated alphas, then repeat this step if necessary. If it is not enough we need to lower values of α_{c_i}.

For solving (20), it is enough to use existing code for the SMO method, the only difference is that initial values of α_i parameters fulfills (IV.25).

IV.4.2 Reduced Optimization Problem as bSVC

The reduced optimization problem OP 18 can be reformulated as bSVC with φ_i weights where

$$\varphi_{c_i} = \varphi_{\text{old}}^{c_i} - y_{c_i} \sum_{\substack{j=1 \\ j \notin P}}^{n} y_j \alpha_j K_{c_i j} \tag{IV.30}$$

and

$$D = -\sum_{\substack{i=1 \\ i \notin P}}^{n} y_i \alpha_i \ . \tag{IV.31}$$

Before running bSVC we set initial values of β to the actual values

$$\beta_i = \alpha_{c_i} \tag{IV.32}$$

for $i = \{1, \ldots, p\}$. We can also use current values of E_{c_i} and track only changes. When the solution is found, then we need to update global values of E_{c_i} for $i = \{1, \ldots, p\}$.

IV.4.3 Comparison of SMS with General Subproblem Solvers

In the second method, subproblems are solved by quadratic programming solvers (for example an *interior point method* solver, see [48]). The third method solves subproblems with the SMO algorithm. So it uses a widely known method for decomposing the original problem into 2-parameter subproblems, for more than 2 parameter subproblems.

IV.4.4 Comparison of SMS with SMO

The second and third solvers solve more than 2 parameter subproblems. For some data sets, problems are computed faster with the second and third solvers than with the first one. For example in [13], it was shown that the second solver with working sets of size 20 was faster than the first solver for some data sets. In [31], we showed that the third solver with working sets of size 5 is faster than the first one.

IV.4.5 Experiments

We compared SMS with SMO and found that in deed SMS is faster than SMO, [31]. The SVM optimization with SMS algorithm was tested with the subproblem size of 5. The size was experimentally chosen as the best size.

IV.5 Heuristic of Alternatives

The SMO standard heuristic chooses parameters in every iteration based on KKT conditions. We proposed in [32] an improvement to SMO that we check additionally growth of an objective function (III.10). The HoA for the selected pairs of parameters computes objective function growth and choose the pair maximizing this growth. Both heuristics try to come close 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 that satisfy SVM optimization possibility conditions the best or almost the best. In the set of pairs there is always a pair, that would be chosen by SMO standard heuristic. So the heuristic of alternatives has two strategies incorporated, one to check optimization possibility conditions and the second to check objective function value growth. The pairs that will be chosen for checking might look like this

$$(s_{11}, s_{21}), (s_{12}, s_{21}), (s_{11}, s_{22}), (s_{13}, s_{21}), \ldots \ . \tag{IV.33}$$

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

$$(s_{11}, s_{21}), (s_{12}, s_{21}), (s_{11}, s_{22}), (s_{12}, s_{21}) \quad . \tag{IV.34}$$

Note that we excluded pairs with both parameters the same.

IV.5.1 Comparison of Time Complexity

In the 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 $O(n)$. For all parameters and all iterations the complexity is $O(kn^2)$, where k is the number of iterations.

In HoA, objective function value growth of OP 12 needs to be computed in every iteration for every alternative pair. From the (IV.1) we get the formula for objective function value growth

$$\begin{aligned}\Delta f_2\left(\vec{\beta}\right) = \sum_{i=1}^{2} \Delta\beta_i - \sum_{j=1}^{2} y_{c_j} \Delta\beta_j \sum_{\substack{i=1 \\ i \notin C}}^{n} y_i \alpha_i K_{c_j i} - \frac{1}{2} \sum_{i=1}^{2} \left(\beta_{i\text{new}}^2 - \beta_{i\text{old}}^2\right) K_{c_i c_i} \\ -y_{c_1 c_2} \left(\beta_{1\text{new}}\beta_{2\text{new}} - \beta_{1\text{old}}\beta_{2\text{old}}\right) K_{c_1 c_2} \quad .\end{aligned} \tag{IV.35}$$

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 $O(kmn)$, where m is the number of alternative pairs in every iteration. Overall complexity of heuristic of alternatives is $O(kn^2 + kmn)$. The complexity of HoA differs from SMO standard heuristic with the kmn part, which has limited influence on overall time when the number of parameters is big enough.

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

IV.5.2 Experiments

The 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 12 out of 16 tests. A strong improvement in the number of iterations leads to the improvement in training time. We can notice the improvement in training time in 6 out of 12 tests.

IV.6 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.

Table IV.1: The HoA performance for real world data sets. Column descriptions: *id* – an id of a test, *dn* – a name of a data set, *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*)

(a)

id	dn	ker
0	a1aAll	denseLinear 0.0
1	a1aAll	denseRBF 0.00813
2	breast-cancer	denseRBF 0.1
3	diabetes	denseRBF 0.125
4	djia	denseRBF 0.08333
5	abalone	denseLinear 0.0
6	abalone	denseRBF 0.125
7	abalone	denseRBF 0.5
8	cadata	denseRBF 0.125
9	djia	denseLinear 0.0
10	djia	denseRBF 0.1
11	djia	denseRBF 0.5
12	housing	denseLinear 0.0
13	housing	densePolynomial 5.0
14	housing	denseRBF 0.077
15	housing	denseRBF 0.5

(b)

idRef	m1it	m2it	m1ctt	m2ctt
0	10291	8004	9.93725	9.6975
1	50775	50775	13582.618	13677.475
2	905	969	0.8405	1.097
3	1045	1066	0.669	0.89
4	1736	1586	1.883	2.035
5	6033	5565	18.487	19.681
6	15382	15879	99.377	122.48
7	10399	10314	67.824	76.006
8	100126	99571	3291.658	3311.83
9	1833	1251	6.202	4.506
10	1154	1073	2.283	2.339
11	2542	2307	4.27	4.243
12	4195	2857	4.555	3.438
13	414420	92011	108.594	31.754
14	318	313	0.901	0.939
15	1209	1048	2.393	2.338

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], the authors found that improving quality of the volume prediction leads to better execution performance, however they had found contradicting results in [10].

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 part of the strategy by using a general purpose machine learning method such as SVR. The work was published in [38].

In [1], the authors predict a volume function by decomposing volume into two parts and using the average method and autoregressive models for prediction. In [2], the 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 ways 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 knowledge about the margin of an example, recently proposed for SVC, [33], for δ-SVR, [35] and for ε-SVR, [37]. It was used for manipulating a decision curve for classification problems, and manipulating a regression function for regression ones.

A test scenario that 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 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 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 pre-

dicting 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 statements 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, $v(i)$ is a volume of the i-th trade, v is a market volume in T, $p(i)$ is a price of the i-th trade. We have

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

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

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

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

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

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

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

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

In the presented strategy, we divide T to some time slices. Below, we list some statements 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 $v(T_i)$, $VWAP$ for all trades in the i-th part is noted as $VWAP(T_i)$, aggregated volume of the order in the i-th part is noted as $v_0(T_i)$. Then a market volume in T is

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

Market VWAP in T is

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

A volume of the order in T is

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

and order VWAP in T is

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

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

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

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

We can reformulate (V.10) by substituting

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

and we get

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

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

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

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 that achieves the ratio equal to 1 by predicting volume participation. Second, we will present an extension of this strategy that 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 that 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

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

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

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

where

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

The r is called *volume participation*, Fig. V.1. We can easily check that for this strategy (V.11) is satisfied. 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 $r(T_i)$ (V.16) for every time slice and try to trade at $VWAP(T_i)$ 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 $v(T_i)$ is more richer prediction than just only ratios $r(T_i)$. For the same ratios, we could have multiple possible values of v. In other words, when we have only ratios $r(T_i)$ it is impossible to conclude about a value of v. There exist multiple different volume shapes with the same ratios $r(T_i)$.

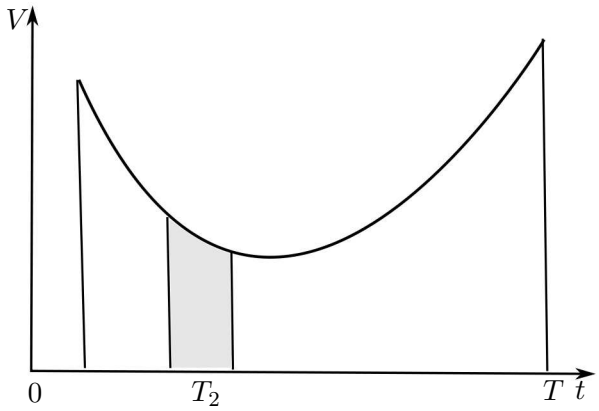


Figure V.1: The idea of volume participation. Volume participation for T_2 is interpreted as a ratio of gray area to the whole area below volume from 0 to T

Note that for different values of a free term a of a volume function we can get different values 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 satisfied. Let's derive the ratio

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

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

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 execution errors in 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 (derivation in E.1)

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 worse performance of prediction of volume participation for TWAP, therefore larger value of ε_1 compared to the VWAP strategy, so we expect larger variance of ε for the 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 the 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 $r(T_{i-1})$ ($r(T_{i-1})$ is predicted as in the second solver) 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 satisfied:

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

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

For the TWAP predictor, they are satisfied 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 satisfy (V.27), and when some values are below zero, we adjust them to zero. We repeat these two steps until both constraints are satisfied. For the last predictor, we propose the direct incorporation of (V.27) by using φ-SVC and modified kernels, III.9. Instead of incorporating directly (V.28) to the optimization problem, we propose soft incorporation proposed for SVC, III.10.

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 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 changes 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 worse volume participation prediction performance, and decrease value of p for the rest. For this purpose, we use SVM with knowledge about the margin of an example introduced for SVC in [33, 35], for ε-SVR in [37] and for δ-SVR in [36]. The technique was used for manipulating classification boundaries, [33], and regression functions, [36]. It has 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 $\varphi_i = r$ 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 example 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 execution error for SVM and null hypotheses. We compare prediction performance of SVM with the following null hypotheses: prediction based on a 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 knowledge about the margin of an example.

For solving ε-SVR and SVC for particular values of parameters we use LibSVM, [3], ported to Java. Data that 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 include 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 Error 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

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%

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 – φ_i 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

difference based on t-test for ε-SVR, with significant difference for δ-SVR. Comparing additional measure of variance of execution error, we achieve slightly better results for ε-SVR than for the first and the third hypotheses, and similar results to the second hypothesis. For δ-SVR, we achieved much larger 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 omits the topic of price prediction. Therefore, we propose the following procedure for generating prior knowledge 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 φ_i weights, 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 worse after adjusting the function, but we can see significant improvement in execution error 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 for predicting volume participation function. We can improve costs of order execution by using prediction of stock prices with SVM.

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

Chapter VI

Summary

The main contributions to the thesis are

1. proposed a novel regression method, called δ-SVR, which transforms regression problems into binary classification problems, analysis of equivalency of Bayes solutions before and after transformation, analysis of sparsity and generalization performance of the transformed problems, and experiments for SVC classifier,
2. analysis of knowledge about margin of an example, applications to reducing complexity of models and incorporating additional constraints to the optimization problem,
3. proposed a speed improvement for the SMO heuristic,
4. application of SVM to reducing cost of executing orders on exchanges.

Appendix A

Introduction to Optimization Theory

This introduction is based partly on [46]. 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*. We will represent these problems by

$$\text{maximize } f(x) \tag{A.1}$$

subject to

$$\vec{x} \in D \tag{A.2}$$

Alternatively

$$\max \{f(\vec{x}) \,|\vec{x} \in D\} \tag{A.3}$$

Such problems are called *maximization problems*. A *solution* to the problem (A.3) is a point $\vec{x} \in D$ such as

$$f(\vec{x}) \geq f(\vec{y}) \tag{A.4}$$

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

$$D = U \cap \{\vec{x} \in R^n | g(\vec{x}) = 0, h(\vec{x}) \geq 0\} \tag{A.5}$$

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

$$D = U \cap \{\vec{x} | g(\vec{x}) = 0\} \tag{A.6}$$

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

$$D = U \cap \{\vec{x} | h(\vec{x}) \geq 0\} \tag{A.7}$$

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*

$$D = U \cap \{\vec{x} | g_i(\vec{x}) = 0, i = 1, \ldots, k\} \quad , \tag{A.8}$$

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

$$Df(x^*) + \sum_{j=1}^{k} \lambda_i^* Dg_i(x^*) = 0 \tag{A.9}$$

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:

$$L\left(\vec{x}, \vec{\lambda}\right) = f(\vec{x}) + \sum_{i=1}^{k} \lambda_i g_i(\vec{x}) \ . \tag{A.10}$$

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*

$$Z(x^*) = \{z \in R^n | Dg(x^*) z = 0\} \tag{A.11}$$

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

$$D^2L(x^*, \lambda^*) = D^2f(x^*) + \sum_{i=1}^{k} \lambda_i^* D^2 g_i(x^*) \tag{A.12}$$

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*

$$D = U \cap \{x \in R^n | h_i(x) \geq 0, i = 1, \ldots, n\} \quad , \tag{A.13}$$

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

$$c_i = \begin{cases} g_i, \ if \ i \in \{1, \ldots, k\} \\ h_{i-k}, \ if \ i \in \{k+1, \ldots, k+n\} \end{cases} \tag{A.14}$$

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*

$$D = U \cap \{\vec{x} \in R^n | c_i(x) = 0, i = 1, \ldots, k, c_j(x) \geq 0, j = k+1, \ldots, k+n\} \tag{A.15}$$

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*

$$h_i(\hat{x}) > 0 \tag{A.16}$$

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

$$D = \{x \in U | h_i(x) \geq 0, i = 1, \ldots, n\} \tag{A.17}$$

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*

$$h\left(\vec{\lambda}\right) = \min_{\vec{x}} L\left(\vec{x}, \vec{\lambda}\right) \tag{A.18}$$

Defining the set

$$D = \left\{\vec{\lambda} | h\left(\vec{\lambda}\right) \exists \ and \ \vec{\lambda} \geq 0\right\} \tag{A.19}$$

allows for the formulation of the dual problem

$$\underset{\vec{\lambda} \in D}{\text{maximize}} \, h\left(\vec{\lambda}\right) \tag{A.20}$$

that is equivalent to

$$\max_{\vec{\lambda} \in D} \left(\min_{\vec{x}} L\left(\vec{x}, \vec{\lambda}\right) \right) \quad . \tag{A.21}$$

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

Regression Based on Binary Classification

B.1 The idea of a Set of Indicator Functions

The classification problem could be defined in terms of minimizing the risk function, [50]

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

where L is *a loss function* defined as

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

The regression problem could be defined as minimizing the risk function

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

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

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

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

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

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

where

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

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

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

where $\alpha \in A$ and

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

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

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

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

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

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.

B.2 A Proof of Thm. II.2.1

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

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

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

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

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

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

We can rewrite it as

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

Both classes have the same number of examples so

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

so

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

Because both distributions are symmetrical and unimodal the above holds for

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

and because of symmetrical translation we get

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

□

B.3 A Proof of Thm. II.2.2

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

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

where

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

for

$$x = m \ . \tag{B.22}$$

So

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

where

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

When $\delta = m$ then

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

if for some value $\delta > m$

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

then from intermediate value theorem there exists the δ.

□

B.4 Solution for (II.64)

The following holds

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

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

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

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

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

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

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

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

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

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

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

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

Appendix C

Knowledge About a Margin

C.1 Derivation of the Dual Form of OP 12

OP 21.

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

where

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

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

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

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

$$\frac{\partial t\left(\vec{w},b,\vec{\alpha},\vec{\xi},\vec{r}\right)}{\partial b} = \sum_{i=1}^{n} \alpha_i y_{\mathrm{c}}^i = 0 \ . \tag{C.3}$$

A partial derivative with respect to ξ_i is

$$\frac{\partial t\left(\vec{w},b,\vec{\alpha},\vec{\xi},\vec{r}\right)}{\partial \xi_i} = C_i - r_i - \alpha_i = 0 \ . \tag{C.4}$$

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

$$\begin{aligned} d(\vec{\alpha}, \vec{r}) &= \tfrac{1}{2} \sum_{i=1}^{m} \left(\sum_{j=1}^{n} \alpha_j y_c^j x_{ji} \right) \left(\sum_{k=1}^{n} \alpha_k y_c^k x_{ki} \right) \\ &- \sum_{i=1}^{n} \alpha_i y_c^i \left(\sum_{j=1}^{m} w_j x_{ij} + b \right) + \sum_{i=1}^{n} \alpha_i (1 + \varphi_i) + C_i \sum_{i=1}^{n} \xi_i \\ &- \sum_{i=1}^{n} \alpha_i \xi_i - \sum_{i=1}^{n} r_i \xi_i \end{aligned} \tag{C.5}$$

$$\begin{aligned} d(\vec{\alpha}, \vec{r}) &= \tfrac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{n} \alpha_k \alpha_j y_c^k y_c^j x_{ki} x_{ji} - \sum_{i=1}^{n} \alpha_i y_c^i \sum_{j=1}^{m} w_j x_{ij} \\ &- b \sum_{i=1}^{n} \alpha_i y_c^i + \sum_{i=1}^{n} \alpha_i (1 + \varphi_i) \end{aligned} \tag{C.6}$$

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

$$\begin{aligned} d(\vec{\alpha}, \vec{r}) &= \tfrac{1}{2} \sum_{j=1}^{n} \sum_{k=1}^{n} \alpha_k \alpha_j y_c^k y_c^j \sum_{i=1}^{m} x_{ji} x_{ki} \\ &- \sum_{i=1}^{n} \sum_{k=1}^{n} \alpha_k \alpha_i y_c^k y_c^i \sum_{j=1}^{m} x_{ij} x_{kj} + \sum_{i=1}^{n} \alpha_i (1 + \varphi_i) \end{aligned} \tag{C.8}$$

$$d(\vec{\alpha}, \vec{r}) = -\frac{1}{2} \sum_{i=1}^{n} \sum_{k=1}^{n} \alpha_k \alpha_i y_c^k y_c^i \sum_{j=1}^{m} x_{ij} x_{kj} + \sum_{i=1}^{n} \alpha_i (1 + \varphi_i) \ . \tag{C.9}$$

The dual form is

OP 22.

$$\max_{\vec{\alpha}, \vec{r}} \quad d(\vec{\alpha}, \vec{r}) = \sum_{i=1}^{n} \alpha_i (1 + \varphi_i) - \frac{1}{2} \sum_{i=1}^{n} \sum_{k=1}^{n} \alpha_k \alpha_i y_c^k y_c^i \sum_{j=1}^{m} x_{ij} x_{kj} \tag{C.10}$$

subject to

$$\sum_{i=1}^{n} \alpha_i y_c^i = 0 \tag{C.11}$$

$$C_i = r_i + \alpha_i \tag{C.12}$$

$$\alpha_i \geq 0 \tag{C.13}$$

$$r_i \geq 0 \tag{C.14}$$

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

C.2 Derivation of ε-SVR Reformulation as φ-SVC

We present derivation of ε-SVR in the form OP 8 as φ-SVC

OP 23.

$$\min_{\vec{w_r}, b_r, \vec{\xi_r}, \vec{\xi_r^*}} \quad f\left(\vec{w_r}, b_r, \vec{\xi_r}, \vec{\xi_r^*}\right) = \frac{1}{2} \|\vec{w_r}\|^2 + C_r \sum_{i=1}^{n} \left(\xi_r^i + \xi_r^{*i}\right) \tag{C.15}$$

subject to

$$y_r^i - g(\vec{x_i}) \leq \varepsilon_u^i + \xi_r^i \tag{C.16}$$

$$g(\vec{x_i}) - y_r^i \leq \varepsilon_d^i + \xi_r^{i*} \tag{C.17}$$

$$\vec{\xi_r} \geq 0 \tag{C.18}$$

$$\vec{\xi_r^*} \geq 0 \tag{C.19}$$

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

$$g\left(\vec{x_i}\right) = \vec{w_r} \cdot \vec{x_i} + b_r \ . \tag{C.20}$$

OP 24.

$$\min_{\vec{w_r}, b_r, \vec{\xi_r}, \vec{\xi_r^*}} \quad f\left(\vec{w_r}, b_r, \vec{\xi_r}, \vec{\xi_r^*}\right) = \frac{1}{2}\left\|\vec{w_r}\right\|^2 + C_r \sum_{i=1}^{2n} \xi_r^i \tag{C.21}$$

subject to

$$g\left(\vec{x_i}\right) \geq y_r^i - \varepsilon_u^i - \xi_r^i \tag{C.22}$$

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

$$g\left(\vec{x_i}\right) \leq \varepsilon_d^i + y_r^i + \xi_r^i \tag{C.23}$$

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

$$\vec{\xi_r} \geq 0 \ , \tag{C.24}$$

where

$$g\left(\vec{x_i}\right) = \vec{w_r} \cdot \vec{x_i} + b_r \ . \tag{C.25}$$

OP 25.

$$\min_{\vec{w_r}, b_r, \vec{\xi_r}, \vec{\xi_r^*}} \quad f\left(\vec{w_r}, b_r, \vec{\xi_r}, \vec{\xi_r^*}\right) = \frac{1}{2}\left\|\vec{w_r}\right\|^2 + C_r \sum_{i=1}^{2n} \xi_r^i \tag{C.26}$$

subject to

$$y_c^i g\left(\vec{x_i}\right) \geq y_r^i - \varepsilon_u^i - \xi_r^i \ \left(y_c^i = 1\right) \tag{C.27}$$

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

$$y_c^i g\left(\vec{x_i}\right) \geq -\varepsilon_d^i - y_r^i - \xi_r^i \ \left(y_c^i = -1\right) \tag{C.28}$$

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

$$\vec{\xi_r} \geq 0 \tag{C.29}$$

where

$$g\left(\vec{x_i}\right) = \vec{w_r} \cdot \vec{x_i} + b_r \ . \tag{C.30}$$

OP 26.

$$\min_{\vec{w_r}, b_r, \vec{\xi_r}, \vec{\xi_r^*}} \quad f\left(\vec{w_r}, b_r, \vec{\xi_r}, \vec{\xi_r^*}\right) = \frac{1}{2}\left\|\vec{w_r}\right\|^2 + C_r \sum_{i=1}^{2n} \xi_r^i \tag{C.31}$$

subject to

$$y_c^i g\left(\vec{x_i}\right) \geq y_c^i y_r^i - \varepsilon_i - \xi_r^i \tag{C.32}$$

$$\vec{\xi_r} \geq 0 \tag{C.33}$$

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

$$\varepsilon_i = \varepsilon_u^i \text{ for } i \in \{1, \ldots, n\} \ , \tag{C.34}$$

$$\varepsilon_i = \varepsilon_d^i \text{ for } i \in \{n+1, \ldots, 2n\} \ , \tag{C.35}$$

$$g\left(\vec{x_i}\right) = \vec{w_r} \cdot \vec{x_i} + b_r \ . \tag{C.36}$$

OP 27.

$$\min_{\vec{w_r}, b_r, \vec{\xi_r}, \vec{\xi_r^*}} \quad f\left(\vec{w_r}, b_r, \vec{\xi_r}, \vec{\xi_r^*}\right) = \frac{1}{2}\left\|\vec{w_r}\right\|^2 + C_r \sum_{i=1}^{2n} \xi_r^i \tag{C.37}$$

subject to

$$y_c^i g\left(\vec{x_i}\right) \geq 1 + y_c^i y_r^i - \varepsilon_i - \xi_r^i - 1 \tag{C.38}$$

$$\vec{\xi_r} \geq 0 \tag{C.39}$$

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

$$\varepsilon_i = \varepsilon_u^i \text{ for } i \in \{1, \ldots, n\} \ , \tag{C.40}$$

$$\varepsilon_i = \varepsilon_d^i \text{ for } i \in \{n+1, \ldots, 2n\} \ , \tag{C.41}$$

$$g\left(\vec{x_i}\right) = \vec{w_r} \cdot \vec{x_i} + b_r \ . \tag{C.42}$$

And we have

$$w_i = \sum_{j=1}^{2n} y_c^j \alpha_j x_{ij} = \sum_{j=1}^{n} \alpha_j x_{ij} - \sum_{j=n+1}^{2n} \alpha_j^* x_{ij} = \sum_{j=1}^{n} \left(\alpha_j - \alpha_j^*\right) x_{ij} \ . \tag{C.43}$$

C.3 Derivation of the Dual Form of OP 17

OP 28.

$$\max_{\vec{\alpha},\vec{r}} \quad d\left(\vec{\alpha},\vec{r}\right) \tag{C.44}$$

where

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

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

$$\frac{\partial h\left(\vec{w},\vec{\alpha},\vec{\xi},\vec{r}\right)}{\partial w_i} = w_i - \sum_{j=1}^{n} \alpha_j y_j x_{ji} = 0 \tag{C.45}$$

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

$$\frac{\partial h\left(\vec{w},\vec{\alpha},\vec{\xi},\vec{r}\right)}{\partial \xi_i} = C_i - r_i - \alpha_i = 0 \ . \tag{C.46}$$

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

$$\begin{aligned} d\left(\vec{\alpha},\vec{r}\right) &= \tfrac{1}{2} \sum_{i=1}^{m} \left(\sum_{j=1}^{n} \alpha_j y_j x_{ji}\right)\left(\sum_{k=1}^{n} \alpha_k y_k x_{ki}\right) \\ &- \sum_{i=1}^{n} \alpha_i y_i \left(\sum_{j=1}^{m} w_j x_{ij}\right) + \sum_{i=1}^{n} \alpha_i \left(1 + \varphi_i\right) + C_i \sum_{i=1}^{n} \xi_i \\ &- \sum_{i=1}^{n} \alpha_i \xi_i - \sum_{i=1}^{n} r_i \xi_i \end{aligned} \tag{C.47}$$

$$\begin{aligned} d\left(\vec{\alpha},\vec{r}\right) &= \tfrac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{n} \alpha_k \alpha_j y_k y_j x_{ki} x_{ji} - \sum_{i=1}^{n} \alpha_i y_i \sum_{j=1}^{m} w_j x_{ij} \\ &+ \sum_{i=1}^{n} \alpha_i \left(1 + \varphi_i\right) \end{aligned} \tag{C.48}$$

$$\begin{aligned} d\left(\vec{\alpha},\vec{r}\right) &= \tfrac{1}{2} \sum_{j=1}^{n} \sum_{k=1}^{n} \alpha_k \alpha_j y_k y_j \sum_{i=1}^{m} x_{ji} x_{ki} \\ &- \sum_{i=1}^{n} \alpha_i y_i \sum_{j=1}^{m} x_{ij} \sum_{k=1}^{n} \alpha_k y_k x_{kj} + \sum_{i=1}^{n} \alpha_i \left(1 + \varphi_i\right) \end{aligned} \tag{C.49}$$

$$d\left(\vec{\alpha},\vec{r}\right) = \frac{1}{2}\sum_{j=1}^{n}\sum_{k=1}^{n}\alpha_k\alpha_j y_k y_j \sum_{i=1}^{m} x_{ji}x_{ki} - \sum_{i=1}^{n}\sum_{k=1}^{n}\alpha_k\alpha_i y_k y_i \sum_{j=1}^{m} x_{ij}x_{kj} + \sum_{i=1}^{n}\alpha_i\left(1+\varphi_i\right) \tag{C.50}$$

$$d\left(\vec{\alpha},\vec{r}\right) = -\frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{n}\alpha_k\alpha_i y_k y_i \sum_{j=1}^{m} x_{ij}x_{kj} + \sum_{i=1}^{n}\alpha_i\left(1+\varphi_i\right) \ . \tag{C.51}$$

The dual form is

OP 29.

$$\max_{\vec{\alpha},\vec{r}} \quad d\left(\vec{\alpha},\vec{r}\right) = \sum_{i=1}^{n}\alpha_i\left(1+\varphi_i\right) - \frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{n}\alpha_k\alpha_i y_k y_i \sum_{j=1}^{m} x_{ij}x_{kj} \tag{C.52}$$

subject to

$$C_i = r_i + \alpha_i \tag{C.53}$$

$$\alpha_i \geq 0 \tag{C.54}$$

$$r_i \geq 0 \tag{C.55}$$

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

C.4 Incorporation of the Linear Dependency to φ-SVC

We will incorporate (III.44) to OP 12. After reformulation

$$\sum_{i=1}^{s} s_i \vec{w}_c \cdot \vec{d}_i + b\sum_{i=1}^{s} s_i = e \tag{C.56}$$

$$b = \frac{1}{\sum_{i=1}^{s} s_i}\left(e - \sum_{i=1}^{s} s_i \vec{w}_c \cdot \vec{d}_i\right) \ . \tag{C.57}$$

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

$$h\left(\vec{x}\right) = \vec{w}_c \cdot \vec{x} + \frac{1}{\sum_{i=1}^{s} s_i}\left(e - \sum_{i=1}^{s} s_i \vec{w}_c \cdot \vec{d}_i\right) \tag{C.58}$$

after reformulation

$$h\left(\vec{x}\right) = \vec{w}_c \cdot \vec{x} - \frac{1}{\sum_{i=1}^{s} s_i}\vec{w}_c \cdot \sum_{i=1}^{s} s_i \vec{d}_i + \frac{e}{\sum_{i=1}^{s} s_i} \ . \tag{C.59}$$

After substituting above to OP 12, we get the φ-SVC problem without the offset with the new kernel in the form of transformation of the input vectors

$$\vec{x} \to \vec{x} - \frac{1}{\sum_{i=1}^{s} s_i}\sum_{i=1}^{s} s_i \vec{d}_i \tag{C.60}$$

$$K\left(\vec{x},\vec{y}\right) = \left(\vec{x} - \frac{1}{\sum_{i=1}^{s} s_i}\sum_{i=1}^{s} s_i \vec{d}_i\right)\left(\vec{y} - \frac{1}{\sum_{i=1}^{s} s_i}\sum_{i=1}^{s} s_i \vec{d}_i\right) \tag{C.61}$$

and φ_i weights set as

$$\varphi_i = \varphi_{old} - y_i \frac{e}{\sum_{i=1}^{s} s_i} \ . \tag{C.62}$$

We get nonlinear solutions by using the following way of further kernelization

$$K\left(\vec{x},\vec{y}\right) = K_o\left(\vec{x},\vec{y}\right) - \frac{1}{\sum_{i=1}^{s} s_i}\sum_{i=1}^{s} s_i K_o\left(\vec{x},\vec{d}_i\right) - \frac{1}{\sum_{i=1}^{s} s_i}\sum_{i=1}^{s} s_i K_o\left(\vec{y},\vec{d}_i\right) \tag{C.63}$$

$$+\frac{1}{\left(\sum_{i=1}^{s} s_i\right)^2}\sum_{i=1}^{s}\sum_{j=1}^{s} s_i s_j K_o\left(\vec{d_i},\vec{d_j}\right) \tag{C.64}$$

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

$$\vec{w_c}=\sum_{j=1}^{n}\alpha_j y_c^j\left(\vec{x_j}-\frac{1}{\sum_{i=1}^{s} s_i}\sum_{i=1}^{s} s_i\vec{d_i}\right) \tag{C.65}$$

we get

$$h\left(\vec{x}\right)=\vec{w_c}\cdot\vec{x}+b=\sum_{j=1}^{n}\alpha_j y_c^j\left(\vec{x_j}-\frac{1}{\sum_{i=1}^{s} s_i}\sum_{i=1}^{s} s_i\vec{d_i}\right)\cdot\vec{x}+b \tag{C.66}$$

$$=\sum_{j=1}^{n}\alpha_j y_c^j\vec{x_j}\cdot\vec{x}-\frac{1}{\sum_{i=1}^{s} s_i}\sum_{j=1}^{n}\sum_{i=1}^{s}\alpha_j y_c^j s_i\vec{d_i}\cdot\vec{x}+b \tag{C.67}$$

and after adding internal kernels we get

$$h\left(\vec{x}\right)=\sum_{j=1}^{n}\alpha_j y_c^j K_o\left(\vec{x_j},\vec{x}\right)-\frac{1}{\sum_{i=1}^{s} s_i}\sum_{j=1}^{n}\sum_{i=1}^{s}\alpha_j y_c^j s_i K_o\left(\vec{d_i},\vec{x}\right)+b \tag{C.68}$$

when b is computed as

$$b=\frac{1}{\sum_{i=1}^{s} s_i}\left(e-\sum_{i=1}^{s} s_i\sum_{j=1}^{n}\alpha_j y_c^j\left(\vec{x_j}-\frac{1}{\sum_{i=1}^{s} s_i}\sum_{k=1}^{s} s_k\vec{d_k}\right)\cdot\vec{d_i}\right) \tag{C.69}$$

$$=\frac{1}{\sum_{i=1}^{s} s_i}\left(e-\sum_{i=1}^{s} s_i\sum_{j=1}^{n}\alpha_j y_c^j K_o\left(\vec{x_j},\vec{d_i}\right)+\frac{1}{\sum_{i=1}^{s} s_i}\sum_{i=1}^{s} s_i\sum_{j=1}^{n}\sum_{k=1}^{s}\alpha_j y_c^j s_k K_o\left(\vec{d_k},\vec{d_i}\right)\right) \tag{C.70}$$

C.5 Incorporation of the Linear Dependency to δ-SVR

We will incorporate (III.43) to δ-SVR. We assume that we use special kernels developed for δ-SVR, (II.8). For δ-SVR the condition (III.43) becomes

$$\sum_{i=1}^{s} s_i\frac{-\vec{w_{\text{red}}}\cdot\vec{d_{\text{i,red}}}-b_{\text{c}}}{w_{\text{c}}^{m+1}}=e\ , \tag{C.71}$$

After transformation

$$ew_{\text{c}}^{m+1}+\sum_{i=1}^{s} s_i\vec{w_{\text{red}}}\cdot\vec{d_{\text{i,red}}}+\sum_{i=1}^{s} s_i b_c=0 \tag{C.72}$$

$$b_c=\frac{1}{\sum_{i=1}^{s} s_i}\left(-ew_{\text{c}}^{m+1}-\sum_{i=1}^{s} s_i\vec{w_{\text{red}}}\cdot\vec{d_{\text{i,red}}}\right) \tag{C.73}$$

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

$$h\left(\vec{x}\right)=\vec{w_c}\cdot\vec{x}+\frac{1}{\sum_{i=1}^{s} s_i}\left(-ew_{\text{c}}^{m+1}-\sum_{i=1}^{s} s_i\vec{w_{\text{red}}}\cdot\vec{d_{\text{i,red}}}\right) \tag{C.74}$$

$$h\left(\vec{x}\right)=\vec{w_{\text{red}}}\cdot\vec{x_{\text{red}}}+w_c^{m+1}x_{m+1}+\frac{1}{\sum_{i=1}^{s} s_i}\left(-ew_{\text{c}}^{m+1}-\sum_{i=1}^{s} s_i\vec{w_{\text{red}}}\cdot\vec{d_{\text{i,red}}}\right) \tag{C.75}$$

$$h\left(\vec{x}\right)=\vec{w_{\text{red}}}\cdot\left(\vec{x_{\text{red}}}-\frac{1}{\sum_{i=1}^{s} s_i}\sum_{i=1}^{s} s_i\vec{d_{\text{i,red}}}\right)+w_c^{m+1}\left(x_{m+1}-\frac{e}{\sum_{i=1}^{s} s_i}\right) \tag{C.76}$$

After substituting above to OP 12, we get the φ-SVC problem without the offset with the new kernel in the form

$$K\left(\vec{x},\vec{y}\right)=\left(x_{\vec{\text{red}}}-\frac{1}{\sum_{i=1}^{s}s_i}\sum_{i=1}^{s}s_i d_{\vec{\text{i,red}}}\right)\left(y_{\vec{\text{red}}}-\frac{1}{\sum_{i=1}^{s}s_i}\sum_{i=1}^{s}s_i d_{\vec{\text{i,red}}}\right) \tag{C.77}$$

$$+\left(x_{m+1}-\frac{e}{\sum_{i=1}^{s}s_i}\right)\left(y_{m+1}-\frac{e}{\sum_{i=1}^{s}s_i}\right) \tag{C.78}$$

Then

$$K\left(\vec{x},\vec{y}\right)=x_{\vec{\text{red}}}\cdot y_{\vec{\text{red}}}-\frac{1}{\sum_{i=1}^{s}s_i}\sum_{i=1}^{s}s_i d_{\vec{\text{i,red}}}\cdot x_{\vec{\text{red}}}-\frac{1}{\sum_{i=1}^{s}s_i}\sum_{i=1}^{s}s_i d_{\vec{\text{i,red}}}\cdot y_{\vec{\text{red}}} \tag{C.79}$$

$$+\frac{1}{\left(\sum_{i=1}^{s}s_i\right)^2}\sum_{i=1}^{s}\sum_{j=1}^{s}s_i s_j d_{\vec{\text{i,red}}}\cdot d_{\vec{\text{j,red}}}+x_{m+1}y_{m+1}-x_{m+1}\frac{e}{\sum_{i=1}^{s}s_i}-y_{m+1}\frac{e}{\sum_{i=1}^{s}s_i} \tag{C.80}$$

$$+\frac{e^2}{\left(\sum_{i=1}^{s}s_i\right)^2} \tag{C.81}$$

We get nonlinear solutions by using the following way of further kernelization

$$K\left(\vec{x},\vec{y}\right)=K_o\left(x_{\vec{\text{red}}},y_{\vec{\text{red}}}\right)-\frac{1}{\sum_{i=1}^{s}s_i}\sum_{i=1}^{s}s_i K_o\left(d_{\vec{\text{i,red}}},x_{\vec{\text{red}}}\right) \tag{C.82}$$

$$-\frac{1}{\sum_{i=1}^{s}s_i}\sum_{i=1}^{s}s_i K_o\left(d_{\vec{\text{i,red}}},y_{\vec{\text{red}}}\right)+\frac{1}{\left(\sum_{i=1}^{s}s_i\right)^2}\sum_{i=1}^{s}\sum_{j=1}^{s}s_i s_j K_o\left(d_{\vec{\text{i,red}}},d_{\vec{\text{j,red}}}\right) \tag{C.83}$$

$$+x_{m+1}y_{m+1}-x_{m+1}\frac{e}{\sum_{i=1}^{s}s_i}-y_{m+1}\frac{e}{\sum_{i=1}^{s}s_i}+\frac{e^2}{\left(\sum_{i=1}^{s}s_i\right)^2}\ . \tag{C.84}$$

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

$$h\left(\vec{x}\right)=\vec{w_c}\cdot\vec{x}+b_c=\sum_{j=1}^{2n}\alpha_j y_c^j\left(x_{\vec{\text{j,red}}}-\frac{1}{\sum_{i=1}^{s}s_i}\sum_{i=1}^{s}s_i d_{\vec{\text{i,red}}}\right)\cdot x_{\vec{\text{red}}} \tag{C.85}$$

$$+\sum_{j=1}^{2n}\alpha_j y_c^j\left(x_j^{m+1}-\frac{e}{\sum_{i=1}^{s}s_i}\right)x_{m+1}+b_c \tag{C.86}$$

with internal kernels

$$h\left(\vec{x}\right)=\sum_{j=1}^{2n}\alpha_j y_c^j\left(K_o\left(x_{\vec{\text{j,red}}},x_{\vec{\text{red}}}\right)-\frac{1}{\sum_{i=1}^{s}s_i}\sum_{i=1}^{s}s_i K_o\left(d_{\vec{\text{i,red}}},x_{\vec{\text{red}}}\right)\right) \tag{C.87}$$

$$+\sum_{j=1}^{2n}\alpha_j y_c^j\left(x_j^{m+1}-\frac{e}{\sum_{i=1}^{s}s_i}\right)x_{m+1}+b_c \tag{C.88}$$

where b_c is computed as

$$b_c=\frac{1}{\sum_{i=1}^{s}s_i}\left(-e\sum_{j=1}^{2n}\alpha_j y_c^j\left(x_j^{m+1}-\frac{e}{\sum_{i=1}^{s}s_i}\right)\right) \tag{C.89}$$

$$-\frac{1}{\sum_{i=1}^{s}s_i}\left(\sum_{i=1}^{s}s_i\sum_{j=1}^{2n}\alpha_j y_c^j\left(x_{\vec{\text{j,red}}}-\frac{1}{\sum_{i=1}^{s}s_i}\sum_{k=1}^{s}s_k d_{\vec{\text{k,red}}}\right)\cdot d_{\vec{\text{i,red}}}\right) \tag{C.90}$$

$$= -\frac{1}{\sum_{i=1}^{s} s_i} e \sum_{j=1}^{2n} \alpha_j y_c^j \left(x_j^{m+1} - \frac{e}{\sum_{i=1}^{s} s_i} \right) - \frac{1}{\sum_{i=1}^{s} s_i} \sum_{i=1}^{s} s_i \sum_{j=1}^{2n} \alpha_j y_c^j K_o \left(x_{\mathrm{j,red}}^{\rightarrow}, d_{\mathrm{i,red}}^{\rightarrow} \right) \tag{C.91}$$

$$+ \frac{1}{\sum_{i=1}^{s} s_i} \sum_{i=1}^{s} s_i \sum_{j=1}^{2n} \alpha_j y_c^j \frac{1}{\sum_{i=1}^{s} s_i} \sum_{k=1}^{s} s_k K_o \left(d_{\mathrm{k,red}}^{\rightarrow}, d_{\mathrm{i,red}}^{\rightarrow} \right) \tag{C.92}$$

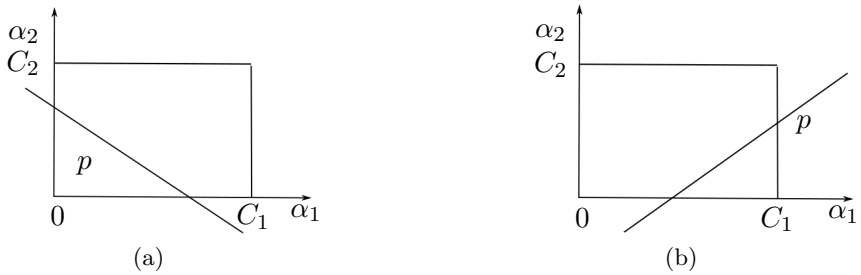


Figure D.1: Visualization of the constraints. We can see a line p with the negative slope in a) and the positive in b)

Appendix D

Solving SVM by Decomposition

D.1 Derivation of SMO β_2 Bounds for φ-SVC

We will derive bounds for β_2 (IV.7):

$$U \leq \alpha_2 \leq V \ , \tag{D.1}$$

where for $y_1 \neq y_2$

$$U = \max\left(0,\ \alpha_2^{old} - \alpha_1^{old}\right), \tag{D.2}$$

$$V = \min\left(C_2,\ C_1 - \alpha_1^{old} + \alpha_2^{old}\right) \ , \tag{D.3}$$

for $y_1 = y_2$

$$U = \max\left(0,\ \alpha_1^{old} + \alpha_2^{old} - C_1\right), \tag{D.4}$$

$$V = \min\left(C_2,\ \alpha_1^{old} + \alpha_2^{old}\right) \ . \tag{D.5}$$

We present two derivations: geometrical and analytical one.

Geometrical Proof. The equality equation of SVM is

$$\alpha_2 = \alpha_1^{\text{old}} y_1 y_2 + \alpha_2^{\text{old}} - \alpha_1 y_1 y_2 \ . \tag{D.6}$$

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.6) we get

$$p : \alpha_2 = \alpha_1^{old} + \alpha_2^{old} - \alpha_1 \ . \tag{D.7}$$

The line p has a negative slope equals to -1, Fig. D.1a.

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 $\alpha_1 = C_1$:

$$\alpha_2 = \alpha_1^{old} + \alpha_2^{old} \tag{D.8}$$

and

$$\alpha_2 = \alpha_1^{old} + \alpha_2^{old} - C_1 \ . \tag{D.9}$$

Because crossing points have to lie in the square we get the following bounds for α_2

$$U = \max\left(0,\ \alpha_1^{old} + \alpha_2^{old} - C_1\right), \tag{D.10}$$

$$V = \min\left(C_2,\ \alpha_1^{old} + \alpha_2^{old}\right) \ . \tag{D.11}$$

When $y_1 \neq y_2$, then $y_1 y_2 = -1$, after substituting it to (D.6)

$$p : \alpha_2 = -\alpha_1^{old} + \alpha_2^{old} + \alpha_1 \ . \tag{D.12}$$

The line p has a positive slope equals to 1, Fig. D.1b. 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 $\alpha_1 = C_1$:

$$\alpha_2 = -\alpha_1^{old} + \alpha_2^{old} \tag{D.13}$$

and

$$\alpha_2 = -\alpha_1^{old} + \alpha_2^{old} + C_1 \ . \tag{D.14}$$

Because crossing points have to lie in the square we get the following bounds for α_2

$$U = \max\left(0,\ \alpha_2^{old} - \alpha_1^{old}\right), \tag{D.15}$$

$$V = \min\left(C_2,\ C_1 - \alpha_1^{old} + \alpha_2^{old}\right) \ . \tag{D.16}$$

□

Analytical Proof. We have an inequality for α_1

$$0 \leq \alpha_1 \leq C_1 \tag{D.17}$$

and a line p

$$\alpha_1 y_1 + \alpha_2 y_2 = \alpha_1^{old} y_1 + \alpha_2^{old} y_2 \ , \tag{D.18}$$

after transformation

$$\alpha_1 = \alpha_1^{old} + y_1 y_2 \alpha_2^{old} - y_1 y_2 \alpha_2 \ , \tag{D.19}$$

after substituting it to the inequality we get

$$0 \leq \alpha_1^{old} + y_1 y_2 \alpha_2^{old} - y_1 y_2 \alpha_2 \leq C_1 \ . \tag{D.20}$$

When $y_1 = y_2$, then $y_1 y_2 = 1$, so

$$0 \leq \alpha_1^{old} + \alpha_2^{old} - \alpha_2 \leq C_1 \ . \tag{D.21}$$

Consider now the first part of the inequality,

$$\alpha_1^{old} + \alpha_2^{old} - \alpha_2 \geq 0 \tag{D.22}$$

$$\alpha_2 \leq \alpha_1^{old} + \alpha_2^{old} \ . \tag{D.23}$$

Because the parameter α_2 has to satisfy the inequality $\alpha_2 \leq C_2$, so the upper bound for α_2 is

$$V = \min\left(C_2,\ \alpha_1^{old} + \alpha_2^{old}\right) \ . \tag{D.24}$$

Considering the second part of the inequality,

$$\alpha_1^{old} + \alpha_2^{old} - \alpha_2 \leq C_1 \tag{D.25}$$

$$\alpha_2 \geq \alpha_1^{old} + \alpha_2^{old} - C_1 \ . \tag{D.26}$$

Because the parameter α_2 has to satisfy the inequality $\alpha_2 \geq 0$, so the lower bound for α_2 is

$$U = \max\left(0,\ \alpha_1^{old} + \alpha_2^{old} - C_1\right) \ . \tag{D.27}$$

When $y_1 \neq y_2$, then $y_1 y_2 = -1$, so

$$0 \leq \alpha_1^{old} - \alpha_2^{old} + \alpha_2 \leq C_1 \ . \tag{D.28}$$

Consider now the first part of the inequality,

$$\alpha_1^{old} - \alpha_2^{old} + \alpha_2 \geq 0 \tag{D.29}$$

$$\alpha_2 \geq \alpha_2^{old} - \alpha_1^{old} \ . \tag{D.30}$$

Because the parameter α_2 has to satisfy the inequality $\alpha_2 \geq 0$, so the lower bound for α_2 is

$$U = \max\left(0,\ \alpha_2^{old} - \alpha_1^{old}\right) \ . \tag{D.31}$$

Considering the second part of the inequality,

$$\alpha_1^{old} - \alpha_2^{old} + \alpha_2 \leq C_1 \tag{D.32}$$

$$\alpha_2 \leq C_1 + \alpha_2^{old} - \alpha_1^{old} \ . \tag{D.33}$$

Because the parameter α_2 has to satisfy the inequality $\alpha_2 \leq C_2$, so the upper bound for α_2 is

$$V = \min\left(C_2,\ C_1 - \alpha_1^{old} + \alpha_2^{old}\right) \ . \tag{D.34}$$

□

D.2 Derivation of the SMO Solution

We have to find new values of parameters in SMO step for SVC. First we compute α_2^{unc}

$$\alpha_2^{\text{unc}} = \alpha_2^{\text{old}} + \frac{y_2\left(E_1 - E_2\right)}{\kappa} \tag{D.35}$$

and then

$$\alpha_2 = \begin{cases} V, & if\ \alpha_2^{\text{new,unc}} > V, \\ \alpha_2^{\text{new,unc}}, & if\ U \leq \alpha_2^{\text{new,unc}} \leq V, \\ U, & if\ \alpha_2^{\text{new,unc}} < U \end{cases} \tag{D.36}$$

$$\alpha_1^{\text{new}} = \alpha_1^{\text{old}} + y_1 y_2\left(\alpha_2^{\text{old}} - \alpha_2^{\text{new}}\right) \ . \tag{D.37}$$

For simplicity of the proof we use a notation

$$K\left(\vec{x_i}, \vec{x_j}\right) \equiv K_{ij} \ , \tag{D.38}$$

where $i, j = 1, 2$, and we define

$$f_i = \sum_{j=1}^{n} y_j \alpha_j K_{ij} \tag{D.39}$$

$$E_i = f_i - y_i = \sum_{j=1}^{n} y_j \alpha_j K_{ij} - y_i \tag{D.40}$$

$$v_i = \sum_{j=3}^{n} y_j \alpha_j K_{ij} = f_i - \sum_{j=1}^{2} y_j \alpha_j K_{ij} \tag{D.41}$$

for $i = 1$ or $i = 2$, where n is the number of all vectors.

The objective function has a form

$$\begin{array}{l} f(\alpha_1, \alpha_2) = \alpha_1 + \alpha_2 - \frac{1}{2} K_{11} \alpha_1^2 - \frac{1}{2} K_{22} \alpha_2^2 - \\ y_1 y_2 K_{12} \alpha_1 \alpha_2 - y_1 \alpha_1 v_1 - y_2 \alpha_2 v_2 + \text{const} \ . \end{array} \tag{D.42}$$

After substituting $s_{ij} = y_i y_j$ for $i, j = 1, 2$ for simplification of notation we get

$$\begin{array}{l} f(\alpha_1, \alpha_2) = \alpha_1 + \alpha_2 - \frac{1}{2} K_{11} \alpha_1^2 - \frac{1}{2} K_{22} \alpha_2^2 - \\ s_{12} K_{12} \alpha_1 \alpha_2 - y_1 \alpha_1 v_1 - y_2 \alpha_2 v_2 + \text{const} \ . \end{array} \tag{D.43}$$

The linear constraint has a form: $\sum_{i=1}^{n} y_i \alpha_i = 0$. It must be satisfied with new values of α_1 and α_2

$$y_1 \alpha_1 + y_2 \alpha_2 = y_1 \alpha_1^{\text{old}} + y_2 \alpha_2^{\text{old}} = \text{const} \ . \tag{D.44}$$

Dividing above by y_1 and noticing that $y_1 y_2 = y_1 / y_2$ we get

$$\alpha_1 + y_1 y_2 \alpha_2 = \alpha_1^{\text{old}} + y_1 y_2 \alpha_2^{\text{old}} \ , \tag{D.45}$$

after simplification

$$\alpha_1 + s_{12} \alpha_2 = \alpha_1^{\text{old}} + s_{12} \alpha_2^{\text{old}} \ . \tag{D.46}$$

Introducing notation

$$\gamma = \alpha_1^{\text{old}} + s_{12} \alpha_2^{\text{old}} \tag{D.47}$$

we have

$$\alpha_1 + s_{12} \alpha_2 = \gamma \tag{D.48}$$

$$\alpha_1 = \gamma - s_{12} \alpha_2 \ . \tag{D.49}$$

The above equation shows how to get α_1 from α_2. After substituting above to the objective function we get

$$\begin{array}{l} f(\alpha_1, \alpha_2) = \gamma - s_{12} \alpha_2 + \alpha_2 - \frac{1}{2} K_{11} (\gamma - s_{12} \alpha_2)^2 - \frac{1}{2} K_{22} \alpha_2^2 - \\ s_{12} K_{12} (\gamma - s_{12} \alpha_2) \alpha_2 - y_1 (\gamma - s_{12} \alpha_2) v_1 - y_2 \alpha_2 v_2 + \text{const} \ . \end{array} \tag{D.50}$$

After transformation

$$\begin{array}{l} f(\alpha_1, \alpha_2) = \gamma - s_{12} \alpha_2 + \alpha_2 - \frac{1}{2} K_{11} (\gamma^2 - 2\gamma s_{12} \alpha_2 + s_{12}^2 \alpha_2^2) - \frac{1}{2} K_{22} \alpha_2^2 - \\ s_{12} K_{12} (\gamma - s_{12} \alpha_2) \alpha_2 - y_1 (\gamma - s_{12} \alpha_2) v_1 - y_2 \alpha_2 v_2 + \text{const} \end{array} \tag{D.51}$$

$$\begin{array}{l} f(\alpha_1, \alpha_2) = \gamma - s_{12} \alpha_2 + \alpha_2 - \frac{1}{2} K_{11} \gamma^2 + K_{11} \gamma s_{12} \alpha_2 - \frac{1}{2} K_{11} \alpha_2^2 - \frac{1}{2} K_{22} \alpha_2^2 \\ - s_{12} K_{12} (\gamma - s_{12} \alpha_2) \alpha_2 - y_1 (\gamma - s_{12} \alpha_2) v_1 - y_2 \alpha_2 v_2 + \text{const} \end{array} \tag{D.52}$$

$$\begin{array}{l} f(\alpha_1, \alpha_2) = \gamma - s_{12} \alpha_2 + \alpha_2 - \frac{1}{2} K_{11} \gamma^2 + K_{11} \gamma s_{12} \alpha_2 - \frac{1}{2} K_{11} \alpha_2^2 - \frac{1}{2} K_{22} \alpha_2^2 \\ - s_{12} K_{12} \gamma \alpha_2 + K_{12} \alpha_2^2 - y_1 \gamma v_1 + y_2 \alpha_2 v_1 - y_2 \alpha_2 v_2 + \text{const} \ . \end{array} \tag{D.53}$$

Now we compute a partial derivative with respect to α_2

$$\begin{array}{l} \frac{\partial f(\alpha_2)}{\partial \alpha_2} = 1 - s_{12} + K_{11} \gamma s_{12} - K_{11} \alpha_2 - K_{22} \alpha_2 \\ - s_{12} K_{12} \gamma + 2 K_{12} \alpha_2 + y_2 v_1 - y_2 v_2 \ . \end{array} \tag{D.54}$$

We are looking for stationary points by equating the derivative to zero

$$\begin{array}{l} \frac{\partial f(\alpha_2)}{\partial \alpha_2} = 1 - s_{12} + K_{11} \gamma s_{12} - K_{11} \alpha_2 - K_{22} \alpha_2 - \\ s_{12} K_{12} \gamma + 2 K_{12} \alpha_2 + y_2 v_1 - y_2 v_2 = 0 \end{array} \tag{D.55}$$

$$\begin{aligned} &1 - s_{12} + K_{11}\gamma s_{12} - K_{11}\alpha_2 - K_{22}\alpha_2 \\ &-s_{12}K_{12}\gamma + 2K_{12}\alpha_2 + y_2 v_1 - y_2 v_2 = 0 \ . \end{aligned} \tag{D.56}$$

Dividing both sides by y_2 we get

$$\begin{aligned} &y_2 - y_1 + K_{11}\gamma y_1 - K_{11} y_2 \alpha_2 - K_{22} y_2 \alpha_2 - \\ &y_1 K_{12}\gamma + 2K_{12} y_2 \alpha_2 + v_1 - v_2 = 0 \ . \end{aligned} \tag{D.57}$$

Adding the new superscript for α_2 we get

$$\begin{aligned} &y_2 - y_1 + K_{11}\gamma y_1 - K_{11} y_2 \alpha_2^{\text{new}} - K_{22} y_2 \alpha_2^{\text{new}} - \\ &y_1 K_{12}\gamma + 2K_{12} y_2 \alpha_2^{\text{new}} + v_1 - v_2 = 0 \ . \end{aligned} \tag{D.58}$$

Substituting for γ, v_1 i v_2

$$\begin{aligned} &y_2 - y_1 + K_{11}\left(\alpha_1 + s_{12}\alpha_2\right) y_1 - K_{11} y_2 \alpha_2^{\text{new}} - K_{22} y_2 \alpha_2^{\text{new}} \\ &-y_1 K_{12}\left(\alpha_1 + s_{12}\alpha_2\right) + 2K_{12} y_2 \alpha_2^{\text{new}} + f_1 - y_1\alpha_1 K_{11} - y_2\alpha_2 K_{12} \\ &-f_2 + y_1\alpha_1 K_{12} + y_2\alpha_2 K_{22} = 0 \end{aligned} \tag{D.59}$$

$$\begin{aligned} &y_2 - y_1 + K_{11}\alpha_1 y_1 + K_{11} y_2 \alpha_2 - K_{11} y_2 \alpha_2^{\text{new}} - K_{22} y_2 \alpha_2^{\text{new}} \\ &-y_1 K_{12}\alpha_1 - K_{12} y_2 \alpha_2 + 2K_{12} y_2 \alpha_2^{\text{new}} + f_1 - y_1\alpha_1 K_{11} - y_2\alpha_2 K_{12} \\ &-f_2 + y_1\alpha_1 K_{12} + y_2\alpha_2 K_{22} = 0 \end{aligned} \tag{D.60}$$

$$\begin{aligned} &y_2 - y_1 + K_{11} y_2 \alpha_2 - K_{11} y_2 \alpha_2^{\text{new}} - K_{22} y_2 \alpha_2^{\text{new}} \\ &-K_{12} y_2 \alpha_2 + 2K_{12} y_2 \alpha_2^{\text{new}} + f_1 - y_2\alpha_2 K_{12} \\ &-f_2 + y_2\alpha_2 K_{22} = 0 \end{aligned} \tag{D.61}$$

$$\begin{aligned} &y_2 - y_1 - y_2\alpha_2^{\text{new}}\left(K_{11} + K_{22} - 2K_{12}\right) + y_2\alpha_2(K_{11} + K_{22} - 2K_{12}) \\ &+f_1 - f_2 = 0 \ . \end{aligned} \tag{D.62}$$

Introducing notation $\kappa = K_{11} + K_{22} - 2K_{12}$ we get

$$y_2 - y_1 - y_2\alpha_2^{\text{new}}\kappa + y_2\alpha_2\kappa + f_1 - f_2 = 0 \ . \tag{D.63}$$

Dividing both sides by y_2 and κ

$$\alpha_2^{\text{new}} = \alpha_2 + \frac{y_2\left(E_1 - E_2\right)}{\kappa} \ . \tag{D.64}$$

After all we have to limit α_2^{new}, so it will lie in $[U, V]$.

D.3 Derivation of the SMO solution for φ-SVC

Compare it with D.2. We have a new objective function

$$f\left(\alpha_1, \alpha_2\right) = \alpha_1\varphi_1 + \alpha_2\varphi_2 + f_{\text{smo}}\left(\alpha_1, \alpha_2\right) \ , \tag{D.65}$$

where f_{smo} is the f function for SMO from D.2. After substituting

$$\alpha_1 = \gamma - y_1 y_2 \alpha_2 \ , \tag{D.66}$$

where

$$\gamma = \alpha_1^{\text{old}} + y_1 y_2 \alpha_2^{\text{old}} \tag{D.67}$$

we get

$$f\left(\alpha_1, \alpha_2\right) = \varphi_1\gamma - \varphi_1 y_1 y_2 \alpha_2 + \alpha_2\varphi_2 + f_{\text{smo}}\left(\alpha_1, \alpha_2\right) \ . \tag{D.68}$$

After differentiating we get

$$\frac{\partial f\left(\alpha_1,\alpha_2\right)}{\partial\alpha_2}=\varphi_2-\varphi_1y_1y_2+\frac{\partial f_{\text{smo}}\left(\alpha_1,\alpha_2\right)}{\partial\alpha_2}\ . \tag{D.69}$$

And a solution is

$$\alpha_2^{\text{new}}=\alpha_2+\frac{y_2\left(E_1-E_2\right)}{\kappa}\ , \tag{D.70}$$

where

$$\begin{aligned} E_i &= \sum_{j=1}^{n}y_j\alpha_jK_{ij}-y_i-y_i\varphi_i \\ \kappa &= K_{11}+K_{22}-2K_{12}\ . \end{aligned} \tag{D.71}$$

D.4 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

$$K\left(\vec{x_i},\vec{x_j}\right)\equiv K_{ij}\ , \tag{D.72}$$

where $i,j=1,2$, and we define

$$E_i=\sum_{j=1}^{n}y_j\alpha_jK_{ij}-y_i \tag{D.73}$$

$$v_1=\sum_{j=2}^{n}y_j\alpha_jK_{ij}=\sum_{j=1}^{n}y_j\alpha_jK_{1j}-y_1\alpha_1K_{11}\ . \tag{D.74}$$

The objective function has a form

$$W\left(\alpha_1\right)=\alpha_1-\tfrac{1}{2}K_{11}\alpha_1^2-y_1\alpha_1v_1+\text{const}\ .\ . \tag{D.75}$$

Now we compute a partial derivative with respect to α_1

$$\tfrac{\partial W(\alpha_1)}{\partial\alpha_1}=1-K_{11}\alpha_1-y_1v_1\ . \tag{D.76}$$

We are looking for stationary points by equating the derivative to zero

$$\frac{\partial W\left(\alpha_1\right)}{\partial\alpha_1}=1-K_{11}\alpha_1^{\text{new}}-y_1v_1=0 \tag{D.77}$$

$$1-K_{11}\alpha_1^{\text{new}}-y_1\sum_{j=1}^{n}y_j\alpha_jK_{1j}+\alpha_1K_{11}=0 \tag{D.78}$$

$$\alpha_1^{\text{new}}=\alpha_1-\frac{y_1E_1}{K_{11}}\ . \tag{D.79}$$

Then we have to bound α_1^{new}:

$$0\leq\alpha_1^{\text{new}}\leq C_1\ . \tag{D.80}$$

D.5 Derivation of SMO Without Offset for φ-SVC

Compare it with *Appendix D.4*. We have a new objective function

$$f\left(\alpha_1\right)=\alpha_1\varphi_1+f_{\text{smo}}\left(\alpha_1,\alpha_2\right)\ , \tag{D.81}$$

where f_{smo} is the f function for SMO from *Appendix D.4*. After differentiating we get

$$\frac{\partial f\left(\alpha_1\right)}{\partial \alpha_1} = \varphi_1 + \frac{\partial f_{\text{smo}}\left(\alpha_1\right)}{\partial \alpha_1} \ . \tag{D.82}$$

And a solution is

$$\alpha_1^{\text{new}} = \alpha_1 - \frac{y_1 E_1}{K_{11}} \ , \tag{D.83}$$

where

$$E_i = \sum_{j=1}^{n} y_j \alpha_j K_{ij} - y_i - y_i \varphi_i \ . \tag{D.84}$$

D.6 Derivation of Optimization Possibility Conditions

D.6.1 Optimization Possibility Conditions Derived Directly

We can transform the linear constraint (IV.2) into the following form

$$\beta_d = -y_{c_d} \sum_{\substack{i=1 \\ i \neq d}}^{p} y_{c_i} \beta_i - y_{c_d} \sum_{\substack{i=1 \\ i \notin C}}^{n} y_i \alpha_i \ , \tag{D.85}$$

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

OP 30.

$$\begin{aligned}
\max_{\vec{\gamma}} \quad & f_3\left(\vec{\gamma}\right) = -y_{c_d} \sum_{\substack{i=1 \\ i \neq d}}^{p} y_{c_i} \gamma_{e_i} - y_{c_d} \sum_{\substack{i=1 \\ i \notin C}}^{n} y_i \alpha_i + \sum_{\substack{i=1 \\ i \neq d}}^{p} \gamma_{e_i} \\
& + \sum_{\substack{i=1 \\ i \notin C}}^{n} \alpha_i - \frac{1}{2} \left(\sum_{\substack{i=1 \\ i \neq d}}^{p} y_{c_i} \gamma_{e_i} + \sum_{\substack{i=1 \\ i \notin C}}^{n} y_i \alpha_i \right)^2 K_{c_d c_d} \\
& + \left(\sum_{\substack{i=1 \\ i \neq d}}^{p} y_{c_i} \gamma_{e_i} + \sum_{\substack{i=1 \\ i \notin C}}^{n} y_i \alpha_i \right) \sum_{\substack{i=1 \\ i \neq d}}^{p} y_{c_i} \gamma_{e_i} K_{c_d c_i} \\
& - \frac{1}{2} \sum_{\substack{i=1 \\ i \neq d}}^{p} y_{c_i} \gamma_{e_i} \sum_{\substack{j=1 \\ j \neq d}}^{p} y_{c_j} \gamma_{e_j} K_{c_i c_j} \\
& + \left(\sum_{\substack{i=1 \\ i \neq d}}^{p} y_{c_i} \gamma_{e_i} + \sum_{\substack{i=1 \\ i \notin C}}^{n} y_i \alpha_i \right) \sum_{\substack{i=1 \\ i \notin C}}^{n} y_i \alpha_i K_{c_d i} \\
& - \sum_{\substack{i=1 \\ i \neq d}}^{p} y_{c_i} \gamma_{e_i} \sum_{\substack{j=1 \\ j \notin C}}^{n} y_j \alpha_j K_{c_i j} - \frac{1}{2} \sum_{\substack{i=1 \\ i \notin C}}^{n} \sum_{\substack{j=1 \\ j \notin C}}^{n} y_{ij} \alpha_i \alpha_j K_{ij}
\end{aligned} \tag{D.86}$$

subject to

$$0 \leq \gamma_{e_i} \leq C, \text{ for } i \in \{1, 2, \ldots, p\} \setminus \{d\}, \ C > 0 \tag{D.87}$$

$$0 \leq c = -y_{c_d} \sum_{\substack{i=1 \\ i \neq d}}^{p} y_{c_i} \gamma_{e_i} - y_{c_d} \sum_{\substack{i=1 \\ i \notin C}}^{n} y_i \alpha_i \leq C \ , \tag{D.88}$$

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 satisfy the constraints from O_1 problem.

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

$$\frac{\partial}{\partial \gamma_{e_k}} f_3(\vec{\gamma}_{\text{old}}) = y_{c_k}(E_{c_d} - E_{c_k}) \quad , \tag{D.89}$$

where E_i is defined in (IV.5).

Let's analyze conditions for optimization possibility. The first 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 satisfying (D.88), the second part consists of conditions based on partial derivatives. Merging all conditions leads to the overall optimization conditions.

The (D.88) must be satisfied after changes, hence we can write

$$\begin{aligned} -\alpha_{c_d}^{\text{old}} \leq \Delta\alpha_{c_d} = -\alpha_{c_d}^{\text{old}} - y_{c_d} \sum_{\substack{i=1 \\ i \neq d}}^{p} y_{c_i} \alpha_{c_i}^{\text{new}} \\ -y_{c_d} \sum_{\substack{i=1 \\ i \notin C}}^{n} y_i \alpha_i \leq C - \alpha_{c_d}^{\text{old}} \quad . \end{aligned} \tag{D.90}$$

After substituting

$$\alpha_{c_d}^{\text{old}} = -y_{c_d} \sum_{\substack{i=1 \\ i \neq d}}^{p} y_{c_i} \alpha_{c_i}^{\text{old}} - y_{c_d} \sum_{\substack{i=1 \\ i \notin C}}^{n} y_i \alpha_i \tag{D.91}$$

we get the following condition

$$-\alpha_{c_d}^{\text{old}} \leq \Delta\alpha_{c_d} = -y_{c_d} \sum_{\substack{i=1 \\ i \neq d}}^{p} y_{c_i} \Delta\alpha_{c_i} \leq C - \alpha_{c_d}^{\text{old}} \quad . \tag{D.92}$$

Theorem D.6.1 (Necessary optimization conditions based on satisfying (D.88))**.** *If the condition* (D.92) *is satisfied, then there exist two parameters c_i, where $i \in \{1, \ldots, p\}$ that belong to the opposite groups G_1 and G_2 defined as*

$$\begin{aligned} G_1 := \{i \in \{1, 2, \ldots, n\} : (y_i = 1 \wedge \alpha_i = 0) \\ \vee (y_i = -1 \wedge \alpha_i = C_i) \vee (0 < \alpha_i < C_i)\} \\ G_2 := \{i \in \{1, 2, \ldots, n\} : (y_i = -1 \wedge \alpha_i = 0) \\ \vee (y_i = 1 \wedge \alpha_i = C_i) \vee (0 < \alpha_i < C_i)\} \quad . \end{aligned} \tag{D.93}$$

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 (D.92) will not be satisfied. We choose parameters belong to the G_1 group. The proof for the 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 values of c_i parameters for $i \in \{1, 2, \ldots, p\} \setminus \{d\}$ remain the same, 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 all values of c_i parameters remain the same 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 (D.92) becomes $0 \leq \Delta\alpha_{c_d} \leq C$, what cannot be true. If $y_{c_d} = -1$, then $\Delta\alpha_{c_d} > 0$ and $\alpha_{c_d}^{\text{old}} = C$. The condition (D.92) becomes $-C \leq \Delta\alpha_{c_d} \leq 0$ what cannot be true. □

Theorem D.6.2 (Sufficient optimization conditions based on satisfying (D.88))**.** *If there exist two parameters c_i, where $i \in \{1, \ldots, p\}$ that belong to the opposite groups G_1 and G_2, then condition* (D.92) *is satisfied 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 (D.92) is satisfied. 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 (D.92) is satisfied. If $0 < \alpha_{c_d}^{\text{old}} < C$, then condition (D.92) is satisfied, 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 (D.92) is satisfied. If $0 < \alpha_{c_d}^{\text{old}} < C$, then condition (D.92) is satisfied, when $\Delta\alpha_{c_b}$ is set to close enough to zero value. □

Theorem D.6.3 (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 satisfy the following condition*

$$\begin{aligned} \frac{f_3(\vec{\gamma})}{\gamma_{e_k}} &> 0 \text{ when } \alpha_{c_k} = 0 \\ \frac{f_3(\vec{\gamma})}{\gamma_{e_k}} &< 0 \text{ when } \alpha_{c_k} = C_{c_k} \\ \frac{f_3(\vec{\gamma})}{\gamma_{e_k}} &\neq 0 \text{ when } 0 < \alpha_{c_k} < C_{c_k} \quad . \end{aligned} \tag{D.94}$$

Proof. We prove that if all partial derivatives of the function f_3 violate the condition (D.94), then optimization will be impossible. If (D.94) is violated, 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 D.6.1. *After substitution (D.89) to (D.94) we get*

$$y_{c_k} \left(E_{c_d} - E_{c_k}\right) > 0 \text{ when } \alpha_{c_k} = 0 \tag{D.95}$$

$$y_{c_k} \left(E_{c_d} - E_{c_k}\right) < 0 \text{ when } \alpha_{c_k} = C_{c_k} \tag{D.96}$$

$$y_{c_k} \left(E_{c_d} - E_{c_k}\right) \neq 0 \text{ when } 0 < \alpha_{c_k} < C_{c_k} \tag{D.97}$$

After simplification:
When $y_{c_k} = 1$:

$$E_{c_k} < E_{c_d} \text{ when } \alpha_{c_k} = 0 \tag{D.98}$$

$$E_{c_k} > E_{c_d} \text{ when } \alpha_{c_k} = C_{c_k} \tag{D.99}$$

$$E_{c_k} \neq E_{c_d} \text{ when } 0 < \alpha_{c_k} < C_{c_k} \tag{D.100}$$

When $y_{c_k} = -1$:

$$E_{c_k} > E_{c_d} \text{ when } \alpha_{c_k} = 0 \tag{D.101}$$

$$E_{c_k} < E_{c_d} \text{ when } \alpha_{c_k} = C_{c_k} \tag{D.102}$$

$$E_{c_k} \neq E_{c_d} \text{ when } 0 < \alpha_{c_k} < C_{c_k} \tag{D.103}$$

Theorem D.6.4 (Sufficient optimization conditions based on partial derivatives)**.** *If one of the partial derivatives of the function f_3 satisfies the condition (D.94), then optimization is possible based on partial derivatives for some parameter changes.*

Proof. We can change the parameter that satisfies the condition (D.94). The remaining parameters, which are attributed to f_3 variables, can stay unchanged, and then f_3 value will grow. □

Theorem D.6.5 (Overall optimization conditions)**.** *Optimization is possible for some parameter changes, if and only if there exist two parameters c_i, where $i \in \{1, 2, \ldots, p\}$ that belong to the opposite groups G_1 and G_2 and one of the partial derivatives of the function f_3 satisfies the condition (D.94).*

Proof. Because of Thm. D.6.1, Thm. D.6.2, Thm. D.6.3, Thm. D.6.4 we only have to prove, that overall optimization is a multiplication of optimization based on (D.88) 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 satisfied with respect to the linear conditions. There is additionally only one nonlinear constraint. When it is also satisfied, then optimization is possible. □

We can see that the Thm. D.6.5 is a different formulation of the Thm. IV.3.1.

D.6.2 Optimization Possibility Conditions Derived From KKT

We can derive the optimization possibility conditions 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)

$$y_i \left(\vec{w} \cdot \vec{x_i} + b\right) \geq 1 + \varphi_i \tag{D.104}$$

  when $y_i = 1$

$$b \geq 1 + \varphi_i - \vec{w} \cdot \vec{x_i} \tag{D.105}$$

  when $y_i = -1$

$$b \leq -\vec{w} \cdot \vec{x_i} - 1 - \varphi_i \ . \tag{D.106}$$

- When $\alpha_i = C_i$

$$y_i \left(\vec{w} \cdot \vec{x_i} + b\right) - 1 - \varphi_i + \xi_i = 0 \tag{D.107}$$

$$\xi_i = -y_i \left(\vec{w} \cdot \vec{x_i} + b\right) + 1 + \varphi_i \ . \tag{D.108}$$

  Because $\xi_i \geq 0$, so

$$-\, y_i \left(\vec{w} \cdot \vec{x_i} + b\right) + 1 + \varphi_i \geq 0 \tag{D.109}$$

$$y_i \left(\vec{w} \cdot \vec{x_i} + b\right) \leq 1 + \varphi_i \tag{D.110}$$

  when $y_i = 1$

$$b \leq 1 + \varphi_i - \vec{w} \cdot \vec{x_i} \tag{D.111}$$

  when $y_i = -1$

$$b \geq -\vec{w} \cdot \vec{x_i} - 1 - \varphi_i \ . \tag{D.112}$$

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

$$y_i \left(\vec{w} \cdot \vec{x_i} + b\right) - 1 - \varphi_i = 0 \tag{D.113}$$

$$b = -\vec{w} \cdot \vec{x_i} + y_i + y_i \varphi_i \ . \tag{D.114}$$

After substituting (I.13) to above equations we get

- when $\alpha_i = 0$ and $y_i = 1$

$$b \geq -E_i \tag{D.115}$$

  when $y_i = -1$

$$b \leq -E_i \ , \tag{D.116}$$

- when $\alpha_i = C_i$ and $y_i = 1$

$$b \leq -E_i \tag{D.117}$$

  when $y_i = -1$

$$b \geq -E_i \ , \tag{D.118}$$

- when $0 < \alpha_i < C_i$

$$b = -E_i \ , \tag{D.119}$$

where E_i is defined in (IV.5).

We will prove that from above equations and (D.92), we can implicate Thm. IV.3.1.

Proof. First, we can notice that the conditions from (D.115) to (D.119) are violated when we have two points from the same group G_1 or G_2. It is a direct conclusion from (D.92). When they come from separated groups and the first one is not greater than 0 or below C_i, and it is from G_1, and the second one is from G_2, then when we analyze (D.115) to (D.119), we can see that $b \geq -E_1$ and $b \leq -E_2$. Merging both inequalities we get $E_2 \leq E_1$. Because it is requirement for an optimal solution, so optimization is possible when $E_2 > E_1$. When both parameters fulfill $0 < \alpha_i < C_i$, then we can see that optimization is possible when $E_1 \neq E_2$ in both approaches. □

D.7 Derivation of the Dual Form of OP 20

OP 31.

$$\max_{\vec{\alpha},\vec{r}} \quad d\left(\vec{\alpha},\vec{r}\right) \tag{D.120}$$

where

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

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

$$\frac{\partial t\left(\vec{w},b,\vec{\alpha},\vec{\xi},\vec{r}\right)}{\partial w_i} = w_i - \sum_{j=1}^{n} \alpha_j y_c^j x_{ji} = 0 \tag{D.121}$$

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

$$\frac{\partial t\left(\vec{w},b,\vec{\alpha},\vec{\xi},\vec{r}\right)}{\partial b} = \sum_{i=1}^{n} \alpha_i y_c^i = D \ . \tag{D.122}$$

A partial derivative with respect to ξ_i is

$$\frac{\partial t\left(\vec{w},b,\vec{\alpha},\vec{\xi},\vec{r}\right)}{\partial \xi_i} = C_i - r_i - \alpha_i = 0 \ . \tag{D.123}$$

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

$$\begin{aligned} d(\vec{\alpha},\vec{r}) &= \tfrac{1}{2}\sum_{i=1}^{m}\left(\sum_{j=1}^{n}\alpha_j y_c^j x_{ji}\right)\left(\sum_{k=1}^{n}\alpha_k y_c^k x_{ki}\right) \\ &- \sum_{i=1}^{n}\alpha_i y_c^i\left(\sum_{j=1}^{m} w_j x_{ij} + b\right) + \sum_{i=1}^{n}\alpha_i(1+\varphi_i) + \sum_{i=1}^{n} C_i\xi_i + Db \\ &- \sum_{i=1}^{n}\alpha_i\xi_i - \sum_{i=1}^{n} r_i\xi_i \end{aligned} \tag{D.124}$$

$$\begin{aligned} d(\vec{\alpha},\vec{r}) &= \tfrac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{n}\alpha_k\alpha_j y_c^k y_c^j x_{ki}x_{ji} - \sum_{i=1}^{n}\alpha_i y_c^i\sum_{j=1}^{m} w_j x_{ij} \\ &-b\sum_{i=1}^{n}\alpha_i y_c^i + Db + \sum_{i=1}^{n}\alpha_i(1+\varphi_i) \end{aligned} \tag{D.125}$$

$$\begin{aligned} d(\vec{\alpha},\vec{r}) &= \tfrac{1}{2}\sum_{j=1}^{n}\sum_{k=1}^{n}\alpha_k\alpha_j y_c^k y_c^j\sum_{i=1}^{m} x_{ji}x_{ki} \\ &- \sum_{i=1}^{n}\alpha_i y_c^i\sum_{j=1}^{m} x_{ij}\sum_{k=1}^{n}\alpha_k y_c^k x_{kj} + \sum_{i=1}^{n}\alpha_i(1+\varphi_i) \end{aligned} \tag{D.126}$$

$$\begin{aligned} d(\vec{\alpha},\vec{r}) &= \tfrac{1}{2}\sum_{j=1}^{n}\sum_{k=1}^{n}\alpha_k\alpha_j y_c^k y_c^j\sum_{i=1}^{m} x_{ji}x_{ki} \\ &- \sum_{i=1}^{n}\sum_{k=1}^{n}\alpha_k\alpha_i y_c^k y_c^i\sum_{j=1}^{m} x_{ij}x_{kj} + \sum_{i=1}^{n}\alpha_i(1+\varphi_i) \end{aligned} \tag{D.127}$$

$$d(\vec{\alpha},\vec{r}) = -\frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{n}\alpha_k\alpha_i y_c^k y_c^i\sum_{j=1}^{m} x_{ij}x_{kj} + \sum_{i=1}^{n}\alpha_i(1+\varphi_i)\ . \tag{D.128}$$

The dual form is

OP 32.

$$\max_{\vec{\alpha},\vec{r}}\quad d(\vec{\alpha},\vec{r}) = \sum_{i=1}^{n}\alpha_i(1+\varphi_i) - \frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{n}\alpha_k\alpha_i y_c^k y_c^i\sum_{j=1}^{m} x_{ij}x_{kj} \tag{D.129}$$

subject to

$$\sum_{i=1}^{n}\alpha_i y_c^i = D \tag{D.130}$$

$$C_i = r_i + \alpha_i \tag{D.131}$$

$$\alpha_i \geq 0 \tag{D.132}$$

$$r_i \geq 0 \tag{D.133}$$

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

D.8 Derivation of OP 20

OP 33.

$$\max_{\vec{\alpha},\vec{r}}\quad d(\vec{\alpha},\vec{r}) \tag{D.134}$$

where

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

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

$$\frac{\partial t\left(\vec{w}, b, \vec{\alpha}, \vec{\xi}, \vec{r}\right)}{\partial w_i} = w_i - \sum_{j=1}^{n} \alpha_j y_{\mathrm{c}}^j x_{ji} = 0 \tag{D.135}$$

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

$$\frac{\partial t\left(\vec{w}, b, \vec{\alpha}, \vec{\xi}, \vec{r}\right)}{\partial b} = \sum_{i=1}^{n} \alpha_i y_{\mathrm{c}}^i = D \ . \tag{D.136}$$

A partial derivative with respect to ξ_i is

$$\frac{\partial t\left(\vec{w}, b, \vec{\alpha}, \vec{\xi}, \vec{r}\right)}{\partial \xi_i} = C_i - r_i - \alpha_i = 0 \ . \tag{D.137}$$

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

$$\begin{aligned} d\left(\vec{\alpha}, \vec{r}\right) = & \tfrac{1}{2} \sum_{i=1}^{m} \left(\sum_{j=1}^{n} \alpha_j y_{\mathrm{c}}^j x_{ji} \right) \left(\sum_{k=1}^{n} \alpha_k y_{\mathrm{c}}^k x_{ki} \right) \\ & - \sum_{i=1}^{n} \alpha_i y_{\mathrm{c}}^i \left(\sum_{j=1}^{m} w_j x_{ij} + b \right) + \sum_{i=1}^{n} \alpha_i \left(1 + \varphi_i\right) + \sum_{i=1}^{n} C_i \xi_i + Db \\ & - \sum_{i=1}^{n} \alpha_i \xi_i - \sum_{i=1}^{n} r_i \xi_i \end{aligned} \tag{D.138}$$

$$\begin{aligned} d\left(\vec{\alpha}, \vec{r}\right) = & \tfrac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{n} \alpha_k \alpha_j y_{\mathrm{c}}^k y_{\mathrm{c}}^j x_{ki} x_{ji} - \sum_{i=1}^{n} \alpha_i y_{\mathrm{c}}^i \sum_{j=1}^{m} w_j x_{ij} \\ & - b \sum_{i=1}^{n} \alpha_i y_{\mathrm{c}}^i + Db + \sum_{i=1}^{n} \alpha_i \left(1 + \varphi_i\right) \end{aligned} \tag{D.139}$$

$$\begin{aligned} d\left(\vec{\alpha}, \vec{r}\right) = & \tfrac{1}{2} \sum_{j=1}^{n} \sum_{k=1}^{n} \alpha_k \alpha_j y_{\mathrm{c}}^k y_{\mathrm{c}}^j \sum_{i=1}^{m} x_{ji} x_{ki} \\ & - \sum_{i=1}^{n} \alpha_i y_{\mathrm{c}}^i \sum_{j=1}^{m} x_{ij} \sum_{k=1}^{n} \alpha_k y_{\mathrm{c}}^k x_{kj} + \sum_{i=1}^{n} \alpha_i \left(1 + \varphi_i\right) \end{aligned} \tag{D.140}$$

$$\begin{aligned} d\left(\vec{\alpha}, \vec{r}\right) = & \tfrac{1}{2} \sum_{j=1}^{n} \sum_{k=1}^{n} \alpha_k \alpha_j y_{\mathrm{c}}^k y_{\mathrm{c}}^j \sum_{i=1}^{m} x_{ji} x_{ki} \\ & - \sum_{i=1}^{n} \sum_{k=1}^{n} \alpha_k \alpha_i y_{\mathrm{c}}^k y_{\mathrm{c}}^i \sum_{j=1}^{m} x_{ij} x_{kj} + \sum_{i=1}^{n} \alpha_i \left(1 + \varphi_i\right) \end{aligned} \tag{D.141}$$

$$d\left(\vec{\alpha}, \vec{r}\right) = -\frac{1}{2} \sum_{i=1}^{n} \sum_{k=1}^{n} \alpha_k \alpha_i y_{\mathrm{c}}^k y_{\mathrm{c}}^i \sum_{j=1}^{m} x_{ij} x_{kj} + \sum_{i=1}^{n} \alpha_i \left(1 + \varphi_i\right) \ . \tag{D.142}$$

The dual form is

OP 34.

$$\max_{\vec{\alpha}, \vec{r}} \quad d\left(\vec{\alpha}, \vec{r}\right) = \sum_{i=1}^{n} \alpha_i \left(1 + \varphi_i\right) - \frac{1}{2} \sum_{i=1}^{n} \sum_{k=1}^{n} \alpha_k \alpha_i y_{\mathrm{c}}^k y_{\mathrm{c}}^i \sum_{j=1}^{m} x_{ij} x_{kj} \tag{D.143}$$

subject to

$$\sum_{i=1}^{n} \alpha_i y_{\mathrm{c}}^i = D \tag{D.144}$$

$$C_i = r_i + \alpha_i \tag{D.145}$$

$$\alpha_i \geq 0 \tag{D.146}$$

$$r_i \geq 0 \tag{D.147}$$

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

Appendix E

Applications: Order Execution Strategies

E.1 Proof of Thm. V.2.1

Proof. The proof is

$$VWAP_0 = \frac{\sum_{i=1}^{m} \left(VWAP\left(T_i\right) + \varepsilon_1\left(T_i\right)\right)\left(v_0\left(r\left(T_i\right) + \varepsilon_2\left(T_i\right)\right)\right)}{v_0} \tag{E.1}$$

$$VWAP_0 = \sum_{i=1}^{m} \left(VWAP\left(T_i\right) + \varepsilon_1\left(T_i\right)\right)\left(r\left(T_i\right) + \varepsilon_2\left(T_i\right)\right) \tag{E.2}$$

$$\frac{VWAP_0}{VWAP} - 1 = \frac{v\sum_{i=1}^{m} \left(VWAP\left(T_i\right) + \varepsilon_1\left(T_i\right)\right)\left(r\left(T_i\right) + \varepsilon_2\left(T_i\right)\right)}{\sum_{i=1}^{m} VWAP\left(T_i\right) v\left(T_i\right)} - 1 = \tag{E.3}$$

$$= \frac{\sum_{i=1}^{m} \left(VWAP\left(T_i\right) + \varepsilon_1\left(T_i\right)\right)\left(r\left(T_i\right) + \varepsilon_2\left(T_i\right)\right)}{\sum_{i=1}^{m} VWAP\left(T_i\right) r\left(T_i\right)} - 1 = \tag{E.4}$$

$$= \frac{\sum_{i=1}^{m} \varepsilon_1\left(T_i\right) r\left(T_i\right)}{\sum_{i=1}^{m} VWAP\left(T_i\right) r\left(T_i\right)} + \frac{\sum_{i=1}^{m} \varepsilon_2\left(T_i\right) VWAP\left(T_i\right)}{\sum_{i=1}^{m} VWAP\left(T_i\right) r\left(T_i\right)} \tag{E.5}$$

$$+ \frac{\sum_{i=1}^{m} \varepsilon_1\left(T_i\right) \varepsilon_2\left(T_i\right)}{\sum_{i=1}^{m} VWAP\left(T_i\right) r\left(T_i\right)} \ . \tag{E.6}$$

□

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. 4, 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. 22, 32, 43, 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 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, 3

[6] 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. 1, 3

[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. 11

[8] L. Hamel. *Knowledge discovery with support vector machines*. Wiley series on methods and applications in data mining. John Wiley & Sons, 2009. ISBN 9780470371923. 54

[9] Ralf Herbrich. *Learning Kernel Classifiers: Theory and Algorithms*. MIT Press, Cambridge, MA, USA, 2001. ISBN 026208306X. 3

[10] D. Hobson. Vwap and volume profiles. *Journal of Trading*, 1(2):38–42, 2006. 61

[11] Te-Ming Huang, Vojislav Kecman, and Ivica Kopriva. *Kernel Based Algorithms for Mining Huge Data Sets*. Springer, 2006. 1

[12] Nitin Indurkhya and Sholom M. Weiss. Solving regression problems with rule-based ensemble classifiers. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '01, pages 287–292, New York, NY, USA, 2001. ACM. ISBN 1-58113-391-X. 11

[13] T. Joachims. Making large-scale support vector machine learning practical, 1998. 53, 54, 57

[14] Thorsten Joachims. *Learning to Classify Text Using Support Vector Machines*. Kluwer Academic Publishers, 2002. 1, 9

[15] 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. 41

[16] 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. ISBN 978-3-540-85566-8. 41

[17] Vojislav Kecman and Tao Yang. Adaptive local hyperplane for regression tasks. In *Proceedings of the 2009 international joint conference on Neural Networks*, IJCNN'09, pages 2371–2375, Piscataway, NJ, USA, 2009. IEEE Press. ISBN 978-1-4244-3549-4. 1, 11

[18] S. Sathiya Keerthi, Shirish Krishnaj Shevade, Chiranjib Bhattacharyya, and K. R. K. Murthy. Improvements to platt's smo algorithm for svm classifier design. *Neural Computation*, 13(3):637–649, 2001. 54

[19] 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. ISSN 1532-4435. 41

[20] 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. 3

[21] Fabien Lauer and Gérard Bloch. Incorporating prior knowledge in support vector machines for classification: A review. *Neurocomputing*, 71(7-9):1578–1594, 2008. ISSN 0925-2312. 1, 9

[22] Fabien Lauer and Gérard Bloch. Incorporating prior knowledge in support vector regression. *Mach. Learn.*, 70:89–118, January 2008. ISSN 0885-6125. 1, 9

[23] 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

[24] LibSVM data sets. Libsvm data sets. www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/, 06 2011. 22, 43

[25] Chun-Fu Lin and Sheng-De Wang. Fuzzy support vector machines. *IEEE Transaction on Neural Networks*, 13(2):464–471, 2002. 9

[26] 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, 11

[27] Olvi L. Mangasarian and Edward W. Wild. Nonlinear knowledge-based classification. *IEEE Transactions on Neural Networks*, 19(10):1826–1832, 2008. 1, 3

[28] Olvi L. Mangasarian, Jude W. Shavlik, and Edward W. Wild. Knowledge-based kernel approximation. *Journal of Machine Learning Research*, 5:1127–1141, 2004. 42

[29] Kuan ming Lin and Chih jen Lin. A study on reduced support vector machines. *IEEE TRANSACTIONS ON NEURAL NETWORKS*, 14:1449–1459, 2003. 41

[30] Tom M. Mitchell. *Machine Learning*. McGraw-Hill, New York, 1997. 3

[31] 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. ISBN 978-1-4244-1514-4. doi: 10.1109/SPA.2007.5903314. URL http://dx.doi.org/10.1109/SPA.2007.5903314. 53, 54, 55, 57

[32] Marcin Orchel. Support vector machines: Heuristic of alternatives. In Ryszard Romaniuk, editor, *Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments 2007*, volume 6937, page 69373E. SPIE, December 2007. doi: 10.1117/12.784837. URL http://dx.doi.org/10.1117/12.784837. 53, 57

[33] Marcin Orchel. Incorporating detractors into svm classification. In Krzysztof Cyran, Stanislaw Kozielski, James Peters, Urszula Stańczyk, and Alicja Wakulicz-Deja, editors, *Man-Machine Interactions*, volume 59 of *Advances in Intelligent and Soft Computing*, pages 361–369. Springer Berlin / Heidelberg, 2009. ISBN 978-3-642-00562-6. doi: 10.1007/978-3-642-00563-3_38. URL http://dx.doi.org/10.1007/978-3-642-00563-3_38. 2, 11, 29, 61, 66

[34] Marcin Orchel. Paper id 55. In *Submitted to European Conference of Machine Learning, ECML 2010*, April 2010. 1, 11

[35] 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. ISBN 978-3-642-20266-7. doi: 10.1007/978-3-642-20267-4_35. URL http://dx.doi.org/10.1007/978-3-642-20267-4_35. 2, 29, 41, 61, 66

[36] Marcin Orchel. Regression based on support vector classification. In Andrej Dobnikar, Uroš Lotric, and Branko Šter, editors, *Adaptive and Natural Computing Algorithms*, volume 6594 of *Lecture Notes in Computer Science*, pages 353–362. Springer Berlin / Heidelberg, 2011. ISBN 978-3-642-20266-7. doi: 10.1007/978-3-642-20267-4_37. URL http://dx.doi.org/10.1007/978-3-642-20267-4_37. 1, 11, 12, 14, 66

[37] Marcin Orchel. Support vector regression as a classification problem with a priori knowledge in the form of detractors. In Tadeusz Czachorski, Stanislaw Kozielski, and Urszula Stańczyk, editors, *Man-Machine Interactions 2*, volume 103 of *Advances in Intelligent and Soft Computing*, pages 353–362. Springer Berlin / Heidelberg, 2011. ISBN 978-3-642-23168-1. doi: 10.1007/978-3-642-23169-8_38. URL http://dx.doi.org/10.1007/978-3-642-23169-8_38. 2, 3, 11, 29, 32, 41, 61, 66

[38] 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. ISBN 978-3-642-25855-8. doi: 10.1007/978-3-642-25856-5_24. URL http://dx.doi.org/10.1007/978-3-642-25856-5_24. 4, 61

[39] Marcin Orchel. Support vector regression based on data shifting. *Neurocomputing*, 96:2–11, 2012. 11, 25

[40] John C. Platt. *Fast training of support vector machines using sequential minimal optimization*, pages 185–208. MIT Press, Cambridge, MA, USA, 1999. ISBN 0-262-19416-3. 31, 53

[41] Jean-Baptiste Pothin and Cedric Richard. Incorporating prior information into support vector machines in the form of ellipsoidal knowledge sets, 2006. 3

[42] Manel Martinez Ramon and Christos Christodoulou. *Support Vector Machines for Antenna Array Processing and Electromagnetics*. Margan & Claypool Publishers, 2006. 1

[43] Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001. ISBN 0262194759. 22, 54

[44] Catarina Silva and Bernardete Ribeiro. *Inductive Inference for Large Scale Text Classification*. Springer, 2010. 1

[45] Ingo Steinwart, Don R. Hush, and Clint Scovel. Training svms without offset. *Journal of Machine Learning Research*, 12:141–202, 2011. 7

[46] Rangarajan Sundaram. *A First Course in Optimization Theory*. Cambridge University Press, 1996. 71

[47] Francis Eng Hock Tay and Lijuan Cao. Modified support vector machines in financial time series forecasting. *Neurocomputing*, 48(1-4):847–861, 2002. 9

[48] R. J. Vanderbei. LOQO: An interior point code for quadratic programming. *Optimization Methods and Software*, 11:451–484, 1999. 57

[49] Vladimir N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995. ISBN 0-387-94559-8. 1, 11, 18

[50] Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, September 1998. ISBN 0471030031. 1, 9, 11, 18, 21, 75

[51] 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. ISBN 0-7695-2128-2. 1, 9

[52] 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

[53] Edward W. Wild. *Optimization-based Machine Learning and Data Mining*. PhD thesis, University of Wisconsin-Madison, 2008. 3

[54] Chang-An Wu and Hong-Bing Liu. An improved support vector regression based on classification. In *Proceedings of the 2007 International Conference on Multimedia and Ubiquitous Engineering*, MUE '07, pages 999–1003, Washington, DC, USA, 2007. IEEE Computer Society. ISBN 0-7695-2777-9. 11

[55] 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. 41

[56] 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. ISBN 1-58113-888-1. 9, 29

[57] 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. 23

[58] L. Yu, S. Wang, K.K. Lai, and L. Zhou. *Bio-inspired credit risk analysis: computational intelligence with support vector machines.* Springer Verlag, 2008. ISBN 9783540778028. 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,

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

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

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

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

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

Optimization theory

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

Order Execution Strategies

$VWAP$ 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-SVM ν support vector machines,

ε-SVR ε-insensitive support vector regression,

φ-SVC φ support vector classification,

bSVC free term support vector classification,

bSVM free term support vector machines,

C-SVM C support vector machines,

CM capacity minimization,

DJIA Dow Jones Industrial Average,

EMS execution management system,

ERM empirical risk minimization,

HoA heuristic of alternatives,

KKT Karush-Kuhn-Tucker,

MSE mean squared error,

NASDAQ National Association of Securities Dealers Automated Quotations,

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,

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,