Solving SVM by Decomposition

Marcin Orchel

Abstract

In this report, we propose two implementation improvements, the first one for speed of training of support vector machines (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 sequential minimal optimization (SMO) method. We achieve simpler implementation with similar speed performance.

Contents

Chapter 1

Introduction

1.1 Support Vector Classification Basics

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

OP 1.

$$\min_{\vec{w_c}, b_c} \quad f\left(\vec{w_c}, b_c\right) = \left\|\vec{w_c}\right\|^2 \tag{1.1}$$

subject to

$$y_c^i h\left(\vec{x_i}\right) \geq 1 \tag{1.2}$$

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

$$h\left(\vec{x_i}\right) = \vec{w_c} \cdot \vec{x_i} + b_c \quad . \tag{1.3}$$

All points must be correctly classified Fig. 1.1a. The SVC soft margin case optimization problem with $\|\cdot\|_1$ norm is

OP 2.

$$\min_{\vec{w_c}, b_c, \vec{\xi_c}} \quad f\left(\vec{w_c}, b_c, \vec{\xi_c}\right) = \frac{1}{2}\left\|\vec{w_c}\right\|^2 + C_c \sum_{i=1}^{n} \xi_c^i \tag{1.4}$$

subject to

$$y_c^i h\left(\vec{x_i}\right) \geq 1 - \xi_c^i \tag{1.5}$$

$$\vec{\xi_c} \geq 0 \tag{1.6}$$

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

$$h\left(\vec{x_i}\right) = \vec{w_c} \cdot \vec{x_i} + b_c \quad , \tag{1.7}$$

$$C_c > 0 \quad . \tag{1.8}$$

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. 1.1b.

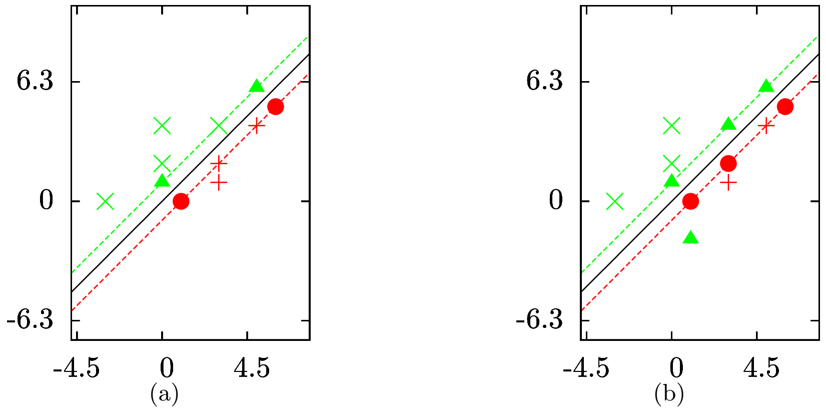


Figure 1.1: Two types of margin classifiers: hard and soft. Example points, support vectors (triangles and circles), solutions (solid lines), margin lines (dashed lines). (a) Hard. (b) Soft. A misclassified point is in (1, -2)

SVC Dual Optimization Problem

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

OP 3.

$$\max_{\vec{\alpha}} \quad f\left(\vec{\alpha}\right) = 1 \cdot \vec{\alpha} - \frac{1}{2} \vec{\alpha}^T \mathbf{Q} \vec{\alpha} \tag{1.9}$$

subject to

$$\vec{\alpha} \cdot \vec{y} = 0 \tag{1.10}$$

$$0 \leq \alpha_i \leq C_c \tag{1.11}$$

where

$$Q_{ij} = y_i y_j K\left(\vec{x_i}, \vec{x_j}\right) \tag{1.12}$$

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

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

$$\vec{w_c^*} \cdot \vec{x} = \sum_{i=1}^{n} y_{\mathrm{c}}^i \alpha_i^* K\left(\vec{x_i}, \vec{x}\right) \ . \tag{1.13}$$

Therefore, the decision curve is

$$h^*\left(\vec{x}\right) = \sum_{i=1}^{n} y_{\mathrm{c}}^i \alpha_i^* K\left(\vec{x_i}, \vec{x}\right) + b_{\mathrm{c}}^* = 0 \ , \tag{1.14}$$

where α_i are Lagrange multipliers of the dual problem, $K\left(\cdot, \cdot\right)$ is a kernel function, which appears only in the dual problem. *Margin boundaries* are defined as the two hyperplanes

$h(\vec{x}) = -1$ and $h(\vec{x}) = 1$. *Optimal margin boundaries* are defined as the two hyperplanes $h^*(\vec{x}) = -1$ and $h^*(\vec{x}) = 1$. *Geometric margin of the hyperplane* h is defined as $1/\|\vec{w_c}\|$. The i-th training example is *a support vector*, when $\alpha_i^* \neq 0$. A set of support vectors contains all training examples lying below optimal margin boundaries ($y_{\mathrm{c}}^i h^*(\vec{x_i}) < 1$), and part of the examples lying exactly on the optimal margin boundaries ($y_{\mathrm{c}}^i h^*(\vec{x_i}) = 1$), Fig. 1.1b.

ν-SVC

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

OP 4.

$$\min_{\vec{w},b,\vec{\xi},p} \quad f\left(\vec{w}, b, \vec{\xi}, p\right) = \frac{1}{2}\|\vec{w}\|^2 - \nu p + \frac{1}{n}\sum_{i=1}^{n} \xi_c^i \tag{1.15}$$

subject to

$$y_i h(\vec{x_i}) \geq p - \xi_i \tag{1.16}$$

$$\vec{\xi} \geq 0 \tag{1.17}$$

$$p \geq 0 \tag{1.18}$$

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

$$h(\vec{x_i}) = \vec{w} \cdot \vec{x_i} + b \ . \tag{1.19}$$

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

1.2 Support Vector Regression Basics

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

OP 5.

$$\min_{\vec{w_{\mathrm{r}}},b_{\mathrm{r}},\vec{\xi_{\mathrm{r}}},\vec{\xi_{\mathrm{r}}^*}} \quad f\left(\vec{w_{\mathrm{r}}}, b_{\mathrm{r}}, \vec{\xi_{\mathrm{r}}}, \vec{\xi_{\mathrm{r}}^*}\right) = \frac{1}{2}\|\vec{w_{\mathrm{r}}}\|^2 + C_{\mathrm{r}}\sum_{i=1}^{n}\left(\xi_{\mathrm{r}}^i + \xi_{\mathrm{r}}^{*i}\right) \tag{1.20}$$

subject to

$$y_{\mathrm{r}}^i - g(\vec{x_i}) \leq \varepsilon + \xi_{\mathrm{r}}^i \tag{1.21}$$

$$g(\vec{x_i}) - y_{\mathrm{r}}^i \leq \varepsilon + \xi_{\mathrm{r}}^{i*} \tag{1.22}$$

$$\vec{\xi_{\mathrm{r}}} \geq 0 \tag{1.23}$$

$$\vec{\xi_{\mathrm{r}}^*} \geq 0 \tag{1.24}$$

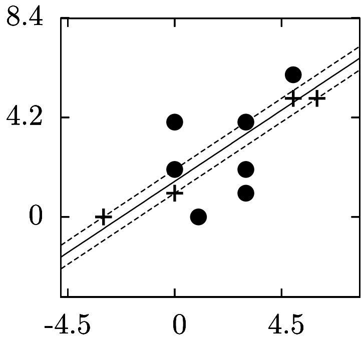


Figure 1.2: The idea of ε-SVR. Points - examples, circles - support vectors, a solid line - a solution, dashed lines - ε boundaries

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

$$g(\vec{x_i}) = \vec{w_\mathrm{r}} \cdot \vec{x_i} + b_\mathrm{r} \quad , \tag{1.25}$$

$$\varepsilon \in \mathbb{R} \quad . \tag{1.26}$$

The $g^*(\vec{x}) = \vec{w_\mathrm{r}^*} \cdot \vec{x} + b_\mathrm{r}^*$ is a regression function. Optimization problem 5 is transformed into an equivalent dual problem. The regression function becomes

$$g^*(\vec{x}) = \sum_{i=1}^{n} (\alpha_i^* - \beta_i^*) K(\vec{x_i}, \vec{x}) + b_\mathrm{r}^* \quad , \tag{1.27}$$

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

1.3 SVM with C_i Weights

A 1-norm soft margin SVC optimization problem for training examples $\vec{x_i}$ with weights C_i is

OP 6.

$$\min_{\vec{w}, b, \vec{\xi}} f\left(\vec{w}, b, \vec{\xi}\right) = \frac{1}{2}\|\vec{w}\|^2 + \vec{C_c} \cdot \vec{\xi} \tag{1.28}$$

subject to

$$y_i h(\vec{x_i}) \geq 1 - \xi_i \tag{1.29}$$

$$\vec{\xi} \geq 0 \tag{1.30}$$

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

$$\vec{C_c} \gg 0 \tag{1.31}$$

$$h\left(\vec{x_i}\right) = \vec{w} \cdot \vec{x_i} + b \ . \tag{1.32}$$

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

Definition 1.3.1. Given some curve $h\left(\vec{x}\right) = 0$, *the margin of the* $\vec{x_p}$ *example* is defined as a value $\left|h\left(\vec{x_p}\right)\right|$.

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.

1.4 Introduction to φ-SVC

The φ support vector classification (φ-SVC) method is a recently proposed method of incorporating knowledge about the margin of an example to SVC, [6, 7, 8]. The φ-SVC optimization problem is defined with an additional parameter per example added in the right side of the inequality (1.5). Another modification of inequality constraints was proposed in [12]. 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 6 as

OP 7.

$$\min_{\vec{w_c}, b_c, \vec{\xi_c}} \quad f\left(\vec{w_c}, b_c, \vec{\xi_c}\right) = \frac{1}{2}\left\|\vec{w_c}\right\|^2 + \vec{C_c} \cdot \vec{\xi_c} \tag{1.33}$$

subject to

$$y_c^i h\left(\vec{x_i}\right) \geq 1 + \varphi_i - \xi_c^i \tag{1.34}$$

$$\vec{\xi_c} \geq 0 \tag{1.35}$$

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

$$\vec{C_c} \gg 0 \tag{1.36}$$

$$\varphi_i \in \mathbb{R} \tag{1.37}$$

$$h\left(\vec{x_i}\right) = \vec{w_c} \cdot \vec{x_i} + b_c \ . \tag{1.38}$$

The new weights φ_i are present only in (1.34). When $\vec{\varphi} = 0$, the OP 7 is equivalent to OP 6. When all φ_i are equal to some constant $\varphi > -1$, we will get the same decision boundary as for $\varphi = 0$ when we change $\vec{C_c}$ to $\vec{C_c} / \left(1 + \varphi\right)$.

Proof. We will prove that we get the same decision boundary when we replace 1 with $1/d$, for some $d > 0$, $\varphi_i = 0$. Let's replace ξ_i with ξ_i/d and we get the inequalities $y_c^i h\left(\vec{x_i}\right) \geq 1/d - \xi_c^i/d$. After multiplying by d we get $y_c^i d h\left(\vec{x_i}\right) \geq 1 - \xi_c^i$. The objective function can be multiplied by d^2 and we get $\frac{1}{2}\left\|d\vec{w_{\mathrm{c}}}\right\|^2 + d\vec{C_{\mathrm{c}}} \cdot \vec{\xi_{\mathrm{c}}}$, 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 $dh\left(\vec{x}\right) = 0$. □

We can notice that when neglecting ξ^i_{c}, we get different bounds for the margin: when $y^i_{\mathrm{c}} = 1$ and $g\left(\vec{x_i}\right) \geq 0$, then we get a lower bound $1 + \varphi_i$, when $y^i_{\mathrm{c}} = -1$ and $g\left(\vec{x_i}\right) \geq 0$, then we get an upper bound $-\left(1 + \varphi_i\right)$, when $y^i_{\mathrm{c}} = 1$ and $g\left(\vec{x_i}\right) < 0$, then we get an upper bound $-\left(1 + \varphi_i\right)$, when $y^i_{\mathrm{c}} = -1$ and $g\left(\vec{x_i}\right) < 0$, 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 $-\left(1 + \varphi_i\right)$, 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 $h\left(\vec{x}\right) = 0$ and the i-th example. We can conduct the same analysis as above for distances by dividing both sides of (1.34) by $\left\|\vec{w_{\mathrm{c}}}\right\|$, so we get $y^i_{\mathrm{c}} h\left(\vec{x_i}\right) / \left\|\vec{w_{\mathrm{c}}}\right\| \geq 1/\left\|\vec{w_{\mathrm{c}}}\right\| + \varphi_i/\left\|\vec{w_{\mathrm{c}}}\right\|$. The conclusion is similar: for the case when $1 + \varphi_i > 0$, we get a lower bound on the distance equal to $\left(1 + \varphi_i\right)/\left\|\vec{w_{\mathrm{c}}}\right\|$. For the case when $1 + \varphi_i < 0$, we get an upper bound on the distance equal to $-\left(1 + \varphi_i\right)/\left\|\vec{w_{\mathrm{c}}}\right\|$. 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^i_{c}, 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^i_{c} 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 7, where φ_i weights are present in the constraints with slack variables

OP 8.

$$\min_{\vec{w_{\mathrm{c}}}, b_{\mathrm{c}}, \vec{\xi_{\mathrm{c}}}} \quad f\left(\vec{w_{\mathrm{c}}}, b_{\mathrm{c}}, \vec{\xi_{\mathrm{c}}}\right) = \frac{1}{2}\left\|\vec{w_{\mathrm{c}}}\right\|^2 + \vec{C_{\mathrm{c}}} \cdot \vec{\xi_{\mathrm{c}}} \tag{1.39}$$

subject to

$$y^i_{\mathrm{c}} h\left(\vec{x_i}\right) \geq 1 - \xi^i_{\mathrm{c}} \tag{1.40}$$

$$\vec{\xi} \geq \varphi_i \tag{1.41}$$

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

In order to construct an efficient algorithm for the OP 7 its dual form was derived The final form of the dual problem is

OP 9.

$$\max_{\vec{\alpha}} \quad d\left(\vec{\alpha}\right) = \vec{\alpha} \cdot \left(1 + \vec{\varphi}\right) - \frac{1}{2}\vec{\alpha}^T Q \vec{\alpha} \tag{1.42}$$

subject to

$$\vec{\alpha} \cdot \vec{y_{\mathrm{c}}} = 0 \tag{1.43}$$

$$0 \leq \vec{\alpha} \leq \vec{C} \ \ , \tag{1.44}$$

where

$$Q_{ij} = y_{\mathrm{c}}^i y_{\mathrm{c}}^j \left(\vec{x}_i \cdot \vec{x}_j\right) \tag{1.45}$$

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

$$h^* \left(\vec{x}\right) = \sum_{i=1}^{n} y_{\mathrm{c}}^i \alpha_i^* K \left(\vec{x}_i, \vec{x}\right) + b^* = 0 \ \ , \tag{1.46}$$

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

$$\alpha_i \left(y_{\mathrm{c}}^i h\left(\vec{x}_i\right) - 1 - \varphi_i + \xi_{\mathrm{c}}^i\right) = 0 \tag{1.47}$$

$$\left(C_i - \alpha_i\right) \xi_{\mathrm{c}}^i = 0 \ \ . \tag{1.48}$$

The conclusions from (1.47) and (1.48) 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 $y_{\mathrm{c}}^i h\left(\vec{x}_i\right) = 1 + \varphi_i$, when $\alpha_i = C_{\mathrm{c}}$, then $y_{\mathrm{c}}^i h\left(\vec{x}_i\right) = 1 + \varphi_i - \xi_{\mathrm{c}}^i$. Moreover, when $\xi_{\mathrm{c}}^i > 0$, then $\alpha_i = C_{\mathrm{c}}$ and $y_{\mathrm{c}}^i h\left(\vec{x}_i\right) = 1 + \varphi_i - \xi_{\mathrm{c}}^i$, when $y_{\mathrm{c}}^i h\left(\vec{x}_i\right) > 1 + \varphi_i - \xi_{\mathrm{c}}^i$, then $\alpha_i = 0$, $\xi_{\mathrm{c}}^i = 0$ and $y_{\mathrm{c}}^i h\left(\vec{x}_i\right) > 1 + \varphi_i$. We can find ξ_{c}^i parameters from the solution of the dual form as following. When

$$y_{\mathrm{c}}^i h^* \left(\vec{x}_i\right) \geq 1 + \varphi_i \ \ , \tag{1.49}$$

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

$$\xi_{\mathrm{c}}^i = 1 + \varphi_i - y_{\mathrm{c}}^i h\left(\vec{x}_i\right) \ \ . \tag{1.50}$$

1.5 Solving φ-SVC Optimization Problem

For solving OP 9, a decomposition method similar to SMO, [9] was derived. In every step, two parameter subproblems are solved. Heuristic and stopping criterion are based on KKT conditions.

Chapter 2

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 [9], which solves 2-parameter subproblems analytically in every iteration. For subproblems with more than 2 parameters, general quadratic programming solvers are used, [2]. We proposed using SMO for solving subproblems with more than 2 parameters, [5]. 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, [4]. The advantage of this strategy is the decreased number of iterations.

We can use all proposed methods with SVC, and therefore also with δ support vector regression (δ-SVR). They also work with φ-SVC, so we can use them with ε-SVR as well.

2.1 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 10.

$$\begin{aligned}\max_{\vec{\beta}} \quad f_2\left(\vec{\beta}\right) = \textstyle\sum_{i=1}^{p} \beta_i \left(1 + \varphi_{c_i}\right) + \textstyle\sum_{\substack{i=1\\ i\notin P}}^{n} \alpha_i \left(1 + \varphi_i\right) - \tfrac{1}{2} \textstyle\sum_{i=1}^{p} y_{c_i}\beta_i \textstyle\sum_{j=1}^{p} y_{c_j}\beta_j K_{c_i c_j} \\ - \textstyle\sum_{i=1}^{p} y_{c_i}\beta_i \textstyle\sum_{\substack{j=1\\ j\notin P}}^{n} y_j\alpha_j K_{c_i j} - \tfrac{1}{2} \textstyle\sum_{\substack{i=1\\ i\notin P}}^{n} \textstyle\sum_{\substack{j=1\\ j\notin P}}^{n} y_{ij}\alpha_i\alpha_j K_{ij}\end{aligned} \tag{2.1}$$

subject to

$$\sum_{i=1}^{p} y_{c_i}\beta_i + \sum_{\substack{i=1\\ i\notin P}}^{n} y_i\alpha_i = 0 \tag{2.2}$$

$$0 \leq \beta_i \leq C_{c_i} \tag{2.3}$$

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 9. It must satisfy the linear constraint (1.43).

After solving OP 10, 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 (1.43).

2.2 Introduction to SMO for φ-SVC

The SMO is a well established method for solving SVC described in [10, 1]. 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 10 when $p = 2$. We can find directly the solution for this case, before clipping it is

$$\beta_2^{\text{unc}} = \alpha_{c_2} + \frac{y_{c_2}\left(E_{c_1} - E_{c_2}\right)}{\kappa} \tag{2.4}$$

where

$$E_i = \sum_{j=1}^{n} y_j \alpha_j K\left(\vec{x_i}, \vec{x_j}\right) - y_i - y_i \varphi_i \quad , \tag{2.5}$$

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

$$\kappa = K\left(\vec{x_{c_1}}, \vec{x_{c_1}}\right) + K\left(\vec{x_{c_2}}, \vec{x_{c_2}}\right) - 2K\left(\vec{x_{c_1}}, \vec{x_{c_2}}\right) \quad . \tag{2.6}$$

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

$$\beta_2 = \begin{cases} V, \ if \ \beta_2^{\text{unc}} > V \\ \beta_2^{\text{unc}}, \ if \ U \leq \beta_2^{\text{unc}} \leq V \\ U, \ if \ \beta_2^{\text{unc}} < U \end{cases} \tag{2.7}$$

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

$$U = \max\left(0, \alpha_{c_2} - \alpha_{c_1}\right) \quad , \tag{2.8}$$

$$V = \min\left(C_{c_2}, C_{c_1} - \alpha_{c_1} + \alpha_{c_2}\right) \quad , \tag{2.9}$$

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

$$U = \max\left(0, \alpha_{c_1} + \alpha_{c_2} - C_{c_1}\right) \quad , \tag{2.10}$$

$$V = \min\left(C_2, \alpha_{c_1} + \alpha_{c_2}\right) \quad . \tag{2.11}$$

A value of the first variable is

$$\beta_1 = \alpha_{c_1} + y_{c_1} y_{c_2}\left(\alpha_{c_2} - \beta_2\right) \quad . \tag{2.12}$$

2.2.1 SMO for SVM without the offset

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

$$\beta_1^{\mathrm{unc}} = \alpha_{c_1} - \frac{y_{c_1} E_{c_1}}{K\left(\vec{x_{c_1}}, \vec{x_{c_1}}\right)} \quad . \tag{2.13}$$

Then we have to bound β_1^{unc} to

$$0 \leq \beta_1^{\mathrm{unc}} \leq C_{c_1} \quad . \tag{2.14}$$

2.3 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, [3]. The multivariable heuristic based on Zoutendijk's method was proposed in [2]. We proposed a simple strategy for multivariable heuristic based on violation of KKT conditions, [5]. 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 2.3.1. *Optimization is possible for reduced optimization problem OP 10 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*

$$\begin{gathered} G_1 := \{i \in \{1, 2, \ldots, n\} : (y_i = 1 \wedge \alpha_i = 0) \\ \vee (y_i = -1 \wedge \alpha_i = C_i) \vee (0 < \alpha_i < C_i)\} \\ G_2 := \{i \in \{1, 2, \ldots, n\} : (y_i = -1 \wedge \alpha_i = 0) \\ \vee (y_i = 1 \wedge \alpha_i = C_i) \vee (0 < \alpha_i < C_i)\} \quad . \end{gathered} \tag{2.15}$$

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 A.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.

2.3.1 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. 2.3.1 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

$$m_{c_d c_k} := E_{c_d} - E_{c_k} \quad , \tag{2.16}$$

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

$$m_{c_d c_k} := E_{c_k} - E_{c_d} \quad , \tag{2.17}$$

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

$$m_{c_d c_k} := |E_{c_k} - E_{c_d}| \quad . \tag{2.18}$$

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.

2.3.2 Choosing Remaining Parameters

In [5], 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.

2.4 Subproblem Solver Based on SMO

There were two basic methods for solving SVM subproblems. A new, third method was proposed by me in [5].

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).

2.4.1 Free Term Support Vector Machines

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

OP 11.

$$\min_{\vec{w_c}, b_c, \vec{\xi_c}} \quad f\left(\vec{w_c}, b_c, \vec{\xi_c}\right) = \frac{1}{2}\left\|\vec{w_c}\right\|^2 + C_c \sum_{i=1}^{n} \xi_c^i + Db \tag{2.19}$$

subject to

$$y_c^i h\left(\vec{x_i}\right) \geq 1 + \varphi_i - \xi_c^i \tag{2.20}$$

$$\vec{\xi_c} \geq 0 \tag{2.21}$$

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

$$h\left(\vec{x_i}\right) = \vec{w_c} \cdot \vec{x_i} + b_c \quad , \tag{2.22}$$

$$C_c > 0 \quad . \tag{2.23}$$

We modified (2.19) by adding the last term. We propose to derive a dual optimization problem (derivation in Appendix A.8) which is

OP 12.

$$\max_{\vec{\alpha}} \quad f\left(\vec{\alpha}\right) = \vec{\alpha} \cdot \left(1 + \vec{\varphi}\right) - \frac{1}{2}\vec{\alpha^T}\mathbf{Q}\vec{\alpha} \tag{2.24}$$

subject to

$$\vec{\alpha} \cdot \vec{y} = D \tag{2.25}$$

$$0 \leq \alpha_i \leq C_c \tag{2.26}$$

where

$$Q_{ij} = y_i y_j K\left(\vec{x_i}, \vec{x_j}\right) \tag{2.27}$$

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

We can notice that the only difference compared to OP 9 is in (2.25). We can use SMO for solving bSVM as well. First note that clipping formulas for bSVM are the same as (2.7), because in derivation (Appendix A.1) we do need to use D parameter. For the similar reason the solution (2.7) is also the same. The only difference is that the initial solution must fulfill the new condition (2.25). We can see that the bounds for D are

$$\sum_{i=1}^{n_2} C_{d_i} \leq D \leq \sum_{i=1}^{n_1} C_{c_i} \quad , \tag{2.28}$$

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

$$\alpha_{d_i} = \frac{\sum_{i=1}^{n_1} C_{c_i} - D}{n_2} \quad . \tag{2.29}$$

If any of constraints (2.26) 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 (12), it is enough to use existing code for the SMO method, the only difference is that initial values of α_i parameters fulfills (2.25).

2.4.2 Reduced Optimization Problem as bSVC

The reduced optimization problem OP 10 can be reformulated as bSVC with φ_i weights where

$$\varphi_{c_i} = \varphi_{\text{old}}^{c_i} - y_{c_i} \sum_{\substack{j=1 \\ j \notin P}}^{n} y_j \alpha_j K_{c_i j} \tag{2.30}$$

and

$$D = -\sum_{\substack{i=1 \\ i \notin P}}^{n} y_i \alpha_i \ . \tag{2.31}$$

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

$$\beta_i = \alpha_{c_i} \tag{2.32}$$

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 $i = \{1, \ldots, p\}$.

2.4.3 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, [11]). 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.

2.4.4 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 [2], it was shown that the second solver with working sets of size 20 was faster than the first solver for some data sets. In [5], we showed that the third solver with working sets of size 5 is faster than the first one.

2.4.5 Experiments

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

2.5 Heuristic of Alternatives

The SMO standard heuristic chooses parameters in every iteration based on KKT conditions. We proposed in [4] an improvement to SMO that we check additionally growth of an objective function (1.42). 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

$$\left(s_{11}, s_{21}\right), \left(s_{12}, s_{21}\right), \left(s_{11}, s_{22}\right), \left(s_{13}, s_{21}\right), \ldots \quad . \tag{2.33}$$

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.

$$\left(s_{11}, s_{21}\right), \left(s_{12}, s_{21}\right), \left(s_{11}, s_{22}\right), \left(s_{12}, s_{21}\right) \quad . \tag{2.34}$$

Note that we excluded pairs with both parameters the same.

2.5.1 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 $O\left(kn^2\right)$, where k is the number of iterations.

In HoA, objective function value growth of OP 7 needs to be computed in every iteration for every alternative pair. From the (2.1) we get the formula for objective function value growth

$$\begin{aligned} \Delta f_2\left(\vec{\beta}\right) = \sum_{i=1}^{2} \Delta\beta_i - \sum_{j=1}^{2} y_{c_j} \Delta\beta_j \sum_{\substack{i=1 \\ i \notin C}}^{n} y_i \alpha_i K_{c_j i} - \frac{1}{2} \sum_{i=1}^{2} \left(\beta_{i\mathrm{new}}^2 - \beta_{i\mathrm{old}}^2\right) K_{c_i c_i} \\ - y_{c_1 c_2} \left(\beta_{1\mathrm{new}} \beta_{2\mathrm{new}} - \beta_{1\mathrm{old}} \beta_{2\mathrm{old}}\right) K_{c_1 c_2} \quad . \end{aligned} \tag{2.35}$$

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 $O\left(kn^2 + kmn\right)$. 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.

2.5.2 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 2.1. 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.

2.6 Summary

In this report, 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 2.1: 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

Appendix A

A.1 Derivation of SMO β_2 Bounds for φ-SVC

We will derive bounds for β_2 (2.7):

$$U \leq \ \alpha_2 \leq V \ \ , \tag{A.1}$$

where for $y_1 \neq y_2$

$$U = \max\left(0,\ \alpha_{_2}^{old} - \alpha_{_1}^{old}\right) , \tag{A.2}$$

$$V = \min\left(C_2,\ C_1 - \alpha_{_1}^{old} + \alpha_{_2}^{old}\right) \ \ , \tag{A.3}$$

for $y_1 = y_2$

$$U = \max\left(0,\ \alpha_{_1}^{old} + \alpha_{_2}^{old} - C_1\right) , \tag{A.4}$$

$$V = \min\left(C_2,\ \alpha_{_1}^{old} + \alpha_{_2}^{old}\right) \ \ . \tag{A.5}$$

We present two derivations: geometrical and analytical one.

Geometrical Proof. The equality equation of SVM is

$$\alpha_2 = \alpha_{_1}^{\text{old}} y_1 y_2 + \alpha_{_2}^{\text{old}} - \alpha_1 y_1 y_2 \ \ . \tag{A.6}$$

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 (A.6) we get

$$p : \alpha_2 = \alpha_{_1}^{old} + \alpha_{_2}^{old} - \alpha_1 \ \ . \tag{A.7}$$

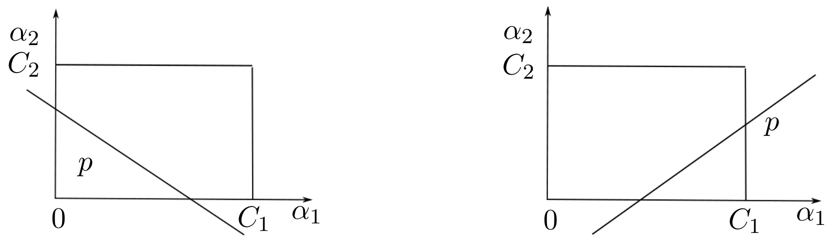
The line p has a negative slope equals to -1, Fig. A.1a.

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 $\alpha_1 = C_1$:

$$\alpha_2 = \alpha_{_1}^{old} + \alpha_{_2}^{old} \tag{A.8}$$

and

$$\alpha_2 = \alpha_{_1}^{old} + \alpha_{_2}^{old} - C_1 \ \ . \tag{A.9}$$



(a)

(b)

Figure A.1: Visualization of the constraints. We can see a line p with the negative slope in a) and the positive in b)

Because crossing points have to lie in the square we get the following bounds for α_2

$$U = \max\left(0,\ \alpha_1^{old} + \alpha_2^{old} - C_1\right), \tag{A.10}$$

$$V = \min\left(C_2,\ \alpha_1^{old} + \alpha_2^{old}\right) \quad . \tag{A.11}$$

When $y_1 \neq y_2$, then $y_1 y_2 = -1$, after substituting it to (A.6)

$$p : \alpha_2 = -\alpha_1^{old} + \alpha_2^{old} + \alpha_1 \quad . \tag{A.12}$$

The line p has a positive slope equals to 1, Fig. A.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 $\alpha_1 = C_1$:

$$\alpha_2 = -\alpha_1^{old} + \alpha_2^{old} \tag{A.13}$$

and

$$\alpha_2 = -\alpha_1^{old} + \alpha_2^{old} + C_1 \quad . \tag{A.14}$$

Because crossing points have to lie in the square we get the following bounds for α_2

$$U = \max\left(0,\ \alpha_2^{old} - \alpha_1^{old}\right), \tag{A.15}$$

$$V = \min\left(C_2,\ C_1 - \alpha_1^{old} + \alpha_2^{old}\right) \quad . \tag{A.16}$$

□

Analytical Proof. We have an inequality for α_1

$$0 \leq \alpha_1 \leq C_1 \tag{A.17}$$

and a line p

$$\alpha_1 y_1 + \alpha_2 y_2 = \alpha_1^{old} y_1 + \alpha_2^{old} y_2 \quad , \tag{A.18}$$

after transformation

$$\alpha_1 = \alpha_1^{old} + y_1 y_2 \alpha_2^{old} - y_1 y_2 \alpha_2 \quad , \tag{A.19}$$

after substituting it to the inequality we get

$$0 \leq \alpha_1^{old} + y_1 y_2 \alpha_2^{old} - y_1 y_2 \alpha_2 \leq C_1 \ . \tag{A.20}$$

When $y_1 = y_2$, then $y_1 y_2 = 1$, so

$$0 \leq \alpha_1^{old} + \alpha_2^{old} - \alpha_2 \leq C_1 \ . \tag{A.21}$$

Consider now the first part of the inequality,

$$\alpha_1^{old} + \alpha_2^{old} - \alpha_2 \geq 0 \tag{A.22}$$

$$\alpha_2 \leq \alpha_1^{old} + \alpha_2^{old} \ . \tag{A.23}$$

Because the parameter α_2 has to satisfy the inequality $\alpha_2 \leq C_2$, so the upper bound for α_2 is

$$V = \min\left(C_2,\ \alpha_1^{old} + \alpha_2^{old}\right) \ . \tag{A.24}$$

Considering the second part of the inequality,

$$\alpha_1^{old} + \alpha_2^{old} - \alpha_2 \leq C_1 \tag{A.25}$$

$$\alpha_2 \geq \alpha_1^{old} + \alpha_2^{old} - C_1 \ . \tag{A.26}$$

Because the parameter α_2 has to satisfy the inequality $\alpha_2 \geq 0$, so the lower bound for α_2 is

$$U = \max\left(0,\ \alpha_1^{old} + \alpha_2^{old} - C_1\right) \ . \tag{A.27}$$

When $y_1 \neq y_2$, then $y_1 y_2 = -1$, so

$$0 \leq \alpha_1^{old} - \alpha_2^{old} + \alpha_2 \leq C_1 \ . \tag{A.28}$$

Consider now the first part of the inequality,

$$\alpha_1^{old} - \alpha_2^{old} + \alpha_2 \geq 0 \tag{A.29}$$

$$\alpha_2 \geq \alpha_2^{old} - \alpha_1^{old} \ . \tag{A.30}$$

Because the parameter α_2 has to satisfy the inequality $\alpha_2 \geq 0$, so the lower bound for α_2 is

$$U = \max\left(0,\ \alpha_2^{old} - \alpha_1^{old}\right) \ . \tag{A.31}$$

Considering the second part of the inequality,

$$\alpha_1^{old} - \alpha_2^{old} + \alpha_2 \leq C_1 \tag{A.32}$$

$$\alpha_2 \leq C_1 + \alpha_2^{old} - \alpha_1^{old} \ . \tag{A.33}$$

Because the parameter α_2 has to satisfy the inequality $\alpha_2 \leq C_2$, so the upper bound for α_2 is

$$V = \min\left(C_2,\ C_1 - \alpha_1^{old} + \alpha_2^{old}\right) \ . \tag{A.34}$$

□

A.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}

$$\alpha_2^{\text{unc}} = \alpha_2^{\text{old}} + \frac{y_2 \left(E_1 - E_2 \right)}{\kappa} \tag{A.35}$$

and then

$$\alpha_2 = \left\{ \begin{array}{l} V, \quad if\ \alpha_2^{\text{new,unc}} > V, \\ \alpha_2^{\text{new,unc}}, \quad if\ U \leq \alpha_2^{\text{new,unc}} \leq V, \\ U, \quad if\ \alpha_2^{\text{new,unc}} < U \end{array} \right. \tag{A.36}$$

$$\alpha_1^{\text{new}} = \alpha_1^{\text{old}} + y_1 y_2 \left(\alpha_2^{\text{old}} - \alpha_2^{\text{new}} \right) \quad . \tag{A.37}$$

For simplicity of the proof we use a notation

$$K \left(\vec{x_i}, \vec{x_j} \right) \equiv K_{ij} \quad , \tag{A.38}$$

where $i, j = 1, 2$, and we define

$$f_i = \sum_{j=1}^{n} y_j \alpha_j K_{ij} \tag{A.39}$$

$$E_i = f_i - y_i = \sum_{j=1}^{n} y_j \alpha_j K_{ij} - y_i \tag{A.40}$$

$$v_i = \sum_{j=3}^{n} y_j \alpha_j K_{ij} = f_i - \sum_{j=1}^{2} y_j \alpha_j K_{ij} \tag{A.41}$$

for $i = 1$ or $i = 2$, where n is the number of all vectors.

The objective function has a form

$$\begin{array}{l} f \left(\alpha_1, \alpha_2 \right) = \alpha_1 + \alpha_2 - \frac{1}{2} K_{11} \alpha_1^2 - \frac{1}{2} K_{22} \alpha_2^2 - \\ y_1 y_2 K_{12} \alpha_1 \alpha_2 - y_1 \alpha_1 v_1 - y_2 \alpha_2 v_2 + \text{const} \quad . \end{array} \tag{A.42}$$

After substituting $s_{ij} = y_i y_j$ for $i, j = 1, 2$ for simplification of notation we get

$$\begin{array}{l} f \left(\alpha_1, \alpha_2 \right) = \alpha_1 + \alpha_2 - \frac{1}{2} K_{11} \alpha_1^2 - \frac{1}{2} K_{22} \alpha_2^2 - \\ s_{12} K_{12} \alpha_1 \alpha_2 - y_1 \alpha_1 v_1 - y_2 \alpha_2 v_2 + \text{const} \quad . \end{array} \tag{A.43}$$

The linear constraint has a form: $\sum\limits_{i=1}^{n} y_i \alpha_i = 0$. It must be satisfied with new values of α_1 and α_2

$$y_1 \alpha_1 + y_2 \alpha_2 = y_1 \alpha_1^{\text{old}} + y_2 \alpha_2^{\text{old}} = \text{const} \quad . \tag{A.44}$$

Dividing above by y_1 and noticing that $y_1 y_2 = y_1 / y_2$ we get

$$\alpha_1 + y_1 y_2 \alpha_2 = \alpha_1^{\text{old}} + y_1 y_2 \alpha_2^{\text{old}} \quad , \tag{A.45}$$

after simplification

$$\alpha_1 + s_{12}\alpha_2 = \alpha_1^{\text{old}} + s_{12}\alpha_2^{\text{old}} \ . \tag{A.46}$$

Introducing notation

$$\gamma = \alpha_1^{\text{old}} + s_{12}\alpha_2^{\text{old}} \tag{A.47}$$

we have

$$\alpha_1 + s_{12}\alpha_2 = \gamma \tag{A.48}$$

$$\alpha_1 = \gamma - s_{12}\alpha_2 \ . \tag{A.49}$$

The above equation shows how to get α_1 from α_2. After substituting above to the objective function we get

$$\begin{array}{l} f\left(\alpha_1, \alpha_2\right) = \gamma - s_{12}\alpha_2 + \alpha_2 - \frac{1}{2}K_{11}\left(\gamma - s_{12}\alpha_2\right)^2 - \frac{1}{2}K_{22}\alpha_2^2 - \\ s_{12}K_{12}\left(\gamma - s_{12}\alpha_2\right)\alpha_2 - y_1\left(\gamma - s_{12}\alpha_2\right)v_1 - y_2\alpha_2 v_2 + \text{const} \ . \end{array} \tag{A.50}$$

After transformation

$$\begin{array}{l} f\left(\alpha_1, \alpha_2\right) = \gamma - s_{12}\alpha_2 + \alpha_2 - \frac{1}{2}K_{11}\left(\gamma^2 - 2\gamma s_{12}\alpha_2 + s_{12}^2\alpha_2^2\right) - \frac{1}{2}K_{22}\alpha_2^2 - \\ s_{12}K_{12}\left(\gamma - s_{12}\alpha_2\right)\alpha_2 - y_1\left(\gamma - s_{12}\alpha_2\right)v_1 - y_2\alpha_2 v_2 + \text{const} \end{array} \tag{A.51}$$

$$\begin{array}{l} f\left(\alpha_1, \alpha_2\right) = \gamma - s_{12}\alpha_2 + \alpha_2 - \frac{1}{2}K_{11}\gamma^2 + K_{11}\gamma s_{12}\alpha_2 - \frac{1}{2}K_{11}\alpha_2^2 - \frac{1}{2}K_{22}\alpha_2^2 \\ -s_{12}K_{12}\left(\gamma - s_{12}\alpha_2\right)\alpha_2 - y_1\left(\gamma - s_{12}\alpha_2\right)v_1 - y_2\alpha_2 v_2 + \text{const} \end{array} \tag{A.52}$$

$$\begin{array}{l} f\left(\alpha_1, \alpha_2\right) = \gamma - s_{12}\alpha_2 + \alpha_2 - \frac{1}{2}K_{11}\gamma^2 + K_{11}\gamma s_{12}\alpha_2 - \frac{1}{2}K_{11}\alpha_2^2 - \frac{1}{2}K_{22}\alpha_2^2 \\ -s_{12}K_{12}\gamma\alpha_2 + K_{12}\alpha_2^2 - y_1\gamma v_1 + y_2\alpha_2 v_1 - y_2\alpha_2 v_2 + \text{const} \ . \end{array} \tag{A.53}$$

Now we compute a partial derivative with respect to α_2

$$\begin{array}{l} \frac{\partial f(\alpha_2)}{\partial \alpha_2} = 1 - s_{12} + K_{11}\gamma s_{12} - K_{11}\alpha_2 - K_{22}\alpha_2 \\ -s_{12}K_{12}\gamma + 2K_{12}\alpha_2 + y_2 v_1 - y_2 v_2 \ . \end{array} \tag{A.54}$$

We are looking for stationary points by equating the derivative to zero

$$\begin{array}{l} \frac{\partial f(\alpha_2)}{\partial \alpha_2} = 1 - s_{12} + K_{11}\gamma s_{12} - K_{11}\alpha_2 - K_{22}\alpha_2 - \\ s_{12}K_{12}\gamma + 2K_{12}\alpha_2 + y_2 v_1 - y_2 v_2 = 0 \end{array} \tag{A.55}$$

$$\begin{array}{l} 1 - s_{12} + K_{11}\gamma s_{12} - K_{11}\alpha_2 - K_{22}\alpha_2 \\ -s_{12}K_{12}\gamma + 2K_{12}\alpha_2 + y_2 v_1 - y_2 v_2 = 0 \ . \end{array} \tag{A.56}$$

Dividing both sides by y_2 we get

$$\begin{array}{l} y_2 - y_1 + K_{11}\gamma y_1 - K_{11}y_2\alpha_2 - K_{22}y_2\alpha_2 - \\ y_1K_{12}\gamma + 2K_{12}y_2\alpha_2 + v_1 - v_2 = 0 \ . \end{array} \tag{A.57}$$

Adding the new superscript for α_2 we get

$$\begin{array}{l} y_2 - y_1 + K_{11}\gamma y_1 - K_{11}y_2\alpha_2^{\text{new}} - K_{22}y_2\alpha_2^{\text{new}} - \\ y_1K_{12}\gamma + 2K_{12}y_2\alpha_2^{\text{new}} + v_1 - v_2 = 0 \ . \end{array} \tag{A.58}$$

Substituting for γ, v_1 i v_2

$$\begin{aligned}&y_2 - y_1 + K_{11}\left(\alpha_1 + s_{12}\alpha_2\right) y_1 - K_{11} y_2 \alpha_2^{\text{new}} - K_{22} y_2 \alpha_2^{\text{new}} \\ &- y_1 K_{12}\left(\alpha_1 + s_{12}\alpha_2\right) + 2K_{12} y_2 \alpha_2^{\text{new}} + f_1 - y_1\alpha_1 K_{11} - y_2\alpha_2 K_{12} \\ &- f_2 + y_1\alpha_1 K_{12} + y_2\alpha_2 K_{22} = 0\end{aligned} \tag{A.59}$$

$$\begin{aligned}&y_2 - y_1 + K_{11}\alpha_1 y_1 + K_{11} y_2\alpha_2 - K_{11} y_2 \alpha_2^{\text{new}} - K_{22} y_2 \alpha_2^{\text{new}} \\ &- y_1 K_{12}\alpha_1 - K_{12} y_2\alpha_2 + 2K_{12} y_2 \alpha_2^{\text{new}} + f_1 - y_1\alpha_1 K_{11} - y_2\alpha_2 K_{12} \\ &- f_2 + y_1\alpha_1 K_{12} + y_2\alpha_2 K_{22} = 0\end{aligned} \tag{A.60}$$

$$\begin{aligned}&y_2 - y_1 + K_{11} y_2\alpha_2 - K_{11} y_2 \alpha_2^{\text{new}} - K_{22} y_2 \alpha_2^{\text{new}} \\ &- K_{12} y_2\alpha_2 + 2K_{12} y_2 \alpha_2^{\text{new}} + f_1 - y_2\alpha_2 K_{12} \\ &- f_2 + y_2\alpha_2 K_{22} = 0\end{aligned} \tag{A.61}$$

$$\begin{aligned}&y_2 - y_1 - y_2\alpha_2^{\text{new}}\left(K_{11} + K_{22} - 2K_{12}\right) + y_2\alpha_2(K_{11} + K_{22} - 2K_{12}) \\ &+ f_1 - f_2 = 0 \ .\end{aligned} \tag{A.62}$$

Introducing notation $\kappa = K_{11} + K_{22} - 2K_{12}$ we get

$$y_2 - y_1 - y_2\alpha_2^{\text{new}}\kappa + y_2\alpha_2\kappa + f_1 - f_2 = 0 \ . \tag{A.63}$$

Dividing both sides by y_2 and κ

$$\alpha_2^{\text{new}} = \alpha_2 + \frac{y_2\left(E_1 - E_2\right)}{\kappa} \ . \tag{A.64}$$

After all we have to limit α_2^{new}, so it will lie in $[U, V]$.

A.3 Derivation of the SMO solution for φ-SVC

Compare it with A.2. We have a new objective function

$$f\left(\alpha_1, \alpha_2\right) = \alpha_1\varphi_1 + \alpha_2\varphi_2 + f_{\text{smo}}\left(\alpha_1, \alpha_2\right) \ , \tag{A.65}$$

where f_{smo} is the f function for SMO from A.2. After substituting

$$\alpha_1 = \gamma - y_1 y_2 \alpha_2 \ , \tag{A.66}$$

where

$$\gamma = \alpha_1^{\text{old}} + y_1 y_2 \alpha_2^{\text{old}} \tag{A.67}$$

we get

$$f\left(\alpha_1, \alpha_2\right) = \varphi_1\gamma - \varphi_1 y_1 y_2 \alpha_2 + \alpha_2\varphi_2 + f_{\text{smo}}\left(\alpha_1, \alpha_2\right) \ . \tag{A.68}$$

After differentiating we get

$$\frac{\partial f\left(\alpha_1, \alpha_2\right)}{\partial\alpha_2} = \varphi_2 - \varphi_1 y_1 y_2 + \frac{\partial f_{\text{smo}}\left(\alpha_1, \alpha_2\right)}{\partial\alpha_2} \ . \tag{A.69}$$

And a solution is

$$\alpha_2^{\text{new}} = \alpha_2 + \frac{y_2\left(E_1 - E_2\right)}{\kappa} \quad , \tag{A.70}$$

where

$$\begin{aligned} E_i &= \sum_{j=1}^{n} y_j \alpha_j K_{ij} - y_i - y_i \varphi_i \\ \kappa &= K_{11} + K_{22} - 2K_{12} \quad . \end{aligned} \tag{A.71}$$

A.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

$$K\left(\vec{x_i}, \vec{x_j}\right) \equiv K_{ij} \quad , \tag{A.72}$$

where $i, j = 1, 2$, and we define

$$E_i = \sum_{j=1}^{n} y_j \alpha_j K_{ij} - y_i \tag{A.73}$$

$$v_1 = \sum_{j=2}^{n} y_j \alpha_j K_{ij} = \sum_{j=1}^{n} y_j \alpha_j K_{1j} - y_1 \alpha_1 K_{11} \quad . \tag{A.74}$$

The objective function has a form

$$W\left(\alpha_1\right) = \alpha_1 - \tfrac{1}{2} K_{11} \alpha_1^2 - y_1 \alpha_1 v_1 + \text{const} \quad . \quad . \tag{A.75}$$

Now we compute a partial derivative with respect to α_1

$$\tfrac{\partial W(\alpha_1)}{\partial \alpha_1} = 1 - K_{11}\alpha_1 - y_1 v_1 \quad . \tag{A.76}$$

We are looking for stationary points by equating the derivative to zero

$$\frac{\partial W\left(\alpha_1\right)}{\partial \alpha_1} = 1 - K_{11}\alpha_1^{\text{new}} - y_1 v_1 = 0 \tag{A.77}$$

$$1 - K_{11}\alpha_1^{\text{new}} - y_1 \sum_{j=1}^{n} y_j \alpha_j K_{1j} + \alpha_1 K_{11} = 0 \tag{A.78}$$

$$\alpha_1^{\text{new}} = \alpha_1 - \frac{y_1 E_1}{K_{11}} \quad . \tag{A.79}$$

Then we have to bound α_1^{new}:

$$0 \le \alpha_1^{\text{new}} \le C_1 \quad . \tag{A.80}$$

A.5 Derivation of SMO Without Offset for φ-SVC

Compare it with *Appendix A.4*. We have a new objective function

$$f\left(\alpha_1\right) = \alpha_1 \varphi_1 + f_{\text{smo}}\left(\alpha_1, \alpha_2\right) \quad , \tag{A.81}$$

where f_{smo} is the f function for SMO from *Appendix A.4*. After differentiating we get

$$\frac{\partial f\left(\alpha_1\right)}{\partial \alpha_1} = \varphi_1 + \frac{\partial f_{\text{smo}}\left(\alpha_1\right)}{\partial \alpha_1} \quad . \tag{A.82}$$

And a solution is

$$\alpha_1^{\text{new}} = \alpha_1 - \frac{y_1 E_1}{K_{11}} \quad , \tag{A.83}$$

where

$$E_i = \sum_{j=1}^{n} y_j \alpha_j K_{ij} - y_i - y_i \varphi_i \quad . \tag{A.84}$$

A.6 Derivation of Optimization Possibility Conditions

A.6.1 Optimization Possibility Conditions Derived Directly

We can transform the linear constraint (2.2) into the following form

$$\beta_d = -y_{c_d} \sum_{\substack{i=1 \\ i \neq d}}^{p} y_{c_i} \beta_i - y_{c_d} \sum_{\substack{i=1 \\ i \notin C}}^{n} y_i \alpha_i \quad , \tag{A.85}$$

where $d \in \{1, 2, \ldots, p\}$ is an arbitrarily chosen parameter. After substituting β_d to the (2.1), we get the following optimization subproblem

OP 13.

$$\begin{aligned}
\max_{\vec{\gamma}} \quad & f_3\left(\vec{\gamma}\right) = -y_{c_d} \sum_{\substack{i=1 \\ i \neq d}}^{p} y_{c_i} \gamma_{e_i} - y_{c_d} \sum_{\substack{i=1 \\ i \notin C}}^{n} y_i \alpha_i + \sum_{\substack{i=1 \\ i \neq d}}^{p} \gamma_{e_i} \\
& + \sum_{\substack{i=1 \\ i \notin C}}^{n} \alpha_i - \tfrac{1}{2} \left(\sum_{\substack{i=1 \\ i \neq d}}^{p} y_{c_i} \gamma_{e_i} + \sum_{\substack{i=1 \\ i \notin C}}^{n} y_i \alpha_i \right)^2 K_{c_d c_d} \\
& + \left(\sum_{\substack{i=1 \\ i \neq d}}^{p} y_{c_i} \gamma_{e_i} + \sum_{\substack{i=1 \\ i \notin C}}^{n} y_i \alpha_i \right) \sum_{\substack{i=1 \\ i \neq d}}^{p} y_{c_i} \gamma_{e_i} K_{c_d c_i} \\
& - \tfrac{1}{2} \sum_{\substack{i=1 \\ i \neq d}}^{p} y_{c_i} \gamma_{e_i} \sum_{\substack{j=1 \\ j \neq d}}^{p} y_{c_j} \gamma_{e_j} K_{c_i c_j} \\
& + \left(\sum_{\substack{i=1 \\ i \neq d}}^{p} y_{c_i} \gamma_{e_i} + \sum_{\substack{i=1 \\ i \notin C}}^{n} y_i \alpha_i \right) \sum_{\substack{i=1 \\ i \notin C}}^{n} y_i \alpha_i K_{c_d i} \\
& - \sum_{\substack{i=1 \\ i \neq d}}^{p} y_{c_i} \gamma_{e_i} \sum_{\substack{j=1 \\ j \notin C}}^{n} y_j \alpha_j K_{c_i j} - \tfrac{1}{2} \sum_{\substack{i=1 \\ i \notin C}}^{n} \sum_{\substack{j=1 \\ j \notin C}}^{n} y_{ij} \alpha_i \alpha_j K_{ij}
\end{aligned} \tag{A.86}$$

subject to

$$0 \leq \gamma_{e_i} \leq C, \text{ for } i \in \{1, 2, \ldots, p\} \setminus \{d\}, \ C > 0 \tag{A.87}$$

$$0 \leq c = -y_{c_d} \sum_{\substack{i=1 \\ i \neq d}}^{p} y_{c_i} \gamma_{e_i} - y_{c_d} \sum_{\substack{i=1 \\ i \notin C}}^{n} y_i \alpha_i \leq C \ \ , \tag{A.88}$$

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

$$\frac{\partial}{\partial \gamma_{e_k}} f_3(\vec{\gamma}_{\text{old}}) = y_{c_k} (E_{c_d} - E_{c_k}) \ \ , \tag{A.89}$$

where E_i is defined in (2.5).

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 (A.88), the second part consists of conditions based on partial derivatives. Merging all conditions leads to the overall optimization conditions.

The (A.88) must be satisfied after changes, hence we can write

$$\begin{aligned} -\alpha_{c_d}^{\text{old}} \leq \Delta\alpha_{c_d} = -\alpha_{c_d}^{\text{old}} - y_{c_d} \sum_{\substack{i=1 \\ i \neq d}}^{p} y_{c_i} \alpha_{c_i}^{\text{new}} \\ -y_{c_d} \sum_{\substack{i=1 \\ i \notin C}}^{n} y_i \alpha_i \leq C - \alpha_{c_d}^{\text{old}} \ \ . \end{aligned} \tag{A.90}$$

After substituting

$$\alpha_{c_d}^{\text{old}} = -y_{c_d} \sum_{\substack{i=1 \\ i \neq d}}^{p} y_{c_i} \alpha_{c_i}^{\text{old}} - y_{c_d} \sum_{\substack{i=1 \\ i \notin C}}^{n} y_i \alpha_i \tag{A.91}$$

we get the following condition

$$-\alpha_{c_d}^{\text{old}} \leq \Delta\alpha_{c_d} = -y_{c_d} \sum_{\substack{i=1 \\ i \neq d}}^{p} y_{c_i} \Delta\alpha_{c_i} \leq C - \alpha_{c_d}^{\text{old}} \ \ . \tag{A.92}$$

Theorem A.6.1 (Necessary optimization conditions based on satisfying (A.88))**.** *If the condition* (A.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*

$$\begin{aligned}
&G_1 := \{i \in \{1, 2, \ldots, n\} : (y_i = 1 \wedge \alpha_i = 0) \\
&\vee (y_i = -1 \wedge \alpha_i = C_i) \vee (0 < \alpha_i < C_i)\} \\
&G_2 := \{i \in \{1, 2, \ldots, n\} : (y_i = -1 \wedge \alpha_i = 0) \\
&\vee (y_i = 1 \wedge \alpha_i = C_i) \vee (0 < \alpha_i < C_i)\} \quad .
\end{aligned} \tag{A.93}$$

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 (A.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 (A.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 (A.92) becomes $-C \leq \Delta\alpha_{c_d} \leq 0$ what cannot be true. □

Theorem A.6.2 (Sufficient optimization conditions based on satisfying (A.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* (A.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 (A.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 (A.92) is satisfied. If $0 < \alpha_{c_d}^{\text{old}} < C$, then condition (A.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 (A.92) is satisfied. If $0 < \alpha_{c_d}^{\text{old}} < C$, then condition (A.92) is satisfied, when $\Delta\alpha_{c_b}$ is set to close enough to zero value. □

Theorem A.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*

$$\begin{aligned} &\frac{f_3(\vec{\gamma})}{\gamma_{e_k}} > 0 \text{ when } \alpha_{c_k} = 0 \\ &\frac{f_3(\vec{\gamma})}{\gamma_{e_k}} < 0 \text{ when } \alpha_{c_k} = C_{c_k} \\ &\frac{f_3(\vec{\gamma})}{\gamma_{e_k}} \neq 0 \text{ when } 0 < \alpha_{c_k} < C_{c_k} \ . \end{aligned} \tag{A.94}$$

Proof. We prove that if all partial derivatives of the function f_3 violate the condition (A.94), then optimization will be impossible. If (A.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 A.6.1. *After substitution* (A.89) *to* (A.94) *we get*

$$y_{c_k} \left(E_{c_d} - E_{c_k}\right) > 0 \ \textit{when} \ \alpha_{c_k} = 0 \tag{A.95}$$

$$y_{c_k} \left(E_{c_d} - E_{c_k}\right) < 0 \ \textit{when} \ \alpha_{c_k} = C_{c_k} \tag{A.96}$$

$$y_{c_k} \left(E_{c_d} - E_{c_k}\right) \neq 0 \ \textit{when} \ 0 < \alpha_{c_k} < C_{c_k} \tag{A.97}$$

After simplification:
When $y_{c_k} = 1$:

$$E_{c_k} < E_{c_d} \ \textit{when} \ \alpha_{c_k} = 0 \tag{A.98}$$

$$E_{c_k} > E_{c_d} \ \textit{when} \ \alpha_{c_k} = C_{c_k} \tag{A.99}$$

$$E_{c_k} \neq E_{c_d} \ \textit{when} \ 0 < \alpha_{c_k} < C_{c_k} \tag{A.100}$$

When $y_{c_k} = -1$:

$$E_{c_k} > E_{c_d} \ \textit{when} \ \alpha_{c_k} = 0 \tag{A.101}$$

$$E_{c_k} < E_{c_d} \ \textit{when} \ \alpha_{c_k} = C_{c_k} \tag{A.102}$$

$$E_{c_k} \neq E_{c_d} \ \textit{when} \ 0 < \alpha_{c_k} < C_{c_k} \tag{A.103}$$

Theorem A.6.4 (Sufficient optimization conditions based on partial derivatives)**.** *If one of the partial derivatives of the function f_3 satisfies the condition* (A.94)*, then optimization is possible based on partial derivatives for some parameter changes.*

Proof. We can change the parameter that satisfies the condition (A.94). The remaining parameters, which are attributed to f_3 variables, can stay unchanged, and then f_3 value will grow. □

Theorem A.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* (A.94)*.*

Proof. Because of Thm. A.6.1, Thm. A.6.2, Thm. A.6.3, Thm. A.6.4 we only have to prove, that overall optimization is a multiplication of optimization based on (A.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. A.6.5 is a different formulation of the Thm. 2.3.1.

A.6.2 Optimization Possibility Conditions Derived From KKT

We can derive the optimization possibility conditions from KKT conditions (1.47), (1.48). We have the following cases:

- When $\alpha_i = 0$, then from (1.48), $\xi_i = 0$. From (1.47)

$$y_i \left(\vec{w} \cdot \vec{x_i} + b\right) \geq 1 + \varphi_i \tag{A.104}$$

when $y_i = 1$

$$b \geq 1 + \varphi_i - \vec{w} \cdot \vec{x_i} \tag{A.105}$$

when $y_i = -1$

$$b \leq -\vec{w} \cdot \vec{x_i} - 1 - \varphi_i \quad . \tag{A.106}$$

- When $\alpha_i = C_i$

$$y_i \left(\vec{w} \cdot \vec{x_i} + b\right) - 1 - \varphi_i + \xi_i = 0 \tag{A.107}$$

$$\xi_i = -y_i \left(\vec{w} \cdot \vec{x_i} + b\right) + 1 + \varphi_i \quad . \tag{A.108}$$

Because $\xi_i \geq 0$, so

$$-y_i \left(\vec{w} \cdot \vec{x_i} + b\right) + 1 + \varphi_i \geq 0 \tag{A.109}$$

$$y_i \left(\vec{w} \cdot \vec{x_i} + b\right) \leq 1 + \varphi_i \tag{A.110}$$

when $y_i = 1$

$$b \leq 1 + \varphi_i - \vec{w} \cdot \vec{x_i} \tag{A.111}$$

when $y_i = -1$

$$b \geq -\vec{w} \cdot \vec{x_i} - 1 - \varphi_i \quad . \tag{A.112}$$

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

$$y_i \left(\vec{w} \cdot \vec{x_i} + b\right) - 1 - \varphi_i = 0 \tag{A.113}$$

$$b = -\vec{w} \cdot \vec{x_i} + y_i + y_i\varphi_i \quad . \tag{A.114}$$

After substituting (1.13) to above equations we get

- when $\alpha_i = 0$ and $y_i = 1$
$$b \geq -E_i \tag{A.115}$$
when $y_i = -1$
$$b \leq -E_i \ , \tag{A.116}$$
- when $\alpha_i = C_i$ and $y_i = 1$
$$b \leq -E_i \tag{A.117}$$
when $y_i = -1$
$$b \geq -E_i \ , \tag{A.118}$$
- when $0 < \alpha_i < C_i$
$$b = -E_i \ , \tag{A.119}$$

where E_i is defined in (2.5).

We will prove that from above equations and (A.92), we can implicate Thm. 2.3.1.

Proof. First, we can notice that the conditions from (A.115) to (A.119) are violated when we have two points from the same group G_1 or G_2. It is a direct conclusion from (A.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 (A.115) to (A.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. □

A.7 Derivation of the Dual Form of OP 12

OP 14.
$$\max_{\vec{\alpha},\vec{r}} \ d\left(\vec{\alpha},\vec{r}\right) \tag{A.120}$$

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

$$\frac{\partial t\left(\vec{w}, b, \vec{\alpha}, \vec{\xi}, \vec{r}\right)}{\partial w_i} = w_i - \sum_{j=1}^{n} \alpha_j y_{\mathrm{c}}^{j} x_{ji} = 0 \tag{A.121}$$

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

$$\frac{\partial t\left(\vec{w}, b, \vec{\alpha}, \vec{\xi}, \vec{r}\right)}{\partial b} = \sum_{i=1}^{n} \alpha_i y_{\mathrm{c}}^{i} = D \ \ . \tag{A.122}$$

A partial derivative with respect to ξ_i is

$$\frac{\partial t\left(\vec{w}, b, \vec{\alpha}, \vec{\xi}, \vec{r}\right)}{\partial \xi_i} = C_i - r_i - \alpha_i = 0 \ \ . \tag{A.123}$$

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

$$\begin{aligned} d\left(\vec{\alpha}, \vec{r}\right) = {} & \tfrac{1}{2} \sum_{i=1}^{m} \left(\sum_{j=1}^{n} \alpha_j y_{\mathrm{c}}^{j} x_{ji} \right) \left(\sum_{k=1}^{n} \alpha_k y_{\mathrm{c}}^{k} x_{ki} \right) \\ & - \sum_{i=1}^{n} \alpha_i y_{\mathrm{c}}^{i} \left(\sum_{j=1}^{m} w_j x_{ij} + b \right) + \sum_{i=1}^{n} \alpha_i \left(1 + \varphi_i\right) + \sum_{i=1}^{n} C_i \xi_i + Db \\ & - \sum_{i=1}^{n} \alpha_i \xi_i - \sum_{i=1}^{n} r_i \xi_i \end{aligned} \tag{A.124}$$

$$\begin{aligned} d\left(\vec{\alpha}, \vec{r}\right) = {} & \tfrac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{n} \alpha_k \alpha_j y_{\mathrm{c}}^{k} y_{\mathrm{c}}^{j} x_{ki} x_{ji} - \sum_{i=1}^{n} \alpha_i y_{\mathrm{c}}^{i} \sum_{j=1}^{m} w_j x_{ij} \\ & - b \sum_{i=1}^{n} \alpha_i y_{\mathrm{c}}^{i} + Db + \sum_{i=1}^{n} \alpha_i \left(1 + \varphi_i\right) \end{aligned} \tag{A.125}$$

$$\begin{aligned} d\left(\vec{\alpha}, \vec{r}\right) = {} & \tfrac{1}{2} \sum_{j=1}^{n} \sum_{k=1}^{n} \alpha_k \alpha_j y_{\mathrm{c}}^{k} y_{\mathrm{c}}^{j} \sum_{i=1}^{m} x_{ji} x_{ki} \\ & - \sum_{i=1}^{n} \alpha_i y_{\mathrm{c}}^{i} \sum_{j=1}^{m} x_{ij} \sum_{k=1}^{n} \alpha_k y_{\mathrm{c}}^{k} x_{kj} + \sum_{i=1}^{n} \alpha_i \left(1 + \varphi_i\right) \end{aligned} \tag{A.126}$$

$$\begin{aligned} d\left(\vec{\alpha}, \vec{r}\right) = {} & \tfrac{1}{2} \sum_{j=1}^{n} \sum_{k=1}^{n} \alpha_k \alpha_j y_{\mathrm{c}}^{k} y_{\mathrm{c}}^{j} \sum_{i=1}^{m} x_{ji} x_{ki} \\ & - \sum_{i=1}^{n} \sum_{k=1}^{n} \alpha_k \alpha_i y_{\mathrm{c}}^{k} y_{\mathrm{c}}^{i} \sum_{j=1}^{m} x_{ij} x_{kj} + \sum_{i=1}^{n} \alpha_i \left(1 + \varphi_i\right) \end{aligned} \tag{A.127}$$

$$d\left(\vec{\alpha}, \vec{r}\right) = -\frac{1}{2} \sum_{i=1}^{n} \sum_{k=1}^{n} \alpha_k \alpha_i y_{\mathrm{c}}^{k} y_{\mathrm{c}}^{i} \sum_{j=1}^{m} x_{ij} x_{kj} + \sum_{i=1}^{n} \alpha_i \left(1 + \varphi_i\right) \ \ . \tag{A.128}$$

The dual form is

OP 15.

$$\max_{\vec{\alpha},\vec{r}} \quad d\left(\vec{\alpha},\vec{r}\right) = \sum_{i=1}^{n} \alpha_i \left(1+\varphi_i\right) - \frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{n} \alpha_k \alpha_i y_{\mathrm{c}}^{k} y_{\mathrm{c}}^{i} \sum_{j=1}^{m} x_{ij} x_{kj} \tag{A.129}$$

subject to

$$\sum_{i=1}^{n} \alpha_i y_{\mathrm{c}}^{i} = D \tag{A.130}$$

$$C_i = r_i + \alpha_i \tag{A.131}$$

$$\alpha_i \geq 0 \tag{A.132}$$

$$r_i \geq 0 \tag{A.133}$$

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

A.8 Derivation of OP 12

OP 16.

$$\max_{\vec{\alpha},\vec{r}} \quad d\left(\vec{\alpha},\vec{r}\right) \tag{A.134}$$

where

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

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

$$\frac{\partial t\left(\vec{w},b,\vec{\alpha},\vec{\xi},\vec{r}\right)}{\partial w_i} = w_i - \sum_{j=1}^{n} \alpha_j y_{\mathrm{c}}^{j} x_{ji} = 0 \tag{A.135}$$

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

$$\frac{\partial t\left(\vec{w},b,\vec{\alpha},\vec{\xi},\vec{r}\right)}{\partial b} = \sum_{i=1}^{n} \alpha_i y_{\mathrm{c}}^{i} = D \ . \tag{A.136}$$

A partial derivative with respect to ξ_i is

$$\frac{\partial t\left(\vec{w}, b, \vec{\alpha}, \vec{\xi}, \vec{r}\right)}{\partial \xi_i} = C_i - r_i - \alpha_i = 0 \ . \tag{A.137}$$

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

$$\begin{aligned} & d\left(\vec{\alpha}, \vec{r}\right) = \tfrac{1}{2} \textstyle\sum_{i=1}^{m} \left(\sum_{j=1}^{n} \alpha_j y_c^j x_{ji}\right) \left(\sum_{k=1}^{n} \alpha_k y_c^k x_{ki}\right) \\ & - \textstyle\sum_{i=1}^{n} \alpha_i y_c^i \left(\sum_{j=1}^{m} w_j x_{ij} + b\right) + \sum_{i=1}^{n} \alpha_i \left(1 + \varphi_i\right) + \sum_{i=1}^{n} C_i \xi_i + Db \\ & - \textstyle\sum_{i=1}^{n} \alpha_i \xi_i - \sum_{i=1}^{n} r_i \xi_i \end{aligned} \tag{A.138}$$

$$\begin{aligned} & d\left(\vec{\alpha}, \vec{r}\right) = \tfrac{1}{2} \textstyle\sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{n} \alpha_k \alpha_j y_c^k y_c^j x_{ki} x_{ji} - \sum_{i=1}^{n} \alpha_i y_c^i \sum_{j=1}^{m} w_j x_{ij} \\ & -b \textstyle\sum_{i=1}^{n} \alpha_i y_c^i + Db + \sum_{i=1}^{n} \alpha_i \left(1 + \varphi_i\right) \end{aligned} \tag{A.139}$$

$$\begin{aligned} & d\left(\vec{\alpha}, \vec{r}\right) = \tfrac{1}{2} \textstyle\sum_{j=1}^{n} \sum_{k=1}^{n} \alpha_k \alpha_j y_c^k y_c^j \sum_{i=1}^{m} x_{ji} x_{ki} \\ & - \textstyle\sum_{i=1}^{n} \alpha_i y_c^i \sum_{j=1}^{m} x_{ij} \sum_{k=1}^{n} \alpha_k y_c^k x_{kj} + \sum_{i=1}^{n} \alpha_i \left(1 + \varphi_i\right) \end{aligned} \tag{A.140}$$

$$\begin{aligned} & d\left(\vec{\alpha}, \vec{r}\right) = \tfrac{1}{2} \textstyle\sum_{j=1}^{n} \sum_{k=1}^{n} \alpha_k \alpha_j y_c^k y_c^j \sum_{i=1}^{m} x_{ji} x_{ki} \\ & - \textstyle\sum_{i=1}^{n} \sum_{k=1}^{n} \alpha_k \alpha_i y_c^k y_c^i \sum_{j=1}^{m} x_{ij} x_{kj} + \sum_{i=1}^{n} \alpha_i \left(1 + \varphi_i\right) \end{aligned} \tag{A.141}$$

$$d\left(\vec{\alpha}, \vec{r}\right) = -\frac{1}{2} \sum_{i=1}^{n} \sum_{k=1}^{n} \alpha_k \alpha_i y_c^k y_c^i \sum_{j=1}^{m} x_{ij} x_{kj} + \sum_{i=1}^{n} \alpha_i \left(1 + \varphi_i\right) \ . \tag{A.142}$$

The dual form is

OP 17.

$$\max_{\vec{\alpha}, \vec{r}} \quad d\left(\vec{\alpha}, \vec{r}\right) = \sum_{i=1}^{n} \alpha_i \left(1 + \varphi_i\right) - \frac{1}{2} \sum_{i=1}^{n} \sum_{k=1}^{n} \alpha_k \alpha_i y_c^k y_c^i \sum_{j=1}^{m} x_{ij} x_{kj} \tag{A.143}$$

subject to

$$\sum_{i=1}^{n} \alpha_i y_c^i = D \tag{A.144}$$

$$C_i = r_i + \alpha_i \tag{A.145}$$

$$\alpha_i \geq 0 \tag{A.146}$$

$$r_i \geq 0 \tag{A.147}$$

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

References

[1] L. Hamel. *Knowledge discovery with support vector machines.* Wiley series on methods and applications in data mining. John Wiley & Sons, 2009. 11

[2] T. Joachims. Making large-scale support vector machine learning practical, 1998. 10, 12, 15

[3] 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. 12

[4] 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. 10, 16

[5] 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. 10, 12, 13, 15

[6] 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. 7

[7] 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. 7

[8] 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. 7

[9] John C. Platt. *Fast training of support vector machines using sequential minimal optimization*, pages 185–208. MIT Press, Cambridge, MA, USA, 1999. 9, 10

[10] Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001. 11

[11] R. J. Vanderbei. LOQO: An interior point code for quadratic programming. *Optimization Methods and Software*, 11:451–484, 1999. 15

[12] 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. 7

Notation and Symbols

Miscellaneous

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

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

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

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

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

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

Optimization theory

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

Abbreviations

δ-SVR δ support vector regression.

ν-SVC ν support vector classification.

ε-SVR ε-insensitive support vector regression.

φ-SVC φ support vector classification.

C-SVC C support vector classification.

HoA heuristic of alternatives.

KKT Karush-Kuhn-Tucker.

RBF radial basis function.

SMO sequential minimal optimization.

SMS Sequential Multidimensional Subsolver.

SVC support vector classification.

SVM support vector machines.

VC Vapnik-Chervonenkis.