TECHNICAL REPORT

No. CCLS-14-02
Title: A Fractional Programming Framework for Support Vector Machine-type Formulations
Authors: Ilia Vovsha

A Fractional Programming Framework for Support Vector Machine-type Formulations

Ilia Vovsha IV2121@COLUMBIA.EDU
Computer Science Department
Columbia University
New York, NY 10027, USA

Abstract

We develop a theoretical framework for relating various formulations of regularization problems through fractional programming. We focus on problems with objective functions of the type $L + \lambda \cdot P$, where the parameter λ lacks intuitive interpretation. We observe that fractional programming is an elegant approach to obtain bounds on the range of the parameter, and then generalize this approach to show that different forms can be obtained from a common fractional program. Furthermore, we apply the proposed framework in two concrete settings; we consider support vector machines (SVMs), where the framework clarifies the relation between various existing soft-margin dual forms for classification, and the SVM+ algorithm (Vapnik and Vashist, 2009), where we use this methodology to derive a new dual formulation, and obtain bounds on the cost parameter.

Keywords: regularization, fractional programming, support vector machines, dual formulations, SVM+ method

1. Introduction

We consider general regularization problems with objective functions of the type $L(x) + \lambda \cdot P(x)$ subject to some set of constraints. Since the parameter λ is balancing between the loss and penalty functions, often two incongruent quantities, it may lack intuitive interpretation. As a result, the task of parameter selection can be non-trivial and computationally inefficient. The relevant range of parameter values can vary significantly from one dataset to another, and it is not always possible to follow a search procedure that is both effective and consistent. Nevertheless, there are several standard approaches to address this issue:

- The simplest approach is to provide a wide range of values $\{\lambda_1, \ldots, \lambda_k\}$, and select the optimal parameter using cross-validation (Shawe-Taylor and Cristianini, 2004; Chang and Lin, 2011). Frequently, the loss and penalty functions have predictable structure (e.g., loss is finite). In this case, we can make the parameter search more effective by first obtaining finite bounds on the parameter $\{\lambda_{min}, \lambda_{max}\}$, and then selecting the values more carefully.

- Although a heuristic approach is often satisfactory in practice, we can try to do better by fitting the entire path of parameters (Gunter and Zhu, 2005; Rosset and Zhu, 2007; Hastie et al., 2004; Loosli et al., 2007). This method can be efficient as long as the cost of fitting an entire path is proportional to solving the problem with a single parameter value (Hastie et al., 2004).

- We could introduce an alternative form (denoted $\mathcal{P}^*$) with a different, more intuitive parameter (λ^*), and then attempt to demonstrate equivalence to the original

form (denoted $\mathcal{P}$). However, even if we examine a pair of forms $\{\mathcal{P}, \mathcal{P}^*\}$ with specific loss/penalty functions, proving equivalence can be quite intricate. One concrete example is SVM for classification (Schölkopf et al., 2000; Chang and Lin, 2001).

We are particularly interested in the third approach, but we aim for an elegant and general method for transforming one form into another. Instead of *introducing* forms and then laboring to prove equivalence, we propose a framework under which we can obtain an equivalent form in essentially two well-justified steps.

We develop this framework based on fractional programming theory and notation, and apply it to dual optimization problems for SVM-type algorithms. Therefore, to simplify the discussion, we flip the sign of the parameter in the objective function and refer to (maximizing) $N(x) - q \cdot D(x)$ rather than (minimizing) $L(x) + \lambda \cdot P(x)$. Formally, the task can be stated as follows:

> *Given an optimization problem $\mathcal{P}$ with objective of the type $N(x) - q \cdot D(x)$ subject to some set of constraints, and where the parameter q lacks intuitive interpretation, we wish to obtain an equivalent form $\mathcal{P}^*$ with a parameter q^* that has a concrete meaning. The forms $\mathcal{P}, \mathcal{P}^*$ should have the same optimal solution sets. In other words, for each fixed parameter q, there exists a corresponding parameter q^* such that the particular solution vector x_0 of $\mathcal{P}$ is also the solution vector of $\mathcal{P}^*$.*

As we discuss in Section 1.2 below, the key to our approach is to establish equivalence for a very specific pair of parameters first, and only then generalize the proof to the entire range. This sub-task is crucial for clarifying the intuitive connection since it is not evident why fractional programming should be considered to begin with.

1.1 Fractional Programming-Based Approach

In *Fractional Programming* (FPG), the focus is on optimization problems characterized by ratio(s) of functions in the objective:

$$(\mathcal{F}) \qquad \max \left\{ R(x) = \frac{N(x)}{D(x)} \,\middle|\, x \in S \right\}$$

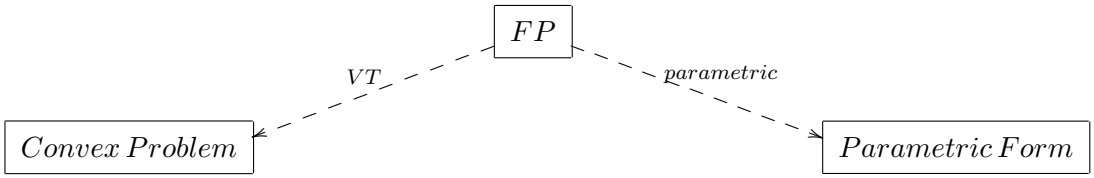
There are multiple approaches for solving a *fractional program* (FP), and the preferable method may depend on the specific structure—the properties of the functions $\{N, D\}$, and the feasible set S—of the problem (Schaible, 1981; Schaible and Ibaraki, 1983; Ibaraki, 1981; Avriel et al., 1988).

One common approach is based on introducing a parametric convex problem related to the given FP:

$$(\mathcal{F}_q) \qquad \max \{N(x) - qD(x) \,|\, x \in S\}$$

The parametric problem is then solved repeatedly for a convergent sequence of parameters $q_1, q_2, \ldots, \bar{q}$ (Schaible and Ibaraki, 1983, section 3.4). However, this *parametric* solution method may not be optimal when the functions that make up the ratio have a certain amount of algebraic structure. Instead, one could use a *variable transformation* (VT) method to transform the FP into an equivalent convex problem that needs to be solved

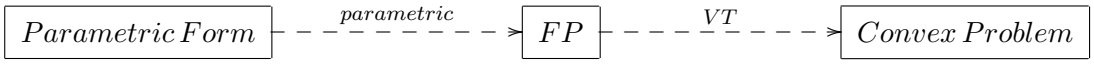
only *once* (Schaible, 1981; Schaible and Ibaraki, 1983, section 3.2). In other words, if the functions $\{N, D\}$ have some "nice" properties (algebraic structure) e.g., linearity, concavity, then solving a single convex problem may be more efficient than iterative search.



As the diagram suggests, we are concerned with solution methods for solving FPs since they provide a template for relating forms. In fact, the similarity of the parametric problem $\mathcal{F}_q$ to the general regularization problem $\mathcal{P}$ is the basis for deriving the framework.

1.2 Two Crucial Observations

When we have an FP, the goal is to choose the most appropriate form for solving the optimization problem. In our setting, we are given a specific parametric problem $\mathcal{P}$, which we would like to replace with an equivalent form. Therefore, it is logical to reverse the argument and recover back the FP associated with $\mathcal{P}$. The recovered FP can then be transformed into an equivalent convex problem $\mathcal{P}^*$ using the VT solution method. $\mathcal{P}^*$ would have a different parameter, which may have a more intuitive meaning than q.



If the functions $\{N, D\}$ have proper structure, then the derivation of $\mathcal{P}^*$ consists of these two steps, both justified by FPG theory.

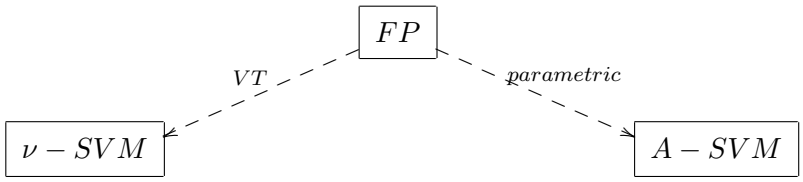
This observation is the blueprint for the general framework. However, in the context of FPG, the parametric problem $\mathcal{F}_q$ is solved repeatedly for a *sequence* of parameters $q_1, q_2, \ldots, \bar{q}$. More precisely, in order to maximize the fraction $N(x)/D(x)$, we are really searching for the maximal value of q (q_{max}) which satisfies the expression $\max\{N(x) - q \cdot D(x)\} \geq 0$. For any $q > q_{\max}$, the expression would be negative (zero, if the feasible set contains a null vector) since we cannot increase the fraction value further. Consequently, if we apply the theory verbatim, we can establish equivalence between two forms only for a very particular pair of parameters, namely q_{max} and q_0^*, where q_0^* is the corresponding parameter for the convex problem, which is solved only once. Formally, we refer to this special case of the framework as the *upper bound problem* (UBP):

> *Given an optimization problem $\mathcal{P}$ with objective of the type $\{\max P_q(x) = N(x) - q \cdot D(x)\}$ subject to some set of constraints, we wish to determine a finite upper bound on the parameter q. In other words, the goal is to find a value $q_{\max}$ such that for any $q \geq q_{\max}$ either $\max P_q(x) = \max P_{q_{\max}}(x)$, or $\max P_q(x) < 0$.*

We observe that it is preferable to solve the UBP first, i.e., establish equivalence for a specific pair of parameters and then generalize the proof to the entire range. Once we formulate the UBP for the original form $\mathcal{P}$, the intuitive link to FPG becomes clear: in the context of the UBP, $\mathcal{P}$ is the parametric form associated with a particular FP. This is the second critical observation.

1.3 Application to SVMs

We apply the proposed framework in two concrete settings. First, we consider SVMs for classification, a well researched problem for which various soft-margin dual forms have been examined. Our main result relates the *second order cone program* (SOCP) form (Vapnik, 1998, chap. 10.2), henceforth denoted A-SVM, and ν-SVM (Schölkopf et al., 2000) through a common FP. The proof is based on a simple application of the framework, since ν-SVM and A-SVM are just distinct solution methods:



The form ν-SVM was originally proposed as an alternative to the standard *quadratic program* (QP) form, which has a non-intuitive parameter C. Moreover, Chang and Lin (2001) later proved that these forms have the same optimal solution sets. Since the QP form, denoted C-SVM, is easily related to A-SVM (Vapnik, 1998), we actually derive the same result. However, we are compelled to formulate the equivalence relation in terms of A-SVM since the objective function of C-SVM is not *scale invariant*, i.e., the structure is not appropriate. On the other hand, our proof is markedly less convoluted, and the technique can be adapted to other problems.

For our second setting, we consider the *learning using privileged information* (LUPI) paradigm, introduced by Vapnik (2006) to incorporate elements of "teaching" into machine learning algorithms. LUPI is a general paradigm, but initially it has been developed for SVM-type algorithms (Vapnik, 2006; Vapnik et al., 2009; Vapnik and Vashist, 2009). From an optimization perspective, the extended method, named SVM+, has a similar structure to the SVM problem.

Once again, we concentrate on dual forms for classification. However, in this case we introduce new SOCP and ν-SVM+ formulations. The derivation is succinct since it follows from the FPG framework, and we obtain bounds on the range of the cost parameter as a by-product. In practice, solving an SVM+ problem may require the tuning of twice as many parameters as the analogous SVM problem. Hence, it is useful to have a form with a more intuitive parameter to search over.

1.4 Outline

The rest of the paper is organized as follows: we begin by reviewing relevant fractional programming theory in Section 2. We then derive the general framework in Section 3. We develop the framework for the upper bound problem (Section 3.1) before generalizing the method to show equivalence for the entire optimal solution set (Section 3.2). In addition, we elaborate on the class of problems with suitable algebraic structure. The general method (stated in Theorem 9) for transforming $\mathcal{P}$ into $\mathcal{P}^*$ is our primary theoretical contribution in this paper.

Subsequent sections are devoted to specific SVM-type formulations. In Section 4, we review various soft-margin dual SVM forms for classification. We then apply the framework

in Section 5 to demonstrate how these forms could be related without any additional effort. This result both clarifies and improves existing SVM theory. In Section 6, we apply the same technique to SVM+ forms. We derive ν-SVM+, a new dual form for the SVM+ algorithm, and obtain bounds on the search range of the parameters. To illustrate that ν-SVM+ can be used in practice, we extend the decomposition method proposed for C-SVM+ (Pechyony et al., 2010) in Section 7. We summarize our work in Section 8, and relegate some of the technical details to the appendices.

2. Fractional Programming Review

We review some basic fractional programming (FPG) theory. These results, which we reference throughout the paper, are discussed in more detail in the FPG literature (Schaible, 1981; Schaible and Ibaraki, 1983; Ibaraki, 1981).

2.1 Fractional Programs

A fractional program is an optimization problem with a ratio of functions appearing in the objective.

Definition 1 (Fractional Program) *Let N, D, denote real valued functions defined on the subset S of the n-dimensional Euclidean space $\Re^n$. The optimization problem $\mathcal{F}$ is called a fractional program (FP)*[1].

$$(\mathcal{F}) \qquad \max \left\{ R(x) = \frac{N(x)}{D(x)} \,\middle|\, x \in S \right\} \quad \text{where } S \subseteq \Re^n, D(x) > 0 \tag{2.1}$$

Based on the properties of the functions that make up the ratio and the feasible region S, we can distinguish between different types of FPs. Since we focus on SVM-type forms, we are particularly interested in concave fractional programs.

Definition 2 (Concave Fractional Program) *Let $N(x)$ be a concave function, and $D(x)$ a convex function, defined on the convex set $S \subseteq \Re^n$. In addition, assume that $N(x) \geq 0$ for non-affine $D(x)$. Then $\mathcal{F}$ is called a concave fractional program (C.F.P).*

There are multiple approaches to solving a C.F.P. We outline two of these strategies; the variable transformation method, which replaces the C.F.P with an equivalent convex program, and the parametric approach which requires iterative search.

2.2 Variable Transformation Solution Method

When the numerator and/or the denominator of the ratio $R(x)$ have a certain amount of algebraic structure, it may be appropriate to transform the C.F.P into an equivalent convex program[2] by applying the variable transformation

$$u = \frac{1}{D(x)} x,\ t = \frac{1}{D(x)},\ x \in S \tag{2.2}$$

1. Maximum can be replaced by supremum.
2. Sometimes called a concave program; terminology is inconsistent in the literature.

This is a generalization of the transformation initially suggested by Charnes and Cooper (1962) for reducing a linear fractional program (where both $N(x), D(x)$ are linear, and S is polyhedral) to a linear program. The equivalence is established with the following lemmas (Schaible, 1976):

Lemma 3 (Equivalence) *Let $\mathcal{F}$ be a C.F.P and consider the problem $\mathcal{F}'$:*

$$(\mathcal{F}') \qquad \max\left\{R'(u,t) = t \cdot N(\frac{u}{t}) \,\middle|\, u \in \Re^n,\, t > 0,\, \frac{u}{t} \in S,\, t \cdot D(\frac{u}{t}) \leq 1\right\} \tag{2.3}$$

(a) *$\mathcal{F}'$ is a convex program.*
(b) *If $\max\{N(x) \,|\, x \in S\} > 0$, then $\mathcal{F}$ has an optimal solution if and only if $\mathcal{F}'$ has one, and optimal solutions of $\mathcal{F}$ and $\mathcal{F}'$ are related by the variable transformation* (2.2).

Proof See (Schaible, 1976, 1974). ■

For an FP with an affine function $D(x)$, we can relax the positivity assumption in (b), and replace the last inequality in $\mathcal{F}'$ with an equality.

Lemma 4 (Equivalence) *Let $\mathcal{F}$ be a C.F.P where $D(x)$ is affine, and consider the problem $\mathcal{F}''$:*

$$(\mathcal{F}'') \qquad \max\left\{R''(u,t) = t \cdot N(\frac{u}{t}) \,\middle|\, u \in \Re^n,\, t > 0,\, \frac{u}{t} \in S,\, t \cdot D(\frac{u}{t}) = 1\right\} \tag{2.4}$$

(a) *$\mathcal{F}''$ is a convex program.*
(b) *$\mathcal{F}$ has an optimal solution if and only if $\mathcal{F}''$ has one, and optimal solutions of $\mathcal{F}$ and $\mathcal{F}''$ are related by the variable transformation* (2.2).

Proof See (Schaible, 1976, 1974). ■

If we also assume that the functions $N(x), D(x)$ are differentiable on S, then the ratio $R(x)$ is a semi-strictly quasi-concave function, and hence every local maximum of $R(x)$ is a global maximum (see, for example, Avriel et al., 1988). In this case, the optimal solutions of $\mathcal{F}$ and $\mathcal{F}'$ ($\mathcal{F}''$) are unique.

2.3 Parametric Solution Method

Another class of solution methods for C.F.Ps is based on the auxiliary problem $\mathcal{F}_q$ defined by

$$(\mathcal{F}_q) \qquad \max\{N(x) - qD(x) \,|\, x \in S\} \tag{2.5}$$

where $q \in \Re$ is a parameter. Let $\Phi(q)$ denote the optimal value of $\mathcal{F}_q$. Theorem 5 shows that $\mathcal{F}_q$ is closely related to $\mathcal{F}$.

Theorem 5 *Let $\mathcal{F}$ be a C.F.P where, in addition, $N(x), D(x)$ are continuous, and let $\mathcal{F}_q$ be the corresponding parametrized problem. Then,*
(a) *$\Phi(q)$ is continuous and convex over $q \in \Re$.*
(b) *$\Phi(q)$ is strictly decreasing over $q \in \Re$ i.e., if $q_1 < q_2$ then $\Phi(q_2) < \Phi(q_1)$, $q_1, q_2 \in \Re$.*
(c) *$\Phi(q) = 0$ has a unique solution, say q_0.*
(d) *$\Phi(q_0) = \max\{N(x) - q_0 D(x) \,|\, x \in S\} = 0$ if and only if $q_0 = \frac{N(x_0)}{D(x_0)} = \max\{R(x) \,|\, x \in S\}$.*

Proof See (Dinkelbach, 1967; Jagannathan, 1966). A typical $\Phi(q)$ curve is shown in (Schaible and Ibaraki, 1983, Fig.1). ■

Theorem 5(c,d) indicates that an optimal solution of $\mathcal{F}_q$ is also optimal for $\mathcal{F}$ when $\Phi(q) = 0$. Therefore, instead of solving $\mathcal{F}$, we could solve $\Phi(q) = 0$ using a parametric approach. Parametric procedures may be preferable when $\mathcal{F}_q$ is a more tractable problem than $\mathcal{F}$ (for more details, see Ibaraki, 1981).

3. Fractional Programming Framework

FPG theory provides a template for relating forms through various transformations. We could exploit these transformations to derive equivalent forms if the structure of the functions is suitable. In particular, Theorem 5 gives us context in which the original problem $\mathcal{P}$ we wish to transform and the auxiliary problem $\mathcal{F}_q$ (2.5) are identical.

3.1 Upper Bound Problem

We consider the problem $\mathcal{P}$ with the parameter fixed to $q = q_{\max}$ under the set of assumptions imposed in the previous section and summarized in Theorem 6 below. We define $q_{\max}$ to be the maximal value of q which satisfies the expression $\max\{N(x) - q \cdot D(x)\} \geq 0$. It follows from the results above that this unknown value must exist (be finite), and that for any $q > q_{\max}$ the expression would be negative. Crucially, in the context of searching for $q_{\max}$, $\mathcal{P}$ has the exact form of the auxiliary problem $\mathcal{F}_q$.

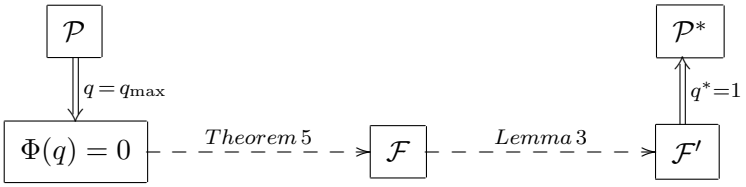
We could solve this *upper bound problem* (UBP) using a parametric procedure, that is, repeatedly solve $\mathcal{F}_q$ for a sequence of parameters $q_1, q_2, \ldots, \bar{q} = q_{\max}$ (Ibaraki, 1981). However, based on Theorem 5, we could also convert $\mathcal{F}_q$ back to the corresponding FP $\mathcal{F}$ and then transform $\mathcal{F}$ to the convex problem $\mathcal{F}'$ ($\mathcal{F}''$) using one of the Lemmas. The convex problem is solved just once since it does not have a parameter (it is implicitly set to one, see below).

The second approach is relevant in our setting since we can trace the same steps to replace $\mathcal{P}_{q=q_{\max}}$ with an equivalent form $\mathcal{P}^*$, where q^* is fixed to some value.

Theorem 6 *Suppose that the given optimization problem $\mathcal{P}$ has an objective function of the type $\{P_q(x) = N(x) - q \cdot D(x)\}$ where $N(x), D(x)$ are continuous functions defined on the convex set $S \subseteq \Re^n$ specified through the constraints of $\mathcal{P}$, and $q \in \Re$ is a parameter. In addition, assume that $N(x)$ is concave and $D(x)$ is convex, $D(x) > 0$, and $N(x) \geq 0$ for non-affine $D(x)$. Let $\Phi(q)$ denote the optimal value of the expression* $\max\{P_q(x)\}$. *Then,*
(a) *Solving $\mathcal{P}$ with $q = q_{\max}$ is equivalent to solving $\Phi(q) = 0$ (an identical solution vector is obtained).*
(b) *Solving $\Phi(q) = 0$ is equivalent to solving the C.F.P $\mathcal{F}$* (2.1).
(c) *By applying the variable transformation* (2.2) *to $\mathcal{F}$, we can obtain the equivalent convex problem $\mathcal{F}'$* (2.3) *or $\mathcal{F}''$* (2.4) *with the additional variable t.*

Proof **(a)** By definition of the UBP. **(b)** Follows from Theorem 5, and Definitions 1, 2. **(c)** Follows from Lemma 3 and Lemma 4. ■

Theorem 6 is a special case of the framework for transforming $\mathcal{P}$ into $\mathcal{P}^*$.



Moreover, the Theorem outlines a constructive method to determine the actual upper bound on the range of the parameter q. For a given $\mathcal{P}$, we could solve $\mathcal{F}'$ ($\mathcal{F}''$) once and then plug the solution vector x_0 into the fraction to get the upper bound: $q_{\max} = \frac{N(x_0)}{D(x_0)}$.

This approach is very effective when $\mathcal{F}'$ can be solved efficiently. However, $\mathcal{F}'$ is inherently more complex than $\mathcal{P}$ due to the extra constraint $t \cdot D(u/t) \leq 1$ and the additional variable t. In fact, this is a critical issue since the (implied) purpose of the framework is to produce equivalent forms that are comparable in complexity. If the particular problem $\mathcal{F}'$ is not "simple", then the general form $\mathcal{P}^*$ is not going to be either. Therefore, we describe the conditions under which $\mathcal{F}'$ can be simplified.

First, we observe that the additional variable t can be partially eliminated if the objective function of $\mathcal{P}$ is *scale invariant*.

Definition 7 (Scale Invariant Function) *A function is scale invariant if the equality*

$$f(\Delta x) = \Delta^d f(x) \tag{3.1}$$

holds for any scale factor Δ and some (fixed) choice of exponent d. The degree of invariance is denoted by d.

In particular, if the degree of invariance is one, then both the objective and the extra constraint of $\mathcal{F}'$ simplify to:

$$t \cdot N(\frac{u}{t}) = t(\frac{1}{t}) \cdot N(u) = N(u) \tag{3.2}$$

$$t \cdot D(\frac{u}{t}) = t(\frac{1}{t}) \cdot D(u) = D(u) \leq 1 \tag{3.3}$$

Typical examples in the FPG literature (Schaible, 1981, example 1) satisfy this property implicitly.

The second observation concerns the (simplified) extra constraint $D(u) \leq 1$. Since it is desirable to have a linear (rather than nonlinear) constraint, it is sometimes useful to flip the ratio of $\mathcal{F}$ and derive $\mathcal{F}'$ from the fraction $R(x) = \frac{-D(x)}{N(x)}$.

Remark 8 *If $N(x)$ is an affine (linear) function but $D(x)$ is not, it is preferable to flip the ratio of $\mathcal{F}$, before applying the variable transformation, in order to have an affine denominator. Moreover, if $D(x)$ is scale invariant [3], then Lemma 4 is invoked to obtain a convex problem with an additional linear constraint.*

3. Henceforth we assume that scale invariance is of degree one, unless stated otherwise.

As we show in Section 5, this remark is very useful for SVM-type forms. In general, the FPG results in Section 2 are well-suited for manipulating forms whose objective is scale invariant. Hence, we insist on this assumption to develop the framework below. However, using a slightly different transformation (Mond and Craven, 1975), it is possible to extend the framework to functions with higher degree of invariance and non-scale invariant functions as well. We discuss this approach in Appendix A.

3.2 General Framework

We could generalize Theorem 6 to derive equivalent forms for the entire parameter range. Although this is the main conceptual result of the paper, the proof requires only minor technical detail on top of the analysis in Section 3.1.

We consider the problem $\mathcal{P}$ with the parameter q fixed to an arbitrary value q_θ (in the relevant range of q). It follows from Theorem 5 that $\forall q_\theta : 0 < q_\theta \leq q_{\max}$, $\Phi(q_\theta) \geq \Phi(q_{max}) = 0$. Therefore, instead of solving $\Phi(q) = 0$ as above, we solve $\Phi(q) - \theta = 0$, $\theta \geq 0$. Accordingly, $N(x)$ is replaced with $N(x) - \theta$. Since q_θ is arbitrary, we can assume without loss of generality (w.l.o.g) that theta is fixed to some value in the feasible range. In other words, $\exists q_\theta : \Phi(q_\theta) = \theta$.

Theorem 9 *Suppose all the assumptions of Theorem 6 hold. Let $\Phi(q)$ denote the optimal value of the expression $\max\{P_q(x)\}$, where $P_q(x)$ is scale invariant, and assume that theta is feasible. Then,*
(a) *Solving $\mathcal{P}$ with $0 < q_\theta \leq q_{\max}$ is equivalent to solving $\Phi(q) - \theta = 0$ (an identical solution vector is obtained).*
(b) *Solving $\Phi(q) - \theta = 0$, is equivalent to solving the C.F.P*

$$(\mathcal{F}_\theta) \qquad \max \left\{ R_\theta(x) = \frac{N(x) - \theta}{D(x)} \,\middle|\, x \in S \right\} \tag{3.4}$$

If $N(x)$ is an affine function but $D(x)$ is not, we flip the ratio and solve

$$(\mathcal{F}_{\theta\star}) \qquad \max \left\{ R_{\theta\star}(x) = \frac{-D(x)}{N(x) - \theta} \,\middle|\, x \in S \right\} \tag{3.5}$$

(c) *By applying the variable transformation (2.2) to $\mathcal{F}_\theta$ or (3.6) to $\mathcal{F}_{\theta\star}$, we can obtain the equivalent convex problems $\mathcal{F}'_\theta$ and $\mathcal{F}''_\theta$ respectively*

$$u = \frac{1}{N(x) - \theta} x,\ t = \frac{1}{N(x) - \theta},\ x \in S \tag{3.6}$$

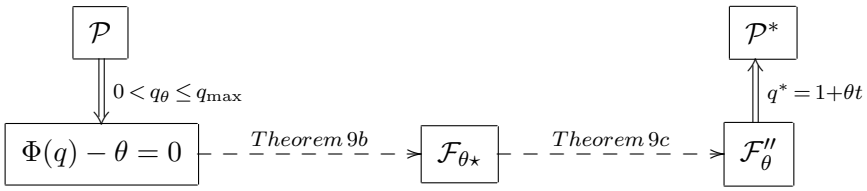
$$(\mathcal{F}'_\theta) \qquad \max \left\{ R'_\theta(u,t) = N(u) - \theta t \,\middle|\, u \in \Re^n,\ t > 0,\ \frac{u}{t} \in S,\ D(u) \leq 1 \right\} \tag{3.7}$$

$$(\mathcal{F}''_\theta) \qquad \max \left\{ R''_\theta(u,t) = -D(u) \,\middle|\, u \in \Re^n,\ t > 0,\ \frac{u}{t} \in S,\ N(u) = 1 + \theta t \right\} \tag{3.8}$$

Proof **(a)** By definition. **(b)** Follows from Theorem 5 and Definitions 1,2, where we have replaced $N(x)$ with $N(x) - \theta$ throughout. The ratio is flipped based on Remark 8. **(c)** Follows from Lemma 3 and Lemma 4, where the simplification occurs due to scale

invariance. ■

Theorem 9 is the general framework for transforming $\mathcal{P}$ into $\mathcal{P}^*$. For simplicity, we present the framework through the C.F.P $\mathcal{F}_{\theta\star}$. However, it may be logical to flip the ratio even if $N(x)$ is not affine (alternatively, we could replace $D(x)$ with $D(x)-\theta$ to begin with). When the variable t is confined to the constraints, the convex problem is easier to manipulate.



In practice, $\mathcal{F}''_\theta$ is simplified further by scaling, and removing t from the constraints if possible. Furthermore, once t has been fixed, we could substitute a parameter $q^* = 1 + \theta t$ instead of theta to get the standard form $\mathcal{P}^*$. This statement is not precise since it is impossible to predict the scaling factor which in general depends on the set of constraints. We elaborate on this issue with two concrete examples in Sections 5 and 6.

4. Soft-Margin Formulations for SVMs

The Support Vector Machine (SVM) is a widely used kernel learning algorithm. The basic idea of SVM is to find the hyperplane which separates the training examples with maximal margin (Vapnik and Lerner, 1963). To obtain non-linear decision boundaries, the optimal separating hyperplane can be constructed in the feature space induced by a kernel function (Boser et al., 1992). Non-separable (noisy) problems can be addressed by introducing a soft-margin formulation (Cortes and Vapnik, 1995).

SVM-type algorithms have been developed for many learning settings (Vapnik, 1998; Schölkopf and Smola, 2002). In this paper, we are concerned with the soft-margin formulations for classification.

4.1 The QP Form (C-SVM)

Suppose we have a set of ℓ observations $(x_1, \ldots, x_\ell)$ with corresponding labels $(y_1, \ldots, y_\ell)$ where, $x_i \in \Re^n$, $y_i \in \{-1, 1\}$. To find the optimal hyperplane in the non-separable case, we could solve the primal optimization problem (Cortes and Vapnik, 1995)

$$(P_C) \qquad \min_{\mathbf{w},b,\xi} \frac{1}{2}(\mathbf{w}\cdot\mathbf{w}) + C\sum_{i=1}^{\ell}\xi_i$$
$$\forall i,\ y_i((\mathbf{w}\cdot\mathbf{x}_i)+b) \geq 1-\xi_i,\ \xi_i \geq 0 \tag{4.1}$$

The standard approach for solving P_c is to construct the Lagrangian and obtain the dual form. In vector notation, the problem is a *quadratic program* (QP)

$$\max_{\boldsymbol{\alpha}} D_C(\boldsymbol{\alpha}) = \mathbf{e}^T\boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha}^T Q\boldsymbol{\alpha}$$
$$\mathbf{y}^T\boldsymbol{\alpha} = 0$$
$$\forall i,\ 0 \leq \alpha_i \leq C \tag{4.2}$$

where vectors are boldface, $\mathbf{e}$ is an ℓ-by-1 vector of ones, Q is an ℓ-by-ℓ *positive semi-definite* (p.s.d) matrix, $Q_{i,j} = y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$, and K is the kernel. Henceforth we assume that the variable bounds hold $\forall i$, unless specified otherwise. The solution vector $\boldsymbol{\alpha}^0$ of (4.2) defines the generalized optimal hyperplane (4.3), and the threshold b is chosen to satisfy the *Karush-Kuhn-Tucker* (KKT) conditions.

$$f(\mathbf{x}) = \sum_{i=1}^{\ell} \alpha_i^0 y_i K(\mathbf{x}, \mathbf{x}_i) + b \tag{4.3}$$

Since the parameter appears in the constraints of the dual (4.2), it is useful to factor it out of the constraints and into the objective. If we substitute variables, that is, $(\forall i)$ replace α_i with $C \cdot \alpha_i$, and divide by C, then we obtain the normalized dual form (4.4), and the corresponding decision function (4.5).

$$\begin{aligned} (D_C) \qquad \max_{\boldsymbol{\alpha}} \; & D_C(\boldsymbol{\alpha}) = \mathbf{e}^T\boldsymbol{\alpha} - C \cdot \frac{1}{2}\boldsymbol{\alpha}^T Q \boldsymbol{\alpha} \\ & \mathbf{y}^T\boldsymbol{\alpha} = 0,\; 0 \le \alpha_i \le 1 \end{aligned} \tag{4.4}$$

$$f_C(\mathbf{x}) = C \cdot \sum_{i=1}^{\ell} \alpha_i^0 y_i K(\mathbf{x}, \mathbf{x}_i) + b \tag{4.5}$$

4.2 The SOCP Form (A-SVM)

The primal form P_C (4.1) is often chosen by default to solve support vector classification problems (Hsu et al., 2003). However, P_C was originally proposed as a computationally efficient alternative to the *second order cone program* (SOCP) that arises when we attempt to find the specific Δ-margin hyperplane in the non-separable case (Vapnik, 1998):

$$\begin{aligned} (P_\Delta) \qquad \min_{\mathbf{w},b,\xi} \; & \sum_{i=1}^{\ell} \xi_i \\ & \forall i,\; y_i((\mathbf{w}\cdot\mathbf{x}_i) + b) \ge 1 - \xi_i,\; \xi_i \ge 0 \\ & (\mathbf{w}\cdot\mathbf{w}) \le \frac{1}{\Delta^2} \end{aligned} \tag{4.6}$$

The standard approach is again to construct the Lagrangian, and find the saddle point. If we denote $A = \frac{1}{\Delta}$, and generalize to the nonlinear case, we obtain the SOCP D_A (4.7), and the decision function (4.8), defined by the solution vector $\boldsymbol{\alpha}^0$.

$$\begin{aligned} (D_A) \qquad \max_{\boldsymbol{\alpha}} \; & D_A(\boldsymbol{\alpha}) = \mathbf{e}^T\boldsymbol{\alpha} - A\sqrt{\boldsymbol{\alpha}^T Q \boldsymbol{\alpha}} \\ & \mathbf{y}^T\boldsymbol{\alpha} = 0,\; 0 \le \alpha_i \le 1 \end{aligned} \tag{4.7}$$

$$f_A(\mathbf{x}) = \frac{A}{\sqrt{(\boldsymbol{\alpha}^0)^T Q \boldsymbol{\alpha}^0}} \sum_{i=1}^{\ell} \alpha_i^0 y_i K(\mathbf{x}, \mathbf{x}_i) + b \tag{4.8}$$

Having obtained $\boldsymbol{\alpha}^0$ (assuming $\boldsymbol{\alpha}^0 \neq 0$) for some A, we could define the parameter $C = {}^{A}/\sqrt{(\boldsymbol{\alpha}^0)^T Q \boldsymbol{\alpha}^0}$. In this case, the solutions of D_A and D_C coincide (Vapnik, 1998, Ch. 10.2).

4.3 ν-SVM: A Form with an Intuitive Parameter

Considerable effort has been devoted to the implementation of efficient optimization methods for solving C-SVM (4.2) (Platt, 1999; Bottou and Lin, 2007). However, this form has an obvious drawback since C is not an intuitive parameter to search over, as the relevant search range can vary greatly between datasets. Schölkopf et al. (2000) tried to address this issue by introducing a primal form with an intuitive parameter:

$$\begin{aligned}(P_\nu) \qquad & \min_{\mathbf{w},b,\xi} \frac{1}{2}(\mathbf{w}\cdot\mathbf{w}) - \nu\rho + \frac{1}{\ell}\sum_{i=1}^{\ell}\xi_i \\ & \forall i,\ y_i((\mathbf{w}\cdot\mathbf{x}_i)+b) \geq \rho - \xi_i,\ \xi_i \geq 0 \\ & \rho \geq 0 \end{aligned} \tag{4.9}$$

where $0 \leq \nu \leq 1$, and ν is a bound on the fraction of margin errors and number of support vectors.

Subsequently, the relation between ν-SVM and C-SVM was studied both geometrically (Crisp and Burges, 1999; Bennett and Bredensteiner, 2000), and analytically (Chang and Lin, 2001). Chang and Lin (2001) proved that these problems have the same optimal solution set and showed that the parameter ν has concrete bounds $0 < \nu_{min} \leq \nu \leq \nu_{max} \leq 1$. As a consequence of this result, they indirectly obtained bounds on the C parameter as well since C_{max} can be derived from ν_{min} and the solution vector of the dual $D_{\nu_{min}}$ below. In a similar fashion, C_{min} follows from ν_{max}, where

$$\nu_{\max} = \frac{2 \times \min(\#y = 1,\ \#y = -1)}{\ell}$$

and $\#y$ denotes the number of examples with a particular label.

For computational reasons, Chang and Lin (2001, 2011) recommend to solve the scaled dual,

$$\begin{aligned}(D_\nu) \qquad & \min_{\boldsymbol{\alpha}} \frac{1}{2}\boldsymbol{\alpha}^T Q \boldsymbol{\alpha} \\ & \mathbf{e}^T\boldsymbol{\alpha} = \nu\ell,\ \mathbf{y}^T\boldsymbol{\alpha} = 0,\ 0 \leq \alpha_i \leq 1 \end{aligned} \tag{4.10}$$

On the other hand, in the course of our FPG-based derivation, we obtain the reparametrized form (Crisp and Burges, 1999, section 2.1),

$$\begin{aligned}(D_\mu) \qquad & \min_{\boldsymbol{\alpha}} \frac{1}{4}\boldsymbol{\alpha}^T Q \boldsymbol{\alpha} \\ & \mathbf{e}^T\boldsymbol{\alpha} = 2,\ \mathbf{y}^T\boldsymbol{\alpha} = 0,\ 0 \leq \alpha_i \leq \mu \end{aligned} \tag{4.11}$$

which is equivalent to D_ν by trivial variable substitution.

5. Fractional Programming Framework for SVMs

We would like to apply the framework developed in Section 3 to the dual SVM forms discussed above. Both D_C (4.4) and D_A (4.7) have the exact form of the auxiliary problem

$\mathcal{F}_q$ (2.5). However, D_C is not scale invariant since the objective function is composed of linear and quadratic terms. Therefore, we are compelled to work with D_A which does have this property.

In order to refine the theory for this particular setting, we must verify that the assumptions in Theorem 6 (Thm. 9) hold for D_A. Indeed, $N(\boldsymbol{\alpha})$ is concave, $D(\boldsymbol{\alpha})$ is convex, and S is polyhedral. However, the requirement $D(\boldsymbol{\alpha}) > 0$ is not satisfied. This condition holds for D_A if we assume that the kernel matrix is strictly *positive definite* (p.d), and add the trivially enforced constraint $\mathbf{e}^T\boldsymbol{\alpha} \geq \varepsilon$, where $\varepsilon > 0$ is some small constant[4]. For example, we could use the Gaussian kernel with distinct examples (Schölkopf and Smola, 2002). In the notation of Section 3:

$$N(\mathbf{x}) = \mathbf{e}^T\mathbf{x},\ D(\mathbf{x}) = \sqrt{\mathbf{x}^TQ\mathbf{x}},\ q = A$$
$$S \equiv \{\mathbf{y}^T\mathbf{x} = 0,\ \mathbf{e}^T\mathbf{x} \geq \varepsilon,\ 0 \leq x_i \leq 1\}$$

Taking Remark 8 into account, we apply Theorem 9 with the ratio of the recovered FP (C.F.P) flipped. C.F.Ps of this form (5.1) can often arise in stochastic nonlinear programming applications and portfolio selection problems (for a very similar example, see Schaible, 1981). Due to the affine denominator, we can invoke Lemma 4 to obtain a convex problem without a nonlinear constraint. In fact, we get exactly D_μ (4.11), the reparametrized form of ν-SVM.

Theorem 10 *Suppose that the kernel matrix K is p.d, and add the constraint $\mathbf{e}^T\boldsymbol{\alpha} \geq \theta + \varepsilon$ to the optimization problem D_A. Let $\Phi(A)$ denote the optimal value of D_A, and assume that theta is feasible (i.e., $\exists A_\theta : \Phi(A_\theta) = \theta$). Then,*
***(a)** Solving $\Phi(A) - \theta = 0$, is equivalent to solving the C.F.P:*

$$(F_{A_\theta}) \qquad \max\left\{ R_{A_\theta}(\mathbf{x}) = \frac{-\sqrt{\mathbf{x}^TQ\mathbf{x}}}{\mathbf{e}^T\mathbf{x} - \theta} \,\middle|\, \mathbf{x} \in S \right\}$$
$$\text{where } S \equiv \{\mathbf{y}^T\mathbf{x} = 0,\ \mathbf{e}^T\mathbf{x} \geq \theta + \varepsilon,\ 0 \leq x_i \leq 1\} \tag{5.1}$$

***(b)** By applying the variable transformation* (5.2) *to F_{A_θ}, we can obtain the equivalent convex problem F''_{A_θ} with the additional variable t:*

$$\boldsymbol{\alpha} = \frac{1}{\mathbf{e}^T\mathbf{x} - \theta}\mathbf{x},\ t = \frac{1}{\mathbf{e}^T\mathbf{x} - \theta} \tag{5.2}$$

$$(F''_{A_\theta}) \qquad \max_{\boldsymbol{\alpha},t} R''_{A_\theta}(\boldsymbol{\alpha}, t) = -\frac{1}{2}\sqrt{\boldsymbol{\alpha}^TQ\boldsymbol{\alpha}}$$
$$\mathbf{e}^T\boldsymbol{\alpha} = 2 + \theta t,\ \mathbf{y}^T\boldsymbol{\alpha} = 0,\ t > 0$$
$$0 \leq \alpha_i \leq t \tag{5.3}$$

4. Select some index i and set $\varepsilon \leq \alpha_i \leq 1$.

Proof **(a)** Follows from Theorem 9(a),(b). **(b)** It follows from Lemma 4, that (5.2) applied to F_{A_θ} yields the functions and constraints

$$\frac{1}{2}t \cdot N(\frac{\boldsymbol{\alpha}}{t}) = -\frac{1}{2}\sqrt{\boldsymbol{\alpha}^T Q \boldsymbol{\alpha}}$$
$$\frac{1}{2}t \cdot D(\frac{\boldsymbol{\alpha}}{t}) = \frac{1}{2}t \cdot (\frac{\mathbf{e}^T\boldsymbol{\alpha}}{t} - \theta) = \frac{\mathbf{e}^T\boldsymbol{\alpha} - \theta t}{2} = 1 \tag{5.4}$$
$$t > 0,\ \frac{\mathbf{y}^T\boldsymbol{\alpha}}{t} = 0,\ \frac{\mathbf{e}^T\boldsymbol{\alpha}}{t} \geq \theta + \varepsilon,\ 0 \leq \frac{\alpha_i}{t} \leq 1 \tag{5.5}$$

where the appended constant 1/2 has no effect on the validity of the lemma. Simplifying (5.4),(5.5) and dropping the redundant constraint $\mathbf{e}^T\boldsymbol{\alpha} \geq (\theta + \varepsilon)t$, leads to the convex problem F''_{A_θ}. ■

If we minimize the squared objective $[R''_{A_\theta}(\boldsymbol{\alpha}, t)]^2$, F''_{A_θ} becomes a quadratic problem subject to linear constraints, which we could solve using any standard technique. Having obtained the solution $\{\boldsymbol{\alpha}_\theta, t_\theta\}$, we could scale alphas by $\frac{2+\theta t_\theta}{2}$ and reparametrize the problem. As the following theorem shows, this leads to the familiar form D_μ (4.11), with the parameter value $\mu = \frac{2t_\theta}{2+\theta t_\theta} \leq 1$.

Theorem 11 *Let $\{\boldsymbol{\alpha}_\theta, t_\theta\}$ denote the solution of the convex problem F''_{A_θ} where theta is a fixed constant from the feasible range. It follows that,*
(a) *$\forall\theta \geq 0$, $t_\theta \leq 1$.*
(b) *Solving F''_{A_θ} is equivalent to solving the form D_μ with $\mu = \frac{2t_\theta}{2+\theta t_\theta} \leq 1$.*

Proof **(a)** Suppose $t_\theta > 1$, and the bound is tight, that is $\max_i \alpha_i = t_\theta$. Then we could divide every alpha by t_θ and increase the objective value. **(b)** Reparametrize F''_{A_θ} as follows. Multiply every alpha by $\frac{2+\theta t}{2}$ and substitute $(\forall i)\ \alpha_i = (\frac{2+\theta t}{2})\alpha_i$, to obtain the scaled problem

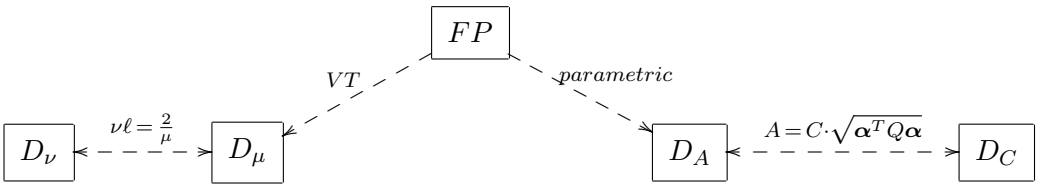
$$\begin{aligned} \max_{\boldsymbol{\alpha}}\ & -\frac{1}{2}(\frac{2+\theta t}{2})\sqrt{\boldsymbol{\alpha}^T Q \boldsymbol{\alpha}} \\ & \mathbf{e}^T\boldsymbol{\alpha} = 2,\ \mathbf{y}^T\boldsymbol{\alpha} = 0,\ t > 0 \\ & 0 \leq \alpha_i \leq \frac{2t}{2+\theta t} \end{aligned} \tag{5.6}$$

Let $\mu = \frac{2t}{2+\theta t}$. Since the inequality $2 + \theta t \geq 2t$ holds for all $\{\theta, t\}$ such that $\theta \geq 0, t \leq 1$, and from part (a), $t_\theta \leq 1$, we know that $\mu \leq 1$ holds. Problems (5.3) and (5.6) have the same unique solution. Therefore, we could solve F''_{A_θ}, and fix the parameter $t = t_\theta$. If we minimize the squared objective (divided by the constant $\frac{2+\theta t_\theta}{2}$) we obtain D_μ. ■

Theorems 10 and 11 establish that for every A_θ such that $0 < A_\theta \leq A_{\max}$, the optimal solutions of D_{A_θ} and F''_{A_θ} are related through the FP F_{A_θ}. Since the functions N, D are differentiable on S, for each A_θ, the optimal solutions are unique.

Thus, considered under the fractional programming framework, D_A and D_μ (and by extension, C-SVM and ν-SVM) are in fact two distinct methods for solving the same FP,

and hence their solution sets are related by definition.



This is the essential benefit of using the framework, as we can obtain equivalent forms directly. Furthermore, the inferred "analytic" insight is in some sense complementary to the geometric interpretation of ν-SVM classifiers (Crisp and Burges, 1999; Bennett and Bredensteiner, 2000).

For the sake of completeness, we show that the upper bound on the parameter A follows from the solution of $F''_{A_{\theta=0}}$.

Corollary 12 (Upper Bound) *Suppose we are given the optimization problem D_A with a p.d kernel matrix K, and let $\boldsymbol{\alpha}_{\max} = \boldsymbol{\alpha}_{\theta=0}$ be the solution vector of the related convex problem $F''_{A_{\theta=0}}$, where we have fixed $\theta = 0$. If we denote $M = \max R''_{A_{\theta=0}}(\boldsymbol{\alpha})$, then the upper bound on the parameter A can be determined from M as follows:*

$$M = \max_{\boldsymbol{\alpha}} R''_{A_{\theta=0}}(\boldsymbol{\alpha}) = -\frac{1}{2}\sqrt{\boldsymbol{\alpha}_{\max}^T Q \boldsymbol{\alpha}_{\max}}$$

$$A_{\max} = \frac{\mathbf{e}^T \boldsymbol{\alpha}_{\max}}{\sqrt{\boldsymbol{\alpha}_{\max}^T Q \boldsymbol{\alpha}_{\max}}} = \frac{2}{-2M} = -\frac{1}{M}$$

If we include the zero vector, $\boldsymbol{\alpha} = 0$, in the feasible region, then for $A \geq A_{\max}$, $\max D_A(\boldsymbol{\alpha}) = 0$, else as $A \longrightarrow \infty$, $\max D_A(\boldsymbol{\alpha}) \longrightarrow -\infty$.

Proof This is a special case of Theorems 10 and 11. ■

6. Soft-Margin Formulations for SVM+

We consider the *learning using privileged information* (LUPI) paradigm, introduced by Vapnik (2006) to incorporate elements of "teaching" into machine learning algorithms. In human learning, a teacher who can provide the student with explanations, comments, and comparisons, has a very important and complex role. Although it is difficult to define the teacher's responsibility precisely, certain aspects can be approximated well in a classical supervised learning framework. One such aspect is to provide the student (algorithm) with additional information about each example during the learning (training) phase. The algorithm could then access and use this *privileged information* (PI) to learn an improved decision rule (function), that would later be evaluated on unseen examples to which PI has not been added. If the teacher is effective (can be trusted), then the algorithm with PI would require significantly fewer training examples than the algorithm without, to perform equally well. In other words, the *rate of convergence* to the optimal (Bayesian) solution would increase.

LUPI is a general paradigm, but initially it has been developed for SVM-type algorithms (Vapnik, 2006; Vapnik et al., 2009; Vapnik and Vashist, 2009). The extended method is

denoted as SVM+ and, from an optimization perspective, has a similar structure to the SVM problem.

6.1 The QP Form (C-SVM+)

Suppose that for each of our observations $(\mathbf{x}_1, \ldots, \mathbf{x}_\ell)$, we have some additional (privileged) information $(\mathbf{x}_1^*, \ldots, \mathbf{x}_\ell^*)$, which we hope could improve our decision rule. In the SVM+ method, we map vectors $\mathbf{x}_i$ into one (decision) space, and vectors $\mathbf{x}_i^*$ into another (correcting) space, where $\mathbf{x}_i, \mathbf{x}_i^*$ can have different dimensionality. To find the decision and correcting (slack) functions, $f(\mathbf{x}) = (\mathbf{w} \cdot \mathbf{x}) + b$ and $\varphi(\mathbf{x}^*) = (\mathbf{w}^* \cdot \mathbf{x}^*) + b^*$ respectively, we could solve the problem (Vapnik and Vashist, 2009, section 3.2):

$$
\begin{aligned}
(P_C+) \qquad \min_{\mathbf{w},b,\mathbf{w}^*,b^*} \; & \frac{1}{2}[(\mathbf{w} \cdot \mathbf{w}) + \gamma(\mathbf{w}^* \cdot \mathbf{w}^*)] + C \cdot \sum_{i=1}^{\ell}[(\mathbf{w}^* \cdot \mathbf{x}_i^*) + b^*] \\
& \forall i,\; y_i[(\mathbf{w} \cdot \mathbf{x}_i) + b] \geq 1 - [(\mathbf{w}^* \cdot \mathbf{x}_i^*) + b^*] \\
& \forall i,\; [(\mathbf{w}^* \cdot \mathbf{x}_i^*) + b^*] \geq 0
\end{aligned} \tag{6.1}
$$

where the user-specified parameter $\gamma > 0$ determines how much weight should be given to the privileged information. By setting γ close to zero, we could reject the PI and reproduce the SVM solution.

We can construct the Lagrangian for P_C+, and obtain the dual in vector notation:

$$
\begin{aligned}
\max_{\boldsymbol{\alpha},\boldsymbol{\beta}} \; & D(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \mathbf{e}^T\boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha}^T Q \boldsymbol{\alpha} - \frac{1}{2\gamma}(\boldsymbol{\alpha} + \boldsymbol{\beta} - C)^T K^*(\boldsymbol{\alpha} + \boldsymbol{\beta} - C) \\
& \mathbf{y}^T\boldsymbol{\alpha} = 0,\; \mathbf{e}^T(\boldsymbol{\alpha} + \boldsymbol{\beta} - C) = 0 \\
& 0 \leq \alpha_i,\; 0 \leq \beta_i
\end{aligned} \tag{6.2}
$$

where $Q_{i,j} = y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$, and $\{K, K^*\}$ are the kernels in decision and correcting space respectively. The solution vectors $\{\boldsymbol{\alpha}^0, \boldsymbol{\beta}^0\}$ of the dual define the decision and correcting functions:

$$
f(\mathbf{x}) = \sum_{i=1}^{\ell} \alpha_i^0 y_i K(\mathbf{x}, \mathbf{x}_i) + b \tag{6.3}
$$

$$
\varphi(\mathbf{x}^*) = \frac{1}{\gamma} \sum_{i=1}^{\ell} (\alpha_i^0 + \beta_i^0 - C) K^*(\mathbf{x}^*, \mathbf{x}_i^*) + b^* \tag{6.4}
$$

If we substitute variables, that is $(\forall i)$ replace α_i with $C \cdot \alpha_i$, β_i with $C \cdot \beta_i$, and divide by C, then we obtain the normalized dual form:

$$
\begin{aligned}
(D_C+) \qquad \max_{\boldsymbol{\alpha},\boldsymbol{\beta}} \; & D_C(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \mathbf{e}^T\boldsymbol{\alpha} - C \cdot \frac{1}{2}[\boldsymbol{\alpha}^T Q \boldsymbol{\alpha} + \frac{1}{\gamma}(\boldsymbol{\alpha} + \boldsymbol{\beta} - 1)^T K^*(\boldsymbol{\alpha} + \boldsymbol{\beta} - 1)] \\
& \mathbf{y}^T\boldsymbol{\alpha} = 0,\; \mathbf{e}^T(\boldsymbol{\alpha} + \boldsymbol{\beta} - 1) = 0 \\
& 0 \leq \alpha_i,\; 0 \leq \beta_i
\end{aligned} \tag{6.5}
$$

6.2 The SOCP Form (A-SVM+)

Solving the problem D_C+ may require the tuning of several parameters: $\{C, \gamma\}$ and the kernel parameters of two kernels. Hence, from a practical perspective, it would be useful to replace C with a more intuitive parameter. However, as in the SVM case, the objective function $D_C(\boldsymbol{\alpha}, \boldsymbol{\beta})$ is not scale invariant. Therefore, we introduce the extension of the primal SOCP formulation P_Δ to the SVM+ method:

$$
\begin{aligned}
(P_\Delta+) \qquad \min_{\mathbf{w},b,\mathbf{w}^*,b^*} & \sum_{i=1}^{\ell} [(\mathbf{w}^* \cdot \mathbf{x}_i^*) + b^*]_+ \\
& \forall i,\ y_i\left[(\mathbf{w} \cdot \mathbf{x}_i) + b\right] \geq 1 - [(\mathbf{w}^* \cdot \mathbf{x}_i^*) + b^*] \\
& (\mathbf{w} \cdot \mathbf{w}) + \gamma(\mathbf{w}^* \cdot \mathbf{w}^*) \leq \frac{1}{\Delta^2}
\end{aligned} \tag{6.6}
$$

where $(x)_+ = \max\{x, 0\}$. Note the slightly modified objective (due to the '+' function). Conceptually, the correcting function is not limited to modeling strictly non-negative variables. Mathematically, the dual differs only in the additional upper bound on betas, see equation (6.8). We can replace the '+' function, and obtain an equivalent problem:

$$
\begin{aligned}
\min_{\mathbf{w},b,\mathbf{w}^*,b^*,\xi^*} & \sum_{i=1}^{\ell} [(\mathbf{w}^* \cdot \mathbf{x}_i^*) + b^* + \xi_i^*] \\
& \forall i,\ y_i\left[(\mathbf{w} \cdot \mathbf{x}_i) + b\right] \geq 1 - [(\mathbf{w}^* \cdot \mathbf{x}_i^*) + b^*] \\
& \forall i,\ (\mathbf{w}^* \cdot \mathbf{x}_i^*) + b^* + \xi_i^* \geq 0,\ \xi_i^* \geq 0 \\
& (\mathbf{w} \cdot \mathbf{w}) + \gamma(\mathbf{w}^* \cdot \mathbf{w}^*) \leq \frac{1}{\Delta^2}
\end{aligned} \tag{6.7}
$$

If we set $A = \frac{1}{\Delta}$, and construct the Lagrangian for (6.7), we can derive the following form in vector notation (see Appendix C):

$$
\begin{aligned}
\max_{\boldsymbol{\alpha},\boldsymbol{\beta}} & D_A(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \mathbf{e}^T\boldsymbol{\alpha} - A\sqrt{\boldsymbol{\alpha}^T Q \boldsymbol{\alpha} + \frac{1}{\gamma}(\boldsymbol{\alpha} + \boldsymbol{\beta} - 1)^T K^* (\boldsymbol{\alpha} + \boldsymbol{\beta} - 1)} \\
& \mathbf{y}^T\boldsymbol{\alpha} = 0,\ \mathbf{e}^T(\boldsymbol{\alpha} + \boldsymbol{\beta} - 1) = 0 \\
& 0 \leq \alpha_i,\ 0 \leq \beta_i \leq 1
\end{aligned} \tag{6.8}
$$

Having obtained the solution vectors $\{\boldsymbol{\alpha}^0, \boldsymbol{\beta}^0\}$ of (6.8) for some A, we could define the parameter

$$
C = \frac{A}{\sqrt{(\boldsymbol{\alpha}^0)^T Q \boldsymbol{\alpha}^0 + \frac{1}{\gamma}(\boldsymbol{\alpha}^0 + \boldsymbol{\beta}^0 - 1)^T K^* (\boldsymbol{\alpha}^0 + \boldsymbol{\beta}^0 - 1)}}
$$

and transform (6.8) to the normalized quadratic dual (6.5), but with the additional upper bound on betas.

For clarity, we prefer to substitute variables: $(\forall i)\ \delta_i = -(\beta_i - 1)$. Then the form (6.8), and corresponding decision and correcting functions can be re-written as

$$\begin{aligned}(D_A+) \qquad \max_{\boldsymbol{\alpha},\boldsymbol{\delta}} D_A(\boldsymbol{\alpha},\boldsymbol{\delta}) &= \mathbf{e}^T\boldsymbol{\alpha} - A\sqrt{\boldsymbol{\alpha}^T Q\boldsymbol{\alpha} + \frac{1}{\gamma}(\boldsymbol{\alpha}-\boldsymbol{\delta})^T K^*(\boldsymbol{\alpha}-\boldsymbol{\delta})} \\ \mathbf{y}^T\boldsymbol{\alpha} &= 0,\ \mathbf{e}^T(\boldsymbol{\alpha}-\boldsymbol{\delta}) = 0 \\ 0 \le \alpha_i,\ 0 \le \delta_i &\le 1 \end{aligned} \tag{6.9}$$

$$f_A(\mathbf{x}) = \frac{A}{\sqrt{\boldsymbol{\alpha}^T Q\boldsymbol{\alpha} + \frac{1}{\gamma}(\boldsymbol{\alpha}-\boldsymbol{\delta})^T K^*(\boldsymbol{\alpha}-\boldsymbol{\delta})}} \sum_{i=1}^{\ell} \alpha_i y_i K(\mathbf{x},\mathbf{x}_i) + b \tag{6.10}$$

$$\varphi_A(\mathbf{x}^*) = \frac{1}{\gamma}\frac{A}{\sqrt{\boldsymbol{\alpha}^T Q\boldsymbol{\alpha} + \frac{1}{\gamma}(\boldsymbol{\alpha}-\boldsymbol{\delta})^T K^*(\boldsymbol{\alpha}-\boldsymbol{\delta})}} \sum_{i=1}^{\ell} (\alpha_i - \delta_i) K^*(\mathbf{x}^*,\mathbf{x}_i^*) + b^* \tag{6.11}$$

We emphasize that the main purpose of tinkering with the notation in this section, is to obtain forms that can be related effortlessly with fractional programming methodology. The conceptual goal is to show equivalence between the forms D_C+, D_A+ and D_v+ below. The technical detail required to accomplish it, consists mainly of variable substitution and taking care of variable bounds.

6.3 Fractional Programming-Based Derivation of ν-SVM+

In order to apply the FPG framework to D_A+, we need to modify the notation. Let

$$\begin{aligned} &\forall i \equiv \{1,\ldots,2\ell\},\ \forall j \equiv \{\ell+1,\ldots,2\ell\}, \\ &\tilde{\boldsymbol{\alpha}} = [\boldsymbol{\alpha};\boldsymbol{\delta}],\ \tilde{\mathbf{e}} = [\mathbf{e};\mathbf{0}],\ \tilde{\mathbf{y}} = [\mathbf{y};\mathbf{0}],\ \mathbf{u} = [\mathbf{e};-\mathbf{e}] \\ &Q_+ = \begin{bmatrix} Q + \frac{K^*}{\gamma} & -\frac{K^*}{\gamma} \\ -\frac{K^*}{\gamma} & \frac{K^*}{\gamma} \end{bmatrix} \end{aligned}$$

where ';' denotes concatenation of vectors, and Q_+ is a $2\ell \times 2\ell$ p.s.d (p.d) matrix since K is p.s.d (p.d). Then D_A+ can be written in the equivalent form:

$$\begin{aligned} \max_{\tilde{\boldsymbol{\alpha}}} D_A(\tilde{\boldsymbol{\alpha}}) &= \tilde{\mathbf{e}}^T\tilde{\boldsymbol{\alpha}} - A\sqrt{\tilde{\boldsymbol{\alpha}}^T Q_+ \tilde{\boldsymbol{\alpha}}} \\ \mathbf{u}^T\tilde{\boldsymbol{\alpha}} &= 0,\ \tilde{\mathbf{y}}^T\tilde{\boldsymbol{\alpha}} = 0 \\ \forall i,\ 0 \le \tilde{\alpha}_i;\ \forall j,\ \tilde{\alpha}_j &\le 1 \end{aligned} \tag{6.12}$$

and we can replace the denominator in (6.10),(6.11) by

$$\sqrt{\boldsymbol{\alpha}^T Q\boldsymbol{\alpha} + \frac{1}{\gamma}(\boldsymbol{\alpha}-\boldsymbol{\delta})^T K^*(\boldsymbol{\alpha}-\boldsymbol{\delta})} = \sqrt{\tilde{\boldsymbol{\alpha}}^T Q_+ \tilde{\boldsymbol{\alpha}}}$$

If we assume that the decision kernel matrix K is p.d, then Theorems 10, 11 can be adapted to hold for D_A+ (6.12).

Theorem 13 *Suppose K is p.d, let $\Phi_+(A)$ denote the optimal value of D_A+, and assume that theta is feasible (i.e., $\exists A_\theta : \Phi_+(A_\theta) = \theta$). Then,*
(a) *Solving $\Phi_+(A) - \theta = 0$, is equivalent to solving the C.F.P:*

$$(F_{A_\theta}+) \qquad \max \left\{ R^+_{A_\theta}(\mathbf{x}) = \frac{-\sqrt{\mathbf{x}^T Q_+ \mathbf{x}}}{\tilde{\mathbf{e}}^T \mathbf{x} - \theta} \,\middle|\, \mathbf{x} \in S \right\}$$
$$\text{where } S \equiv \{\mathbf{u}^T\mathbf{x} = 0,\ \tilde{\mathbf{y}}^T\mathbf{x} = 0,\ \tilde{\mathbf{e}}^T\mathbf{x} \geq \theta + \varepsilon,\ \forall i : 0 \leq x_i,\ \forall j : x_j \leq 1\} \quad (6.13)$$

(b) *By applying the variable transformation* (6.14) *to $F_{A_\theta}+$, we can obtain the equivalent convex problem $F''_{A_\theta}+$ with the additional variable t:*

$$\tilde{\boldsymbol{\alpha}} = \frac{1}{\tilde{\mathbf{e}}^T\mathbf{x} - \theta}\mathbf{x},\ t = \frac{1}{\tilde{\mathbf{e}}^T\mathbf{x} - \theta} \tag{6.14}$$

$$\begin{aligned} (F''_{A_\theta}+) \qquad & \max_{\tilde{\boldsymbol{\alpha}}, t} R''_{A_\theta}(\tilde{\boldsymbol{\alpha}}, t) = -\frac{1}{2}\sqrt{\tilde{\boldsymbol{\alpha}}^T Q_+ \tilde{\boldsymbol{\alpha}}} \\ & \tilde{\mathbf{e}}^T\tilde{\boldsymbol{\alpha}} = 2 + \theta t,\ \mathbf{u}^T\tilde{\boldsymbol{\alpha}} = 0,\ \tilde{\mathbf{y}}^T\tilde{\boldsymbol{\alpha}} = 0,\ t > 0 \\ & \forall i,\ 0 \leq \tilde{\alpha}_i;\ \forall j,\ \tilde{\alpha}_j \leq t \end{aligned} \tag{6.15}$$

Proof **(a)**,**(b)** Repeat the argument in Theorem 10. ■

Theorem 14 *Let $\{\tilde{\boldsymbol{\alpha}}_\theta, t_\theta\}$ denote the solution of the convex problem $F''_{A_\theta}+$, where theta is feasible. It follows that,*
(a) *$\forall \theta \geq 0$, $t_\theta \leq 1$.*
(b) *Solving $F''_{A_\theta}+$ is equivalent to solving the problem* (6.16) *with $\mu = \frac{2t_\theta}{2+\theta t_\theta} \leq 1$.*

Proof **(a)** See Theorem 11(a). **(b)** Multiply $\tilde{\boldsymbol{\alpha}}$ by $\frac{1}{2}(2 + \theta t)$ and substitute $(\forall i)\ \tilde{\alpha}_i = \frac{1}{2}(2 + \theta t)\tilde{\alpha}_i$, to obtain the problem

$$\begin{aligned} \max_{\tilde{\boldsymbol{\alpha}}}\ & -\frac{1}{2}\left(\frac{2+\theta t}{2}\right)\sqrt{\tilde{\boldsymbol{\alpha}}^T Q_+ \tilde{\boldsymbol{\alpha}}} \\ & \tilde{\mathbf{e}}^T\tilde{\boldsymbol{\alpha}} = 2,\ \mathbf{u}^T\tilde{\boldsymbol{\alpha}} = 0,\ \tilde{\mathbf{y}}^T\tilde{\boldsymbol{\alpha}} = 0,\ t > 0 \\ & \forall i,\ 0 \leq \tilde{\alpha}_i;\ \forall j,\ \tilde{\alpha}_j \leq \frac{2t}{2+\theta t} \end{aligned}$$

Let $\mu = \frac{2t}{2+\theta t}$. Repeat the argument in Theorem 11(b) to obtain the problem:

$$\begin{aligned} \max_{\tilde{\boldsymbol{\alpha}}}\ & -\frac{1}{2}\sqrt{\tilde{\boldsymbol{\alpha}}^T Q_+ \tilde{\boldsymbol{\alpha}}} \\ & \tilde{\mathbf{e}}^T\tilde{\boldsymbol{\alpha}} = 2,\ \mathbf{u}^T\tilde{\boldsymbol{\alpha}} = 0,\ \tilde{\mathbf{y}}^T\tilde{\boldsymbol{\alpha}} = 0 \\ & \forall i,\ 0 \leq \tilde{\alpha}_i;\ \forall j,\ \tilde{\alpha}_j \leq \mu \end{aligned} \tag{6.16}$$

■

From a computational point of view, it is preferable to solve the scaled version of (6.16), which we denote as the dual ν-SVM+ form,

$$\begin{aligned}\min_{\tilde{\boldsymbol{\alpha}}} \; & \frac{1}{2}\,\tilde{\boldsymbol{\alpha}}^T Q_+ \tilde{\boldsymbol{\alpha}} \\ & \tilde{\mathbf{e}}^T\tilde{\boldsymbol{\alpha}} = \nu_+\ell,\; \mathbf{u}^T\tilde{\boldsymbol{\alpha}} = 0,\; \tilde{\mathbf{y}}^T\tilde{\boldsymbol{\alpha}} = 0 \\ & \forall i,\; 0 \le \tilde{\alpha}_i;\; \forall j,\; \tilde{\alpha}_j \le 1\end{aligned}$$

or reverting to the usual notation,

$$\begin{aligned}(D_\nu+) \qquad \min_{\boldsymbol{\alpha},\boldsymbol{\delta}} \; & \frac{1}{2}\cdot\left[\boldsymbol{\alpha}^T Q\boldsymbol{\alpha} + \frac{1}{\gamma}(\boldsymbol{\alpha}-\boldsymbol{\delta})^T K^*(\boldsymbol{\alpha}-\boldsymbol{\delta})\right] \\ & \mathbf{e}^T\boldsymbol{\alpha} = \nu_+\ell,\; \mathbf{e}^T(\boldsymbol{\alpha}-\boldsymbol{\delta}) = 0,\; \mathbf{y}^T\boldsymbol{\alpha} = 0 \\ & 0 \le \alpha_i,\; 0 \le \delta_i \le 1 \end{aligned} \tag{6.17}$$

Notice that since there is no upper bound on alphas in (6.17) (that is, the bound is loose), the parameter ν_+ does not have exactly the same interpretation as ν in ν-SVM. In particular, the range of ν is upper bounded by $\nu_{\max}$ (see Section 4.3), while $\nu_+ \le 1$. The bound on ν_+ holds due to the constraints $\mathbf{e}^T\boldsymbol{\alpha} = \nu_+\ell$, $\mathbf{e}^T(\boldsymbol{\alpha}-\boldsymbol{\delta}) = 0$, and the upper bound on deltas.

6.4 Mixture Model of Slacks Extension

Schölkopf et al. (2000, proposition 5) showed that the parameter ν lets one control the number of support vectors and errors. An analogous result does not hold for ν_+, though in practice it is a good approximation since alphas do not exceed one by much (in order to minimize the objective value). However, if we consider the mixture model of slacks extension to SVM+ (Vapnik and Vashist, 2009, section 4.1), then (6.17) is obtained with an upper bound on alphas, $\alpha_i \le \theta$. The additional parameter θ ($\theta > 1$) is introduced to reinforce the smooth part of the function which models the slacks. If theta is not much larger than one, then ν_+ controls the number of support vectors in a similar way to ν.

6.5 Lower Bound on the Parameter ν_+

In Section 4.3, we observed that the parameter ν has a concrete lower bound, $0 < \nu_{min} \le \nu$. Intuitively, we would expect to obtain a lower bound on ν_+ as well, though it might depend on the particular γ parameter. However, the following theorem shows that the bound is independent of γ, and in fact exactly ν_{min}.

Theorem 15 *Suppose that all kernel parameters (if any exist) are fixed to some values, for both kernels $\{K, K^*\}$. In addition, assume that the kernel matrix K is p.d. Then for any $\gamma > 0$, ν_+ is bounded from below by ν_{min}, where ν_{min} is the lower bound on the parameter ν for the corresponding ν-SVM problem* (4.10)*, with the identical matrix Q. That is, $\forall\gamma > 0$, $\nu_{min} \le \nu_+ \le 1$.*

Proof Set $\nu_0 = \frac{2}{\ell}$, and solve the problem D_{ν_0} (4.10). Let $\boldsymbol{\alpha}^0$ denote the solution vector of D_{ν_0}, and let $\alpha_{max} = \max_i \alpha_i^0$. It follows that, $\nu_{min} = \frac{2}{\ell}\cdot\frac{1}{\alpha_{max}}$, and $\boldsymbol{\alpha}^{min} = \boldsymbol{\alpha}^0\cdot\frac{1}{\alpha_{max}}$ is the solution vector of $D_{\nu_{min}}$. In other words, ν_{min} is the smallest value of ν for which the

solution has at least one alpha at the upper bound. For any $0 < \nu < \nu_{min}$, the solution is just $\boldsymbol{\alpha}^{min}$ scaled by some constant (< 1).

Now fix gamma to any positive value, $\gamma = \gamma_0 > 0$, and solve $D_\nu+$ (6.17) with $\nu_+ = \nu_{min}$. It is clear that the second term in the objective $\{\frac{1}{\gamma_0}(\boldsymbol{\alpha} - \boldsymbol{\delta})^T K^*(\boldsymbol{\alpha} - \boldsymbol{\delta})\}$ is minimized when $\boldsymbol{\alpha} = \boldsymbol{\delta}$. Moreover, we already know that the first term $\{\boldsymbol{\alpha}^T Q \boldsymbol{\alpha}\}$ is minimized when $\boldsymbol{\alpha} = \boldsymbol{\alpha}^{min}$. Since $(\forall i)\, \alpha_i^{min} \leq 1$, we can choose $\boldsymbol{\delta} = \boldsymbol{\alpha}^{min}$ without violating the constraints $\delta_i \leq 1$. Similar to the argument above, ν_{min} is the smallest value of ν_+ for which the solution has at least one delta at the upper bound. For any $0 < \nu_+ < \nu_{min}$, the solution is just $\{\boldsymbol{\alpha}^{min}, \boldsymbol{\delta} = \boldsymbol{\alpha}^{min}\}$ scaled by some constant (< 1). Since we made no assumptions about the gamma parameter, the bound holds $\forall \gamma > 0$. ■

From a practical point of view, the theorem implies that the search interval over ν_+ can uniformly begin at ν_{min}, once all other parameters (kernels and γ) are fixed. Furthermore, the solution of $D_\nu+$ at ν_{min} can be computed once, and holds $\forall \gamma$. This is no longer the case in the interval $(\nu_{min}, 1]$, where the solution depends on γ.

When applying Theorem 15 in practice, there is one caveat to be aware of: it can be difficult to obtain an accurate lower bound due to the numerical issues that arise when we solve the problem D_ν with the parameter $\nu = \frac{2}{\ell}$. We could improve the accuracy—at the cost of significantly increasing computation time—by taking the square-root of the objective and solving the resulting SOCP form. However, it is preferable to implement a decomposition method (Chang and Lin, 2001, section 4), and set a lower than usual stopping tolerance.

7. A Decomposition Method for ν-SVM+

It is clear that ν-SVM+ offers a trade-off between intuitive search and simplicity of the optimization problem. In order to illustrate that ν-SVM+ can be used in practice, we extend the *alternating SMO* (aSMO) decomposition method, proposed by Pechyony et al. (2010)[5] for SVM+, to ν-SVM+. Further discussion on the computational merits of each method can be found in Appendix B.

7.1 Modified aSMO

Modifying aSMO to work with the $D_\nu+$ form is not difficult, but some care is needed to compute the thresholds:

(a) The sets of maximally sparse feasible directions I_1, I_2 are identical. The set I_3 is not considered since the constraint $\mathbf{e}^T \boldsymbol{\alpha} = \nu_+ \ell$ must hold throughout.

A move in the direction $u_{i,j} \in I_1$ increases δ_i and decreases δ_j by the same quantity, while the rest of the variables remain fixed. Likewise, a move in the direction $u_{i,j} \in I_2$ increases α_i and decreases α_j. Excluding I_3 implies that we cannot modify both alphas and deltas in the same iteration. For a formal definition of the sets, see (Pechyony et al., 2010, section 4).

5. The longer version can be found at www.nec-labs.com/∼pechyony/svmplus_smo.pdf

(b) The gradient (assuming maximization) is slightly different,

$$\forall i,\ g_i^{\alpha} = -(Q\boldsymbol{\alpha})_i - \frac{1}{\gamma}(K^*(\boldsymbol{\alpha} - \boldsymbol{\delta}))_i \tag{7.1}$$

$$\forall i,\ g_i^{\delta} = \frac{1}{\gamma}(K^*(\boldsymbol{\alpha} - \boldsymbol{\delta}))_i \tag{7.2}$$

But the update equations are similar, where we need to replace betas with deltas throughout (note that the sign should be reversed).

(c) The step size in the direction $u_{i,j} \in I_1$ must be clipped, due to the additional bound on deltas (in aSMO, betas are only bounded from below).

(d) The working set selection procedure is the same, but we do not consider set I_3.

(e) Convergence follows from the results of Bordes et al. (2005, appendix), where we can adapt the proof of Pechyony et al. (2010, section 5) to show that $\mathcal{U} = I_1 \cup I_2$ is a finite witness family.

7.2 Computation of the Thresholds

Suppose that $\{\boldsymbol{\alpha}, \boldsymbol{\beta}\}$ are the solution vectors of $D_\nu+$ (for some ν_+). By the KKT conditions,

$$\alpha_i > 0 \implies y_i f_A(\mathbf{x}_i) = 1 - \varphi_A(\mathbf{x}_i^*) \tag{7.3}$$

where $f_A(x), \varphi_A(x^*)$ are defined in (6.10) and (6.11) respectively—recall that solving $D_\nu+$ is equivalent to solving D_A+ for some appropriately chosen parameter A. To simplify the expression, define a constant C, and factor it out:

$$C = \frac{A}{\sqrt{\boldsymbol{\alpha}^T Q \boldsymbol{\alpha} + \frac{1}{\gamma}(\boldsymbol{\alpha} + \boldsymbol{\beta} - 1)^T K^*(\boldsymbol{\alpha} + \boldsymbol{\beta} - 1)}}$$

$$\begin{aligned} f_A(\mathbf{x}_i) &= C \cdot \sum_{j=1}^{\ell} \alpha_j y_j K(\mathbf{x}_i, \mathbf{x}_j) + b \\ &= C \cdot [\sum_{j=1}^{\ell} \alpha_j y_j K(\mathbf{x}_i, \mathbf{x}_j) + b] \end{aligned} \tag{7.4}$$

$$\begin{aligned} \varphi_A(\mathbf{x}_i^*) &= C \cdot \frac{1}{\gamma} \sum_{j=1}^{\ell} (\alpha_j - \delta_j) K^*(\mathbf{x}_i^*, \mathbf{x}_j^*) + b^* \\ &= C \cdot [\frac{1}{\gamma} \sum_{j=1}^{\ell} (\alpha_j - \delta_j) K^*(\mathbf{x}_i^*, \mathbf{x}_j^*) + b^*] \end{aligned} \tag{7.5}$$

Substitute (7.4) and (7.5) into (7.3) and obtain

$$\begin{aligned}
\alpha_i > 0 \Longrightarrow C \cdot [(Q\mathbf{a})_i + y_i b] &= 1 - C \cdot [\frac{1}{\gamma}(K^*(\boldsymbol{\alpha} - \boldsymbol{\delta}))_i + b^*] \\
(Q\mathbf{a})_i + y_i b &= \frac{1}{C} - \frac{1}{\gamma}(K^*(\boldsymbol{\alpha} - \boldsymbol{\delta}))_i - b^* \\
b^* + y_i b &= \frac{1}{C} - (Q\boldsymbol{\alpha})_i - \frac{1}{\gamma}(K^*(\boldsymbol{\alpha} - \boldsymbol{\delta}))_i \\
b^* + y_i b &= \frac{1}{C} + g_i^\alpha \\
\alpha_i > 0, y_i = 1 \Longrightarrow b^* + b &= \frac{1}{C} + g_i^\alpha \qquad (7.6) \\
\alpha_i > 0, y_i = -1 \Longrightarrow b^* - b &= \frac{1}{C} + g_i^\alpha \qquad (7.7)
\end{aligned}$$

Although C is an unknown constant, it is not required to compute the threshold. Subtracting (7.7) from (7.6), and averaging the gradient over the examples to avoid numerical errors, we have

$$\begin{aligned}
n_+ = \sum_{0<\alpha_i, y_i=1} 1,\ n_- &= \sum_{0<\alpha_i, y_i=-1} 1 \\
b = \frac{1}{2}(\frac{\sum_{0<\alpha_i, y_i=1} g_i^\alpha}{n_+} &- \frac{\sum_{0<\alpha_i, y_i=-1} g_i^\alpha}{n_-})
\end{aligned}$$

Since alphas are not upper-bounded, there must be at least one unbounded support vector satisfying each condition. If we also wish to compute b^*, then we could apply the KKT conditions for the correcting space:

$$\begin{aligned}
\delta_i < 1 \Longrightarrow \varphi(\mathbf{x}_i^*) = 0 &\Longrightarrow -g_i^\delta = b^* \\
b^* = -\frac{(\sum_{\delta_i<1} g_i^\delta)}{\sum_{\delta_i<1} 1}&
\end{aligned}$$

Deltas are bounded; therefore, if $(\forall i)\ \delta_i = 1$, then we set $b^* = \min_i g_i^\delta$.

7.3 Initialization Step

The optimization problem $D_\nu+$ (6.17) has one additional constraint ($\mathbf{e}^T\boldsymbol{\alpha} = \nu_+\ell$) compared to the original form D_C+ (6.5). As a result, the initial point must satisfy this constraint to be feasible. This is an inherent disadvantage of using ν-SVM+ (as well as ν-SVM), since a non-zero initial point implies a non-zero gradient that must be computed.

We can simplify the gradient computation by using the following procedure to specify the initial point: let $t = \frac{\nu_+\ell}{2}$, and select the first $\lfloor t \rfloor$ indices for which $y_i = 1$. Set the corresponding variables $\{\alpha_i, \delta_i\}$ to $(1, \ldots, 1, t - \lfloor t \rfloor)$, where $\forall i,\ \alpha_i = \delta_i$. Repeat the above with indices for which $y_i = -1$, and set the remaining variables to zero. It follows that,

$$\begin{aligned}
\boldsymbol{\alpha} = \boldsymbol{\delta} &\Longrightarrow \forall i,\ \frac{1}{\gamma}(K^*(\boldsymbol{\alpha} - \boldsymbol{\delta}))_i = 0 \\
&\Longrightarrow \forall i,\ g_i^\delta = 0,\ g_i^\alpha = -(Q\boldsymbol{\alpha})_i
\end{aligned}$$

Thus, in order to compute the initial gradient (7.1),(7.2), we only need to retrieve columns of the matrix Q. The cost of the initialization step is then similar to ν-SVM (see Chang and Lin, 2001, section 5). Furthermore, we could incorporate a "hot start" (that is, adjust the previous solution vector and set it as the current initial point) to reduce this cost when the search is over a range of ν_+ parameter values.

8. Conclusion

In this paper, we developed a theoretical framework for relating various formulations of regularization problems. As our main result, we presented a general template for transforming a given problem $\mathcal{P}$ into an equivalent form $\mathcal{P}^*$ in essentially two well-justified steps. In addition, we derived an upper bound on the search range of the parameter q as a special case.

To demonstrate that the proposed framework is practically useful, we considered dual SVM-type forms for classification. For the well-researched SVM algorithm, we were able to reproduce established theory for relating various forms in an elegant and concise manner. We then applied the framework to the recently introduced SVM+ method and obtained new forms by nearly identical technique. Since SVM+ has not been used extensively yet, we also included a decomposition method for the new ν-SVM+ form we derived.

Throughout the discussion, we made several simplifying assumptions to highlight the intuitive connection to fractional programming theory. However, we also showed in the Appendix that the framework can be extended further based on similar ideas. We briefly mention two more relevant aspects which we did not elaborate on.

When we discussed dual SVM and SVM+ forms, we explicitly assumed that the kernel matrix is not positive-semi-definite. Strict positive definiteness is not a very stringent requirement in general, and it is not difficult to relax it within the framework. From a theoretical perspective, the main difference would be the feasible range of theta and the behavior of the curve $\Phi(q)$, but the proof method remains similar.

Furthermore, the framework does not have to be limited to SVM-type forms. There are many settings in which forms have been manipulated and examined in relation to each other, for example, problems involving L1 and L2-norm-based regularization. In light of this, it is interesting to evaluate whether the framework is effective for a particular problem. In other words, does the equivalent form $\mathcal{P}^*$ have a parameter q^* with an intuitive meaning?

The answer for SVM and SVM+ forms is clearly affirmative. However, in general, the 'quality' of the parameter q^* (is it intuitive, simple to tune?) depends on the properties of the functions $\{N, D\}$ and the availability of efficient methods for solving $\mathcal{P}^*$. In the simplest case, we can completely eliminate the variable t due to scale invariance and polyhedral constraints. The transformation then amounts to dropping one of the functions from the objective of $\mathcal{P}$ into an extra constraint:

$$(\mathcal{P}^*) \qquad \begin{aligned} &\max \{-D(x)\} \\ &\text{subject to: } N(x) = q^*,\ x \in S \end{aligned}$$

More frequently, $\mathcal{P}^*$ is as complex as $\mathcal{F}'_\theta$. In this case, the answer may be negative unless we can simplify or scale the form, and the equivalence to $\mathcal{P}$ is not obvious without the framework.

Perhaps this is one explanation why we were not able to find previous work on the relation between SVM-type regularization forms through fractional programming. On the one hand, the connection is only mentioned when fractional programs are formulated from the outset (Mangasarian, 2005), but on the other hand, the parameter search task is not prioritized if there are few parameters and heuristic approaches are sufficient for comprehensive grid search. Nevertheless, we believe that the framework is intuitive and useful both to complement heuristic search and to simplify it when the optimization problem has many parameters.

Acknowledgments

This work was partially funded by NSF grant IIS-0916200. The author would like to thank Haimonti Dutta and Ansaf Salleb-Aouissi for thoroughly reviewing the paper. Section 6.2 is credited to Vladimir Vapnik and is based on his work.

Appendix A. Approach for Problems Lacking Scale Invariance

We discuss an approach for extending the FPG framework to functions whose degree of invariance is not one. If the given optimization problem $\mathcal{P}$ has an objective function with degree of invariance $d > 1$, then the condition $f(\Delta x) = \Delta^d f(x)$ holds. Therefore, it is logical to consider a slightly different variable transformation than (2.2):

$$u = t^{d-1} \cdot \frac{1}{D(x)} x,\ t^d = \frac{1}{D(x)},\ x \in S \tag{A.1}$$

The rest of the derivation of the equivalent form $\mathcal{P}^*$ is analagous to the one in Section 3. Likewise, if we flip the ratio, the corresponding transformation is

$$u = t^{d-1} \frac{1}{N(x) - \theta} x,\ t^d = \frac{1}{N(x) - \theta},\ x \in S \tag{A.2}$$

But the proof idea is the same.

In fact, Mond and Craven (1975) studied nonlinear FPs and converted them to convex problems with essentially the same method. They used different notation to denote multiplier functions for the objective ($\phi_0(t)$) and each constraint ($\phi_i(t)$). The multiplier functions are chosen to match the degrees of the functions that make up the optimization problem. We follow their approach to show that it is possible to apply the framework to non-scale invariant functions as well. We derive an analogous result to Theorem 11 for the dual SVM form D_C (4.4).

Theorem 16 *Suppose that the kernel matrix K is p.d, and add the constraint $\mathbf{e}^T\boldsymbol{\alpha} \geq \theta + \varepsilon$ to the problem D_C. Let $\Phi(C)$ denote the optimal value of D_C, and assume that theta is feasible ($\exists C_\theta : \Phi(C_\theta) = \theta$). Then,*
***(a)** Solving $\Phi(C) - \theta = 0$, is equivalent to solving the C.F.P:*

$$(F_{C_\theta}) \qquad \max \left\{ R_{C_\theta}(\mathbf{x}) = \frac{-\mathbf{x}^T Q \mathbf{x}}{\mathbf{e}^T\mathbf{x} - \theta} \,\middle|\, \mathbf{x} \in S \right\}$$
$$\text{where } S \equiv \{\mathbf{y}^T\mathbf{x} = 0,\ \mathbf{e}^T\mathbf{x} \geq \theta + \varepsilon,\ 0 \leq x_i \leq 1\} \tag{A.3}$$

(b) *By applying the multiplier functions* (A.4) *to* F_{C_θ}*, we can obtain the equivalent convex problem* F''_{C_θ} *with the additional variable* t*:*

$$\phi_0(t) = \frac{1}{2}t^2,\ (\forall i)\ \phi_i(t) = t \tag{A.4}$$

$$\begin{aligned}(F''_{C_\theta}) \qquad \max_{\boldsymbol{\alpha},t}\ & R''_{C_\theta}(\boldsymbol{\alpha},t) = -\frac{1}{2}\boldsymbol{\alpha}^T Q\boldsymbol{\alpha} \\ & \mathbf{e}^T\boldsymbol{\alpha} = \frac{2}{t} + \theta t,\ \mathbf{y}^T\boldsymbol{\alpha} = 0,\ t > 0 \\ & 0 \le \alpha_i \le t \end{aligned} \tag{A.5}$$

Proof **(a)** Follows from Theorem 9(a),(b), but without assuming scale invariance. **(b)** It follows from (Mond and Craven, 1975) that (A.4) applied to F_{C_θ} yields the functions and constraints

$$\begin{aligned} & N(\frac{\boldsymbol{\alpha}}{t}) \cdot \phi_0(t) = -\frac{1}{2}\boldsymbol{\alpha}^T Q\boldsymbol{\alpha} \\ & D(\frac{\boldsymbol{\alpha}}{t}) \cdot \phi_0(t) = \frac{1}{2}t^2 \cdot (\frac{\mathbf{e}^T\boldsymbol{\alpha}}{t} - \theta) = t \cdot \frac{\mathbf{e}^T\boldsymbol{\alpha} - \theta t}{2} = 1 \\ & t > 0,\ \phi_i(t) \cdot \frac{\mathbf{y}^T\boldsymbol{\alpha}}{t} = 0,\ \phi_i(t) \cdot (\frac{\mathbf{e}^T\boldsymbol{\alpha}}{t} - \theta - \varepsilon) \ge 0 \end{aligned} \tag{A.6}$$

$$0 \le \phi_i(t) \cdot \frac{\alpha_i}{t},\ \phi_i(t) \cdot (\frac{\alpha_i}{t} - 1) \le 0 \tag{A.7}$$

Simplifying (A.6),(A.7) and dropping the constraint $\mathbf{e}^T\boldsymbol{\alpha} \ge (\theta + \varepsilon)t$, leads to the convex problem F''_{C_θ}. ■

The deficiency of the approach for non-scale invariant functions is evident if we compare the convex problems F''_{C_θ} and F''_{A_θ} (5.3). The former has a more complicated form, since the extra constraint is quadratic (not linear). However, having obtained the solution vector $\{\boldsymbol{\alpha}_\theta, t_\theta\}$ of F''_{C_θ}, the approach in Theorem 11 (scale alphas and reparametrize the problem) is still applicable.

Theorem 17 *Let* $\{\boldsymbol{\alpha}_\theta, t_\theta\}$ *denote the solution of the convex problem* F''_{C_θ}*, where theta is feasible. It follows that,*
(a) $\forall\theta \ge 0,\ t_\theta \le 1$.
(b) *Solving* F''_{C_θ} *is equivalent to solving the form* D_μ (4.11) *with* $\mu = \frac{2t_\theta^2}{2+\theta t_\theta^2} \le 1$.

Proof **(a)** See Theorem 11(a). **(b)** Multiply $\boldsymbol{\alpha}$ by $\frac{1}{2}(\frac{2}{t} + \theta t)$ and substitute $(\forall i)\ \alpha_i = \frac{1}{2}(\frac{2}{t} + \theta t)\alpha_i$ to obtain the problem

$$\begin{aligned} \max_{\boldsymbol{\alpha}}\ & -\frac{1}{2}(\frac{2+\theta t^2}{2t})^2\boldsymbol{\alpha}^T Q\boldsymbol{\alpha} \\ & \mathbf{e}^T\boldsymbol{\alpha} = 2,\ \mathbf{y}^T\boldsymbol{\alpha} = 0,\ t > 0 \\ & 0 \le \alpha_i \le \frac{2t^2}{2+\theta t^2} \end{aligned}$$

Let $\mu = \frac{2t^2}{2+\theta t^2}$. Since the inequality $2 + \theta t^2 \geq 2t^2$ holds for all $\{\theta, t\}$ such that $\theta \geq 0$, $t \leq 1$, and from part (a), $t_\theta \leq 1$, we know that $\mu \leq 1$ holds. The rest of the argument is similar to Theorem 11(b). ■

We could derive the upper bound on the parameter C from the solution of $F''_{C_{\theta=0}}$. However, we omit this result since it is trivial to adapt Corollary 12.

Appendix B. Remarks on Decomposition Methods for SVM+

It is interesting to compare C-SVM+ and ν-SVM+ from a computational perspective. However, since it is difficult to quantitatively evaluate the trade-off between intuitive search and simplicity of the optimization problem, we focused on comparing the convergence rate (number of iterations) of the respective decomposition methods, rather than the actual running time.

We have implemented the aSMO decomposition method according to the specifications of Pechyony et al. (2010), and the extension described in Section 7, for C-SVM+ and ν-SVM+ respectively. Both methods were implemented without shrinking, and we solved the optimization problem D_C+ (6.5) with the additional upper bound on betas ($\beta_i \leq 1$).

For each comparison run, we solved $D_\nu+$ (6.17) with the stopping tolerance set to $\tau = 10^{-6}$ or $\tau = 10^{-7}$, and recorded the number of iterations it took aSMO to converge. We then computed the corresponding C parameter (see Section 6.2) from equations (7.6),(7.7), and solved D_C+ (where $\beta_i \leq 1$), with tolerance set to $C \times \tau$, to ensure a fair comparison. In other words, if the standard approach when solving C-SVM (C-SVM+) is to set the tolerance to 10^{-3} (Chang and Lin, 2011); then by analogy, the tolerance for ν-SVM+ should be set to some value in the range $[10^{-7}, 10^{-5}]$, where the smaller the parameter (ν_+), the smaller the tolerance. To verify that the relative stopping tolerance was appropriate, we checked the accuracy of the term $\{\boldsymbol{\alpha}^T Q \boldsymbol{\alpha} + \frac{1}{\gamma}(\boldsymbol{\alpha} - \boldsymbol{\delta})^T K^* (\boldsymbol{\alpha} - \boldsymbol{\delta})\}$, at the respective solution vectors. Excepting a handful of runs, the term was consistently accurate for at least five decimal digits.

For almost all the experiments, C-SVM+ converged faster than ν-SVM+. This is not surprising, since the main deficiency of the aSMO method for ν-SVM+ is the lack of a working set which includes both alphas and deltas (betas); recall from Section 7, that the set I_3 is not considered, as the constraint $\mathbf{e}^T \boldsymbol{\alpha} = \nu_+ \ell$ must hold throughout. As a result, ν-SVM+ cannot take a "combined" step, and thus requires approximately 1.5-2.0 times as many steps (iterations) as C-SVM+. For this reason, aSMO might not be the most effective method for the optimization problem $D_\nu+$. It is therefore interesting to consider a variant of aSMO, by adjusting the method to work with the inequality $\mathbf{e}^T \boldsymbol{\alpha} \geq \nu_+ \ell$, instead of the equality. Another possible approach is to extend the gSMO decomposition method (Pechyony et al., 2010).

In general, when numerical optimization methods are used, the QP forms (C-SVM, C-SVM+) are more efficient than equivalent forms (ν-SVM, ν-SVM+) which have an extra constraint. SOCP forms (A-SVM, A-SVM+) are also less efficient due to the second order cone constraint (Lobo et al., 1998). However, decomposition methods diminish this computational advantage (Vovsha, 2011). As a result, it is debatable whether QP forms should be preferred by default (Hsu et al., 2003), especially when the parameter search is sparse (few parameter values).

Appendix C. Derivation of the Form D_A+

We give the full derivation of the form D_A+ (6.8) from the primal $P_\Delta+$ (6.6). Construct lagrangian for $P_\Delta+$:

$$\begin{aligned}
L &= L(\mathbf{w}, b, \mathbf{w}^*, b^*, \boldsymbol{\xi}^*, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\mu}, \lambda) \\
&= \sum_{i=1}^{\ell}[(\mathbf{w}^* \cdot \mathbf{x}_i^*) + b^* + \xi_i^*] - \sum_{i=1}^{\ell} \alpha_i [y_i((\mathbf{w} \cdot \mathbf{x}_i) + b) - 1 + ((\mathbf{w}^* \cdot \mathbf{x}_i^*) + b^*)] \\
&\quad - \frac{1}{2}\lambda(\frac{1}{\Delta^2} - (\mathbf{w} \cdot \mathbf{w}) - \gamma(\mathbf{w}^* \cdot \mathbf{w}^*)) - \sum_{i=1}^{\ell} \beta_i[(\mathbf{w}^* \cdot \mathbf{x}_i^*) + b^* + \xi_i^*] - \sum_{i=1}^{\ell} \mu_i \xi_i^* \\
&= \sum_{i=1}^{\ell} \alpha_i + \frac{1}{2}\lambda(\mathbf{w} \cdot \mathbf{w}) + \frac{1}{2}\lambda\gamma(\mathbf{w}^* \cdot \mathbf{w}^*) - \frac{1}{2}\lambda\frac{1}{\Delta^2} - \sum_{i=1}^{\ell} \alpha_i y_i[(\mathbf{w} \cdot x_i) + b] \\
&\quad - \sum_{i=1}^{\ell}(\alpha_i + \beta_i - 1)[(\mathbf{w}^* \cdot \mathbf{x}_i^*) + b^*] - \sum_{i=1}^{\ell}(\beta_i + \mu_i - 1)\xi_i^*
\end{aligned}$$

Minimize L with respect to $\mathbf{w}, \mathbf{w}^*, b, b^*, \xi_i^*$ and maximize it with respect to $\{\alpha, \beta, \lambda, \mu\} \geq 0$:

$$\begin{aligned}
&\frac{\partial L}{\partial \mathbf{w}} = \lambda \mathbf{w} - \sum_{i=1}^{\ell} \alpha_i y_i \mathbf{x}_i = 0 \\
&\frac{\partial L}{\partial \mathbf{w}^*} = \lambda\gamma \mathbf{w}^* - \sum_{i=1}^{\ell}(\alpha_i + \beta_i - 1)\mathbf{x}_i^* = 0 \\
&\frac{\partial L}{\partial b} = -\sum_{i=1}^{\ell} \alpha_i y_i = 0 \qquad\qquad \frac{\partial L}{\partial b^*} = -\sum_{i=1}^{\ell}(\alpha_i + \beta_i - 1) = 0 \\
&\forall_i,\ \frac{\partial L}{\partial \xi_i^*} = -(\beta_i + \mu_i - 1) = 0 \\
&0 \leq \{\alpha, \beta, \lambda, \mu\}
\end{aligned}$$

This yields the following conditions:

$$\begin{aligned}
&\mathbf{w} = \frac{1}{\lambda}\sum_{i=1}^{\ell} \alpha_i y_i \mathbf{x}_i \\
&\mathbf{w}^* = \frac{1}{\lambda\gamma}\sum_{i=1}^{\ell}(\alpha_i + \beta_i - 1)\mathbf{x}_i^* \\
&\sum_{i=1}^{\ell} \alpha_i y_i = 0 \\
&\sum_{i=1}^{\ell}(\alpha_i + \beta_i - 1) = 0 \\
&\forall_i,\ 0 \leq \beta_i \leq 1,\ 0 \leq \alpha_i,\ 0 \leq \lambda
\end{aligned}$$

From which we obtain by substitution (and setting $A = \frac{1}{\Delta}$):

$$\begin{aligned}
\max_{\alpha,\beta,\lambda} D(\alpha,\beta,\lambda) &= \sum_{i=1}^{\ell}\alpha_i - \frac{1}{2\lambda}\sum_{i,j=1}^{\ell}\alpha_i\alpha_j y_i y_j(\mathbf{x}_i\cdot\mathbf{x}_j) \\
&\quad - \frac{1}{2\lambda\gamma}\sum_{i,j=1}^{\ell}(\alpha_i+\beta_i-1)(\alpha_j+\beta_j-1)(\mathbf{x}_i^*\cdot\mathbf{x}_j^*) - \frac{1}{2}\lambda A^2 \\
&= \sum_{i=1}^{\ell}\alpha_i - \frac{1}{2\lambda}\sum_{i,j=1}^{\ell}[\alpha_i\alpha_j y_i y_j(\mathbf{x}_i\cdot\mathbf{x}_j) + \frac{1}{\gamma}(\alpha_i+\beta_i-1)(\alpha_j+\beta_j-1)(\mathbf{x}_i^*\cdot\mathbf{x}_j^*)] - \frac{1}{2}\lambda A^2 \\
&\text{subject to :} \\
&\quad \sum_{i=1}^{\ell}\alpha_i y_i = 0,\ \sum_{i=1}^{\ell}(\alpha_i+\beta_i-1) = 0,\ \gamma \geq 0 \\
&\quad \forall_i,\ 0 \leq \alpha_i,\ 0 \leq \beta_i \leq 1
\end{aligned}$$

We can maximize with respect to λ:

$$\begin{aligned}
&\frac{\partial D(\alpha,\beta,\lambda)}{\partial\lambda} = \frac{1}{2\lambda^2}\sum_{i,j=1}^{\ell}[\alpha_i\alpha_j y_i y_j(\mathbf{x}_i\cdot\mathbf{x}_j) + \frac{1}{\gamma}(\alpha_i+\beta_i-1)(\alpha_j+\beta_j-1)(\mathbf{x}_i^*\cdot\mathbf{x}_j^*)] - \frac{1}{2}A^2 \\
&\frac{\partial D(\alpha,\beta,\lambda)}{\partial\lambda} = 0 \Longrightarrow \\
&\quad \frac{1}{\lambda^2} = \frac{A^2}{\sum_{i,j=1}^{\ell}[\alpha_i\alpha_j y_i y_j(\mathbf{x}_i\cdot\mathbf{x}_j) + \frac{1}{\gamma}(\alpha_i+\beta_i-1)(\alpha_j+\beta_j-1)(\mathbf{x}_i^*\cdot\mathbf{x}_j^*)]} \\
&\quad \frac{1}{\lambda} = \frac{A}{\sqrt{\sum_{i,j=1}^{\ell}[\alpha_i\alpha_j y_i y_j(\mathbf{x}_i\cdot\mathbf{x}_j) + \frac{1}{\gamma}(\alpha_i+\beta_i-1)(\alpha_j+\beta_j-1)(\mathbf{x}_i^*\cdot\mathbf{x}_j^*)]}}
\end{aligned}$$

If we substitute $\frac{1}{\lambda}$ back, we get:

$$\begin{aligned}
\max_{\alpha,\beta} D(\alpha,\beta) &= \sum_{i=1}^{\ell}\alpha_i - A\sqrt{\sum_{i,j=1}^{\ell}[\alpha_i\alpha_j y_i y_j(\mathbf{x}_i\cdot\mathbf{x}_j) + \frac{1}{\gamma}(\alpha_i+\beta_i-1)(\alpha_j+\beta_j-1)(\mathbf{x}_i^*\cdot\mathbf{x}_j^*)]} \\
&\text{subject to :} \\
&\sum_{i=1}^{\ell}\alpha_i y_i = 0,\ \sum_{i=1}^{\ell}(\alpha_i+\beta_i-1) = 0 \\
&\forall_i,\ 0 \leq \alpha_i,\ 0 \leq \beta_i \leq 1
\end{aligned}$$

If we generalize to the nonlinear case, where $Q_{i,j} = y_i y_j K(\mathbf{x}_i,\mathbf{x}_j)$, K is the kernel in decision space, and K^* is the kernel in correcting space, in vector notation, we obtain the problem (6.8).

References

M. Avriel, W. E. Diewert, S. Schaible, and I. Zang. *Generalized Concavity*. Plenum Press, New York, 1988.

K. P. Bennett and E. J. Bredensteiner. Duality and geometry in svm classifiers. In *In Proc. 17th International Conf. on Machine Learning*, pages 57–64. Morgan Kaufmann, 2000.

A. Bordes, S. Ertekin, J. Weston, and L. Bottou. Fast kernel classifiers with online and active learning. *Journal of Machine Learning Research*, 6:1579–1619, 2005.

B. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, COLT '92, pages 144–152, New York, NY, USA, 1992. ACM.

L. Bottou and C-J. Lin. Support vector machine solvers. In L. Bottou, O. Chapelle, D. DeCoste, and J. Weston, editors, *Large Scale Kernel Machines*, pages 301–320. MIT Press, Cambridge, MA., 2007.

C-C. Chang and C-J. Lin. Training v-support vector classifiers: Theory and algorithms. *Neural Computation*, 13(9):2119–2147, 2001.

C-C. Chang and C-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

A. Charnes and W. W. Cooper. Programming with linear fractional functionals. *Naval Research Logistics Quarterly*, 9(3-4):181–186, 1962.

C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

D. J. Crisp and C. J. C. Burges. A geometric interpretation of v-svm classifiers. In *NIPS*, pages 244–250. The MIT Press, 1999.

W. Dinkelbach. On nonlinear fractional programming. *Management Science*, 12(7):492–498, 1967.

L. Gunter and J. Zhu. Computing the solution path for the regularized support vector regression. In *NIPS*, 2005.

T. Hastie, S. Rosset, R. Tibshirani, and J. Zhu. The entire regularization path for the support vector machine. *Journal of Machine Learning Research*, 5:1391–1415, 2004.

C-W. Hsu, C-C. Chang, and C-J. Lin. A practical guide to support vector classification. Technical report, National Taiwan University, 2003.

T. Ibaraki. Solving mathematical programming problems with fractional objective functions. In S. Schaible and W. T. Ziemba, editors, *Generalized Concavity in Optimization and Economics*, pages 441–472. Academic Press, 1981.

R. Jagannathan. On some properties of programming problems in parametric form pertaining to fractional programming. *Management Science*, 12(7):609–615, 1966.

M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret. Applications of second-order cone programming. *Linear Algebra and its Applications*, 284(1-3):193–228, 1998.

G. Loosli, G. Gasso, and S. Canu. Regularization paths for v-svm and v-svr. In Derong Liu, Shumin Fei, Zengguang Hou, Huaguang Zhang, and Changyin Sun, editors, *Advances in Neural Networks ISNN 2007*, volume 4493 of *Lecture Notes in Computer Science*, pages 486–496. Springer Berlin Heidelberg, 2007.

O. L. Mangasarian. Support vector machine classification via parameterless robust linear programming. *Optimization Methods and Software*, 20(1):115–125, 2005.

B. Mond and B. D. Craven. Nonlinear fractional programming. *Bulletin of the Australian Mathematical Society*, 12(3):391–397, 1975.

D. Pechyony, R. Izmailov, A. Vashist, and V. Vapnik. Smo-style algorithms for learning using privileged information. In *DMIN*, pages 235–241. CSREA Press, 2010.

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

S. Rosset and J. Zhu. Piecewise linear regularized solution paths. *The Annals of Statistics*, pages 1012–1030, 2007.

S. Schaible. Parameter-free convex equivalent and dual programs of fractional programming problems. *Mathematical Methods of Operations Research*, 18(5):187–196, 1974.

S. Schaible. Fractional programming: Applications and algorithms. *European Journal of Operational Research*, 7(2):111–120, 1981.

S. Schaible. Fractional programming. i. duality. *Management Science*, 22(8):858–867, 1976.

S. Schaible and T. Ibaraki. Fractional programming. *European Journal of Operational Research*, 12(4):325–338, 1983.

B. Schölkopf and A. J. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, USA, 2002.

B. Schölkopf, A. J. Smola, R. C. Williamson, and P. L. Bartlett. New support vector algorithms. *Neural Computation*, 12(5):1207–1245, 2000.

J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, New York, NY, USA, 2004.

V. Vapnik. *Estimation of Dependences Based on Empirical Data*. Information Science and Statistics. Springer-Verlag New York, Inc., 2nd edition, 2006.

V. Vapnik. *Statistical Learning Theory*. John Wiley and Sons, 1998.

V. Vapnik and A. Lerner. Pattern recognition using generalized portrait method. *Automation and Remote Control*, 24:774–780, 1963.

V. Vapnik and A. Vashist. A new learning paradigm: Learning using privileged information. *Neural Networks*, 22(5-6):544–557, 2009.

V. Vapnik, A. Vashist, and N. Pavlovitch. Learning using hidden information (learning with teacher). In *IJCNN*, pages 3188–3195. IEEE, 2009.

I. Vovsha. Conic smo. Technical Report CCLS-11-03, Columbia University, 2011.