Chapter 16

Dual Ascent Methods; ADMM

This chapter is devoted to the presentation of one of the best methods known at the present for solving optimization problems involving equality constraints. In fact, this method can also handle more general constraints, namely, membership in a convex set. It can also be used to solve a range of problems arising in machine learning including *lasso minimization*, *elastic net regression*, *support vector machine (SVM)*, and *ν-SV regression*. In order to obtain a good understanding of this method, called the *alternating direction method of multipliers*, for short *ADMM*, we review two precursors of ADMM, the *dual ascent method* and the *method of multipliers*.

ADMM is not a new method. In fact, it was developed in the 1970's. It has been revived as a very effective method to solve problems in statistical and machine learning dealing with very large data because it is well suited to distributed (convex) optimization. An extensive presentation of ADMM, its variants, and its applications, is given in the excellent paper by Boyd, Parikh, Chu, Peleato and Eckstein [Boyd *et al.* (2010)]. This paper is essentially a book on the topic of ADMM, and our exposition is deeply inspired by it.

In this chapter, we consider the problem of minimizing a convex function J (not necessarily differentiable) under the equality constraints $Ax = b$. In Section 16.1 we discuss the dual ascent method. It is essentially gradient descent applied to the dual function G, but since G is maximized, gradient descent becomes gradient ascent.

In order to make the minimization step of the dual ascent method more robust, one can use the trick of adding the penalty term $(\rho/2) \left\| Au - b \right\|_2^2$ to the Lagrangian. We obtain the *augmented Lagrangian*

$$L_\rho(u, \lambda) = J(u) + \lambda^\top (Au - b) + (\rho/2) \left\| Au - b \right\|_2^2,$$

with $\lambda \in \mathbb{R}^m$, and where $\rho > 0$ is called the *penalty parameter*. We obtain the minimization Problem (P_ρ),

$$\begin{aligned} &\text{minimize} \quad J(u) + (\rho/2)\left\|Au - b\right\|_2^2 \\ &\text{subject to} \quad Au = b, \end{aligned}$$

which is equivalent to the original problem.

The benefit of adding the penalty term $(\rho/2)\left\|Au - b\right\|_2^2$ is that by Proposition 15.25, Problem (P_ρ) has a unique optimal solution under mild conditions on A. Dual ascent applied to the dual of (P_ρ) is called the *method of multipliers* and is discussed in Section 16.2.

The alternating direction method of multipliers, for short ADMM, combines the decomposability of dual ascent with the superior convergence properties of the method of multipliers. The idea is to split the function J into two independent parts, as $J(x, z) = f(x) + g(z)$, and to consider the Minimization Problem (P_{admm}),

$$\begin{aligned} &\text{minimize} \quad f(x) + g(z) \\ &\text{subject to} \quad Ax + Bz = c, \end{aligned}$$

for some $p \times n$ matrix A, some $p \times m$ matrix B, and with $x \in \mathbb{R}^n$, $z \in \mathbb{R}^m$, and $c \in \mathbb{R}^p$. We also assume that f and g are convex. Further conditions will be added later.

As in the method of multipliers, we form the augmented Lagrangian

$$L_\rho(x, z, \lambda) = f(x) + g(z) + \lambda^\top(Ax + Bz - c) + (\rho/2)\left\|Ax + Bz - c\right\|_2^2,$$

with $\lambda \in \mathbb{R}^p$ and for some $\rho > 0$. The major difference with the method of multipliers is that instead of performing a minimization step jointly over x and z, ADMM *first performs an x-minimization step and then a z-minimization step.* Thus x and z are updated in an alternating or sequential fashion, which accounts for the term *alternating direction*. Because the Lagrangian is augmented, some mild conditions on A and B imply that these minimization steps are guaranteed to terminate. ADMM is presented in Section 16.3.

In Section 16.4 we prove the convergence of ADMM under the following assumptions:

(1) The functions $f \colon \mathbb{R} \to \mathbb{R} \cup \{+\infty\}$ and $g \colon \mathbb{R} \to \mathbb{R} \cup \{+\infty\}$ are proper and closed convex functions (see Section 15.1) such that $\mathbf{relint}(\text{dom}(f)) \cap \mathbf{relint}(\text{dom}(g)) \neq \emptyset$.
(2) The $n \times n$ matrix $A^\top A$ is invertible and the $m \times m$ matrix $B^\top B$ is invertible. Equivalently, the $p \times n$ matrix A has rank n and the $p \times m$ matrix has rank m.

(3) The unaugmented Lagrangian $L_0(x,z,\lambda) = f(x) + g(z) + \lambda^\top(Ax + Bz - c)$ has a saddle point, which means there exists x^*, z^*, λ^* (not necessarily unique) such that

$$L_0(x^*, z^*, \lambda) \leq L_0(x^*, z^*, \lambda^*) \leq L_0(x, z, \lambda^*)$$

for all x, z, λ.

By Theorem 15.10, Assumption (3) is equivalent to the fact that the KKT equations are satisfied by some triple (x^*, z^*, λ^*), namely

$$Ax^* + Bz^* - c = 0 \tag{$*$}$$

and

$$0 \in \partial f(x^*) + \partial g(z^*) + A^\top \lambda^* + B^\top \lambda^*, \tag{$\dagger$}$$

Assumption (3) is also equivalent to Conditions (a) and (b) of Theorem 15.10. In particular, our program has an optimal solution (x^*, z^*). By Theorem 15.12, λ^* is maximizer of the dual function $G(\lambda) = \inf_{x,z} L_0(x, z, \lambda)$ and strong duality holds, that is, $G(\lambda^*) = f(x^*) + g(z^*)$ (the duality gap is zero).

We will show after the proof of Theorem 16.1 that Assumption (2) is actually implied by Assumption (3). This allows us to prove a convergence result stronger than the convergence result proven in Boyd et al. [Boyd *et al.* (2010)] (under the exact same assumptions (1) and (3)). In particular, we prove that *all* of the sequences (x^k), (z^k), and (λ^k) converge to optimal solutions $(\widetilde{x}, \widetilde{z})$, and $\widetilde{\lambda}$. The core of our proof is due to Boyd et al. [Boyd *et al.* (2010)], but there are new steps because we have the stronger hypothesis (2).

In Section 16.5, we discuss stopping criteria.

In Section 16.6 we present some applications of ADMM, in particular, minimization of a proper closed convex function f over a closed convex set C in $\mathbb{R}^n$ and quadratic programming. The second example provides one of the best methods for solving quadratic problems, including the SVM problems discussed in Chapter 18, the elastic net method in Section 19.6, and ν-SV regression in Chapter 20.

Section 16.8 gives applications of ADMM to ℓ^1-norm problems, in particular, lasso regularization, which plays an important role in machine learning.

16.1 Dual Ascent

Our goal is to solve the minimization problem, Problem (P),

$$\begin{aligned} &\text{minimize} && J(u)\\ &\text{subject to} && Au = b, \end{aligned}$$

with affine equality constraints (with A an $m \times n$ matrix and $b \in \mathbb{R}^m$). The Lagrangian $L(u, \lambda)$ of Problem (P) is given by

$$L(u, \lambda) = J(u) + \lambda^\top (Au - b).$$

with $\lambda \in \mathbb{R}^m$. From Proposition 14.13, the dual function $G(\lambda) = \inf_{u \in \mathbb{R}^n} L(u, \lambda)$ is given by

$$G(\lambda) = \begin{cases} -b^\top \lambda - J^*(-A^\top \lambda) & \text{if } -A^\top \lambda \in \mathrm{dom}(J^*),\\ -\infty & \text{otherwise,} \end{cases}$$

for all $\lambda \in \mathbb{R}^m$, where J^* is the conjugate of J. Recall that by Definition 14.11, the *conjugate* f^* of a function $f \colon U \to \mathbb{R}$ defined on a subset U of $\mathbb{R}^n$ is the partial function $f^* \colon \mathbb{R}^n \to \mathbb{R}$ defined by

$$f^*(y) = \sup_{x \in U} (y^\top x - f(x)), \quad y \in \mathbb{R}^n.$$

If the conditions of Theorem 14.7(1) hold, which in our case means that for every $\lambda \in \mathbb{R}^m$, there is a unique $u_\lambda \in \mathbb{R}^n$ such that

$$G(\lambda) = L(u_\lambda, \lambda) = \inf_{u \in \mathbb{R}^n} L(u, \lambda),$$

and that the function $\lambda \mapsto u_\lambda$ is continuous, then G is differentiable. Furthermore, we have

$$\nabla G_\lambda = Au_\lambda - b,$$

and for any solution $\mu = \lambda^*$ of the dual problem

$$\begin{aligned} &\text{maximize} && G(\lambda)\\ &\text{subject to} && \lambda \in \mathbb{R}^m, \end{aligned}$$

the vector $u^* = u_\mu$ is a solution of the primal Problem (P). Furthermore, $J(u^*) = G(\lambda^*)$, that is, the duality gap is zero.

The dual ascent method is essentially gradient descent applied to the dual function G. But since G is maximized, gradient descent becomes gradient ascent. Also, we no longer worry that the minimization problem $\inf_{u \in \mathbb{R}^n} L(u, \lambda)$ has a unique solution, so we denote by u^+ some minimizer of the above problem, namely

$$u^+ = \arg\min_u L(u, \lambda).$$

Given some initial dual variable λ^0, the *dual ascent method* consists of the following two steps:

$$\begin{aligned}u^{k+1} &= \arg\min_u L(u, \lambda^k)\\ \lambda^{k+1} &= \lambda^k + \alpha^k(Au^{k+1} - b),\end{aligned}$$

where $\alpha^k > 0$ is a step size. The first step is used to compute the "new gradient" (indeed, if the minimizer u^{k+1} is unique, then $\nabla G_{\lambda^k} = Au^{k+1} - b$), and the second step is a dual variable update.

Example 16.1. Let us look at a very simple example of the gradient ascent method applied to a problem we first encountered in Section 6.1, namely minimize $J(x, y) = (1/2)(x^2 + y^2)$ subject to $2x - y = 5$. The Lagrangian is

$$L(x, y, \lambda) = \frac{1}{2}(x^2 + y^2) + \lambda(2x - y - 5).$$

See Figure 16.1.

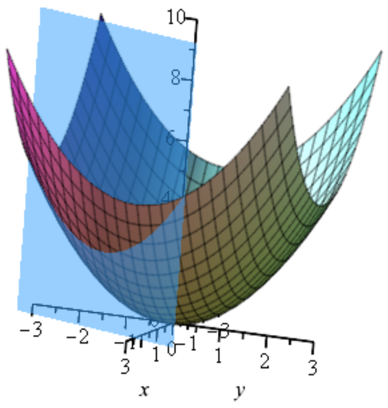


Fig. 16.1 The graph of $J(x, y) = (1/2)(x^2 + y^2)$ is the parabolic surface while the graph of $2x - y = 5$ is the transparent blue plane. The solution to Example 16.1 is apex of the intersection curve, namely the point $(2, -1, \frac{5}{2})$.

The method of Lagrangian duality says first calculate

$$G(\lambda) = \inf_{(x,y)\in\mathbb{R}^2} L(x, y, \lambda).$$

Since

$$J(x, y) = \frac{1}{2}\begin{pmatrix} x & y \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix},$$

we observe that $J(x,y)$ is a quadratic function determined by the positive definite matrix $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, and hence to calculate $G(\lambda)$, we must set $\nabla L_{x,y} = 0$. By calculating $\frac{\partial J}{\partial x} = 0$ and $\frac{\partial J}{\partial y} = 0$, we find that $x = -2\lambda$ and $y = \lambda$. Then $G(\lambda) = -5/2\lambda^2 - 5\lambda$, and we must calculate the maximum of $G(\lambda)$ with respect to $\lambda \in \mathbb{R}$. This means calculating $G'(\lambda) = 0$ and obtaining $\lambda = -1$ for the solution of $(x, y, \lambda) = (-2\lambda, \lambda, \lambda) = (2, -1, -1)$.

Instead of solving *directly* for λ in terms of (x, y), the method of dual ascent begins with a *numerical* estimate for λ, namely λ^0, and forms the "numerical" Lagrangian

$$L(x, y, \lambda^0) = \frac{1}{2}(x^2 + y^2) + \lambda^0(2x - y - 5).$$

With this numerical value λ^0, we minimize $L(x, y, \lambda^0)$ with respect to (x, y). This calculation will be identical to that used to form $G(\lambda)$ above, and as such, we obtain the iterative step $(x^1, y^1) = (-2\lambda^0, \lambda^0)$. So if we replace λ^0 by λ^k, we have the first step of the dual ascent method, namely

$$u^{k+1} = \begin{pmatrix} x^{k+1} \\ y^{k+1} \end{pmatrix} = \begin{pmatrix} -2 \\ 1 \end{pmatrix} \lambda^k.$$

The second step of the dual ascent method refines the numerical estimate of λ by calculating

$$\lambda^{k+1} = \lambda^k + \alpha^k \left(\begin{pmatrix} 2 & -1 \end{pmatrix} \begin{pmatrix} x^{k+1} \\ y^{k+1} \end{pmatrix} - 5 \right).$$

(Recall that in our original problem the constraint is $2x - y = 5$ or $\begin{pmatrix} 2 & -1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} - 5$, so $A = \begin{pmatrix} 2 & -1 \end{pmatrix}$ and $b = 5$.) By simplifying the above equation, we find that

$$\lambda^{k+1} = (1 - \beta)\lambda^k - \beta, \qquad \beta = 5\alpha^k.$$

Back substituting for λ^k in the preceding equation shows that

$$\lambda^{k+1} = (1 - \beta)^{k+1}\lambda^0 + (1 - \beta)^{k+1} - 1.$$

If $0 < \beta \leq 1$, the preceding line implies that λ^{k+1} converges to $\lambda = -1$, which coincides with the answer provided by the original Lagrangian duality method. Observe that if $\beta = 1$ or $\alpha^k = \frac{1}{5}$, the dual ascent method terminates in one step.

With an appropriate choice of α^k, we have $G(\lambda^{k+1}) > G(\lambda^k)$, so the method makes progress. Under certain assumptions, for example, that J is strictly convex and some conditions of the α^k, it can be shown that dual ascent converges to an optimal solution (both for the primal and the dual). However, the main flaw of dual ascent is that the minimization step may diverge. For example, this happens is J is a nonzero affine function of one of its components. The remedy is to add a penalty term to the Lagrangian.

On the positive side, the dual ascent method leads to a decentralized algorithm if the function J is separable. Suppose that u can be split as $u = \sum_{i=1}^N u_i$, with $u_i \in \mathbb{R}^{n_i}$ and $n = \sum_{i=1}^N n_i$, that

$$J(u) = \sum_{i=1}^N J_i(u_i),$$

and that A is split into N blocks A_i (with A_i a $m \times n_i$ matrix) as $A = [A_1 \cdots A_N]$, so that $Au = \sum_{k=1}^N A_i u_i$. Then the Lagrangian can be written as

$$L(u, \lambda) = \sum_{i=1}^N L_i(u_i, \lambda),$$

with

$$L_i(u_i, \lambda) = J_i(u_i) + \lambda^\top \left(A_i u_i - \frac{1}{N} b \right).$$

it follows that the minimization of $L(u, \lambda)$ with respect to the primal variable u can be split into N separate minimization problems that can be solved in parallel. The algorithm then performs the N updates

$$u_i^{k+1} = \arg\min_{u_i} L_i(u_i, \lambda^k)$$

in parallel, and then the step

$$\lambda^{k+1} = \lambda^k + \alpha^k (Au^{k+1} - b).$$

16.2 Augmented Lagrangians and the Method of Multipliers

In order to make the minimization step of the dual ascent method more robust, one can use the trick of adding the penalty term $(\rho/2) \left\| Au - b \right\|_2^2$ to the Lagrangian.

Definition 16.1. Given the Optimization Problem (P),

$$\begin{aligned} &\text{minimize} \quad J(u) \\ &\text{subject to} \quad Au = b, \end{aligned}$$

the *augmented Lagrangian* is given by

$$L_\rho(u, \lambda) = J(u) + \lambda^\top (Au - b) + (\rho/2) \|Au - b\|_2^2,$$

with $\lambda \in \mathbb{R}^m$, and where $\rho > 0$ is called the *penalty parameter*.

The augmented Lagrangian $L_\rho(u, \lambda)$ can be viewed as the ordinary Lagrangian of the Minimization Problem (P_ρ),

$$\begin{aligned} &\text{minimize} \quad J(u) + (\rho/2) \|Au - b\|_2^2 \\ &\text{subject to} \quad Au = b. \end{aligned}$$

The above problem is equivalent to Program (P), since for any feasible solution of (P_ρ), we must have $Au - b = 0$.

The benefit of adding the penalty term $(\rho/2) \|Au - b\|_2^2$ is that by Proposition 15.25, Problem (P_ρ) has a unique optimal solution under mild conditions on A.

Dual ascent applied to the dual of (P_ρ) yields the the *method of multipliers*, which consists of the following steps, given some initial λ^0:

$$\begin{aligned} u^{k+1} &= \arg\min_u L_\rho(u, \lambda^k) \\ \lambda^{k+1} &= \lambda^k + \rho(Au^{k+1} - b). \end{aligned}$$

Observe that the second step uses the parameter ρ. The reason is that it can be shown that choosing $\alpha^k = \rho$ guarantees that (u^{k+1}, λ^{k+1}) satisfies the equation

$$\nabla J_{u^{k+1}} + A^\top \lambda^{k+1} = 0,$$

which means that (u^{k+1}, λ^{k+1}) is dual feasible; see Boyd, Parikh, Chu, Peleato and Eckstein [Boyd *et al.* (2010)], Section 2.3.

Example 16.2. Consider the minimization problem

$$\begin{aligned} &\text{minimize} \quad y^2 + 2x \\ &\text{subject to} \quad 2x - y = 0. \end{aligned}$$

See Figure 16.2.

The quadratic function

$$J(x, y) = y^2 + 2x = \begin{pmatrix} x & y \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 2 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

is convex but not strictly convex. Since $y = 2x$, the problem is equivalent to minimizing $y^2 + 2x = 4x^2 + 2x$, whose minimum is achieved for $x = -1/4$ (since setting the derivative of the function $x \mapsto 4x^2 + 2$ yields $8x + 2 = 0$).

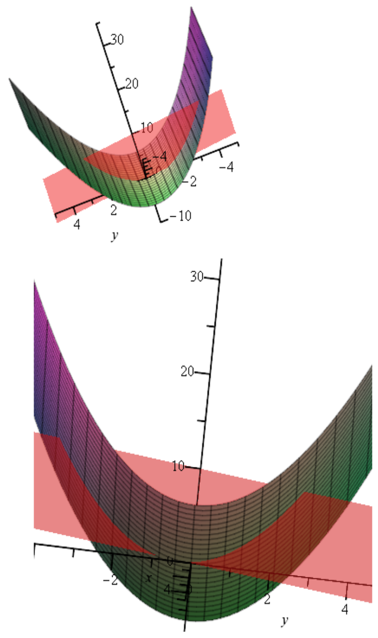


Fig. 16.2 Two views of the graph of $y^2 + 2x$ intersected with the transparent red plane $2x - y = 0$. The solution to Example 16.2 is apex of the intersection curve, namely the point $(-\frac{1}{4}, -\frac{1}{2}, -\frac{15}{16})$.

Thus, the unique minimum of our problem is achieved for $(x = -1/4, y = -1/2)$. The Largrangian of our problem is

$$L(x, y, \lambda) = y^2 + 2x + \lambda(2x - y).$$

If we apply the dual ascent method, minimization of $L(x, y, \lambda)$ with respect to x and y holding λ constant yields the equations

$$\begin{aligned} 2 + 2\lambda &= 0 \\ 2y - \lambda &= 0, \end{aligned}$$

obtained by setting the gradient of L (with respect to x and y) to zero. If $\lambda \neq -1$, the problem has no solution. Indeed, if $\lambda \neq -1$, minimizing $L(x, y, \lambda) = y^2 + 2x + \lambda(2x - y)$ with respect to x and y yields $-\infty$.

The augmented Lagrangian is

$$\begin{aligned} L_\rho(x, y, \lambda) &= y^2 + 2x + \lambda(2x - y) + (\rho/2)(2x - y)^2 \\ &= 2\rho x^2 - 2\rho xy + 2(1 + \lambda)x - \lambda y + \left(1 + \frac{\rho}{2}\right) y^2, \end{aligned}$$

which in matrix form is

$$L_\rho(x, y, \lambda) = \begin{pmatrix} x & y \end{pmatrix} \begin{pmatrix} 2\rho^2 & -\rho \\ -\rho & 1 + \dfrac{\rho}{2} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 2(1+\lambda) & -\lambda \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}.$$

The trace of the above matrix is $1 + \frac{\rho}{2} + 2\rho^2 > 0$, and the determinant is

$$2\rho^2 \left(1 + \frac{\rho}{2}\right) - \rho^2 = \rho^2(1 + \rho) > 0,$$

since $\rho > 0$. Therefore, the above matrix is symmetric positive definite. Minimizing $L_\rho(x, y, \lambda)$ with respect to x and y, we set the gradient of $L_\rho(x, y, \lambda)$ (with respect to x and y) to zero, and we obtain the equations:

$$\begin{aligned} 2\rho x - \rho y + (1 + \lambda) &= 0 \\ -2\rho x + (2 + \rho)y - \lambda &= 0. \end{aligned}$$

The solution is

$$x = -\frac{1}{4} - \frac{1 + \lambda}{2\rho}, \quad y = -\frac{1}{2}.$$

Thus the steps for the method of multipliers are

$$\begin{aligned} x^{k+1} &= -\frac{1}{4} - \frac{1 + \lambda^k}{2\rho} \\ y^{k+1} &= -\frac{1}{2} \\ \lambda^{k+1} &= \lambda^k + \rho \begin{pmatrix} 2 & -1 \end{pmatrix} \begin{pmatrix} -\frac{1}{4} - \frac{1+\lambda^k}{2\rho} \\ -\frac{1}{2} \end{pmatrix}, \end{aligned}$$

and the second step simplifies to

$$\lambda^{k+1} = -1.$$

Consequently, we see that the method converges after two steps for any initial value of λ^0, and we get

$$x = -\frac{1}{4} \quad y = -\frac{1}{2}, \quad \lambda = -1.$$

The method of multipliers also converges for functions J that are not even convex, as illustrated by the next example.

Example 16.3. Consider the minimization problem

$$\begin{aligned} &\text{minimize} \quad 2\beta xy \\ &\text{subject to} \quad 2x - y = 0, \end{aligned}$$

with $\beta > 0$. See Figure 16.3.

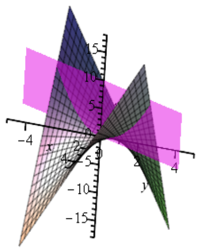
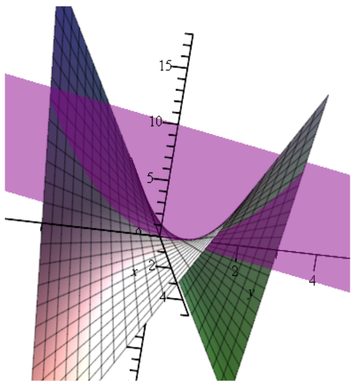



Fig. 16.3 Two views of the graph of the saddle of $2xy$ ($\beta = 1$) intersected with the transparent magenta plane $2x - y = 0$. The solution to Example 16.3 is apex of the intersection curve, namely the point $(0, 0, 0)$.

The quadratic function

$$J(x, y) = 2\beta xy = \begin{pmatrix} x & y \end{pmatrix} \begin{pmatrix} 0 & \beta \\ \beta & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

is not convex because the above matrix is not even positive semidefinite (the eigenvalues of the matrix are $-\beta$ and $+\beta$). The augmented Lagrangian is

$$\begin{aligned} L_\rho(x, y, \lambda) &= 2\beta xy + \lambda(2x - y) + (\rho/2)(2x - y)^2 \\ &= 2\rho x^2 + 2(\beta - \rho)xy + 2\lambda x - \lambda y + \frac{\rho}{2} y^2, \end{aligned}$$

which in matrix form is

$$L_\rho(x, y, \lambda) = \begin{pmatrix} x & y \end{pmatrix} \begin{pmatrix} 2\rho & \beta - \rho \\ \beta - \rho & \dfrac{\rho}{2} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 2\lambda & -\lambda \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}.$$

The trace of the above matrix is $2\rho + \frac{\rho}{2} = \frac{5}{2}\rho > 0$, and the determinant is

$$\rho^2 - (\beta - \rho)^2 = \beta(2\rho - \beta).$$

This determinant is positive if $\rho > \beta/2$, in which case the matrix is symmetric positive definite. Minimizing $L_\rho(x, y, \lambda)$ with respect to x and y, we set the gradient of $L_\rho(x, y, \lambda)$ (with respect to x and y) to zero, and we obtain the equations:

$$\begin{aligned} 2\rho x + (\beta - \rho)y + \lambda &= 0 \\ 2(\beta - \rho)x + \rho y - \lambda &= 0. \end{aligned}$$

Since we are assuming that $\rho > \beta/2$, the solutions are

$$x = -\frac{\lambda}{2(2\rho - \beta)}, \quad y = \frac{\lambda}{(2\rho - \beta)}.$$

Thus the steps for the method of multipliers are

$$\begin{aligned} x^{k+1} &= -\frac{\lambda^k}{2(2\rho - \beta)} \\ y^{k+1} &= \frac{\lambda^k}{(2\rho - \beta)} \\ \lambda^{k+1} &= \lambda^k + \frac{\rho}{2(2\rho - \beta)} \begin{pmatrix} 2 & -1 \end{pmatrix} \begin{pmatrix} -\lambda^k \\ 2\lambda^k \end{pmatrix}, \end{aligned}$$

and the second step simplifies to

$$\lambda^{k+1} = \lambda^k + \frac{\rho}{2(2\rho - \beta)}(-4\lambda^k),$$

that is,

$$\lambda^{k+1} = -\frac{\beta}{2\rho - \beta}\lambda^k.$$

If we pick $\rho > \beta > 0$, which implies that $\rho > \beta/2$, then

$$\frac{\beta}{2\rho - \beta} < 1,$$

and the method converges for any intial value λ^0 to the solution

$$x = 0, \quad y = 0, \quad \lambda = 0.$$

Indeed, since the constraint $2x - y = 0$ holds, $2\beta xy = 4\beta x^2$, and the minimum of the function $x \mapsto 4\beta x^2$ is achieved for $x = 0$ (since $\beta > 0$).

As an exercise, the reader should verify that dual ascent (with $\alpha^k = \rho$) yields the equations

$$\begin{aligned} x^{k+1} &= \frac{\lambda^k}{2\beta} \\ y^{k+1} &= -\frac{\lambda^k}{\beta} \\ \lambda^{k+1} &= \left(1 + \frac{2\rho}{\beta}\right)\lambda^k, \end{aligned}$$

and so the method diverges, except for $\lambda^0 = 0$, which is the optimal solution.

The method of multipliers converges under conditions that are far more general than the dual ascent. However, the addition of the penalty term has the negative effect that even if J is separable, then the Lagrangian L_ρ is not separable. Thus the basic method of multipliers cannot be used for decomposition and is not parallelizable. The next method deals with the problem of separability.

16.3 ADMM: Alternating Direction Method of Multipliers

The alternating direction method of multipliers, for short ADMM, combines the decomposability of dual ascent with the superior convergence properties of the method of multipliers. It can be viewed as an approximation of the method of multipliers, but it is generally superior.

The idea is to split the function J into two independent parts, as $J(x, z) = f(x) + g(z)$, and to consider the Minimization Problem (P_{admm}),

$$\begin{aligned} &\text{minimize} && f(x) + g(z) \\ &\text{subject to} && Ax + Bz = c, \end{aligned}$$

for some $p \times n$ matrix A, some $p \times m$ matrix B, and with $x \in \mathbb{R}^n$, $z \in \mathbb{R}^m$, and $c \in \mathbb{R}^p$. We also assume that f and g are convex. Further conditions will be added later.

As in the method of multipliers, we form the *augmented Lagrangian*

$$L_\rho(x, z, \lambda) = f(x) + g(z) + \lambda^\top (Ax + Bz - c) + (\rho/2) \left\| Ax + Bz - c \right\|_2^2,$$

with $\lambda \in \mathbb{R}^p$ and for some $\rho > 0$.

Given some initial values (z^0, λ^0), the *ADMM method* consists of the following iterative steps:

$$\begin{aligned} x^{k+1} &= \arg\min_x L_\rho(x, z^k, \lambda^k) \\ z^{k+1} &= \arg\min_z L_\rho(x^{k+1}, z, \lambda^k) \\ \lambda^{k+1} &= \lambda^k + \rho(Ax^{k+1} + Bz^{k+1} - c). \end{aligned}$$

Instead of performing a minimization step jointly over x and z, as the method of multipliers would in the step

$$(x^{k+1}, z^{k+1}) = \arg\min_{x,z} L_\rho(x, z, \lambda^k),$$

ADMM first performs an x-minimization step, and then a z-minimization step. Thus x and z are updated in an alternating or sequential fashion, which accounts for the term *alternating direction*.

The algorithm state in ADMM is (z^k, λ^k), in the sense that (z^{k+1}, λ^{k+1}) is a function of (z^k, λ^k). The variable x^{k+1} is an auxiliary variable which is used to compute z^{k+1} from (z^k, λ^k). The roles of x and z are not quite symmetric, since the update of x is done before the update of λ. By switching x and z, f and g and A and B, we obtain a variant of ADMM in which the order of the x-update step and the z-update step are reversed.

Example 16.4. Let us reconsider the problem of Example 16.2 to solve it using ADMM. We formulate the problem as

$$\begin{aligned} \text{minimize} \quad & 2x + z^2 \\ \text{subject to} \quad & 2x - z = 0, \end{aligned}$$

with $f(x) = 2x$ and $g(z) = z^2$. The augmented Lagrangian is given by

$$L_\rho(x, z, \lambda) = 2x + z^2 + 2\lambda x - \lambda z + 2\rho x^2 - 2\rho x z + \frac{\rho}{2} z^2.$$

The ADMM steps are as follows. The x-update is

$$x^{k+1} = \arg\min_x \left(2\rho x^2 - 2\rho x z^k + 2\lambda^k x + 2x\right),$$

and since this is a quadratic function in x, its minimum is achieved when the derivative of the above function (with respect to x) is zero, namely

$$x^{k+1} = \frac{1}{2} z^k - \frac{1}{2\rho}\lambda^k - \frac{1}{2\rho}. \tag{1}$$

The z-update is

$$z^{k+1} = \arg\min_z \left(z^2 + \frac{\rho}{2} z^2 - 2\rho x^{k+1} z - \lambda^k z\right),$$

and as for the x-step, the minimum is achieved when the derivative of the above function (with respect to z) is zero, namely

$$z^{k+1} = \frac{2\rho x^{k+1}}{\rho + 2} + \frac{\lambda^k}{\rho + 2}. \tag{2}$$

The λ-update is

$$\lambda^{k+1} = \lambda^k + \rho(2x^{k+1} - z^{k+1}). \tag{3}$$

Substituting the right hand side of (1) for x^{k+1} in (2) yields

$$z^{k+1} = \frac{\rho z^k}{\rho + 2} - \frac{1}{\rho + 2}. \tag{4}$$

Using (2), we obtain

$$2x^{k+1} - z^{k+1} = \frac{4x^{k+1}}{\rho + 2} - \frac{\lambda^k}{\rho + 2}, \tag{5}$$

and then using (3) we get

$$\lambda^{k+1} = \frac{2\lambda^k}{\rho+2} + \frac{4\rho x^{k+1}}{\rho+2}. \tag{6}$$

Substituting the right hand side of (1) for x^{k+1} in (6), we obtain

$$\lambda^{k+1} = \frac{2\rho z^k}{\rho+2} - \frac{2}{\rho+2}. \tag{7}$$

Equation (7) shows that z^k determines λ^{k+1}, and Equation (1) for $k+2$, along with Equation (4), shows that z^k also determines x^{k+2}. In particular, we find that

$$\begin{aligned} x^{k+2} &= \frac{1}{2} z^{k+1} - \frac{1}{2\rho}\lambda^{k+1} - \frac{1}{2\rho} \\ &= \frac{(\rho-2)z^k}{2(\rho+2)} - \frac{1}{\rho+2}. \end{aligned}$$

Thus is suffices to find the limit of the sequence (z^k). Since we already know from Example 16.2 that this limit is $-1/2$, using (4), we write

$$z^{k+1} = -\frac{1}{2} + \frac{\rho z^k}{\rho+2} - \frac{1}{\rho+2} + \frac{1}{2} = -\frac{1}{2} + \frac{\rho}{\rho+2}\left(\frac{1}{2} + z^k\right).$$

By induction, we deduce that

$$z^{k+1} = -\frac{1}{2} + \left(\frac{\rho}{\rho+2}\right)^{k+1}\left(\frac{1}{2} + z^0\right),$$

and since $\rho > 0$, we have $\rho/(\rho+2) < 1$, so the limit of the sequence (z^{k+1}) is indeed $-1/2$, and consequently the limit of (λ^{k+1}) is -1 and the limit of x^{k+2} is $-1/4$.

For ADMM to be practical, the x-minimization step and the z-minimization step have to be doable efficiently.

It is often convenient to write the ADMM updates in terms of the *scaled dual variable* $\mu = (1/\rho)\lambda$. If we define the *residual* as

$$r = Ax + bz - c,$$

then we have

$$\begin{aligned} \lambda^\top r + (\rho/2)\left\|r\right\|_2^2 &= (\rho/2)\left\|r + (1/\rho)\lambda\right\|_2^2 - (1/(2\rho))\left\|\lambda\right\|_2^2 \\ &= (\rho/2)\left\|r + \mu\right\|_2^2 - (\rho/2)\left\|\mu\right\|_2^2. \end{aligned}$$

The *scaled form of ADMM* consists of the following steps:

$$\begin{aligned}
x^{k+1} &= \arg\min_x \Big(f(x) + (\rho/2) \left\| Ax + Bz^k - c + \mu^k \right\|_2^2 \Big) \\
z^{k+1} &= \arg\min_z \Big(g(z) + (\rho/2) \left\| Ax^{k+1} + Bz - c + \mu^k \right\|_2^2 \Big) \\
\mu^{k+1} &= \mu^k + Ax^{k+1} + Bz^{k+1} - c.
\end{aligned}$$

If we define the *residual* r^k *at step* k as

$$r^k = Ax^k + Bz^k - c = \mu^k - \mu^{k-1} = (1/\rho)(\lambda^k - \lambda^{k-1}),$$

then we see that

$$r = u^0 + \sum_{j=1}^{k} r^j.$$

The formulae in the scaled form are often shorter than the formulae in the unscaled form.

We now discuss the convergence of ADMM.

16.4 Convergence of ADMM ⊛

Let us repeat the steps of ADMM: Given some initial (z^0, λ^0), do:

$$\begin{aligned}
x^{k+1} &= \arg\min_x L_\rho(x, z^k, \lambda^k) && (x\text{-update}) \\
z^{k+1} &= \arg\min_z L_\rho(x^{k+1}, z, \lambda^k) && (z\text{-update}) \\
\lambda^{k+1} &= \lambda^k + \rho(Ax^{k+1} + Bz^{k+1} - c). && (\lambda\text{-update})
\end{aligned}$$

The convergence of ADMM can be proven under the following three assumptions:

(1) The functions $f\colon \mathbb{R} \to \mathbb{R} \cup \{+\infty\}$ and $g\colon \mathbb{R} \to \mathbb{R} \cup \{+\infty\}$ are proper and closed convex functions (see Section 15.1) such that $\mathbf{relint}(\mathrm{dom}(f)) \cap \mathbf{relint}(\mathrm{dom}(g)) \neq \emptyset$.
(2) The $n \times n$ matrix $A^\top A$ is invertible and the $m \times m$ matrix $B^\top B$ is invertible. Equivalently, the $p \times n$ matrix A has rank n and the $p \times m$ matrix has rank m.
(3) The unaugmented Lagrangian $L_0(x, z, \lambda) = f(x) + g(z) + \lambda^\top(Ax + Bz - c)$ has a saddle point, which means there exists x^*, z^*, λ^* (not necessarily unique) such that

$$L_0(x^*, z^*, \lambda) \leq L_0(x^*, z^*, \lambda^*) \leq L_0(x, z, \lambda^*)$$

for all x, z, λ.

Recall that the augmented Lagrangian is given by

$$L_\rho(x, z, \lambda) = f(x) + g(z) + \lambda^\top (Ax + Bz - c) + (\rho/2) \left\| Ax + Bz - c \right\|_2^2 .$$

For z (and λ) fixed, we have

$$\begin{aligned} L_\rho(x, z, \lambda) &= f(x) + g(z) + \lambda^\top (Ax + Bz - c) \\ &\quad + (\rho/2)(Ax + Bz - c)^\top (Ax + Bz - c) \\ &= f(x) + (\rho/2)x^\top A^\top Ax + (\lambda^\top + \rho(Bz - c)^\top)Ax \\ &\quad + g(z) + \lambda^\top (Bz - c) + (\rho/2)(Bz - c)^\top (Bz - c). \end{aligned}$$

Assume that (1) and (2) hold. Since $A^\top A$ is invertible, then it is symmetric positive definite, and by Proposition 15.25 the x-minimization step has a unique solution (the minimization problem succeeds with a unique minimizer).

Similarly, for x (and λ) fixed, we have

$$\begin{aligned} L_\rho(x, z, \lambda) &= f(x) + g(z) + \lambda^\top (Ax + Bz - c) \\ &\quad + (\rho/2)(Ax + Bz - c)^\top (Ax + Bz - c) \\ &= g(z) + (\rho/2)z^\top B^\top Bz + (\lambda^\top + \rho(Ax - c)^\top)Bz \\ &\quad + f(x) + \lambda^\top (Ax - c) + (\rho/2)(Ax - c)^\top (Ax - c). \end{aligned}$$

Since $B^\top B$ is invertible, then it is symmetric positive definite, and by Proposition 15.25 the z-minimization step has a unique solution (the minimization problem succeeds with a unique minimizer).

By Theorem 15.10, Assumption (3) is equivalent to the fact that the KKT equations are satisfied by some triple (x^*, z^*, λ^*), namely

$$Ax^* + Bz^* - c = 0 \tag{$*$}$$

and

$$0 \in \partial f(x^*) + \partial g(z^*) + A^\top \lambda^* + B^\top \lambda^*, \tag{$\dagger$}$$

Assumption (3) is also equivalent to Conditions (a) and (b) of Theorem 15.10. In particular, our program has an optimal solution (x^*, z^*). By Theorem 15.12, λ^* is maximizer of the dual function $G(\lambda) = \inf_{x,z} L_0(x, z, \lambda)$ and strong duality holds, that is, $G(\lambda^*) = f(x^*) + g(z^*)$ (the duality gap is zero).

We will see after the proof of Theorem 16.1 that Assumption (2) is actually implied by Assumption (3). This allows us to prove a convergence result stronger than the convergence result proven in Boyd et al. [Boyd *et al.* (2010)] under the exact same assumptions (1) and (3).

Let p^* be the minimum value of $f + g$ over the convex set $\{(x, z) \in \mathbb{R}^{m+p} \mid Ax + Bz - c = 0\}$, and let (p^k) be the sequence given by $p^k = f(x^k) + g(z^k)$, and recall that $r^k = Ax^k + Bz^k - c$.

Our main goal is to prove the following result.

Theorem 16.1. *Suppose the following assumptions hold:*

(1) The functions $f\colon \mathbb{R} \to \mathbb{R} \cup \{+\infty\}$ and $g\colon \mathbb{R} \to \mathbb{R} \cup \{+\infty\}$ are proper and closed convex functions (see Section 15.1) such that $\mathbf{relint}(\mathrm{dom}(f)) \cap \mathbf{relint}(\mathrm{dom}(g)) \neq \emptyset$.

(2) The $n \times n$ matrix $A^\top A$ is invertible and the $m \times m$ matrix $B^\top B$ is invertible. Equivalently, the $p \times n$ matrix A has rank n and the $p \times m$ matrix has rank m. (This assumption is actually redundant, because it is implied by Assumption (3)).

(3) The unaugmented Lagrangian $L_0(x, z, \lambda) = f(x) + g(z) + \lambda^\top(Ax + Bz - c)$ has a saddle point, which means there exists x^*, z^*, λ^* (not necessarily unique) such that

$$L_0(x^*, z^*, \lambda) \leq L_0(x^*, z^*, \lambda^*) \leq L_0(x, z, \lambda^*)$$

for all x, z, λ.

Then for any initial values (z^0, λ^0), the following properties hold:

(1) The sequence (r^k) converges to 0 (residual convergence).

(2) The sequence (p^k) converge to p^* (objective convergence).

(3) The sequences (x^k) and (z^k) converge to an optimal solution $(\widetilde{x}, \widetilde{z})$ of Problem (P_{admm}) and the sequence (λ^k) converges an optimal solution $\widetilde{\lambda}$ of the dual problem (primal and dual variable convergence).

Proof. The core of the proof is due to Boyd et al. [Boyd *et al.* (2010)], but there are new steps because we have the stronger hypothesis (2), which yield the stronger result (3).

The proof consists of several steps. It is not possible to prove directly that the sequences (x^k), (z^k), and (λ^k) converge, so first we prove that the sequence (r^{k+1}) converges to zero, and that the sequences (Ax^{k+1}) and (Bz^{k+1}) also converge.

Step 1. Prove the inequality (A1) below.

Consider the sequence of reals (V^k) given by

$$V^k = (1/\rho) \left\|\lambda^k - \lambda^*\right\|_2^2 + \rho \left\|B(z^k - z^*)\right\|_2^2.$$

It can be shown that the V^k satisfy the following inequality:

$$V^{k+1} \leq V^k - \rho \left\|r^{k+1}\right\|_2^2 - \rho \left\|B(z^{k+1} - z^k)\right\|_2^2. \tag{A1}$$

This is rather arduous. Since a complete proof is given in Boyd et al. [Boyd *et al.* (2010)], we will only provide some of the key steps later.

Inequality (A1) shows that the sequence (V^k) in nonincreasing. If we write these inequalities for $k, k-1, \ldots, 0$, we have

$$\begin{aligned} V^{k+1} &\leq V^k - \rho \left\| r^{k+1} \right\|_2^2 - \rho \left\| B(z^{k+1} - z^k) \right\|_2^2 \\ V^k &\leq V^{k-1} - \rho \left\| r^k \right\|_2^2 - \rho \left\| B(z^k - z^{k-1}) \right\|_2^2 \\ &\vdots \\ V^1 &\leq V^0 - \rho \left\| r^1 \right\|_2^2 - \rho \left\| B(z^1 - z^0) \right\|_2^2, \end{aligned}$$

and by adding up these inequalities, we obtain

$$V^{k+1} \leq V^0 - \rho \sum_{j=0}^{k} \left(\left\| r^{j+1} \right\|_2^2 + \left\| B(z^{j+1} - z^j) \right\|_2^2 \right),$$

which implies that

$$\rho \sum_{j=0}^{k} \left(\left\| r^{j+1} \right\|_2^2 + \left\| B(z^{j+1} - z^j) \right\|_2^2 \right) \leq V_0 - V^{k+1} \leq V^0, \tag{B}$$

since $V^{k+1} \leq V^0$.

Step 2. Prove that the sequence (r^k) converges to 0, and that the sequences (Ax^{k+1}) and (Bz^{k+1}) also converge.

Inequality (B) implies that the series $\sum_{k=1}^{\infty} r^k$ and $\sum_{k=0}^{\infty} B(z^{k+1} - z^k)$ converge absolutely. In particular, the sequence (r^k) converges to 0.

The nth partial sum of the series $\sum_{k=0}^{\infty} B(z^{k+1} - z^k)$ is

$$\sum_{k=0}^{n} B(z^{k+1} - z^k) = B(z^{n+1} - z^0),$$

and since the series $\sum_{k=0}^{\infty} B(z^{k+1} - z^k)$ converges, we deduce that the sequence (Bz^{k+1}) converges. Since $Ax^{k+1} + Bz^{k+1} - c = r^{k+1}$, the convergence of (r^{k+1}) and (Bz^{k+1}) implies that the sequence (Ax^{k+1}) also converges.

Step 3. Prove that the sequences (x^{k+1}) and (z^{k+1}) converge. By Assumption (2), the matrices $A^\top A$ and $B^\top B$ are invertible, so multiplying each vector Ax^{k+1} by $(A^\top A)^{-1} A^\top$, if the sequence (Ax^{k+1}) converges to u, then the sequence (x^{k+1}) converges to $(A^\top A)^{-1} A^\top u$. Siimilarly, if the sequence (Bz^{k+1}) converges to v, then the sequence (z^{k+1}) converges to $(B^\top B)^{-1} B^\top v$.

Step 4. Prove that the sequence (λ^k) converges.

Recall that

$$\lambda^{k+1} = \lambda^k + \rho r^{k+1}.$$

It follows by induction that

$$\lambda^{k+p} = \lambda^k + \rho(r^{k+1} + \cdots + \rho^{k+p}), \quad p \geq 2.$$

As a consequence, we get

$$\left\|\lambda^{k+p} - \lambda^k\right\| \leq \rho(\left\|r^{k+1}\right\| + \cdots + \left\|r^{k+p}\right\|).$$

Since the series $\sum_{k=1}^{\infty} \left\|r^k\right\|$ converges, the partial sums form a Cauchy sequence, and this immediately implies that for any $\epsilon > 0$ we can find $N > 0$ such that

$$\rho(\left\|r^{k+1}\right\| + \cdots + \left\|r^{k+p}\right\|) < \epsilon, \quad \text{for all } k, p + k \geq N,$$

so the sequence (λ^k) is also a Cauchy sequence, thus it converges.

Step 5. Prove that the sequence (p^k) converges to p^*.

For this, we need two more inequalities. Following Boyd et al. [Boyd *et al.* (2010)], we need to prove that

$$p^{k+1} - p^* \leq -(\lambda^{k+1})^\top r^{k+1} - \rho(B(z^{k+1} - z^k))^\top(-r^{k+1} + B(z^{k+1} - z^*)) \quad \text{(A2)}$$

and

$$p^* - p^{k+1} \leq (\lambda^*)^\top r^{k+1}. \quad \text{(A3)}$$

Since we proved that the sequence (r^k) and $B(z^{k+1} - z^k)$ converge to 0, and that the sequence (λ^{k+1}) converges, from

$$\begin{aligned}(\lambda^{k+1})^\top r^{k+1} + \rho(B(z^{k+1} - z^k))^\top(-r^{k+1} + B(z^{k+1} - z^*)) &\leq p^* - p^{k+1} \\ &\leq (\lambda^*)^\top r^{k+1},\end{aligned}$$

we deduce that in the limit, p^{k+1} converges to p^*.

Step 6. Prove (A3).

Since (x^*, y^*, λ^*) is a saddle point, we have

$$L_0(x^*, z^*, \lambda^*) \leq L_0(x^{k+1}, z^{k+1}, \lambda^*).$$

Since $Ax^* + Bz^* = c$, we have $L_0(x^*, z^*, \lambda^*) = p^*$, and since $p^{k+1} = f(x^{k+1}) + g(z^{k+1})$, we have

$$L_0(x^{k+1}, z^{k+1}, \lambda^*) = p^{k+1} + (\lambda^*)^\top r^{k+1},$$

so we obtain

$$p^* \leq p^{k+1} + (\lambda^*)^\top r^{k+1},$$

which yields (A3).

Step 7. Prove (A2).

By Proposition 15.24, z^{k+1} minimizes $L_\rho(x^{k+1}, z, \lambda^k)$ iff

$$\begin{aligned}0 &\in \partial g(z^{k+1}) + B^\top \lambda^k + \rho B^\top (Ax^{k+1} + Bz^{k+1} - c)\\ &= \partial g(z^{k+1}) + B^\top \lambda^k + \rho B^\top r^{k+1}\\ &= \partial g(z^{k+1}) + B^\top \lambda^{k+1},\end{aligned}$$

since $r^{k+1} = Ax^{k+1} + Bz^{k+1} - c$ and $\lambda^{k+1} = \lambda^k + \rho(Ax^{k+1} + Bz^{k+1} - c)$.

In summary, we have

$$0 \in \partial g(z^{k+1}) + B^\top \lambda^{k+1}, \tag{$\dagger_1$}$$

which shows that z^{k+1} minimizes the function

$$z \mapsto g(z) + (\lambda^{k+1})^\top Bz.$$

Consequently, we have

$$g(z^{k+1}) + (\lambda^{k+1})^\top Bz^{k+1} \leq g(z^*) + (\lambda^{k+1})^\top Bz^*. \tag{B1}$$

Similarly, x^{k+1} minimizes $L_\rho(x, z^k, \lambda^k)$ iff

$$\begin{aligned}0 &\in \partial f(x^{k+1}) + A^\top \lambda^k + \rho A^\top (Ax^{k+1} + Bz^k - c)\\ &= \partial f(x^{k+1}) + A^\top (\lambda^k + \rho r^{k+1} + \rho B(z^k - z^{k+1}))\\ &= \partial f(x^{k+1}) + A^\top \lambda^{k+1} + \rho A^\top B(z^k - z^{k+1})\end{aligned}$$

since $r^{k+1} - Bz^{k+1} = Ax^{k+1} - c$ and $\lambda^{k+1} = \lambda^k + \rho(Ax^{k+1} + Bz^{k+1} - c) = \lambda^k + \rho r^{k+1}$.

Equivalently, the above derivation shows that

$$0 \in \partial f(x^{k+1}) + A^\top (\lambda^{k+1} - \rho B(z^{k+1} - z^k)), \tag{$\dagger_2$}$$

which shows that x^{k+1} minimizes the function

$$x \mapsto f(x) + (\lambda^{k+1} - \rho B(z^{k+1} - z^k))^\top Ax.$$

Consequently, we have

$$\begin{aligned}&f(x^{k+1}) + (\lambda^{k+1} - \rho B(z^{k+1} - z^k))^\top Ax^{k+1}\\ &\qquad\qquad \leq f(x^*) + (\lambda^{k+1} - \rho B(z^{k+1} - z^k))^\top Ax^*.\end{aligned} \tag{B2}$$

Adding up Inequalities (B1) and (B2), using the equation $Ax^* + Bz^* = c$, and rearranging, we obtain inequality (A2).

Step 8. Prove that $(x^k), (z^k)$, and (λ^k) converge to optimal solutions.

Recall that (r^k) converges to 0, and that (x^k), (z^k), and (λ^k) converge to limits $\widetilde{x}$, $\widetilde{z}$, and $\widetilde{\lambda}$. Since $r^k = Ax^k + Bz^k - c$, in the limit, we have

$$A\widetilde{x} + B\widetilde{z} - c = 0. \tag{$*_1$}$$

Using ($\dagger_1$), in the limit, we obtain

$$0 \in \partial g(\widetilde{z}) + B^\top \widetilde{\lambda}. \tag{$*_2$}$$

Since $(B(z^{k+1} - z^k))$ converges to 0, using ($\dagger_2$), in the limit, we obtain

$$0 \in \partial f(\widetilde{x}) + A^\top \widetilde{\lambda}. \tag{$*_3$}$$

From ($*_2$) and ($*_3$), we obtain

$$0 \in \partial f(\widetilde{x}) + \partial g(\widetilde{z}) + A^\top \widetilde{\lambda} + B^\top \widetilde{\lambda}. \tag{$*_4$}$$

But ($*_1$) and ($*_4$) are exactly the KKT equations, and by Theorem 15.10, we conclude that $\widetilde{x}, \widetilde{z}, \widetilde{\lambda}$ are optimal solutions.

Step 9. Prove (A1). This is the most tedious step of the proof. We begin by adding up (A2) and (A3), and then perform quite a bit or rewriting and manipulation. The complete derivation can be found in Boyd et al. [Boyd *et al.* (2010)]. $\square$

Remarks:

(1) In view of Theorem 15.11, we could replace Assumption (3) by the slightly stronger assumptions that the optimum value of our program is finite and that the constraints are qualified. Since the constraints are affine, this means that there is some feasible solution in $\mathbf{relint}(\mathrm{dom}(f)) \cap \mathbf{relint}(\mathrm{dom}(g))$. These assumptions are more practical than Assumption (3).

(2) Actually, Assumption (3) implies Assumption (2). Indeed, we know from Theorem 15.10 that the existence of a saddle point implies that our program has a finite optimal solution. However, if either $A^\top A$ or $B^\top B$ is not invertible, then Program (P) may not have a finite optimal solution, as shown by the following counterexample.

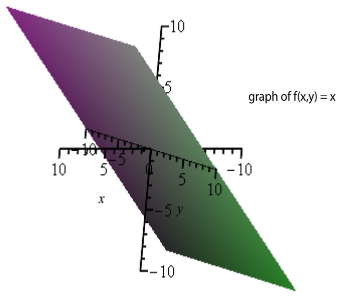
Example 16.5. Let

$$f(x, y) = x, \quad g(z) = 0, \quad y - z = 0.$$

Then

$$L_\rho(x, y, z, \lambda) = x + \lambda(y - z) + (\rho/2)(y - z)^2,$$

but minimizing over (x, y) with z held constant yields $-\infty$, which implies that the above program has no finite optimal solution. See Figure 16.4.



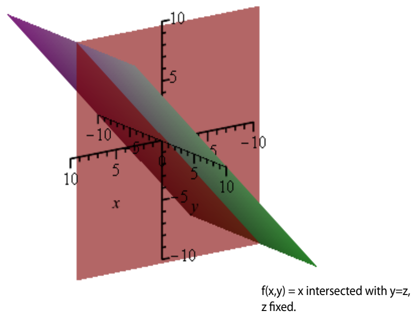


Fig. 16.4 A graphical representation of the Example 16.5. This is an illustration of the x minimization step when z is held fixed. Since the intersection of the two planes is an unbounded line, we "see" that minimizing over x yields $-\infty$.

The problem is that

$$A = \begin{pmatrix} 0 \ 1 \end{pmatrix}, \quad B = \begin{pmatrix} -1 \end{pmatrix},$$

but

$$A^\top A = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$$

is not invertible.

(3) Proving (A1), (A2), (A3), and the convergence of (r^k) to 0 and of (p^k) to p^* does not require Assumption (2). The proof, using the ingeneous Inequality (A1) (and (B)) is the proof given in Boyd et al. [Boyd *et al.* (2010)]. We were also able to prove that (λ^k), (Ax^k) and (Bz^k) converge without Assumption (2), but to prove that (x^k), (y^k), and (λ^k) converge to optimal solutions, we had to use Assumption (2).

(4) Bertsekas discusses ADMM in [Bertsekas (2015)], Sections 2.2 and 5.4. His formulation of ADMM is slightly different, namely

$$\begin{aligned} &\text{minimize} && f(x) + g(z) \\ &\text{subject to} && Ax = z. \end{aligned}$$

Bertsekas states a convergence result for this version of ADMM under the hypotheses that either $\text{dom}(f)$ is compact or that $A^\top A$ is invertible, and that a saddle point exists; see Proposition 5.4.1. The proof is given in Bertsekas [Bertsekas and Tsitsiklis (1997)], Section 3.4, Proposition 4.2. It appears that the proof makes use of gradients, so it is not clear that it applies in the more general case where f and g are not differentiable.

(5) Versions of ADMM are discussed in Gabay [Gabay (1983)] (Sections 4 and 5). They are more general than the version discussed here. Some convergence proofs are given, but because Gabay's framework is more general, it is not clear that they apply to our setting. Also, these proofs rely on earlier result by Lions and Mercier, which makes the comparison difficult.

(5) Assumption (2) does not imply that the system $Ax + Bz = c$ has any solution. For example, if

$$A = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad B = \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \quad c = \begin{pmatrix} 1 \\ 0 \end{pmatrix},$$

the system

$$\begin{aligned} x - z &= 1 \\ x - z &= 0 \end{aligned}$$

has no solution. However, since Assumption (3) implies that the program has an optimal solution, it implies that c belongs to the column space of the $p \times (n + m)$ matrix $\begin{pmatrix} A & B \end{pmatrix}$.

Here is an example where ADMM diverges for a problem whose optimum value is $-\infty$.

Example 16.6. Consider the problem given by

$$f(x) = x, \quad g(z) = 0, \quad x - z = 0.$$

Since $f(x) + g(z) = x$, and $x = z$, the variable x is unconstrained and the above function goes to $-\infty$ when x goes to $-\infty$. The augmented Lagrangian is

$$\begin{aligned}L_\rho(x,z,\lambda) &= x + \lambda(x-z) + \frac{\rho}{2}(x-z)^2\\ &= \frac{\rho}{2}x^2 - \rho xz + \frac{\rho}{2}z^2 + x + \lambda x - \lambda z.\end{aligned}$$

The matrix

$$\begin{pmatrix} \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{pmatrix}$$

is singular and $L_\rho(x,z,\lambda)$ goes to $-\infty$ in when $(x,z) = t(1,1)$ and t goes to $-\infty$. The ADMM steps are:

$$\begin{aligned}x^{k+1} &= z^k - \frac{1}{\rho}\lambda^k - \frac{1}{\rho}\\ z^{k+1} &= x^{k+1} + \frac{1}{\rho}\lambda^k\\ \lambda^{k+1} &= \lambda^k + \rho(x^{k+1} - z^{k+1}),\end{aligned}$$

and these equations hold for all $k \geq 0$. From the last two equations we deduce that

$$\lambda^{k+1} = \lambda^k + \rho(x^{k+1} - z^{k+1}) = \lambda^k + \rho(-\frac{1}{\rho}\lambda^k) = 0, \quad \text{for all } k \geq 0,$$

so

$$z^{k+2} = x^{k+2} + \frac{1}{\rho}\lambda^{k+1} = x^{k+2}, \quad \text{for all } k \geq 0.$$

Consequently we find that

$$x^{k+3} = z^{k+2} + \frac{1}{\rho}\lambda^{k+2} - \frac{1}{\rho} = x^{k+2} - \frac{1}{\rho}.$$

By induction, we obtain

$$x^{k+3} = x^2 - \frac{k+1}{\rho}, \quad \text{for all } k \geq 0,$$

which shows that x^{k+3} goes to $-\infty$ when k goes to infinity, and since $x^{k+2} = z^{k+2}$, similarly z^{k+3} goes to $-\infty$ when k goes to infinity.

16.5 Stopping Criteria

Going back to Inequality (A2),

$$p^{k+1} - p^* \leq -(\lambda^{k+1})^\top r^{k+1} - \rho(B(z^{k+1} - z^k))^\top(-r^{k+1} + B(z^{k+1} - z^*)), \quad \text{(A2)}$$

using the fact that $Ax^* + Bz^* - c = 0$ and $r^{k+1} = Ax^{k+1} + Bz^{k+1} - c$, we have

$$\begin{aligned}-r^{k+1} + B(z^{k+1} - z^*) &= -Ax^{k+1} - Bz^{k+1} + c + B(z^{k+1} - z^*)\\ &= -Ax^{k+1} + c - Bz^*\\ &= -Ax^{k+1} + Ax^* = -A(x^{k+1} - x^*),\end{aligned}$$

so (A2) can be rewritten as

$$p^{k+1} - p^* \leq -(\lambda^{k+1})^\top r^{k+1} + \rho(B(z^{k+1} - z^k))^\top A(x^{k+1} - x^*),$$

or equivalently as

$$p^{k+1} - p^* \leq -(\lambda^{k+1})^\top r^{k+1} + (x^{k+1} - x^*)^\top \rho A^\top B(z^{k+1} - z^k). \tag{s_1}$$

We define the *dual residual* as

$$s^{k+1} = \rho A^\top B(z^{k+1} - z^k),$$

the quantity $r^{k+1} = Ax^{k+1} + Bz^{k+1} - c$ being the *primal residual*. Then (s_1) can be written as

$$p^{k+1} - p^* \leq -(\lambda^{k+1})^\top r^{k+1} + (x^{k+1} - x^*)^\top s^{k+1}. \tag{s}$$

Inequality (s) shows that when the residuals r^k and s^k are small, then p^k is close to p^* from below. Since x^* is unknown, we can't use this inequality, but if we have a guess that $\left\|x^k - x^*\right\| \leq d$, then using Cauchy–Schwarz we obtain

$$p^{k+1} - p^* \leq \left\|\lambda^{k+1}\right\| \left\|r^{k+1}\right\| + d\left\|s^{k+1}\right\|.$$

The above suggests that a reasonable termination criterion is that $\left\|r^k\right\|$ and $\left\|s^k\right\|$ should be small, namely that

$$\left\|r^k\right\| \leq \epsilon^{\text{pri}} \quad \text{and} \quad \left\|s^k\right\| \leq \epsilon^{\text{dual}},$$

for some chosen feasibility tolerances ϵ^{pri} and ϵ^{dual}. Further discussion for choosing these parameters can be found in Boyd et al. [Boyd *et al.* (2010)] (Section 3.3.1).

Various extensions and variations of ADMM are discussed in Boyd et al. [Boyd *et al.* (2010)] (Section 3.4). In order to accelerate convergence of the method, one may choose a different ρ at each step (say ρ^k), although proving the convergence of such a method may be difficult. If we assume that ρ^k becomes constant after a number of iterations, then the proof that we gave still applies. A simple scheme is this:

$$\rho^{k+1} = \begin{cases} \tau^{\text{incr}}\rho^k & \text{if } \left\|r^k\right\| > \mu\left\|s^k\right\| \\ \rho^k/\tau^{\text{decr}} & \text{if } \left\|s^k\right\| > \mu\left\|r^k\right\| \\ \rho^k & \text{otherwise,} \end{cases}$$

where $\tau^{\text{incr}} > 1, \tau^{\text{decr}} > 1$, and $\mu > 1$ are some chosen parameters. Again, we refer the interested reader to Boyd et al. [Boyd *et al.* (2010)] (Section 3.4).

16.6 Some Applications of ADMM

Structure in f, g, A, and B can often be exploited to yield more efficient methods for performing the x-update and the z-update. We focus on the x-update, but the discussion applies just as well to the z-update. Since z and λ are held constant during minimization over x, it is more convenient to use the scaled form of ADMM. Recall that

$$x^{k+1} = \arg\min_x \Big(f(x) + (\rho/2) \left\|Ax + Bz^k - c + u^k\right\|_2^2\Big)$$

(here we use u instead of μ), so we can express the x-update step as

$$x^+ = \arg\min_x \Big(f(x) + (\rho/2) \left\|Ax - v\right\|_2^2\Big),$$

with $v = -Bz^k + c - u^k$.

Example 16.7. A first simplification arises when $A = I$, in which case the x-update is

$$x^+ = \arg\min_x \Big(f(x) + (\rho/2) \left\|x - v\right\|_2^2\Big) = \mathbf{prox}_{f,\rho}(v).$$

The map $v \mapsto \mathbf{prox}_{f,\rho}(v)$ is known as the *proximity operator of f with penalty ρ*. The above minimization is generally referred to as *proximal minimization*.

Example 16.8. When the function f is simple enough, the proximity operator can be computed analytically. This is the case in particular when $f = I_C$, the indicator function of a nonempty closed convex set C. In this case, it is easy to see that

$$x^+ = \arg\min_x \Big(I_C(x) + (\rho/2) \left\|x - v\right\|_2^2\Big) = \Pi_C(v),$$

the orthogonal projection of v onto C. In the special case where $C = \mathbb{R}^n_+$ (the first orthant), then

$$x^+ = (v)_+,$$

the vector obtained by setting the negative components of v to zero.

Example 16.9. A second case where simplifications arise is the case where f is a convex quadratic functional of the form

$$f(x) = \frac{1}{2}x^\top Px + q^\top x + r,$$

where P is an $n \times n$ symmetric positive semidefinite matrix, $q \in \mathbb{R}^n$ and $r \in \mathbb{R}$. In this case the gradient of the map

$$x \mapsto f(x) + (\rho/2) \left\|Ax - v\right\|_2^2 = \frac{1}{2}x^\top Px + q^\top x + r + \frac{\rho}{2}x^\top(A^\top A)x \\ - \rho x^\top A^\top v + \frac{\rho}{2}v^\top v$$

is given by

$$(P + \rho A^\top A)x + q - \rho A^\top v,$$

and since A has rank n, the matrix $A^\top A$ is symmetric positive definite, so we get

$$x^+ = (P + \rho A^\top A))^{-1}(\rho A^\top v - q).$$

Methods from numerical linear algebra can be used so compute x^+ fairly efficiently; see Boyd et al. [Boyd *et al.* (2010)] (Section 4).

Example 16.10. A third case where simplifications arise is the variation of the previous case where f is a convex quadratic functional of the form

$$f(x) = \frac{1}{2}x^\top Px + q^\top x + r,$$

except that f is constrained by equality constraints $Cx = b$, as in Section 14.4, which means that $\text{dom}(f) = \{x \in \mathbb{R}^n \mid Cx = b\}$, and $A = I$. The x-minimization step consists in minimizing the function

$$J(x) = \frac{1}{2}x^\top Px + q^\top x + r + \frac{\rho}{2}x^\top x - \rho x^\top v + \frac{\rho}{2}v^\top v$$

subject to the constraint

$$Cx = b,$$

so by the results of Section 14.4, x^+ is a component of the solution of the KKT-system

$$\begin{pmatrix} P + \rho I & C^\top \\ C & 0 \end{pmatrix} \begin{pmatrix} x^+ \\ y \end{pmatrix} = \begin{pmatrix} -q + \rho v \\ b \end{pmatrix}.$$

The matrix $P + \rho I$ is symmetric positive definite, so the KKT-matrix is invertible.

We can now describe how ADMM is used to solve two common problems of convex optimization.

(1) *Minimization of a proper closed convex function f over a closed convex set C in $\mathbb{R}^n$.* This is the following problem

$$\begin{aligned} &\text{minimize} \quad f(x) \\ &\text{subject to} \ \ x \in C, \end{aligned}$$

which can be rewritten in AADM form as

$$\begin{aligned} &\text{minimize} \quad f(x) + I_C(z) \\ &\text{subject to} \ \ x - z = 0. \end{aligned}$$

Using the scaled dual variable $u = \lambda/\rho$, the augmented Lagrangian is

$$L_\rho(x, z, u) = f(x) + I_C(z) + \frac{\rho}{2} \|x - z + u\|_2^2 - \frac{\rho}{2} \|u\|^2 .$$

In view of Example 16.8, the scaled form of ADMM for this problem is

$$\begin{aligned} x^{k+1} &= \arg\min_x \Big(f(x) + (\rho/2) \left\|x - z^k + u^k\right\|_2^2 \Big) \\ z^{k+1} &= \Pi_C(x^{k+1} + u^k) \\ u^{k+1} &= u^k + x^{k+1} - z^{k+1}. \end{aligned}$$

The x-update involves evaluating a proximal operator. Note that the function f need not be differentiable. Of course, these minimizations depend on having efficient computational procedures for the proximal operator and the projection operator.

(2) *Quadratic Programming.* Here the problem is

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2} x^\top P x + q^\top x + r \\ &\text{subject to} \ \ Ax = b, \ x \geq 0, \end{aligned}$$

where P is an $n \times n$ symmetric positive semidefinite matrix, $q \in \mathbb{R}^n$, $r \in \mathbb{R}$, and A is an $m \times n$ matrix of rank m.
The above program is converted in ADMM form as follows:

$$\begin{aligned} &\text{minimize} \quad f(x) + g(z) \\ &\text{subject to} \ \ x - z = 0, \end{aligned}$$

with

$$f(x) = \frac{1}{2} x^\top P x + q^\top x + r, \quad \text{dom}(f) = \{x \in \mathbb{R}^n \mid Ax = b\},$$

and

$$g = I_{\mathbb{R}^n_+},$$

the indicator function of the positive orthant $\mathbb{R}^n_+$. In view of Example 16.8 and Example 16.10, the scaled form of ADMM consists of the following steps:

$$\begin{aligned}
x^{k+1} &= \arg\min_x \left(f(x) + (\rho/2)\left\|x - z^k + u^k\right\|_2^2\right)\\
z^{k+1} &= (x^{k+1} + u^k)_+\\
u^{k+1} &= u^k + x^{k+1} - z^{k+1}.
\end{aligned}$$

The x-update involves solving the KKT equations

$$\begin{pmatrix} P + \rho I & A^\top \\ A & 0 \end{pmatrix}\begin{pmatrix} x^{k+1} \\ y \end{pmatrix} = \begin{pmatrix} -q + \rho(z^k - u^k) \\ b \end{pmatrix}.$$

This is an important example because it provides one of the best methods for solving quadratic problems, in particular, the SVM problems discussed in Chapter 18.

We programmed the above method in `Matlab` as the function `qsolve1`, see Appendix B, Section B.1. Here are two examples.

Example 16.11. Consider the quadratic program for which

$$P_1 = \begin{pmatrix} 4 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 4 \end{pmatrix} \qquad q_1 = -\begin{pmatrix} 4 \\ 4 \\ 4 \end{pmatrix}$$

$$A_1 = \begin{pmatrix} 1 & 1 & -1 \\ 1 & -1 & -1 \end{pmatrix} \qquad b_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

We see immediately that the constraints

$$\begin{aligned}
x + y - z &= 0\\
x - y - z &= 0
\end{aligned}$$

imply that $z = x$ and $y = 0$. Then it is easy using calculus to find that the unique minimum is given by $(x, y, z) = (1, 0, 1)$. Running `qsolve1` on P_1, q_1, A_1, b_1 with $\rho = 10$, $tolr = tols = 10^{-12}$ and $iternum = 10000$, we find that after 83 iterations the primal and the dual residuals are less than 10^{-12}, and we get

$$(x, y, z) = (1.000000000000149, 0.000000000000000, 1.000000000000148).$$

Example 16.12. Consider the quadratic program for which

$$P_2 = \begin{pmatrix} 4 & 1 & 0 & 0 \\ 1 & 4 & 1 & 0 \\ 0 & 1 & 4 & 1 \\ 0 & 0 & 1 & 4 \end{pmatrix} \qquad q_2 = -\begin{pmatrix} 4 \\ 4 \\ 4 \\ 4 \end{pmatrix}$$

$$A_2 = \begin{pmatrix} 1 & 1 & -1 & 0 \\ 1 & -1 & -1 & 0 \end{pmatrix} \qquad b_2 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Again, we see immediately that the constraints imply that $z = x$ and $y = 0$. Then it is easy using calculus to find that the unique minimum is given by $(x, y, z) = (28/31, 0, 28/31, 24/31)$. Running `qsolve1` on P_2, q_2, A_2, b_2 with $\rho = 10$, $tolr = tols = 10^{-12}$ and $iternum = 10000$, we find that after 95 iterations the primal and the dual residuals are less than 10^{-12}, and we get

$$(x, y, z, t) = (0.903225806451495, 0.000000000000000,\\ 0.903225806451495, 0.774193548387264),$$

which agrees with the answer found earlier up to 11 decimals.

As an illustration of the wide range of applications of ADMM we show in the next section how to solve the hard margin SVM (SVM_{h2}) discussed in Section 14.6.

16.7 Solving Hard Margin (SVM_{h2}) Using ADMM

Recall that we would like to solve the following optimization problem (see Section 14.6):

Hard margin SVM (SVM_{h2}):

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2}\|w\|^2 \\ &\text{subject to} \\ &\qquad w^\top u_i - b \geq 1 \qquad i = 1, \ldots, p \\ &\qquad -w^\top v_j + b \geq 1 \qquad j = 1, \ldots, q. \end{aligned}$$

The margin is $\delta = 1/\|w\|$. The separating hyperplane $H_{w,b}$ is the hyperplane of equation $w^\top x - b = 0$, and the margin hyperplanes are the hyperplanes $H_{w,b+1}$ (the blue hyperplane) of equation $w^\top x - b - 1 = 0$ and $H_{w,b-1}$ (the red hyperplane) of equation $w^\top x - b + 1 = 0$. The dual program derived in Section 14.10 is the following program:

Dual of the Hard margin SVM (SVM$_{h2}$):

$$\begin{aligned}
&\text{minimize} \quad \frac{1}{2}\begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix} X^\top X \begin{pmatrix}\lambda \\ \mu\end{pmatrix} - \begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix} \mathbf{1}_{p+q} \\
&\text{subject to} \\
&\qquad \sum_{i=1}^p \lambda_i - \sum_{j=1}^q \mu_j = 0 \\
&\qquad \lambda \geq 0, \; \mu \geq 0,
\end{aligned}$$

where X is the $n \times (p+q)$ matrix given by

$$X = \begin{pmatrix} -u_1 & \cdots & -u_p & v_1 & \cdots & v_q \end{pmatrix}.$$

Then w is determined as follows:

$$w = -X\begin{pmatrix}\lambda \\ \mu\end{pmatrix} = \sum_{i=1}^p \lambda_i u_i - \sum_{j=1}^q \mu_j v_j.$$

To solve the dual using ADMM we need to determine the matrices P, q A and c as in Section 16.6(2). We renamed b as c to avoid a clash since b is already used. We have

$$P = X^\top X, \quad q = -\mathbf{1}_{p+q},$$

and since the only constraint is

$$\sum_{i=1}^p \lambda_i - \sum_{j=1}^q \mu_j = 0,$$

the matrix A is the $1 \times (p+q)$ row vector

$$A = \begin{pmatrix}\mathbf{1}_p^\top & -\mathbf{1}_q^\top\end{pmatrix},$$

and the right-hand side c is the scalar

$$c = 0.$$

Obviously the matrix A has rank 1. We obtain b using any i_0 such that $\lambda_{i_0} > 0$ and any j_0 such that $\mu_{j_0} > 0$. Since the corresponding constraints are active, we have

$$w^\top u_{i_0} - b = 1, \quad -w^\top v_{j_0} + b = 1,$$

so we obtain

$$b = w^\top (u_{i_0} + v_{j_0})/2.$$

For improved numerical stability, we can average over the sets of indices defined as $I_{\lambda>0} = \{i \in \{1, \ldots, p\} \mid \lambda_i > 0\}$ and $I_{\mu>0} = \{j \in \{1, \ldots, q\} \mid \mu_j > 0\}$. We obtain

$$b = w^\top \left(\Big(\sum_{i \in I_{\lambda>0}} u_i\Big)/|I_{\lambda>0}| + \Big(\sum_{j \in I_{\mu>0}} v_j\Big)/|I_{\mu>0}| \right)/2.$$

The `Matlab` programs implementing the above method are given in Appendix B, Section B.1. This should convince the reader that there is very little gap between theory and practice, although it is quite consuming to tune the tolerance parameters needed to deal with floating-point arithmetic.

Figure 16.5 shows the result of running the `Matlab` program implementing the above method using ADMM on two sets of points of 50 points each generated at random using the following `Matlab` code.

```
u14 = 10.1*randn(2,50)+18;
v14 = -10.1*randn(2,50)-18;
```

The function `SVMhard2` is called with $\rho = 10$ as follows

```
[lamb,mu,w] = SVMhard2(10,u14,v14)
```

and produces the output shown in Figure 16.5. Observe that there is one blue support vector and two red support vectors.

16.8 Applications of ADMM to ℓ^1-Norm Problems

Another important application of ADMM is to ℓ^1-norm minimization problems, especially lasso minimization, discussed below and in Section 19.4. This involves the special case of ADMM where $f(x) = \tau \|x\|_1$ and $A = I$. In particular, in the one-dimensional case, we need to solve the minimization problem: find

$$x^* = \arg\min_x \left(\tau|x| + (\rho/2)(x - v)^2\right),$$

with $x, v \in \mathbb{R}$, and $\rho, \tau > 0$. Let $c = \tau/\rho$ and write

$$f(x) = \frac{\tau}{2c}\left(2c|x| + (x - v)^2\right).$$

Minimizing f over x is equivalent to minimizing

$$g(x) = 2c|x| + (x - v)^2 = 2c|x| + x^2 - 2xv + v^2,$$

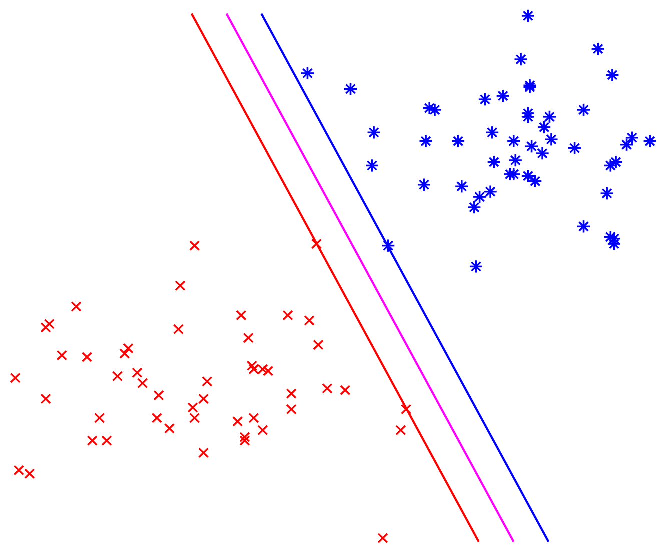


Fig. 16.5 An example of hard margin SVM.

which is equivalent to minimizing

$$h(x) = x^2 + 2(c|x| - xv)$$

over x. If $x \geq 0$, then

$$h(x) = x^2 + 2(cx - xv) = x^2 + 2(c - v)x = (x - (v - c))^2 - (v - c)^2.$$

If $v - c > 0$, that is, $v > c$, since $x \geq 0$, the function $x \mapsto (x - (v - c))^2$ has a minimum for $x = v - c > 0$, else if $v - c \leq 0$, then the function $x \mapsto (x - (v - c))^2$ has a minimum for $x = 0$.

If $x \leq 0$, then

$$h(x) = x^2 + 2(-cx - xv) = x^2 - 2(c + v)x = (x - (v + c))^2 - (v + c)^2.$$

if $v+c < 0$, that is, $v < -c$, since $x \leq 0$, the function $x \mapsto (x-(v+c))^2$ has a minimum for $x = v+c$, else if $v+c \geq 0$, then the function $x \mapsto (x-(v+c))^2$ has a minimum for $x = 0$.

In summary, $\inf_x h(x)$ is the function of v given by

$$S_c(v) = \begin{cases} v - c & \text{if } v > c \\ 0 & \text{if } |v| \leq c \\ v + c & \text{if } v < -c. \end{cases}$$

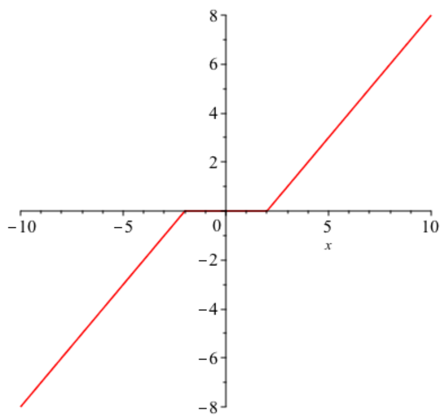


Fig. 16.6 The graph of S_c (when $c = 2$).

The function S_c is known as a *soft thresholding operator*. The graph of S_c shown in Figure 16.6.

One can check that

$$S_c(v) = (v - c)_+ - (-v - c)_+,$$

and also

$$S_c(v) = (1 - c/|v|)_+ v, \quad v \neq 0,$$

which shows that S_c is a *shrinkage operator* (it moves a point toward zero).

The operator S_c is extended to vectors in $\mathbb{R}^n$ component wise, that is, if $x = (x_1, \ldots, x_n)$, then

$$S_c(x) = (S_c(x_1), \ldots, S_c(x_n)).$$

We now consider several ℓ^1-norm problems.

(1) *Least absolute deviation.*
This is the problem of minimizing $\|Ax - b\|_1$, rather than $\|Ax - b\|_2$. Least absolute deviation is more robust than least squares fit because it deals better with outliers. The problem can be formulated in ADMM form as follows:

$$\begin{aligned} &\text{minimize} && \|z\|_1 \\ &\text{subject to} && Ax - z = b, \end{aligned}$$

with $f = 0$ and $g = \|\ \|_1$. As usual, we assume that A is an $m \times n$ matrix of rank n, so that $A^\top A$ is invertible. ADMM (in scaled form) can be expressed as

$$\begin{aligned} x^{k+1} &= (A^\top A)^{-1}A^\top(b + z^k - u^k) \\ z^{k+1} &= S_{1/\rho}(Ax^{k+1} - b + u^k) \\ u^{k+1} &= u^k + Ax^{k+1} - z^{k+1} - b. \end{aligned}$$

(2) *Basis pursuit.*
This is the following minimization problem:

$$\begin{aligned} &\text{minimize} \quad \|x\|_1 \\ &\text{subject to} \quad Ax = b, \end{aligned}$$

where A is an $m \times n$ matrix of rank $m < n$, and $b \in \mathbb{R}^m$, $x \in \mathbb{R}^n$. The problem is to find a sparse solution to an underdetermined linear system, which means a solution x with many zero coordinates. This problem plays a central role in compressed sensing and statistical signal processing.
Basis pursuit can be expressed in ADMM form as the problem

$$\begin{aligned} &\text{minimize} \quad I_C(x) + \|z\|_1 \\ &\text{subject to} \quad x - z = 0, \end{aligned}$$

with $C = \{x \in \mathbb{R}^n \mid Ax = b\}$. It is easy to see that the ADMM procedure (in scaled form) is

$$\begin{aligned} x^{k+1} &= \Pi_C(z^k - u^k) \\ z^{k+1} &= S_{1/\rho}(x^{k+1} + u^k) \\ u^{k+1} &= u^k + x^{k+1} - z^{k+1}, \end{aligned}$$

where Π_C is the orthogonal projection onto the subspace C. In fact, it is not hard to show that

$$x^{k+1} = (I - A^\top(AA^\top)^{-1}A)(z^k - u^k) + A^\top(AA^\top)^{-1}b.$$

In some sense, an ℓ^1-minimization problem is reduced to a sequence of ℓ^2-norm problems. There are ways of improving the efficiency of the method; see Boyd et al. [Boyd *et al.* (2010)] (Section 6.2)

(3) *General ℓ^1-regularized loss minimization.*
This is the following minimization problem:

$$\text{minimize} \quad l(x) + \tau\, \|x\|_1\,,$$

where l is any proper closed and convex loss function, and $\tau > 0$. We convert the problem to the ADMM problem:

$$\begin{aligned} &\text{minimize} \quad l(x) + \tau \left\|z\right\|_1 \\ &\text{subject to} \quad x - z = 0. \end{aligned}$$

The ADMM procedure (in scaled form) is

$$\begin{aligned} x^{k+1} &= \operatorname*{arg\,min}_x \Big(l(x) + (\rho/2) \left\|x - z^k + u^k\right\|_2^2 \Big) \\ z^{k+1} &= S_{\tau/\rho}(x^{k+1} + u^k) \\ u^{k+1} &= u^k + x^{k+1} - z^{k+1}. \end{aligned}$$

The x-update is a proximal operator evaluation. In general, one needs to apply a numerical procedure to compute x^{k+1}, for example, a version of Newton's method. The special case where $l(x) = (1/2) \left\|Ax - b\right\|_2^2$ is particularly important.

(4) *Lasso regularization.*
This is the following minimization problem:

$$\text{minimize} \quad (1/2) \left\|Ax - b\right\|_2^2 + \tau \left\|x\right\|_1 .$$

This is a linear regression with the regularizing term $\tau \left\|x\right\|_1$ instead of $\tau \left\|x\right\|_2$, to encourage a sparse solution. This method was first proposed by Tibshirani around 1996, under the name *lasso*, which stands for "least absolute selection and shrinkage operator." This method is also known as *ℓ^1-regularized regression*, but this is not as cute as "lasso," which is used predominantly. This method is discussed extensively in Hastie, Tibshirani, and Wainwright [Hastie *et al.* (2015)].
The lasso minimization is converted to the following problem in ADMM form:

$$\begin{aligned} &\text{minimize} \quad \left\|Ax - b\right\|_2^2 + \tau \left\|z\right\|_1 \\ &\text{subject to} \quad x - z = 0. \end{aligned}$$

Then the ADMM procedure (in scaled form) is

$$\begin{aligned} x^{k+1} &= (A^\top A + \rho I)^{-1}(A^\top b + \rho(z^k - u^k)) \\ z^{k+1} &= S_{\tau/\rho}(x^{k+1} + u^k) \\ u^{k+1} &= u^k + x^{k+1} - z^{k+1}. \end{aligned}$$

Since $\rho > 0$, the matrix $A^\top A + \rho I$ is symmetric positive definite. Note that the x-update looks like a *ridge regression step* (see Section 19.1). There are various generalizations of lasso.

(5) *Generalized Lasso regularization.*
This is the following minimization problem:

$$\text{minimize} \quad (1/2)\left\|Ax-b\right\|_2^2+\tau\left\|Fx\right\|_1,$$

where A is an $m\times n$ matrix, F is a $p\times n$ matrix, and either A has rank n or F has rank n. This problem is converted to the ADMM problem

$$\begin{aligned}&\text{minimize} && \left\|Ax-b\right\|_2^2+\tau\left\|z\right\|_1\\&\text{subject to} && Fx-z=0,\end{aligned}$$

and the corresponding ADMM procedure (in scaled form) is

$$\begin{aligned}x^{k+1}&=(A^\top A+\rho F^\top F)^{-1}(A^\top b+\rho F^\top(z^k-u^k))\\z^{k+1}&=S_{\tau/\rho}(Fx^{k+1}+u^k)\\u^{k+1}&=u^k+Fx^{k+1}-z^{k+1}.\end{aligned}$$

(6) *Group Lasso.*
This a generalization of (3). Here we assume that x is split as $x=(x_1,\ldots,x_N)$, with $x_i\in\mathbb{R}^{n_i}$ and $n_1+\cdots+x_N=n$, and the regularizing term $\left\|x\right\|_1$ is replaced by $\sum_{i=1}^N\left\|x_i\right\|_2$. When $n_i=1$, this reduces to (3). The z-update of the ADMM procedure needs to modified. We define the soft thresholding operator $\mathcal{S}_c\colon\mathbb{R}^m\to\mathbb{R}^m$ given by

$$\mathcal{S}_c(v)=\left(1-\frac{c}{\left\|v\right\|_2}\right)_+v,$$

with $\mathcal{S}_c(0)=0$. Then the z-update consists of the N updates

$$z_i^{k+1}=\mathcal{S}_{\tau/\rho}(x_i^{k+1}+u^k),\quad i=1,\ldots,N.$$

The method can be extended to deal with overlapping groups; see Boyd et al. [Boyd *et al.* (2010)] (Section 6.4).

There are many more applications of ADMM discussed in Boyd et al. [Boyd *et al.* (2010)], including consensus and sharing. See also Strang [Strang (2019)] for a brief overview.

16.9 Summary

The main concepts and results of this chapter are listed below:

- Dual ascent.
- Augmented Lagrangian.
- Penalty parameter.

- Method of multipliers.
- ADMM (alternating direction method of multipliers).
- x-update, z-update, λ-update.
- Scaled form of ADMM.
- Residual, dual residual.
- Stopping criteria.
- Proximity operator, proximal minimization.
- Quadratic programming.
- KKT equations.
- Soft thresholding operator.
- Shrinkage operator.
- Least absolute deviation.
- Basis pursuit.
- General ℓ^1-regularized loss minimization.
- Lasso regularization.
- Generalized lasso regularization.
- Group lasso.

16.10 Problems

Problem 16.1. In the method of multipliers described in Section 16.2, prove that choosing $\alpha^k = \rho$ guarantees that (u^{k+1}, λ^{k+1}) satisfies the equation

$$\nabla J_{u^{k+1}} + A^{\top} \lambda^{k+1} = 0.$$

Problem 16.2. Prove that the Inequality (A1) follows from the Inequalities (A2) and (A3) (see the proof of Theorem 16.1). For help consult Appendix A of Boyd et al. [Boyd *et al.* (2010)].

Problem 16.3. Consider Example 16.8. Prove that if $f = I_C$, the indicator function of a nonempty closed convex set C, then

$$x^+ = \arg\min_x \Big(I_C(x) + (\rho/2) \left\| x - v \right\|_2^2 \Big) = \Pi_C(v),$$

the orthogonal projection of v onto C. In the special case where $C = \mathbb{R}^n_+$ (the first orthant), then

$$x^+ = (v)_+,$$

the vector obtained by setting the negative components of v to zero.

Problem 16.4. Prove that the soft thresholding operator S_c from Section 16.8 satisfies the equations

$$S_c(v) = (v - c)_+ - (-v - c)_+,$$

and

$$S_c(v) = (1 - c/|v|)_+ v, \quad v \neq 0.$$

Problem 16.5. Rederive the formula

$$S_c(v) = \begin{cases} v - c & \text{if } v > c \\ 0 & \text{if } |v| \leq c \\ v + c & \text{if } v < -c \end{cases}$$

using subgradients.

Problem 16.6. In basis pursuit (see Section 16.8 (2)) prove that

$$x^{k+1} = (I - A^\top (AA^\top)^{-1} A)(z^k - u^k) + A^\top (AA^\top)^{-1} b.$$

Problem 16.7. Implement (in `Matlab`) ADMM applied to lasso regularization as described in Section 16.6 (4). The stopping criterion should be based on feasibility tolerances ϵ^{pri} and ϵ^{dual}, say 10^{-4}, and on a maximum number of iteration steps, say 10000. There is a build in `Matlab` function `wthresh` implementing soft thresholding. You may use the `Matlab` command `randn` to create a random data set X and a random response vector y (see the help menu in `Matlab` under `lasso`). Try various values of ρ and τ. You will observe that the choice of ρ greatly affects the rate of convergence of the procedure.

PART 4

Applications to Machine Learning

Chapter 17

Positive Definite Kernels

This chapter is an introduction to positive definite kernels and the use of kernel functions in machine learning.

Let X be a nonempty set. If the set X represents a set of highly nonlinear data, it may be advantageous to map X into a space F of much higher dimension called the *feature space*, using a function $\varphi\colon X \to F$ called a *feature map*. This idea is that φ "unwinds" the description of the objects in F in an attempt to make it linear. The space F is usually a vector space equipped with an inner product $\langle -, -\rangle$. If F is infinite dimensional, then we assume that it is a Hilbert space.

Many algorithms that analyze or classify data make use of the inner products $\langle\varphi(x), \varphi(y)\rangle$, where $x, y \in X$. These algorithms make use of the function $\kappa\colon X \times X \to \mathbb{C}$ given by

$$\kappa(x, y) = \langle\varphi(x), \varphi(y)\rangle, \qquad x, y \in X,$$

called a *kernel function*.

The kernel trick is to pretend that we have a feature embedding $\varphi\colon X \to F$ (actually unknown), but to only use inner products $\langle\varphi(x), \varphi(y)\rangle$ that can be evaluated using the original data through the known kernel function κ. It turns out that the functions of the form κ as above can be defined in terms of a condition which is reminiscent of positive semidefinite matrices (see Definition 17.2). Furthermore, every function satisfying Definition 17.2 arises from a suitable feature map into a Hilbert space; see Theorem 17.1.

We illustrate the kernel methods on kernel PCA (see Section 17.4).

17.1 Feature Maps and Kernel Functions

Definition 17.1. Let X be a nonempty set, let H be a (complex) Hilbert space, and let $\varphi\colon X \to H$ be a function called a *feature map*. The function

$\kappa\colon X \times X \to \mathbb{C}$ given by

$$\kappa(x, y) = \langle \varphi(x), \varphi(y) \rangle, \qquad x, y \in X,$$

is called a *kernel function*.

Remark: A *feature map* is often called a *feature embedding*, but this terminology is a bit misleading because it suggests that such a map is injective, which is not necessarily the case. Unfortunately this terminology is used by most people.

Example 17.1. Suppose we have two feature maps $\varphi_1\colon X \to \mathbb{R}^{n_1}$ and $\varphi_2\colon X \to \mathbb{R}^{n_2}$, and let $\kappa_1(x, y) = \langle \varphi_1(x), \varphi_1(y) \rangle$ and $\kappa_2(x, y) = \langle \varphi_2(x), \varphi_2(y) \rangle$ be the corresponding kernel functions (where $\langle -, - \rangle$ is the standard inner product on $\mathbb{R}^n$). Define the feature map $\varphi\colon X \to \mathbb{R}^{n_1+n_2}$ by

$$\varphi(x) = (\varphi_1(x), \varphi_2(x)),$$

an $(n_1 + n_2)$-tuple. We have

$$\begin{aligned}
\langle \varphi(x), \varphi(y) \rangle &= \langle (\varphi_1(x), \varphi_2(x)), (\varphi_1(y), \varphi_2(y)) \rangle \\
&= \langle \varphi_1(x), \varphi_1(y) \rangle + \langle \varphi_2(x), \varphi_2(y) \rangle \\
&= \kappa_1(x, y) + \kappa_2(x, y),
\end{aligned}$$

which shows that the map κ given by

$$\kappa(x, y) = \kappa_1(x, y) + \kappa_2(x, y)$$

is the kernel function corresponding to the feature map $\varphi\colon X \to \mathbb{R}^{n_1+n_2}$.

Example 17.2. Let X be a subset of $\mathbb{R}^2$, and let $\varphi_1\colon X \to \mathbb{R}^3$ be the map given by

$$\varphi_1(x_1, x_2) = (x_1^2, x_2^2, \sqrt{2}x_1x_2).$$

Figure 17.1 illustrates $\varphi_1\colon X \to \mathbb{R}^3$ when $X = \{((x_1, x_2) \mid -10 \leq x_1 \leq 10, -10 \leq x_2 \leq 10\}$.

Observe that linear relations in the feature space $H = \mathbb{R}^3$ correspond to quadratic relations in the input space (of data). We have

$$\begin{aligned}
\langle \varphi_1(x), \varphi_1(y) \rangle &= \langle (x_1^2, x_2^2, \sqrt{2}x_1x_2), (y_1^2, y_2^2, \sqrt{2}y_1y_2) \rangle \\
&= x_1^2y_1^2 + x_2^2y_2^2 + 2x_1x_2y_1y_2 \\
&= (x_1y_1 + x_2y_2)^2 = \langle x, y \rangle^2,
\end{aligned}$$

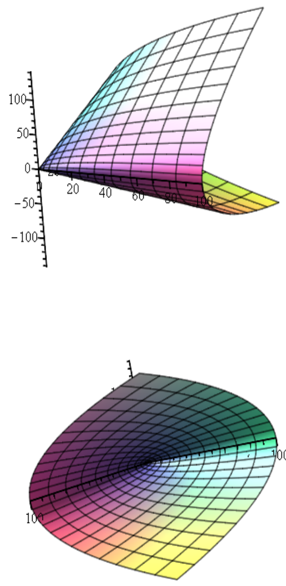


Fig. 17.1 The parametric surface $\varphi_1(x_1, x_2) = (x_1^2, x_2^2, \sqrt{2}x_1x_2)$ where $-10 \leq x_1 \leq 10$ and $-10 \leq x_2 \leq 10$.

where $\langle x, y\rangle$ is the usual inner product on $\mathbb{R}^2$. Hence the function

$$\kappa(x, y) = \langle x, y\rangle^2$$

is a kernel function associated with the feature space $\mathbb{R}^3$.

If we now consider the map $\varphi_2 \colon X \to \mathbb{R}^4$ given by

$$\varphi_2(x_1, x_2) = (x_1^2, x_2^2, x_1x_2, x_1x_2),$$

we check immediately that

$$\langle \varphi_2(x), \varphi_2(y)\rangle = \kappa(x, y) = \langle x, y\rangle^2,$$

which shows that the same kernel can arise from different maps into different feature spaces.

Example 17.3. Example 17.2 can be generalized as follows. Suppose we have a feature map $\varphi_1 \colon X \to \mathbb{R}^n$ and let $\kappa_1(x, y) = \langle \varphi_1(x), \varphi_1(y)\rangle$ be the corresponding kernel function (where $\langle -, -\rangle$ is the standard inner product on $\mathbb{R}^n$). Define the feature map $\varphi \colon X \to \mathbb{R}^n \times \mathbb{R}^n$ by its n^2 components

$$\varphi(x)_{(i,j)} = (\varphi_1(x))_i(\varphi_1(x))_j, \qquad 1 \leq i, j \leq n,$$

with the inner product on $\mathbb{R}^n \times \mathbb{R}^n$ given by

$$\langle u, v\rangle = \sum_{i,j=1}^{n} u_{(i,j)} v_{(i,j)}.$$

Then we have

$$\begin{aligned}\langle \varphi(x), \varphi(y)\rangle &= \sum_{i,j=1}^{n} \varphi_{(i,j)}(x)\varphi_{(i,j)}(y)\\ &= \sum_{i,j=1}^{n} (\varphi_1(x))_i(\varphi_1(x))_j(\varphi_1(y))_i(\varphi_1(y))_j\\ &= \sum_{i=1}^{n}(\varphi_1(x))_i(\varphi_1(y))_i \sum_{j=1}^{n}(\varphi_1(x))_j(\varphi_1(y))_j\\ &= (\kappa_1(x,y))^2.\end{aligned}$$

Thus the map κ given by $\kappa(x, y) = (\kappa_1(x, y))^2$ is a kernel map associated with the feature map $\varphi\colon X \to \mathbb{R}^n \times \mathbb{R}^n$. The feature map φ is a direct generalization of the feature map φ_2 of Example 17.2.

The above argument is immediately adapted to show that if $\varphi_1\colon X \to \mathbb{R}^{n_1}$ and $\varphi_2\colon X \to \mathbb{R}^{n_2}$ are two feature maps and if $\kappa_1(x, y) = \langle\varphi_1(x), \varphi_1(y)\rangle$ and $\kappa_2(x, y) = \langle\varphi_2(x), \varphi_2(y)\rangle$ are the corresponding kernel functions, then the map defined by

$$\kappa(x, y) = \kappa_1(x, y)\kappa_2(x, y)$$

is a kernel function for the feature space $\mathbb{R}^{n_1} \times \mathbb{R}^{n_2}$ and the feature map

$$\varphi(x)_{(i,j)} = (\varphi_1(x))_i(\varphi_2(x))_j, \qquad 1 \leq i \leq n_1,\ 1 \leq j \leq n_2.$$

Example 17.4. Note that the feature map $\varphi\colon X \to \mathbb{R}^n \times \mathbb{R}^n$ is not very economical because if $i \neq j$ then the components $\varphi_{(i,j)}(x)$ and $\varphi_{(j,i)}(x)$ are both equal to $(\varphi_1(x))_i(\varphi_1(x))_j$. Therefore we can define the more economical embedding $\varphi'\colon X \to \mathbb{R}^{\binom{n+1}{2}}$ given by

$$\varphi'(x)_{(i,j)} = \begin{cases}(\varphi_1(x))_i^2 & i = j,\\ \sqrt{2}(\varphi_1(x))_i(\varphi_1(x))_j & i < j,\end{cases}$$

where the pairs (i, j) with $1 \leq i \leq j \leq n$ are ordered lexicographically. The feature map φ is a direct generalization of the feature map φ_1 of Example 17.2.

Observe that φ' can also be defined in the following way which makes it easier to come up with the generalization to any power:

$$\varphi'_{(i_1,\ldots,i_n)}(x) = \binom{2}{i_1 \cdots i_n}^{1/2} (\varphi_1(x))_1^{i_1} (\varphi_1(x))_1^{i_2} \cdots (\varphi_1(x))_1^{i_n}, \qquad i_1 + i_2 + \cdots + i_n = 2,\ i_j \in \mathbb{N},$$

where the n-tuples $(i_1, \ldots, i_n)$ are ordered lexicographically. Recall that for any $m \geq 1$ and any $(i_1, \ldots, i_n) \in \mathbb{N}^m$ such that $i_1 + i_2 + \cdots + i_n = m$, we have

$$\binom{m}{i_1 \cdots i_n} = \frac{m!}{i_1! \cdots i_n!}.$$

More generally, for any $m \geq 2$, using the multinomial theorem, we can define a feature embedding $\varphi\colon X \to \mathbb{R}^{\binom{n+m-1}{m}}$ defining the kernel function κ given by $\kappa(x, y) = (\kappa_1(x, y))^m$, with φ given by

$$\varphi_{(i_1,\ldots,i_n)}(x) = \binom{m}{i_1 \cdots i_n}^{1/2} (\varphi_1(x))_1^{i_1} (\varphi_1(x))_1^{i_2} \cdots (\varphi_1(x))_1^{i_n}, \qquad i_1 + i_2 + \cdots + i_n = m,\ i_j \in \mathbb{N},$$

where the n-tuples $(i_1, \ldots, i_n)$ are ordered lexicographically.

Example 17.5. For any positive real constant $R > 0$, the constant function $\kappa(x, y) = R$ is a kernel function corresponding to the feature map $\varphi\colon X \to \mathbb{R}$ given by $\varphi(x, y) = \sqrt{R}$.

By definition, the function $\kappa'_1\colon \mathbb{R}^n \to \mathbb{R}$ given by $\kappa'_1(x, y) = \langle x, y\rangle$ is a kernel function (the feature map is the identity map from $\mathbb{R}^n$ to itself). We just saw that for any positive real constant $R > 0$, the constant $\kappa'_2(x, y) = R$ is a kernel function. By Example 17.1, the function $\kappa'_3(x, y) = \kappa'_1(x, y) + \kappa'_2(x, y)$ is a kernel function, and for any integer $d \geq 1$, by Example 17.4, the function κ_d given by

$$\kappa_d(x, y) = (\kappa'_3(x, y))^d = (\langle x, y\rangle + R)^d,$$

is a kernel function on $\mathbb{R}^n$. By the binomial formula,

$$\kappa_d(x, y) = \sum_{m=0}^{d} R^{d-m} \langle x, y\rangle^m.$$

By Example 17.1, the feature map of this kernel function is the concatenation of the features of the $d + 1$ kernel maps $R^{d-m}\langle x, y\rangle^m$. By Example

17.3, the components of the feature map of the kernel map $R^{d-m}\langle x, y\rangle^m$ are reweightings of the functions

$$\varphi_{(i_1,\ldots,i_n)}(x) = x_1^{i_1}x_2^{i_2}\cdots x_n^{i_n}, \quad i_1+i_2+\cdots+i_n = m,$$

with $(i_1,\ldots,i_n)\in\mathbb{N}^n$. Thus the components of the feature map of the kernel function κ_d are reweightings of the functions

$$\varphi_{(i_1,\ldots,i_n)}(x) = x_1^{i_1}x_2^{i_2}\cdots x_n^{i_n}, \quad i_1+i_2+\cdots+i_n \leq d,$$

with $(i_1,\ldots,i_n)\in\mathbb{N}^n$. It is easy to see that the dimension of this feature space is $\binom{m+d}{d}$.

There are a number of variations of the polynomial kernel κ_d; all-subsets embedding kernels, ANOVA kernels; see Shawe–Taylor and Christianini [Shawe-Taylor and Cristianini (2004)], Chapter III.

In the next example the set X is not a vector space.

Example 17.6. Let D be a finite set and let $X = 2^D$ be its power set. If $|D| = n$, let $H = \mathbb{R}^X \cong \mathbb{R}^{2^n}$. We are assuming that the subsets of D are enumerated in some fashion so that each coordinate of $\mathbb{R}^{2^n}$ corresponds to one of these subsets. For example, if $D = \{1, 2, 3, 4\}$, let

$$\begin{array}{llll}
U_1 = \emptyset & U_2 = \{1\} & U_3 = \{2\} & U_4 = \{3\} \\
U_5 = \{4\} & U_6 = \{1,2\} & U_7 = \{1,3\} & U_8 = \{1,4\} \\
U_9 = \{2,3\} & U_{10} = \{2,4\} & U_{11} = \{3,4\} & U_{12} = \{1,2,3\} \\
U_{13} = \{1,2,4\} & U_{14} = \{1,3,4\} & U_{15} = \{2,3,4\} & U_{16} = \{1,2,3,4\}.
\end{array}$$

Let $\varphi\colon X \to H$ be the feature map defined as follows: for any subsets $A, U \in X$,

$$\varphi(A)_U = \begin{cases} 1 & \text{if } U \subseteq A \\ 0 & \text{otherwise.} \end{cases}$$

For example, if $A_1 = \{1, 2, 3\}$, we obtain the vector

$$\varphi(\{1,2,3\}) = (1,1,1,1,0,1,1,0,1,0,0,1,0,0,0,0),$$

and if $A_2 = \{2, 3, 4\}$, we obtain the vector

$$\varphi(\{2,3,4\}) = (1,0,1,1,1,0,0,0,1,1,1,0,0,0,1,0).$$

For any two subsets A_1 and A_2 of D, it is easy to check that

$$\langle\varphi(A_1), \varphi(A_2)\rangle = 2^{|A_1\cap A_2|},$$

the number of common subsets of A_1 and A_2. For example, $A_1 \cap A_2 = \{2, 3\}$, and

$$\langle \varphi(A_1), \varphi(A_2) \rangle = 4.$$

Therefore, the function $\kappa\colon X \times X \to \mathbb{R}$ given by

$$\kappa(A_1, A_2) = 2^{|A_1 \cap A_2|}, \qquad A_1, A_2 \subseteq D$$

is a kernel function.

Kernels on collections of sets can be defined in terms of measures.

Example 17.7. Let $(D, \mathcal{A})$ be a measurable space, where D is a nonempty set and $\mathcal{A}$ is a σ-algebra on D (the measurable sets). Let X be a subset of $\mathcal{A}$. If μ is a positive measure on $(D, \mathcal{A})$ and if μ is finite, which means that $\mu(D)$ is finite, then we can define the map $\kappa_1\colon X \times X \to \mathbb{R}$ given by

$$\kappa_1(A_1, A_2) = \mu(A_1 \cap A_2), \qquad A_1, A_2 \in X.$$

We can show that κ is a kernel function as follows. Let $H = \mathrm{L}^2_\mu(D, \mathcal{A}, \mathbb{R})$ be the Hilbert space of μ-square-integrable functions with the inner product

$$\langle f, g \rangle = \int_D f(s)g(s)\, d\mu(s),$$

and let $\varphi\colon X \to H$ be the feature embedding given by

$$\varphi(A) = \chi_A, \qquad A \in X,$$

the characteristic function of A. Then we have

$$\begin{aligned}
\kappa_1(A_1, A_2) = \mu(A_1 \cap A_2) &= \int_D \chi_{A_1 \cap A_2}(s)\, d\mu(s) \\
&= \int_D \chi_{A_1}(s)\chi_{A_2}(s)\, d\mu(s) = \langle \chi_{A_1}, \chi_{A_2} \rangle \\
&= \langle \varphi(A_1), \varphi(A_2) \rangle.
\end{aligned}$$

The above kernel is called the *intersection kernel*. If we assume that μ is normalized so that $\mu(D) = 1$, then we also have the *union complement kernel*:

$$\kappa_2(A_1, A_2) = \mu(\overline{A_1} \cap \overline{A_2}) = 1 - \mu(A_1 \cup A_2).$$

The sum κ_3 of the kernels κ_1 and κ_2 is the *agreement kernel*:

$$\kappa_s(A_1, A_2) = 1 - \mu(A_1 - A_2) - \mu(A_2 - A_1).$$

Many other kinds of kernels can be designed, in particular, graph kernels. For comprehensive presentations of kernels, see Schölkopf and Smola [Schölkopf and Smola (2002)] and Shawe–Taylor and Christianini [Shawe-Taylor and Cristianini (2004)].

Kernel functions have the following important property.

Proposition 17.1. *Let X be any nonempty set, let H be any (complex) Hilbert space, let $\varphi\colon X \to H$ be any function, and let $\kappa\colon X \times X \to \mathbb{C}$ be the kernel given by*

$$\kappa(x, y) = \langle \varphi(x), \varphi(y)\rangle, \qquad x, y \in X.$$

For any finite subset $S = \{x_1, \ldots, x_p\}$ of X, if K_S is the $p \times p$ matrix

$$K_S = (\kappa(x_j, x_i))_{1\leq i,j\leq p} = (\langle \varphi(x_j), \varphi(x_i)\rangle)_{1\leq i,j\leq p},$$

then we have

$$u^* K_S\, u \geq 0, \qquad \textit{for all } u \in \mathbb{C}^p.$$

Proof. We have

$$\begin{aligned} u^* K_S\, u = u^\top K_S^\top\, \overline{u} &= \sum_{i,j=1}^{p} \kappa(x_i, x_j) u_i \overline{u_j} \\ &= \sum_{i,j=1}^{p} \langle \varphi(x), \varphi(y)\rangle u_i \overline{u_j} \\ &= \left\langle \sum_{i=1}^{p} u_i \varphi(x_i), \sum_{j=1}^{p} u_j \varphi(x_j) \right\rangle = \left\| \sum_{i=1}^{p} u_i \varphi(x_i) \right\|^2 \geq 0, \end{aligned}$$

as claimed. ☐

17.2 Basic Properties of Positive Definite Kernels

Proposition 17.1 suggests a second approach to kernel functions which does not assume that a feature space and a feature map are provided. We will see in Section 17.3 that the two approaches are equivalent. The second approach is useful in practice because it is often difficult to define a feature space and a feature map in a simple manner.

Definition 17.2. Let X be a nonempty set. A function $\kappa\colon X \times X \to \mathbb{C}$ is a *positive definite kernel* if for every finite subset $S = \{x_1, \ldots, x_p\}$ of X, if K_S is the $p \times p$ matrix

$$K_S = (\kappa(x_j, x_i))_{1\leq i,j\leq p}$$

called a *Gram matrix*, then we have

$$u^* K_S\, u = \sum_{i,j=1}^{p} \kappa(x_i, x_j) u_i \overline{u_j} \geq 0, \qquad \text{for all } u \in \mathbb{C}^p.$$

Observe that Definition 17.2 does not require that $u^* K_S\, u > 0$ if $u \neq 0$, so the terminology *positive definite* is a bit abusive, and it would be more appropriate to use the terminology *positive semidefinite*. However, it seems customary to use the term *positive definite kernel*, or even *positive kernel*.

Proposition 17.2. *Let $\kappa\colon X \times X \to \mathbb{C}$ be a positive definite kernel. Then $\kappa(x,x) \geq 0$ for all $x \in X$, and for any finite subset $S = \{x_1, \ldots, x_p\}$ of X, the $p \times p$ matrix K_S given by*

$$K_S = (\kappa(x_j, x_i))_{1 \leq i,j \leq p}$$

is Hermitian, that is, $K_S^* = K_S$.

Proof. The first property is obvious by choosing $S = \{x\}$. To prove that K_S is Hermitian, observe that we have

$$(u+v)^* K_S(u+v) = u^* K_S u + u^* K_S v + v^* K_S u + v^* K_S v,$$

and since $(u+v)^* K_S(u+v), u^* K_S u, v^* K_S v \geq 0$, we deduce that

$$2A = u^* K_S v + v^* K_S u \tag{1}$$

must be real. By replacing u by iu, we see that

$$2B = -iu^* K_S v + iv^* K_S u \tag{2}$$

must also be real. By multiplying Equation (2) by i and adding it to Equation (1) we get

$$u^* K_S v = A + iB. \tag{3}$$

By subtracting Equation (3) from Equation (1) we get

$$v^* K_S u = A - iB.$$

Then

$$u^* K_S^* v = \overline{v^* K_S u} = \overline{A - iB} = A + iB = u^* K_S v,$$

for all $u, v \in \mathbb{C}^*$, which implies $K_S^* = K_S$. $\square$

If the map $\kappa\colon X \times X \to \mathbb{R}$ is real-valued, then we have the following criterion for κ to be a positive definite kernel that only involves real vectors.

Proposition 17.3. *If $\kappa\colon X \times X \to \mathbb{R}$, then κ is a positive definite kernel iff for any finite subset $S = \{x_1, \ldots, x_p\}$ of X, the $p \times p$ real matrix K_S given by*

$$K_S = (\kappa(x_k, x_j))_{1 \leq j,k \leq p}$$

is symmetric, that is, $K_S^\top = K_S$, and

$$u^\top K_S\, u = \sum_{j,k=1}^{p} \kappa(x_j, x_k) u_j u_k \geq 0, \qquad \textit{for all } u \in \mathbb{R}^p.$$

Proof. If κ is a real-valued positive definite kernel, then the proposition is a trivial consequence of Proposition 17.2.

For the converse assume that κ is symmetric and that it satisfies the second condition of the proposition. We need to show that κ is a positive definite kernel with respect to complex vectors. If we write $u_k = a_k + ib_k$, then

$$\begin{aligned} u^* K_S\, u &= \sum_{j,k=1}^{p} \kappa(x_j, x_k)(a_j + ib_j)(a_k - ib_k) \\ &= \sum_{j,k=1}^{p} (a_j a_k + b_j b_k)\kappa(x_j, x_k) + i \sum_{j,k=1}^{p} (b_j a_k - a_j b_k)\kappa(x_j, x_k) \\ &= \sum_{j,k=1}^{p} (a_j a_k + b_j b_k)\kappa(x_j, x_k) \\ &\quad + i \sum_{1 \le j < k \le p} b_j a_k(\kappa(x_j, x_k) - \kappa(x_k, x_j)). \end{aligned}$$

Thus $u^* K_S u$ is real iff K_S is symmetric. ☐

Consequently we make the following definition.

Definition 17.3. Let X be a nonempty set. A function $\kappa \colon X \times X \to \mathbb{R}$ is a *(real) positive definite kernel* if $\kappa(x, y) = \kappa(y, x)$ for all $x, y \in X$, and for every finite subset $S = \{x_1, \ldots, x_p\}$ of X, if K_S is the $p \times p$ real symmetric matrix

$$K_S = (\kappa(x_i, x_j))_{1 \le i,j \le p},$$

then we have

$$u^\top K_S\, u = \sum_{i,j=1}^{p} \kappa(x_i, x_j) u_i u_j \ge 0, \qquad \text{for all } u \in \mathbb{R}^p.$$

Among other things, the next proposition shows that a positive definite kernel satisfies the Cauchy–Schwarz inequality.

Proposition 17.4. *A Hermitian* 2×2 *matrix*

$$A = \begin{pmatrix} a & \overline{b} \\ b & d \end{pmatrix}$$

is positive semidefinite if and only if $a \ge 0$, $d \ge 0$, *and* $ad - |b|^2 \ge 0$.

Let $\kappa \colon X \times X \to \mathbb{C}$ *be a positive definite kernel. For all* $x, y \in X$, *we have*

$$|\kappa(x, y)|^2 \le \kappa(x, x)\kappa(y, y).$$

Proof. For all $x, y \in \mathbb{C}$, we have

$$\begin{aligned}(\overline{x}\ \overline{y}) \begin{pmatrix} a & \overline{b} \\ b & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} &= (\overline{x}\ \overline{y}) \begin{pmatrix} ax + \overline{b}y \\ bx + dy \end{pmatrix} \\ &= a|x|^2 + bx\overline{y} + \overline{bx\overline{y}} + d|y|^2.\end{aligned}$$

If A is positive semidefinite, then we already know that $a \geq 0$ and $d \geq 0$. If $a = 0$, then we must have $b = 0$, since otherwise we can make $bx\overline{y} + \overline{bx\overline{y}}$, which is twice the real part of $bx\overline{y}$, as negative as we want. In this case, $ad - |b|^2 = 0$.

If $a > 0$, then

$$a|x|^2 + bx\overline{y} + \overline{bx\overline{y}} + d|y|^2 = a\left|x + \frac{\overline{b}}{a}y\right|^2 + \frac{|y|^2}{a}(ad - |b|^2).$$

If $ad - |b|^2 < 0$, we can pick $y \neq 0$ and $x = -(\overline{b}y)/a$, so that the above expression is negative. Therefore, $ad - |b|^2 \geq 0$. The converse is trivial.

If $x = y$, the inequality $|\kappa(x, y)|^2 \leq \kappa(x, x)\kappa(y, y)$ is trivial. If $x \neq y$, the inequality follows by applying the criterion for being positive semidefinite to the matrix

$$\begin{pmatrix} \kappa(x, x) & \overline{\kappa(x, y)} \\ \kappa(x, y) & \kappa(y, y) \end{pmatrix},$$

as claimed. $\square$

The following property due to I. Schur (1911) shows that the pointwise product of two positive definite kernels is also a positive definite kernel.

Proposition 17.5. *(I. Schur) If $\kappa_1 \colon X \times X \to \mathbb{C}$ and $\kappa_2 \colon X \times X \to \mathbb{C}$ are two positive definite kernels, then the function $\kappa \colon X \times X \to \mathbb{C}$ given by $\kappa(x, y) = \kappa_1(x, y)\kappa_2(x, y)$ for all $x, y \in X$ is also a positive definite kernel.*

Proof. It suffices to prove that if $A = (a_{jk})$ and $B = (b_{jk})$ are two Hermitian positive semidefinite $p \times p$ matrices, then so is their pointwise product $C = A \circ B = (a_{jk}b_{jk})$ (also known as Hadamard or Schur product). Recall that a Hermitian positive semidefinite matrix A can be diagonalized as $A = U\Lambda U^*$, where Λ is a diagonal matrix with nonnegative entries and U is a unitary matrix. Let $\Lambda^{1/2}$ be the diagonal matrix consisting of the positive square roots of the diagonal entries in Λ. Then we have

$$A = U\Lambda U^* = U\Lambda^{1/2}\Lambda^{1/2}U^* = U\Lambda^{1/2}(U\Lambda^{1/2})^*.$$

Thus if we set $R = U\Lambda^{1/2}$, we have

$$A = RR^*,$$

which means that

$$a_{jk} = \sum_{h=1}^{p} r_{jh}\overline{r_{kh}}.$$

Then for any $u \in \mathbb{C}^p$, we have

$$\begin{aligned}
u^*(A \circ B)u &= \sum_{j,k=1}^{p} a_{jk}b_{jk}u_j\overline{u_k} \\
&= \sum_{j,k=1}^{p}\sum_{h=1}^{p} r_{jh}\overline{r_{kh}}b_{jk}u_j\overline{u_k} \\
&= \sum_{h=1}^{p}\sum_{j,k=1}^{p} b_{jk}u_jr_{jh}\overline{u_kr_{kh}}.
\end{aligned}$$

Since B is positive semidefinite, for each fixed h, we have

$$\sum_{j,k=1}^{p} b_{jk}u_jr_{jh}\overline{u_kr_{kh}} = \sum_{j,k=1}^{p} b_{jk}z_j\overline{z_k} \geq 0,$$

as we see by letting $z = (u_1r_{1h}, \ldots, u_pr_{ph})$, $\square$

In contrast, the ordinary product AB of two symmetric positive semidefinite matrices A and B may not be symmetric positive semidefinite; see Section 7.9 in Volume I for an example.

Here are other ways of obtaining new positive definite kernels from old ones.

Proposition 17.6. *Let $\kappa_1 \colon X \times X \to \mathbb{C}$ and $\kappa_2 \colon X \times X \to \mathbb{C}$ be two positive definite kernels, $f \colon X \to \mathbb{C}$ be a function, $\psi \colon X \to \mathbb{R}^N$ be a function, $\kappa_3 \colon \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{C}$ be a positive definite kernel, and $a \in \mathbb{R}$ be any positive real number. Then the following functions are positive definite kernels:*

(1) $\kappa(x,y) = \kappa_1(x,y) + \kappa_2(x,y)$.
(2) $\kappa(x,y) = a\kappa_1(x,y)$.
(3) $\kappa(x,y) = f(x)\overline{f(y)}$.
(4) $\kappa(x,y) = \kappa_3(\psi(x), \psi(y))$.
(5) *If B is a symmetric positive semidefinite $n \times n$ matrix, then the map $\kappa \colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ given by*

$$\kappa(x,y) = x^\top By$$

is a positive definite kernel.

Proof. (1) For every finite subset $S = \{x_1, \ldots, x_p\}$ of X, if K_1 is the $p \times p$ matrix

$$K_1 = (\kappa_1(x_k, x_j))_{1 \leq j,k \leq p}$$

and if if K_2 is the $p \times p$ matrix

$$K_2 = (\kappa_2(x_k, x_j))_{1 \leq j,k \leq p},$$

then for any $u \in \mathbb{C}^p$, we have

$$u^*(K_1 + K_2)u = u^* K_1 u + u^* K_2 u \geq 0,$$

since $u^* K_1 u \geq 0$ and $u^* K_2 u \geq 0$ because κ_2 and κ_2 are positive definite kernels, which means that K_1 and K_2 are positive semidefinite.

(2) We have

$$u^*(aK_1)u = au^* K_1 u \geq 0,$$

since $a > 0$ and $u^* K_1 u \geq 0$.

(3) For every finite subset $S = \{x_1, \ldots, x_p\}$ of X, if K is the $p \times p$ matrix

$$K = (\kappa(x_k, x_j))_{1 \leq j,k \leq p} = (\overline{f(x_k)} f(x_j))_{1 \leq j,k \leq p}$$

then we have

$$\begin{aligned} u^* K u = u^\top K^\top \overline{u} = \sum_{j,k=1}^{p} \kappa(x_j, x_k) u_j \overline{u_k} &= \sum_{j,k=1}^{p} u_j f(x_j) \overline{u_k f(x_k)} \\ &= \left| \sum_{j=1}^{p} u_j f(x_j) \right|^2 \geq 0. \end{aligned}$$

(4) For every finite subset $S = \{x_1, \ldots, x_p\}$ of X, the $p \times p$ matrix K given by

$$K = (\kappa(x_k, x_j))_{1 \leq j,k \leq p} = (\kappa_3(\psi(x_k), \psi(x_j)))_{1 \leq j,k \leq p}$$

is symmetric positive semidefinite since κ_3 is a positive definite kernel.

(5) As in the proof of Proposition 17.5 (adapted to the real case) there is a matrix R such that

$$B = RR^\top,$$

so

$$\kappa(x, y) = x^\top B y = x^\top R R^\top y = (R^\top x)^\top R^\top y = \langle R^\top x, R^\top y \rangle,$$

so κ is the kernel function given by the feature map $\varphi(x) = R^\top x$ from $\mathbb{R}^n$ to itself, and by Proposition 17.1, it is a symmetric positive definite kernel. $\square$

Proposition 17.7. *Let $\kappa_1 \colon X \times X \to \mathbb{C}$ be a positive definite kernel, and let $p(z)$ be a polynomial with nonnegative coefficients. Then the following functions κ defined below are also positive definite kernels.*

(1) $\kappa(x, y) = p(\kappa_1(x, y))$.
(2) $\kappa(x, y) = e^{\kappa_1(x,y)}$.
(3) If X is real Hilbert space with inner product $\langle -, - \rangle_X$ and corresponding norm $\| \, \|_X$,

$$\kappa(x, y) = e^{-\frac{\|x-y\|_X^2}{2\sigma^2}}$$

for any $\sigma > 0$.

Proof. (1) If $p(z) = a_m z^m + \cdots + a_1 z + a_0$, then

$$p(\kappa_1(x, y)) = a_m \kappa_1(x, y)^m + \cdots + a_1 \kappa_1(x, y) + a_0.$$

Since $a_k \geq 0$ for $k = 0, \ldots, m$, by Proposition 17.5 and Proposition 17.6(2), each function $a_k \kappa_i(x, y)^k$ with $1 \leq k \leq m$ is a positive definite kernel, by Proposition 17.6(3) with $f(x) = \sqrt{a_0}$, the constant function a_0 is a positive definite kernel, and by Proposition 17.6(1), $p(\kappa_1(x, y))$ is a positive definite kernel.

(2) We have

$$e^{\kappa_1(x,y)} = \sum_{k=0}^{\infty} \frac{\kappa_1(x, y)^k}{k!}.$$

By (1), the partial sums

$$\sum_{k=0}^{m} \frac{\kappa_1(x, y)^k}{k!}$$

are positive definite kernels, and since $e^{\kappa_1(x,y)}$ is the (uniform) pointwise limit of positive definite kernels, it is also a positive definite kernel.

(3) By Proposition 17.6(2), since the map $(x, y) \mapsto \langle x, y \rangle_X$ is obviously a positive definite kernel (the feature map is the identity) and since $\sigma \neq 0$, the function $(x, y) \mapsto \langle x, y \rangle_X / \sigma^2$ is a positive definite kernel (by Proposition 17.6(2)), so by (2),

$$\kappa_1(x, y) = e^{\frac{\langle x,y \rangle_X}{\sigma^2}}$$

is a positive definite kernel. Let $f \colon X \to \mathbb{R}$ be the function given by

$$f(x) = e^{-\frac{\|x\|^2}{2\sigma^2}}.$$

Then by Proposition 17.6(3),

$$\kappa_2(x,y) = f(x)f(y) = e^{-\frac{\|x\|^2}{2\sigma^2}}e^{-\frac{\|y\|^2}{2\sigma^2}} = e^{-\frac{\|x\|_X^2+\|y\|_X^2}{2\sigma^2}}$$

is a positive definite kernel. By Proposition 17.5, the function $\kappa_1\kappa_2$ is a positive definite kernel, that is

$$\kappa_1(x,y)\kappa_2(x,y) = e^{\frac{\langle x,y\rangle_X}{\sigma^2}}e^{-\frac{\|x\|_X^2+\|y\|_X^2}{2\sigma^2}} = e^{\frac{\langle x,y\rangle_X}{\sigma^2}-\frac{\|x\|_X^2+\|y\|_X^2}{2\sigma^2}} = e^{-\frac{\|x-y\|_X^2}{2\sigma^2}}$$

is a positive definite kernel. ☐

Definition 17.4. The positive definite kernel

$$\kappa(x,y) = e^{-\frac{\|x-y\|_X^2}{2\sigma^2}}$$

is called a *Gaussian kernel*.

This kernel requires a feature map in an infinite-dimensional space because it is an infinite sum of distinct kernels.

Remark: If κ_1 is a positive definite kernel, the proof of Proposition 17.7(3) is immediately adapted to show that

$$\kappa(x,y) = e^{-\frac{\kappa_1(x,x)+\kappa_1(y,y)-2\kappa_1(x,y)}{2\sigma^2}}$$

is a positive definite kernel.

Next we prove that every positive definite kernel arises from a feature map in a Hilbert space which is a function space.

17.3 Hilbert Space Representation of a Positive Definite Kernel ⊛

The following result shows how to construct a so-called *reproducing kernel Hilbert space*, for short RKHS, from a positive definite kernel.

Theorem 17.1. *Let $\kappa\colon X\times X\to\mathbb{C}$ be a positive definite kernel on a nonempty set X. For every $x\in X$, let $\kappa_x\colon X\to\mathbb{C}$ be the function given by*

$$\kappa_x(y) = \kappa(x,y), \qquad y\in X.$$

Let H_0 be the subspace of the vector space $\mathbb{C}^X$ of functions from X to $\mathbb{C}$ spanned by the family of functions $(\kappa_x)_{\in X}$, and let $\varphi\colon X\to H_0$ be the map given by $\varphi(x)=\kappa_x$. There is a Hermitian inner product $\langle -,-\rangle$ on H_0 such that

$$\kappa(x,y) = \langle\varphi(x),\varphi(y)\rangle, \qquad \textit{for all } x,y\in X.$$

The completion H of H_0 is a Hilbert space, and the map $\eta\colon H \to \mathbb{C}^X$ given by

$$\eta(f)(x) = \langle f, \kappa_x \rangle, \qquad x \in X,$$

is linear and injective, so H can be identified with a subspace of $\mathbb{C}^X$. We also have

$$\kappa(x, y) = \langle \varphi(x), \varphi(y) \rangle, \qquad \textit{for all } x, y \in X.$$

For all $f \in H_0$ and all $x \in X$,

$$\langle f, \kappa_x \rangle = f(x), \tag{$*$}$$

a property known as the **reproducing property**.

Proof.

Step 1. Define a candidate inner product.

For any two linear combinations $f = \sum_{j=1}^{p} \alpha_j \kappa_{x_j}$ and $g = \sum_{k=1}^{q} \beta_k \kappa_{y_k}$ in H_0, with $x_j, y_k \in X$ and $\alpha_j, \beta_k \in \mathbb{C}$, define $\langle f, g \rangle$ by

$$\langle f, g \rangle = \sum_{j=1}^{p} \sum_{k=1}^{q} \alpha_j \overline{\beta_k} \kappa(x_j, y_k). \tag{$\dagger$}$$

At first glance, the above expression appears to depend on the expression of f and g as linear combinations, but since $\kappa(x_j, y_k) = \overline{\kappa(y_k, x_j)}$, observe that

$$\sum_{k=1}^{q} \overline{\beta_k} f(y_k) = \sum_{j=1}^{p} \sum_{k=1}^{q} \alpha_j \overline{\beta_k} \kappa(x_j, y_k) = \sum_{j=1}^{p} \alpha_j \overline{g(x_j)}, \tag{$*$}$$

and since the first and the third term are equal for all linear combinations representing f and g, we conclude that ($\dagger$) depends only on f and g and not on their representation as a linear combination.

Step 2. Prove that the inner product defined by ($\dagger$) is Hermitian semidefinite.

Obviously ($\dagger$) defines a Hermitian sequilinear form. For every $f \in H_0$, we have

$$\langle f, f \rangle = \sum_{j,k=1}^{p} \alpha_j \overline{\alpha_k} \kappa(x_j, x_k) \geq 0,$$

since κ is a positive definite kernel.

Step 3. Prove that the inner product defined by ($\dagger$) is positive definite.

For any finite subset $\{f_1, \ldots, f_n\}$ of H_0 and any $z \in \mathbb{C}^n$, we have

$$\sum_{j,k=1}^{n} \langle f_j, f_k\rangle z_j \overline{z_k} = \left\langle \sum_{j=1}^{n} z_j f_j, \sum_{j=1}^{n} z_j f_j \right\rangle \geq 0,$$

which shows that the map $(f, g) \mapsto \langle f, g\rangle$ from $H_0 \times H_0$ to $\mathbb{C}$ is a positive definite kernel.

Observe that for all $f \in H_0$ and all $x \in X$, (†) implies that

$$\langle f, \kappa_x\rangle = \sum_{j=1}^{k} \alpha_j \kappa(x_j, x) = f(x),$$

a property known as the *reproducing property*. The above implies that

$$\langle \kappa_x, \kappa_y\rangle = \kappa(x, y). \tag{$**$}$$

By Proposition 17.4 applied to the positive definite kernel $(f, g) \mapsto \langle f, g\rangle$, we have

$$|\langle f, \kappa_x\rangle|^2 \leq \langle f, f\rangle\langle \kappa_x, \kappa_x\rangle,$$

that is,

$$|f(x)|^2 \leq \langle f, f\rangle \kappa(x, x),$$

so $\langle f, f\rangle = 0$ implies that $f(x) = 0$ for all $x \in X$, which means that $\langle -, -\rangle$ as defined by (†) is positive definite. Therefore, $\langle -, -\rangle$ is a Hermitian inner product on H_0, and by $(**)$ and since $\varphi(x) = \kappa_x$, we have

$$\kappa(x, y) = \langle \varphi(x), \varphi(y)\rangle, \qquad \text{for all } x, y \in X.$$

Step 4. Define the embedding η.

Let H be the Hilbert space which is the completion of H_0, so that H_0 is dense in H. The map $\eta\colon H \to \mathbb{C}^X$ given by

$$\eta(f)(x) = \langle f, \kappa_x\rangle$$

is obviously linear, and it is injective because the family $(\kappa_x)_{x\in X}$ spans H_0 which is dense in H, thus it is also dense in H, so if $\langle f, \kappa_x\rangle = 0$ for all $x \in X$, then $f = 0$. ☐

Corollary 17.1. *If we identify a function $f \in H$ with the function $\eta(f)$, then we have the reproducing property*

$$\langle f, \kappa_x\rangle = f(x), \qquad \textit{for all } f \in H \textit{ and all } x \in X.$$

If X is finite, then $\mathbb{C}^X$ is finite-dimensional. If X is a separable topological space and if κ is continuous, then it can be shown that H is a separable Hilbert space.

Also, if $\kappa\colon X \times X \to \mathbb{R}$ is a real symmetric positive definite kernel, then we see immediately that Theorem 17.1 holds with H_0 a real Euclidean space and H a real Hilbert space.

$\circledast$ **Remark:** If $X = G$, where G is a locally compact group, then a function $p\colon G \to \mathbb{C}$ (not necessarily continuous) is *positive semidefinite* if for all $s_1, \ldots, s_n \in G$ and all $\xi_1, \ldots, \xi_n \in \mathbb{C}$, we have

$$\sum_{j,k=1}^{n} p(s_j^{-1}s_k)\xi_k\overline{\xi_j} \geq 0.$$

So if we define $\kappa\colon G \times G \to \mathbb{C}$ by

$$\kappa(s,t) = p(t^{-1}s),$$

then κ is a positive definite kernel on G. If p is continuous, then it is known that p arises from a unitary representation $U\colon G \to \mathbf{U}(H)$ of the group G in a Hilbert space H with inner product $\langle -, - \rangle$ (a homomorphism with a certain continuity property), in the sense that there is some vector $x_0 \in H$ such that

$$p(s) = \langle U(s)(x_0), x_0 \rangle, \qquad \text{for all } s \in G.$$

Since the $U(s)$ are unitary operators on H,

$$\begin{aligned} p(t^{-1}s) &= \langle U(t^{-1}s)(x_0), x_0 \rangle = \langle U(t^{-1})(U(s)(x_0)), x_0 \rangle \\ &= \langle U(t)^*(U(s)(x_0)), x_0 \rangle = \langle U(s)(x_0), U(t)(x_0) \rangle, \end{aligned}$$

which shows that

$$\kappa(s,t) = \langle U(s)(x_0), U(t)(x_0) \rangle,$$

so the map $\varphi\colon G \to H$ given by

$$\varphi(s) = U(s)(x_0)$$

is a feature map into the feature space H. This theorem is due to Gelfand and Raikov (1943).

The proof of Theorem 17.1 is essentially identical to part of Godement's proof of the above result about the correspondence between functions of positive type and unitary representations; see Helgason [Helgason (2000)], Chapter IV, Theorem 1.5. Theorem 17.1 is a little more general since it does not assume that X is a group, but when G is a group, the feature map arises from a unitary representation.

17.4 Kernel PCA

As an application of kernel functions, we discuss a generalization of the method of principal component analysis (PCA). Suppose we have a set of data $S = \{x_1, \ldots, x_n\}$ in some input space $\mathcal{X}$, and pretend that we have an embedding $\varphi\colon \mathcal{X} \to F$ of $\mathcal{X}$ in a (real) feature space $(F, \langle -, - \rangle)$, but that we only have access to the kernel function $\kappa(x, y) = \langle \varphi(x), \varphi(y) \rangle$. We would like to do PCA analysis on the set $\varphi(S) = \{\varphi(x_1), \ldots, \varphi(x_n)\}$.

There are two obstacles:

(1) We need to center the data and compute the inner products of pairs of centered data. More precisely, if the centroid of $\varphi(S)$ is
$$\mu = \frac{1}{n}(\varphi(x_1) + \cdots + \varphi(x_n)),$$
then we need to compute the inner products $\langle \varphi(x) - \mu, \varphi(y) - \mu \rangle$.

(2) Let us assume that $F = \mathbb{R}^d$ with the standard Euclidean inner product and that the data points $\varphi(x_i)$ are expressed as *row vectors* X_i of an $n \times d$ matrix X (as it is customary). Then the inner products $\kappa(x_i, x_j) = \langle \varphi(x_i), \varphi(x_j) \rangle$ are given by the *kernel matrix* $\mathbf{K} = XX^\top$. Be aware that with this representation, in the expression $\langle \varphi(x_i), \varphi(x_j) \rangle$, $\varphi(x_i)$ is a d-dimensional column vector, while $\varphi(x_i) = X_i^\top$. However, the jth component $(Y_k)_j$ of the principal component Y_k (viewed as a n-dimensional column vector) is given by the projection of $\widehat{X}_j = X_j - \mu$ onto the direction u_k (viewing μ as a d-dimensional row vector), which is a unit eigenvector of the matrix $(X - \mu)^\top (X - \mu)$ (where $\widehat{X} = X - \mu$ is the matrix whose jth row is $\widehat{X}_j = X_j - \mu$), is given by the inner product
$$\langle X_j - \mu, u_k \rangle = (Y_k)_j;$$
see Definition 21.2 (Vol. I) and Theorem 21.11 (Vol. I). The problem is that we know what the matrix $(X - \mu)(X - \mu)^\top$ is from (1), because it can be expressed in terms of $\mathbf{K}$, but we don't know what $(X - \mu)^\top (X - \mu)$ is because we don't have access to $\widehat{X} = X - \mu$.

Both difficulties are easily overcome. For (1) we have
$$\begin{aligned}\langle \varphi(x) - \mu, \varphi(y) - \mu \rangle &= \left\langle \varphi(x) - \frac{1}{n}\sum_{k=1}^n \varphi(x_k), \varphi(y) - \frac{1}{n}\sum_{k=1}^n \varphi(x_k) \right\rangle \\ &= \kappa(x, y) - \frac{1}{n}\sum_{i=1}^n \kappa(x, x_i) - \frac{1}{n}\sum_{j=1}^n \kappa(x_j, y) + \frac{1}{n^2}\sum_{i,j=1}^n \kappa(x_i, x_j).\end{aligned}$$

For (2), if $\mathbf{K}$ is the kernel matrix $\mathbf{K} = (\kappa(x_i, x_j))$, then the kernel matrix $\widehat{\mathbf{K}}$ corresponding to the kernel function $\widehat{\kappa}$ given by

$$\widehat{\kappa}(x, y) = \langle \varphi(x) - \mu, \varphi(y) - \mu \rangle$$

can be expressed in terms of $\mathbf{K}$. Let $\mathbf{1}$ be the column vector (of dimension n) whose entries are all 1. Then $\mathbf{1}\mathbf{1}^\top$ is the $n \times n$ matrix whose entries are all 1. If A is an $n \times n$ matrix, then $\mathbf{1}^\top A$ is the row vector consisting of the sums of the columns of A, $A\mathbf{1}$ is the column vector consisting of the sums of the rows of A, and $\mathbf{1}^\top A\mathbf{1}$ is the sum of all the entries in A. Then it is easy to see that the kernel matrix corresponding to the kernel function $\widehat{\kappa}$ is given by

$$\widehat{\mathbf{K}} = \mathbf{K} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top\mathbf{K} - \frac{1}{n}\mathbf{K}\mathbf{1}\mathbf{1}^\top + \frac{1}{n^2}(\mathbf{1}^\top\mathbf{K}\mathbf{1})\mathbf{1}\mathbf{1}^\top.$$

Suppose $\widehat{X} = X - \mu$ has rank r. To overcome the second problem, note that if

$$\widehat{X} = VDU^\top$$

is an SVD for $\widehat{X}$, then

$$\widehat{X}^\top = UD^\top V^\top$$

is an SVD for $\widehat{X}^\top$, and the $r \times r$ submatrix of $D^\top$ consisting of the first r rows and r columns of $D^\top$ (and D), is the diagonal Σ_r matrix consisting of the singular values $\sigma_1 \geq \cdots \geq \sigma_r$ of $\widehat{X}$, so we can express the matrix U_r consisting of the first r columns u_k of U in terms of the matrix V_r consisting of the first r columns v_k of V $(1 \leq k \leq r)$ as

$$U_r = \widehat{X}^\top V_r \Sigma_r^{-1}.$$

Furthermore, $\sigma_1^2 \geq \cdots \geq \sigma_r^2$ are the nonzero eigenvalues of $\widehat{\mathbf{K}} = \widehat{X}\widehat{X}^\top$, and the columns of V_r are corresponding unit eigenvectors of $\widehat{\mathbf{K}}$. From

$$U_r = \widehat{X}^\top V_r \Sigma_r^{-1}$$

the kth column u_k of U_r (which is a unit eigenvector of $\widehat{X}^\top\widehat{X}$ associated with the eigenvalue σ_k^2) is given by

$$u_k = \sum_{i=1}^n \sigma_k^{-1}(v_k)_i \widehat{X}_i^\top = \sum_{i=1}^n \sigma_k^{-1}(v_k)_i \widehat{\varphi(x_i)}, \quad 1 \leq k \leq r,$$

so the projection of $\widehat{\varphi(x)}$ onto u_k is given by

$$\begin{aligned}\langle \widehat{\varphi(x)}, u_k \rangle &= \left\langle \widehat{\varphi(x)}, \sum_{i=1}^n \sigma_k^{-1}(v_k)_i \widehat{\varphi(x_i)} \right\rangle \\ &= \sum_{i=1}^n \sigma_k^{-1}(v_k)_i \left\langle \widehat{\varphi(x)}, \widehat{\varphi(x_i)} \right\rangle = \sum_{i=1}^n \sigma_k^{-1}(v_k)_i \widehat{\kappa}(x, x_i).\end{aligned}$$

Therefore, the jth component of the principal component Y_k in the principal direction u_k is given by

$$(Y_k)_j = \langle X_j - \mu, u_k\rangle = \sum_{i=1}^n \sigma_k^{-1}(v_k)_i \widehat{\kappa}(x_j, x_i) = \sum_{i=1}^n \sigma_k^{-1}(v_k)_i \widehat{\mathbf{K}}_{ij}.$$

The generalization of kernel PCA to a general embedding $\varphi\colon \mathcal{X} \to F$ of $\mathcal{X}$ in a (real) feature space $(F, \langle -, -\rangle)$ (where F is not restricted to be equal to $\mathbb{R}^d$) with the kernel matrix $\mathbf{K}$ given by

$$\mathbf{K}_{ij} = \langle \varphi(x_i), \varphi(x_j)\rangle,$$

goes as follows.

- Let r be the rank of $\widehat{\mathbf{K}}$, where
$$\widehat{\mathbf{K}} = \mathbf{K} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top \mathbf{K} - \frac{1}{n}\mathbf{K}\mathbf{1}\mathbf{1}^\top + \frac{1}{n^2}(\mathbf{1}^\top \mathbf{K}\mathbf{1})\mathbf{1}\mathbf{1}^\top,$$
let $\sigma_1^2 \geq \cdots \geq \sigma_r^2$ be the nonzero eigenvalues of $\widehat{\mathbf{K}}$, and let $v_1, \ldots, v_r$ be corresponding unit eigenvectors. The notation
$$\alpha_k = \sigma_k^{-1} v_k$$
is often used, where the α_k are called the *dual variables*.
- The column vector Y_k $(1 \leq k \leq r)$ defined by
$$Y_k = \left(\sum_{i=1}^n (\alpha_k)_i \widehat{\mathbf{K}}_{ij}\right)_{j=1}^n$$
is called the *kth kernel principal component* (for short *kth kernel PCA*) of the data set $S = \{x_1, \ldots, x_n\}$ in the direction $u_k = \sum_{i=1}^n \sigma_k^{-1}(v_k)_i \widehat{X}_i^\top$ (even though the matrix $\widehat{X}$ is *not known*).

17.5 Summary

The main concepts and results of this chapter are listed below:

- Feature map, feature embedding, feature space.
- Kernel function.
- Positive definite kernel, real positive definite kernel.
- Gram matrix.
- Hadamard product, Schur product.
- Gaussian kernel.
- Reproducing kernel Hilbert space (RKHS).
- Reproducing property.
- Intersection kernel, union complement kernel, agreement kernel.
- Kernel PCA.
- k-th kernel PCA.

17.6 Problems

Problem 17.1. Referring back to Example 17.3, prove that if $\varphi_1\colon X \to \mathbb{R}^{n_1}$ and $\varphi_2\colon X \to \mathbb{R}^{n_2}$ are two feature maps and if $\kappa_1(x,y) = \langle \varphi_1(x), \varphi_1(y)\rangle$ and $\kappa_2(x,y) = \langle \varphi_2(x), \varphi_2(y)\rangle$ are the corresponding kernel functions, then the map defined by

$$\kappa(x,y) = \kappa_1(x,y)\kappa_2(x,y)$$

is a kernel function, for the feature space $\mathbb{R}^{n_1} \times \mathbb{R}^{n_2}$ and the feature map

$$\varphi(x)_{(i,j)} = (\varphi_1(x))_i(\varphi_2(x))_j, \qquad 1 \le i \le n_1,\ 1 \le j \le n_2.$$

Problem 17.2. Referring back to Example 17.3, prove that the feature embedding $\varphi\colon X \to \mathbb{R}^{\binom{n+m-1}{m}}$ given by

$$\varphi_{(i_1,\ldots,i_n)}(x) = \binom{m}{i_1 \cdots i_n}^{1/2} (\varphi_1(x))_1^{i_1}(\varphi_1(x))_1^{i_2} \cdots (\varphi_1(x))_1^{i_n},$$
$$i_1 + i_2 + \cdots + i_n = m,\ i_j \in \mathbb{N},$$

where the n-tuples $(i_1,\ldots,i_n)$ are ordered lexicographically, defines the kernel function κ given by $\kappa(x,y) = (\kappa_1(x,y))^m$.

Problem 17.3. In Example 17.6, prove that for any two subsets A_1 and A_2 of D,

$$\langle \varphi(A_1), \varphi(A_2)\rangle = 2^{|A_1 \cap A_2|},$$

the number of common subsets of A_1 and A_2.

Problem 17.4. Prove that the pointwise limit of positive definite kernels is also a positive definite kernel.

Problem 17.5. Prove that if κ_1 is a positive definite kernel, then

$$\kappa(x,y) = e^{-\frac{\kappa_1(x,x)+\kappa_1(y,y)-2\kappa_1(x,y)}{2\sigma^2}}$$

is a positive definite kernel.

Chapter 18

Soft Margin Support Vector Machines

In Sections 14.5 and 14.6 we considered the problem of separating two nonempty disjoint finite sets of p *blue* points $\{u_i\}_{i=1}^p$ and q *red* points $\{v_j\}_{j=1}^q$ in $\mathbb{R}^n$. The goal is to find a hyperplane H of equation $w^\top x - b = 0$ (where $w \in \mathbb{R}^n$ is a nonzero vector and $b \in \mathbb{R}$), such that all the blue points u_i are in one of the two open half-spaces determined by H, and all the red points v_j are in the other open half-space determined by H. SVM picks a hyperplane which maximizes the minimum distance from these points to the hyperplane. See Figure 18.1.

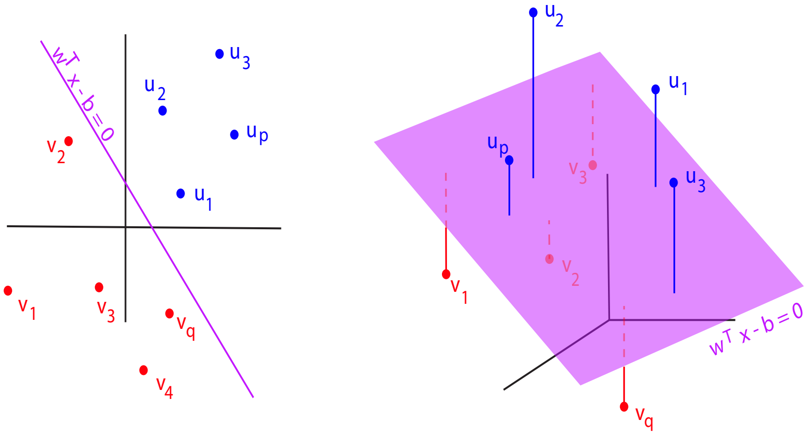


Fig. 18.1 Two examples of the SVM separation problem. The left figure is SVM in $\mathbb{R}^2$, while the right figure is SVM in $\mathbb{R}^3$.

In this chapter we return to the problem of separating two disjoint sets of points, $\{u_i\}_{i=1}^p$ and $\{v_j\}_{j=1}^q$, but this time we do not assume that these two sets are separable. To cope with nonseparability, we allow points to invade the safety zone around the separating hyperplane, and even points on the wrong side of the hyperplane. Such a method is called *soft margin support vector machine*. We discuss variations of this method, including ν-SV classification. In each case we present a careful derivation of the dual.

If the sets of points $\{u_1, \ldots, u_p\}$ and $\{v_1, \ldots, v_q\}$ are not linearly separable (with $u_i, v_j \in \mathbb{R}^n$), we can use a trick from linear programming which is to introduce nonnegative "slack variables" $\epsilon = (\epsilon_1, \ldots, \epsilon_p) \in \mathbb{R}^p$ and $\xi = (\xi_1, \ldots, \xi_q) \in \mathbb{R}^q$ to relax the "hard" constraints

$$\begin{aligned} w^\top u_i - b &\geq \delta & i &= 1, \ldots, p \\ -w^\top v_j + b &\geq \delta & j &= 1, \ldots, q \end{aligned}$$

of Problem (SVM$_{h1}$) from Section 14.5 to the "soft" constraints

$$\begin{aligned} w^\top u_i - b &\geq \delta - \epsilon_i, & \epsilon_i &\geq 0 & i &= 1, \ldots, p \\ -w^\top v_j + b &\geq \delta - \xi_j, & \xi_j &\geq 0 & j &= 1, \ldots, q. \end{aligned}$$

Recall that $w \in \mathbb{R}^n$ and $b, \delta \in \mathbb{R}$.

If $\epsilon_i > 0$, the point u_i may be misclassified, in the sense that it can belong to the margin (the slab), or even to the wrong half-space classifying the negative (red) points. See Figures 18.5 (2) and (3). Similarly, if $\xi_j > 0$, the point v_j may be misclassified, in the sense that it can belong to the margin (the slab), or even to the wrong half-space classifying the positive (blue) points. We can think of ϵ_i as a measure of how much the constraint $w^\top u_i - b \geq \delta$ is violated, and similarly of ξ_j as a measure of how much the constraint $-w^\top v_j + b \geq \delta$ is violated. If $\epsilon = 0$ and $\xi = 0$, then we recover the original constraints. By making ϵ and ξ large enough, these constraints can always be satisfied. We add the constraint $w^\top w \leq 1$ and we minimize $-\delta$.

If instead of the constraints of Problem (SVM$_{h1}$) we use the hard constraints

$$\begin{aligned} w^\top u_i - b &\geq 1 & i &= 1, \ldots, p \\ -w^\top v_j + b &\geq 1 & j &= 1, \ldots, q \end{aligned}$$

of Problem (SVM$_{h2}$) (see Example 14.6), then we relax to the soft constraints

$$\begin{aligned} w^\top u_i - b &\geq 1 - \epsilon_i, & \epsilon_i &\geq 0 & i &= 1, \ldots, p \\ -w^\top v_j + b &\geq 1 - \xi_j, & \xi_j &\geq 0 & j &= 1, \ldots, q. \end{aligned}$$

In this case there is no constraint on w, but we minimize $(1/2)w^\top w$.

Ideally we would like to find a separating hyperplane that *minimizes the number of misclassified points*, which means that the variables ϵ_i and ξ_j should be as small as possible, but there is a trade-off in maximizing the margin (the thickness of the slab), and minimizing the number of misclassified points. This is reflected in the choice of the objective function, and there are several options, depending on whether we minimize a linear function of the variables ϵ_i and ξ_j, or a quadratic functions of these variables, or whether we include the term $(1/2)b^2$ in the objective function. These methods are known as *support vector classification* algorithms (for short *SVC* algorithms).

SVC algorithms seek an "optimal" separating hyperplane H of equation $w^\top x - b = 0$. If some new data $x \in \mathbb{R}^n$ comes in, we can classify it by determining in which of the two half spaces determined by the hyperplane H they belong by computing the sign of the quantity $w^\top x - b$. The function $\text{sgn}\colon \mathbb{R} \to \{-1, 1\}$ is given by

$$\text{sgn}(x) = \begin{cases} +1 & \text{if } x \geq 0 \\ -1 & \text{if } x < 0. \end{cases}$$

Then we define the *(binary) classification function* associated with the hyperplane H of equation $w^\top x - b = 0$ as

$$f(x) = \text{sgn}(w^\top x - b).$$

Remarkably, all the known optimization problems for finding this hyperplane share the property that the weight vector w and the constant b are given by expressions that *only involves inner products of the input data points u_i and v_j*, and so does the classification function

$$f(x) = \text{sgn}(w^\top x - b).$$

This is a key fact that allows a far reaching generalization of the support vector machine using the method of *kernels*.

The method of kernels consists in assuming that the input space $\mathbb{R}^n$ is embedded in a larger (possibly infinite dimensional) Euclidean space F (with an inner product $\langle -, - \rangle$) usually called a *feature space*, using a function

$$\varphi\colon \mathbb{R}^n \to F$$

called a *feature map*. The function $\kappa\colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ given by

$$\kappa(x, y) = \langle \varphi(x), \varphi(y) \rangle$$

is the kernel function associated with the embedding φ; see Chapter 17. The idea is that the feature map φ "unwinds" the input data, making it somehow more linear in the higher dimensional space F. Now even if we don't know what the feature space F is and what the embedding map φ is, we can pretend to solve our separation problem in F for the embedded data points $\varphi(u_i)$ and $\varphi(v_j)$. Thus we seek a hyperplane H of equation

$$\langle w, \zeta\rangle - b = 0, \quad \zeta \in F,$$

in the feature space F, to attempt to separate the points $\varphi(u_i)$ and the points $\varphi(v_j)$. As we said, it turns out that w and b are given by expression involving only the inner products $\kappa(u_i, u_j) = \langle\varphi(u_i), \varphi(u_j)\rangle, \kappa(u_i, v_j) = \langle\varphi(u_i), \varphi(v_j)\rangle$, and $\kappa(v_i, v_j) = \langle\varphi(v_i), \varphi(v_j)\rangle$, which form the symmetric $(p+q)\times(p+q)$ matrix $\mathbf{K}$ (a kernel matrix) given by

$$\mathbf{K}_{ij} = \begin{cases} \kappa(u_i, u_j) & 1 \leq i \leq p,\ 1 \leq j \leq q \\ -\kappa(u_i, v_{j-p}) & 1 \leq i \leq p,\ p+1 \leq j \leq p+q \\ -\kappa(v_{i-p}, u_j) & p+1 \leq i \leq p+q,\ 1 \leq j \leq p \\ \kappa(v_{i-p}, v_{j-q}) & p+1 \leq i \leq p+q,\ p+1 \leq j \leq p+q. \end{cases}$$

For example, if $p = 2$ and $q = 3$, we have the matrix

$$\mathbf{K} = \begin{pmatrix} \kappa(u_1, u_1) & \kappa(u_1, u_2) & -\kappa(u_1, v_1) & -\kappa(u_1, v_2) & -\kappa(u_1, v_3) \\ \kappa(u_2, u_1) & \kappa(u_2, u_2) & -\kappa(u_2, v_1) & -\kappa(u_2, v_2) & -\kappa(u_2, v_3) \\ -\kappa(v_1, u_1) & -\kappa(v_1, u_2) & \kappa(v_1, v_1) & \kappa(v_1, v_2) & \kappa(v_1, v_3) \\ -\kappa(v_2, u_1) & -\kappa(v_2, u_2) & \kappa(v_2, v_1) & \kappa(v_2, v_2) & \kappa(v_2, v_3) \\ -\kappa(v_3, u_1) & -\kappa(v_3, u_2) & \kappa(v_3, v_1) & \kappa(v_3, v_2) & \kappa(v_3, v_3) \end{pmatrix}.$$

Then the classification function

$$f(x) = \operatorname{sgn}(\langle w, \varphi(x)\rangle - b)$$

for points in the original data space $\mathbb{R}^n$ is also expressed solely in terms of the matrix $\mathbf{K}$ and the inner products $\kappa(u_i, x) = \langle\varphi(u_i), \varphi(x)\rangle$ and $\kappa(v_j, x) = \langle\varphi(v_j), \varphi(x)\rangle$. As a consequence, in the original data space $\mathbb{R}^n$, the hypersurface

$$\mathcal{S} = \{x \in \mathbb{R}^n \mid \langle w, \varphi(x)\rangle - b = 0\}$$

separates the data points u_i and v_j, but it is not an affine subspace of $\mathbb{R}^n$. The classification function f tells us on which "side" of $\mathcal{S}$ is a new data point $x \in \mathbb{R}^n$. Thus, we managed to separate the data points u_i and v_j that are not separable by an affine hyperplane, by a *nonaffine hypersurface* $\mathcal{S}$, by assuming that an embdedding $\varphi\colon \mathbb{R}^n \to F$ exists, even though we don't know what it is, but having access to F through the kernel function $\kappa\colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ given by the inner products $\kappa(x, y) = \langle \varphi(x), \varphi(y) \rangle$.

In practice the art of using the kernel method is to choose the right kernel (as the knight says in Indiana Jones, to "choose wisely.").

The method of kernels is very flexible. It also applies to the soft margin versions of SVM, but also to regression problems, to principal component analysis (PCA), and to other problems arising in machine learning.

We discussed the method of kernels in Chapter 17. Other comprehensive presentations of the method of kernels are found in Schölkopf and Smola [Schölkopf and Smola (2002)] and Shawe–Taylor and Christianini [Shawe-Taylor and Cristianini (2004)]. See also Bishop [Bishop (2006)].

We first consider the soft margin SVM arising from Problem (SVM$_{h1}$).

18.1 Soft Margin Support Vector Machines; (SVM$_{s1}$)

In this section we derive the dual function G associated with the following version of the soft margin SVM coming from Problem (SVM$_{h1}$), where the maximization of the margin δ has been replaced by the minimization of $-\delta$, and where we added a "regularizing term" $K\Big(\sum_{i=1}^p \epsilon_i + \sum_{j=1}^q \xi_j\Big)$ whose purpose is to make $\epsilon \in \mathbb{R}^p$ and $\xi \in \mathbb{R}^q$ *sparse* (that is, try to make ϵ_i and ξ_j have as many zeros as possible), where $K > 0$ is a fixed constant that can be adjusted to determine the influence of this regularizing term. If the primal problem (SVM$_{s1}$) has an optimal solution $(w, \delta, b, \epsilon, \xi)$, we attempt to use the dual function G to obtain it, but we will see that with this particular formulation of the problem, the constraint $w^\top w \leq 1$ causes troubles even though it is convex.

Soft margin SVM (SVM$_{s1}$):

$$\begin{aligned}
&\text{minimize} \quad -\delta + K\left(\sum_{i=1}^p \epsilon_i + \sum_{j=1}^q \xi_j\right)\\
&\text{subject to}\\
&\qquad w^\top u_i - b \geq \delta - \epsilon_i, \quad \epsilon_i \geq 0 \qquad i = 1, \ldots, p\\
&\qquad -w^\top v_j + b \geq \delta - \xi_j, \quad \xi_j \geq 0 \qquad j = 1, \ldots, q\\
&\qquad w^\top w \leq 1.
\end{aligned}$$

It is customary to write $\ell = p+q$. Figure 18.2 illustrates the correct margin half space associated with $w^\top x - b - \delta = 0$ while Figure 18.3 illustrates the correct margin half space associated with $w^\top x - b + \delta = 0$. Ideally, all the points should be contained in one of the two correct shifted margin regions described by affine constraints $w^\top u_i - b \geq \delta - \epsilon_i$, or $-w^\top v_j + b \geq \delta - \xi_j$.

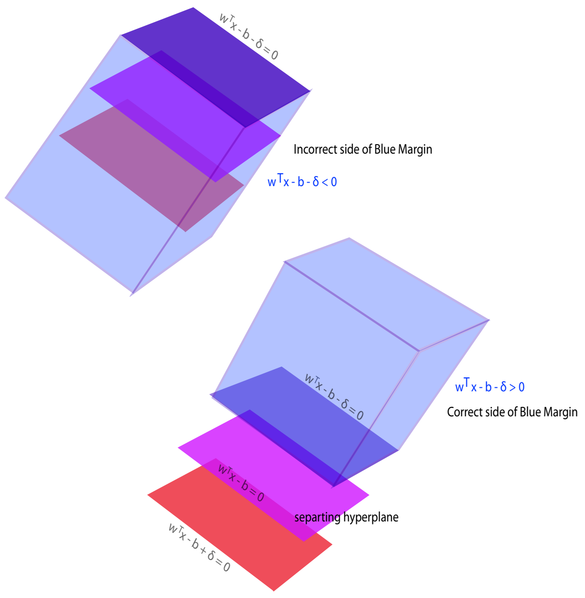


Fig. 18.2 The blue margin half space associated with $w^\top x - b - \delta = 0$.

For this problem, the primal problem may have an optimal solution $(w, \delta, b, \epsilon, \xi)$ with $\|w\| = 1$ and $\delta > 0$, but if the sets of points are not linearly separable then an optimal solution of the dual may not yield w.

The objective function of our problem is affine and the only nonaffine constraint $w^\top w \leq 1$ is convex. This constraint is qualified because for any $w \neq 0$ such that $w^\top w < 1$ and for any $\delta > 0$ and any b we can pick ϵ and ξ large enough so that the constraints are satisfied. Consequently, by Theorem 14.5(2) *if* the primal problem (SVM$_{s1}$) has an optimal solution, *then* the dual problem has a solution too, and the duality gap is zero.

Unfortunately this does not imply that an optimal solution of the dual yields an optimal solution of the primal because the hypotheses of Theorem 14.5(1) fail to hold. In general, there may not be a unique vector $(w, \epsilon, \xi, b, \delta)$ such that

$$\inf_{w,\epsilon,\xi,b,\delta} L(w, \epsilon, \xi, b, \delta, \lambda, \mu, \alpha, \beta, \gamma) = G(\lambda, \mu, \alpha, \beta, \gamma).$$

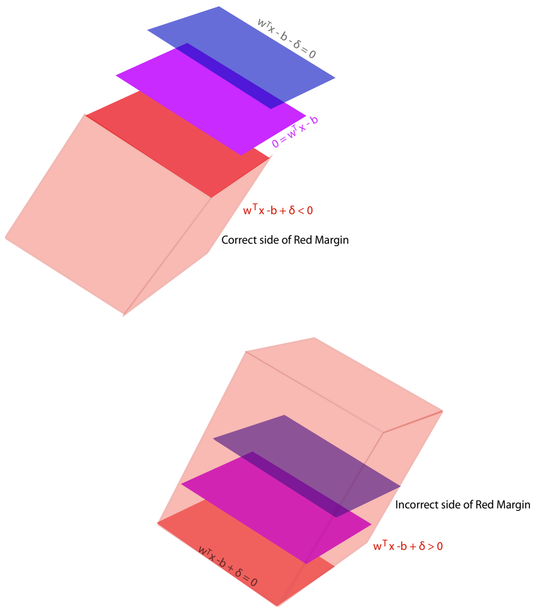


Fig. 18.3 The red margin half space associated with $w^\top x - b + \delta = 0$.

If the sets $\{u_i\}$ and $\{v_j\}$ are *not* linearly separable, then the dual problem may have a solution for which $\gamma = 0$,

$$\sum_{i=1}^{p} \lambda_i = \sum_{j=1}^{q} \mu_j = \frac{1}{2},$$

and

$$\sum_{i=1}^{p} \lambda_i u_i = \sum_{j=1}^{q} \mu_j v_j,$$

so that the dual function $G(\lambda, \mu, \alpha, \beta, \ \gamma)$, which is a *partial function*, is defined and has the value $G(\lambda, \mu, \alpha, \beta, 0) = 0$. Such a pair (λ, μ) corresponds to the coefficients of two convex combinations

$$\sum_{i=1}^{p} 2\lambda_i u_i = \sum_{j=1}^{q} 2\mu_j v_j$$

which correspond to the *same point* in the (nonempty) intersection of the convex hulls $\mathrm{conv}(u_1, \ldots, u_p)$ and $\mathrm{conv}(v_1, \ldots, v_q)$. It turns out that the only connection between w and the dual function is the equation

$$2\gamma w = \sum_{i=1}^{p} \lambda_i u_i - \sum_{j=1}^{q} \mu_j v_j,$$

and when $\gamma = 0$ this is equation is $0 = 0$, *so the dual problem is useless to determine w.* This point seems to have been missed in the literature (for example, in Shawe–Taylor and Christianini [Shawe-Taylor and Cristianini (2004)], Section 7.2). What the dual problem does show is that $\delta \geq 0$. However, if $\gamma \neq 0$, then w is determined by any solution (λ, μ) of the dual.

It still remains to compute δ and b, which can be done under a mild hypothesis that we call the **Standard Margin Hypothesis**.

Let $\lambda \in \mathbb{R}^p_+$ be the Lagrange multipliers associated with the inequalities $w^\top u_i - b \geq \delta - \epsilon_i$, let $\mu \in \mathbb{R}^q_+$ be the Lagrange multipliers are associated with the inequalities $-w^\top v_j + b \geq \delta - \xi_j$, let $\alpha \in \mathbb{R}^p_+$ be the Lagrange multipliers associated with the inequalities $\epsilon_i \geq 0$, $\beta \in \mathbb{R}^q_+$ be the Lagrange multipliers associated with the inequalities $\xi_j \geq 0$, and let $\gamma \in \mathbb{R}^+$ be the Lagrange multiplier associated with the inequality $w^\top w \leq 1$.

The linear constraints are given by the $2(p+q) \times (n+p+q+2)$ matrix given in block form by

$$C = \begin{pmatrix} X^\top & -I_{p+q} & \begin{matrix} \mathbf{1}_p \\ -\mathbf{1}_q \end{matrix} & \mathbf{1}_{p+q} \\ 0_{p+q,n} & -I_{p+q} & 0_{p+q} & 0_{p+q} \end{pmatrix},$$

where X is the $n \times (p+q)$ matrix

$$X = \begin{pmatrix} -u_1 & \cdots & -u_p & v_1 & \cdots & v_q \end{pmatrix},$$

and the linear constraints are expressed by

$$\begin{pmatrix} X^\top & -I_{p+q} & \begin{matrix} \mathbf{1}_p \\ -\mathbf{1}_q \end{matrix} & \mathbf{1}_{p+q} \\ 0_{p+q,n} & -I_{p+q} & 0_{p+q} & 0_{p+q} \end{pmatrix} \begin{pmatrix} w \\ \epsilon \\ \xi \\ b \\ \delta \end{pmatrix} \leq \begin{pmatrix} 0_{p+q} \\ 0_{p+q} \end{pmatrix}.$$

More explicitly, C is the following matrix:

$$C = \begin{pmatrix} -u_1^\top & -1 & \cdots & 0 & 0 & \cdots & 0 & 1 & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ -u_p^\top & 0 & \cdots & -1 & 0 & \cdots & 0 & 1 & 1 \\ v_1^\top & 0 & \cdots & 0 & -1 & \cdots & 0 & -1 & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ v_q^\top & 0 & \cdots & 0 & 0 & \cdots & -1 & -1 & 1 \\ 0 & -1 & \cdots & 0 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & -1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & \cdots & 0 & -1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & -1 & 0 & 0 \end{pmatrix}.$$

The objective function is given by

$$J(w, \epsilon, \xi, b, \delta) = -\delta + K \left(\epsilon^\top \; \xi^\top\right) \mathbf{1}_{p+q}.$$

The Lagrangian $L(w, \epsilon, \xi, b, \delta, \lambda, \mu, \alpha, \beta, \gamma)$ with $\lambda, \alpha \in \mathbb{R}^p_+$, $\mu, \beta \in \mathbb{R}^q_+$, and $\gamma \in \mathbb{R}^+$ is given by

$$\begin{aligned} L(w, \epsilon, \xi, b, \delta, \lambda, \mu, \alpha, \beta, \gamma) &= -\delta + K \left(\epsilon^\top \; \xi^\top\right) \mathbf{1}_{p+q} \\ &\quad + \left(w^\top \; \left(\epsilon^\top \; \xi^\top\right) \; b \; \delta\right) C^\top \begin{pmatrix} \lambda \\ \mu \\ \alpha \\ \beta \end{pmatrix} + \gamma(w^\top w - 1). \end{aligned}$$

Since

$$\begin{aligned} \left(w^\top \; \left(\epsilon^\top \; \xi^\top\right) \; b \; \delta\right) C^\top \begin{pmatrix} \lambda \\ \mu \\ \alpha \\ \beta \end{pmatrix} &= w^\top X \begin{pmatrix} \lambda \\ \mu \end{pmatrix} - \epsilon^\top(\lambda + \alpha) - \xi^\top(\mu + \beta) \\ &\quad + b(\mathbf{1}_p^\top \lambda - \mathbf{1}_q^\top \mu) + \delta(\mathbf{1}_p^\top \lambda + \mathbf{1}_q^\top \mu), \end{aligned}$$

the Lagrangian can be written as

$$\begin{aligned} L(w, \epsilon, \xi, b, \delta, \lambda, \mu, \alpha, \beta, \gamma) &= -\delta + K(\epsilon^\top \mathbf{1}_p + \xi^\top \mathbf{1}_q) + w^\top X \begin{pmatrix} \lambda \\ \mu \end{pmatrix} \\ + \gamma(w^\top w - 1) - \epsilon^\top(\lambda + \alpha) &- \xi^\top(\mu + \beta) + b(\mathbf{1}_p^\top \lambda - \mathbf{1}_q^\top \mu) + \delta(\mathbf{1}_p^\top \lambda + \mathbf{1}_q^\top \mu) \\ &= (\mathbf{1}_p^\top \lambda + \mathbf{1}_q^\top \mu - 1)\delta + w^\top X \begin{pmatrix} \lambda \\ \mu \end{pmatrix} + \gamma(w^\top w - 1) \\ &\quad + \epsilon^\top(K\mathbf{1}_p - (\lambda + \alpha)) + \xi^\top(K\mathbf{1}_q - (\mu + \beta)) + b(\mathbf{1}_p^\top \lambda - \mathbf{1}_q^\top \mu). \end{aligned}$$

To find the dual function $G(\lambda, \mu, \alpha, \beta, \gamma)$ we minimize $L(w, \epsilon, \xi, b, \delta, \lambda, \mu, \alpha, \beta, \gamma)$ with respect to w, ϵ, ξ, b, and δ. Since the Lagrangian is convex and $(w, \epsilon, \xi, b, \delta) \in \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^q \times \mathbb{R} \times \mathbb{R}$, a convex open set, by Theorem 4.5, the Lagrangian has a minimum in $(w, \epsilon, \xi, b, \delta)$ iff $\nabla L_{w,\epsilon,\xi,b,\delta} = 0$, so we compute the gradient with respect to $w, \epsilon, \xi, b, \delta$, and we get

$$\nabla L_{w,\epsilon,\xi,b,\delta} = \begin{pmatrix} X \begin{pmatrix} \lambda \\ \mu \end{pmatrix} + 2\gamma w \\ K\mathbf{1}_p - (\lambda + \alpha) \\ K\mathbf{1}_q - (\mu + \beta) \\ \mathbf{1}_p^\top \lambda - \mathbf{1}_q^\top \mu \\ \mathbf{1}_p^\top \lambda + \mathbf{1}_q^\top \mu - 1 \end{pmatrix}.$$

By setting $\nabla L_{w,\epsilon,\xi,b,\delta} = 0$ we get the equations

$$\begin{aligned} 2\gamma w &= -X \begin{pmatrix} \lambda \\ \mu \end{pmatrix} \\ \lambda + \alpha &= K\mathbf{1}_p \\ \mu + \beta &= K\mathbf{1}_q \\ \mathbf{1}_p^\top \lambda &= \mathbf{1}_q^\top \mu \\ \mathbf{1}_p^\top \lambda + \mathbf{1}_q^\top \mu &= 1. \end{aligned} \qquad (*_w)$$

The second and third equations are equivalent to the inequalities

$$0 \leq \lambda_i, \mu_j \leq K, \quad i = 1, \ldots, p,\ j = 1, \ldots, q,$$

often called *box constraints*, and the fourth and fifth equations yield

$$\mathbf{1}_p^\top \lambda = \mathbf{1}_q^\top \mu = \frac{1}{2}.$$

First let us consider the singular case $\gamma = 0$. In this case, $(*_w)$ implies that

$$X \begin{pmatrix} \lambda \\ \mu \end{pmatrix} = 0,$$

and the term $\gamma(w^\top w - 1)$ is missing from the Lagrangian, which in view of the other four equations above reduces to

$$L(w, \epsilon, \xi, b, \delta, \lambda, \mu, \alpha, \beta, 0) = w^\top X \begin{pmatrix} \lambda \\ \mu \end{pmatrix} = 0.$$

In summary, we proved that if $\gamma = 0$, then

$$G(\lambda, \mu, \alpha, \beta, 0) = \begin{cases} 0 & \text{if } \begin{cases} \sum_{i=1}^p \lambda_i = \sum_{j=1}^q \mu_j = \frac{1}{2} \\ 0 \leq \lambda_i \leq K,\ i = 1, \ldots, p \\ 0 \leq \mu_j \leq K,\ j = 1, \ldots, q \end{cases} \\ -\infty & \text{otherwise} \\ & \text{and } \sum_{i=1}^p \lambda_i u_i - \sum_{j=1}^q \mu_j v_j = 0. \end{cases}$$

Geometrically, (λ, μ) corresponds to the coefficients of two convex combinations

$$\sum_{i=1}^{p} 2\lambda_i u_i = \sum_{j=1}^{q} 2\mu_j v_j$$

which correspond to the *same point* in the intersection of the convex hulls $\mathrm{conv}(u_1, \ldots, u_p)$ and $\mathrm{conv}(v_1, \ldots, v_q)$ iff the sets $\{u_i\}$ and $\{v_j\}$ are *not linearly separable*. If the sets $\{u_i\}$ and $\{v_j\}$ are *linearly separable*, then the convex hulls $\mathrm{conv}(u_1, \ldots, u_p)$ and $\mathrm{conv}(v_1, \ldots, v_q)$ are disjoint, which implies that $\gamma > 0$.

Let us now assume that $\gamma > 0$. Plugging back w from equation $(*_w)$ into the Lagrangian, after simplifications we get

$$\begin{aligned} G(\lambda, \mu, \alpha, \beta, \gamma) &= -\frac{1}{2\gamma} \begin{pmatrix} \lambda^\top & \mu^\top \end{pmatrix} X^\top X \begin{pmatrix} \lambda \\ \mu \end{pmatrix} + \frac{\gamma}{4\gamma^2} \begin{pmatrix} \lambda^\top & \mu^\top \end{pmatrix} X^\top X \begin{pmatrix} \lambda \\ \mu \end{pmatrix} - \gamma \\ &= -\frac{1}{4\gamma} \begin{pmatrix} \lambda^\top & \mu^\top \end{pmatrix} X^\top X \begin{pmatrix} \lambda \\ \mu \end{pmatrix} - \gamma, \end{aligned}$$

so if $\gamma > 0$ the dual function is independent of α, β and is given by

$$\begin{aligned} &G(\lambda, \mu, \alpha, \beta, \gamma) \\ &= \begin{cases} -\frac{1}{4\gamma} \begin{pmatrix} \lambda^\top & \mu^\top \end{pmatrix} X^\top X \begin{pmatrix} \lambda \\ \mu \end{pmatrix} - \gamma & \text{if } \begin{cases} \sum_{i=1}^{p} \lambda_i = \sum_{j=1}^{q} \mu_j = \frac{1}{2} \\ 0 \leq \lambda_i \leq K, \ i = 1, \ldots, p \\ 0 \leq \mu_j \leq K, \ j = 1, \ldots, q \end{cases} \\ -\infty & \text{otherwise.} \end{cases} \end{aligned}$$

Since $X^\top X$ is symmetric positive semidefinite and $\gamma \geq 0$, obviously

$$G(\lambda, \mu, \alpha, \beta, \gamma) \leq 0$$

for all $\gamma > 0$.

The dual program is given by

$$\begin{aligned}
&\text{maximize} \quad -\frac{1}{4\gamma}\begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix} X^\top X \begin{pmatrix}\lambda \\ \mu\end{pmatrix} - \gamma \quad \text{if } \gamma > 0 \\
&\qquad\qquad 0 \quad \text{if} \quad \gamma = 0 \\
&\text{subject to} \\
&\qquad \sum_{i=1}^p \lambda_i - \sum_{j=1}^q \mu_j = 0 \\
&\qquad \sum_{i=1}^p \lambda_i + \sum_{j=1}^q \mu_j = 1 \\
&\qquad 0 \leq \lambda_i \leq K, \quad i = 1, \ldots, p \\
&\qquad 0 \leq \mu_j \leq K, \quad j = 1, \ldots, q.
\end{aligned}$$

Also, if $\gamma = 0$, then $X \begin{pmatrix}\lambda \\ \mu\end{pmatrix} = 0$.

Maximizing with respect to $\gamma > 0$ by setting $\frac{\partial}{\partial\gamma} G(\lambda, \mu, \alpha, \beta, \gamma) = 0$ yields

$$\gamma^2 = \frac{1}{4}\begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix} X^\top X \begin{pmatrix}\lambda \\ \mu\end{pmatrix},$$

so we obtain

$$G(\lambda, \mu) = -\left(\begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix} X^\top X \begin{pmatrix}\lambda \\ \mu\end{pmatrix}\right)^{1/2}.$$

Finally, since $G(\lambda, \mu) = 0$ and $X \begin{pmatrix}\lambda \\ \mu\end{pmatrix} = 0$ if $\gamma = 0$, the dual program is equivalent to the following minimization program:

Dual of Soft margin SVM (SVM_{s1}):

$$\begin{aligned}
&\text{minimize} \quad \begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix} X^\top X \begin{pmatrix}\lambda \\ \mu\end{pmatrix} \\
&\text{subject to} \\
&\qquad \sum_{i=1}^p \lambda_i - \sum_{j=1}^q \mu_j = 0 \\
&\qquad \sum_{i=1}^p \lambda_i + \sum_{j=1}^q \mu_j = 1 \\
&\qquad 0 \leq \lambda_i \leq K, \quad i = 1, \ldots, p \\
&\qquad 0 \leq \mu_j \leq K, \quad j = 1, \ldots, q.
\end{aligned}$$

Observe that the constraints imply that K must be chosen so that

$$K \geq \max\left\{\frac{1}{2p}, \frac{1}{2q}\right\}.$$

If $(w, \delta, b, \epsilon, \xi)$ is an optimal solution of Problem (SVM$_{s1}$), then the complementary slackness conditions yield a classification of the points u_i and v_j in terms of the values of λ and μ. Indeed, we have $\epsilon_i \alpha_i = 0$ for $i = 1, \ldots, p$ and $\xi_j \beta_j = 0$ for $j = 1, \ldots, q$. Also, if $\lambda_i > 0$, then corresponding constraint is active, and similarly if $\mu_j > 0$. Since $\lambda_i + \alpha_i = K$, it follows that $\epsilon_i \alpha_i = 0$ iff $\epsilon_i(K - \lambda_i) = 0$, and since $\mu_j + \beta_j = K$, we have $\xi_j \beta_j = 0$ iff $\xi_j(K - \mu_j) = 0$. Thus if $\epsilon_i > 0$, then $\lambda_i = K$, and if $\xi_j > 0$, then $\mu_j = K$. Consequently, if $\lambda_i < K$, then $\epsilon_i = 0$ and u_i is correctly classified, and similarly if $\mu_j < K$, then $\xi_j = 0$ and v_j is correctly classified. We have the following classification:

(1) If $0 < \lambda_i < K$, then $\epsilon_i = 0$ and the i-th inequality is active, so

$$w^\top u_i - b - \delta = 0.$$

This means that u_i is on the blue margin (the hyperplane $H_{w,b+\delta}$ of equation $w^\top x = b + \delta$) and is classified correctly.
Similarly, if $0 < \mu_j < K$, then $\xi_j = 0$ and

$$w^\top v_j - b + \delta = 0,$$

so v_j is on the red margin (the hyperplane $H_{w,b-\delta}$ of equation $w^\top x = b - \delta$) and is classified correctly. See Figure 18.4.

(2) If $\lambda_i = K$, then the i-th inequality is active, so

$$w^\top u_i - b - \delta = -\epsilon_i.$$

If $\epsilon_i = 0$, then the point u_i is on the blue margin. If $\epsilon_i > 0$, then u_i is within the open half space bounded by the blue margin hyperplane $H_{w,b+\delta}$ and containing the separating hyperplane $H_{w,b}$; if $\epsilon_i \leq \delta$, then u_i is classified correctly, and if $\epsilon_i > \delta$, then u_i is misclassified (u_i lies on the wrong side of the separating hyperplane, the red side). See Figure 18.5.
Similarly, if $\mu_j = K$, then

$$w^\top v_j - b + \delta = \xi_j.$$

If $\xi_j = 0$, then the point v_j is on the red margin. If $\xi_j > 0$, then v_j is within the open half space bounded by the red margin hyperplane $H_{w,b-\delta}$ and containing the separating hyperplane $H_{w,b}$; if $\xi_j \leq \delta$, then

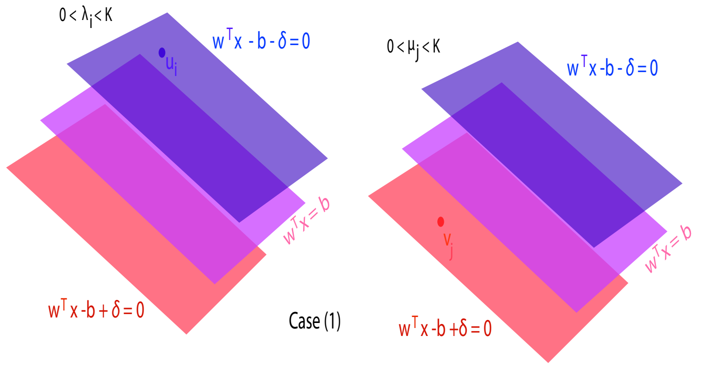


Fig. 18.4 When $0 < \lambda_i < K$, u_i is contained within the blue margin hyperplane. When $0 < \mu_j < K$, v_j is contained within the red margin hyperplane.

v_j is classified correctly, and if $\xi_j > \delta$, then v_j is misclassified (v_j lies on the wrong side of the separating hyperplane, the blue side). See Figure 18.5.

(3) If $\lambda_i = 0$, then $\epsilon_i = 0$ and the i-th inequality may or may not be active, so

$$w^\top u_i - b - \delta \geq 0.$$

Thus u_i is in the closed half space on the blue side bounded by the blue margin hyperplane $H_{w,b+\delta}$ (of course, classified correctly).
Similarly, if $\mu_j = 0$, then

$$w^\top v_j - b + \delta \leq 0$$

and v_j is in the closed half space on the red side bounded by the red margin hyperplane $H_{w,b-\delta}$ (of course, classified correctly). See Figure 18.6.

Definition 18.1. The vectors u_i on the blue margin $H_{w,b+\delta}$ and the vectors v_j on the red margin $H_{w,b-\delta}$ are called *support vectors*. Support vectors correspond to vectors u_i for which $w^\top u_i - b - \delta = 0$ (which implies $\epsilon_i = 0$), and vectors v_j for which $w^\top v_j - b + \delta = 0$ (which implies $\xi_j = 0$). Support vectors u_i such that $0 < \lambda_i < K$ and support vectors v_j such that $0 < \mu_j < K$ are *support vectors of type 1*. Support vectors of type 1 play a special

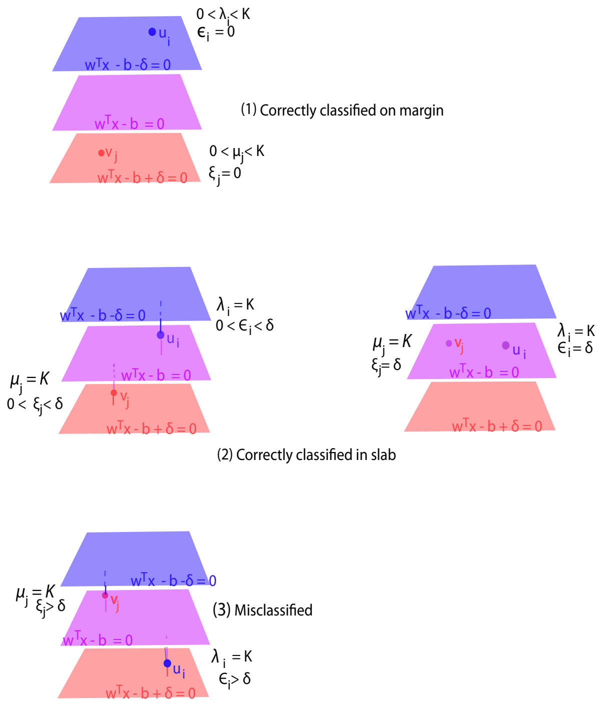


Fig. 18.5 Figure (1) illustrates the case of u_i contained in the margin and occurs when $\epsilon_i = 0$. Figure (1) also illustrates the case of v_j contained in the margin when $\xi_j = 0$. The left illustration of Figure (2) is when u_i is inside the margin yet still on the correct side of the separating hyperplane $w^\top x - b = 0$. Similarly, v_j is inside the margin on the correct side of the separating hyperplane. The right illustration depicts u_i and v_j on the separating hyperplane. Figure (3) illustrations a misclassification of u_i and v_j.

role so we denote the sets of indices associated with them by

$$I_\lambda = \{i \in \{1, \dots, p\} \mid 0 < \lambda_i < K\}$$
$$I_\mu = \{j \in \{1, \dots, q\} \mid 0 < \mu_j < K\}.$$

We denote their cardinalities by $numsvl_1 = |I_\lambda|$ and $numsvm_1 = |I_\mu|$. Support vectors u_i such that $\lambda_i = K$ and support vectors v_j such that $\mu_j = K$ are *support vectors of type 2*. Those support vectors u_i such that $\lambda_i = 0$ and those support vectors v_j such that $\mu_j = 0$ are called *exceptional support vectors*.

The vectors u_i for which $\lambda_i = K$ and the vectors v_j for which $\mu_j = K$ are said to *fail the margin*. The sets of indices associated with the vectors

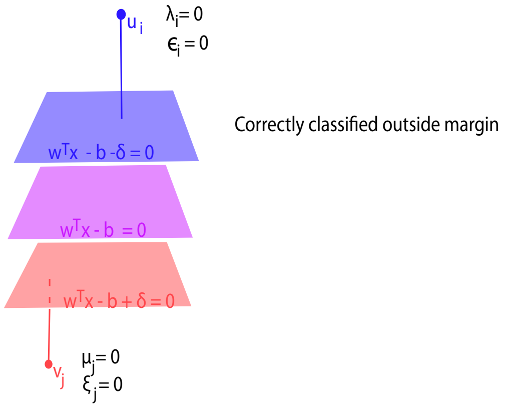


Fig. 18.6 When $\lambda_i = 0$, u_i is correctly classified outside the blue margin. When $\mu_j = 0$, v_j is correctly classified outside the red margin.

failing the margin are denoted by

$$\begin{aligned}K_\lambda &= \{i \in \{1, \ldots, p\} \mid \lambda_i = K\}\\ K_\mu &= \{j \in \{1, \ldots, q\} \mid \mu_j = K\}.\end{aligned}$$

We denote their cardinalities by $p_f = |K_\lambda|$ and $q_f = |K_\mu|$.

Vectors u_i such that $\lambda_i > 0$ and vectors v_j such that $\mu_j > 0$ are said to *have margin at most* δ. The sets of indices associated with these vectors are denoted by

$$\begin{aligned}I_{\lambda>0} &= \{i \in \{1, \ldots, p\} \mid \lambda_i > 0\}\\ I_{\mu>0} &= \{j \in \{1, \ldots, q\} \mid \mu_j > 0\}.\end{aligned}$$

We denote their cardinalities by $p_m = |I_{\lambda>0}|$ and $q_m = |I_{\mu>0}|$.

Obviously, $I_{\lambda>0} = I_\lambda \cup K_\lambda$ and $I_{\mu>0} = I_\mu \cup K_\mu$, so $p_f \leq p_m$ and $q_f \leq q_m$. *Intuitively a blue point that fails the margin is on the wrong side of the blue margin and a red point that fails the margin is on the wrong side of the red margin*. The points in $I_{\lambda>0}$ not in K_λ are on the blue margin and the points in $I_{\mu>0}$ not in K_μ are on the red margin. There are $p - p_m$ points u_i classified correctly on the blue side and outside the δ-slab and there are $q - q_m$ points v_j classified correctly on the red side and outside the δ-slab.

It is easy to show that we have the following bounds on K:

$$\max\left\{\frac{1}{2p_m}, \frac{1}{2q_m}\right\} \leq K \leq \min\left\{\frac{1}{2p_f}, \frac{1}{2q_f}\right\}.$$

These inequalities restrict the choice of K quite heavily.

It will also be useful to understand how points are classified in terms of ϵ_i (or ξ_j).

(1) If $\epsilon_i > 0$, then by complementary slackness $\lambda_i = K$, so the ith equation is active and by (2) above,

$$w^\top u_i - b - \delta = -\epsilon_i.$$

Since $\epsilon_i > 0$, the point u_i is within the open half space bounded by the blue margin hyperplane $H_{w,b+\delta}$ and containing the separating hyperplane $H_{w,b}$; if $\epsilon_i \leq \delta$, then u_i is classified correctly, and if $\epsilon_i > \delta$, then u_i is misclassified.
Similarly, if $\xi_j > 0$, then v_j is within the open half space bounded by the red margin hyperplane $H_{w,b-\delta}$ and containing the separating hyperplane $H_{w,b}$; if $\xi_j \leq \delta$, then v_j is classified correctly, and if $\xi_j > \delta$, then v_j is misclassified.

(2) If $\epsilon_i = 0$, then the point u_i is correctly classified. If $\lambda_i = 0$, then by (3) above, u_i is in the closed half space on the blue side bounded by the blue margin hyperplane $H_{w,b+\delta}$. If $\lambda_i > 0$, then by (1) and (2) above, the point u_i is on the blue margin.
Similarly, if $\xi_j = 0$, then the point v_j is correctly classified. If $\mu_j = 0$, then v_j is in the closed half space on the red side bounded by the red margin hyperplane $H_{w,b-\delta}$, and if $\mu_j > 0$, then the point v_j is on the red margin.

It shown in Section 18.2 how the dual program is solved using ADMM from Section 16.6. If the primal problem is solvable, this yields solutions for λ and μ.

If the optimal value is 0, then $\gamma = 0$ and $X \begin{pmatrix} \lambda \\ \mu \end{pmatrix} = 0$, so in this case it is not possible to determine w. However, if the optimal value is > 0, then once a solution for λ and μ is obtained, by $(*_w)$, we have

$$\begin{aligned} \gamma &= \frac{1}{2}\left(\begin{pmatrix} \lambda^\top & \mu^\top \end{pmatrix} X^\top X \begin{pmatrix} \lambda \\ \mu \end{pmatrix} \right)^{1/2} \\ w &= \frac{1}{2\gamma}\left(\sum_{i=1}^{p} \lambda_i u_i - \sum_{j=1}^{q} \mu_j v_j \right), \end{aligned}$$

so we get

$$w = \frac{\sum_{i=1}^{p} \lambda_i u_i - \sum_{j=1}^{q} \mu_j v_j}{\left(\begin{pmatrix} \lambda^\top & \mu^\top \end{pmatrix} X^\top X \begin{pmatrix} \lambda \\ \mu \end{pmatrix} \right)^{1/2}},$$

which is the result of making $\sum_{i=1}^{p} \lambda_i u_i - \sum_{j=1}^{q} \mu_j v_j$ a unit vector, since

$$X = \begin{pmatrix} -u_1 & \cdots & -u_p & v_1 & \cdots & v_q \end{pmatrix}.$$

It remains to find b and δ, which are not given by the dual program and for this we use the complementary slackness conditions.

The equations

$$\sum_{i=1}^{p} \lambda_i = \sum_{j=1}^{q} \mu_j = \frac{1}{2}$$

imply that there is some i_0 such that $\lambda_{i_0} > 0$ and some j_0 such that $\mu_{j_0} > 0$, but a priori, nothing prevents the situation where $\lambda_i = K$ for all nonzero λ_i or $\mu_j = K$ for all nonzero μ_j. If this happens, we can rerun the optimization method with a larger value of K. If the following mild hypothesis holds, then b and δ can be found.

Standard Margin Hypothesis for (SVM$_{s1}$). There is some index i_0 such that $0 < \lambda_{i_0} < K$ and there is some index j_0 such that $0 < \mu_{j_0} < K$. This means that some u_{i_0} is a support vector of type 1 on the blue margin, and some v_{j_0} is a support of type 1 on the red margin.

If the **Standard Margin Hypothesis** for (SVM$_{s1}$) holds, then $\epsilon_{i_0} = 0$ and $\mu_{j_0} = 0$, and then we have the active equations

$$w^\top u_{i_0} - b = \delta \quad \text{and} \quad -w^\top v_{j_0} + b = \delta,$$

and we obtain the values of b and δ as

$$\begin{aligned} b &= \frac{1}{2}(w^\top u_{i_0} + w^\top v_{j_0}) \\ \delta &= \frac{1}{2}(w^\top u_{i_0} - w^\top v_{j_0}). \end{aligned}$$

Due to numerical instability, when writing a computer program it is preferable to compute the lists of indices I_λ and I_μ given by

$$\begin{aligned} I_\lambda &= \{i \in \{1, \ldots, p\} \mid 0 < \lambda_i < K\} \\ I_\mu &= \{j \in \{1, \ldots, q\} \mid 0 < \mu_j < K\}. \end{aligned}$$

Then it is easy to see that we can compute b and δ using the following averaging formulae:

$$b = w^\top \left(\Big(\sum_{i \in I_\lambda} u_i \Big) / |I_\lambda| + \Big(\sum_{j \in I_\mu} v_j \Big) / |I_\mu| \right) /2$$
$$\delta = w^\top \left(\Big(\sum_{i \in I_\lambda} u_i \Big) / |I_\lambda| - \Big(\sum_{j \in I_\mu} v_j \Big) / |I_\mu| \right) /2.$$

As we said earlier, the hypotheses of Theorem 14.5(2) hold, so *if* the primal problem (SVM$_{s1}$) has an optimal solution with $w \neq 0$, *then* the dual problem has a solution too, and the duality gap is zero. Therefore, for optimal solutions we have

$$L(w, \epsilon, \xi, b, \delta, \lambda, \mu, \alpha, \beta, \gamma) = G(\lambda, \mu, \alpha, \beta, \gamma),$$

which means that

$$-\delta + K\Big(\sum_{i=1}^p \epsilon_i + \sum_{j=1}^q \xi_j \Big) = - \left(\begin{pmatrix} \lambda^\top & \mu^\top \end{pmatrix} X^\top X \begin{pmatrix} \lambda \\ \mu \end{pmatrix} \right)^{1/2},$$

so we get

$$\delta = K\Big(\sum_{i=1}^p \epsilon_i + \sum_{j=1}^q \xi_j \Big) + \left(\begin{pmatrix} \lambda^\top & \mu^\top \end{pmatrix} X^\top X \begin{pmatrix} \lambda \\ \mu \end{pmatrix} \right)^{1/2}.$$

Therefore, we confirm that $\delta \geq 0$.

It is important to note that the objective function of the dual program

$$-G(\lambda, \mu) = \left(\begin{pmatrix} \lambda^\top & \mu^\top \end{pmatrix} X^\top X \begin{pmatrix} \lambda \\ \mu \end{pmatrix} \right)^{1/2}$$

only involves the inner products of the u_i and the v_j through the matrix $X^\top X$, and similarly, the equation of the optimal hyperplane can be written as

$$\sum_{i=1}^p \lambda_i u_i^\top x - \sum_{j=1}^q \mu_j v_j^\top x - \left(\begin{pmatrix} \lambda^\top & \mu^\top \end{pmatrix} X^\top X \begin{pmatrix} \lambda \\ \mu \end{pmatrix} \right)^{1/2} b = 0,$$

an expression that only involves inner products of x with the u_i and the v_j and inner products of the u_i and the v_j.

As explained at the beginning of this chapter, this is a key fact that allows a generalization of the support vector machine using the method of *kernels*. We can define the following "kernelized" version of Problem (SVM$_{s1}$):

Soft margin kernel SVM (SVM_{s1}):

$$\begin{aligned}
&\text{minimize} \quad -\delta + K\left(\sum_{i=1}^{p}\epsilon_i + \sum_{j=1}^{q}\xi_j\right)\\
&\text{subject to}\\
&\qquad \langle w, \varphi(u_i)\rangle - b \geq \delta - \epsilon_i, \quad \epsilon_i \geq 0 \qquad i = 1, \ldots, p\\
&\qquad -\langle w, \varphi(v_j)\rangle + b \geq \delta - \xi_j, \quad \xi_j \geq 0 \qquad j = 1, \ldots, q\\
&\qquad \langle w, w\rangle \leq 1.
\end{aligned}$$

Tracing through the computation that led us to the dual program with u_i replaced by $\varphi(u_i)$ and v_j replaced by $\varphi(v_j)$, we find the following version of the dual program:

Dual of Soft margin kernel SVM (SVM_{s1}):

$$\begin{aligned}
&\text{minimize} \quad \begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix} \mathbf{K} \begin{pmatrix}\lambda\\ \mu\end{pmatrix}\\
&\text{subject to}\\
&\qquad \sum_{i=1}^{p}\lambda_i - \sum_{j=1}^{q}\mu_j = 0\\
&\qquad \sum_{i=1}^{p}\lambda_i + \sum_{j=1}^{q}\mu_j = 1\\
&\qquad 0 \leq \lambda_i \leq K, \quad i = 1, \ldots, p\\
&\qquad 0 \leq \mu_j \leq K, \quad j = 1, \ldots, q,
\end{aligned}$$

where $\mathbf{K}$ is the $\ell \times \ell$ kernel symmetric matrix (with $\ell = p + q$) given by

$$\mathbf{K}_{ij} = \begin{cases} \kappa(u_i, u_j) & 1 \leq i \leq p,\ 1 \leq j \leq q\\ -\kappa(u_i, v_{j-p}) & 1 \leq i \leq p,\ p+1 \leq j \leq p+q\\ -\kappa(v_{i-p}, u_j) & p+1 \leq i \leq p+q,\ 1 \leq j \leq p\\ \kappa(v_{i-p}, v_{j-q}) & p+1 \leq i \leq p+q,\ p+1 \leq j \leq p+q. \end{cases}$$

We also find that

$$w = \frac{\displaystyle\sum_{i=1}^{p}\lambda_i\varphi(u_i) - \sum_{j=1}^{q}\mu_j\varphi(v_j)}{\left(\begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix} K \begin{pmatrix}\lambda\\ \mu\end{pmatrix}\right)^{1/2}}.$$

Under the Standard Margin Hypothesis, there is some index i_0 such that $0 < \lambda_{i_0} < K$ and there is some index j_0 such that $0 < \mu_{j_0} < K$, and we obtain the value of b and δ as

$$\begin{aligned} b &= \frac{1}{2}(\langle w, \varphi(u_{i_0}\rangle + \langle w, \varphi(v_{j_0})\rangle) \\ \delta &= \frac{1}{2}(\langle w, \varphi(u_{i_0})\rangle - \langle w, \varphi(v_{j_0})\rangle). \end{aligned}$$

Using the above value for w, we obtain

$$b = \frac{\sum_{i=1}^p \lambda_i(\kappa(u_i, u_{i_0}) + \kappa(u_i, v_{j_0})) - \sum_{j=1}^q \mu_j(\kappa(v_j, u_{i_0}) + \kappa(v_j, v_{j_0}))}{2\left(\begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix} K \begin{pmatrix}\lambda \\ \mu\end{pmatrix}\right)^{1/2}}.$$

It follows that the classification function

$$f(x) = \operatorname{sgn}(\langle w, \varphi(x)\rangle - b)$$

is given by

$$\begin{aligned} f(x) = \operatorname{sgn}\Bigg(&\sum_{i=1}^p \lambda_i(2\kappa(u_i, x) - \kappa(u_i, u_{i_0}) - \kappa(u_i, v_{j_0})) \\ &- \sum_{j=1}^q \mu_j(2\kappa(v_j, x) - \kappa(v_j, u_{i_0}) - \kappa(v_j, v_{j_0}))\Bigg), \end{aligned}$$

which is solely expressed in terms of the kernel κ.

Kernel methods for SVM are discussed in Schölkopf and Smola [Schölkopf and Smola (2002)] and Shawe–Taylor and Christianini [Shawe-Taylor and Cristianini (2004)].

18.2 Solving SVM (SVM$_{s1}$) Using ADMM

In order to solve (SVM$_{s1}$) using ADMM we need to write the matrix corresponding to the constraints in equational form,

$$\begin{aligned} \sum_{i=1}^p \lambda_i - \sum_{j=1}^q \mu_j &= 0 \\ \sum_{i=1}^p \lambda_i + \sum_{j=1}^q \mu_j &= 1 \\ \lambda_i + \alpha_i = K, &\quad i = 1, \ldots, p \\ \mu_j + \beta_j = K, &\quad j = 1, \ldots, q. \end{aligned}$$

This is the $(p+q+2) \times 2(p+q)$ matrix A given by

$$A = \begin{pmatrix} \mathbf{1}_p^\top & -\mathbf{1}_q^\top & 0_p^\top & 0_q^\top \\ \mathbf{1}_p^\top & \mathbf{1}_q^\top & 0_p^\top & 0_q^\top \\ I_p & 0_{p,q} & I_p & 0_{p,q} \\ 0_{q,p} & I_q & 0_{q,p} & I_q \end{pmatrix}.$$

We leave it as an exercise to prove that A has rank $p+q+2$. The right-hand side is

$$c = \begin{pmatrix} 0 \\ 1 \\ K\mathbf{1}_{p+q} \end{pmatrix}.$$

The symmetric positive semidefinite $(p+q) \times (p+q)$ matrix P defining the quadratic functional is

$$P = 2X^\top X, \quad \text{with} \quad X = \begin{pmatrix} -u_1 \cdots -u_p \; v_1 \cdots v_q \end{pmatrix},$$

and

$$q = 0_{p+q}.$$

Since there are $2(p+q)$ Lagrange multipliers $(\lambda, \mu, \alpha, \beta)$, the $(p+q) \times (p+q)$ matrix $X^\top X$ must be augmented with zero's to make it a $2(p+q) \times 2(p+q)$ matrix P_a given by

$$P_a = \begin{pmatrix} X^\top X & 0_{p+q,p+q} \\ 0_{p+q,p+q} & 0_{p+q,p+q} \end{pmatrix},$$

and similarly q is augmented with zeros as the vector $q_a = 0_{2(p+q)}$.

Since the constraint $w^\top w \leq 1$ causes troubles, we trade it for a different objective function in which $-\delta$ is replaced by $(1/2)\left\|w\right\|_2^2$. This way we are left with purely affine constraints. In the next section we discuss a generalization of Problem (SVM$_{h2}$) obtained by adding a linear regularizing term.

18.3 Soft Margin Support Vector Machines; (SVM$_{s2}$)

In this section we consider the generalization of Problem (SVM$_{h2}$) where we minimize $(1/2)w^\top w$ by adding the "regularizing term" $K\Big(\sum_{i=1}^p \epsilon_i + \sum_{j=1}^q \xi_j,\Big)$ for some $K > 0$. Recall that the margin δ is given by $\delta = 1/\left\|w\right\|$.

Soft margin SVM (SVM$_{s2}$):

$$\begin{aligned}
&\text{minimize} \quad \frac{1}{2} w^\top w + K \left(\epsilon^\top \ \xi^\top\right) \mathbf{1}_{p+q} \\
&\text{subject to} \\
&\qquad w^\top u_i - b \geq 1 - \epsilon_i, \quad \epsilon_i \geq 0 \qquad i = 1, \ldots, p \\
&\qquad -w^\top v_j + b \geq 1 - \xi_j, \quad \xi_j \geq 0 \qquad j = 1, \ldots, q.
\end{aligned}$$

This is the classical problem discussed in all books on machine learning or pattern analysis, for instance Vapnik [Vapnik (1998)], Bishop [Bishop (2006)], and Shawe–Taylor and Christianini [Shawe-Taylor and Cristianini (2004)]. The trivial solution where all variables are 0 is ruled out because of the presence of the 1 in the inequalities, but it is not clear that if (w, b, ϵ, ξ) is an optimal solution, then $w \neq 0$.

We prove that if the primal problem has an optimal solution (w, ϵ, ξ, b) with $w \neq 0$, then w is determined by any optimal solution (λ, μ) of the dual. We also prove that there is some i for which $\lambda_i > 0$ and some j for which $\mu_j > 0$. Under a mild hypothesis that we call the **Standard Margin Hypothesis**, b can be found.

Note that this framework is still somewhat sensitive to outliers because the penalty for misclassification is linear in ϵ and ξ.

First we write the constraints in matrix form. The $2(p+q) \times (n+p+q+1)$ matrix C is written in block form as

$$C = \begin{pmatrix} X^\top & -I_{p+q} & \begin{matrix} \mathbf{1}_p \\ -\mathbf{1}_q \end{matrix} \\ 0_{p+q,n} & -I_{p+q} & 0_{p+q} \end{pmatrix},$$

where X is the $n \times (p+q)$ matrix

$$X = \begin{pmatrix} -u_1 & \cdots & -u_p & v_1 & \cdots & v_q \end{pmatrix},$$

and the constraints are expressed by

$$\begin{pmatrix} X^\top & -I_{p+q} & \begin{matrix} \mathbf{1}_p \\ -\mathbf{1}_q \end{matrix} \\ 0_{p+q,n} & -I_{p+q} & 0_{p+q} \end{pmatrix} \begin{pmatrix} w \\ \epsilon \\ \xi \\ b \end{pmatrix} \leq \begin{pmatrix} -\mathbf{1}_{p+q} \\ 0_{p+q} \end{pmatrix}.$$

The objective function $J(w, \epsilon, \xi, b)$ is given by

$$J(w, \epsilon, \xi, b) = \frac{1}{2} w^\top w + K \left(\epsilon^\top \ \xi^\top\right) \mathbf{1}_{p+q}.$$

The Lagrangian $L(w, \epsilon, \xi, b, \lambda, \mu, \alpha, \beta)$ with $\lambda, \alpha \in \mathbb{R}^p_+$ and with $\mu, \beta \in \mathbb{R}^q_+$ is given by

$$\begin{aligned} L(w, \epsilon, \xi, b, \lambda, \mu, \alpha, \beta) &= \frac{1}{2} w^\top w + K \left(\epsilon^\top \; \xi^\top\right) \mathbf{1}_{p+q} \\ &\quad + \left(w^\top \; \left(\epsilon^\top \; \xi^\top\right) \; b\right) C^\top \begin{pmatrix} \lambda \\ \mu \\ \alpha \\ \beta \end{pmatrix} + \left(\mathbf{1}^\top_{p+q} \; 0^\top_{p+q}\right) \begin{pmatrix} \lambda \\ \mu \\ \alpha \\ \beta \end{pmatrix}. \end{aligned}$$

Since

$$\left(w^\top \; \left(\epsilon^\top \; \xi^\top\right) \; b\right) C^\top \begin{pmatrix} \lambda \\ \mu \\ \alpha \\ \beta \end{pmatrix} = \left(w^\top \; \left(\epsilon^\top \; \xi^\top\right) \; b\right) \begin{pmatrix} X & 0_{n,p+q} \\ -I_{p+q} & -I_{p+q} \\ \mathbf{1}^\top_p \; -\mathbf{1}^\top_q & 0^\top_{p+q} \end{pmatrix} \begin{pmatrix} \lambda \\ \mu \\ \alpha \\ \beta \end{pmatrix},$$

we get

$$\begin{aligned} \left(w^\top \; \left(\epsilon^\top \; \xi^\top\right) \; b\right) C^\top \begin{pmatrix} \lambda \\ \mu \\ \alpha \\ \beta \end{pmatrix} &= \left(w^\top \; \left(\epsilon^\top \; \xi^\top\right) \; b\right) \begin{pmatrix} X \begin{pmatrix} \lambda \\ \mu \end{pmatrix} \\ -\begin{pmatrix} \lambda + \alpha \\ \mu + \beta \end{pmatrix} \\ \mathbf{1}^\top_p \lambda - \mathbf{1}^\top_q \mu \end{pmatrix} \\ &= w^\top X \begin{pmatrix} \lambda \\ \mu \end{pmatrix} - \epsilon^\top(\lambda + \alpha) - \xi^\top(\mu + \beta) + b(\mathbf{1}^\top_p \lambda - \mathbf{1}^\top_q \mu), \end{aligned}$$

and since

$$\left(\mathbf{1}^\top_{p+q} \; 0^\top_{p+q}\right) \begin{pmatrix} \lambda \\ \mu \\ \alpha \\ \beta \end{pmatrix} = \mathbf{1}^\top_{p+q} \begin{pmatrix} \lambda \\ \mu \end{pmatrix} = \left(\lambda^\top \; \mu^\top\right) \mathbf{1}_{p+q},$$

the Lagrangian can be rewritten as

$$\begin{aligned} L(w, \epsilon, \xi, b, \lambda, \mu, \alpha, \beta) &= \frac{1}{2} w^\top w + w^\top X \begin{pmatrix} \lambda \\ \mu \end{pmatrix} + \epsilon^\top (K\mathbf{1}_p - (\lambda + \alpha)) \\ &\quad + \xi^\top (K\mathbf{1}_q - (\mu + \beta)) + b(\mathbf{1}^\top_p \lambda - \mathbf{1}^\top_q \mu) + \left(\lambda^\top \; \mu^\top\right) \mathbf{1}_{p+q}. \end{aligned}$$

To find the dual function $G(\lambda, \mu, \alpha, \beta)$ we minimize $L(w, \epsilon, \xi, b, \lambda, \mu, \alpha, \beta)$ with respect to w, ϵ, ξ and b. Since the Lagrangian is convex and $(w, \epsilon, \xi, b) \in \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^q \times \mathbb{R}$, a convex open set, by Theorem 4.5, the Lagrangian has a minimum in (w, ϵ, ξ, b) iff $\nabla L_{w,\epsilon,\xi,b} = 0$, so we compute its gradient with respect to w, ϵ, ξ and b, and we get

$$\nabla L_{w,\epsilon,\xi,b} = \begin{pmatrix} w + X \begin{pmatrix} \lambda \\ \mu \end{pmatrix} \\ K\mathbf{1}_p - (\lambda + \alpha) \\ K\mathbf{1}_q - (\mu + \beta) \\ \mathbf{1}_p^\top \lambda - \mathbf{1}_q^\top \mu \end{pmatrix}.$$

By setting $\nabla L_{w,\epsilon,\xi,b} = 0$ we get the equations

$$w = -X \begin{pmatrix} \lambda \\ \mu \end{pmatrix} \tag{$*_w$}$$

$$\begin{aligned} \lambda + \alpha &= K\mathbf{1}_p \\ \mu + \beta &= K\mathbf{1}_q \\ \mathbf{1}_p^\top \lambda &= \mathbf{1}_q^\top \mu. \end{aligned}$$

The first and the fourth equation are identical to the Equations $(*_1)$ and $(*_2)$ that we obtained in Example 14.10. Since $\lambda, \mu, \alpha, \beta \geq 0$, the second and the third equation are equivalent to the box constraints

$$0 \leq \lambda_i, \mu_j \leq K, \quad i = 1, \ldots, p,\ j = 1, \ldots, q.$$

Using the equations that we just derived, after simplifications we get

$$G(\lambda, \mu, \alpha, \beta) = -\frac{1}{2} \begin{pmatrix} \lambda^\top & \mu^\top \end{pmatrix} X^\top X \begin{pmatrix} \lambda \\ \mu \end{pmatrix} + \begin{pmatrix} \lambda^\top & \mu^\top \end{pmatrix} \mathbf{1}_{p+q},$$

which is independent of α and β and is identical to the dual function obtained in $(*_4)$ of Example 14.10. To be perfectly rigorous,

$$G(\lambda, \mu) = -\frac{1}{2} \begin{pmatrix} \lambda^\top & \mu^\top \end{pmatrix} X^\top X \begin{pmatrix} \lambda \\ \mu \end{pmatrix} + \begin{pmatrix} \lambda^\top & \mu^\top \end{pmatrix} \mathbf{1}_{p+q}$$

$$\text{if} \quad \begin{cases} \sum_{i=1}^p \lambda_i = \sum_{j=1}^q \mu_j \\ 0 \leq \lambda_i \leq K,\ i = 1, \ldots, p \\ 0 \leq \mu_j \leq K,\ j = 1, \ldots, q \end{cases}$$

$$-\infty \quad \text{otherwise.}$$

As in Example 14.10, the the dual program can be formulated as

$$\text{maximize} \quad -\frac{1}{2} \begin{pmatrix} \lambda^\top & \mu^\top \end{pmatrix} X^\top X \begin{pmatrix} \lambda \\ \mu \end{pmatrix} + \begin{pmatrix} \lambda^\top & \mu^\top \end{pmatrix} \mathbf{1}_{p+q}$$

subject to

$$\begin{aligned} &\sum_{i=1}^p \lambda_i - \sum_{j=1}^q \mu_j = 0 \\ &0 \leq \lambda_i \leq K, \quad i = 1, \ldots, p \\ &0 \leq \mu_j \leq K, \quad j = 1, \ldots, q, \end{aligned}$$

or equivalently

Dual of Soft margin SVM (SVM_{s2}):

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2}\begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix} X^\top X \begin{pmatrix}\lambda \\ \mu\end{pmatrix} - \begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix} \mathbf{1}_{p+q} \\ &\text{subject to} \\ &\qquad \sum_{i=1}^p \lambda_i - \sum_{j=1}^q \mu_j = 0 \\ &\qquad 0 \leq \lambda_i \leq K, \quad i = 1, \ldots, p \\ &\qquad 0 \leq \mu_j \leq K, \quad j = 1, \ldots, q. \end{aligned}$$

If (w, ϵ, ξ, b) is an optimal solution of Problem (SVM_{s2}), then the complementary slackness conditions yield a classification of the points u_i and v_j in terms of the values of λ and μ. Indeed, we have $\epsilon_i \alpha_i = 0$ for $i = 1, \ldots, p$ and $\xi_j \beta_j = 0$ for $j = 1, \ldots, q$. Also, if $\lambda_i > 0$, then corresponding constraint is active, and similarly if $\mu_j > 0$. Since $\lambda_i + \alpha_i = K$, it follows that $\epsilon_i \alpha_i = 0$ iff $\epsilon_i(K - \lambda_i) = 0$, and since $\mu_j + \beta_j = K$, we have $\xi_j \beta_j = 0$ iff $\xi_j(K - \mu_j) = 0$. Thus if $\epsilon_i > 0$, then $\lambda_i = K$, and if $\xi_j > 0$, then $\mu_j = K$. Consequently, if $\lambda_i < K$, then $\epsilon_i = 0$ and u_i is correctly classified, and similarly if $\mu_j < K$, then $\xi_j = 0$ and v_j is correctly classified.

We have a classification of the points u_i and v_j in terms of λ and μ obtained from the classification given in Section 18.1 by replacing δ with 1. Since it is so similar, it is omitted. Let us simply recall that the vectors u_i on the blue margin and the vectors v_j on the red margin are called *support vectors*; these are the vectors u_i for which $w^\top u_i - b - 1 = 0$ (which implies $\epsilon_i = 0$), and the vectors v_j for which $w^\top v_j - b + 1 = 0$ (which implies $\xi_j = 0$). Those support vectors u_i such that $\lambda_i = 0$ and those support vectors such that $\mu_j = 0$ are called *exceptional support vectors*.

We also have the following classification of the points u_i and v_j terms of ϵ_i (or ξ_j) obtained by replacing δ with 1.

(1) If $\epsilon_i > 0$, then by complementary slackness $\lambda_i = K$, so the ith equation is active and by (2) above,

$$w^\top u_i - b - 1 = -\epsilon_i.$$

Since $\epsilon_i > 0$, the point u_i is within the open half space bounded by the blue margin hyperplane $H_{w,b+1}$ and containing the separating hyperplane $H_{w,b}$; if $\epsilon_i \leq 1$, then u_i is classified correctly, and if $\epsilon_i > 1$, then u_i is misclassified.

Similarly, if $\xi_j > 0$, then v_j is within the open half space bounded by the red margin hyperplane $H_{w,b-1}$ and containing the separating hyperplane $H_{w,b}$; if $\xi_j \leq 1$, then v_j is classified correctly, and if $\xi_j > 1$, then v_j is misclassified.

(2) If $\epsilon_i = 0$, then the point u_i is correctly classified. If $\lambda_i = 0$, then by (3) above, u_i is in the closed half space on the blue side bounded by the blue margin hyperplane $H_{w,b+\eta}$. If $\lambda_i > 0$, then by (1) and (2) above, the point u_i is on the blue margin.
Similarly, if $\xi_j = 0$, then the point v_j is correctly classified. If $\mu_j = 0$, then v_j is in the closed half space on the red side bounded by the red margin hyperplane $H_{w,b-\eta}$, and if $\mu_j > 0$, then the point v_j is on the red margin. See Figure 18.5 (3).

Vectors u_i for which $\lambda_i = K$ and vectors v_j such that $\xi_j = K$ are said to *fail the margin*.

It is shown in Section 18.4 how the dual program is solved using ADMM from Section 16.6. If the primal problem is solvable, this yields solutions for λ and μ.

Remark. The hard margin Problem (SVM$_{h2}$) corresponds to the special case of Problem (SVM$_{s2}$) in which $\epsilon = 0$, $\xi = 0$, and $K = +\infty$. Indeed, in Problem (SVM$_{h2}$) the terms involving ϵ and ξ are missing from the Lagrangian and the effect is that the box constraints are missing; we simply have $\lambda_i \geq 0$ and $\mu_j \geq 0$.

We can use the dual program to solve the primal. Once $\lambda \geq 0, \mu \geq 0$ have been found, w is given by

$$w = -X \begin{pmatrix} \lambda \\ \mu \end{pmatrix} = \sum_{i=1}^{p} \lambda_i u_i - \sum_{j=1}^{q} \mu_j v_j.$$

To find b we use the complementary slackness conditions.

If the primal has a solution $w \neq 0$, then the equation

$$w = \sum_{i=1}^{p} \lambda_i u_i - \sum_{j=1}^{q} \mu_j v_j$$

implies that either there is some index i_0 such that $\lambda_{i_0} > 0$ or there is some index j_0 such that $\mu_{j_0} > 0$. The constraint

$$\sum_{i=1}^{p} \lambda_i - \sum_{j=1}^{q} \mu_j = 0$$

implies that there is some index i_0 such that $\lambda_{i_0} > 0$ and there is some index j_0 such that $\mu_{j_0} > 0$. However, a priori, nothing prevents the situation where $\lambda_i = K$ for all nonzero λ_i or $\mu_j = K$ for all nonzero μ_j. If this happens, we can rerun the optimization method with a larger value of K. Observe that the equation

$$\sum_{i=1}^{p} \lambda_i - \sum_{j=1}^{q} \mu_j = 0$$

implies that if there is some index i_0 such that $0 < \lambda_{i_0} < K$, then there is some index j_0 such that $0 < \mu_{j_0} < K$, and vice-versa. If the following mild hypothesis holds, then b can be found.

Standard Margin Hypothesis for (SVM$_{s2}$). There is some index i_0 such that $0 < \lambda_{i_0} < K$ and there is some index j_0 such that $0 < \mu_{j_0} < K$. This means that some u_{i_0} is a support vector of type 1 on the blue margin, and some v_{j_0} is a support vector of type 1 on the red margin.

If the **Standard Margin Hypothesis** for (SVM$_{s2}$) holds, then $\epsilon_{i_0} = 0$ and $\mu_{j_0} = 0$, and then we have the active equations

$$w^\top u_{i_0} - b = 1 \quad \text{and} \quad -w^\top v_{j_0} + b = 1,$$

and we obtain

$$b = \frac{1}{2}(w^\top u_{i_0} + w^\top v_{j_0}).$$

Due to numerical instability, when writing a computer program it is preferable to compute the lists of indices I_λ and I_μ given by

$$\begin{aligned} I_\lambda &= \{i \in \{1, \ldots, p\} \mid 0 < \lambda_i < K\} \\ I_\mu &= \{j \in \{1, \ldots, q\} \mid 0 < \mu_j < K\}. \end{aligned}$$

Then it is easy to see that we can compute b using the following averaging formula

$$b = w^\top \left(\Big(\sum_{i \in I_\lambda} u_i \Big) / |I_\lambda| + \Big(\sum_{j \in I_\mu} v_j \Big) / |I_\mu| \right) /2.$$

Recall that $\delta = 1/\left\|w\right\|$.

Remark: There is a cheap version of Problem (SVM$_{s2}$) which consists in dropping the term $(1/2)w^\top w$ from the objective function:

Soft margin classifier (SVM$_{s2l}$):

$$\begin{aligned}
&\text{minimize} \quad \sum_{i=1}^{p} \epsilon_i + \sum_{j=1}^{q} \xi_j \\
&\text{subject to} \\
&\qquad\qquad w^\top u_i - b \geq 1 - \epsilon_i, \quad \epsilon_i \geq 0 \qquad i = 1, \ldots, p \\
&\qquad\qquad - w^\top v_j + b \geq 1 - \xi_j, \quad \xi_j \geq 0 \qquad j = 1, \ldots, q.
\end{aligned}$$

The above program is a linear program that minimizes the number of misclassified points but does not care about enforcing a minimum margin. An example of its use is given in Boyd and Vandenberghe; see [Boyd and Vandenberghe (2004)], Section 8.6.1.

The "kernelized" version of Problem (SVM$_{s2}$) is the following:

Soft margin kernel SVM (SVM$_{s2}$):

$$\begin{aligned}
&\text{minimize} \quad \frac{1}{2}\langle w, w\rangle + K \begin{pmatrix} \epsilon^\top & \xi^\top \end{pmatrix} \mathbf{1}_{p+q} \\
&\text{subject to} \\
&\qquad\qquad \langle w, \varphi(u_i)\rangle - b \geq 1 - \epsilon_i, \quad \epsilon_i \geq 0 \qquad i = 1, \ldots, p \\
&\qquad\qquad - \langle w, \varphi(v_j)\rangle + b \geq 1 - \xi_j, \quad \xi_j \geq 0 \qquad j = 1, \ldots, q.
\end{aligned}$$

Redoing the computation of the dual function, we find that the dual program is given by

Dual of Soft margin kernel SVM (SVM$_{s2}$):

$$\begin{aligned}
&\text{minimize} \quad \frac{1}{2} \begin{pmatrix} \lambda^\top & \mu^\top \end{pmatrix} \mathbf{K} \begin{pmatrix} \lambda \\ \mu \end{pmatrix} - \begin{pmatrix} \lambda^\top & \mu^\top \end{pmatrix} \mathbf{1}_{p+q} \\
&\text{subject to} \\
&\qquad\qquad \sum_{i=1}^{p} \lambda_i - \sum_{j=1}^{q} \mu_j = 0 \\
&\qquad\qquad 0 \leq \lambda_i \leq K, \quad i = 1, \ldots, p \\
&\qquad\qquad 0 \leq \mu_j \leq K, \quad j = 1, \ldots, q,
\end{aligned}$$

where $\mathbf{K}$ is the $\ell \times \ell$ kernel symmetric matrix (with $\ell = p + q$) given at the end of Section 18.1. We also find that

$$w = \sum_{i=1}^{p} \lambda_i \varphi(u_i) - \sum_{j=1}^{q} \mu_j \varphi(v_j),$$

so

$$b = \frac{1}{2}\Bigg(\sum_{i=1}^{p} \lambda_i(\kappa(u_i, u_{i_0}) + \kappa(u_i, v_{j_0})) - \sum_{j=1}^{q} \mu_j(\kappa(v_j, u_{i_0}) + \kappa(v_j, v_{j_0}))\Bigg),$$

and the classification function

$$f(x) = \mathrm{sgn}(\langle w, \varphi(x)\rangle - b)$$

is given by

$$\begin{aligned} f(x) = \mathrm{sgn}\Bigg(&\sum_{i=1}^{p} \lambda_i(2\kappa(u_i, x) - \kappa(u_i, u_{i_0}) - \kappa(u_i, v_{j_0})) \\ &- \sum_{j=1}^{q} \mu_j(2\kappa(v_j, x) - \kappa(v_j, u_{i_0}) - \kappa(v_j, v_{j_0}))\Bigg). \end{aligned}$$

18.4 Solving SVM (SVM$_{s2}$) Using ADMM

In order to solve (SVM$_{s2}$) using ADMM we need to write the matrix corresponding to the constraints in equational form,

$$\begin{aligned} \sum_{i=1}^{p} \lambda_i - \sum_{j=1}^{q} \mu_j &= 0 \\ \lambda_i + \alpha_i &= K, \quad i = 1, \ldots, p \\ \mu_j + \beta_j &= K, \quad j = 1, \ldots, q. \end{aligned}$$

This is the $(p + q + 1) \times 2(p + q)$ matrix A given by

$$A = \begin{pmatrix} \mathbf{1}_p^\top & -\mathbf{1}_q^\top & 0_p^\top & 0_q^\top \\ I_p & 0_{p,q} & I_p & 0_{p,q} \\ 0_{q,p} & I_q & 0_{q,p} & I_q \end{pmatrix}.$$

We leave it as an exercise to prove that A has rank $p+q+1$. The right-hand side is

$$c = \begin{pmatrix} 0 \\ K\mathbf{1}_{p+q} \end{pmatrix}.$$

The symmetric positive semidefinite $(p+q) \times (p+q)$ matrix P defining the quadratic functional is

$$P = X^\top X, \quad \text{with} \quad X = \begin{pmatrix} -u_1 & \cdots & -u_p & v_1 & \cdots & v_q \end{pmatrix},$$

and

$$q = -\mathbf{1}_{p+q}.$$

Since there are $2(p+q)$ Lagrange multipliers $(\lambda, \mu, \alpha, \beta)$, the $(p+q)\times(p+q)$ matrix $X^\top X$ must be augmented with zero's to make it a $2(p+q)\times 2(p+q)$ matrix P_a given by

$$P_a = \begin{pmatrix} X^\top X & 0_{p+q,p+q} \\ 0_{p+q,p+q} & 0_{p+q,p+q} \end{pmatrix},$$

and similarly q is augmented with zeros as the vector

$$q_a = \begin{pmatrix} -\mathbf{1}_{p+q} \\ 0_{p+q}. \end{pmatrix}$$

18.5 Soft Margin Support Vector Machines; (SVM$_{s2'}$)

In this section we consider a generalization of Problem (SVM$_{s2}$) for a version of the soft margin SVM coming from Problem (SVM$_{h2}$) by adding an extra degree of freedom, namely instead of the margin $\delta = 1/\|w\|$, we use the margin $\delta = \eta/\|w\|$ where η is some positive constant that we wish to maximize. To do so, we add a term $-K_m\eta$ to the objective function $(1/2)w^\top w$ as well as the "regularizing term" $K_s\left(\sum_{i=1}^p \epsilon_i + \sum_{j=1}^q \xi_j\right)$ whose purpose is to make ϵ and ξ sparse, where $K_m > 0$ (m refers to margin) and $K_s > 0$ (s refers to sparse) are fixed constants that can be adjusted to determine the influence of η and the regularizing term.

Soft margin SVM (SVM$_{s2'}$):

$$\begin{aligned}
&\text{minimize} \quad \frac{1}{2}w^\top w - K_m\eta + K_s\left(\epsilon^\top \;\; \xi^\top\right)\mathbf{1}_{p+q}\\
&\text{subject to}\\
&\qquad w^\top u_i - b \geq \eta - \epsilon_i, \quad \epsilon_i \geq 0 \qquad i = 1, \ldots, p\\
&\qquad -w^\top v_j + b \geq \eta - \xi_j, \quad \xi_j \geq 0 \qquad j = 1, \ldots, q\\
&\qquad \eta \geq 0.
\end{aligned}$$

This version of the SVM problem was first discussed in Schölkopf, Smola, Williamson, and Bartlett [Schölkopf *et al.* (2000)] under the name of *ν-SVC* (or *ν-SVM*), and also used in Schölkopf, Platt, Shawe–Taylor, and Smola [Schölkopf *et al.* (2001)]. The ν-SVC method is also presented in Schölkopf and Smola [Schölkopf and Smola (2002)] (which contains much more). The difference between the ν-SVC method and the method presented in Section 18.3, sometimes called the C-SVM method, was thoroughly investigated by Chan and Lin [Chang and Chih-Jen (2001)].

For this problem it is no longer clear that if $(w, \eta, b, \epsilon, \xi)$ is an optimal solution, then $w \neq 0$ and $\eta > 0$. In fact, if the sets of points are not linearly separable and if K_s is chosen too big, Problem (SVM$_{s2'}$) may fail to have an optimal solution.

We show that in order for the problem to have a solution we must pick K_m and K_s so that

$$K_m \leq \min\{2pK_s, 2qK_s\}.$$

If we define ν by

$$\nu = \frac{K_m}{(p+q)K_s},$$

then $K_m \leq \min\{2pK_s, 2qK_s\}$ is equivalent to

$$\nu \leq \min\left\{\frac{2p}{p+q}, \frac{2q}{p+q}\right\} \leq 1.$$

The reason for introducing ν is that $\nu(p+q)/2$ can be interpreted as the maximum number of points failing to achieve the margin $\delta = \eta / \|w\|$. We will show later that if the points u_i and v_j are not separable, then we must pick ν so that $\nu \geq 2/(p+q)$ for the method to have a solution for which $w \neq 0$ and $\eta > 0$.

The objective function of our problem is convex and the constraints are affine. Consequently, by Theorem 14.5(2) *if* the Primal Problem (SVM$_{s2'}$) has an optimal solution, *then* the dual problem has a solution too, and the duality gap is zero. This does not immediately imply that an optimal solution of the dual yields an optimal solution of the primal because the hypotheses of Theorem 14.5(1) fail to hold.

We show that if the primal problem has an optimal solution $(w, \eta, \epsilon, \xi, b)$ with $w \neq 0$, then any optimal solution of the dual problem determines λ and μ, which in turn determine w *via* the equation

$$w = -X \begin{pmatrix} \lambda \\ \mu \end{pmatrix} = \sum_{i=1}^{p} \lambda_i u_i - \sum_{j=1}^{q} \mu_j v_j, \tag{$*_w$}$$

and $\eta \geq 0$.

It remains to determine b, η, ϵ and ξ. The solution of the dual does not determine b, η, ϵ, ξ directly, and we are not aware of necessary and sufficient conditions that ensure that they can be determined. The best we can do is to use the KKT conditions.

The simplest sufficient condition is what we call the

Standard Margin Hypothesis for (SVM$_{s2'}$): There is some i_0 such that $0 < \lambda_{i_0} < K_s$, and there is some μ_{j_0} such that $0 < \mu_{j_0} < K_s$. This means that there is some support vector u_{i_0} of type 1 *and* there is some support vector v_{j_0} of type 1.

In this case, then by complementary slackness, it can be shown that $\epsilon_{i_0} = 0$, $\xi_{i_0} = 0$, and the corresponding inequalities are active, that is we have the equations

$$w^\top u_{i_0} - b = \eta, \quad -w^\top v_{j_0} + b = \eta,$$

so we can solve for b and η. Then since by complementary slackness, if $\epsilon_i > 0$, then $\lambda_i = K_s$ and if $\xi_j > 0$, then $\mu_j = K_s$, all inequalities corresponding to such $\epsilon_i > 0$ and $\mu_j > 0$ are active, and we can solve for ϵ_i and ξ_j.

The linear constraints are given by the $(2(p+q)+1) \times (n+p+q+2)$ matrix given in block form by

$$C = \begin{pmatrix} X^\top & -I_{p+q} & \begin{matrix} \mathbf{1}_p \\ -\mathbf{1}_q \end{matrix} & \mathbf{1}_{p+q} \\ 0_{p+q,n} & -I_{p+q} & 0_{p+q} & 0_{p+q} \\ 0_n^\top & 0_{p+q}^\top & 0 & -1 \end{pmatrix},$$

where X is the $n \times (p+q)$ matrix

$$X = \begin{pmatrix} -u_1 & \cdots & -u_p & v_1 & \cdots & v_q \end{pmatrix},$$

and the linear constraints are expressed by

$$\begin{pmatrix} X^\top & -I_{p+q} & \begin{matrix} \mathbf{1}_p \\ -\mathbf{1}_q \end{matrix} & \mathbf{1}_{p+q} \\ 0_{p+q,n} & -I_{p+q} & 0_{p+q} & 0_{p+q} \\ 0_n^\top & 0_{p+q}^\top & 0 & -1 \end{pmatrix} \begin{pmatrix} w \\ \epsilon \\ \xi \\ b \\ \eta \end{pmatrix} \leq \begin{pmatrix} 0_{p+q} \\ 0_{p+q} \\ 0 \end{pmatrix}.$$

The objective function is given by

$$J(w, \epsilon, \xi, b, \eta) = \frac{1}{2} w^\top w - K_m \eta + K_s \begin{pmatrix} \epsilon^\top & \xi^\top \end{pmatrix} \mathbf{1}_{p+q}.$$

The Lagrangian $L(w, \epsilon, \xi, b, \eta, \lambda, \mu, \alpha, \beta, \gamma)$ with $\lambda, \alpha \in \mathbb{R}_+^p$, $\mu, \beta \in \mathbb{R}_+^q$, and $\gamma \in \mathbb{R}_+$ is given by

$$\begin{aligned} L(w, \epsilon, \xi, b, \eta, \lambda, \mu, \alpha, \beta, \gamma) &= \frac{1}{2} w^\top w - K_m \eta + K_s \begin{pmatrix} \epsilon^\top & \xi^\top \end{pmatrix} \mathbf{1}_{p+q} \\ &\quad + \begin{pmatrix} w^\top & \begin{pmatrix} \epsilon^\top & \xi^\top \end{pmatrix} & b & \eta \end{pmatrix} C^\top \begin{pmatrix} \lambda \\ \mu \\ \alpha \\ \beta \\ \gamma \end{pmatrix}. \end{aligned}$$

Since

$$\begin{pmatrix} w^\top & \begin{pmatrix} \epsilon^\top & \xi^\top \end{pmatrix} & b & \eta \end{pmatrix} C^\top \begin{pmatrix} \lambda \\ \mu \\ \alpha \\ \beta \\ \gamma \end{pmatrix} = w^\top X \begin{pmatrix} \lambda \\ \mu \end{pmatrix} - \epsilon^\top(\lambda + \alpha) - \xi^\top(\mu + \beta)$$
$$+ b(\mathbf{1}_p^\top \lambda - \mathbf{1}_q^\top \mu) + \eta(\mathbf{1}_p^\top \lambda + \mathbf{1}_q^\top \mu) - \gamma\eta,$$

the Lagrangian can be written as

$$\begin{aligned} L(w, \epsilon, \xi, b, \eta, \lambda, \mu, \alpha, \beta, \gamma) &= \frac{1}{2} w^\top w + K_s(\epsilon^\top \mathbf{1}_p + \xi^\top \mathbf{1}_q) + w^\top X \begin{pmatrix} \lambda \\ \mu \end{pmatrix} \\ - K_m\eta - \epsilon^\top(\lambda + \alpha) &- \xi^\top(\mu + \beta) + b(\mathbf{1}_p^\top \lambda - \mathbf{1}_q^\top \mu) + \eta(\mathbf{1}_p^\top \lambda + \mathbf{1}_q^\top \mu) - \gamma\eta \\ &= \frac{1}{2} w^\top w + w^\top X \begin{pmatrix} \lambda \\ \mu \end{pmatrix} + (\mathbf{1}_p^\top \lambda + \mathbf{1}_q^\top \mu - K_m - \gamma)\eta \\ &+ \epsilon^\top(K_s \mathbf{1}_p - (\lambda + \alpha)) + \xi^\top(K_s \mathbf{1}_q - (\mu + \beta)) + b(\mathbf{1}_p^\top \lambda - \mathbf{1}_q^\top \mu). \end{aligned}$$

To find the dual function $G(\lambda, \mu, \alpha, \beta, \gamma)$ we minimize $L(w, \epsilon, \xi, b, \eta, \lambda, \mu, \alpha, \beta, \gamma)$ with respect to w, ϵ, ξ, b, and η. Since the Lagrangian is convex and $(w, \epsilon, \xi, b, \eta) \in \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^q \times \mathbb{R} \times \mathbb{R}$, a convex open set, by Theorem 4.5, the Lagrangian has a minimum in $(w, \epsilon, \xi, b, \eta)$ iff $\nabla L_{w,\epsilon,\xi,b,\eta} = 0$, so we compute its gradient with respect to $w, \epsilon, \xi, b, \eta$, and we get

$$\nabla L_{w,\epsilon,\xi,b,\eta} = \begin{pmatrix} X \begin{pmatrix} \lambda \\ \mu \end{pmatrix} + w \\ K_s \mathbf{1}_p - (\lambda + \alpha) \\ K_s \mathbf{1}_q - (\mu + \beta) \\ \mathbf{1}_p^\top \lambda - \mathbf{1}_q^\top \mu \\ \mathbf{1}_p^\top \lambda + \mathbf{1}_q^\top \mu - K_m - \gamma \end{pmatrix}.$$

By setting $\nabla L_{w,\epsilon,\xi,b,\eta} = 0$ we get the equations

$$w = -X \begin{pmatrix} \lambda \\ \mu \end{pmatrix} \qquad (*_w)$$

$$\begin{aligned} \lambda + \alpha &= K_s \mathbf{1}_p \\ \mu + \beta &= K_s \mathbf{1}_q \\ \mathbf{1}_p^\top \lambda &= \mathbf{1}_q^\top \mu, \end{aligned}$$

and

$$\mathbf{1}_p^\top \lambda + \mathbf{1}_q^\top \mu = K_m + \gamma. \qquad (*_\gamma)$$

The second and third equations are equivalent to the box constraints

$$0 \leq \lambda_i, \mu_j \leq K_s, \quad i = 1, \ldots, p,\ j = 1, \ldots, q,$$

and since $\gamma \geq 0$ equation $(*_\gamma)$ is equivalent to

$$\mathbf{1}_p^\top \lambda + \mathbf{1}_q^\top \mu \geq K_m.$$

Plugging back w from $(*_w)$ into the Lagrangian, after simplifications we get

$$\begin{aligned} G(\lambda, \mu, \alpha, \beta) &= \frac{1}{2} \begin{pmatrix} \lambda^\top & \mu^\top \end{pmatrix} X^\top X \begin{pmatrix} \lambda \\ \mu \end{pmatrix} - \begin{pmatrix} \lambda^\top & \mu^\top \end{pmatrix} X^\top X \begin{pmatrix} \lambda \\ \mu \end{pmatrix} \\ &= -\frac{1}{2} \begin{pmatrix} \lambda^\top & \mu^\top \end{pmatrix} X^\top X \begin{pmatrix} \lambda \\ \mu \end{pmatrix}, \end{aligned}$$

so the dual function is independent of α, β and is given by

$$G(\lambda, \mu) = -\frac{1}{2} \begin{pmatrix} \lambda^\top & \mu^\top \end{pmatrix} X^\top X \begin{pmatrix} \lambda \\ \mu \end{pmatrix}.$$

The dual program is given by

$$\begin{aligned} &\text{maximize} \quad -\frac{1}{2} \begin{pmatrix} \lambda^\top & \mu^\top \end{pmatrix} X^\top X \begin{pmatrix} \lambda \\ \mu \end{pmatrix} \\ &\text{subject to} \\ &\qquad \sum_{i=1}^p \lambda_i - \sum_{j=1}^q \mu_j = 0 \\ &\qquad \sum_{i=1}^p \lambda_i + \sum_{j=1}^q \mu_j \geq K_m \\ &\qquad 0 \leq \lambda_i \leq K_s, \quad i = 1, \ldots, p \\ &\qquad 0 \leq \mu_j \leq K_s, \quad j = 1, \ldots, q. \end{aligned}$$

Finally, the dual program is equivalent to the following minimization program:

Dual of Soft margin SVM ($\mathrm{SVM}_{s2'}$):

$$\begin{aligned}
&\text{minimize} \quad \frac{1}{2}\begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix} X^\top X \begin{pmatrix}\lambda \\ \mu\end{pmatrix}\\
&\text{subject to}\\
&\qquad \sum_{i=1}^{p}\lambda_i - \sum_{j=1}^{q}\mu_j = 0\\
&\qquad \sum_{i=1}^{p}\lambda_i + \sum_{j=1}^{q}\mu_j \geq K_m\\
&\qquad 0 \leq \lambda_i \leq K_s, \quad i = 1, \ldots, p\\
&\qquad 0 \leq \mu_j \leq K_s, \quad j = 1, \ldots, q.
\end{aligned}$$

If $(w, \eta, \epsilon, \xi, b)$ is an optimal solution of Problem ($\mathrm{SVM}_{s2'}$) with $w \neq 0$ and $\eta \neq 0$, then the complementary slackness conditions yield a classification of the points u_i and v_j in terms of the values of λ and μ. Indeed, we have $\epsilon_i\alpha_i = 0$ for $i = 1, \ldots, p$ and $\xi_j\beta_j = 0$ for $j = 1, \ldots, q$. Also, if $\lambda_i > 0$, then the corresponding constraint is active, and similarly if $\mu_j > 0$. Since $\lambda_i + \alpha_i = K_s$, it follows that $\epsilon_i\alpha_i = 0$ iff $\epsilon_i(K_s - \lambda_i) = 0$, and since $\mu_j + \beta_j = K_s$, we have $\xi_j\beta_j = 0$ iff $\xi_j(K_s - \mu_j) = 0$. Thus if $\epsilon_i > 0$, then $\lambda_i = K_s$, and if $\xi_j > 0$, then $\mu_j = K_s$. Consequently, if $\lambda_i < K_s$, then $\epsilon_i = 0$ and u_i is correctly classified, and similarly if $\mu_j < K_s$, then $\xi_j = 0$ and v_j is correctly classified.

We have the following classification which is basically the classification given in Section 18.1 obtained by replacing δ with η (recall that $\eta > 0$ and $\delta = \eta / \|w\|$) .

(1) If $0 < \lambda_i < K_s$, then $\epsilon_i = 0$ and the i-th inequality is active, so

$$w^\top u_i - b - \eta = 0.$$

This means that u_i is on the blue margin (the hyperplane $H_{w,b+\eta}$ of equation $w^\top x = b + \eta$) and is classified correctly.
Similarly, if $0 < \mu_j < K_s$, then $\xi_j = 0$ and

$$w^\top v_j - b + \eta = 0,$$

so v_j is on the red margin (the hyperplane $H_{w,b-\eta}$ of equation $w^\top x = b - \eta$) and is classified correctly.

(2) If $\lambda_i = K_s$, then the i-th inequality is active, so

$$w^\top u_i - b - \eta = -\epsilon_i.$$

If $\epsilon_i = 0$, then the point u_i is on the blue margin. If $\epsilon_i > 0$, then u_i is within the open half space bounded by the blue margin hyperplane $H_{w,b+\eta}$ and containing the separating hyperplane $H_{w,b}$; if $\epsilon_i \leq \eta$, then u_i is classified correctly, and if $\epsilon_i > \eta$, then u_i is misclassified (u_i lies on the wrong side of the separating hyperplane, the red side).
Similarly, if $\mu_j = K_s$, then

$$w^\top v_j - b + \eta = \xi_j.$$

If $\xi_j = 0$, then the point v_j is on the red margin. If $\xi_j > 0$, then v_j is within the open half space bounded by the red margin hyperplane $H_{w,b-\eta}$ and containing the separating hyperplane $H_{w,b}$; if $\xi_j \leq \eta$, then v_j is classified correctly, and if $\xi_j > \eta$, then v_j is misclassified (v_j lies on the wrong side of the separating hyperplane, the blue side).

(3) If $\lambda_i = 0$, then $\epsilon_i = 0$ and the i-th inequality may or may not be active, so

$$w^\top u_i - b - \eta \geq 0.$$

Thus u_i is in the closed half space on the blue side bounded by the blue margin hyperplane $H_{w,b+\eta}$ (of course, classified correctly).
Similarly, if $\mu_j = 0$, then

$$w^\top v_j - b + \eta \leq 0$$

and v_j is in the closed half space on the red side bounded by the red margin hyperplane $H_{w,b-\eta}$ (of course, classified correctly).

Definition 18.2. The vectors u_i on the blue margin $H_{w,b+\eta}$ and the vectors v_j on the red margin $H_{w,b-\eta}$ are called *support vectors*. Support vectors correspond to vectors u_i for which $w^\top u_i - b - \eta = 0$ (which implies $\epsilon_i = 0$), and vectors v_j for which $w^\top v_j - b + \eta = 0$ (which implies $\xi_j = 0$). Support vectors u_i such that $0 < \lambda_i < K_s$ and support vectors v_j such that $0 < \mu_j < K_s$ are *support vectors of type 1*. Support vectors of type 1 play a special role so we denote the sets of indices associated with them by

$$\begin{aligned}
I_\lambda &= \{i \in \{1, \ldots, p\} \mid 0 < \lambda_i < K_s\} \\
I_\mu &= \{j \in \{1, \ldots, q\} \mid 0 < \mu_j < K_s\}.
\end{aligned}$$

We denote their cardinalities by $\mathit{numsvl}_1 = |I_\lambda|$ and $\mathit{numsvm}_1 = |I_\mu|$. Support vectors u_i such that $\lambda_i = K_s$ and support vectors v_j such that $\mu_j = K_s$ are *support vectors of type 2*. Those support vectors u_i such that $\lambda_i = 0$ and those support vectors v_j such that $\mu_j = 0$ are called *exceptional support vectors*.

The vectors u_i for which $\lambda_i = K_s$ and the vectors v_j for which $\mu_j = K_s$ are said to *fail the margin*. The sets of indices associated with the vectors failing the margin are denoted by

$$K_\lambda = \{i \in \{1, \ldots, p\} \mid \lambda_i = K_s\}$$
$$K_\mu = \{j \in \{1, \ldots, q\} \mid \mu_j = K_s\}.$$

We denote their cardinalities by $p_f = |K_\lambda|$ and $q_f = |K_\mu|$.

It will also be useful to understand how points are classified in terms of ϵ_i (or ξ_j).

(1) If $\epsilon_i > 0$, then by complementary slackness $\lambda_i = K_s$, so the ith equation is active and by (2) above,
$$w^\top u_i - b - \eta = -\epsilon_i.$$
Since $\epsilon_i > 0$, the point u_i is strictly within the open half space bounded by the blue margin hyperplane $H_{w,b+\eta}$ and containing the separating hyperplane $H_{w,b}$ (excluding the blue hyperplane $H_{w,b+\eta}$); if $\epsilon_i \leq \eta$, then u_i is classified correctly, and if $\epsilon_i > \eta$, then u_i is misclassified.
Similarly, if $\xi_j > 0$, then v_j is strictly within the open half space bounded by the red margin hyperplane $H_{w,b-\eta}$ and containing the separating hyperplane $H_{w,b}$ (excluding the red hyperplane $H_{w,b-\eta}$); if $\xi_j \leq \eta$, then v_j is classified correctly, and if $\xi_j > \eta$, then v_j is misclassified.

(2) If $\epsilon_i = 0$, then the point u_i is correctly classified. If $\lambda_i = 0$, then by (3) above, u_i is in the closed half space on the blue side bounded by the blue margin hyperplane $H_{w,b+\eta}$. If $\lambda_i > 0$, then by (1) and (2) above, the point u_i is on the blue margin.
Similarly, if $\xi_j = 0$, then the point v_j is correctly classified. If $\mu_j = 0$, then v_j is in the closed half space on the red side bounded by the red margin hyperplane $H_{w,b-\eta}$, and if $\mu_j > 0$, then the point v_j is on the red margin.

Also observe that if $\lambda_i > 0$, then u_i is in the closed half space bounded by the blue hyperplane $H_{w,b+\eta}$ and containing the separating hyperplane $H_{w,b}$ (including the blue hyperplane $H_{w,b+\eta}$).

Similarly, if $\mu_j > 0$, then v_j is in the closed half space bounded by the red hyperplane $H_{w,b+\eta}$ and containing the separating hyperplane $H_{w,b}$ (including the red hyperplane $H_{w,b+\eta}$).

Definition 18.3. Vectors u_i such that $\lambda_i > 0$ and vectors v_j such that $\mu_j > 0$ are said to *have margin at most* δ. The sets of indices associated with these vectors are denoted by

$$I_{\lambda>0} = \{i \in \{1, \ldots, p\} \mid \lambda_i > 0\}$$
$$I_{\mu>0} = \{j \in \{1, \ldots, q\} \mid \mu_j > 0\}.$$

We denote their cardinalities by $p_m = |I_{\lambda>0}|$ and $q_m = |I_{\mu>0}|$.

Vectors u_i such that $\epsilon_i > 0$ and vectors v_j such that $\xi_j > 0$ are said to *strictly fail the margin*. The corresponding sets of indices are denoted by

$$E_\lambda = \{i \in \{1, \ldots, p\} \mid \epsilon_i > 0\}$$
$$E_\mu = \{j \in \{1, \ldots, q\} \mid \xi_j > 0\}.$$

We write $p_{sf} = |E_\lambda|$ and $q_{sf} = |E_\mu|$.

We have the inclusions $E_\lambda \subseteq K_\lambda$ and $E_\mu \subseteq K_\mu$. The difference between the first sets and the second sets is that the second sets may contain support vectors such that $\lambda_i = K_s$ and $\epsilon_i = 0$, or $\mu_j = K_s$ and $\xi_j = 0$. We also have the equations $I_\lambda \cup K_\lambda = I_{\lambda>0}$ and $I_\mu \cup K_\mu = I_{\mu>0}$, and the inequalities $p_{sf} \le p_f \le p_m$ and $q_{sf} \le q_f \le q_m$.

It is shown in Section 18.8 how the dual program is solved using ADMM from Section 16.6. If the primal problem is solvable, this yields solutions for λ and μ. Once a solution for λ and μ is obtained, we have

$$w = -X \begin{pmatrix} \lambda \\ \mu \end{pmatrix} = \sum_{i=1}^{p} \lambda_i u_i - \sum_{j=1}^{q} \mu_j v_j.$$

As we said earlier, the hypotheses of Theorem 14.5(2) hold, so *if* the primal problem $(\mathrm{SVM}_{s2'})$ has an optimal solution with $w \neq 0$, *then* the dual problem has a solution too, and the duality gap is zero. Therefore, for optimal solutions we have

$$L(w, \epsilon, \xi, b, \eta, \lambda, \mu, \alpha, \beta, \gamma) = G(\lambda, \mu, \alpha, \beta, \gamma),$$

which means that

$$\frac{1}{2} w^\top w - K_m \eta + K_s \left(\sum_{i=1}^{p} \epsilon_i + \sum_{j=1}^{q} \xi_j \right) = -\frac{1}{2} \begin{pmatrix} \lambda^\top & \mu^\top \end{pmatrix} X^\top X \begin{pmatrix} \lambda \\ \mu \end{pmatrix},$$

and since

$$w = -X \begin{pmatrix} \lambda \\ \mu \end{pmatrix},$$

we get

$$\frac{1}{2} \begin{pmatrix} \lambda^\top & \mu^\top \end{pmatrix} X^\top X \begin{pmatrix} \lambda \\ \mu \end{pmatrix} - K_m \eta + K_s \left(\sum_{i=1}^{p} \epsilon_i + \sum_{j=1}^{q} \xi_j \right)$$
$$= -\frac{1}{2} \begin{pmatrix} \lambda^\top & \mu^\top \end{pmatrix} X^\top X \begin{pmatrix} \lambda \\ \mu \end{pmatrix},$$

which yields

$$\eta = \frac{K_s}{K_m}\left(\sum_{i=1}^{p}\epsilon_i + \sum_{j=1}^{q}\xi_j\right) + \frac{1}{K_m}\begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix} X^\top X \begin{pmatrix}\lambda \\ \mu\end{pmatrix}. \qquad (*)$$

Therefore, we confirm that $\eta \geq 0$.

Remarks: Since we proved that if the Primal Problem (SVM$_{s2'}$) has an optimal solution with $w \neq 0$, then $\eta \geq 0$, one might wonder why the constraint $\eta \geq 0$ was included. If we delete this constraint, it is easy to see that the only difference is that instead of the equation

$$\mathbf{1}_p^\top \lambda + \mathbf{1}_q^\top \mu = K_m + \gamma \qquad (*_1)$$

we obtain the equation

$$\mathbf{1}_p^\top \lambda + \mathbf{1}_q^\top \mu = K_m. \qquad (*_2)$$

If $\eta > 0$, then by complementary slackness $\gamma = 0$, in which case $(*_1)$ and $(*_2)$ are equivalent. But if $\eta = 0$, then γ could be strictly positive.

The option to omit the constraint $\eta \geq 0$ in the primal is slightly advantageous because then the dual involves $2(p+q)$ instead of $2(p+q)+1$ Lagrange multipliers, so the constraint matrix is a $(p+q+2) \times 2(p+q)$ matrix instead of a $(p+q+2) \times (2(p+q)+1)$ matrix and the matrix defining the quadratic functional is a $2(p+q) \times 2(p+q)$ matrix instead of a $(2(p+q)+1) \times (2(p+q)+1)$ matrix; see Section 18.8.

Under the **Standard Margin Hypothesis** for (SVM$_{s2'}$), there is some i_0 such that $0 < \lambda_{i_0} < K_s$ and some j_0 such that $0 < \mu_{j_0} < K_s$, and by the complementary slackness conditions $\epsilon_{i_0} = 0$ and $\xi_{j_0} = 0$, so we have the two active constraints

$$w^\top u_{i_0} - b = \eta, \quad -w^\top v_{j_0} + b = \eta,$$

and we can solve for b and η and we get

$$\begin{aligned} b &= \frac{w^\top u_{i_0} + w^\top v_{j_0}}{2} \\ \eta &= \frac{w^\top u_{i_0} - w^\top v_{j_0}}{2} \\ \delta &= \frac{\eta}{\|w\|}. \end{aligned}$$

Due to numerical instability, when writing a computer program it is preferable to compute the lists of indices I_λ and I_μ given by

$$\begin{aligned} I_\lambda &= \{i \in \{1, \ldots, p\} \mid 0 < \lambda_i < K_s\} \\ I_\mu &= \{j \in \{1, \ldots, q\} \mid 0 < \mu_j < K_s\}. \end{aligned}$$

Then it is easy to see that we can compute b and η using the following averaging formulae:

$$b = w^\top \left(\left(\sum_{i \in I_\lambda} u_i \right) / |I_\lambda| + \left(\sum_{j \in I_\mu} v_j \right) / |I_\mu| \right) /2$$

$$\eta = w^\top \left(\left(\sum_{i \in I_\lambda} u_i \right) / |I_\lambda| - \left(\sum_{j \in I_\mu} v_j \right) / |I_\mu| \right) /2.$$

The "kernelized" version of Problem $(\mathrm{SVM}_{s2'})$ is the following:

Soft margin kernel SVM $(\mathrm{SVM}_{s2'})$:

$$\begin{aligned}
&\text{minimize} \quad \frac{1}{2}\langle w, w\rangle - K_m\eta + K_s\left(\epsilon^\top\ \xi^\top\right)\mathbf{1}_{p+q}\\
&\text{subject to}\\
&\qquad \langle w, \varphi(u_i)\rangle - b \geq \eta - \epsilon_i, \quad \epsilon_i \geq 0 \qquad i = 1, \ldots, p\\
&\qquad -\langle w, \varphi(v_j)\rangle + b \geq \eta - \xi_j, \quad \xi_j \geq 0 \qquad j = 1, \ldots, q\\
&\qquad \eta \geq 0.
\end{aligned}$$

Tracing through the derivation of the dual program we obtain

Dual of the Soft margin kernel SVM $(\mathrm{SVM}_{s2'})$:

$$\begin{aligned}
&\text{minimize} \quad \frac{1}{2}\left(\lambda^\top\ \mu^\top\right)\mathbf{K}\begin{pmatrix}\lambda\\ \mu\end{pmatrix}\\
&\text{subject to}\\
&\qquad \sum_{i=1}^p \lambda_i - \sum_{j=1}^q \mu_j = 0\\
&\qquad \sum_{i=1}^p \lambda_i + \sum_{j=1}^q \mu_j \geq K_m\\
&\qquad 0 \leq \lambda_i \leq K_s, \quad i = 1, \ldots, p\\
&\qquad 0 \leq \mu_j \leq K_s, \quad j = 1, \ldots, q,
\end{aligned}$$

where $\mathbf{K}$ is the kernel matrix of Section 18.1.

As in Section 18.3, we obtain

$$w = \sum_{i=1}^p \lambda_i \varphi(u_i) - \sum_{j=1}^q \mu_j \varphi(v_j),$$

so

$$b = \frac{1}{2}\biggl(\sum_{i=1}^{p} \lambda_i(\kappa(u_i, u_{i_0}) + \kappa(u_i, v_{j_0})) - \sum_{j=1}^{q} \mu_j(\kappa(v_j, u_{i_0}) + \kappa(v_j, v_{j_0}))\biggr),$$

and the classification function

$$f(x) = \mathrm{sgn}(\langle w, \varphi(x)\rangle - b)$$

is given by

$$\begin{aligned} f(x) = \mathrm{sgn}\biggl(&\sum_{i=1}^{p} \lambda_i(2\kappa(u_i, x) - \kappa(u_i, u_{i_0}) - \kappa(u_i, v_{j_0})) \\ &- \sum_{j=1}^{q} \mu_j(2\kappa(v_j, x) - \kappa(v_j, u_{i_0}) - \kappa(v_j, v_{j_0}))\biggr). \end{aligned}$$

18.6 Classification of the Data Points in Terms of ν ($\mathbf{SVM}_{s2'}$)

For a finer classification of the points it turns out to be convenient to consider the ratio

$$\nu = \frac{K_m}{(p+q)K_s}.$$

First note that in order for the constraints to be satisfied, some relationship between K_s and K_m must hold. In addition to the constraints

$$0 \leq \lambda_i \leq K_s, \quad 0 \leq \mu_j \leq K_s,$$

we also have the constraints

$$\sum_{i=1}^{p} \lambda_i = \sum_{j=1}^{q} \mu_j$$

$$\sum_{i=1}^{p} \lambda_i + \sum_{j=1}^{q} \mu_j \geq K_m$$

which imply that

$$\sum_{i=1}^{p} \lambda_i \geq \frac{K_m}{2} \quad \text{and} \quad \sum_{j=1}^{q} \mu_j \geq \frac{K_m}{2}. \tag{$\dagger$}$$

Since λ, μ are all nonnegative, if $\lambda_i = K_s$ for all i and if $\mu_j = K_s$ for all j, then

$$\frac{K_m}{2} \leq \sum_{i=1}^{p} \lambda_i \leq pK_s \quad \text{and} \quad \frac{K_m}{2} \leq \sum_{j=1}^{q} \mu_j \leq qK_s,$$

so these constraints are not satisfied unless $K_m \leq \min\{2pK_s, 2qK_s\}$, so we assume that $K_m \leq \min\{2pK_s, 2qK_s\}$. The equations in (†) also imply that there is some i_0 such that $\lambda_{i_0} > 0$ and some j_0 such that $\mu_{j_0} > 0$, and so $p_m \geq 1$ and $q_m \geq 1$.

For a finer classification of the points we find it convenient to define $\nu > 0$ such that

$$\nu = \frac{K_m}{(p+q)K_s},$$

so that the objective function $J(w, \epsilon, \xi, b, \eta)$ is given by

$$J(w, \epsilon, \xi, b, \eta) = \frac{1}{2} w^\top w + (p+q)K_s \left(-\nu\eta + \frac{1}{p+q} \begin{pmatrix} \epsilon^\top & \xi^\top \end{pmatrix} \mathbf{1}_{p+q} \right).$$

Observe that the condition $K_m \leq \min\{2pK_s, 2qK_s\}$ is equivalent to

$$\nu \leq \min\left\{ \frac{2p}{p+q}, \frac{2q}{p+q} \right\} \leq 1.$$

Since we obtain an equivalent problem by rescaling by a common positive factor, theoretically it is convenient to normalize K_s as

$$K_s = \frac{1}{p+q},$$

in which case $K_m = \nu$. This method is called the *ν-support vector machine*. Actually, to program the method, it may be more convenient assume that K_s is arbitrary. This helps in avoiding λ_i and μ_j to become to small when $p + q$ is relatively large.

The equations (†) and the box inequalities

$$0 \leq \lambda_i \leq K_s, \quad 0 \leq \mu_j \leq K_s$$

also imply the following facts:

Proposition 18.1. *If Problem* (SVM$_{s2'}$) *has an optimal solution with $w \neq 0$ and $\eta > 0$, then the following facts hold:*

(1) Let p_f be the number of points u_i such that $\lambda_i = K_s$, and let q_f the number of points v_j such that $\mu_j = K_s$. Then $p_f, q_f \leq \nu(p+q)/2$.
(2) Let p_m be the number of points u_i such that $\lambda_i > 0$, and let q_m the number of points v_j such that $\mu_j > 0$. Then $p_m, q_m \geq \nu(p+q)/2$. We have $p_m \geq 1$ and $q_m \geq 1$.
(3) If $p_f \geq 1$ or $q_f \geq 1$, then $\nu \geq 2/(p+q)$.

Proof. (1) Recall that for an optimal solution with $w \neq 0$ and $\eta > 0$, we have $\gamma = 0$, so by $(*_\gamma)$ we have the equations

$$\sum_{i=1}^{p} \lambda_i = \frac{K_m}{2} \quad \text{and} \quad \sum_{j=1}^{q} \mu_j = \frac{K_m}{2}.$$

The point u_i fails to achieve the margin iff $\lambda_i = K_s = K_m/(\nu(p+q))$, so if there are p_f such points then

$$\frac{K_m}{2} = \sum_{i=1}^{p} \lambda_i \geq \frac{K_m p_f}{\nu(p+q)},$$

so

$$p_f \leq \frac{\nu(p+q)}{2}.$$

A similar reasoning applies if v_j fails to achieve the margin δ with $\sum_{i=1}^{p} \lambda_i$ replaced by $\sum_{j=1}^{q} \mu_j$.

(2) A point u_i has margin at most δ iff $\lambda_i > 0$. If

$$I_{\lambda>0} = \{i \in \{1, \ldots, p\} \mid \lambda_i > 0\} \quad \text{and} \quad p_m = |I_{\lambda>0}|,$$

then

$$\frac{K_m}{2} = \sum_{i=1}^{p} \lambda_i = \sum_{i \in I_{\lambda>0}} \lambda_i,$$

and since $\lambda_i \leq K_s = K_m/(\nu(p+q))$, we have

$$\frac{K_m}{2} = \sum_{i \in I_{\lambda>0}} \lambda_i \leq \frac{K_m p_m}{\nu(p+q)},$$

which yields

$$p_m \geq \frac{\nu(p+q)}{2}.$$

A similar reasoning applies if a point v_j has margin at most δ. We already observed that (†) implies that $p_m \geq 1$ and $q_m \geq 1$.

(3) This follows immediately from (1). ☐

Observe that $p_f = q_f = 0$ means that there are no points in the open slab containing the separating hyperplane, namely, the points u_i and the points v_j are separable. So if the points u_i and the points v_j are not separable, then we must pick ν such that $2/(p+q) \leq \nu \leq \min\{2p/(p+q), 2q/(p+q)\}$ for the method to succeed. Otherwise, the method is trying to produce a solution where $w = 0$ and $\eta = 0$, and it does not converge (γ

is nonzero). Actually, Proposition 18.1 yields more accurate bounds on ν for the method to converge, namely

$$\max\left\{\frac{2p_f}{p+q}, \frac{2q_f}{p+q}\right\} \leq \nu \leq \min\left\{\frac{2p_m}{p+q}, \frac{2q_m}{p+q}\right\}.$$

By a previous remark, $p_f \leq p_m$ and $q_f \leq q_m$, the first inequality being strict if there is some i such that $0 < \lambda_i < K$, and the second inequality being strict if there is some j such that $0 < \mu_j < K$. This will be the case under the **Standard Margin Hypothesis**.

Observe that a small value of ν keeps p_f and q_f small, which is achieved if the δ-slab is narrow (to avoid having points on the wrong sides of the margin hyperplanes). A large value of ν allows p_m and q_m to be fairly large, which is achieved if the δ-slab is wide. Thus the smaller ν is, the narrower the δ-slab is, and the larger ν is, the wider the δ-slab is. This is the opposite of the behavior that we witnessed in ν-regression (see Section 20.1).

18.7 Existence of Support Vectors for (SVM$_{s2'}$)

We now consider the issue of the existence of support vectors. We will show that in the "generic case" there is always some blue support vector and some red support vector. The term generic has to do with the choice of ν and will be explained below.

Given any real numbers u, v, x, y, if $\max\{u, v\} < \min\{x, y\}$, then $u < x$ and $v < y$. This is because $u, v \leq \max\{u, v\} < \min\{x, y\} \leq x, y$. Consequently, since by Proposition 18.1, $\max\{2p_f/(p+q), 2q_f/(p+q)\} \leq \nu$, if $\nu < \min\{2p/(p+q), 2q/(p+q)\}$, then $p_f < p$ and $q_f < q$, and since $p_{sf} \leq p_f$ and $q_{sf} \leq q_f$, we also have $p_{sf} < p$ and $q_{sf} < q$. This implies that there are constraints corresponding to some $i \notin E_\lambda$ (in which case $\epsilon_i = 0$) and to some $j \notin E_\mu$ (in which case $\xi_j = 0$), of the form

$$\begin{aligned} w^\top u_i - b &\geq \eta & \qquad i &\notin E_\lambda \\ -w^\top v_j + b &\geq \eta & \qquad j &\notin E_\mu. \end{aligned}$$

If $w^\top u_i - b = \eta$ for some $i \notin E_\lambda$ and $-w^\top v_j + b = \eta$ for some $j \notin E_\mu$, then we have a blue support vector and a red support vector. Otherwise, we show how to modify b and η to obtain an optimal solution with a blue support vector and a red support vector.

Proposition 18.2. *For every optimal solution* $(w, b, \eta, \epsilon, \xi)$ *of Problem* (SVM$_{s2'}$) *with* $w \neq 0$ *and* $\eta > 0$, *if*

$$\nu < \min\{2p/(p+q), 2q/(p+q)\}$$

and if either no u_i is a support vector or no v_j is a support vector, then there is another optimal solution (for the same w) with some i_0 such that $\epsilon_{i_0} = 0$ and $w^\top u_{i_0} - b = \eta$, and there is some j_0 such that $\xi_{j_0} = 0$ and $-w^\top v_{j_0} + b = \eta$; in other words, some u_{i_0} and some v_{j_0} is a support vector; in particular, $p_{sf} < p$ and $q_{sf} < q$.

Proof. We just explained that $p_{sf} < p$ and $q_{sf} < q$, so the following constraints hold:

$$\begin{aligned}
w^\top u_i - b &= \eta - \epsilon_i & \epsilon_i &> 0 & i &\in E_\lambda \\
-w^\top v_j + b &= \eta - \xi_j & \xi_j &> 0 & j &\in E_\mu \\
w^\top u_i - b &\geq \eta & & & i &\notin E_\lambda \\
-w^\top v_j + b &\geq \eta & & & j &\notin E_\mu,
\end{aligned}$$

where there is some $i \notin E_\lambda$ and some $j \notin E_\mu$.

If our optimal solution does not have a blue support vector and a red support vector, then either $w^\top u_i - b > \eta$ for all $i \notin E_\lambda$ or $-w^\top v_j + b > \eta$ for all $j \notin E_\mu$.

Case 1. We have

$$\begin{aligned}
w^\top u_i - b &> \eta & i &\notin E_\lambda \\
-w^\top v_j + b &\geq \eta & j &\notin E_\mu.
\end{aligned}$$

There are two subcases.

Case 1a. Assume that there is some $j \notin E_\mu$ such that $-w^\top v_j + b = \eta$. Our strategy is to increase η and b by a small amount θ in such a way that some inequality becomes an equation for some $i \notin E_\lambda$. Geometrically, this amounts to raising the separating hyperplane $H_{w,b}$ and increasing the width of the slab, keeping the red margin hyperplane unchanged. See Figure 18.7.

Let us pick θ such that

$$\theta = (1/2) \min\{w^\top u_i - b - \eta \mid i \notin E_\lambda\}.$$

Our hypotheses imply that $\theta > 0$. We can write

$$\begin{aligned}
w^\top u_i - (b + \theta) &= \eta + \theta - (\epsilon_i + 2\theta) & \epsilon_i &> 0 & i &\in E_\lambda \\
-w^\top v_j + b + \theta &= \eta + \theta - \xi_j & \xi_j &> 0 & j &\in E_\mu \\
w^\top u_i - (b + \theta) &\geq \eta + \theta & & & i &\notin E_\lambda \\
-w^\top v_j + b + \theta &\geq \eta + \theta & & & j &\notin E_\mu.
\end{aligned}$$

By hypothesis

$$-w^\top v_j + b + \theta = \eta + \theta \quad \text{for some } j \notin E_\mu,$$

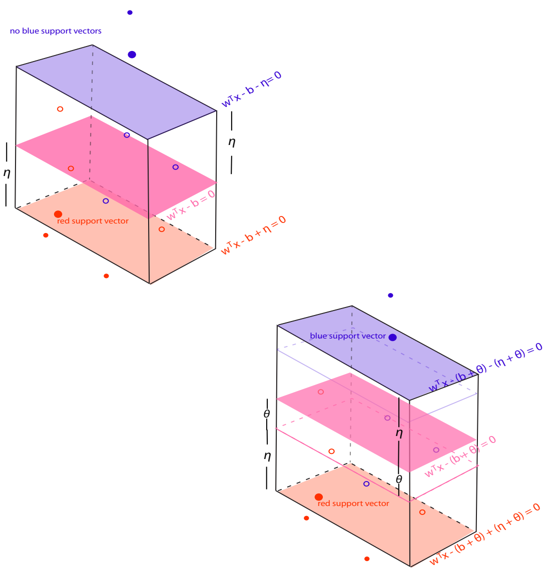


Fig. 18.7 In this illustration points with errors are denoted by open circles. In the original, upper left configuration, there is no blue support vector. By raising the pink separating hyperplane and increasing the margin, we end up with a blue support vector.

and by the choice of θ,

$$w^\top u_i - (b + \theta) = \eta + \theta \quad \text{for some } i \notin E_\lambda.$$

The new value of the objective function is

$$\begin{aligned}\omega(\theta) &= \frac{1}{2} w^\top w - \nu(\eta + \theta) + \frac{1}{p+q}\left(\sum_{i \in E_\lambda} (\epsilon_i + 2\theta) + \sum_{j \in E_\mu} \xi_j\right) \\ &= \frac{1}{2} w^\top w - \nu\eta + \frac{1}{p+q}\left(\sum_{i \in E_\lambda} \epsilon_i + \sum_{j \in E_\mu} \xi_j\right) - \left(\nu - \frac{2p_{sf}}{p+q}\right)\theta.\end{aligned}$$

By Proposition 18.1 we have

$$\max\left\{\frac{2p_f}{p+q}, \frac{2q_f}{p+q}\right\} \leq \nu$$

and $p_{sf} \leq p_f$ and $q_{sf} \leq q_f$, which implies that

$$\nu - \frac{2p_{sf}}{p+q} \geq 0, \tag{$*_1$}$$

and so $\omega(\theta) \leq \omega(0)$. If inequality $(*_1)$ is strict, then this contradicts the optimality of the original solution. Therefore, $\nu = 2p_{sf}/(p+q)$, $\omega(\theta) = \omega(0)$, and $(w, b+\theta, \eta+\theta, \epsilon+2\theta, \xi)$ is an optimal solution such that

$$\begin{aligned} w^\top u_i - (b+\theta) &= \eta + \theta \\ -w^\top v_j + b + \theta &= \eta + \theta \end{aligned}$$

for some $i \notin E_\lambda$ and some $j \notin E_\mu$.

Case 1b. We have $-w^\top v_j + b > \eta$ for all $j \notin E_\mu$. Our strategy is to increase η and the errors by a small θ in such a way that some inequality becomes an equation for some $i \notin E_\lambda$ or for some $j \notin E_\mu$. Geometrically, this corresponds to increasing the width of the slab, keeping the separating hyperplane unchanged. See Figures 18.8 and 18.9. Then we are reduced to *Case 1a* or *Case 2a*.

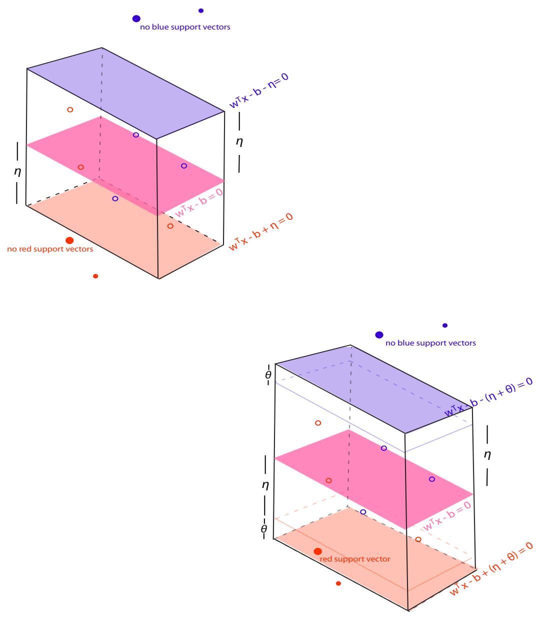


Fig. 18.8 In this illustration points with errors are denoted by open circles. In the original, upper left configuration, there is no blue support vector and no red support vector. By increasing the margin, we end up with a red support vector and reduce to Case 1a.

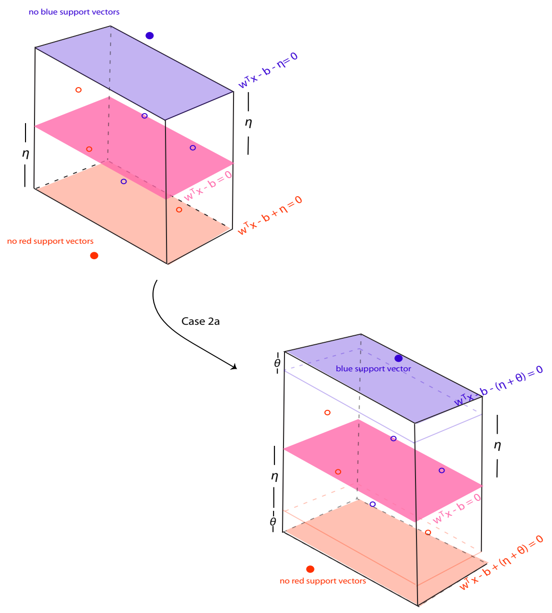


Fig. 18.9 In this illustration points with errors are denoted by open circles. In the original, upper left configuration, there is no blue support vector and no red support vector. By increasing the margin, we end up with a blue support vector and reduce to Case 2a.

We have

$$\begin{aligned}
w^\top u_i - b &= \eta - \epsilon_i & \epsilon_i &> 0 & i &\in E_\lambda\\
-w^\top v_j + b &= \eta - \xi_j & \xi_j &> 0 & j &\in E_\mu\\
w^\top u_i - b &> \eta & & & i &\notin E_\lambda\\
-w^\top v_j + b &> \eta & & & j &\notin E_\mu.
\end{aligned}$$

Let us pick θ such that

$$\theta = \min\{w^\top u_i - b - \eta,\ -w^\top v_j + b - \eta \mid i \notin E_\lambda, j \notin E_\mu\}.$$

Our hypotheses imply that $\theta > 0$. We can write

$$\begin{aligned}
w^\top u_i - b &= \eta + \theta - (\epsilon_i + \theta) & \epsilon_i &> 0 & i &\in E_\lambda\\
-w^\top v_j + b &= \eta + \theta - (\xi_j + \theta) & \xi_j &> 0 & j &\in E_\mu\\
w^\top u_i - b &\geq \eta + \theta & & & i &\notin E_\lambda\\
-w^\top v_j + b &\geq \eta + \theta & & & j &\notin E_\mu,
\end{aligned}$$

and by the choice of θ, either

$$w^\top u_i - b = \eta + \theta \quad \text{for some } i \notin E_\lambda$$

or

$$-w^\top v_j + b = \eta + \theta \quad \text{for some } j \notin E_\mu.$$

The new value of the objective function is

$$\begin{aligned}\omega(\theta) &= \frac{1}{2}w^\top w - \nu(\eta+\theta) + \frac{1}{p+q}\biggl(\sum_{i\in E_\lambda}(\epsilon_i+\theta) + \sum_{j\in E_\mu}(\xi_j+\theta)\biggr)\\ &= \frac{1}{2}w^\top w - \nu\eta + \frac{1}{p+q}\biggl(\sum_{i\in E_\lambda}\epsilon_i + \sum_{j\in E_\mu}\xi_j\biggr) - \biggl(\nu - \frac{p_{sf}+q_{sf}}{p+q}\biggr)\theta.\end{aligned}$$

Since $\max\{2p_f/(p+q), 2q_f/(p+q)\} \leq \nu$ implies that $(p_f+q_f)/(p+q) \leq \nu$ and $p_{sf} \leq p_f$, $q_{sf} \leq q_f$, we have

$$\nu - \frac{p_{sf}+q_{sf}}{p+q} \geq 0, \tag{$*_2$}$$

and so $\omega(\theta) \leq \omega(0)$. If inequality $(*_2)$ is strict, then this contradicts the optimality of the original solution. Therefore, $\nu = (p_{sf}+q_{sf})/(p+q)$, $\omega(\theta) = \omega(0)$ and $(w, b, \eta+\theta, \epsilon+\theta, \xi+\theta)$ is an optimal solution such that either

$$w^\top u_i - b = \eta + \theta \quad \text{for some } i \notin E_\lambda$$

or

$$-w^\top v_j + b = \eta + \theta \quad \text{for some } j \notin E_\mu.$$

We are now reduced to *Case 1a* or *Case 2a*.

Case 2. We have

$$\begin{aligned} w^\top u_i - b &\geq \eta & i &\notin E_\lambda\\ -w^\top v_j + b &> \eta & j &\notin E_\mu.\end{aligned}$$

There are two subcases.

Case 2a. Assume that there is some $i \notin E_\lambda$ such that $w^\top u_i - b = \eta$. Our strategy is to increase η and decrease b by a small amount θ in such a way that some inequality becomes an equation for some $j \notin E_\mu$. Geometrically, this amounts to lowering the separating hyperplane $H_{w,b}$ and increasing the width of the slab, keeping the blue margin hyperplane unchanged. See Figure 18.10.

Let us pick θ such that

$$\theta = (1/2)\min\{-w^\top v_j + b - \eta \mid j \notin E_\mu\}.$$

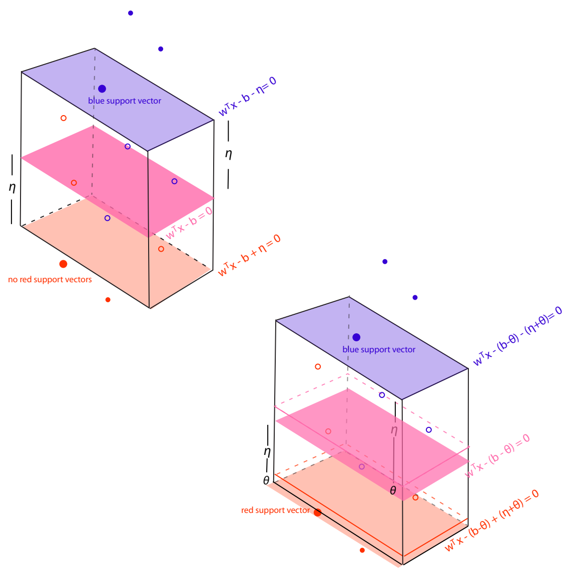


Fig. 18.10 In this illustration points with errors are denoted by open circles. In the original, upper left configuration, there is no red support vector. By lowering the pink separating hyperplane and increasing the margin, we end up with a red support vector.

Our hypotheses imply that $\theta > 0$. We can write

$$\begin{aligned}
w^\top u_i - (b-\theta) &= \eta + \theta - \epsilon_i & \epsilon_i &> 0 & i &\in E_\lambda \\
-w^\top v_j + b - \theta &= \eta + \theta - (\xi_j + 2\theta) & \xi_j &> 0 & j &\in E_\mu \\
w^\top u_i - (b-\theta) &\geq \eta + \theta & & & i &\notin E_\lambda \\
-w^\top v_j + b - \theta &\geq \eta + \theta & & & j &\notin E_\mu .
\end{aligned}$$

By hypothesis

$$w^\top u_i - (b-\theta) = \eta + \theta \quad \text{for some } i \notin E_\lambda,$$

and by the choice of θ,

$$-w^\top v_j + b - \theta = \eta + \theta \quad \text{for some } j \notin E_\mu.$$

The new value of the objective function is

$$\begin{aligned}
\omega(\theta) &= \frac{1}{2} w^\top w - \nu(\eta+\theta) + \frac{1}{p+q}\left(\sum_{i \in E_\lambda} \epsilon_i + \sum_{j \in E_\mu} (\xi_j + 2\theta)\right) \\
&= \frac{1}{2} w^\top w - \nu\eta + \frac{1}{p+q}\left(\sum_{i \in E_\lambda} \epsilon_i + \sum_{j \in E_\mu} \xi_j\right) - \left(\nu - \frac{2q_{sf}}{p+q}\right)\theta.
\end{aligned}$$

The rest of the proof is similar to Case 1a with p_{sf} replaced by q_{sf}.

Case 2b. We have $w^\top u_i - b > \eta$ for all $i \notin E_\lambda$. Since by hypothesis $-w^\top v_j + b > \eta$ for all $j \notin E_\mu$, Case 2b is identical to Case 1b, and we are done. ☐

A subtle point here is that Proposition 18.2 shows that if there is an optimal solution, then there is one with a blue and a red support vector, but it does not guarantee that these are support vectors of type 1. Since the dual program does not determine b and η unless these support vectors are of type 1, from a practical point of view this proposition is not helpful.

The proof of Proposition 18.2 reveals that there are three critical values for ν:

$$\frac{2p_{sf}}{p+q}, \ \frac{2q_{sf}}{p+q}, \ \frac{p_{sf}+q_{sf}}{p+q}.$$

These values can be avoided by requiring the strict inequality

$$\max\left\{\frac{2p_{sf}}{p+q}, \frac{2q_{sf}}{p+q}\right\} < \nu.$$

Then the following corollary holds.

Theorem 18.1. *For every optimal solution $(w, b, \eta, \epsilon, \xi)$ of Problem $(\mathrm{SVM}_{s2'})$ with $w \neq 0$ and $\eta > 0$, if*

$$\max\{2p_f/(p+q), 2q_f/(p+q)\} < \nu < \min\{2p/(p+q), 2q/(p+q)\},$$

then some u_{i_0} and some v_{j_0} is a support vector.

Proof. We proceed by contradiction. Suppose that for every optimal solution with $w \neq 0$ and $\eta > 0$ no u_i is a blue support vector or no v_j is a red support vector. Since $\nu < \min\{2p/(p+q), 2q/(p+q)\}$, Proposition 18.2 holds, so there is another optimal solution. But since the critical values of ν are avoided, the proof of Proposition 18.2 shows that the value of the objective function for this new optimal solution is strictly smaller than the original optimal value, a contradiction. ☐

We also have the following proposition that gives a sufficient condition implying that η and b can be found in terms of an optimal solution (λ, μ) of the dual.

Proposition 18.3. *If $(w, b, \eta, \epsilon, \xi)$ is an optimal solution of Problem $(\mathrm{SVM}_{s2'})$ with $w \neq 0$ and $\eta > 0$, if*

$$\max\{2p_f/(p+q), 2q_f/(p+q)\} < \nu < \min\{2p/(p+q), 2q/(p+q)\},$$

then η and b can always be determined from an optimal solution (λ, μ) of the dual in terms of a single support vector.

Proof. By Theorem 18.1 some u_{i_0} and some v_{j_0} is a support vector. As we already explained, Problem (SVM$_{s2'}$) satisfies the conditions for having a zero duality gap. Therefore, for optimal solutions we have

$$L(w, \epsilon, \xi, b, \eta, \lambda, \mu, \alpha, \beta) = G(\lambda, \mu, \alpha, \beta),$$

which means that

$$\frac{1}{2} w^\top w - \nu\eta + \frac{1}{p+q}\left(\sum_{i=1}^{p} \epsilon_i + \sum_{j=1}^{q} \xi_j\right) = -\frac{1}{2}\begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix} X^\top X \begin{pmatrix}\lambda \\ \mu\end{pmatrix},$$

and since

$$w = -X\begin{pmatrix}\lambda \\ \mu\end{pmatrix},$$

we get

$$\frac{1}{p+q}\left(\sum_{i=1}^{p} \epsilon_i + \sum_{j=1}^{q} \xi_j\right) = \nu\eta - \begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix} X^\top X \begin{pmatrix}\lambda \\ \mu\end{pmatrix}. \qquad (*)$$

Let $K_\lambda = \{i \in \{1, \ldots, p\} \mid \lambda_i = K_s\}$ and $K_\mu = \{j \in \{1, \ldots, q\} \mid \mu_j = K_s\}$. By definition, $p_f = |K_\lambda|$ and $q_f = |K_\mu|$ (here we assuming that $K_s = 1/(p+q)$). By complementary slackness the following equations are active:

$$\begin{aligned} w^\top u_i - b &= \eta - \epsilon_i & \quad & i \in K_\lambda \\ -w^\top v_j + b &= \eta - \xi_j & & j \in K_\mu. \end{aligned}$$

But $(*)$ can be written as

$$\frac{1}{p+q}\left(\sum_{i\in K_\lambda} \epsilon_i + \sum_{j\in K_\mu} \xi_j\right) = \nu\eta - \begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix} X^\top X \begin{pmatrix}\lambda \\ \mu\end{pmatrix}, \qquad (**)$$

and since

$$\begin{aligned} \epsilon_i &= \eta - w^\top u_i + b & \quad & i \in K_\lambda \\ \xi_j &= \eta + w^\top v_j - b & & j \in K_\mu, \end{aligned}$$

by substituting in the Equation $(**)$ we get

$$\begin{aligned} \left(\nu - \frac{p_f + q_f}{p+q}\right)\eta = \frac{p_f - q_f}{p+q} b + \frac{1}{p+q} w^\top \left(\sum_{i\in K_\mu} v_j - \sum_{i\in K_\lambda} u_i\right) \\ + \begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix} X^\top X \begin{pmatrix}\lambda \\ \mu\end{pmatrix}. \end{aligned}$$

We also know that $w^\top u_{i_0} - b = \eta$ and $-w^\top v_{j_0} + b = \eta$ for some i_0 and some j_0. In the first case $b = -\eta + w^\top u_{i_0}$, and by substituting b in the above equation we get the equation

$$\begin{aligned}\left(\nu - \frac{p_f + q_f}{p+q}\right)\eta &= -\frac{p_f - q_f}{p+q}\eta + \frac{p_f - q_f}{p+q} w^\top u_{i_0} \\ &\quad + \frac{1}{p+q} w^\top \left(\sum_{i \in K_\mu} v_j - \sum_{i \in K_\lambda} u_i\right) + \left(\lambda^\top\ \mu^\top\right) X^\top X \begin{pmatrix}\lambda \\ \mu\end{pmatrix},\end{aligned}$$

that is,

$$\begin{aligned}\left(\nu - \frac{2q_f}{p+q}\right)\eta &= \frac{p_f - q_f}{p+q} w^\top u_{i_0} + \frac{1}{p+q} w^\top \left(\sum_{i \in K_\mu} v_j - \sum_{i \in K_\lambda} u_i\right) \\ &\quad + \left(\lambda^\top\ \mu^\top\right) X^\top X \begin{pmatrix}\lambda \\ \mu\end{pmatrix}.\end{aligned}$$

In the second case $b = \eta + w^\top v_{j_0}$, and we get the equation

$$\begin{aligned}\left(\nu - \frac{p_f + q_f}{p+q}\right)\eta &= \frac{p_f - q_f}{p+q}\eta + \frac{p_f - q_f}{p+q} w^\top v_{j_0} \\ &\quad + \frac{1}{p+q} w^\top \left(\sum_{i \in K_\mu} v_j - \sum_{i \in K_\lambda} u_i\right) + \left(\lambda^\top\ \mu^\top\right) X^\top X \begin{pmatrix}\lambda \\ \mu\end{pmatrix},\end{aligned}$$

that is,

$$\begin{aligned}\left(\nu - \frac{2p_f}{p+q}\right)\eta &= \frac{p_f - q_f}{p+q} w^\top v_{j_0} + \frac{1}{p+q} w^\top \left(\sum_{i \in K_\mu} v_j - \sum_{i \in K_\lambda} u_i\right) \\ &\quad + \left(\lambda^\top\ \mu^\top\right) X^\top X \begin{pmatrix}\lambda \\ \mu\end{pmatrix}.\end{aligned}$$

We need to choose ν such that $2p_f/(p+q) - \nu \neq 0$ and $2q_f/(p+q) - \nu \neq 0$. Since by Proposition 18.1, we have $\max\{2p_f/(p+q), 2q_f/(p+q)\} \leq \nu$, it suffices to pick ν such that $\max\{2p_f/(p+q), 2q_f/(p+q)\} < \nu$. If this condition is satisfied we can solve for η, and then we find b from either $b = -\eta + w^\top u_{i_0}$ or $b = \eta + w^\top v_{j_0}$. $\square$

Remark: Of course the hypotheses of the proposition imply that $w^\top u_{i_0} - b = \eta$ and $-w^\top v_{j_0} + b = \eta$ for some i_0 and some j_0. Thus we can also compute b and η using the formulae

$$\begin{aligned} b &= \frac{w^\top (u_{i_0} + v_{j_0})}{2} \\ \eta &= \frac{w^\top (u_{i_0} - v_{j_0})}{2}.\end{aligned}$$

The interest of Proposition 18.3 lies in the fact that it allows us to compute b and η knowing only a *single* support vector.

In practice we can only find support vectors of type 1 so Proposition 18.3 is useful if we can only find some blue support vector of type 1 *or* some red support vector of type 1.

As earlier, if we define I_λ and I_μ as

$$\begin{aligned} I_\lambda &= \{i \in \{1, \ldots, p\} \mid 0 < \lambda_i < K_s\} \\ I_\mu &= \{j \in \{1, \ldots, q\} \mid 0 < \mu_j < K_s\}, \end{aligned}$$

then we have the following cases to compute η and b.

(1) If $I_\lambda \neq \emptyset$ and $I_\mu \neq \emptyset$, then

$$\begin{aligned} b &= w^\top \left(\left(\sum_{i \in I_\lambda} u_i \right) / |I_\lambda| + \left(\sum_{j \in I_\mu} v_j \right) / |I_\mu| \right) /2 \\ \eta &= w^\top \left(\left(\sum_{i \in I_\lambda} u_i \right) / |I_\lambda| - \left(\sum_{j \in I_\mu} v_j \right) / |I_\mu| \right) /2. \end{aligned}$$

(2) If $I_\lambda \neq \emptyset$ and $I_\mu = \emptyset$, then

$$\begin{aligned} b &= -\eta + w^\top \left(\sum_{i \in I_\lambda} u_i \right) / |I_\lambda| \\ ((p+q)\nu - 2q_f)\eta &= (p_f - q_f) w^\top \left(\sum_{i \in I_\lambda} u_i \right) / |I_\lambda| \\ &\quad + w^\top \left(\sum_{i \in K_\mu} v_j - \sum_{i \in K_\lambda} u_i \right) \\ &\quad + (p+q) \begin{pmatrix} \lambda^\top & \mu^\top \end{pmatrix} X^\top X \begin{pmatrix} \lambda \\ \mu \end{pmatrix}. \end{aligned}$$

(3) If $I_\lambda = \emptyset$ and $I_\mu \neq \emptyset$, then

$$\begin{aligned} b &= \eta + w^\top \left(\sum_{j \in I_\mu} v_j \right) / |I_\mu| \\ ((p+q)\nu - 2p_f)\eta &= (p_f - q_f) w^\top \left(\sum_{j \in I_\mu} v_j \right) / |I_\mu| \\ &\quad + w^\top \left(\sum_{i \in K_\mu} v_j - \sum_{i \in K_\lambda} u_i \right) \\ &\quad + (p+q) \begin{pmatrix} \lambda^\top & \mu^\top \end{pmatrix} X^\top X \begin{pmatrix} \lambda \\ \mu \end{pmatrix}. \end{aligned}$$

The above formulae correspond to $K_s = 1/(p+q)$. In general we have to replace the rightmost $(p+q)$ by $1/K_s$.

We have examples where there is a single support vector of type 1 and $\nu = 2q_f/(p+q)$, so the above method fails. Curiously, perturbing ν slightly yields a solution with some blue support vector of type 1 and some red support vector of type 1, and so we have not yet found an example where the above method succeeds with a single support vector of type 1. This suggests to conduct some perturbation analysis but it appears to be nontrivial.

Among its advantages, the support vector machinery is conducive to finding interesting statistical bounds in terms of the *VC dimension*, a notion invented by Vapnik and Chernovenkis. We will not go into this here and instead refer the reader to Vapnik [Vapnik (1998)] (especially, Chapter 4 and Chapters 9-13).

18.8 Solving SVM (SVM$_{s2'}$) Using ADMM

In order to solve (SVM$_{s2'}$) using ADMM we need to write the matrix corresponding to the constraints in equational form,

$$\begin{aligned}
\sum_{i=1}^{p} \lambda_i - \sum_{j=1}^{q} \mu_j &= 0 \\
\sum_{i=1}^{p} \lambda_i + \sum_{j=1}^{q} \mu_j - \gamma &= K_m \\
\lambda_i + \alpha_i &= K_s, \quad i = 1, \dots, p \\
\mu_j + \beta_j &= K_s, \quad j = 1, \dots, q,
\end{aligned}$$

with $K_m = (p+q)K_s\nu$. This is the $(p+q+2) \times (2(p+q)+1)$ matrix A given by

$$A = \begin{pmatrix} \mathbf{1}_p^\top & -\mathbf{1}_q^\top & 0_p^\top & 0_q^\top & 0 \\ \mathbf{1}_p^\top & \mathbf{1}_q^\top & 0_p^\top & 0_q^\top & -1 \\ I_p & 0_{p,q} & I_p & 0_{p,q} & 0_p \\ 0_{q,p} & I_q & 0_{q,p} & I_q & 0_q \end{pmatrix}.$$

Observe the remarkable analogy with the matrix arising in ν-regression in Section 20.3, except that $p = q = m$ and that -1 is replaced by $+1$. We leave it as an exercise to prove that A has rank $p+q+2$. The right-hand side is

$$c = \begin{pmatrix} 0 \\ K_m \\ K_s \mathbf{1}_{p+q} \end{pmatrix}.$$

The symmetric positive semidefinite $(p+q) \times (p+q)$ matrix P defining the quadratic functional is

$$P = X^\top X, \quad \text{with} \quad X = \begin{pmatrix} -u_1 \cdots -u_p \; v_1 \cdots v_q \end{pmatrix},$$

and

$$q = 0_{p+q}.$$

Since there are $2(p+q)+1$ Lagrange multipliers $(\lambda, \mu, \alpha, \beta, \gamma)$, the $(p+q) \times (p+q)$ matrix $X^\top X$ must be augmented with zero's to make it a $(2(p+q)+1) \times (2(p+q)+1)$ matrix P_a given by

$$P_a = \begin{pmatrix} X^\top X & 0_{p+q,p+q+1} \\ 0_{p+q+1,p+q} & 0_{p+q+1,p+q+1} \end{pmatrix},$$

and similarly q is augmented with zeros as the vector $q_a = 0_{2(p+q)+1}$.

As we mentioned in Section 18.5, since $\eta \geq 0$ for an optimal solution, we can drop the constraint $\eta \geq 0$ from the primal problem. In this case there are $2(p+q)$ Lagrange multipliers $(\lambda, \mu, \alpha, \beta)$. It is easy to see that the objective function of the dual is unchanged and the set of constraints is

$$\begin{aligned} \sum_{i=1}^p \lambda_i - \sum_{j=1}^q \mu_j &= 0 \\ \sum_{i=1}^p \lambda_i + \sum_{j=1}^q \mu_j &= K_m \\ \lambda_i + \alpha_i &= K_s, \quad i = 1, \ldots, p \\ \mu_j + \beta_j &= K_s, \quad j = 1, \ldots, q, \end{aligned}$$

with $K_m = (p+q)K_s\nu$. The constraint matrix corresponding to this system of equations is the $(p+q+2) \times 2(p+q)$ matrix A_2 given by

$$A_2 = \begin{pmatrix} \mathbf{1}_p^\top & -\mathbf{1}_q^\top & 0_p^\top & 0_q^\top \\ \mathbf{1}_p^\top & \mathbf{1}_q^\top & 0_p^\top & 0_q^\top \\ I_p & 0_{p,q} & I_p & 0_{p,q} \\ 0_{q,p} & I_q & 0_{q,p} & I_q \end{pmatrix}.$$

We leave it as an exercise to prove that A_2 has rank $p+q+2$. The right-hand side is

$$c_2 = \begin{pmatrix} 0 \\ K_m \\ K_s \mathbf{1}_{p+q} \end{pmatrix}.$$

The symmetric positive semidefinite $(p+q) \times (p+q)$ matrix P defining the quadratic functional is

$$P = X^\top X, \quad \text{with} \quad X = \begin{pmatrix} -u_1 & \cdots & -u_p & v_1 & \cdots & v_q \end{pmatrix},$$

and

$$q = 0_{p+q}.$$

Since there are $2(p+q)$ Lagrange multipliers the $(p+q) \times (p+q)$ matrix $X^\top X$ must be augmented with zero's to make it a $2(p+q) \times 2(p+q)$ matrix P_{2a} given by

$$P_{2a} = \begin{pmatrix} X^\top X & 0_{p+q,p+q} \\ 0_{p+q,p+q} & 0_{p+q,p+q} \end{pmatrix},$$

and similarly q is augmented with zeros as the vector $q_{2a} = 0_{2(p+q)}$.

The `Matlab` programs implementing the above method are given in Appendix B, Section B.2. We ran our program on two sets of 30 points each generated at random using the following code which calls the function `runSVMs2pbv3`:

```
rho = 10;
u16 = 10.1*randn(2,30)+7 ;
v16 = -10.1*randn(2,30)-7;
[~,~,~,~,~,~,w3] = runSVMs2pbv3(0.37,rho,u16,v16,1/60)
```

We picked $K = 1/60$ and various values of ν starting with $\nu = 0.37$, which appears to be the smallest value for which the method converges; see Figure 18.11.

In this example, $p_f = 10, q_f = 11, p_m = 12, q_m = 12$. The quadratic solver converged after 8121 steps to reach primal and dual residuals smaller than 10^{-10}.

Reducing ν below $\nu = 0.37$ has the effect that p_f, q_f, p_m, q_m decrease but the following situation arises. Shrinking η a little bit has the effect that $p_f = 9, q_f = 10, p_m = 10, q_m = 11$. Then $\max\{p_f, q_f\} = \min\{p_m, q_m\} = 10$, so the only possible value for ν is $\nu = 20/60 = 1/3 = 0.3333333\cdots$. When we run our program with $\nu = 1/3$, it returns a value of η less than

10^{-13} and a value of w whose components are also less than 10^{-13}. This is probably due to numerical precision. Values of ν less than 1/3 cause the same problem. It appears that the geometry of the problem constrains the values of p_f, q_f, p_m, q_m in such a way that it has no solution other than $w = 0$ and $\eta = 0$.

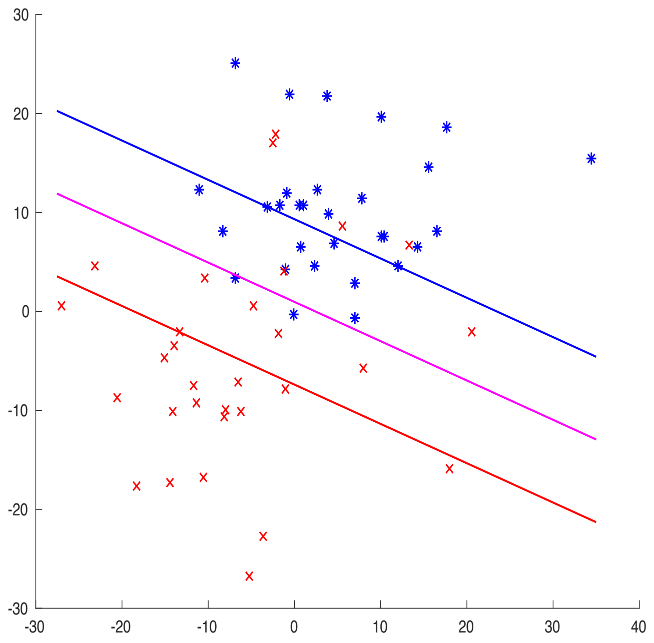


Fig. 18.11 Running (SVM$_{s2'}$) on two sets of 30 points; $\nu = 0.37$.

Figure 18.12 shows the result of running the program with $\nu = 0.51$. We have $p_f = 15, q_f = 16, p_m = 16, q_m = 16$. Interestingly, for $\nu = 0.5$, we run into the singular situation where there is only one support vector and $\nu = 2p_f/(p + q)$.

Next Figure 18.13 shows the result of running the program with $\nu = 0.71$. We have $p_f = 21, q_f = 21, p_m = 22, q_m = 23$. Interestingly, for $\nu = 0.7$, we run into the singular situation where there are no support vectors.

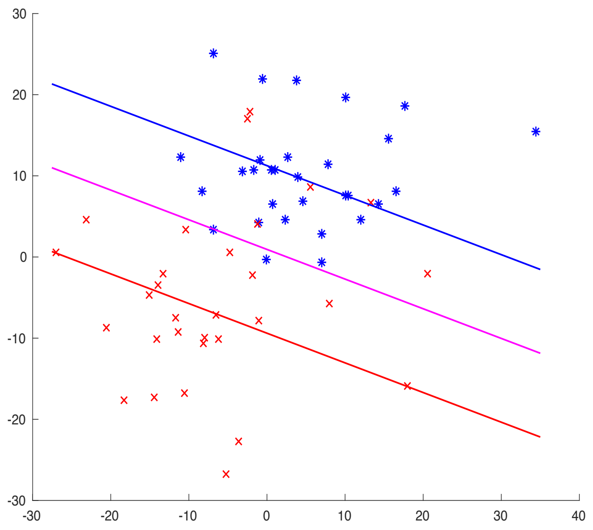


Fig. 18.12 Running ($\text{SVM}_{s2'}$) on two sets of 30 points; $\nu = 0.51$.

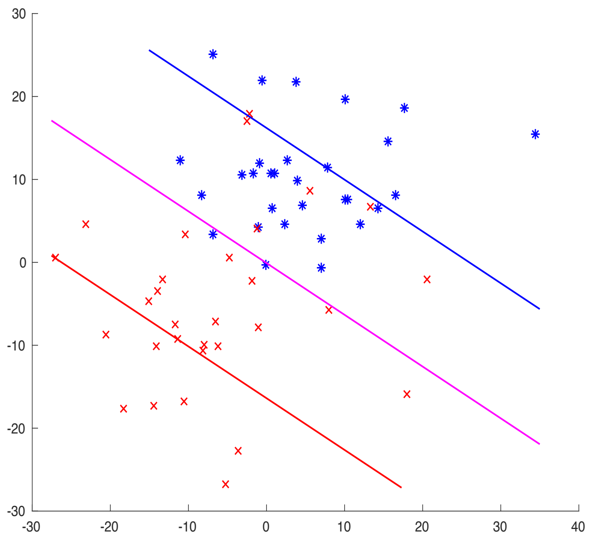


Fig. 18.13 Running ($\text{SVM}_{s2'}$) on two sets of 30 points; $\nu = 0.71$.

For our next to the last run, Figure 18.14 shows the result of running the program with $\nu = 0.95$. We have $p_f = 28, q_f = 28, p_m = 29, q_m = 29$.

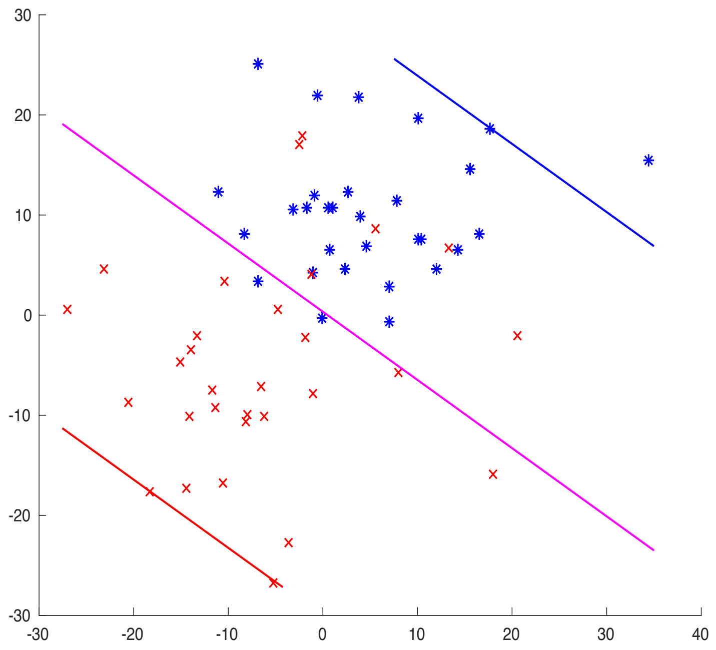


Fig. 18.14 Running (SVM$_{s2'}$) on two sets of 30 points; $\nu = 0.95$.

Figure 18.15 shows the result of running the program with $\nu = 0.97$. We have $p_f = 29, q_f = 29, p_m = 30, q_m = 30$, which shows that the largest margin has been achieved. However, after 80000 iterations the dual residual is less than 10^{-12} but the primal residual is approximately 10^{-4} (our tolerance for convergence is 10^{-10}, which is quite high). Nevertheless the result is visually very good.

18.9 Soft Margin Support Vector Machines; (SVM$_{s3}$)

In this section we consider a variation of Problem (SVM$_{s2'}$) by adding the term $(1/2)b^2$ to the objective function. The result is that in minimizing the Lagrangian to find the dual function G, not just w but also b is determined and η is determined under a mild condition on ν. We also suppress the

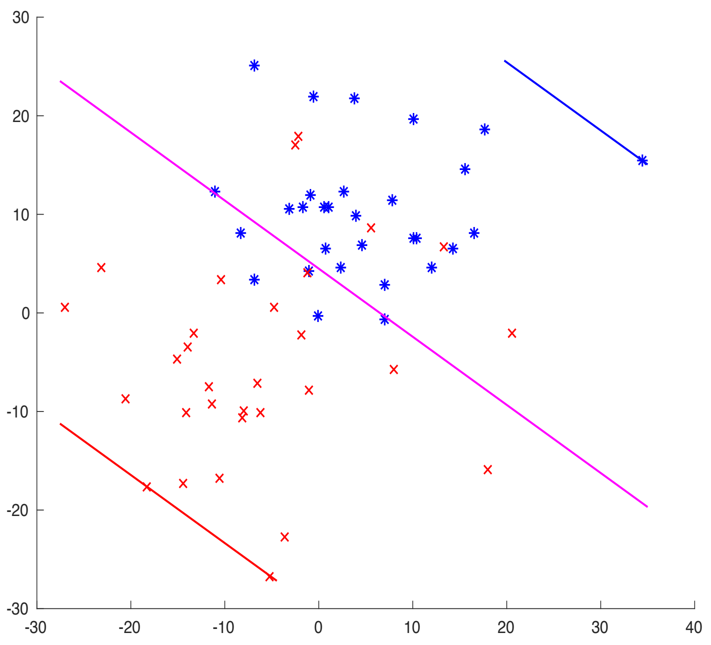


Fig. 18.15 Running $(\mathrm{SVM}_{s2'})$ on two sets of 30 points; $\nu = 0.97$.

constraint $\eta \geq 0$ which turns out to be redundant.

Soft margin SVM (SVM_{s3}):

$$\begin{aligned}
&\text{minimize} \quad \frac{1}{2}w^\top w + \frac{1}{2}b^2 + (p+q)K_s\left(-\nu\eta + \frac{1}{p+q}\begin{pmatrix}\epsilon^\top & \xi^\top\end{pmatrix}\mathbf{1}_{p+q}\right)\\
&\text{subject to}\\
&\qquad w^\top u_i - b \geq \eta - \epsilon_i, \quad \epsilon_i \geq 0 \qquad i = 1, \ldots, p\\
&\quad - w^\top v_j + b \geq \eta - \xi_j, \quad \xi_j \geq 0 \qquad j = 1, \ldots, q.
\end{aligned}$$

To simplify the presentation we assume that $K_s = 1/(p+q)$. When writing a computer program it is more convenient to assume that K_s is arbitrary. In this case, ν needs to be replaced by $(p+q)K_s\nu$ in all the formulae.

The Lagrangian $L(w, \epsilon, \xi, b, \eta, \lambda, \mu, \alpha, \beta)$ with $\lambda, \alpha \in \mathbb{R}^p_+$, $\mu, \beta \in \mathbb{R}^q_+$ is given by

$$\begin{aligned}L(w,\epsilon,\xi,b,\eta,\lambda,\mu,\alpha,\beta) &= \frac{1}{2}w^\top w + w^\top X\begin{pmatrix}\lambda\\ \mu\end{pmatrix} + \frac{b^2}{2} + K_s(\epsilon^\top \mathbf{1}_p + \xi^\top \mathbf{1}_q)\\ &\quad - \epsilon^\top(\lambda+\alpha) - \nu\eta - \xi^\top(\mu+\beta) + b(\mathbf{1}_p^\top\lambda - \mathbf{1}_q^\top\mu) + \eta(\mathbf{1}_p^\top\lambda + \mathbf{1}_q^\top\mu)\\ &= \frac{1}{2}w^\top w + w^\top X\begin{pmatrix}\lambda\\ \mu\end{pmatrix} + \frac{b^2}{2} + b(\mathbf{1}_p^\top\lambda - \mathbf{1}_q^\top\mu) + \eta(\mathbf{1}_p^\top\lambda + \mathbf{1}_q^\top\mu - \nu)\\ &\qquad + \epsilon^\top(K_s\mathbf{1}_p - (\lambda+\alpha)) + \xi^\top(K_s\mathbf{1}_q - (\mu+\beta)).\end{aligned}$$

To find the dual function $G(\lambda,\mu,\alpha,\beta)$, we minimize $L(w,\epsilon,\xi,b,\eta,\lambda,\mu,\alpha,\beta)$ with respect to w,ϵ,ξ,b, and η. Since the Lagrangian is convex and $(w,\epsilon,\xi,b,\eta)\in\mathbb{R}^n\times\mathbb{R}^p\times\mathbb{R}^q\times\mathbb{R}\times\mathbb{R}$, a convex open set, by Theorem 4.5, the Lagrangian has a minimum in (w,ϵ,ξ,b,η) iff $\nabla L_{w,\epsilon,\xi,b,\eta}=0$, so we compute its gradient with respect to w,ϵ,ξ,b,η, and we get

$$\nabla L_{w,\epsilon,\xi,b,\eta} = \begin{pmatrix} X\begin{pmatrix}\lambda\\ \mu\end{pmatrix} + w\\ K_s\mathbf{1}_p - (\lambda+\alpha)\\ K_s\mathbf{1}_q - (\mu+\beta)\\ b + \mathbf{1}_p^\top\lambda - \mathbf{1}_q^\top\mu\\ \mathbf{1}_p^\top\lambda + \mathbf{1}_q^\top\mu - \nu\end{pmatrix}.$$

By setting $\nabla L_{w,\epsilon,\xi,b,\eta}=0$ we get the equations

$$w = -X\begin{pmatrix}\lambda\\ \mu\end{pmatrix} \tag{$*_w$}$$

$$\begin{aligned}\lambda+\alpha &= K_s\mathbf{1}_p\\ \mu+\beta &= K_s\mathbf{1}_q\\ \mathbf{1}_p^\top\lambda + \mathbf{1}_q^\top\mu &= \nu,\end{aligned}$$

and

$$b = -(\mathbf{1}_p^\top\lambda - \mathbf{1}_q^\top\mu). \tag{$*_b$}$$

The second and third equations are equivalent to the box constraints

$$0\leq \lambda_i,\mu_j\leq K_s,\quad i=1,\ldots,p,\ j=1,\ldots,q.$$

Since we assumed that the primal problem has an optimal solution with $w\neq 0$, we have

$$X\begin{pmatrix}\lambda\\ \mu\end{pmatrix}\neq 0.$$

Plugging back w from $(*_w)$ and b from $(*_b)$ into the Lagrangian, we get

$$\begin{aligned} G(\lambda, \mu, \alpha, \beta) &= \frac{1}{2}\begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix} X^\top X \begin{pmatrix}\lambda \\ \mu\end{pmatrix} - \begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix} X^\top X \begin{pmatrix}\lambda \\ \mu\end{pmatrix} + \frac{1}{2}b^2 - b^2 \\ &= -\frac{1}{2}\begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix} X^\top X \begin{pmatrix}\lambda \\ \mu\end{pmatrix} - \frac{1}{2}b^2 \\ &= -\frac{1}{2}\begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix}\left(X^\top X + \begin{pmatrix}\mathbf{1}_p\mathbf{1}_p^\top & -\mathbf{1}_p\mathbf{1}_q^\top \\ -\mathbf{1}_q\mathbf{1}_p^\top & \mathbf{1}_q\mathbf{1}_q^\top\end{pmatrix}\right)\begin{pmatrix}\lambda \\ \mu\end{pmatrix}, \end{aligned}$$

so the dual function is independent of α, β and is given by

$$G(\lambda, \mu) = -\frac{1}{2}\begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix}\left(X^\top X + \begin{pmatrix}\mathbf{1}_p\mathbf{1}_p^\top & -\mathbf{1}_p\mathbf{1}_q^\top \\ -\mathbf{1}_q\mathbf{1}_p^\top & \mathbf{1}_q\mathbf{1}_q^\top\end{pmatrix}\right)\begin{pmatrix}\lambda \\ \mu\end{pmatrix}.$$

The dual program is given by

$$\begin{aligned} &\text{maximize} \quad -\frac{1}{2}\begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix}\left(X^\top X + \begin{pmatrix}\mathbf{1}_p\mathbf{1}_p^\top & -\mathbf{1}_p\mathbf{1}_q^\top \\ -\mathbf{1}_q\mathbf{1}_p^\top & \mathbf{1}_q\mathbf{1}_q^\top\end{pmatrix}\right)\begin{pmatrix}\lambda \\ \mu\end{pmatrix} \\ &\text{subject to} \\ &\qquad \sum_{i=1}^p \lambda_i + \sum_{j=1}^q \mu_j = \nu \\ &\qquad 0 \leq \lambda_i \leq K_s, \quad i = 1, \ldots, p \\ &\qquad 0 \leq \mu_j \leq K_s, \quad j = 1, \ldots, q. \end{aligned}$$

Finally, the dual program is equivalent to the following minimization program:

Dual of the Soft margin SVM (SVM_{s3}):

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2}\begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix}\left(X^\top X + \begin{pmatrix}\mathbf{1}_p\mathbf{1}_p^\top & -\mathbf{1}_p\mathbf{1}_q^\top \\ -\mathbf{1}_q\mathbf{1}_p^\top & \mathbf{1}_q\mathbf{1}_q^\top\end{pmatrix}\right)\begin{pmatrix}\lambda \\ \mu\end{pmatrix} \\ &\text{subject to} \\ &\qquad \sum_{i=1}^p \lambda_i + \sum_{j=1}^q \mu_j = \nu \\ &\qquad 0 \leq \lambda_i \leq K_s, \quad i = 1, \ldots, p \\ &\qquad 0 \leq \mu_j \leq K_s, \quad j = 1, \ldots, q. \end{aligned}$$

The classification of the points u_i and v_j in terms of the values of λ and μ and Definition 18.2 and Definition 18.3 are unchanged.

It is shown in Section 18.12 how the dual program is solved using ADMM from Section 16.6. If the primal problem is solvable, this yields solutions for λ and μ. Once a solution for λ and μ is obtained, we have

$$\begin{aligned} w &= -X\begin{pmatrix}\lambda\\ \mu\end{pmatrix} = \sum_{i=1}^p \lambda_i u_i - \sum_{j=1}^q \mu_j v_j \\ b &= -(\mathbf{1}_p^\top \lambda - \mathbf{1}_q^\top \mu) = -\sum_{i=1}^p \lambda_i + \sum_{j=1}^q \mu_j. \end{aligned}$$

We can compute η using duality. As we said earlier, the hypotheses of Theorem 14.5(2) hold, so *if* the primal problem (SVM$_{s3}$) has an optimal solution with $w \neq 0$, *then* the dual problem has a solution too, and the duality gap is zero. Therefore, for optimal solutions we have

$$L(w, \epsilon, \xi, b, \eta, \lambda, \mu, \alpha, \beta) = G(\lambda, \mu, \alpha, \beta),$$

which means that

$$\begin{aligned} &\frac{1}{2}w^\top w + \frac{b^2}{2} - (p+q)K_s\nu\eta + K_s\left(\sum_{i=1}^p \epsilon_i + \sum_{j=1}^q \xi_j\right) \\ &\qquad = -\frac{1}{2}\begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix}\left(X^\top X + \begin{pmatrix}\mathbf{1}_p\mathbf{1}_p^\top & -\mathbf{1}_p\mathbf{1}_q^\top \\ -\mathbf{1}_q\mathbf{1}_p^\top & \mathbf{1}_q\mathbf{1}_q^\top\end{pmatrix}\right)\begin{pmatrix}\lambda\\ \mu\end{pmatrix}. \end{aligned}$$

We can use the above equation to determine η.

Since

$$\frac{1}{2}w^\top w + \frac{b^2}{2} = \frac{1}{2}\begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix}\left(X^\top X + \begin{pmatrix}\mathbf{1}_p\mathbf{1}_p^\top & -\mathbf{1}_p\mathbf{1}_q^\top \\ -\mathbf{1}_q\mathbf{1}_p^\top & \mathbf{1}_q\mathbf{1}_q^\top\end{pmatrix}\right)\begin{pmatrix}\lambda\\ \mu\end{pmatrix},$$

we get

$$\begin{aligned} (p+q)K_s\nu\eta &= K_s\left(\sum_{i=1}^p \epsilon_i + \sum_{j=1}^q \xi_j\right) \\ &\quad + \begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix}\left(X^\top X + \begin{pmatrix}\mathbf{1}_p\mathbf{1}_p^\top & -\mathbf{1}_p\mathbf{1}_q^\top \\ -\mathbf{1}_q\mathbf{1}_p^\top & \mathbf{1}_q\mathbf{1}_q^\top\end{pmatrix}\right)\begin{pmatrix}\lambda\\ \mu\end{pmatrix}. \qquad (*) \end{aligned}$$

Since

$$X^\top X + \begin{pmatrix}\mathbf{1}_p\mathbf{1}_p^\top & -\mathbf{1}_p\mathbf{1}_q^\top \\ -\mathbf{1}_q\mathbf{1}_p^\top & \mathbf{1}_q\mathbf{1}_q^\top\end{pmatrix}$$

is positive semidefinite, we confirm that $\eta \geq 0$.

Since nonzero ϵ_i and ξ_j may only occur for vectors u_i and v_j that fail the margin, namely $\lambda_i = K_s$, $\mu_j = K_s$, the corresponding constraints are active and we can solve for ϵ_i and ξ_j in terms of b and η. Let K_λ and K_μ be the sets of indices corresponding to points failing the margin,

$$\begin{aligned}K_\lambda &= \{i \in \{1, \ldots, p\} \mid \lambda_i = K_s\}\\ K_\mu &= \{j \in \{1, \ldots, q\} \mid \mu_j = K_s\}.\end{aligned}$$

By definition $p_f = |K_\lambda|, q_f = |K_\mu|$. Then for every $i \in K_\lambda$ we have

$$\epsilon_i = \eta + b - w^\top u_i$$

and for every $j \in K_\mu$ we have

$$\xi_j = \eta - b + w^\top v_j.$$

Using the above formulae we obtain

$$\begin{aligned}\sum_{i=1}^p \epsilon_i + \sum_{j=1}^q \xi_j &= \sum_{i\in K_\lambda} \epsilon_i + \sum_{j\in K_\mu} \xi_j\\ &= \sum_{i\in K_\lambda} (\eta + b - w^\top u_i) + \sum_{j\in K_\mu} (\eta - b + w^\top v_j)\\ &= (p_f + q_f)\eta + (p_f - q_f)b + w^\top \left(\sum_{j\in K_\mu} v_j - \sum_{i\in K_\lambda} u_i\right)\end{aligned}$$

Substituting this expression in $(*)$ we obtain

$$\begin{aligned}(p+q)K_s\nu\eta &= K_s\left(\sum_{i=1}^p \epsilon_i + \sum_{j=1}^q \xi_j\right)\\ &\quad + \begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix}\left(X^\top X + \begin{pmatrix}\mathbf{1}_p\mathbf{1}_p^\top & -\mathbf{1}_p\mathbf{1}_q^\top\\ -\mathbf{1}_q\mathbf{1}_p^\top & \mathbf{1}_q\mathbf{1}_q^\top\end{pmatrix}\right)\begin{pmatrix}\lambda\\ \mu\end{pmatrix}\\ &= K_s\left((p_f + q_f)\eta + (p_f - q_f)b + w^\top\left(\sum_{j\in K_\mu} v_j - \sum_{i\in K_\lambda} u_i\right)\right)\\ &\quad + \begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix}\left(X^\top X + \begin{pmatrix}\mathbf{1}_p\mathbf{1}_p^\top & -\mathbf{1}_p\mathbf{1}_q^\top\\ -\mathbf{1}_q\mathbf{1}_p^\top & \mathbf{1}_q\mathbf{1}_q^\top\end{pmatrix}\right)\begin{pmatrix}\lambda\\ \mu\end{pmatrix},\end{aligned}$$

which yields

$$\begin{aligned}((p+q)\nu - p_f - q_f)\eta &= (p_f - q_f)b + w^\top\left(\sum_{j\in K_\mu} v_j - \sum_{i\in K_\lambda} u_i\right)\\ &\quad + \frac{1}{K_s}\begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix}\left(X^\top X + \begin{pmatrix}\mathbf{1}_p\mathbf{1}_p^\top & -\mathbf{1}_p\mathbf{1}_q^\top\\ -\mathbf{1}_q\mathbf{1}_p^\top & \mathbf{1}_q\mathbf{1}_q^\top\end{pmatrix}\right)\begin{pmatrix}\lambda\\ \mu\end{pmatrix}.\end{aligned}$$

We show in Proposition 18.4 that $p_f + q_f \leq (p+q)\nu$, so if $\nu > (p_f + q_f)/(p+q)$, we can solve for η in terms of b, w, and λ, μ. But b and w are expressed in terms of λ, μ as

$$\begin{aligned} w &= -X \begin{pmatrix} \lambda \\ \mu \end{pmatrix} \\ b &= -\sum_{i=1}^{p} \lambda_i + \sum_{j=1}^{q} \mu_j = -\mathbf{1}_p^\top \lambda + \mathbf{1}_q^\top \mu \end{aligned}$$

so η is also expressed in terms of λ, μ.

The condition $\nu > (p_f + q_f)/(p+q)$ cannot be satisfied if $p_f + q_f = p+q$, but in this case all points fail the margin, which indicates that δ is too big, so we reduce ν and try again.

Remark: The equation

$$\sum_{i=1}^{p} \lambda_i + \sum_{j=1}^{q} \mu_j = \nu$$

implies that either there is some i_0 such that $\lambda_{i_0} > 0$ *or* there is some j_0 such that $\mu_{j_0} > 0$, which implies that $p_m + q_m \geq 1$.

Another way to compute η is to assume the Standard Margin Hypothesis for (SVM$_{s3}$). Under the **Standard Margin Hypothesis** for (SVM$_{s3}$), either there is some i_0 such that $0 < \lambda_{i_0} < K_s$ *or* there is some j_0 such that $0 < \mu_{j_0} < K_s$, in other words, there is some support vector of type 1. By the complementary slackness conditions $\epsilon_{i_0} = 0$ or $\xi_{j_0} = 0$, so we have

$$w^\top u_{i_0} - b = \eta, \quad \text{or} \quad -w^\top v_{j_0} + b = \eta,$$

and we can solve for η.

Due to numerical instability, when writing a computer program it is preferable to compute the lists of indices I_λ and I_μ given by

$$\begin{aligned} I_\lambda &= \{i \in \{1, \ldots, p\} \mid 0 < \lambda_i < K_s\} \\ I_\mu &= \{j \in \{1, \ldots, q\} \mid 0 < \mu_j < K_s\}. \end{aligned}$$

Then it is easy to see that we can compute η using the following averaging formulae: If $I_\lambda \neq \emptyset$, then

$$\eta = w^\top \left(\sum_{i \in I_\lambda} u_i \right) / |I_\lambda| - b,$$

and if $I_\mu \neq \emptyset$, then

$$\eta = b - w^\top \left(\sum_{j \in I_\mu} v_j \right) / |I_\mu|.$$

Theoretically the condition $\nu > (p_f + q_f)/(p+q)$ is less restrictive that the **Standard Margin Hypothesis** but in practice we have never observed an example for which $\nu > (p_f+q_f)/(p+q)$ and yet the **Standard Margin Hypothesis** fails.

The "kernelized" version of Problem (SVM$_{s3}$) is the following:

Soft margin kernel SVM (SVM$_{s3}$):

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2}\langle w, w\rangle + \frac{1}{2}b^2 - \nu\eta + K_s\left(\epsilon^\top \ \xi^\top\right)\mathbf{1}_{p+q} \\ &\text{subject to} \\ &\qquad \langle w, \varphi(u_i)\rangle - b \geq \eta - \epsilon_i, \quad \epsilon_i \geq 0 \qquad i = 1, \ldots, p \\ &\qquad -\langle w, \varphi(v_j)\rangle + b \geq \eta - \xi_j, \quad \xi_j \geq 0 \qquad j = 1, \ldots, q, \end{aligned}$$

with $K_s = 1/(p+q)$.

Tracing through the derivation of the dual program, we obtain

Dual of the Soft margin kernel SVM (SVM$_{s3}$):

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2}\left(\lambda^\top \ \mu^\top\right)\left(\mathbf{K} + \begin{pmatrix} \mathbf{1}_p\mathbf{1}_p^\top & -\mathbf{1}_p\mathbf{1}_q^\top \\ -\mathbf{1}_q\mathbf{1}_p^\top & \mathbf{1}_q\mathbf{1}_q^\top \end{pmatrix}\right)\begin{pmatrix}\lambda \\ \mu\end{pmatrix} \\ &\text{subject to} \\ &\qquad \sum_{i=1}^p \lambda_i + \sum_{j=1}^q \mu_j = \nu \\ &\qquad 0 \leq \lambda_i \leq K_s, \quad i = 1, \ldots, p \\ &\qquad 0 \leq \mu_j \leq K_s, \quad j = 1, \ldots, q, \end{aligned}$$

where $\mathbf{K}$ is the kernel matrix of Section 18.1.

We obtain

$$\begin{aligned} w &= \sum_{i=1}^p \lambda_i\varphi(u_i) - \sum_{j=1}^q \mu_j\varphi(v_j) \\ b &= -\sum_{i=1}^p \lambda_i + \sum_{j=1}^q \mu_j. \end{aligned}$$

The classification function

$$f(x) = \operatorname{sgn}(\langle w, \varphi(x)\rangle - b)$$

is given by

$$f(x) = \operatorname{sgn}\left(\sum_{i=1}^p \lambda_i(\kappa(u_i, x) + 1) - \sum_{j=1}^q \mu_j(\kappa(v_j, x) + 1)\right).$$

18.10 Classification of the Data Points in Terms of ν (SVM$_{s3}$)

The equations (†) and the box inequalities

$$0 \leq \lambda_i \leq K_s, \quad 0 \leq \mu_j \leq K_s$$

also imply the following facts (recall that $\delta = \eta / \|w\|$):

Proposition 18.4. *If Problem* (SVM$_{s3}$) *has an optimal solution with $w \neq 0$ and $\eta > 0$ then the following facts hold:*

(1) Let p_f be the number of points u_i such that $\lambda_i = K_s$, and let q_f the number of points v_j such that $\mu_j = K_s$. Then $p_f + q_f \leq (p+q)\nu$.
(2) Let p_m be the number of points u_i such that $\lambda_i > 0$, and let q_m the number of points v_j such that $\mu_j > 0$. Then $p_m + q_m \geq (p+q)\nu$. We have $p_m + q_m \geq 1$.
(3) If $p_f \geq 1$ or $q_f \geq 1$, then $\nu \geq 1/(p+q)$.

Proof. (1) Recall that for an optimal solution with $w \neq 0$ and $\eta > 0$ we have the equation

$$\sum_{i=1}^{p} \lambda_i + \sum_{j=1}^{q} \mu_j = \nu.$$

Since there are p_f points u_i such that $\lambda_i = K_s = 1/(p+q)$ and q_f points v_j such that $\mu_j = K_s = 1/(p+q)$, we have

$$\nu = \sum_{i=1}^{p} \lambda_i + \sum_{j=1}^{q} \mu_j \geq \frac{p_f + q_f}{p+q},$$

so

$$p_f + q_f \leq \nu(p+q).$$

(2) If

$$I_{\lambda>0} = \{i \in \{1, \ldots, p\} \mid \lambda_i > 0\} \quad \text{and} \quad p_m = |I_{\lambda>0}|$$

and

$$I_{\mu>0} = \{j \in \{1, \ldots, q\} \mid \mu_j > 0\} \quad \text{and} \quad q_m = |I_{\mu>0}|,$$

then

$$\nu = \sum_{i=1}^{p} \lambda_i + \sum_{j=1}^{q} \mu_j = \sum_{i \in I_{\lambda>0}} \lambda_i + \sum_{j \in I_{\mu>0}} \mu_j,$$

and since $\lambda_i, \mu_j \leq K_s = 1/(p+q)$, we have

$$\nu = \sum_{i \in I_{\lambda>0}} \lambda_i + \sum_{j \in I_{\mu>0}} \mu_j \leq \frac{p_m + q_m}{p+q},$$

which yields

$$p_m + q_m \geq \nu(p+q).$$

We already noted earlier that $p_m + q_m \geq 1$.

(3) This follows immediately from (1). ☐

Note that if ν is chosen so that $\nu < 1/(p+q)$, then $p_f = q_f = 0$, which means that none of the data points are misclassified; in other words, the u_is and v_js are linearly separable. Thus we see that if the u_is and v_js are not linearly separable we must pick ν such that $1/(p+q) \leq \nu \leq 1$ for the method to succeed. In fact, by Proposition 18.4, we must choose ν so that

$$\frac{p_f + q_f}{p+q} \leq \nu \leq \frac{p_m + q_m}{p+q}.$$

Furthermore, in order to be able to determine b, we must have the strict inequality

$$\frac{p_f + q_f}{p+q} < \nu.$$

18.11 Existence of Support Vectors for (SVM$_{s3}$)

The following proposition is the version of Proposition 18.2 for Problem (SVM$_{s3}$).

Proposition 18.5. *For every optimal solution* $(w, b, \eta, \epsilon, \xi)$ *of Problem* (SVM$_{s3}$) *with* $w \neq 0$ *and* $\eta > 0$*, if* $\nu < 1$ *and if no* u_i *is a support vector and no* v_j *is a support vector, then there is another optimal solution such that some* u_{i_0} *or some* v_{j_0} *is a support vector.*

Proof. We may assume that $K_s = 1/(p+q)$ and we proceed by contradiction. Thus we assume that for all $i \in \{1, \ldots, p\}$, if $\epsilon_i = 0$, then the constraint $w^\top u_i - b \geq \eta$ is not active, namely $w^\top u_i - b > \eta$, *and* for all $j \in \{1, \ldots, q\}$, if $\xi_j = 0$, then the constraint $-w^\top v_j + b \geq \eta$ is not active, namely $-w^\top v_j + b > \eta$.

Let $E_\lambda = \{i \in \{1, \ldots, p\} \mid \epsilon_i > 0\}$ and let $E_\mu = \{j \in \{1, \ldots, q\} \mid \xi_j > 0\}$. By definition, $p_{sf} = |E_\lambda|$, $q_{sf} = |E_\mu|$, $p_{sf} \leq p_f$ and $q_{sf} \leq q_f$, so by Proposition 18.1,

$$\frac{p_{sf} + q_{sf}}{p+q} \leq \frac{p_f + q_f}{p+q} \leq \nu.$$

Therefore, if $\nu < 1$, then $p_{sf} + q_{sf} < p + q$, which implies that either there is some $i \notin E_\lambda$ such that $\epsilon_i = 0$ or there is some $j \notin E_\mu$ such that $\xi_j = 0$.

By complementary slackness all the constraints for which $i \in E_\lambda$ and $j \in E_\mu$ are active, so our hypotheses are

$$\begin{aligned}
w^\top u_i - b &= \eta - \epsilon_i & \epsilon_i &> 0 & i &\in E_\lambda\\
-w^\top v_j + b &= \eta - \xi_j & \xi_j &> 0 & j &\in E_\mu\\
w^\top u_i - b &> \eta & & & i &\notin E_\lambda\\
-w^\top v_j + b &> \eta & & & j &\notin E_\mu,
\end{aligned}$$

and either there is some $i \notin E_\lambda$ or there is some $j \notin E_\mu$. Our strategy, as illustrated in Figures 18.8 and 18.9, is to increase the width η of the slab keeping the separating hyperplane unchanged. Let us pick θ such that

$$\theta = \min\{w^\top u_i - b - \eta,\ -w^\top v_j + b - \eta \mid i \notin E_\lambda, j \notin E_\mu\}.$$

Our hypotheses imply that $\theta > 0$. We can write

$$\begin{aligned}
w^\top u_i - b &= \eta + \theta - (\epsilon_i + \theta) & \epsilon_i + \theta &> 0 & i &\in E_\lambda\\
-w^\top v_j + b &= \eta + \theta - (\xi_j + \theta) & \xi_j + \theta &> 0 & j &\in E_\mu\\
w^\top u_i - b &\geq \eta + \theta & & & i &\notin E_\lambda\\
-w^\top v_j + b &\geq \eta + \theta & & & j &\notin E_\mu,
\end{aligned}$$

and by the choice of θ, either

$$w^\top u_i - b = \eta + \theta \quad \text{for some } i \notin E_\lambda$$

or

$$-w^\top v_j + b = \eta + \theta \quad \text{for some } j \notin E_\mu.$$

The original value of the objective function is

$$\omega(0) = \frac{1}{2}w^\top w + \frac{1}{2}b^2 - \nu\eta + \frac{1}{p+q}\left(\sum_{i\in E_\lambda} \epsilon_i + \sum_{j \in E_\mu} \xi_j\right),$$

and the new value is

$$\begin{aligned}
\omega(\theta) &= \frac{1}{2}w^\top w + \frac{1}{2}b^2 - \nu(\eta+\theta) + \frac{1}{p+q}\left(\sum_{i\in E_\lambda} (\epsilon_i + \theta) + \sum_{j \in E_\mu} (\xi_j + \theta)\right)\\
&= \frac{1}{2}w^\top w + \frac{1}{2}b^2 - \nu\eta + \frac{1}{p+q}\left(\sum_{i\in E_\lambda} \epsilon_i + \sum_{j \in E_\mu} \xi_j\right) - \left(\nu - \frac{p_{sf}+q_{sf}}{p+q}\right)\theta.
\end{aligned}$$

By Proposition 18.1,

$$\frac{p_{sf}+q_{sf}}{p+q} \leq \frac{p_f + q_f}{p+q} \leq \nu,$$

so

$$\nu - \frac{p_{sf} + q_{sf}}{p+q} \geq 0,$$

and so $\omega(\theta) \leq \omega(0)$. If the inequality is strict, then this contradicts the optimality of the original solution. Therefore, $\omega(\theta) = \omega(0)$ and $(w, b, \eta + \theta, \epsilon + \theta, \xi + \theta)$ is an optimal solution such that either

$$w^\top u_i - b = \eta + \theta \quad \text{for some } i \notin E_\lambda$$

or

$$-w^\top v_j + b = \eta + \theta \quad \text{for some } j \notin E_\mu,$$

as desired. ☐

Proposition 18.5 cannot be strengthened to claim that there is some support vector u_{i_0} *and* some support vector v_{j_0}. We found examples for which the above condition fails for ν large enough.

The proof of Proposition 18.5 reveals that $(p_{sf} + q_{sf})/(p+q)$ is a critical value for ν. if this value is avoided we have the following corollary.

Theorem 18.2. *For every optimal solution $(w, b, \eta, \epsilon, \xi)$ of Problem* (SVM$_{s3}$) *with $w \neq 0$ and $\eta > 0$, if*

$$(p_{sf} + q_{sf})/(p+q) < \nu < 1,$$

then some u_{i_0} or some v_{j_0} is a support vector.

The proof proceeds by contradiction using Proposition 18.5 (for a very similar proof, see the proof of Theorem 18.1).

18.12 Solving SVM (SVM$_{s3}$) Using ADMM

In order to solve (SVM$_{s3}$) using ADMM we need to write the matrix corresponding to the constraints in equational form,

$$\begin{aligned} \sum_{i=1}^{p} \lambda_i + \sum_{j=1}^{q} \mu_j &= K_m \\ \lambda_i + \alpha_i &= K_s, \quad i = 1, \ldots, p \\ \mu_j + \beta_j &= K_s, \quad j = 1, \ldots, q \end{aligned}$$

with $K_m = (p+q)K_s\nu$. This is the $(p+q+1) \times 2(p+q)$ matrix A given by

$$A = \begin{pmatrix} \mathbf{1}_p^\top & \mathbf{1}_q^\top & 0_p^\top & 0_q^\top \\ I_p & 0_{p,q} & I_p & 0_{p,q} \\ 0_{q,p} & I_q & 0_{q,p} & I_q \end{pmatrix}.$$

We leave it as an exercise to prove that A has rank $p+q+1$. The right-hand side is

$$c = \begin{pmatrix} K_m \\ K_s\mathbf{1}_{p+q} \end{pmatrix}.$$

The symmetric positive semidefinite $(p+q)\times(p+q)$ matrix P defining the quadratic functional is

$$P = X^\top X + \begin{pmatrix} \mathbf{1}_p\mathbf{1}_p^\top & -\mathbf{1}_p\mathbf{1}_q^\top \\ -\mathbf{1}_q\mathbf{1}_p^\top & \mathbf{1}_q\mathbf{1}_q^\top \end{pmatrix}, \quad \text{with} \quad X = \begin{pmatrix} -u_1 & \cdots & -u_p & v_1 & \cdots & v_q \end{pmatrix},$$

and

$$q = 0_{p+q}.$$

Since there are $2(p+q)$ Lagrange multipliers $(\lambda, \mu, \alpha, \beta)$, the $(p+q)\times(p+q)$ matrix P must be augmented with zero's to make it a $2(p+q)\times 2(p+q)$ matrix P_a given by

$$P_a = \begin{pmatrix} P & 0_{p+q,p+q} \\ 0_{p+q,p+q} & 0_{p+q,p+q} \end{pmatrix},$$

and similarly q is augmented with zeros as the vector

$$q_a = 0_{2(p+q)}.$$

The `Matlab` programs implementing the above method are given in Appendix B, Section B.3. We ran our program on the same input data points used in Section 18.8, namely

```
u16 = 10.1*randn(2,30)+7 ;
v16 = -10.1*randn(2,30)-7;
[~,~,~,~,~,~,w3] = runSVMs3b(0.365,rho,u16,v16,1/60)
```

We picked $K = 1/60$ and various values of ν starting with $\nu = 0.365$, which appears to be the smallest value for which the method converges; see Figure 18.16.

We have $p_f = 10, q_f = 10, p_m = 12$ and $q_m = 11$, as opposed to $p_f = 10, q_f = 11, p_m = 12, q_m = 12$, which was obtained by running (SVM$_{s2'}$); see Figure 18.11. A slightly narrower margin is achieved.

Next we ran our program with $\nu = 0.5$, see Figure 18.17. We have $p_f = 13, q_f = 16, p_m = 14$ and $q_m = 17$.

We also ran our program with $\nu = 0.71$, see Figure 18.18. We have $p_f = 21, q_f = 21, p_m = 22$ and $q_m = 22$. The value $\nu = 0.7$ is a singular value for which there are no support vectors and $\nu = (p_f + q_f)/(p+q)$.

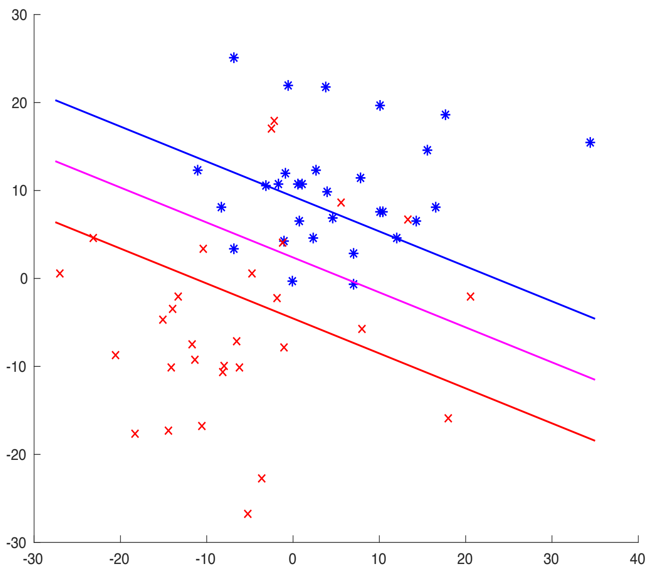


Fig. 18.16 Running (SVM$_{s3}$) on two sets of 30 points; $\nu = 0.365$.

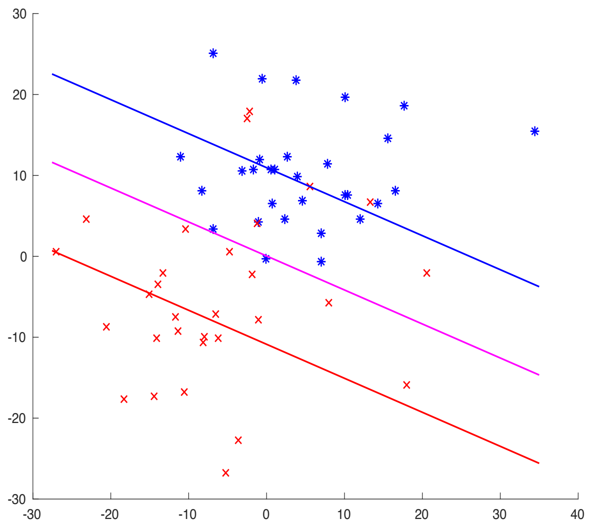


Fig. 18.17 Running (SVM$_{s3}$) on two sets of 30 points; $\nu = 0.5$.

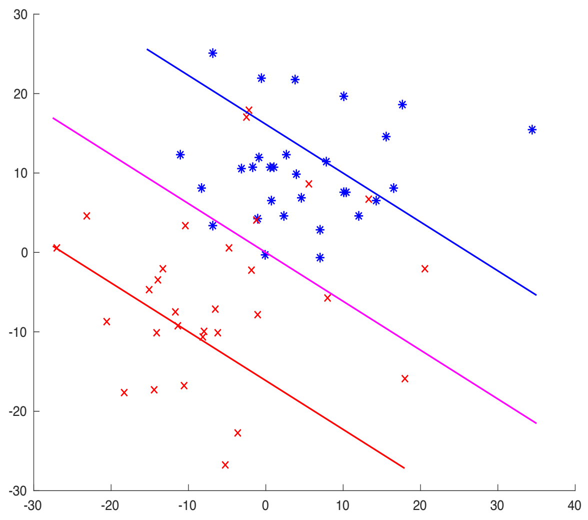


Fig. 18.18 Running (SVM$_{s3}$) on two sets of 30 points; $\nu = 0.71$.

Finally we ran our program with $\nu = 0.98$, see Figure 18.19. We have $p_f = 28, q_f = 30, p_m = 29$ and $q_m = 30$.

Because the term $(1/2)b^2$ is added to the objective function to be minimized, it turns out that (SVM$_{s3}$) yields values of b and η that are smaller than the values returned by (SVM$_{s2'}$). This is the reason why a smaller margin width could be obtained for $\nu = 0.365$. On the other hand, (SVM$_{s3}$) is unable to achieve as big a margin as (SVM$_{s2'}$) for values of $\nu \geq 0.97$, because the separating line produced by (SVM$_{s3}$) is lower than the the separating line produced by (SVM$_{s2'}$).

18.13 Soft Margin SVM; (SVM$_{s4}$)

In this section we consider the version of Problem (SVM$_{s2'}$) in which instead of using the function $K\left(\sum_{i=1}^{p} \epsilon_i + \sum_{j=1}^{q} \xi_j\right)$ as a regularizing function we use the quadratic function $K(\|\epsilon\|_2^2 + \|\xi\|_2^2)$.

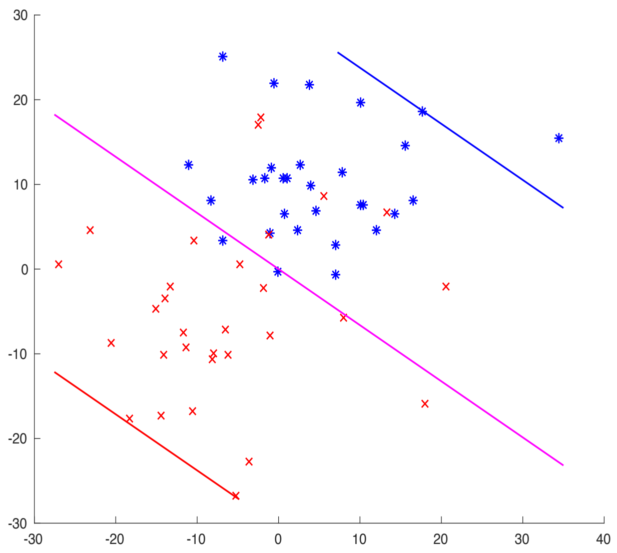


Fig. 18.19 Running (SVM$_{s3}$) on two sets of 30 points; $\nu = 0.98$.

Soft margin SVM (SVM$_{s4}$):

$$\begin{aligned}&\text{minimize} \quad \frac{1}{2}w^\top w + (p+q)K_s\left(-\nu\eta + \frac{1}{p+q}(\epsilon^\top\epsilon + \xi^\top\xi)\right)\\ &\text{subject to}\\ &\qquad w^\top u_i - b \geq \eta - \epsilon_i, \qquad i = 1,\ldots,p\\ &\quad -w^\top v_j + b \geq \eta - \xi_j, \qquad j = 1,\ldots,q\\ &\qquad\qquad \eta \geq 0,\end{aligned}$$

where ν and K_s are two given positive constants. As we saw earlier, theoretically, it is convenient to pick $K_s = 1/(p+q)$. When writing a computer program, it is preferable to assume that K_s is arbitrary. In this case ν needs to be replaced by $(p+q)K_s\nu$ in all the formulae obtained with $K_s = 1/(p+q)$.

The new twist with this formulation of the problem is that if $\epsilon_i < 0$, then the corresponding inequality $w^\top u_i - b \geq \eta - \epsilon_i$ implies the inequality $w^\top u_i - b \geq \eta$ obtained by setting ϵ_i to zero while reducing the value of $\|\epsilon\|^2$, and similarly if $\xi_j < 0$, then the corresponding inequality $-w^\top v_j + b \geq \eta - \xi_j$ implies the inequality $-w^\top v_j + b \geq \eta$ obtained by setting ξ_j to zero while reducing the value of $\|\xi\|^2$. Therefore, if (w, b, ϵ, ξ) is an optimal solution of Problem (SVM$_{s4}$), it is not necessary to restrict the slack variables ϵ_i and ξ_j to the nonnegative, which simplifies matters a bit. In fact, we will see that for an optimal solution, $\epsilon = \lambda/(2K_s)$ and $\xi = \mu/(2K_s)$. The variable η can also be determined by expressing that the duality gap is zero.

One of the advantages of this methods is that ϵ is determined by λ, ξ is determined by μ, and η and b are determined by λ and μ. This method *does not* require support vectors to compute b. We can omit the constraint $\eta \geq 0$, because for an optimal solution it can be shown using duality that $\eta \geq 0$; see Section 18.14.

A drawback of Program (SVM$_{s4}$) is that for fixed K_s, the quantity $\delta = \eta/\|w\|$ and the hyperplanes $H_{w,b}, H_{w,b+\eta}$ and $H_{w,b-\eta}$ are independent of ν. This will be shown in Theorem 18.3. Thus this method is less flexible than (SVM$_{s2'}$) and (SVM$_{s3}$).

The Lagrangian is given by

$$\begin{aligned}
L(w, \epsilon, \xi, b, \eta, \lambda, \mu, \gamma) &= \frac{1}{2} w^\top w - \nu\eta + K_s(\epsilon^\top \epsilon + \xi^\top \xi) + w^\top X \begin{pmatrix} \lambda \\ \mu \end{pmatrix} - \gamma\eta \\
&\quad - \epsilon^\top \lambda - \xi^\top \mu + b(\mathbf{1}_p^\top \lambda - \mathbf{1}_q^\top \mu) + \eta(\mathbf{1}_p^\top \lambda + \mathbf{1}_q^\top \mu) \\
&= \frac{1}{2} w^\top w + w^\top X \begin{pmatrix} \lambda \\ \mu \end{pmatrix} + \eta(\mathbf{1}_p^\top \lambda + \mathbf{1}_q^\top \mu - \nu - \gamma) \\
&\quad + K_s(\epsilon^\top \epsilon + \xi^\top \xi) - \epsilon^\top \lambda - \xi^\top \mu + b(\mathbf{1}_p^\top \lambda - \mathbf{1}_q^\top \mu).
\end{aligned}$$

To find the dual function $G(\lambda, \mu, \gamma)$ we minimize $L(w, \epsilon, \xi, b, \eta, \lambda, \mu, \gamma)$ with respect to w, ϵ, ξ, b, and η. Since the Lagrangian is convex and $(w, \epsilon, \xi, b, \eta) \in \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^q \times \mathbb{R} \times \mathbb{R}$, a convex open set, by Theorem 4.5, the Lagrangian has a minimum in $(w, \epsilon, \xi, b, \eta)$ iff $\nabla L_{w,\epsilon,\xi,b,\eta} = 0$, so we compute $\nabla L_{w,\epsilon,\xi,b,\eta}$. The gradient $\nabla L_{w,\epsilon,\xi,b,\eta}$ is given by

$$\nabla L_{w,\epsilon,\xi,b,\eta} = \begin{pmatrix} w + X \begin{pmatrix} \lambda \\ \mu \end{pmatrix} \\ 2K_s\epsilon - \lambda \\ 2K_s\xi - \mu \\ \mathbf{1}_p^\top \lambda - \mathbf{1}_q^\top \mu \\ \mathbf{1}_p^\top \lambda + \mathbf{1}_q^\top \mu - \nu - \gamma \end{pmatrix}.$$

By setting $\nabla L_{w,\epsilon,\xi,b,\eta} = 0$ we get the equations

$$\begin{aligned} w &= -X \begin{pmatrix} \lambda \\ \mu \end{pmatrix}, & (*_w) \\ 2K_s \epsilon &= \lambda \\ 2K_s \xi &= \mu \\ \mathbf{1}_p^\top \lambda &= \mathbf{1}_q^\top \mu \\ \mathbf{1}_p^\top \lambda + \mathbf{1}_q^\top \mu &= \nu + \gamma. \end{aligned}$$

The last two equations are identical to the last two equations obtained in Problem (SVM$_{s2'}$). We can use the other equations to obtain the following expression for the dual function $G(\lambda, \mu, \gamma)$,

$$\begin{aligned} G(\lambda, \mu, \gamma) &= -\frac{1}{4K_s}(\lambda^\top \lambda + \mu^\top \mu) - \frac{1}{2} \begin{pmatrix} \lambda^\top & \mu^\top \end{pmatrix} X^\top X \begin{pmatrix} \lambda \\ \mu \end{pmatrix} \\ &= -\frac{1}{2} \begin{pmatrix} \lambda^\top & \mu^\top \end{pmatrix} \left(X^\top X + \frac{1}{2K_s} I_{p+q} \right) \begin{pmatrix} \lambda \\ \mu \end{pmatrix}. \end{aligned}$$

Consequently the dual program is equivalent to the minimization program

Dual of the Soft margin SVM (SVM$_{s4}$):

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2} \begin{pmatrix} \lambda^\top & \mu^\top \end{pmatrix} \left(X^\top X + \frac{1}{2K_s} I_{p+q} \right) \begin{pmatrix} \lambda \\ \mu \end{pmatrix} \\ &\text{subject to} \\ &\qquad \sum_{i=1}^p \lambda_i - \sum_{j=1}^q \mu_j = 0 \\ &\qquad \sum_{i=1}^p \lambda_i + \sum_{j=1}^q \mu_j \geq \nu \\ &\qquad \lambda_i \geq 0, \quad i = 1, \ldots, p \\ &\qquad \mu_j \geq 0, \quad j = 1, \ldots, q. \end{aligned}$$

The above program is similar to the program that was obtained for Problem (SVM$_{s2'}$) but the matrix $X^\top X$ is replaced by the matrix $X^\top X + (1/2K_s) I_{p+q}$, which is positive definite since $K_s > 0$, and also the inequalities $\lambda_i \leq K_s$ and $\mu_j \leq K_s$ no longer hold.

It is shown in Section 18.14 how the dual program is solved using ADMM from Section 16.6. If the primal problem is solvable, this yields solutions for λ and μ. We obtain w from λ and μ, as in Problem (SVM$_{s2'}$); namely,

$$w = -X \begin{pmatrix} \lambda \\ \mu \end{pmatrix} = \sum_{i=1}^{p} \lambda_i u_i - \sum_{j=1}^{q} \mu_j v_j.$$

Since the variables ϵ_i and ξ_j are not restricted to be nonnegative we no longer have complementary slackness conditions involving them, but we know that

$$\epsilon = \frac{\lambda}{2K_s}, \quad \xi = \frac{\mu}{2K_s}.$$

Also since the constraints

$$\sum_{i=1}^{p} \lambda_i \geq \frac{\nu}{2} \quad \text{and} \quad \sum_{j=1}^{q} \mu_j \geq \frac{\nu}{2}$$

imply that there is some i_0 such that $\lambda_{i_0} > 0$ and some j_0 such that $\mu_{j_0} > 0$, we have $\epsilon_{i_0} > 0$ and $\xi_{j_0} > 0$, which means that at least two points are misclassified, so Problem (SVM$_{s4}$) should only be used when the sets $\{u_i\}$ and $\{v_j\}$ are *not* linearly separable.

Because $\epsilon_i = \lambda_i/(2K_s)$, $\xi_j = \mu_j/(2K_s)$, and there is no upper bound K_s on λ_i and μ_j, the classification of the points is simpler than in the previous cases.

(1) If $\lambda_i = 0$, then $\epsilon_i = 0$ and the inequality $w^\top u_i - b - \eta \geq 0$ holds. If equality holds then u_i is a support vector on the blue margin (the hyperplane $H_{w,b+\eta}$). Otherwise u_i is in the blue open half-space bounded by the margin hyperplane $H_{w,b+\eta}$ (not containing the separating hyperplane $H_{w,b}$). See Figure 18.20.
Similarly, if $\mu_j = 0$, then $\xi_j = 0$ and the inequality $-w^\top v_j + b - \eta \geq$ holds. If equality holds then v_j is a support vector on the red margin (the hyperplane $H_{w,b-\eta}$). Otherwise v_j is in the red open half-space bounded by the margin hyperplane $H_{w,b-\eta}$ (not containing the separating hyperplane $H_{w,b}$). See Figure 18.20.

(2) If $\lambda_i > 0$, then $\epsilon_i = \lambda_i/(2K_s) > 0$. The corresponding constraint is active, so we have

$$w^\top u_i - b = \eta - \epsilon_i.$$

If $\epsilon_i \leq \eta$, then the points u_i is inside the slab bounded by the blue margin hyperplane $H_{w,b+\eta}$ and the separating hyperplane $H_{w,b}$. If $\epsilon_i > \eta$, then the point u_i belongs to the open half-space bounded by the separating hyperplane and containing the red margin hyperplane (the red side); it is misclassified. See Figure 18.21.

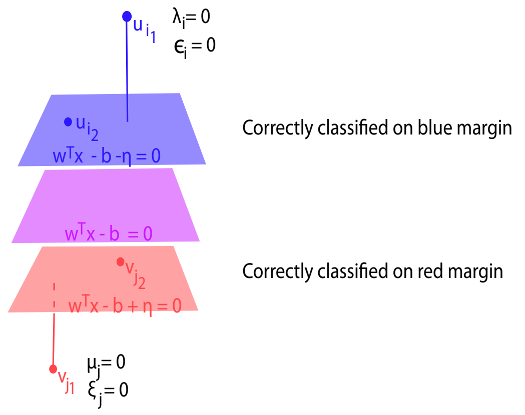


Fig. 18.20 When $\lambda_i = 0$, u_i is correctly classified on or outside the blue margin. When $\mu_j = 0$, v_j is correctly classified on or outside outside the red margin.

Similarly, if $\mu_j > 0$, then $\xi_j = \mu_j/(2K_s) > 0$. The corresponding constraint is active, so we have

$$-w^\top v_j + b = \eta - \xi_j.$$

If $\xi_j \leq \eta$, then the points v_j is inside the slab bounded by the red margin hyperplane $H_{w,b-\eta}$ and the separating hyperplane $H_{w,b}$. If $\xi_j > \eta$, then the point v_j belongs to the open half-space bounded by the separating hyperplane and containing the blue margin hyperplane (the blue side); it is misclassified. See Figure 18.21.

We can use the fact that the duality gap is 0 to find η. We have

$$\frac{1}{2}w^\top w - \nu\eta + K_s(\epsilon^\top\epsilon + \xi^\top\xi) = -\frac{1}{2}\begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix}\left(X^\top X + \frac{1}{2K_s}I_{p+q}\right)\begin{pmatrix}\lambda \\ \mu\end{pmatrix},$$

and since

$$w = -X\begin{pmatrix}\lambda \\ \mu\end{pmatrix}$$

we get

$$\begin{aligned}\nu\eta &= K_s(\epsilon^\top\epsilon + \xi^\top\xi) + \begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix}\left(X^\top X + \frac{1}{4K_s}I_{p+q}\right)\begin{pmatrix}\lambda \\ \mu\end{pmatrix} \\ &= \begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix}\left(X^\top X + \frac{1}{2K_s}I_{p+q}\right)\begin{pmatrix}\lambda \\ \mu\end{pmatrix}.\end{aligned}$$

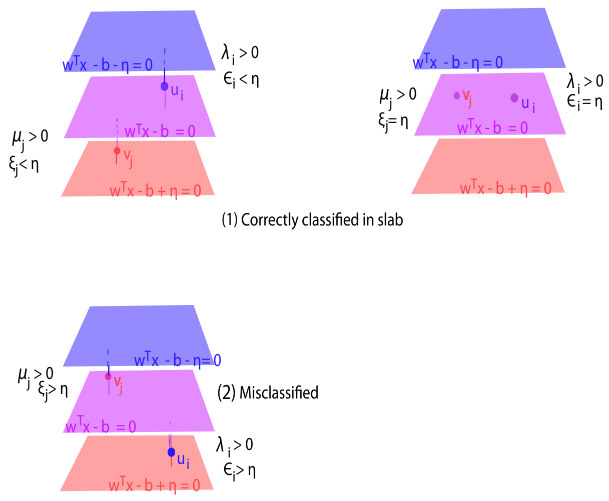


Fig. 18.21 The classification of points for SVM$_{s4}$ when the Lagrange multipliers are positive. The left illustration of Figure (1) is when u_i is inside the margin yet still on the correct side of the separating hyperplane $w^\top x - b = 0$. Similarly, v_j is inside the margin on the correct side of the separating hyperplane. The right illustration depicts u_i and v_j on the separating hyperplane. Figure (2) illustrations a misclassification of u_i and v_j.

The above confirms that at optimality we have $\eta \geq 0$.

Remark: If we do not assume that $K_s = 1/(p+q)$, then the above formula must be replaced by

$$(p+q)K_s\nu\eta = \begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix} \left(X^\top X + \frac{1}{2K_s} I_{p+q}\right) \begin{pmatrix}\lambda \\ \mu\end{pmatrix}.$$

Since η is determined independently of the existence of support vectors, the margin hyperplane $H_{w,b+\eta}$ may not contain any point u_i and the margin hyperplane $H_{w,b-\eta}$ may not contain any point v_j.

We can solve for b using some active constraint corresponding to any i_0 such that $\lambda_{i_0} > 0$ and any j_0 such that $\mu_{j_0} > 0$ (by a previous remark, the constraints imply that such i_0 and j_0 must exist). To improve numerical stability we average over the following sets of indices. Let I_λ and I_μ be the set of indices given by

$$\begin{aligned} I_\lambda &= \{i \in \{1,\ldots,p\} \mid \lambda_i > 0\} \\ I_\mu &= \{j \in \{1,\ldots,q\} \mid \mu_j > 0\}, \end{aligned}$$

and let $p_m = |I_\lambda|$ and $q_m = |I_\mu|$. We obtain the formula

$$\begin{aligned} b = w^\top \bigg(\Big(\sum_{i \in I_\lambda} u_i\Big)/p_m + \Big(\sum_{j \in I_\mu} v_j\Big)/q_m\bigg)/2 \\ + \bigg(\Big(\sum_{i \in I_\lambda} \epsilon_i\Big)/p_m - \Big(\sum_{j \in I_\mu} \xi_j\Big)/q_m\bigg)/2. \end{aligned}$$

We now prove that for a fixed K_s, the solution to Problem (SVM$_{s4}$) is unique and independent of the value of ν.

Theorem 18.3. *For K_s and ν fixed, if Problem* (SVM$_{s4}$) *succeeds, then it has a unique solution. If Problem* (SVM$_{s4}$) *succeeds and returns $(\lambda, \mu, \eta, w, b)$ for the value ν and $(\lambda^\kappa, \mu^\kappa, \eta^\kappa, w^\kappa, b^\kappa)$ for the value $\kappa\nu$ with $\kappa > 0$, then*

$$\lambda^\kappa = \kappa\lambda, \;\; \mu^\kappa = \kappa\mu, \;\; \eta^\kappa = \kappa\eta, \;\; w^\kappa = \kappa w, \;\; b^\kappa = \kappa b.$$

As a consequence, $\delta = \eta/\left\|w\right\| = \eta^\kappa/\left\|w^\kappa\right\| = \delta^\kappa$, and the hyperplanes $H_{w,b}, H_{w,b+\eta}$ and $H_{w,b-\eta}$ are independent of ν.

Proof. We already observed that for an optimal solution with $\eta > 0$, we have $\gamma = 0$. This means that (λ, μ) is a solution of the problem

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2}\begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix}\left(X^\top X + \frac{1}{2K_s} I_{p+q}\right)\begin{pmatrix}\lambda \\ \mu\end{pmatrix} \\ &\text{subject to} \\ &\qquad \sum_{i=1}^p \lambda_i - \sum_{j=1}^q \mu_j = 0 \\ &\qquad \sum_{i=1}^p \lambda_i + \sum_{j=1}^q \mu_j = \nu \\ &\qquad \lambda_i \geq 0, \quad i = 1, \ldots, p \\ &\qquad \mu_j \geq 0, \quad j = 1, \ldots, q. \end{aligned}$$

Since $K_s > 0$ and $X^\top X$ is symmetric positive semidefinite, the matrix $P = X^\top X + \frac{1}{2K_s} I_{p+q}$ is symmetric positive definite. Let $\Omega = \mathbb{R}^{p+q}$ and let U be the convex set given by

$$U = \left\{ \begin{pmatrix}\lambda \\ \mu\end{pmatrix} \in \mathbb{R}^{p+q}_+ \;\middle|\; \begin{pmatrix}\mathbf{1}_p^\top & -\mathbf{1}_q^\top \\ \mathbf{1}_p^\top & \mathbf{1}_q^\top\end{pmatrix}\begin{pmatrix}\lambda \\ \mu\end{pmatrix} = \begin{pmatrix}0 \\ (p+q)K_s\nu\end{pmatrix} \right\}.$$

Since the matrix P is symmetric positive definite, the functional

$$F(\lambda,\mu) = -G(\lambda,\mu) = \frac{1}{2}\begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix} P \begin{pmatrix}\lambda \\ \mu\end{pmatrix}$$

is strictly convex and U is convex, so by Theorem 4.5(2,4), if it has a minimum, then it is unique. Consider the convex set

$$U^\kappa = \left\{ \begin{pmatrix}\lambda \\ \mu\end{pmatrix} \in \mathbb{R}^{p+q}_+ \;\middle|\; \begin{pmatrix}\mathbf{1}_p^\top & -\mathbf{1}_q^\top \\ \mathbf{1}_p^\top & \mathbf{1}_q^\top\end{pmatrix} \begin{pmatrix}\lambda \\ \mu\end{pmatrix} = \begin{pmatrix}0 \\ (p+q)K_s\kappa\nu\end{pmatrix} \right\}.$$

Observe that

$$\kappa U = \left\{ \begin{pmatrix}\kappa\lambda \\ \kappa\mu\end{pmatrix} \in \mathbb{R}^{p+q}_+ \;\middle|\; \begin{pmatrix}\mathbf{1}_p^\top & -\mathbf{1}_q^\top \\ \mathbf{1}_p^\top & \mathbf{1}_q^\top\end{pmatrix} \begin{pmatrix}\kappa\lambda \\ \kappa\mu\end{pmatrix} = \begin{pmatrix}0 \\ (p+q)K_s\kappa\nu\end{pmatrix} \right\} = U^\kappa.$$

By Theorem 4.5(3), $(\lambda,\mu) \in U$ is a minimum of F over U iff

$$dF_{\lambda,\mu}\begin{pmatrix}\lambda' - \lambda \\ \mu' - \mu\end{pmatrix} \geq 0 \quad \text{for all} \quad \begin{pmatrix}\lambda' \\ \mu'\end{pmatrix} \in U.$$

Since

$$dF_{\lambda,\mu}\begin{pmatrix}\lambda' - \lambda \\ \mu' - \mu\end{pmatrix} = \begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix} P \begin{pmatrix}\lambda' - \lambda \\ \mu' - \mu\end{pmatrix}$$

the above conditions are equivalent to

$$\begin{aligned}
\begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix} P \begin{pmatrix}\lambda' - \lambda \\ \mu' - \mu\end{pmatrix} &\geq 0 \\
\begin{pmatrix}\mathbf{1}_p^\top & -\mathbf{1}_q^\top \\ \mathbf{1}_p^\top & \mathbf{1}_q^\top\end{pmatrix} \begin{pmatrix}\lambda \\ \mu\end{pmatrix} &= \begin{pmatrix}0 \\ (p+q)K_s\nu\end{pmatrix} \\
\lambda, \lambda' \in \mathbb{R}^p_+,\ \mu, \mu' &\in \mathbb{R}^q_+.
\end{aligned}$$

Since $\kappa > 0$, by multiplying the above inequality by κ^2 and the equations by κ, the following conditions hold:

$$\begin{aligned}
\begin{pmatrix}\kappa\lambda^\top & \kappa\mu^\top\end{pmatrix} P \begin{pmatrix}\kappa\lambda' - \kappa\lambda \\ \kappa\mu' - \kappa\mu\end{pmatrix} &\geq 0 \\
\begin{pmatrix}\mathbf{1}_p^\top & -\mathbf{1}_q^\top \\ \mathbf{1}_p^\top & \mathbf{1}_q^\top\end{pmatrix} \begin{pmatrix}\kappa\lambda \\ \kappa\mu\end{pmatrix} &= \begin{pmatrix}0 \\ (p+q)K_s\kappa\nu\end{pmatrix} \\
\kappa\lambda, \kappa\lambda' \in \mathbb{R}^p_+,\ \kappa\mu, \kappa\mu' &\in \mathbb{R}^q_+.
\end{aligned}$$

By Theorem 4.5(3), $(\kappa\lambda, \kappa\mu) \in U^\kappa$ is a minimum of F over U^κ, and because F is strictly convex and U^κ is convex, if F has a minimum over U^κ, then $(\kappa\lambda, \kappa\mu) \in U^\kappa$ is the unique minimum. Therefore, $\lambda^\kappa = \kappa\lambda,\ \mu^\kappa = \kappa\mu$.

Since w is given by the equation

$$w = -X \begin{pmatrix} \lambda \\ \mu \end{pmatrix}$$

and since we just showed that $\lambda^\kappa = \kappa\lambda$, $\mu^\kappa = \kappa\mu$, we deduce that $w^\kappa = \kappa w$.

We showed earlier that η is given by the equation

$$(p+q)K_s\nu\eta = \begin{pmatrix} \lambda^\top & \mu^\top \end{pmatrix} \left(X^\top X + \frac{1}{2K_s} I_{p+q} \right) \begin{pmatrix} \lambda \\ \mu \end{pmatrix}.$$

If we replace ν by $\kappa\nu$, since λ is replaced by $\kappa\lambda$ and μ by $\kappa\nu$, we see that $\eta^\kappa = \kappa\eta$. Finally, b is given by the equation

$$b = \frac{w^\top(u_{i_0} + v_{j_0}) + \epsilon_{i_0} - \xi_{j_0}}{2}$$

for and i_0 such that $\lambda_{i_0} > 0$ and any j_0 such that $\mu_{j_0} > 0$. If λ is replaced by $\kappa\lambda$ and μ by $\kappa\mu$, since $\epsilon = \lambda/(2K_s)$ and $\xi = \mu/(2K_s)$, we see that ϵ is replaced by $\kappa\epsilon$ and ξ by $\kappa\xi$, so $b^\kappa = \kappa b$.

Since $w^\kappa = \kappa w$ and $\eta^\kappa = \kappa\eta$ we obtain $\delta = \eta/\left\|w\right\| = \eta^\kappa/\left\|w^\kappa\right\| = \delta^\kappa$. Since $w^\kappa = \kappa w$, $\eta^\kappa = \kappa\eta$ and $b^\kappa = \kappa b$, the normalized equations of the hyperplanes $H_{w,b}, H_{w,b+\eta}$ and $H_{w,b-\eta}$ (obtained by dividing by $\left\|w\right\|$) are all identical, so the hyperplanes $H_{w,b}, H_{w,b+\eta}$ and $H_{w,b-\eta}$ are independent of ν. $\square$

The width of the slab is controlled by K. The larger K is the smaller is the width of the slab. Theoretically, since this method does not rely on support vectors to compute b, it cannot fail if a solution exists, but in practice the quadratic solver does not converge for values of K that are too large. However, the method handles very small values of K, which can yield slabs of excessive width.

The "kernelized" version of Problem (SVM$_{s4}$) is the following:

Soft margin kernel SVM (SVM$_{s4}$):

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2}\langle w, w\rangle - \nu\eta + K_s(\epsilon^\top\epsilon + \xi^\top\xi) \\ &\text{subject to} \\ &\qquad \langle w, \varphi(u_i)\rangle - b \geq \eta - \epsilon_i, \qquad i = 1, \ldots, p \\ &\qquad -\langle w, \varphi(v_j)\rangle + b \geq \eta - \xi_j, \qquad j = 1, \ldots, q \\ &\qquad \eta \geq 0, \end{aligned}$$

with $K_s = 1/(p+q)$.

By going over the derivation of the dual program, we obtain

Dual of the Soft margin kernel SVM (SVM$_{s4}$):

$$\begin{aligned}
&\text{minimize} \quad \frac{1}{2}\begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix}\left(\mathbf{K} + \frac{1}{2K_s} I_{p+q}\right)\begin{pmatrix}\lambda \\ \mu\end{pmatrix}\\
&\text{subject to}\\
&\qquad \sum_{i=1}^p \lambda_i - \sum_{j=1}^q \mu_j = 0\\
&\qquad \sum_{i=1}^p \lambda_i + \sum_{j=1}^q \mu_j \geq \nu\\
&\qquad \lambda_i \geq 0, \quad i = 1, \ldots, p\\
&\qquad \mu_j \geq 0, \quad j = 1, \ldots, q,
\end{aligned}$$

where $\mathbf{K}$ is the kernel matrix of Section 18.1. Then w, b, and $f(x)$ are obtained exactly as in Section 18.5.

18.14 Solving SVM (SVM$_{s4}$) Using ADMM

In order to solve (SVM$_{s4}$) using ADMM we need to write the matrix corresponding to the constraints in equational form,

$$\begin{aligned}
\sum_{i=1}^p \lambda_i - \sum_{j=1}^q \mu_j &= 0\\
\sum_{i=1}^p \lambda_i + \sum_{j=1}^q \mu_j - \gamma &= K_m,
\end{aligned}$$

with $K_m = (p+q)K_s\nu$. This is the $2 \times (p+q+1)$ matrix A given by

$$A = \begin{pmatrix} \mathbf{1}_p^\top & -\mathbf{1}_q^\top & 0 \\ \mathbf{1}_p^\top & \mathbf{1}_q^\top & -1 \end{pmatrix}.$$

We leave it as an exercise to prove that A has rank 2. The right-hand side is

$$c = \begin{pmatrix} 0 \\ K_m \end{pmatrix}.$$

The symmetric positive semidefinite $(p+q)\times(p+q)$ matrix P defining the quadratic functional is

$$P = X^\top X + \frac{1}{2K_s} I_{p+q}, \quad \text{with} \quad X = \begin{pmatrix} -u_1 & \cdots & -u_p & v_1 & \cdots & v_q \end{pmatrix},$$

and

$$q = 0_{p+q}.$$

Since there are $p+q+1$ Lagrange multipliers (λ, μ, γ), the $(p+q)\times(p+q)$ matrix P must be augmented with zero's to make it a $(p+q+1)\times(p+q+1)$ matrix P_a given by

$$P_a = \begin{pmatrix} X^\top X & 0_{p+q} \\ 0_{p+q}^\top & 0 \end{pmatrix},$$

and similarly q is augmented with zeros as the vector $q_a = 0_{p+q+1}$.

As in Section 18.8, since $\eta \geq 0$ for an optimal solution, we can drop the constraint $\eta \geq 0$ from the primal problem. In this case, there are $p+q$ Lagrange multipliers (λ, μ). It is easy to see that the objective function of the dual is unchanged and the set of constraints is

$$\begin{aligned} \sum_{i=1}^p \lambda_i - \sum_{j=1}^q \mu_j &= 0 \\ \sum_{i=1}^p \lambda_i + \sum_{j=1}^q \mu_j &= K_m, \end{aligned}$$

with $K_m = (p+q)K_s\nu$. The matrix corresponding to the above equations is the $2\times(p+q)$ matrix A_2 given by

$$A_2 = \begin{pmatrix} \mathbf{1}_p^\top & -\mathbf{1}_q^\top \\ \mathbf{1}_p^\top & \mathbf{1}_q^\top \end{pmatrix}.$$

We leave it as an exercise to prove that A_2 has rank 2. The right-hand side is

$$c_2 = \begin{pmatrix} 0 \\ K_m \end{pmatrix}.$$

The symmetric positive semidefinite $(p+q)\times(p+q)$ matrix P defining the quadratic functional is

$$P = X^\top X + \frac{1}{2K_s} I_{p+q}, \quad \text{with} \quad X = \begin{pmatrix} -u_1 & \cdots & -u_p & v_1 & \cdots & v_q \end{pmatrix},$$

and

$$q = 0_{p+q}.$$

Since there are $p+q$ Lagrange multipliers (λ, μ), the $(p+q)\times(p+q)$ matrix P need not be augmented with zero's, so $P_{2a} = P$ and similarly $q_{2a} = 0_{p+q}$.

We ran our `Matlab` implementation of the above version of (SVM$_{s4}$) on the data set of Section 18.12. Since the value of ν is irrelevant, we picked $\nu = 1$. First we ran our program with $K = 190$; see Figure 18.22. We have $p_m = 23$ and $q_m = 18$. The program does not converge for $K \geq 200$.

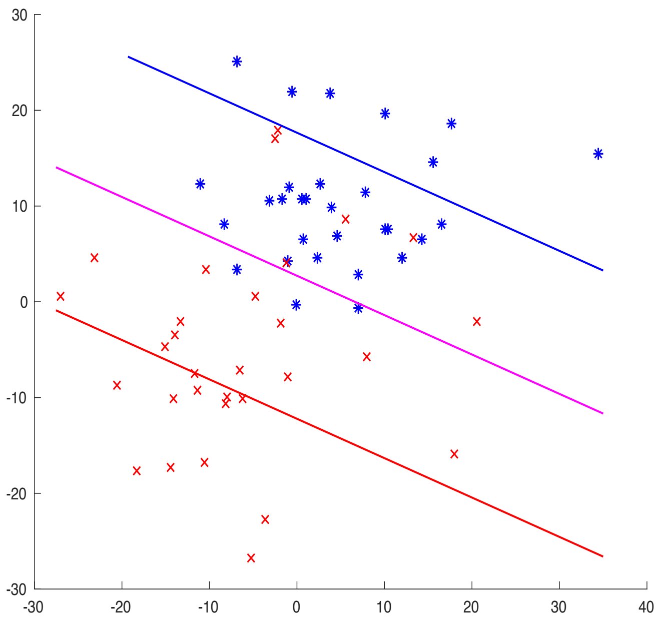


Fig. 18.22 Running (SVM$_{s4}$) on two sets of 30 points; $K = 190$.

Our second run was made with $K = 1/12000$; see Figure 18.23. We have $p_m = 30$ and $q_m = 30$ and we see that the width of the slab is a bit excessive. This example demonstrates that the margin lines need not contain data points.

18.15 Soft Margin SVM; (SVM$_{s5}$)

In this section we consider the version of Problem (SVM$_{s4}$) in which we add the term $(1/2)b^2$ to the objective function. We also drop the constraint $\eta \geq 0$ which is redundant.

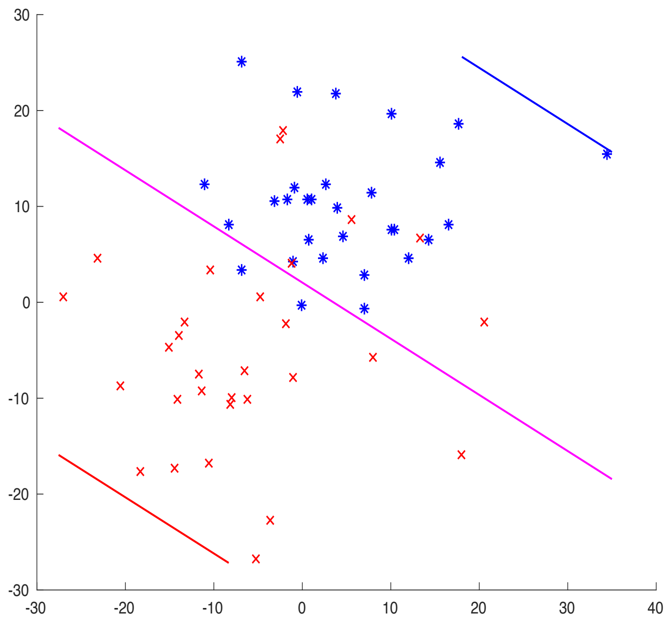


Fig. 18.23 Running (SVM$_{s4}$) on two sets of 30 points; $K = 1/12000$.

Soft margin SVM (SVM$_{s5}$):

$$\begin{aligned}
&\text{minimize} \quad \frac{1}{2}w^\top w + \frac{1}{2}b^2 + (p+q)K_s\left(-\nu\eta + \frac{1}{p+q}(\epsilon^\top\epsilon + \xi^\top\xi)\right)\\
&\text{subject to}\\
&\qquad w^\top u_i - b \geq \eta - \epsilon_i, \qquad i = 1,\ldots,p\\
&\qquad -w^\top v_j + b \geq \eta - \xi_j, \qquad j = 1,\ldots,q,
\end{aligned}$$

where ν and K_s are two given positive constants. As we saw earlier, it is convenient to pick $K_s = 1/(p+q)$. When writing a computer program, it is preferable to assume that K_s is arbitrary. In this case ν must be replaced by $(p+q)K_s\nu$ in all the formulae.

One of the advantages of this methods is that ϵ is determined by λ, ξ is determined by μ (as in (SVM$_{s4}$)), and both η and b determined by λ and μ. As the previous method, this method *does not* require support vectors to compute b. We can omit the constraint $\eta \geq 0$, because for an optimal solution it can be shown using duality that $\eta \geq 0$.

A drawback of Program (SVM$_{s5}$) is that for fixed K_s, the quantity $\delta = \eta/\left\|w\right\|$ and the hyperplanes $H_{w,b}, H_{w,b+\eta}$ and $H_{w,b-\eta}$ are independent of ν. This will be shown in Theorem 18.4. Thus this method is less flexible than (SVM$_{s2'}$) and (SVM$_{s3}$).

The Lagrangian is given by

$$\begin{aligned}L(w,\epsilon,\xi,b,\eta,\lambda,\mu) &= \frac{1}{2}w^\top w + \frac{1}{2}b^2 - \nu\eta + K_s(\epsilon^\top\epsilon + \xi^\top\xi) + w^\top X\begin{pmatrix}\lambda\\ \mu\end{pmatrix}\\ &\quad - \epsilon^\top\lambda - \xi^\top\mu + b(\mathbf{1}_p^\top\lambda - \mathbf{1}_q^\top\mu) + \eta(\mathbf{1}_p^\top\lambda + \mathbf{1}_q^\top\mu)\\ &= \frac{1}{2}w^\top w + w^\top X\begin{pmatrix}\lambda\\ \mu\end{pmatrix} + \eta(\mathbf{1}_p^\top\lambda + \mathbf{1}_q^\top\mu - \nu) + \frac{1}{2}b^2\\ &\quad + K_s(\epsilon^\top\epsilon + \xi^\top\xi) - \epsilon^\top\lambda - \xi^\top\mu + b(\mathbf{1}_p^\top\lambda - \mathbf{1}_q^\top\mu).\end{aligned}$$

To find the dual function $G(\lambda,\mu)$ we minimize $L(w,\epsilon,\xi,b,\eta,\lambda,\mu)$ with respect to w,ϵ,ξ, b, and η. Since the Lagrangian is convex and $(w,\epsilon,\xi,b,\eta)\in\mathbb{R}^n\times\mathbb{R}^p\times\mathbb{R}^q\times\mathbb{R}\times\mathbb{R}$, a convex open set, by Theorem 4.5, the Lagrangian has a minimum in (w,ϵ,ξ,b,η) iff $\nabla L_{w,\epsilon,\xi,b,\eta} = 0$, so we compute $\nabla L_{w,\epsilon,\xi,b,\eta}$. The gradient $\nabla L_{w,\epsilon,\xi,b,\eta}$ is given by

$$\nabla L_{w,\epsilon,\xi,b,\eta} = \begin{pmatrix} w + X\begin{pmatrix}\lambda\\ \mu\end{pmatrix}\\ 2K_s\epsilon - \lambda\\ 2K_s\xi - \mu\\ b + \mathbf{1}_p^\top\lambda - \mathbf{1}_q^\top\mu\\ \mathbf{1}_p^\top\lambda + \mathbf{1}_q^\top\mu - \nu\end{pmatrix}.$$

By setting $\nabla L_{w,\epsilon,\xi,b,\eta} = 0$ we get the equations

$$\begin{aligned} w &= -X\begin{pmatrix}\lambda\\ \mu\end{pmatrix}, && (*_w)\\ 2K_s\epsilon &= \lambda\\ 2K_s\xi &= \mu\\ b &= -(\mathbf{1}_p^\top\lambda - \mathbf{1}_q^\top\mu)\\ \mathbf{1}_p^\top\lambda + \mathbf{1}_q^\top\mu &= \nu.\end{aligned}$$

As we said earlier, both w an b are determined by λ and μ. We can use the equations to obtain the following expression for the dual function $G(\lambda,\mu,\gamma)$,

$$\begin{aligned}G(\lambda,\mu,\gamma) &= -\frac{1}{4K_s}(\lambda^\top\lambda + \mu^\top\mu) - \frac{1}{2}\begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix}X^\top X\begin{pmatrix}\lambda\\ \mu\end{pmatrix} - \frac{b^2}{2}\\ &= -\frac{1}{2}\begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix}\left(X^\top X + \begin{pmatrix}\mathbf{1}_p\mathbf{1}_p^\top & -\mathbf{1}_p\mathbf{1}_q^\top\\ -\mathbf{1}_q\mathbf{1}_p^\top & \mathbf{1}_q\mathbf{1}_q^\top\end{pmatrix} + \frac{1}{2K_s}I_{p+q}\right)\begin{pmatrix}\lambda\\ \mu\end{pmatrix}.\end{aligned}$$

Consequently the dual program is equivalent to the minimization program

Dual of the Soft margin SVM (SVM$_{s5}$):

$$\begin{aligned}
&\text{minimize} \quad \frac{1}{2}\begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix}\left(X^\top X + \begin{pmatrix}\mathbf{1}_p\mathbf{1}_p^\top & -\mathbf{1}_p\mathbf{1}_q^\top \\ -\mathbf{1}_q\mathbf{1}_p^\top & \mathbf{1}_q\mathbf{1}_q^\top\end{pmatrix} + \frac{1}{2K_s}I_{p+q}\right)\begin{pmatrix}\lambda \\ \mu\end{pmatrix}\\
&\text{subject to}\\
&\qquad \sum_{i=1}^p \lambda_i + \sum_{j=1}^q \mu_j = \nu\\
&\qquad \lambda_i \geq 0, \quad i = 1, \ldots, p\\
&\qquad \mu_j \geq 0, \quad j = 1, \ldots, q.
\end{aligned}$$

It is shown in Section 18.16 how the dual program is solved using ADMM from Section 16.6. If the primal problem is solvable, this yields solutions for λ and μ.

The constraint

$$\sum_{i=1}^p \lambda_i + \sum_{j=1}^q \mu_j = \nu$$

implies that either there is some i_0 such that $\lambda_{i_0} > 0$ or there is some j_0 such that $\mu_{j_0} > 0$, so we have $\epsilon_{i_0} > 0$ or $\xi_{j_0} > 0$, which means that at least one point is misclassified. Thus Problem (SVM$_{s5}$) should only be used when the sets $\{u_i\}$ and $\{v_j\}$ are *not* linearly separable.

We can use the fact that the duality gap is 0 to find η. We have

$$\begin{aligned}
&\frac{1}{2}w^\top w + \frac{b^2}{2} - \nu\eta + K_s(\epsilon^\top\epsilon + \xi^\top\xi)\\
&\qquad = -\frac{1}{2}\begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix}\left(X^\top X + \begin{pmatrix}\mathbf{1}_p\mathbf{1}_p^\top & -\mathbf{1}_p\mathbf{1}_q^\top \\ -\mathbf{1}_q\mathbf{1}_p^\top & \mathbf{1}_q\mathbf{1}_q^\top\end{pmatrix} + \frac{1}{2K_s}I_{p+q}\right)\begin{pmatrix}\lambda \\ \mu\end{pmatrix},
\end{aligned}$$

so we get

$$\begin{aligned}
\nu\eta &= K_s(\epsilon^\top\epsilon + \xi^\top\xi)\\
&\quad + \begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix}\left(X^\top X + \begin{pmatrix}\mathbf{1}_p\mathbf{1}_p^\top & -\mathbf{1}_p\mathbf{1}_q^\top \\ -\mathbf{1}_q\mathbf{1}_p^\top & \mathbf{1}_q\mathbf{1}_q^\top\end{pmatrix} + \frac{1}{4K_s}I_{p+q}\right)\begin{pmatrix}\lambda \\ \mu\end{pmatrix}\\
&= \begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix}\left(X^\top X + \begin{pmatrix}\mathbf{1}_p\mathbf{1}_p^\top & -\mathbf{1}_p\mathbf{1}_q^\top \\ -\mathbf{1}_q\mathbf{1}_p^\top & \mathbf{1}_q\mathbf{1}_q^\top\end{pmatrix} + \frac{1}{2K_s}I_{p+q}\right)\begin{pmatrix}\lambda \\ \mu\end{pmatrix}.
\end{aligned}$$

The above confirms that at optimality we have $\eta \geq 0$.

Remark: If we do not assume that $K_s = 1/(p+q)$, then the above formula must be replaced by

$$(p+q)K_s\nu\eta = \begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix}\left(X^\top X + \begin{pmatrix}\mathbf{1}_p\mathbf{1}_p^\top & -\mathbf{1}_p\mathbf{1}_q^\top \\ -\mathbf{1}_q\mathbf{1}_p^\top & \mathbf{1}_q\mathbf{1}_q^\top\end{pmatrix} + \frac{1}{2K_s}I_{p+q}\right)\begin{pmatrix}\lambda \\ \mu\end{pmatrix}.$$

There is a version of Theorem 18.3 stating that for a fixed K_s, the solution to Problem (SVM$_{s5}$) is unique and independent of the value of ν.

Theorem 18.4. *For K_s and ν fixed, if Problem* (SVM$_{s5}$) *succeeds then it has a unique solution. If Problem* (SVM$_{s5}$) *succeeds and returns $(\lambda, \mu, \eta, w, b)$ for the value ν and $(\lambda^\kappa, \mu^\kappa, \eta^\kappa, w^\kappa, b^\kappa)$ for the value $\kappa\nu$ with $\kappa > 0$, then*

$$\lambda^\kappa = \kappa\lambda,\ \ \mu^\kappa = \kappa\mu,\ \ \eta^\kappa = \kappa\eta,\ \ w^\kappa = \kappa w,\ \ b^\kappa = \kappa b.$$

As a consequence, $\delta = \eta/\left\|w\right\| = \eta^\kappa/\left\|w^\kappa\right\| = \delta^\kappa$, and the hyperplanes $H_{w,b}, H_{w,b+\eta}$ and $H_{w,b-\eta}$ are independent of ν.

Proof. The proof is an easy adaptation of the proof of Theorem 18.3 so we only give a sketch. The two crucial points are that the matrix

$$P = X^\top X + \begin{pmatrix}\mathbf{1}_p\mathbf{1}_p^\top & -\mathbf{1}_p\mathbf{1}_q^\top \\ -\mathbf{1}_q\mathbf{1}_p^\top & \mathbf{1}_q\mathbf{1}_q^\top\end{pmatrix} + \frac{1}{2K_s}I_{p+q}$$

is symmetric positive definite and that we have the single equational constraint

$$\mathbf{1}_p^\top\lambda + \mathbf{1}_q^\top\mu = (p+q)K_s\nu$$

defining the convex set

$$U = \left\{\begin{pmatrix}\lambda \\ \mu\end{pmatrix} \in \mathbb{R}_+^{p+q} \mid \mathbf{1}_p^\top\lambda + \mathbf{1}_q^\top\mu = (p+q)K_s\nu\right\}.$$

The proof is essentially the proof of 18.3 using the above SPD matrix and convex set. $\square$

The "kernelized" version of Problem (SVM$_{s5}$) is the following:

Soft margin kernel SVM (SVM$_{s5}$):

$$\begin{aligned}&\text{minimize} \quad \frac{1}{2}\langle w, w\rangle + \frac{1}{2}b^2 - \nu\eta + K_s(\epsilon^\top\epsilon + \xi^\top\xi)\\ &\text{subject to}\\ &\qquad \langle w, \varphi(u_i)\rangle - b \geq \eta - \epsilon_i, \qquad i = 1, \ldots, p\\ &\qquad -\langle w, \varphi(v_j)\rangle + b \geq \eta - \xi_j, \qquad j = 1, \ldots, q,\end{aligned}$$

with $K_s = 1/(p+q)$.

Tracing through the derivation of the dual program, we obtain

Dual of the Soft margin kernel SVM (SVM$_{s5}$):

$$\begin{aligned}
&\text{minimize} \quad \frac{1}{2}\begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix}\left(\mathbf{K} + \begin{pmatrix}\mathbf{1}_p\mathbf{1}_p^\top & -\mathbf{1}_p\mathbf{1}_q^\top \\ -\mathbf{1}_q\mathbf{1}_p^\top & \mathbf{1}_q\mathbf{1}_q^\top\end{pmatrix} + \frac{1}{2K_s}I_{p+q}\right)\begin{pmatrix}\lambda \\ \mu\end{pmatrix} \\
&\text{subject to} \\
&\qquad \sum_{i=1}^p \lambda_i + \sum_{j=1}^q \mu_j = \nu \\
&\qquad \lambda_i \geq 0, \quad i = 1, \ldots, p \\
&\qquad \mu_j \geq 0, \quad j = 1, \ldots, q,
\end{aligned}$$

where $\mathbf{K}$ is the kernel matrix of Section 18.1. Then w, b, and $f(x)$ are obtained exactly as in Section 18.13.

18.16 Solving SVM (SVM$_{s5}$) Using ADMM

In order to solve (SVM$_5$) using ADMM we need to write the matrix corresponding to the constraints in equational form,

$$\sum_{i=1}^p \lambda_i + \sum_{j=1}^q \mu_j = K_m,$$

with $K_m = (p+q)K_s\nu$. This is the $1 \times (p+q)$ matrix A given by

$$A = \begin{pmatrix}\mathbf{1}_p^\top & \mathbf{1}_q^\top\end{pmatrix}.$$

Obviously, A has rank 1. The right-hand side is

$$c = K_m.$$

The symmetric positive definite $(p+q) \times (p+q)$ matrix P defining the quadratic functional is

$$P = X^\top X + \begin{pmatrix}\mathbf{1}_p\mathbf{1}_p^\top & -\mathbf{1}_p\mathbf{1}_q^\top \\ -\mathbf{1}_q\mathbf{1}_p^\top & \mathbf{1}_q\mathbf{1}_q^\top\end{pmatrix} + \frac{1}{2K_s}I_{p+q},$$

with

$$X = \begin{pmatrix}-u_1 & \cdots & -u_p & v_1 & \cdots & v_q\end{pmatrix} \quad \text{and} \quad q = 0_{p+q}.$$

Since there are $p+q$ Lagrange multipliers (λ, μ), the $(p+q) \times (p+q)$ matrix P does not have to be augmented with zero's.

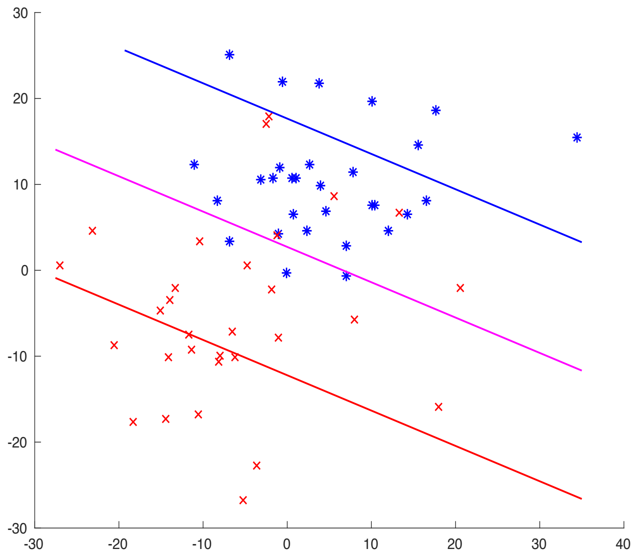


Fig. 18.24 Running (SVM$_{s5}$) on two sets of 30 points; $K = 190$.

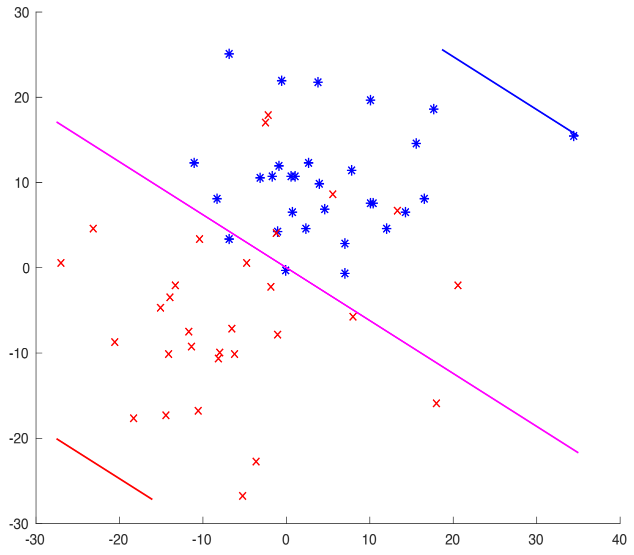


Fig. 18.25 Running (SVM$_{s5}$) on two sets of 30 points; $K = 1/13000$.

We ran our `Matlab` implementation of the above version of (SVM$_{s5}$) on the data set of Section 18.14. Since the value of ν is irrelevant, we picked $\nu = 1$. First we ran our program with $K = 190$; see Figure 18.24. We have $p_m = 23$ and $q_m = 18$. The program does not converge for $K \geq 200$.

Our second run was made with $K = 1/13000$; see Figure 18.25. We have $p_m = 30$ and $q_m = 30$ and we see that the width of the slab is a bit excessive. This example demonstrates that the margin lines need not contain data points.

Method (SVM$_{s5}$) always returns a value for b and η smaller than the value returned by (SVM$_{s4}$) (because of the term $(1/2)b^2$ added to the objective function) but in this example the difference is too small to be noticed.

18.17 Summary and Comparison of the SVM Methods

In this chapter we considered six variants for solving the soft margin binary classification problem for two sets of points $\{u_i\}_{i=1}^p$ and $\{v_j\}_{j=1}^q$ using support vector classification methods. The objective is to find a separating hyperplane $H_{w,b}$ of equation $w^\top x - b = 0$. We also try to find two "margin hyperplanes" $H_{w,b+\delta}$ of equation $w^\top x - b - \delta = 0$ (the blue margin hyperplane) and $H_{w,b-\delta}$ of equation $w^\top x - b + \delta = 0$ (the red margin hyperplane) such that δ is as big as possible and yet the number of misclassified points is minimized, which is achieved by allowing an error $\epsilon_i \geq 0$ for every point u_i, in the sense that the constraint

$$w^\top u_i - b \geq \delta - \epsilon_i$$

should hold, and an error $\xi_j \geq 0$ for every point v_j, in the sense that the constraint

$$-w^\top v_j + b \geq \delta - \xi_j$$

should hold.

The goal is to design an objective function that minimizes ϵ and ξ and maximizes δ. The optimization problem should also solve for w and b, and for this some constraint has to be placed on w. Another goal is to try to use the dual program to solve the optimization problem, because the solutions involve inner products, and thus the problem is amenable to a generalization using kernel functions.

The first attempt, which is to use the objective function

$$J(w, \epsilon, \xi, b, \delta) = -\delta + K \left(\epsilon^\top \; \xi^\top\right) \mathbf{1}_{p+q}$$

and the constraint $w^\top w \leq 1$, does not work very well because this constraint needs to be guarded by a Lagrange multiplier $\gamma \geq 0$, and as a result, minimizing the Lagrangian L to find the dual function G gives an equation for solving w of the form

$$2\gamma w = -X^\top \begin{pmatrix} \lambda \\ \mu \end{pmatrix},$$

but if the sets $\{u_i\}_{i=1}^p$ and $\{v_j\}_{j=1}^q$ are not linearly separable, then an optimal solution may occurs for $\gamma = 0$, in which case it is impossible to determine w. This is Problem (SVM$_{s1}$) considered in Section 18.1.

Soft margin SVM (SVM$_{s1}$):

$$\begin{aligned} &\text{minimize} \quad -\delta + K\left(\sum_{i=1}^p \epsilon_i + \sum_{j=1}^q \xi_j\right) \\ &\text{subject to} \\ &\qquad w^\top u_i - b \geq \delta - \epsilon_i, \quad \epsilon_i \geq 0 \qquad i = 1, \ldots, p \\ &\qquad -w^\top v_j + b \geq \delta - \xi_j, \quad \xi_j \geq 0 \qquad j = 1, \ldots, q \\ &\qquad w^\top w \leq 1. \end{aligned}$$

It is customary to write $\ell = p + q$.

It is shown in Section 18.1 that the dual program is equivalent to the following minimization program:

Dual of the Soft margin SVM (SVM$_{s1}$):

$$\begin{aligned} &\text{minimize} \quad \begin{pmatrix} \lambda^\top & \mu^\top \end{pmatrix} X^\top X \begin{pmatrix} \lambda \\ \mu \end{pmatrix} \\ &\text{subject to} \\ &\qquad \sum_{i=1}^p \lambda_i - \sum_{j=1}^q \mu_j = 0 \\ &\qquad \sum_{i=1}^p \lambda_i + \sum_{j=1}^q \mu_j = 1 \\ &\qquad 0 \leq \lambda_i \leq K, \quad i = 1, \ldots, p \\ &\qquad 0 \leq \mu_j \leq K, \quad j = 1, \ldots, q. \end{aligned}$$

The points u_i and v_j are naturally classified in terms of the values of λ_i and μ_j. The numbers of points in each category have a direct influence on the choice of the parameter K. Let us summarize some of the keys items from Definition 18.1.

The vectors u_i on the blue margin $H_{w,b+\delta}$ and the vectors v_j on the red margin $H_{w,b-\delta}$ are called *support vectors*. Support vectors correspond to vectors u_i for which $w^\top u_i - b - \delta = 0$ (which implies $\epsilon_i = 0$), and vectors v_j for which $w^\top v_j - b + \delta = 0$ (which implies $\xi_j = 0$). Support vectors u_i such that $0 < \lambda_i < K$ and support vectors v_j such that $0 < \mu_j < K$ are *support vectors of type 1*. Support vectors of type 1 play a special role so we denote the sets of indices associated with them by

$$\begin{aligned} I_\lambda &= \{i \in \{1, \ldots, p\} \mid 0 < \lambda_i < K\} \\ I_\mu &= \{j \in \{1, \ldots, q\} \mid 0 < \mu_j < K\}. \end{aligned}$$

We denote their cardinalities by $numsvl_1 = |I_\lambda|$ and $numsvm_1 = |I_\mu|$.

The vectors u_i for which $\lambda_i = K$ and the vectors v_j for which $\mu_j = K$ are said to *fail the margin*. The sets of indices associated with the vectors failing the margin are denoted by

$$\begin{aligned} K_\lambda &= \{i \in \{1, \ldots, p\} \mid \lambda_i = K\} \\ K_\mu &= \{j \in \{1, \ldots, q\} \mid \mu_j = K\}. \end{aligned}$$

We denote their cardinalities by $p_f = |K_\lambda|$ and $q_f = |K_\mu|$.

Vectors u_i such that $\lambda_i > 0$ and vectors v_j such that $\mu_j > 0$ are said to *have margin at most δ*. The sets of indices associated with these vectors are denoted by

$$\begin{aligned} I_{\lambda>0} &= \{i \in \{1, \ldots, p\} \mid \lambda_i > 0\} \\ I_{\mu>0} &= \{j \in \{1, \ldots, q\} \mid \mu_j > 0\}. \end{aligned}$$

We denote their cardinalities by $p_m = |I_{\lambda>0}|$ and $q_m = |I_{\mu>0}|$.

Obviously, $p_f \leq p_m$ and $q_f \leq q_m$. There are $p - p_m$ points u_i classified correctly on the blue side and outside the δ-slab and there are $q - q_m$ points v_j classified correctly on the red side and outside the δ-slab. Intuitively a blue point that fails the margin is on the wrong side of the blue margin and a red point that fails the margin is on the wrong side of the red margin.

It can be shown that that K must be chosen so that

$$\max\left\{\frac{1}{2p_m}, \frac{1}{2q_m}\right\} \leq K \leq \min\left\{\frac{1}{2p_f}, \frac{1}{2q_f}\right\}.$$

If the optimal value is 0, then $\gamma = 0$ and $X \begin{pmatrix} \lambda \\ \mu \end{pmatrix} = 0$, so in this case it is not possible to determine w. However, if the optimal value is > 0, then once a solution for λ and μ is obtained, we have

$$\gamma = \frac{1}{2}\left(\begin{pmatrix} \lambda^\top & \mu^\top \end{pmatrix} X^\top X \begin{pmatrix} \lambda \\ \mu \end{pmatrix} \right)^{1/2}$$
$$w = \frac{1}{2\gamma}\left(\sum_{i=1}^p \lambda_i u_i - \sum_{j=1}^q \mu_j v_j \right),$$

so we get

$$w = \frac{\displaystyle\sum_{i=1}^p \lambda_i u_i - \sum_{j=1}^q \mu_j v_j}{\left(\begin{pmatrix} \lambda^\top & \mu^\top \end{pmatrix} X^\top X \begin{pmatrix} \lambda \\ \mu \end{pmatrix} \right)^{1/2}},$$

If the following mild hypothesis holds, then b and δ can be found.

Standard Margin Hypothesis for (SVM$_{s1}$). There is some index i_0 such that $0 < \lambda_{i_0} < K$ and there is some index j_0 such that $0 < \mu_{j_0} < K$. This means that some u_{i_0} is a support vector of type 1 on the blue margin, and some v_{j_0} is a support vector of type 1 on the red margin.

If the **Standard Margin Hypothesis** for (SVM$_{s1}$) holds, then $\epsilon_{i_0} = 0$ and $\mu_{j_0} = 0$, and we have the active equations

$$w^\top u_{i_0} - b = \delta \quad \text{and} \quad -w^\top v_{j_0} + b = \delta,$$

and we obtain the value of b and δ as

$$b = \frac{1}{2} w^\top (u_{i_0} + v_{j_0})$$
$$\delta = \frac{1}{2} w^\top (u_{i_0} - v_{j_0}).$$

The second more successful approach is to add the term $(1/2)w^\top w$ to the objective function and to drop the constraint $w^\top w \leq 1$. There are several variants of this method, depending on the choice of the regularizing term involving ϵ and ξ (linear or quadratic), how the margin is dealt with (implicitly with the term 1 or explicitly with a term η), and whether the term $(1/2)b^2$ is added to the objective function or not.

These methods all share the property that if the primal problem has an optimal solution with $w \neq 0$, then the dual problem always determines w, and then under mild conditions which we call standard margin hypotheses, b and η can be determined. Then ϵ and ξ can be determined using the constraints that are active. When $(1/2)b^2$ is added to the objective function, b is determined by the equation

$$b = -(\mathbf{1}_p^\top \lambda - \mathbf{1}_q^\top \mu).$$

All these problems are convex and the constraints are qualified, so the duality gap is zero, and if the primal has an optimal solution with $w \neq 0$, then it follows that $\eta \geq 0$.

We now consider five variants in more details.

(1) **Basic Soft margin SVM**: (SVM$_{s2}$).
This is the optimization problem in which the regularization term $K \begin{pmatrix} \epsilon^\top & \xi^\top \end{pmatrix} \mathbf{1}_{p+q}$ is linear and the margin δ is given by $\delta = 1/\|w\|$:

$$\begin{aligned}
&\text{minimize} \quad \frac{1}{2} w^\top w + K \begin{pmatrix} \epsilon^\top & \xi^\top \end{pmatrix} \mathbf{1}_{p+q} \\
&\text{subject to} \\
&\qquad w^\top u_i - b \geq 1 - \epsilon_i, \quad \epsilon_i \geq 0 \qquad i = 1, \ldots, p \\
&\qquad - w^\top v_j + b \geq 1 - \xi_j, \quad \xi_j \geq 0 \qquad j = 1, \ldots, q.
\end{aligned}$$

This problem is the classical one discussed in all books on machine learning or pattern analysis, for instance Vapnik [Vapnik (1998)], Bishop [Bishop (2006)], and Shawe–Taylor and Christianini [Shawe-Taylor and Cristianini (2004)]. It is shown in Section 18.3 that the dual program is

Dual of the Basic Soft margin SVM: (SVM$_{s2}$):

$$\begin{aligned}
&\text{minimize} \quad \frac{1}{2} \begin{pmatrix} \lambda^\top & \mu^\top \end{pmatrix} X^\top X \begin{pmatrix} \lambda \\ \mu \end{pmatrix} - \begin{pmatrix} \lambda^\top & \mu^\top \end{pmatrix} \mathbf{1}_{p+q} \\
&\text{subject to} \\
&\qquad \sum_{i=1}^{p} \lambda_i - \sum_{j=1}^{q} \mu_j = 0 \\
&\qquad 0 \leq \lambda_i \leq K, \quad i = 1, \ldots, p \\
&\qquad 0 \leq \mu_j \leq K, \quad j = 1, \ldots, q.
\end{aligned}$$

We can use the dual program to solve the primal. Once $\lambda \geq 0, \mu \geq 0$ have been found, w is given by

$$w = -X \begin{pmatrix} \lambda \\ \mu \end{pmatrix} = \sum_{i=1}^{p} \lambda_i u_i - \sum_{j=1}^{q} \mu_j v_j,$$

but b is not determined by the dual.

The complementary slackness conditions imply that if $\epsilon_i > 0$, then $\lambda_i = K$, and if $\xi_j > 0$, then $\mu_j = K$. Consequently, if $\lambda_i < K$, then $\epsilon_i = 0$ and u_i is correctly classified, and similarly if $\mu_j < K$, then $\xi_j = 0$ and v_j is correctly classified.
A priori nothing prevents the situation where $\lambda_i = K$ for all nonzero λ_i or $\mu_j = K$ for all nonzero μ_j. If this happens, we can rerun the optimization method with a larger value of K. If the following mild hypothesis holds then b can be found.
Standard Margin Hypothesis for (SVM$_{s2}$). There is some support vector u_{i_0} of type 1 on the blue margin, and some support vector v_{j_0} of type 1 on the red margin.
If the **Standard Margin Hypothesis** for (SVM$_{s2}$) holds then $\epsilon_{i_0} = 0$ and $\mu_{j_0} = 0$, and then we have the active equations

$$w^\top u_{i_0} - b = 1 \quad \text{and} \quad -w^\top v_{j_0} + b = 1,$$

and we obtain

$$b = \frac{1}{2} w^\top (u_{i_0} + v_{j_0}).$$

(2) **Basic Soft margin ν-SVM Problem** (SVM$_{s2'}$).
This a generalization of Problem (SVM$_{s2}$) for a version of the soft margin SVM coming from Problem (SVM$_{h2}$), obtained by adding an extra degree of freedom, namely instead of the margin $\delta = 1/\|w\|$, we use the margin $\delta = \eta/\|w\|$ where η is some positive constant that we wish to maximize. To do so, we add a term $-K_m\eta$ to the objective function. We have the following optimization problem:

$$\begin{aligned}
&\text{minimize} \quad \frac{1}{2} w^\top w - K_m \eta + K_s \begin{pmatrix} \epsilon^\top & \xi^\top \end{pmatrix} \mathbf{1}_{p+q} \\
&\text{subject to} \\
&\qquad w^\top u_i - b \geq \eta - \epsilon_i, \quad \epsilon_i \geq 0 \qquad i = 1, \ldots, p \\
&\qquad -w^\top v_j + b \geq \eta - \xi_j, \quad \xi_j \geq 0 \qquad j = 1, \ldots, q \\
&\qquad \eta \geq 0,
\end{aligned}$$

where $K_m > 0$ and $K_s > 0$ are fixed constants that can be adjusted to determine the influence of η and the regularizing term.
This version of the SVM problem was first discussed in Schölkopf, Smola, Williamson, and Bartlett [Schölkopf *et al.* (2000)] under the name of ν-*SVC*, and also used in Schölkopf, Platt, Shawe–Taylor, and Smola [Schölkopf *et al.* (2001)].

In order for the problem to have a solution we must pick K_m and K_s so that

$$K_m \leq \min\{2pK_s, 2qK_s\}.$$

It is shown in Section 18.5 that the dual program is

Dual of the Basic Soft margin ν-SVM Problem (SVM$_{s2'}$):

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2}\begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix} X^\top X \begin{pmatrix}\lambda \\ \mu\end{pmatrix} \\ &\text{subject to} \\ &\qquad \sum_{i=1}^p \lambda_i - \sum_{j=1}^q \mu_j = 0 \\ &\qquad \sum_{i=1}^p \lambda_i + \sum_{j=1}^q \mu_j \geq K_m \\ &\qquad 0 \leq \lambda_i \leq K_s, \quad i = 1, \ldots, p \\ &\qquad 0 \leq \mu_j \leq K_s, \quad j = 1, \ldots, q. \end{aligned}$$

If the primal problem has an optimal solution with $w \neq 0$, then using the fact that the duality gap is zero we can show that $\eta \geq 0$. Thus constraint $\eta \geq 0$ could be omitted. As in the previous case w is given by

$$w = -X\begin{pmatrix}\lambda \\ \mu\end{pmatrix} = \sum_{i=1}^p \lambda_i u_i - \sum_{j=1}^q \mu_j v_j,$$

but b and η are not determined by the dual.

If we drop the constraint $\eta \geq 0$, then the inequality

$$\sum_{i=1}^p \lambda_i + \sum_{j=1}^q \mu_j \geq K_m$$

is replaced by the equation

$$\sum_{i=1}^p \lambda_i + \sum_{j=1}^q \mu_j = K_m.$$

It convenient to define $\nu > 0$ such that

$$\nu = \frac{K_m}{(p+q)K_s},$$

so that the objective function $J(w, \epsilon, \xi, b, \eta)$ is given by

$$J(w, \epsilon, \xi, b, \eta) = \frac{1}{2} w^\top w + (p+q)K_s \left(-\nu\eta + \frac{1}{p+q} \left(\epsilon^\top \; \xi^\top \right) \mathbf{1}_{p+q} \right).$$

Since we obtain an equivalent problem by rescaling by a common positive factor, theoretically it is convenient to normalize K_s as

$$K_s = \frac{1}{p+q},$$

in which case $K_m = \nu$. This method is called the *ν-support vector machine*.
Under the **Standard Margin Hypothesis** for (SVM$_{s2'}$), there is some support vector u_{i_0} of type 1 and some support vector v_{j_0} of type 1, and by the complementary slackness conditions $\epsilon_{i_0} = 0$ and $\xi_{j_0} = 0$, so we have the two active constraints

$$w^\top u_{i_0} - b = \eta, \quad -w^\top v_{j_0} + b = \eta,$$

and we can solve for b and η and we get

$$\begin{aligned} b &= \frac{w^\top (u_{i_0} + v_{j_0})}{2} \\ \eta &= \frac{w^\top (u_{i_0} - v_{j_0})}{2}. \end{aligned}$$

Due to numerical instability, when writing a computer program it is preferable to compute the lists of indices I_λ and I_μ given by

$$I_\lambda = \{i \in \{1, \ldots, p\} \mid 0 < \lambda_i < K_s\}, \quad I_\mu = \{j \in \{1, \ldots, q\} \mid 0 < \mu_j < K_s\}.$$

Then b and η are given by the following averaging formulae:

$$\begin{aligned} b &= w^\top \left(\left(\sum_{i \in I_\lambda} u_i \right) / |I_\lambda| + \left(\sum_{j \in I_\mu} v_j \right) / |I_\mu| \right) /2 \\ \eta &= w^\top \left(\left(\sum_{i \in I_\lambda} u_i \right) / |I_\lambda| - \left(\sum_{j \in I_\mu} v_j \right) / |I_\mu| \right) /2. \end{aligned}$$

Proposition 18.1 yields bounds on ν for the method to converge, namely

$$\max\left\{ \frac{2p_f}{p+q}, \frac{2q_f}{p+q} \right\} \leq \nu \leq \min\left\{ \frac{2p_m}{p+q}, \frac{2q_m}{p+q} \right\}.$$

In Section 18.7 we investigate conditions on ν that ensure that some point u_{i_0} and some point v_{j_0} is a support vector. Theorem 18.1 shows that for every optimal solution $(w, b, \eta, \epsilon, \xi)$ of Problem (SVM$_{s2'}$) with $w \neq 0$ and $\eta > 0$, if

$$\max\{2p_f/(p+q), 2q_f/(p+q)\} < \nu < \min\{2p/(p+q), 2q/(p+q)\},$$

then some u_{i_0} and some v_{j_0} is a support vector. Under the same conditions on ν Proposition 18.3 shows that η and b can always be determined in terms of (λ, μ) using a single support vector.

(3) **Soft margin ν-SVM Problem** (SVM$_{s3}$). This is the variation of Problem (SVM$_{s2'}$) obtained by adding the term $(1/2)b^2$ to the objective function. The result is that in minimizing the Lagrangian to find the dual function G, not just w but also b is determined. We also suppress the constraint $\eta \geq 0$ which turns out to be redundant. If $\nu > (p_f + q_f)/(p+q)$, then η is also determined. The fact that b and η are determined by the dual seems to be an advantage of Problem (SVM$_{s3}$). The optimization problem is

$$\begin{aligned}
&\text{minimize} \quad \frac{1}{2}w^\top w + \frac{1}{2}b^2 + (p+q)K_s\left(-\nu\eta + \frac{1}{p+q}\begin{pmatrix}\epsilon^\top & \xi^\top\end{pmatrix}\mathbf{1}_{p+q}\right)\\
&\text{subject to}\\
&\qquad\quad w^\top u_i - b \geq \eta - \epsilon_i, \quad \epsilon_i \geq 0 \qquad i = 1, \ldots, p\\
&\qquad -w^\top v_j + b \geq \eta - \xi_j, \quad \xi_j \geq 0 \qquad j = 1, \ldots, q.
\end{aligned}$$

Theoretically it is convenient to assume that $K_s = 1/(p+q)$. Otherwise, ν needs to be replaced by $(p+q)K_s\nu$ in all the formulae below.
It is shown in Section 18.13 that the dual is given by

Dual of the Soft margin ν-SVM Problem (SVM$_{s3}$):

$$\begin{aligned}
&\text{minimize} \quad \frac{1}{2}\begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix}\left(X^\top X + \begin{pmatrix}\mathbf{1}_p\mathbf{1}_p^\top & -\mathbf{1}_p\mathbf{1}_q^\top\\ -\mathbf{1}_q\mathbf{1}_p^\top & \mathbf{1}_q\mathbf{1}_q^\top\end{pmatrix}\right)\begin{pmatrix}\lambda\\ \mu\end{pmatrix}\\
&\text{subject to}\\
&\qquad \sum_{i=1}^p \lambda_i + \sum_{j=1}^q \mu_j = \nu\\
&\qquad 0 \leq \lambda_i \leq K_s, \quad i = 1, \ldots, p\\
&\qquad 0 \leq \mu_j \leq K_s, \quad j = 1, \ldots, q.
\end{aligned}$$

Once a solution for λ and μ is obtained, we have

$$w = -X \begin{pmatrix} \lambda \\ \mu \end{pmatrix} = \sum_{i=1}^{p} \lambda_i u_i - \sum_{j=1}^{q} \mu_j v_j$$

$$b = -\sum_{i=1}^{p} \lambda_i + \sum_{j=1}^{q} \mu_j.$$

Note that the constraint

$$\sum_{i=1}^{p} \lambda_i - \sum_{j=1}^{q} \mu_j = 0$$

occurring in the dual of Program ($\mathrm{SVM}_{s2'}$) has been traded for the equation

$$b = -\sum_{i=1}^{p} \lambda_i + \sum_{j=1}^{q} \mu_j$$

determining b.

If $\nu > (p_f + q_f)/(p+q)$, then η is determined by expressing that the duality gap is zero. We obtain

$$((p+q)\nu - p_f - q_f)\eta = (p_f - q_f)b + w^\top \left(\sum_{j \in K_\mu} v_j - \sum_{i \in K_\lambda} u_i \right) + \frac{1}{K_s} \begin{pmatrix} \lambda^\top & \mu^\top \end{pmatrix} \left(X^\top X + \begin{pmatrix} \mathbf{1}_p \mathbf{1}_p^\top & -\mathbf{1}_p \mathbf{1}_q^\top \\ -\mathbf{1}_q \mathbf{1}_p^\top & \mathbf{1}_q \mathbf{1}_q^\top \end{pmatrix} \right) \begin{pmatrix} \lambda \\ \mu \end{pmatrix}.$$

In practice another way to compute η is to assume the Standard Margin Hypothesis for (SVM_{s3}). Under the **Standard Margin Hypothesis** for (SVM_{s3}), either some u_{i_0} is a support vector of type 1 *or* some v_{j_0} is a support vector of type 1. By the complementary slackness conditions $\epsilon_{i_0} = 0$ or $\xi_{j_0} = 0$, so we have

$$w^\top u_{i_0} - b = \eta, \quad \text{or} \quad -w^\top v_{j_0} + b = \eta,$$

and we can solve for η. As in ($\mathrm{SVM}_{s2'}$) we get more numerically stable formulae by averaging over the sets I_λ and I_μ.

Proposition 18.4 gives bounds ν, namely

$$\frac{p_f + q_f}{p+q} \leq \nu \leq \frac{p_m + q_m}{p+q}.$$

In Section 18.11 we investigate conditions on ν that ensure that either there is some blue support vector u_{i_0} *or* there is some red support vector v_{j_0}. Theorem 18.2 shows that for every optimal solution $(w, b, \eta, \epsilon, \xi)$ of Problem (SVM_{s3}) with $w \neq 0$ and $\eta > 0$, if

$$(p_{sf} + q_{sf})/(p+q) < \nu < 1,$$

then some u_{i_0} or some v_{j_0} is a support vector.

(4) **Basic Quadratic Soft margin ν-SVM Problem** (SVM$_{s4}$). This is the version of Problem (SVM$_{s2'}$) in which instead of using the linear function $K_s \left(\epsilon^\top \; \xi^\top\right) \mathbf{1}_{p+q}$ as a regularizing function we use the quadratic function $K(\|\epsilon\|_2^2 + \|\xi\|_2^2)$. The optimization problem is

$$\begin{aligned}
&\text{minimize} \quad \frac{1}{2}w^\top w + (p+q)K_s\left(-\nu\eta + \frac{1}{p+q}(\epsilon^\top\epsilon + \xi^\top\xi)\right)\\
&\text{subject to}\\
&\qquad w^\top u_i - b \geq \eta - \epsilon_i, \qquad i = 1,\dots,p\\
&\qquad -w^\top v_j + b \geq \eta - \xi_j, \qquad j = 1,\dots,q\\
&\qquad \eta \geq 0,
\end{aligned}$$

where ν and K_s are two given positive constants. As we saw earlier, theoretically, it is convenient to pick $K_s = 1/(p+q)$. When writing a computer program, it is preferable to assume that K_s is arbitrary. In this case ν needs to be replaced by $(p+q)K_s\nu$ in all the formulae obtained with $K_s = 1/(p+q)$.
In this method, it is no longer necessary to require $\epsilon \geq 0$ and $\xi \geq 0$, because an optimal solution satisfies these conditions.
One of the advantages of this methods is that ϵ is determined by λ, ξ is determined by μ, and η and b are determined by λ and μ. We can omit the constraint $\eta \geq 0$, because for an optimal solution it can be shown using duality that $\eta \geq 0$; see Section 18.14. For K_s and ν fixed, if Program (SVM$_{s4}$) has an optimal solution, then it is unique; see Theorem 18.3.
A drawback of Program (SVM$_{s4}$) is that for fixed K_s, the quantity $\delta = \eta/\|w\|$ and the hyperplanes $H_{w,b}, H_{w,b+\eta}$ and $H_{w,b-\eta}$ are *independent* of ν. This is shown in Theorem 18.3. Thus this method is less flexible than (SVM$_{s2'}$) and (SVM$_{s3}$).
It is shown in Section 18.9 that the dual is given by

Dual of the Basic Quadratic Soft margin ν-SVM Problem (SVM$_{s4}$):

$$\begin{aligned}
&\text{minimize} \quad \frac{1}{2}\left(\lambda^\top \; \mu^\top\right)\left(X^\top X + \frac{1}{2K}I_{p+q}\right)\begin{pmatrix}\lambda\\ \mu\end{pmatrix}\\
&\text{subject to}
\end{aligned}$$

$$\begin{aligned}
&\sum_{i=1}^{p} \lambda_i - \sum_{j=1}^{q} \mu_j = 0\\
&\sum_{i=1}^{p} \lambda_i + \sum_{j=1}^{q} \mu_j \geq \nu\\
&\lambda_i \geq 0, \quad i = 1, \ldots, p\\
&\mu_j \geq 0, \quad j = 1, \ldots, q.
\end{aligned}$$

The above program is similar to the program that was obtained for Problem (SVM$_{s2'}$) but the matrix $X^\top X$ is replaced by the matrix $X^\top X + (1/2K)I_{p+q}$, which is positive definite since $K > 0$, and also the inequalities $\lambda_i \leq K$ and $\mu_j \leq K$ no longer hold. If the constraint $\eta \geq 0$ is dropped, then the inequality

$$\sum_{i=1}^{p} \lambda_i + \sum_{j=1}^{q} \mu_j \geq \nu$$

is replaced by the equation

$$\sum_{i=1}^{p} \lambda_i + \sum_{j=1}^{q} \mu_j = \nu.$$

We obtain w from λ and μ, and γ, as in Problem (SVM$_{s2'}$); namely,

$$w = -X \begin{pmatrix} \lambda \\ \mu \end{pmatrix} = \sum_{i=1}^{p} \lambda_i u_i - \sum_{j=1}^{q} \mu_j v_j$$

and η is given by

$$(p+q)K_s \nu \eta = \begin{pmatrix} \lambda^\top & \mu^\top \end{pmatrix} \left(X^\top X + \frac{1}{2K_s} I_{p+q} \right) \begin{pmatrix} \lambda \\ \mu \end{pmatrix}.$$

The constraints imply that there is some i_o such that $\lambda_{i_0} > 0$ *and* some j_0 such that $\mu_{j_0} > 0$, which means that at least two points are misclassified, so Problem (SVM$_{s4}$) should only be used when the sets $\{u_i\}$ and $\{v_j\}$ are *not* linearly separable. We can solve for b using the active constraints corresponding to any i_0 such that $\lambda_{i_0} > 0$ and any j_0 such that $\mu_{j_0} > 0$. To improve numerical stability we average over the sets of indices I_λ and I_μ.

(5) **Quadratic Soft margin ν-SVM Problem** (SVM$_{s5}$). This is the variant of Problem (SVM$_{s4}$) in which we add the term $(1/2)b^2$ to the objective function. We also drop the constraint $\eta \geq 0$ which is redundant. We have the following optimization problem:

$$\begin{aligned}
&\text{minimize} \quad \frac{1}{2}w^\top w + \frac{1}{2}b^2 + (p+q)K_s\left(-\nu\eta + \frac{1}{p+q}(\epsilon^\top \epsilon + \xi^\top \xi)\right)\\
&\text{subject to}\\
&\qquad w^\top u_i - b \geq \eta - \epsilon_i, \qquad i = 1, \ldots, p\\
&\qquad -w^\top v_j + b \geq \eta - \xi_j, \qquad j = 1, \ldots, q,
\end{aligned}$$

where ν and K_s are two given positive constants. As we saw earlier, it is convenient to pick $K_s = 1/(p+q)$. When writing a computer program, it is preferable to assume that K_s is arbitrary. In this case ν must be replaced by $(p+q)K_s\nu$ in all the formulae.
One of the advantages of this methods is that ϵ is determined by λ, ξ is determined by μ (as in (SVM$_{s4}$)), and both η and b determined by λ and μ. We can omit the constraint $\eta \geq 0$, because for an optimal solution it can be shown using duality that $\eta \geq 0$. For K_s and ν fixed, if Program (SVM$_{s5}$) has an optimal solution, then it is unique; see Theorem 18.4.
A drawback of Program (SVM$_{s5}$) is that for fixed K_s, the quantity $\delta = \eta/\left\|w\right\|$ and the hyperplanes $H_{w,b}, H_{w,b+\eta}$ and $H_{w,b-\eta}$ are *independent* of ν. This is shown in Theorem 18.4. Thus this method is less flexible than (SVM$_{s2'}$) and (SVM$_{s3}$).
It is shown in Section 18.15 that the dual of Program (SVM$_{s5}$) is given by

Dual of the Quadratic Soft margin ν-SVM Problem (SVM$_{s5}$):

$$\begin{aligned}
&\text{minimize} \quad \frac{1}{2}\begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix}\left(X^\top X + \begin{pmatrix}\mathbf{1}_p\mathbf{1}_p^\top & -\mathbf{1}_p\mathbf{1}_q^\top \\ -\mathbf{1}_q\mathbf{1}_p^\top & \mathbf{1}_q\mathbf{1}_q^\top\end{pmatrix} + \frac{1}{2K}I_{p+q}\right)\begin{pmatrix}\lambda \\ \mu\end{pmatrix}\\
&\text{subject to}\\
&\qquad \sum_{i=1}^p \lambda_i + \sum_{j=1}^q \mu_j = \nu\\
&\qquad \lambda_i \geq 0, \quad i = 1, \ldots, p\\
&\qquad \mu_j \geq 0, \quad j = 1, \ldots, q.
\end{aligned}$$

This time we obtain w, b, η, ϵ and ξ from λ and μ:

$$\begin{aligned} w &= \sum_{i=1}^{p} \lambda_i u_i - \sum_{j=1}^{q} \mu_j v_j \\ b &= -\sum_{i=1}^{p} \lambda_i + \sum_{j=1}^{q} \mu_j \\ \epsilon &= \frac{\lambda}{2K} \\ \xi &= \frac{\mu}{2K}, \end{aligned}$$

and

$$(p+q)K_s\nu\eta = \begin{pmatrix} \lambda^\top & \mu^\top \end{pmatrix} \left(X^\top X + \begin{pmatrix} \mathbf{1}_p\mathbf{1}_p^\top & -\mathbf{1}_p\mathbf{1}_q^\top \\ -\mathbf{1}_q\mathbf{1}_p^\top & \mathbf{1}_q\mathbf{1}_q^\top \end{pmatrix} + \frac{1}{2K_s} I_{p+q} \right) \begin{pmatrix} \lambda \\ \mu \end{pmatrix}.$$

The constraint

$$\sum_{i=1}^{p} \lambda_i + \sum_{j=1}^{q} \mu_j = \nu$$

implies that either there is some i_0 such that $\lambda_{i_0} > 0$ or there is some j_0 such that $\mu_{j_0} > 0$, we have $\epsilon_{i_0} > 0$ or $\xi_{j_0} > 0$, which means that at least one point is misclassified, so Problem (SVM$_{s5}$) should only be used when the sets $\{u_i\}$ and $\{v_j\}$ are *not* linearly separable.

These methods all have a kernelized version.

We implemented all these methods in `Matlab`, solving the dual using ADMM.

From a theoretical point of view, Problems (SVM$_{s4}$) and (SVM$_{s5}$) seem to have more advantages than the others since they determine w, b, η and b without requiring any condition about support vectors of type 1. However, from a practical point of view, Problems (SVM$_{s4}$) and (SVM$_{s5}$) are less flexible that (SVM$_{s2'}$) and (SVM$_{s3}$), and we have observed that (SVM$_{s4}$) and (SVM$_{s5}$) are unable to produce as small a margin δ as (SVM$_{s2'}$) and (SVM$_{s3}$).

18.18 Problems

Problem 18.1. Prove the following inequality

$$\max\left\{\frac{1}{2p_m}, \frac{1}{2q_m}\right\} \leq K \leq \min\left\{\frac{1}{2p_f}, \frac{1}{2q_f}\right\}$$

stated just after Definition 18.1.

Problem 18.2. Prove the averaging formulae

$$b = w^\top \left(\Big(\sum_{i \in I_\lambda} u_i \Big) / |I_\lambda| + \Big(\sum_{j \in I_\mu} v_j \Big) / |I_\mu| \right) /2$$

$$\delta = w^\top \left(\Big(\sum_{i \in I_\lambda} u_i \Big) / |I_\lambda| - \Big(\sum_{j \in I_\mu} v_j \Big) / |I_\mu| \right) /2$$

stated at the end of Section 18.1.

Problem 18.3. Prove that the matrix

$$A = \begin{pmatrix} \mathbf{1}_p^\top & -\mathbf{1}_q^\top & 0_p^\top & 0_q^\top \\ \mathbf{1}_p^\top & \mathbf{1}_q^\top & 0_p^\top & 0_q^\top \\ I_p & 0_{p,q} & I_p & 0_{p,q} \\ 0_{q,p} & I_q & 0_{q,p} & I_q \end{pmatrix}$$

has rank $p + q + 2$.

Problem 18.4. Prove that the dual program of the kernel version of (SVM$_{s1}$) is given by:

Dual of Soft margin kernel SVM (SVM$_{s1}$):

$$\begin{aligned} &\text{minimize} \quad \begin{pmatrix} \lambda^\top & \mu^\top \end{pmatrix} \mathbf{K} \begin{pmatrix} \lambda \\ \mu \end{pmatrix} \\ &\text{subject to} \\ &\qquad \sum_{i=1}^p \lambda_i = \sum_{j=1}^q \mu_j = \frac{1}{2} \\ &\qquad 0 \leq \lambda_i \leq K, \quad i = 1, \ldots, p \\ &\qquad 0 \leq \mu_j \leq K, \quad j = 1, \ldots, q, \end{aligned}$$

where $\mathbf{K}$ is the $\ell \times \ell$ kernel symmetric matrix (with $\ell = p + q$) given by

$$\mathbf{K}_{ij} = \begin{cases} \kappa(u_i, u_j) & 1 \leq i \leq p,\ 1 \leq j \leq q \\ -\kappa(u_i, v_{j-p}) & 1 \leq i \leq p,\ p+1 \leq j \leq p+q \\ -\kappa(v_{i-p}, u_j) & p+1 \leq i \leq p+q,\ 1 \leq j \leq p \\ \kappa(v_{i-p}, v_{j-q}) & p+1 \leq i \leq p+q,\ p+1 \leq j \leq p+q. \end{cases}$$

Problem 18.5. Prove the averaging formula

$$b = w^\top \left(\left(\sum_{i \in I_\lambda} u_i \right) / |I_\lambda| + \left(\sum_{j \in I_\mu} v_j \right) / |I_\mu| \right) /2$$

stated in Section 18.3.

Problem 18.6. Prove that the kernel version of Program (SVM$_{s2}$) is given by:

Dual of Soft margin kernel SVM (SVM$_{s2}$):

$$\begin{aligned}
&\text{minimize} \quad \frac{1}{2} \begin{pmatrix} \lambda^\top & \mu^\top \end{pmatrix} \mathbf{K} \begin{pmatrix} \lambda \\ \mu \end{pmatrix} - \begin{pmatrix} \lambda^\top & \mu^\top \end{pmatrix} \mathbf{1}_{p+q} \\
&\text{subject to} \\
&\qquad \sum_{i=1}^p \lambda_i - \sum_{j=1}^q \mu_j = 0 \\
&\qquad 0 \leq \lambda_i \leq K, \quad i = 1, \ldots, p \\
&\qquad 0 \leq \mu_j \leq K, \quad j = 1, \ldots, q,
\end{aligned}$$

where $\mathbf{K}$ is the $\ell \times \ell$ kernel symmetric matrix (with $\ell = p + q$) given at the end of Section 18.1.

Problem 18.7. Prove that the matrix

$$A = \begin{pmatrix} \mathbf{1}_p^\top & -\mathbf{1}_q^\top & 0_p^\top & 0_q^\top \\ I_p & 0_{p,q} & I_p & 0_{p,q} \\ 0_{q,p} & I_q & 0_{q,p} & I_q \end{pmatrix}$$

has rank $p + q + 1$.

Problem 18.8. Prove that the matrices

$$A = \begin{pmatrix} \mathbf{1}_p^\top & -\mathbf{1}_q^\top & 0_p^\top & 0_q^\top & 0 \\ \mathbf{1}_p^\top & \mathbf{1}_q^\top & 0_p^\top & 0_q^\top & -1 \\ I_p & 0_{p,q} & I_p & 0_{p,q} & 0_p \\ 0_{q,p} & I_q & 0_{q,p} & I_q & 0_q \end{pmatrix} \quad \text{and} \quad A_2 = \begin{pmatrix} \mathbf{1}_p^\top & -\mathbf{1}_q^\top & 0_p^\top & 0_q^\top \\ \mathbf{1}_p^\top & \mathbf{1}_q^\top & 0_p^\top & 0_q^\top \\ I_p & 0_{p,q} & I_p & 0_{p,q} \\ 0_{q,p} & I_q & 0_{q,p} & I_q \end{pmatrix}$$

have rank $p + q + 2$.

Problem 18.9. Prove that the kernel version of Program (SVM$_{s2'}$) is given by:

Dual of the Soft margin kernel SVM (SVM$_{s2'}$):

$$\begin{aligned}
&\text{minimize} \quad \frac{1}{2}\begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix}\mathbf{K}\begin{pmatrix}\lambda\\ \mu\end{pmatrix}\\
&\text{subject to}\\
&\qquad \sum_{i=1}^p \lambda_i - \sum_{j=1}^q \mu_j = 0\\
&\qquad \sum_{i=1}^p \lambda_i + \sum_{j=1}^q \mu_j \geq K_m\\
&\qquad 0 \leq \lambda_i \leq K_s, \quad i = 1, \ldots, p\\
&\qquad 0 \leq \mu_j \leq K_s, \quad j = 1, \ldots, q,
\end{aligned}$$

where $\mathbf{K}$ is the kernel matrix of Section 18.1.

Problem 18.10. Prove the formulae determining b in terms of η stated just before Theorem 18.3.

Problem 18.11. Prove that the matrix

$$A = \begin{pmatrix} \mathbf{1}_p^\top & \mathbf{1}_q^\top & 0_p^\top & 0_q^\top \\ I_p & 0_{p,q} & I_p & 0_{p,q} \\ 0_{q,p} & I_q & 0_{q,p} & I_q \end{pmatrix}$$

has rank $p + q + 1$.

Problem 18.12. Prove that the kernel version of Program (SVM$_{s3}$) is given by:

Dual of the Soft margin kernel SVM (SVM$_{s3}$):

$$\begin{aligned}
&\text{minimize} \quad \frac{1}{2}\begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix}\left(\mathbf{K} + \begin{pmatrix}\mathbf{1}_p\mathbf{1}_p^\top & -\mathbf{1}_p\mathbf{1}_q^\top \\ -\mathbf{1}_q\mathbf{1}_p^\top & \mathbf{1}_q\mathbf{1}_q^\top\end{pmatrix}\right)\begin{pmatrix}\lambda\\ \mu\end{pmatrix}\\
&\text{subject to}\\
&\qquad \sum_{i=1}^p \lambda_i + \sum_{j=1}^q \mu_j = \nu\\
&\qquad 0 \leq \lambda_i \leq K_s, \quad i = 1, \ldots, p\\
&\qquad 0 \leq \mu_j \leq K_s, \quad j = 1, \ldots, q,
\end{aligned}$$

where $\mathbf{K}$ is the kernel matrix of Section 18.1.

Problem 18.13. Prove that the matrices

$$A = \begin{pmatrix} \mathbf{1}_p^\top & -\mathbf{1}_q^\top & 0 \\ \mathbf{1}_p^\top & \mathbf{1}_q^\top & -1 \end{pmatrix} \quad \text{and} \quad A_2 = \begin{pmatrix} \mathbf{1}_p^\top & -\mathbf{1}_q^\top \\ \mathbf{1}_p^\top & \mathbf{1}_q^\top \end{pmatrix}$$

have rank 2.

Problem 18.14. Implement Program (SVM$_{s4}$) in `Matlab`. You may adapt the programs given in Section B.2 and Section B.3.

Problem 18.15. Prove that the kernel version of Program (SVM$_{s4}$) is given by:

Dual of the Soft margin kernel SVM (SVM$_{s4}$):

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2} \begin{pmatrix} \lambda^\top & \mu^\top \end{pmatrix} \left(\mathbf{K} + \frac{p+q}{2} I_{p+q} \right) \begin{pmatrix} \lambda \\ \mu \end{pmatrix} \\ \text{subject to} \quad & \\ & \sum_{i=1}^p \lambda_i - \sum_{j=1}^q \mu_j = 0 \\ & \sum_{i=1}^p \lambda_i + \sum_{j=1}^q \mu_j \geq \nu \\ & \lambda_i \geq 0, \quad i = 1, \ldots, p \\ & \mu_j \geq 0, \quad j = 1, \ldots, q, \end{aligned}$$

where $\mathbf{K}$ is the kernel matrix of Section 18.1.

Problem 18.16. Implement Program (SVM$_{s5}$) in `Matlab`. You may adapt the programs given in Section B.2 and Section B.3.

Problem 18.17. Prove that the kernel version of Program (SVM$_{s5}$) is given by:

Dual of the Soft margin kernel SVM (SVM$_{s5}$):

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2} \begin{pmatrix} \lambda^\top & \mu^\top \end{pmatrix} \left(\mathbf{K} + \begin{pmatrix} \mathbf{1}_p \mathbf{1}_p^\top & -\mathbf{1}_p \mathbf{1}_q^\top \\ -\mathbf{1}_q \mathbf{1}_p^\top & \mathbf{1}_q \mathbf{1}_q^\top \end{pmatrix} + \frac{p+q}{2} I_{p+q} \right) \begin{pmatrix} \lambda \\ \mu \end{pmatrix} \\ \text{subject to} \quad & \\ & \sum_{i=1}^p \lambda_i + \sum_{j=1}^q \mu_j = \nu \\ & \lambda_i \geq 0, \quad i = 1, \ldots, p \\ & \mu_j \geq 0, \quad j = 1, \ldots, q, \end{aligned}$$

where $\mathbf{K}$ is the kernel matrix of Section 18.1.

Chapter 19

Ridge Regression, Lasso, Elastic Net

In this chapter we discuss linear regression. This problem can be cast as a learning problem. We observe a sequence of (distinct) pairs $((x_1, y_1), \ldots, (x_m, y_m))$ called a *set of training data* (or *predictors*), where $x_i \in \mathbb{R}^n$ and $y_i \in \mathbb{R}$, viewed as input-output pairs of some unknown function f that we are trying to infer. The simplest kind of function is a linear function $f(x) = x^\top w$, where $w \in \mathbb{R}^n$ is a vector of coefficients usually called a *weight vector*. Since the problem is overdetermined and since our observations may be subject to errors, we can't solve for w exactly as the solution of the system $Xw = y$, where X is the $m \times n$ matrix

$$X = \begin{pmatrix} x_1^\top \\ \vdots \\ x_m^\top \end{pmatrix},$$

where the row vectors $x_i^\top$ are the rows of X, and thus the $x_i \in \mathbb{R}^n$ are column vectors. So instead we solve the least-squares problem of minimizing $\|Xw - y\|_2^2$. In general there are still infinitely many solutions so we add a regularizing term. If we add the term $K \|w\|_2^2$ to the objective function $J(w) = \|Xw - y\|_2^2$, then we have *ridge regression*. This problem is discussed in Section 19.1 where we derive the dual program. The dual has a unique solution which yields a solution of the primal. However, the solution of the dual is given in terms of the matrix $XX^\top$ (whereas the solution of the primal is given in terms of $X^\top X$), and since our data points x_i are represented by the rows of the matrix X, we see that this solution only involves inner products of the x_i. This observation is the core of the idea of kernel functions, which were discussed in Chapter 17. We also explain how to solve the problem of learning an affine function $f(x) = x^\top w + b$.

In general the vectors w produced by ridge regression have few zero entries. In practice it is highly desirable to obtain sparse solutions, that is vectors w with many components equal to zero. This can be achieved by replacing the regularizing term $K \|w\|_2^2$ by the regularizing term $K \|w\|_1$; that is, to use the ℓ^1-norm instead of the ℓ^2-norm; see Section 19.4. This method has the exotic name of *lasso regression*. This time there is no closed-form solution, but this is a convex optimization problem and there are efficient iterative methods to solve it. One of the best methods relies on ADMM (see Section 16.8) and is discussed in Section 19.4. The lasso method has some limitations, in particular when the number m of data is smaller than the dimension n of the data. This happens in some applications in genetics and medicine. Fortunately there is a way to combine the best features of ridge regression and lasso, which is to use *two* regularizing terms:

(1) An ℓ^2-term $(1/2)K \|w\|_2^2$ as in ridge regression (with $K > 0$).
(2) An ℓ^1-term $\tau \|w\|_1$ as in lasso.

This method is known as *elastic net regression* and is discussed in Section 19.6. It retains most of the desirable features of ridge regression and lasso, and eliminates some of their weaknesses. Furthermore, it is effectively solved by ADMM.

19.1 Ridge Regression

The problem of solving an overdetermined or underdetermined linear system $Aw = y$, where A is an $m \times n$ matrix, arises as a "learning problem" in which we observe a sequence of data $((a_1, y_1), \ldots, (a_m, y_m))$, viewed as input-output pairs of some unknown function f that we are trying to infer, where the a_i are the *rows* of the matrix A and $y_i \in \mathbb{R}$. The values y_i are sometimes called *labels* or *responses*. The simplest kind of function is a linear function $f(x) = x^\top w$, where $w \in \mathbb{R}^n$ is a vector of coefficients usually called a *weight vector*, or sometimes an *estimator*. In the statistical literature w is often denoted by β. Since the problem is overdetermined and since our observations may be subject to errors, we can't solve for w exactly as the solution of the system $Aw = y$, so instead we solve the least-square problem of minimizing $\|Aw - y\|_2^2$.

In Section 21.1 (Vol. I) we showed that this problem can be solved using the pseudo-inverse. We know that the minimizers w are solutions of the normal equations $A^\top Aw = A^\top y$, but when $A^\top A$ is not invertible, such a solution is not unique so some criterion has to be used to choose among these solutions.

One solution is to pick the unique vector w^+ of smallest Euclidean norm

$\|w^+\|_2$ that minimizes $\|Aw - y\|_2^2$. The solution w^+ is given by $w^+ = A^+y$, where A^+ is the pseudo-inverse of A. The matrix A^+ is obtained from an SVD of A, say $A = V\Sigma U^\top$. Namely, $A^+ = U\Sigma^+V^\top$, where Σ^+ is the matrix obtained from Σ by replacing every nonzero singular value σ_i in Σ by σ_i^{-1}, leaving all zeros in place, and then transposing. The difficulty with this approach is that it requires knowing whether a singular value is zero or very small but nonzero. A very small nonzero singular value σ in Σ yields a very large value σ^{-1} in Σ^+, but $\sigma = 0$ remains 0 in Σ^+.

This discontinuity phenomenon is not desirable and another way is to control the size of w by adding a regularization term to $\|Aw - y\|^2$, and a natural candidate is $\|w\|^2$.

It is customary to rename each column vector $a_i^\top$ as x_i (where $x_i \in \mathbb{R}^n$) and to rename the input data matrix A as X, so that the row vector $x_i^\top$ are the *rows* of the $m \times n$ matrix X

$$X = \begin{pmatrix} x_1^\top \\ \vdots \\ x_m^\top \end{pmatrix}.$$

Our optimization problem, called *ridge regression*, is

Program (RR1):

$$\text{minimize} \quad \|y - Xw\|^2 + K\|w\|^2,$$

which by introducing the new variable $\xi = y - Xw$ can be rewritten as

Program (RR2):

$$\begin{aligned}&\text{minimize} \quad \xi^\top\xi + Kw^\top w\\ &\text{subject to}\\ &\qquad\qquad y - Xw = \xi,\end{aligned}$$

where $K > 0$ is some constant determining the influence of the regularizing term $w^\top w$, and we minimize over ξ and w.

The objective function of the first version of our minimization problem can be expressed as

$$\begin{aligned}J(w) &= \|y - Xw\|^2 + K\|w\|^2\\ &= (y - Xw)^\top(y - Xw) + Kw^\top w\\ &= y^\top y - 2w^\top X^\top y + w^\top X^\top Xw + Kw^\top w\\ &= w^\top(X^\top X + KI_n)w - 2w^\top X^\top y + y^\top y.\end{aligned}$$

The matrix $X^\top X$ is symmetric positive semidefinite and $K > 0$, so the matrix $X^\top X + KI_n$ is positive definite. It follows that

$$J(w) = w^\top (X^\top X + KI_n)w - 2w^\top X^\top y + y^\top y$$

is strictly convex, so by Theorem 4.5(2)-(4), it has a unique minimum iff $\nabla J_w = 0$. Since

$$\nabla J_w = 2(X^\top X + KI_n)w - 2X^\top y,$$

we deduce that

$$w = (X^\top X + KI_n)^{-1} X^\top y. \tag{$*_{wp}$}$$

There is an interesting connection between the matrix $(X^\top X + KI_n)^{-1} X^\top$ and the pseudo-inverse X^+ of X.

Proposition 19.1. *The limit of the matrix $(X^\top X + KI_n)^{-1} X^\top$ when $K > 0$ goes to zero is the pseudo-inverse X^+ of X.*

Proof. To show this let $X = V\Sigma U^\top$ be a SVD of X. Then

$$(X^\top X + KI_n) = U\Sigma^\top V^\top V\Sigma U^\top + KI_n = U(\Sigma^\top\Sigma + KI_n)U^\top,$$

so

$$\begin{aligned}(X^\top X + KI_n)^{-1} X^\top &= U(\Sigma^\top\Sigma + KI_n)^{-1} U^\top U\Sigma^\top V^\top \\ &= U(\Sigma^\top\Sigma + KI_n)^{-1}\Sigma^\top V^\top.\end{aligned}$$

The diagonal entries in the matrix $(\Sigma^\top\Sigma + KI_n)^{-1}\Sigma^\top$ are

$$\frac{\sigma_i}{\sigma_i^2 + K}, \quad \text{if } \sigma_i > 0,$$

and zero if $\sigma_i = 0$. All nondiagonal entries are zero. When $\sigma_i > 0$ and $K > 0$ goes to 0,

$$\lim_{K \mapsto 0} \frac{\sigma_i}{\sigma_i^2 + K} = \sigma_i^{-1},$$

so

$$\lim_{K \mapsto 0} (\Sigma^\top\Sigma + KI_n)^{-1}\Sigma^\top = \Sigma^+,$$

which implies that

$$\lim_{K \mapsto 0} (X^\top X + KI_n)^{-1} X^\top = X^+. \qquad \square$$

The dual function of the first formulation of our problem is a constant function (with value the minimum of J) so it is not useful, but the second formulation of our problem yields an interesting dual problem. The Lagrangian is

$$\begin{aligned} L(\xi, w, \lambda) &= \xi^\top \xi + Kw^\top w + (y - Xw - \xi)^\top \lambda \\ &= \xi^\top \xi + Kw^\top w - w^\top X^\top \lambda - \xi^\top \lambda + \lambda^\top y, \end{aligned}$$

with $\lambda, \xi, y \in \mathbb{R}^m$. The Lagrangian $L(\xi, w, \lambda)$, as a function of ξ and w with λ held fixed, is obviously convex, in fact strictly convex.

To derive the dual function $G(\lambda)$ we minimize $L(\xi, w, \lambda)$ with respect to ξ and w. Since $L(\xi, w, \lambda)$ is (strictly) convex as a function of ξ and w, by Theorem 4.5(4), it has a minimum iff its gradient $\nabla L_{\xi,w}$ is zero (in fact, by Theorem 4.5(2), a unique minimum since the function is strictly convex). Since

$$\nabla L_{\xi,w} = \begin{pmatrix} 2\xi - \lambda \\ 2Kw - X^\top \lambda \end{pmatrix},$$

we get

$$\begin{aligned} \lambda &= 2\xi \\ w &= \frac{1}{2K} X^\top \lambda = X^\top \frac{\xi}{K}. \end{aligned}$$

The above suggests defining the variable α so that $\xi = K\alpha$, so we have $\lambda = 2K\alpha$ and $w = X^\top \alpha$. Then we obtain the dual function as a function of α by substituting the above values of ξ, λ and w back in the Lagrangian and we get

$$\begin{aligned} G(\alpha) &= K^2 \alpha^\top \alpha + K\alpha^\top XX^\top \alpha - 2K\alpha^\top XX^\top \alpha - 2K^2 \alpha^\top \alpha + 2K\alpha^\top y \\ &= -K\alpha^\top (XX^\top + KI_m)\alpha + 2K\alpha^\top y. \end{aligned}$$

This is a strictly concave function so by Theorem 4.5(4), its maximum is achieved iff $\nabla G_\alpha = 0$, that is,

$$2K(XX^\top + KI_m)\alpha = 2Ky,$$

which yields

$$\alpha = (XX^\top + KI_m)^{-1} y.$$

Putting everything together we obtain

$$\begin{aligned} \alpha &= (XX^\top + KI_m)^{-1} y \\ w &= X^\top \alpha \\ \xi &= K\alpha, \end{aligned}$$

which yields

$$w = X^\top (XX^\top + KI_m)^{-1} y. \tag{$*_{wd}$}$$

Earlier in $(*_{wp})$ we found that

$$w = (X^\top X + KI_n)^{-1} X^\top y,$$

and it is easy to check that

$$(X^\top X + KI_n)^{-1} X^\top = X^\top (XX^\top + KI_m)^{-1}.$$

If $n < m$ it is cheaper to use the formula on the left-hand side, but if $m < n$ it is cheaper to use the formula on the right-hand side.

19.2 Ridge Regression; Learning an Affine Function

It is easy to adapt the above method to learn an affine function $f(x) = x^\top w + b$ instead of a linear function $f(x) = x^\top w$, where $b \in \mathbb{R}$. We have the following optimization program

Program (RR3):

$$\begin{aligned} &\text{minimize} \quad \xi^\top \xi + K w^\top w \\ &\text{subject to} \\ &\qquad\qquad y - Xw - b\mathbf{1} = \xi, \end{aligned}$$

with $y, \xi, \mathbf{1} \in \mathbb{R}^m$ and $w \in \mathbb{R}^n$. Note that in Program (**RR3**) minimization is performed over ξ, w and b, but b is not penalized in the objective function. As in Section 19.1, the objective function is strictly convex.

The Lagrangian associated with this program is

$$L(\xi, w, b, \lambda) = \xi^\top \xi + K w^\top w - w^\top X^\top \lambda - \xi^\top \lambda - b\mathbf{1}^\top \lambda + \lambda^\top y.$$

Since L is (strictly) convex as a function of ξ, b, w, by Theorem 4.5(4), it has a minimum iff $\nabla L_{\xi,b,w} = 0$. We get

$$\begin{aligned} \lambda &= 2\xi \\ \mathbf{1}^\top \lambda &= 0 \\ w &= \frac{1}{2K} X^\top \lambda = X^\top \frac{\xi}{K}. \end{aligned}$$

As before, if we set $\xi = K\alpha$ we obtain $\lambda = 2K\alpha$, $w = X^\top \alpha$, and

$$G(\alpha) = -K\alpha^\top (XX^\top + KI_m)\alpha + 2K\alpha^\top y.$$

Since $K > 0$ and $\lambda = 2K\alpha$, the dual to ridge regression is the following program

Program (DRR3):

$$\begin{aligned}&\text{minimize} \quad \alpha^\top (XX^\top + KI_m)\alpha - 2\alpha^\top y\\ &\text{subject to}\\ &\qquad\qquad \mathbf{1}^\top \alpha = 0,\end{aligned}$$

where the minimization is over α.

Observe that up to the factor 1/2, this problem satisfies the conditions of Proposition 6.3 with

$$\begin{aligned}A &= (XX^\top + KI_m)^{-1}\\ b &= y\\ B &= \mathbf{1}_m\\ f &= 0,\end{aligned}$$

and x renamed as α. Therefore, it has a unique solution (α, μ) (beware that $\lambda = 2K\alpha$ is **not** the λ used in Proposition 6.3, which we rename as μ), which is the unique solution of the KKT-equations

$$\begin{pmatrix} XX^\top + KI_m & \mathbf{1}_m \\ \mathbf{1}_m^\top & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ \mu \end{pmatrix} = \begin{pmatrix} y \\ 0 \end{pmatrix}.$$

Since the solution given by Proposition 6.3 is

$$\mu = (B^\top AB)^{-1}(B^\top Ab - f), \quad \alpha = A(b - B\mu),$$

we get

$$\begin{aligned}\mu &= (\mathbf{1}^\top (XX^\top + KI_m)^{-1}\mathbf{1})^{-1}\mathbf{1}^\top (XX^\top + KI_m)^{-1} y\\ \alpha &= (XX^\top + KI_m)^{-1}(y - \mu\mathbf{1}).\end{aligned}$$

Note that the matrix $B^\top AB$ is the scalar $\mathbf{1}^\top (XX^\top + KI_m)^{-1}\mathbf{1}$, which is the negative of the Schur complement of $XX^\top + KI_m$.

Interestingly $b = \mu$, which is not obvious a priori.

Proposition 19.2. *We have $b = \mu$.*

Proof. To prove this result we need to express α differently. Since μ is a scalar, $\mu\mathbf{1} = \mathbf{1}\mu$, so

$$\mu\mathbf{1} = \mathbf{1}\mu = (\mathbf{1}^\top (XX^\top + KI_m)^{-1}\mathbf{1})^{-1}\mathbf{1}\mathbf{1}^\top (XX^\top + KI_m)^{-1} y,$$

and we obtain

$$\alpha = (XX^\top + KI_m)^{-1}(I_m - (\mathbf{1}^\top (XX^\top + KI_m)^{-1}\mathbf{1})^{-1}\mathbf{1}\mathbf{1}^\top (XX^\top + KI_m)^{-1})y. \quad (*_{\alpha_3})$$

Since $w = X^\top \alpha$, we have

$$w = X^\top (XX^\top + KI_m)^{-1}(I_m - (\mathbf{1}^\top (XX^\top + KI_m)^{-1}\mathbf{1})^{-1}\mathbf{1}\mathbf{1}^\top (XX^\top + KI_m)^{-1})y. \quad (*_{w_3})$$

From $\xi = K\alpha$, we deduce that b is given by the equation

$$b\mathbf{1} = y - Xw - K\alpha.$$

Since $w = X^\top \alpha$, using $(*_{\alpha_3})$ we obtain

$$\begin{aligned} b\mathbf{1} &= y - Xw - K\alpha \\ &= y - (XX^\top + KI_m)\alpha \\ &= y - (I_m - (\mathbf{1}^\top (XX^\top + KI_m)^{-1}\mathbf{1})^{-1}\mathbf{1}\mathbf{1}^\top (XX^\top + KI_m)^{-1})y \\ &= (\mathbf{1}^\top (XX^\top + KI_m)^{-1}\mathbf{1})^{-1}\mathbf{1}\mathbf{1}^\top (XX^\top + KI_m)^{-1})y \\ &= \mu\mathbf{1}, \end{aligned}$$

and thus

$$b = \mu = (\mathbf{1}^\top (XX^\top + KI_m)^{-1}\mathbf{1})^{-1}\mathbf{1}^\top (XX^\top + KI_m)^{-1}y, \qquad (*_{b_3})$$

as claimed. ☐

In summary the KKT-equations determine both α and μ, and so $w = X^\top \alpha$ and b as well.

There is also a useful expression of b as an average.

Since $\mathbf{1}^\top \mathbf{1} = m$ and $\mathbf{1}^\top \alpha = 0$, we get

$$b = \frac{1}{m}\mathbf{1}^\top y - \frac{1}{m}\mathbf{1}^\top Xw - \frac{1}{m}K\mathbf{1}^\top \alpha = \overline{y} - \sum_{j=1}^{n} \overline{X^j} w_j,$$

where $\overline{y}$ is the mean of y and $\overline{X^j}$ is the mean of the jth column of X. Therefore,

$$b = \overline{y} - \sum_{j=1}^{n} \overline{X^j} w_j = \overline{y} - (\overline{X^1} \cdots \overline{X^n})w,$$

where $(\overline{X^1} \cdots \overline{X^n})$ is the $1 \times n$ row vector whose jth entry is $\overline{X^j}$.

We will now show that solving the dual (**DRR3**) for α and obtaining $w = X^\top \alpha$ is equivalent to solving our previous ridge regression Problem (**RR2**) applied to the centered data $\widehat{y} = y - \overline{y}\mathbf{1}_m$ and $\widehat{X} = X - \overline{X}$, where $\overline{X}$ is the $m \times n$ matrix whose jth column is $\overline{X^j}\mathbf{1}_m$, the vector whose coordinates are all equal to the mean $\overline{X^j}$ of the jth column X^j of X.

The expression

$$b = \overline{y} - (\overline{X^1} \cdots \overline{X^n})w$$

suggests looking for an intercept term b (also called bias) of the above form, namely

Program (RR4):

$$\begin{aligned}
&\text{minimize} \quad \xi^\top \xi + K w^\top w \\
&\text{subject to} \\
&\qquad y - Xw - b\mathbf{1} = \xi \\
&\qquad b = \widehat{b} + \overline{y} - (\overline{X^1} \cdots \overline{X^n})w,
\end{aligned}$$

with $\widehat{b} \in \mathbb{R}$. Again, in Program (**RR4**), minimization is performed over ξ, w, b and $\widehat{b}$, but b and $\widehat{b}$ are not penalized.

Since

$$b\mathbf{1} = \widehat{b}\mathbf{1} + \overline{y}\mathbf{1} - (\overline{X^1}\mathbf{1} \cdots \overline{X^n}\mathbf{1})w,$$

if $\overline{X} = (\overline{X^1}\mathbf{1} \cdots \overline{X^n}\mathbf{1})$ is the $m \times n$ matrix whose jth column is the vector $\overline{X^j}\mathbf{1}$, then the above program is equivalent to the program

Program (RR5):

$$\begin{aligned}
&\text{minimize} \quad \xi^\top \xi + K w^\top w \\
&\text{subject to} \\
&\qquad y - Xw - \overline{y}\mathbf{1} + \overline{X}w - \widehat{b}\mathbf{1} = \xi,
\end{aligned}$$

where minimization is performed over ξ, w and $\widehat{b}$. If we write $\widehat{y} = y - \overline{y}\mathbf{1}$ and $\widehat{X} = X - \overline{X}$, then the above program becomes

Program (RR6):

$$\begin{aligned}
&\text{minimize} \quad \xi^\top \xi + K w^\top w \\
&\text{subject to} \\
&\qquad \widehat{y} - \widehat{X}w - \widehat{b}\mathbf{1} = \xi,
\end{aligned}$$

minimizing over ξ, w and $\widehat{b}$. If the solution to this program is $\widehat{w}$, then $\widehat{b}$ is given by

$$\widehat{b} = \overline{\widehat{y}} - (\overline{\widehat{X}^1} \cdots \overline{\widehat{X}^n})\widehat{w} = 0,$$

since the data $\widehat{y}$ and $\widehat{X}$ are centered. Therefore (**RR6**) *is equivalent to ridge regression without an intercept term applied to the centered data* $\widehat{y} = y - \overline{y}\mathbf{1}$ *and* $\widehat{X} = X - \overline{X}$,

Program (RR6′):

$$\begin{aligned}&\text{minimize} \quad \xi^\top \xi + K w^\top w\\ &\text{subject to}\\ &\qquad\qquad \widehat{y} - \widehat{X} w = \xi,\end{aligned}$$

minimizing over ξ and w.

If $\widehat{w}$ is the optimal solution of this program given by

$$\widehat{w} = \widehat{X}^\top (\widehat{X}\widehat{X}^\top + K I_m)^{-1} \widehat{y}, \qquad (*_{w_6})$$

then b is given by

$$b = \overline{y} - (\overline{X^1} \cdots \overline{X^n})\widehat{w}.$$

Remark: Although this is not obvious a priori, the optimal solution w^* of the Program (**RR3**) given by $(*_{w_3})$ is equal to the optimal solution $\widehat{w}$ of Program (**RR6′**) given by $(*_{w_6})$. We believe that it should be possible to prove the equivalence of these formulae but a proof eludes us at this time. We leave this as an open problem. In practice the Program (**RR6′**) involving the centered data appears to be the preferred one.

Example 19.1. Consider the data set (X, y_1) with

$$X = \begin{pmatrix} -10 & 11\\ -6 & 5\\ -2 & 4\\ 0 & 0\\ 1 & 2\\ 2 & -5\\ 6 & -4\\ 10 & -6 \end{pmatrix}, \quad y_1 = \begin{pmatrix} 0\\ -2.5\\ 0.5\\ -2\\ 2.5\\ -4.2\\ 1\\ 4 \end{pmatrix}$$

as illustrated in Figure 19.1. We find that $\overline{y} = -0.0875$ and $(\overline{X^1}, \overline{X^2}) = (0.125, 0.875)$. For the value $K = 5$, we obtain

$$w = \begin{pmatrix} 0.9207\\ 0.8677 \end{pmatrix}, \quad b = -0.9618,$$

for $K = 0.1$, we obtain

$$w = \begin{pmatrix} 1.1651\\ 1.1341 \end{pmatrix}, \quad b = -1.2255,$$

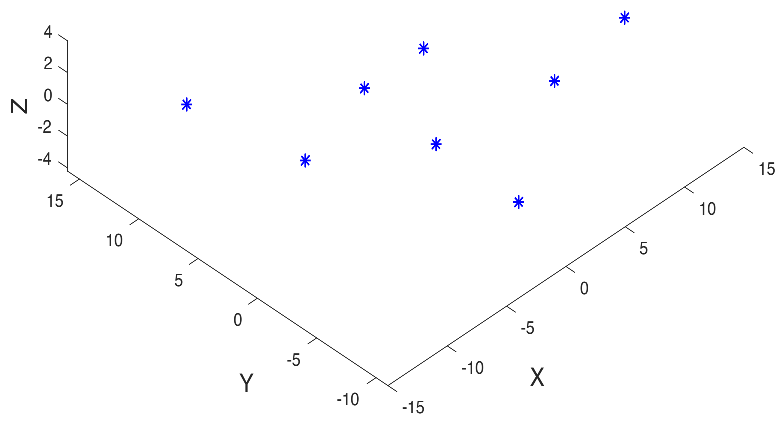


Fig. 19.1 The data set (X, y_1) of Example 19.1.

and for $K = 0.01$,

$$w = \begin{pmatrix} 1.1709 \\ 1.1405 \end{pmatrix}, \quad b = -1.2318.$$

See Figure 19.2.

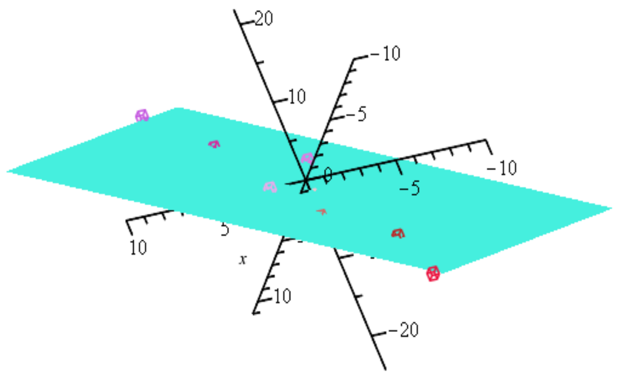


Fig. 19.2 The graph of the plane $f(x, y) = 1.1709x + 1.1405y - 1.2318$ as an approximate fit to the data (X, y_1) of Example 19.1.

We conclude that the points (X_i, y_i) (where X_i is the ith row of X) almost lie on the plane of equation

$$x + y - z - 1 = 0,$$

and that f is almost the function given by $f(x, y) = 1.1x + 1.1y - 1.2$. See Figures 19.3 and 19.4.

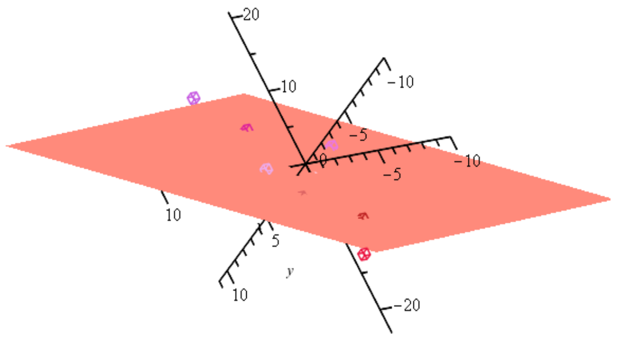


Fig. 19.3 The graph of the plane $f(x, y) = 1.1x + 1.1y - 1.2$ as an approximate fit to the data (X, y_1) of Example 19.1.

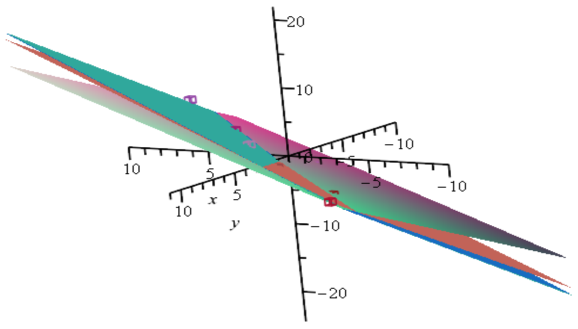


Fig. 19.4 A comparison of how the graphs of the planes corresponding to $K = 1, 0.1, 0.01$ and the salmon plane of equation $f(x, y) = 1.1x + 1.1y - 1.2$ approximate the data (X, y_1) of Example 19.1.

If we change y_1 to

$$y_2 = \begin{pmatrix}0 & -2 & 1 & -1 & 2 & -4 & 1 & 3\end{pmatrix}^\top,$$

as evidenced by Figure 19.5, the exact solution is

$$w = \begin{pmatrix}1\\1\end{pmatrix}, \quad b = -1,$$

and for $K = 0.01$, we find that

$$w = \begin{pmatrix}0.9999\\0.9999\end{pmatrix}, \quad b = -0.9999.$$

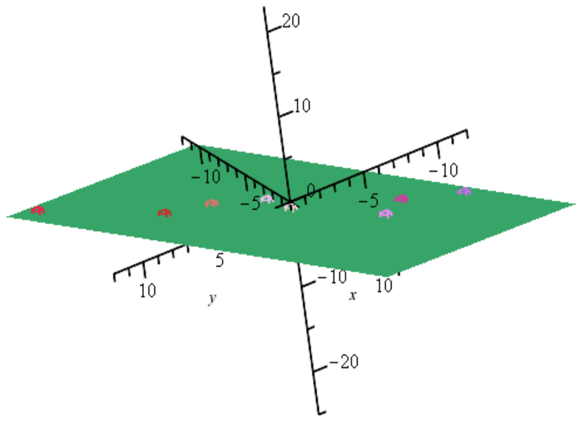


Fig. 19.5 The data (X, y_2) of Example 19.1 is contained within the graph of the plane $f(x, y) = x + y - 1$.

We can see how the choice of K affects the quality of the solution (w, b) by computing the norm $\|\xi\|_2$ of the error vector $\xi = \widehat{y} - \widehat{X}w$. We notice that the smaller K is, the smaller is this norm.

It is natural to wonder what happens *if we also penalize* b in program (**RR3**). Let us add the term Kb^2 to the objective function. Then we obtain the program

$$\begin{aligned}&\text{minimize} \quad \xi^\top \xi + K w^\top w + K b^2\\&\text{subject to}\\&\qquad\qquad y - Xw - b\mathbf{1} = \xi,\end{aligned}$$

minimizing over ξ, w and b.

This suggests treating b an an extra component of the weight vector w and by forming the $m \times (n+1)$ matrix $[X\ \mathbf{1}]$ obtained by adding a column of 1's (of dimension m) to the matrix X, we obtain

Program (RR3b):

$$\begin{aligned}&\text{minimize} \quad \xi^\top \xi + K w^\top w + K b^2\\ &\text{subject to}\end{aligned}$$

$$y - [X\ \mathbf{1}] \begin{pmatrix} w \\ b \end{pmatrix} = \xi,$$

minimizing over ξ, w and b.

This program is solved just as Program (**RR2**). In terms of the dual variable α, we get

$$\begin{aligned}\alpha &= ([X\ \mathbf{1}][X\ \mathbf{1}]^\top + K I_m)^{-1} y\\ \begin{pmatrix} w \\ b \end{pmatrix} &= [X\ \mathbf{1}]^\top \alpha\\ \xi &= K\alpha.\end{aligned}$$

Thus $b = \mathbf{1}^\top \alpha$. Observe that $[X\ \mathbf{1}][X\ \mathbf{1}]^\top = XX^\top + \mathbf{1}\mathbf{1}^\top$.

If $n < m$, it is preferable to use the formula

$$\begin{pmatrix} w \\ b \end{pmatrix} = ([X\ \mathbf{1}]^\top [X\ \mathbf{1}] + K I_{n+1})^{-1} [X\ \mathbf{1}]^\top y.$$

Since we also have the equation

$$y - Xw - b\mathbf{1} = \xi,$$

we obtain

$$\frac{1}{m}\mathbf{1}^\top y - \frac{1}{m}\mathbf{1}^\top X w - \frac{1}{m} b \mathbf{1}^\top \mathbf{1} = \frac{1}{m}\mathbf{1}^\top K\alpha,$$

so

$$\overline{y} - (\overline{X^1} \cdots \overline{X^n}) w - b = \frac{1}{m} K b,$$

which yields

$$b = \frac{m}{m+K}(\overline{y} - (\overline{X^1} \cdots \overline{X^n}) w).$$

Remark: As a least squares problem, the solution is given in terms of the pseudo-inverse $[X\ \mathbf{1}]^+$ of $[X\ \mathbf{1}]$ by

$$\begin{pmatrix} w \\ b \end{pmatrix} = [X\ \mathbf{1}]^+ y.$$

Example 19.2. Applying Program (**RR3**b) to the data set of Example 19.1 with $K = 0.01$ yields

$$w = \begin{pmatrix} 1.1706 \\ 1.1401 \end{pmatrix}, \quad b = -1.2298.$$

See Figure 19.6. We can see how the choice of K affects the quality of

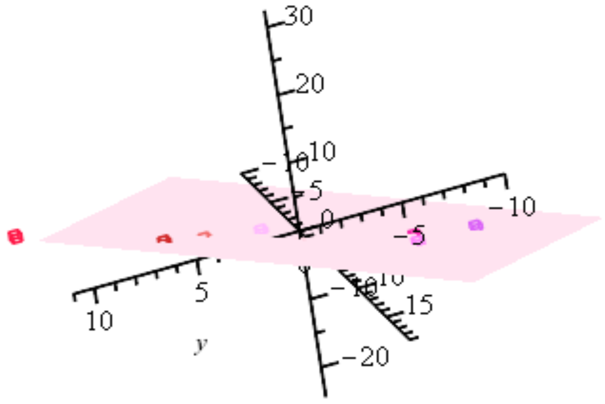


Fig. 19.6 The graph of the plane $f(x, y) = 1.1706x + 1.1401y - 1.2298$ as an approximate fit to the data (X, y_1) of Example 19.1.

the solution (w, b) by computing the norm $\|\xi\|_2$ of the error vector $\xi = y - Xw - b\mathbf{1}_m$. As in Example 19.1 we notice that the smaller K is, the smaller is this norm. We also observe that for a given value of K, Program (**RR6**$'$) gives a slightly smaller value of $\|\xi\|_2$ than (**RR3**b) does.

As pointed out by Hastie, Tibshirani, and Friedman [Hastie *et al.* (2009)] (Section 3.4), a defect of the approach where b is also penalized is that the solution for b is not invariant under adding a constant c to each value y_i. This is not the case for the approach using Program (**RR6**$'$).

19.3 Kernel Ridge Regression

One interesting aspect of the dual (of either (**RR2**) or (**RR3**)) is that it shows that the solution w being of the form $X^\top \alpha$, is a linear combination

$$w = \sum_{i=1}^{m} \alpha_i x_i$$

of the data points x_i, with the coefficients α_i corresponding to the dual variable $\lambda = 2K\alpha$ of the dual function, and with

$$\alpha = (XX^\top + KI_m)^{-1}y.$$

If m is smaller than n, then it is more advantageous to solve for α. But what really makes the dual interesting is that with our definition of X as

$$X = \begin{pmatrix} x_1^\top \\ \vdots \\ x_m^\top \end{pmatrix},$$

the matrix $XX^\top$ consists of the inner products $x_i^\top x_j$, and similarly the function learned $f(x) = x^\top w$ can be expressed as

$$f(x) = \sum_{i=1}^m \alpha_i x_i^\top x,$$

namely that both w and $f(x)$ are given *in terms of the inner products* $x_i^\top x_j$ and $x_i^\top x$.

This fact is the key to a generalization to ridge regression in which the input space $\mathbb{R}^n$ is embedded in a larger (possibly infinite dimensional) Euclidean space F (with an inner product $\langle -, - \rangle$) usually called a *feature space*, using a function

$$\varphi\colon \mathbb{R}^n \to F.$$

The problem becomes (*kernel ridge regression*)

Program (KRR2):

$$\begin{aligned} &\text{minimize} \quad \xi^\top \xi + K\langle w, w\rangle \\ &\text{subject to} \\ &\qquad\qquad y_i - \langle w, \varphi(x_i)\rangle = \xi_i, \quad i = 1, \ldots, m, \end{aligned}$$

minimizing over ξ and w. Note that $w \in F$. This problem is discussed in Shawe–Taylor and Christianini [Shawe-Taylor and Cristianini (2004)] (Section 7.3).

We will show below that the solution is exactly the same:

$$\begin{aligned} \alpha &= (\mathbf{G} + KI_m)^{-1}y \\ w &= \sum_{i=1}^m \alpha_i \varphi(x_i) \\ \xi &= K\alpha, \end{aligned}$$

where $\mathbf{G}$ is the Gram matrix given by $\mathbf{G}_{ij} = \langle \varphi(x_i), \varphi(x_j)\rangle$. This matrix is also called the *kernel matrix* and is often denoted by $\mathbf{K}$ instead of $\mathbf{G}$.

In this framework we have to be a little careful in using gradients since the inner product $\langle -, -\rangle$ on F is involved and F could be infinite dimensional, but this causes no problem because we can use derivatives, and by Proposition 3.5 we have

$$d\langle -, -\rangle_{(u,v)}(x, y) = \langle x, v\rangle + \langle u, y\rangle.$$

This implies that the derivative of the map $u \mapsto \langle u, u\rangle$ is

$$d\langle -, -\rangle_u(x) = 2\langle x, u\rangle. \tag{d_1}$$

Since the map $u \mapsto \langle u, v\rangle$ (with v fixed) is linear, its derivative is

$$d\langle -, v\rangle_u(x) = \langle x, v\rangle. \tag{d_2}$$

The derivative of the Lagrangian

$$L(\xi, w, \lambda) = \xi^\top \xi + K\langle w, w\rangle - \sum_{i=1}^m \lambda_i \langle \varphi(x_i), w\rangle - \xi^\top \lambda + \lambda^\top y$$

with respect to ξ and w is

$$dL_{\xi,w}\big(\widetilde{\xi}, \widetilde{w}\big) = 2(\widetilde{\xi})^\top \xi - (\widetilde{\xi})^\top \lambda + \left\langle 2Kw - \sum_{i=1}^m \lambda_i \varphi(x_i), \widetilde{w} \right\rangle,$$

where we used (d_1) to calculate the derivative of $\xi^\top \xi + K\langle w, w\rangle$ and (d_2) to calculate the derivative of $-\sum_{i=1}^m \lambda_i \langle \varphi(x_i), w\rangle - \xi^\top \lambda$. We have $dL_{\xi,w}\big(\widetilde{\xi}, \widetilde{w}\big) = 0$ for all $\widetilde{\xi}$ and $\widetilde{w}$ iff

$$\begin{aligned} 2Kw &= \sum_{i=1}^m \lambda_i \varphi(x_i) \\ \lambda &= 2\xi. \end{aligned}$$

Again we define $\xi = K\alpha$, so we have $\lambda = 2K\alpha$, and

$$w = \sum_{i=1}^m \alpha_i \varphi(x_i).$$

Plugging back into the Lagrangian we get

$$\begin{aligned} G(\alpha) &= K^2 \alpha^\top \alpha + K \sum_{i,j=1}^m \alpha_i \alpha_j \langle \varphi(x_i), \varphi(x_j)\rangle - 2K \sum_{i,j=1}^m \alpha_i \alpha_j \langle \varphi(x_i), \varphi(x_j)\rangle \\ &\quad - 2K^2 \alpha^\top \alpha + 2K\alpha^\top y \\ &= -K^2 \alpha^\top \alpha - K \sum_{i,j=1}^m \alpha_i \alpha_j \langle \varphi(x_i), \varphi(x_j)\rangle + 2K\alpha^\top y. \end{aligned}$$

If $\mathbf{G}$ is the matrix given by $\mathbf{G}_{ij} = \langle \varphi(x_i), \varphi(x_j)\rangle$, then we have

$$G(\alpha) = -K\alpha^\top (\mathbf{G} + KI_m)\alpha + 2K\alpha^\top y.$$

The function G is strictly concave, so by Theorem 4.5(4) it has a maximum for

$$\alpha = (\mathbf{G} + KI_m)^{-1} y,$$

as claimed earlier.

As in the standard case of ridge regression, if $F = \mathbb{R}^n$ (but the inner product $\langle -, -\rangle$ is arbitrary), we can adapt the above method to learn an affine function $f(w) = x^\top w + b$ instead of a linear function $f(w) = x^\top w$, where $b \in \mathbb{R}$. This time we assume that b is of the form

$$b = \overline{y} - \langle w, (\overline{X^1} \cdots \overline{X^n})\rangle,$$

where X^j is the j column of the $m \times n$ matrix X whose ith row is the transpose of the column vector $\varphi(x_i)$, and where $(\overline{X^1} \cdots \overline{X^n})$ is viewed as a column vector. We have the minimization problem

Program (KRR6$'$):

$$\begin{aligned} &\text{minimize} \quad \xi^\top \xi + K\langle w, w\rangle \\ &\text{subject to} \\ &\qquad \widehat{y}_i - \langle w, \widehat{\varphi(x_i)}\rangle = \xi_i, \quad i = 1, \ldots, m, \end{aligned}$$

minimizing over ξ and w, where $\widehat{\varphi(x_i)}$ is the n-dimensional vector $\varphi(x_i) - (\overline{X^1} \cdots \overline{X^n})$.

The solution is given in terms of the matrix $\widehat{\mathbf{G}}$ defined by

$$\widehat{\mathbf{G}}_{ij} = \langle \widehat{\varphi(x_i)}, \widehat{\varphi(x_j)}\rangle,$$

as before. We get

$$\alpha = (\widehat{\mathbf{G}} + KI_m)^{-1}\widehat{y},$$

and according to a previous computation, b is given by

$$b = \overline{y} - \frac{1}{m}\mathbf{1}\widehat{\mathbf{G}}\alpha.$$

We explained in Section 17.4 how to compute the matrix $\widehat{\mathbf{G}}$ from the matrix $\mathbf{G}$.

Since the dimension of the feature space F may be very large, one might worry that computing the inner products $\langle \varphi(x_i), \varphi(x_j)\rangle$ might be very expensive. This is where kernel functions come to the rescue. A *kernel function* κ for an embedding $\varphi\colon \mathbb{R}^n \to F$ is a map $\kappa\colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ with the property that

$$\kappa(u, v) = \langle \varphi(u), \varphi(v)\rangle \quad \text{for all } u, v \in \mathbb{R}^n.$$

If $\kappa(u, v)$ can be computed in a reasonably cheap way, and if $\varphi(u)$ can be computed cheaply, then the inner products $\langle \varphi(x_i), \varphi(x_j)\rangle$ (and $\langle \varphi(x_i), \varphi(x)\rangle$) can be computed cheaply; see Chapter 17. Fortunately there are good kernel functions. Two very good sources on kernel methods are Schölkopf and Smola [Schölkopf and Smola (2002)] and Shawe–Taylor and Christianini [Shawe-Taylor and Cristianini (2004)].

19.4 Lasso Regression (ℓ^1-Regularized Regression)

The main weakness of ridge regression is that the estimated weight vector w usually has many nonzero coefficients. As a consequence, ridge regression does not scale up well. In practice we need methods capable of handling millions of parameters, or more. A way to encourage sparsity of the vector w, which means that many coordinates of w are zero, is to replace the quadratic penalty function $\tau w^\top w = \tau \left\lVert w\right\rVert_2^2$ by the penalty function $\tau \left\lVert w\right\rVert_1$, with the ℓ^2-norm replaced by the ℓ^1-norm.

This method was first proposed by Tibshirani around 1996, under the name *lasso*, which stands for "least absolute selection and shrinkage operator." This method is also known as *ℓ^1-regularized regression*, but this is not as cute as "lasso," which is used predominantly.

Given a set of training data $\{(x_1, y_1), \ldots, (x_m, y_m)\}$, with $x_i \in \mathbb{R}^n$ and $y_i \in \mathbb{R}$, if X is the $m \times n$ matrix

$$X = \begin{pmatrix} x_1^\top \\ \vdots \\ x_m^\top \end{pmatrix},$$

in which the row vectors $x_i^\top$ are the rows of X, then *lasso regression* is the following optimization problem

Program (lasso1):

$$\begin{aligned} &\text{minimize} && \frac{1}{2}\xi^\top \xi + \tau \left\lVert w\right\rVert_1 \\ &\text{subject to} && \\ & && y - Xw = \xi, \end{aligned}$$

minimizing over ξ and w, where $\tau > 0$ is some constant determining the influence of the regularizing term $\left\lVert w\right\rVert_1$.

The difficulty with the regularizing term $\|w\|_1 = |w_1| + \cdots + |w_n|$ is that the map $w \mapsto \|w\|_1$ is not differentiable for all w. This difficulty can be overcome by using subgradients, but the dual of the above program can also be obtained in an elementary fashion by using a trick that we already used, which is that if $x \in \mathbb{R}$, then

$$|x| = \max\{x, -x\}.$$

Using this trick, by introducing a vector $\epsilon \in \mathbb{R}^n$ of nonnegative variables, we can rewrite lasso minimization as follows:

Program lasso regularization (lasso2):

$$\begin{aligned}
&\text{minimize} \quad \frac{1}{2}\xi^\top \xi + \tau \mathbf{1}_n^\top \epsilon \\
&\text{subject to} \\
&\qquad\qquad y - Xw = \xi \\
&\qquad\qquad w \le \epsilon \\
&\qquad\qquad -w \le \epsilon.
\end{aligned}$$

minimizing over ξ, w and ϵ, with $y, \xi \in \mathbb{R}^m$, and $w, \epsilon, \mathbf{1}_n \in \mathbb{R}^n$.

The constraints $w \le \epsilon$ and $-w \le \epsilon$ are equivalent to $|w_i| \le \epsilon_i$ for $i = 1, \ldots, n$, so for an optimal solution we must have $\epsilon \ge 0$ and $|w_i| = \epsilon_i$, that is, $\|w\|_1 = \epsilon_1 + \cdots + \epsilon_n$.

The Lagrangian $L(\xi, w, \epsilon, \lambda, \alpha_+, \alpha_-)$ is given by

$$\begin{aligned}
L(\xi, w, \epsilon, \lambda, \alpha_+, \alpha_-) &= \frac{1}{2}\xi^\top \xi + \tau \mathbf{1}_n^\top \epsilon + \lambda^\top (y - Xw - \xi) \\
&\quad + \alpha_+^\top (w - \epsilon) + \alpha_-^\top (-w - \epsilon) \\
&= \frac{1}{2}\xi^\top \xi - \xi^\top \lambda + \lambda^\top y \\
&\quad + \epsilon^\top (\tau \mathbf{1}_n - \alpha_+ - \alpha_-) + w^\top (\alpha_+ - \alpha_- - X^\top \lambda),
\end{aligned}$$

with $\lambda \in \mathbb{R}^m$ and $\alpha_+, \alpha_- \in \mathbb{R}^n_+$. Since the objective function is convex and the constraints are affine (and thus qualified), the Lagrangian L has a minimum with respect to the primal variables, ξ, w, ϵ iff $\nabla L_{\xi, w, \epsilon} = 0$. Since the gradient $\nabla L_{\xi, w, \epsilon}$ is given by

$$\nabla L_{\xi, w, \epsilon} = \begin{pmatrix} \xi - \lambda \\ \alpha_+ - \alpha_- - X^\top \lambda \\ \tau \mathbf{1}_n - \alpha_+ - \alpha_- \end{pmatrix},$$

we obtain the equations

$$\begin{aligned}
\xi &= \lambda \\
\alpha_+ - \alpha_- &= X^\top \lambda \\
\alpha_+ + \alpha_- &= \tau \mathbf{1}_n.
\end{aligned}$$

Using these equations, the dual function $G(\lambda, \alpha_+, \alpha_-) = \min_{\xi,w,\epsilon} L$ is given by

$$\begin{aligned} G(\lambda, \alpha_+, \alpha_-) &= \frac{1}{2}\xi^\top\xi - \xi^\top\lambda + \lambda^\top y = \frac{1}{2}\lambda^\top\lambda - \lambda^\top\lambda + \lambda^\top y \\ &= -\frac{1}{2}\lambda^\top\lambda + \lambda^\top y = -\frac{1}{2}\left(\|y-\lambda\|_2^2 - \|y\|_2^2\right), \end{aligned}$$

so

$$G(\lambda, \alpha_+, \alpha_-) = -\frac{1}{2}\left(\|y-\lambda\|_2^2 - \|y\|_2^2\right).$$

Since $\alpha_+, \alpha_- \geq 0$, for any $i \in \{1, \ldots, n\}$ the minimum of $(\alpha_+)_i - (\alpha_-)_i$ is $-\tau$, and the maximum is τ. If we recall that for any $z \in \mathbb{R}^n$,

$$\|z\|_\infty = \max_{1 \leq i \leq n} |z_i|,$$

it follows that the constraints

$$\begin{aligned} \alpha_+ + \alpha_- &= \tau \mathbf{1}_n \\ X^\top\lambda &= \alpha_+ - \alpha_- \end{aligned}$$

are equivalent to

$$\left\|X^\top\lambda\right\|_\infty \leq \tau.$$

The above is equivalent to the $2n$ constraints

$$-\tau \leq (X^\top\lambda)_i \leq \tau, \quad 1 \leq i \leq n.$$

Therefore, the dual lasso program is given by

$$\begin{aligned} &\text{maximize} \quad -\frac{1}{2}\left(\|y-\lambda\|_2^2 - \|y\|_2^2\right) \\ &\text{subject to} \\ &\qquad\qquad \left\|X^\top\lambda\right\|_\infty \leq \tau, \end{aligned}$$

which (since $\|y\|_2^2$ is a constant term) is equivalent to

Program (Dlasso2):

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2}\|y-\lambda\|_2^2 \\ &\text{subject to} \\ &\qquad\qquad \left\|X^\top\lambda\right\|_\infty \leq \tau, \end{aligned}$$

minimizing over $\lambda \in \mathbb{R}^m$.

One way to solve lasso regression is to use the dual program to find $\lambda = \xi$, and then to use linear programming to find w by solving the linear program arising from the lasso primal by holding ξ constant. The best way is to use ADMM as explained in Section 16.8(4). There are also a number of variations of gradient descent; see Hastie, Tibshirani, and Wainwright [Hastie *et al.* (2015)].

In theory, if we know the support of w and the signs of its components, then w is determined as we now explain.

In view of the constraint $y - Xw = \xi$ and the fact that for an optimal solution we must have $\xi = \lambda$, the following condition must hold:

$$\left\|X^\top (Xw - y)\right\|_\infty \leq \tau. \tag{$*$}$$

Also observe that for an optimal solution, we have

$$\begin{aligned}\frac{1}{2}\left\|y - Xw\right\|_2^2 + w^\top X^\top (y - Xw) &= \frac{1}{2}\left\|y\right\|^2 - w^\top X^\top y + \frac{1}{2} w^\top X^\top Xw \\ &\quad + w^\top X^\top y - w^\top X^\top Xw \\ &= \frac{1}{2}\left(\left\|y\right\|_2^2 - \left\|Xw\right\|_2^2\right) \\ &= \frac{1}{2}\left(\left\|y\right\|_2^2 - \left\|y - \lambda\right\|_2^2\right) = G(\lambda).\end{aligned}$$

Since the objective function is convex and the constaints are qualified, by Theorem 14.7(2) the duality gap is zero, so for optimal solutions of the primal and the dual, $G(\lambda) = L(\xi, w, \epsilon)$, that is

$$\frac{1}{2}\left\|y - Xw\right\|_2^2 + w^\top X^\top (y - Xw) = \frac{1}{2}\left\|\xi\right\|_2^2 + \tau\left\|w\right\|_1 = \frac{1}{2}\left\|y - Xw\right\|_2^2 + \tau\left\|w\right\|_1,$$

which yields the equation

$$w^\top X^\top (y - Xw) = \tau\left\|w\right\|_1. \tag{$**_1$}$$

The above is the inner product of w and $X^\top(y - Xw)$, so whenever $w_i \neq 0$, since $\left\|w\right\|_1 = |w_1| + \cdots + |w_n|$, in view of $(*)$, we must have $(X^\top(y - Xw))_i = \tau \mathrm{sgn}(w_i)$. If

$$S = \{i \in \{1, \ldots, n\} \mid w_i \neq 0\}, \tag{$\dagger$}$$

if X_S denotes the matrix consisting of the columns of X indexed by S, and if w_S denotes the vector consisting of the nonzero components of w, then we have

$$X_S^\top (y - X_S w_S) = \tau \mathrm{sgn}(w_S). \tag{$**_2$}$$

We also have

$$\left\|X_{\overline{S}}^\top (y - X_S w_S)\right\|_\infty \leq \tau, \tag{$**_3$}$$

where $\overline{S}$ is the complement of S.

Equation $(**_2)$ yields

$$X_S^\top X_S w_S = X_S^\top y - \tau \mathrm{sgn}(w_S),$$

so if $X_S^\top X_S$ is invertible (which will be the case if the columns of X are linearly independent), we get

$$w_S = (X_S^\top X_S)^{-1}(X_S^\top y - \tau \mathrm{sgn}(w_S)). \tag{$**_4$}$$

In theory, if we know the support of w and the signs of its components, then w_S is determined, but in practice this is useless since the problem is to find the support and the sign of the solution.

19.5 Lasso Regression; Learning an Affine Function

In the preceding section we made the simplifying assumption that we were trying to learn a linear function $f(x) = x^\top w$. To learn an affine function $f(x) = x^\top w + b$, we solve the following optimization problem

Program (lasso3):

$$\begin{aligned}
&\text{minimize} \quad \frac{1}{2}\xi^\top \xi + \tau \mathbf{1}_n^\top \epsilon\\
&\text{subject to}\\
&\qquad\qquad y - Xw - b\mathbf{1}_m = \xi\\
&\qquad\qquad w \leq \epsilon\\
&\qquad\qquad - w \leq \epsilon.
\end{aligned}$$

Observe that as in the case of ridge regression, minimization is performed over ξ, w, ϵ and b, but b is not penalized in the objective function.

The Lagrangian associated with this optimization problem is

$$\begin{aligned}
L(\xi, w, \epsilon, b, \lambda, \alpha_+, \alpha_-) &= \frac{1}{2}\xi^\top \xi - \xi^\top \lambda + \lambda^\top y - b\mathbf{1}_m^\top \lambda\\
&\quad + \epsilon^\top(\tau \mathbf{1}_n - \alpha_+ - \alpha_-) + w^\top(\alpha_+ - \alpha_- - X^\top \lambda),
\end{aligned}$$

so by setting the gradient $\nabla L_{\xi, w, \epsilon, b}$ to zero we obtain the equations

$$\begin{aligned}
\xi &= \lambda\\
\alpha_+ - \alpha_- &= X^\top \lambda\\
\alpha_+ + \alpha_- &= \tau \mathbf{1}_n\\
\mathbf{1}_m^\top \lambda &= 0.
\end{aligned}$$

Using these equations, we find that the dual function is also given by

$$G(\lambda, \alpha_+, \alpha_-) = -\frac{1}{2}\left(\|y - \lambda\|_2^2 - \|y\|_2^2\right),$$

and the dual lasso program is given by

$$\begin{aligned} &\text{maximize} \quad -\frac{1}{2}\left(\|y - \lambda\|_2^2 - \|y\|_2^2\right) \\ &\text{subject to} \\ &\qquad\qquad \left\|X^\top \lambda\right\|_\infty \le \tau \\ &\qquad\qquad \mathbf{1}_m^\top \lambda = 0, \end{aligned}$$

which is equivalent to

Program (Dlasso3):

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2}\|y - \lambda\|_2^2 \\ &\text{subject to} \\ &\qquad\qquad \left\|X^\top \lambda\right\|_\infty \le \tau \\ &\qquad\qquad \mathbf{1}_m^\top \lambda = 0, \end{aligned}$$

minimizing over $\lambda \in \mathbb{R}^m$.

Once $\lambda = \xi$ and w are determined, we obtain b using the equation

$$b\mathbf{1}_m = y - Xw - \xi,$$

and since $\mathbf{1}_m^\top \mathbf{1}_m = m$ and $\mathbf{1}_m^\top \xi = \mathbf{1}_m^\top \lambda = 0$, the above yields

$$b = \frac{1}{m}\mathbf{1}_m^\top y - \frac{1}{m}\mathbf{1}_m^\top Xw - \frac{1}{m}\mathbf{1}_m^\top \xi = \overline{y} - \sum_{j=1}^n \overline{X^j} w_j,$$

where $\overline{y}$ is the mean of y and $\overline{X^j}$ is the mean of the jth column of X.

The equation

$$b = \widehat{b} + \overline{y} - \sum_{j=1}^n \overline{X^j} w_j = \widehat{b} + \overline{y} - (\overline{X^1} \cdots \overline{X^n})w,$$

can be used as in ridge regression, (see Section 19.2), to show that the Program (**lasso3**) is equivalent to applying lasso regression (**lasso2**) without an intercept term to the centered data, by replacing y by $\widehat{y} = y - \overline{y}\mathbf{1}$ and X by $\widehat{X} = X - \overline{X}$. Then b is given by

$$b = \overline{y} - (\overline{X^1} \cdots \overline{X^n})\widehat{w},$$

where $\widehat{w}$ is the solution given by (**lasso2**). This is the method described by Hastie, Tibshirani, and Wainwright [Hastie *et al.* (2015)] (Section 2.2).

Example 19.3. We can create a data set (X, y) where X a 100×5 matrix and y is a 100×1 vector using the following `Matlab` program in which the command `randn` creates an array of normally distributed numbers.

```
X = randn(100,5);
ww = [0; 2; 0; -3; 0];
y = X*ww + randn(100,1)*0.1;
```

The purpose of the third line is to add some small noise to the "output" $X * ww$. The first five rows of X are

$$\begin{pmatrix} -1.1658 & -0.0679 & -1.6118 & 0.3199 & 0.4400 \\ -1.1480 & -0.1952 & -0.0245 & -0.5583 & -0.6169 \\ 0.1049 & -0.2176 & -1.9488 & -0.3114 & 0.2748 \\ 0.7223 & -0.3031 & 1.0205 & -0.5700 & 0.6011 \\ 2.5855 & 0.0230 & 0.8617 & -1.0257 & 0.0923 \end{pmatrix},$$

and the first five rows of y are

$$y = \begin{pmatrix} -1.0965 \\ 1.2155 \\ 0.4324 \\ 1.1902 \\ 3.1346 \end{pmatrix}.$$

We ran the program for lasso using ADMM (see Problem 16.7) with various values of ρ and τ, including $\rho = 1$ and $\rho = 10$. We observed that the program converges a lot faster for $\rho = 10$ than for $\rho = 1$. We plotted the values of the five components of $w(\tau)$ for values of τ from $\tau = 0$ to $\tau = 0.5$ by increment of 0.02, and observed that the first, third, and fifth coordinate drop basically linearly to zero (a value less that 10^{-4}) around $\tau = 0.2$. See Figures 19.7, 19.8, and 19.9. This behavior is also observed in Hastie, Tibshirani, and Wainwright [Hastie *et al.* (2015)] (Section 2.2).

For $\tau = 0.02$, we have

$$w = \begin{pmatrix} 0.00003 \\ 2.01056 \\ -0.00004 \\ -2.99821 \\ 0.00000 \end{pmatrix}, \quad b = 0.00135.$$

This weight vector w is very close to the original vector $ww = [0; 2; 0; -3; 0]$ that we used to create y. For large values of τ, the weight vector is essentially the zero vector. This happens for $\tau = 235$, where every component of w is less than 10^{-5}.

Another way to find b is to add the term $(C/2)b^2$ to the objective function, for some positive constant C, obtaining the program

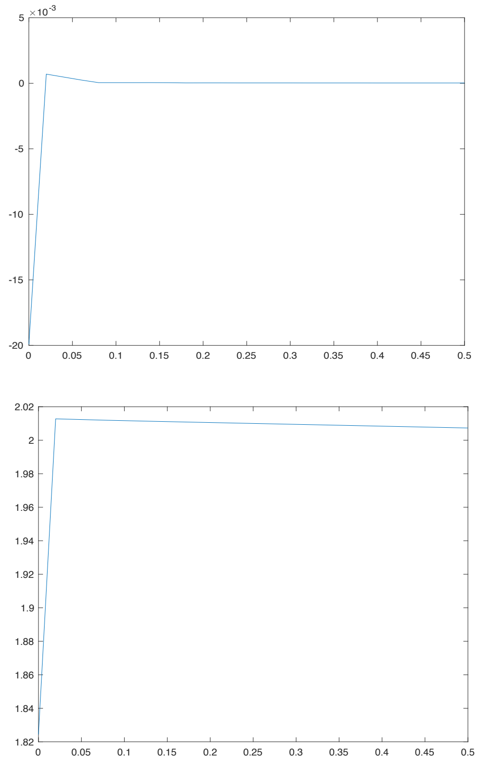


Fig. 19.7 First and second component of w.

Program(lasso4):

$$\begin{aligned}
&\text{minimize} && \frac{1}{2}\xi^\top \xi + \tau \mathbf{1}_n^\top \epsilon + \frac{1}{2} C b^2 \\
&\text{subject to} && \\
& && y - Xw - b\mathbf{1}_m = \xi \\
& && w \leq \epsilon \\
& && -w \leq \epsilon,
\end{aligned}$$

minimizing over ξ, w, ϵ and b.

This time the Lagrangian is

$$\begin{aligned}
L(\xi, w, \epsilon, b, \lambda, \alpha_+, \alpha_-) &= \frac{1}{2}\xi^\top \xi - \xi^\top \lambda + \lambda^\top y + \frac{C}{2} b^2 - b\mathbf{1}_m^\top \lambda \\
&\quad + \epsilon^\top (\tau \mathbf{1}_n - \alpha_+ - \alpha_-) + w^\top (\alpha_+ - \alpha_- - X^\top \lambda),
\end{aligned}$$

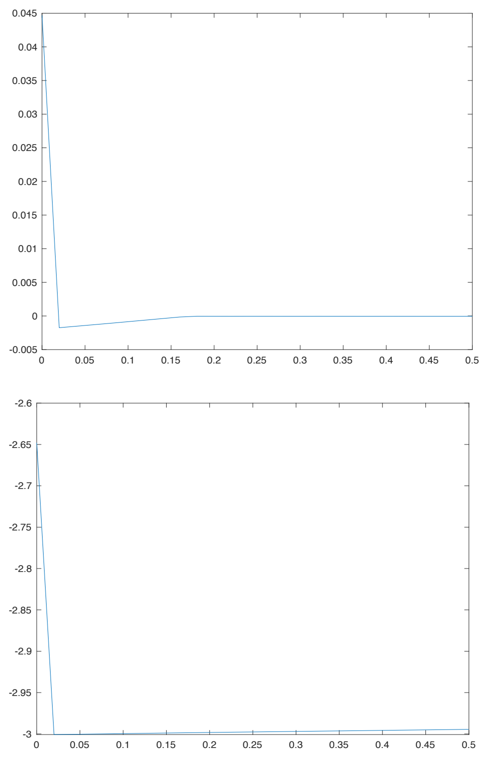


Fig. 19.8 Third and fourth component of w.

so by setting the gradient $\nabla L_{\xi,w,\epsilon,b}$ to zero we obtain the equations

$$\begin{aligned}\xi &= \lambda\\ \alpha_+ - \alpha_- &= X^\top \lambda\\ \alpha_+ + \alpha_- &= \tau \mathbf{1}_n\\ Cb &= \mathbf{1}_m^\top \lambda.\end{aligned}$$

Thus b is also determined, and the dual lasso program is identical to the first lasso dual (**Dlasso2**), namely

$$\begin{aligned}&\text{minimize} \quad \frac{1}{2}\left\|y - \lambda\right\|_2^2\\ &\text{subject to}\\ &\qquad\qquad \left\|X^\top \lambda\right\|_\infty \leq \tau,\end{aligned}$$

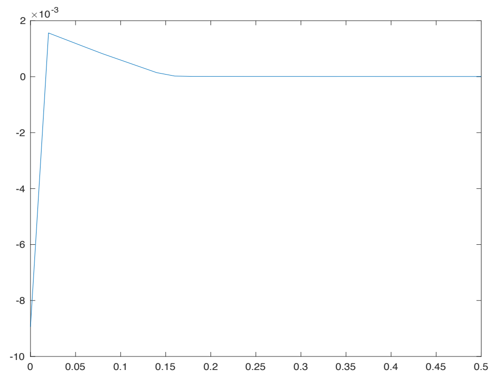


Fig. 19.9 Fifth component of w.

minimizing over λ.

Since the equations $\xi = \lambda$ and

$$y - Xw - b\mathbf{1}_m = \xi$$

hold, from $Cb = \mathbf{1}_m^\top \lambda$ we get

$$\frac{1}{m}\mathbf{1}_m^\top y - \frac{1}{m}\mathbf{1}_m^\top Xw - b\frac{1}{m}\mathbf{1}_m^\top \mathbf{1} = \frac{1}{m}\mathbf{1}_m^\top \lambda,$$

that is

$$\overline{y} - (\overline{X^1} \ \cdots \ \overline{X^n})w - b = \frac{C}{m}b,$$

which yields

$$b = \frac{m}{m+C}(\overline{y} - (\overline{X^1} \ \cdots \ \overline{X^n})w).$$

As in the case of ridge regression, a defect of the approach where b is also penalized is that the solution for b is not invariant under adding a constant c to each value y_i

It is interesting to compare the behavior of the methods:

(1) Ridge regression (**RR6**$'$) (which is equivalent to (**RR3**)).
(2) Ridge regression (**RR3**b), with b penalized (by adding the term Kb^2 to the objective function).
(3) Least squares applied to $[X \ \ \mathbf{1}]$.
(4) (**lasso3**).

When $n \leq 2$ and K and τ are small and of the same order of magnitude, say 0.1 or 0.01, there is no noticeable difference. We ran out programs on the data set of 200 points generated by the following `Matlab` program:

```
X14 = 15*randn(200,1);
ww14 = 1;
y14 = X14*ww14 + 10*randn(200,1) + 20;
```

The result is shown in Figure 19.10, with the following colors: Method (1) in magenta, Method (2) in red, Method (3) in blue, and Method (4) in cyan. All four lines are identical.

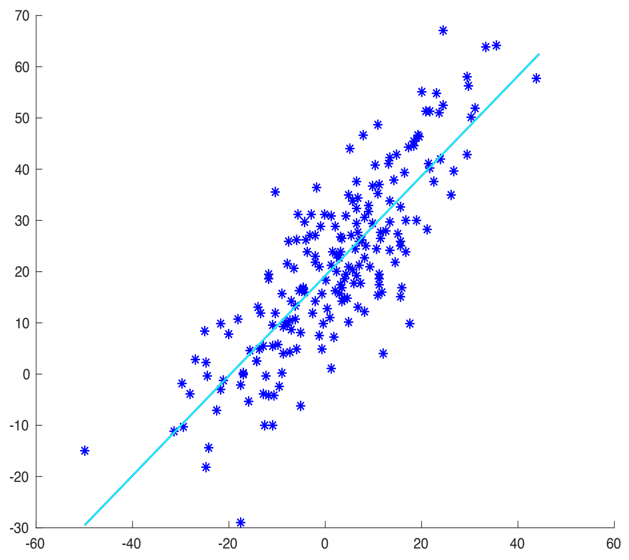


Fig. 19.10 Comparison of the four methods with $K = \tau = 0.1$.

In order to see a difference we also ran our programs with $K = 1000$ and $\tau = 10000$; see Figure 19.11.

As expected, due to the penalization of b, Method (3) yields a significantly lower line (in red), and due to the large value of τ, the line corresponding to lasso (in cyan) has a smaller slope. Method (1) (in magenta) also has a smaller slope but still does not deviate that much from least squares (in blue). It is also interesting to experiment on data sets where n is larger (say $20, 50$).

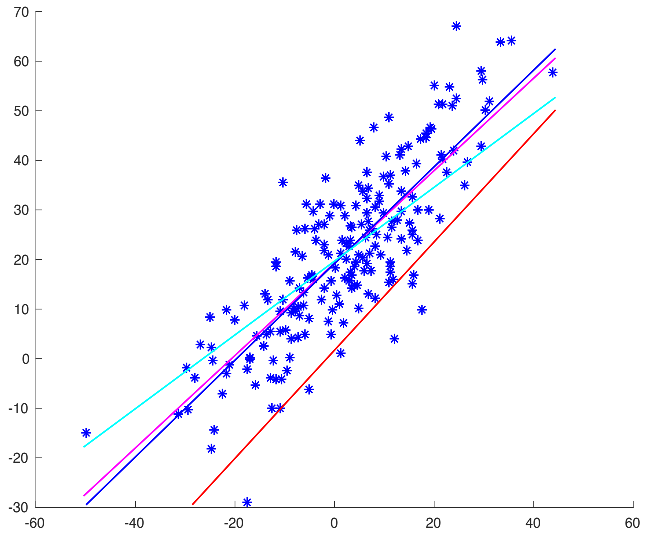


Fig. 19.11 Comparison of the four methods with $K = 1000, \tau = 10000$.

19.6 Elastic Net Regression

The lasso method is unsatisfactory when n (the dimension of the data) is much larger than the number m of data, because it only selects m coordinates and sets the others to values close to zero. It also has problems with groups of highly correlated variables. A way to overcome this problem is to add a "ridge-like" term $(1/2)Kw^\top w$ to the objective function. This way we obtain a hybrid of lasso and ridge regression called the *elastic net method* and defined as follows:

Program (elastic net):

$$\begin{aligned}
&\text{minimize} \quad \frac{1}{2}\xi^\top\xi + \frac{1}{2}Kw^\top w + \tau\mathbf{1}_n^\top\epsilon \\
&\text{subject to} \\
&\qquad y - Xw - b\mathbf{1}_m = \xi \\
&\qquad w \leq \epsilon \\
&\qquad -w \leq \epsilon,
\end{aligned}$$

where $K > 0$ and $\tau \geq 0$ are two constants controlling the influence of the

ℓ^2-regularization and the ℓ^1-regularization.[1] Observe that as in the case of ridge regression, minimization is performed over ξ, w, ϵ and b, but b is not penalized in the objective function. The objective function is strictly convex so if an optimal solution exists, then it is unique; the proof is left as an exercise.

The Lagrangian associated with this optimization problem is

$$\begin{aligned} L(\xi, w, \epsilon, b, \lambda, \alpha_+, \alpha_-) &= \frac{1}{2}\xi^\top \xi - \xi^\top \lambda + \lambda^\top y - b\mathbf{1}_m^\top \lambda + \frac{1}{2}Kw^\top w \\ &\quad + \epsilon^\top(\tau \mathbf{1}_n - \alpha_+ - \alpha_-) + w^\top(\alpha_+ - \alpha_- - X^\top \lambda), \end{aligned}$$

so by setting the gradient $\nabla L_{\xi,w,\epsilon,b}$ to zero we obtain the equations

$$\begin{aligned} \xi &= \lambda \\ Kw &= -(\alpha_+ - \alpha_- - X^\top \lambda) \qquad (*_w) \\ \alpha_+ + \alpha_- - \tau\mathbf{1}_n &= 0 \\ \mathbf{1}_m^\top \lambda &= 0. \end{aligned}$$

We find that $(*_w)$ determines w. Using these equations, we can find the dual function but in order to solve the dual using ADMM, since $\lambda \in \mathbb{R}^m$, it is more convenient to write $\lambda = \lambda_+ - \lambda_-$, with $\lambda_+, \lambda_- \in \mathbb{R}^m_+$ (recall that $\alpha_+, \alpha_- \in \mathbb{R}^n_+$). As in the derivation of the dual of ridge regression, we rescale our variables by defining $\beta_+, \beta_-, \mu_+, \mu_-$ such that

$$\alpha_+ = K\beta_+, \;\; \alpha_- = K\beta_-, \;\; \lambda_+ = K\mu_+, \;\; \lambda_- = K\mu_-.$$

We also let $\mu = \mu_+ - \mu_-$ so that $\lambda = K\mu$. Then $\mathbf{1}_m^\top \lambda = 0$ is equivalent to

$$\mathbf{1}_m^\top \mu_+ - \mathbf{1}_m^\top \mu_- = 0,$$

and since $\xi = \lambda = K\mu$, we have

$$\begin{aligned} \xi &= K(\mu_+ - \mu_-) \\ \beta_+ + \beta_- &= \frac{\tau}{K}\mathbf{1}_n. \end{aligned}$$

Using $(*_w)$ we can write

$$\begin{aligned} w &= -(\beta_+ - \beta_- - X^\top \mu) = -\beta_+ + \beta_- + X^\top \mu_+ - X^\top \mu_- \\ &= \begin{pmatrix} -I_n & I_n & X^\top & -X^\top \end{pmatrix} \begin{pmatrix} \beta_+ \\ \beta_- \\ \mu_+ \\ \mu_- \end{pmatrix}. \end{aligned}$$

[1] Some of the literature denotes K by λ_2 and τ by λ_1, but we prefer not to adopt this notation since we use λ, μ *etc.* to denote Lagrange multipliers.

Then we have

$$\begin{aligned}\begin{pmatrix}-I_n & I_n & X^\top & -X^\top\end{pmatrix}^\top \begin{pmatrix}-I_n & I_n & X^\top & -X^\top\end{pmatrix} &= \begin{pmatrix}-I_n \\ I_n \\ X \\ -X\end{pmatrix}\begin{pmatrix}-I_n & I_n & X^\top & -X^\top\end{pmatrix} \\ &= \begin{pmatrix} I_n & -I_n & -X^\top & X^\top \\ -I_n & I_n & X^\top & -X^\top \\ -X & X & XX^\top & -XX^\top \\ X & -X & -XX^\top & XX^\top\end{pmatrix}.\end{aligned}$$

If we define the symmetric positive semidefinite $2(n+m)\times 2(n+m)$ matrix Q as

$$Q = \begin{pmatrix} I_n & -I_n & -X^\top & X^\top \\ -I_n & I_n & X^\top & -X^\top \\ -X & X & XX^\top & -XX^\top \\ X & -X & -XX^\top & XX^\top\end{pmatrix},$$

then

$$\frac{1}{2}w^\top w = \frac{1}{2}\begin{pmatrix}\beta_+^\top & \beta_-^\top & \mu_+^\top & \mu_-^\top\end{pmatrix} Q \begin{pmatrix}\beta_+ \\ \beta_- \\ \mu_+ \\ \mu_-\end{pmatrix}.$$

As a consequence, using $(*_w)$ and the fact that $\xi = K\mu$, we find that the dual function is given by

$$\begin{aligned} G(\mu, \beta_+, \beta_-) &= \frac{1}{2}\xi^\top\xi - \xi^\top\lambda + \lambda^\top y + w^\top(\alpha_+ - \alpha_- - X^\top\lambda) + \frac{1}{2}Kw^\top w \\ &= \frac{1}{2}\xi^\top\xi - K\xi^\top\mu + K\mu^\top y + Kw^\top(\beta_+ - \beta_- - X^\top\mu) + \frac{1}{2}Kw^\top w \\ &= \frac{1}{2}K^2\mu^\top\mu - K^2\mu^\top\mu + Ky^\top\mu - Kw^\top w + \frac{1}{2}Kw^\top w \\ &= -\frac{1}{2}K^2\mu^\top\mu - \frac{1}{2}Kw^\top w + Ky^\top\mu. \end{aligned}$$

But

$$\mu = \begin{pmatrix}I_m & -I_m\end{pmatrix}\begin{pmatrix}\mu_+ \\ \mu_-\end{pmatrix},$$

so

$$\frac{1}{2}\mu^\top\mu = \frac{1}{2}\begin{pmatrix}\mu_+^\top & \mu_-^\top\end{pmatrix}\begin{pmatrix}I_m & -I_m \\ -I_m & I_m\end{pmatrix}\begin{pmatrix}\mu_+ \\ \mu_-\end{pmatrix},$$

so we get

$$G(\beta_+, \beta_-, \mu_+, \mu_-) = -\frac{1}{2}K \begin{pmatrix}\beta_+^\top & \beta_-^\top & \mu_+^\top & \mu_-^\top\end{pmatrix} P \begin{pmatrix}\beta_+ \\ \beta_- \\ \mu_+ \\ \mu_-\end{pmatrix} - Kq^\top \begin{pmatrix}\beta_+ \\ \beta_- \\ \mu_+ \\ \mu_-\end{pmatrix}$$

with

$$\begin{aligned} P &= Q + K \begin{pmatrix} 0_{n,n} & 0_{n,n} & 0_{n,m} & 0_{n,m} \\ 0_{n,n} & 0_{n,n} & 0_{n,m} & 0_{n,m} \\ 0_{m,n} & 0_{m,n} & I_m & -I_m \\ 0_{m,n} & 0_{m,n} & -I_m & I_m \end{pmatrix} \\ &= \begin{pmatrix} I_n & -I_n & -X^\top & X^\top \\ -I_n & I_n & X^\top & -X^\top \\ -X & X & XX^\top + KI_m & -XX^\top - KI_m \\ X & -X & -XX^\top - KI_m & XX^\top + KI_m \end{pmatrix}, \end{aligned}$$

and

$$q = \begin{pmatrix} 0_n \\ 0_n \\ -y \\ y \end{pmatrix}.$$

The constraints are the equations

$$\begin{aligned} \beta_+ + \beta_- &= \frac{\tau}{K}\mathbf{1}_n \\ \mathbf{1}_m^\top \mu_+ - \mathbf{1}_m^\top \mu_- &= 0, \end{aligned}$$

which correspond to the $(n+1) \times 2(n+m)$ matrix

$$A = \begin{pmatrix} I_n & I_n & 0_{n,m} & 0_{n,m} \\ 0_n^\top & 0_n^\top & \mathbf{1}_m^\top & -\mathbf{1}_m^\top \end{pmatrix}$$

and the right-hand side

$$c = \begin{pmatrix} \frac{\tau}{K}\mathbf{1}_n \\ 0 \end{pmatrix}.$$

Since $K > 0$, the dual of elastic net is equivalent to

Program (Dual Elastic Net):

$$\text{minimize} \quad \frac{1}{2}\begin{pmatrix}\beta_+^\top & \beta_-^\top & \mu_+^\top & \mu_-^\top\end{pmatrix} P \begin{pmatrix}\beta_+ \\ \beta_- \\ \mu_+ \\ \mu_-\end{pmatrix} + q^\top \begin{pmatrix}\beta_+ \\ \beta_- \\ \mu_+ \\ \mu_-\end{pmatrix}$$

subject to

$$A \begin{pmatrix}\beta_+ \\ \beta_- \\ \mu_+ \\ \mu_-\end{pmatrix} = c,$$

$\beta_+, \beta_- \in \mathbb{R}^n_+, \mu_+, \mu_- \in \mathbb{R}^m_+$.

Once $\xi = K\mu = K(\mu_+ - \mu_-)$ and w are determined by $(*_w)$, we obtain b using the equation

$$b\mathbf{1}_m = y - Xw - \xi,$$

which as in Section 19.5 yields

$$b = \overline{y} - \sum_{j=1}^n \overline{X^j} w_j,$$

where $\overline{y}$ is the mean of y and $\overline{X^j}$ is the mean of the jth column of X.

We leave it as an easy exercise to show that A has rank $n+1$. Then we can solve the above program using ADMM, and we have done so. This very similar to what is done in Section 20.3, and hence the details are left as an exercise.

Observe that when $\tau = 0$, the elastic net method reduces to ridge regression. As K tends to 0 the elastic net method tends to lasso, but $K = 0$ is not an allowable value since $\tau/0$ is undefined. Anyway, if we get rid of the constraint

$$\beta_+ + \beta_- = \frac{\tau}{K}\mathbf{1}_n$$

the corresponding optimization program not does determine w. Experimenting with our program we found that convergence becomes very slow for $K < 10^{-3}$. What we can do if K is small, say $K < 10^{-3}$, is to run lasso. A nice way to "blend" ridge regression and lasso is to call the elastic net method with $K = C(1-\theta)$ and $\tau = C\theta$, where $0 \leq \theta < 1$ and $C > 0$.

Running the elastic net method on the data set $(X14, y14)$ of Section 19.5 with $K = \tau = 0.5$ shows absolutely no difference, but the reader should conduct more experiments to see how elastic net behaves as K and τ are varied (the best way to do this is to use θ as explained above). Here is an example involving a data set $(X20, y20)$ where $X20$ is a 200×30 matrix generated as follows:

```
X20 = randn(50,30);
ww20 = [0; 2; 0; -3; 0; -4; 1; 0; 2; 0; 2; 3; 0; -5; 6; 0; 1;
2; 0; 10; 0; 0; 3; 4; 5; 0; 0; -6; -8; 0];
y20 = X20*ww20 + randn(50,1)*0.1 + 5;
```

Running our program with $K = 0.01$ and $K = 0.99$, and then with $K = 0.99$ and $K = 0.01$, we get the following weight vectors (in the left column is the weight vector corresponding to $K = 0.01$ and $K = 0.99$):

```
0.0254            0.2007
1.9193            2.0055
0.0766            0.0262
-3.0014           -2.8008
0.0512            0.0089
-3.8815           -3.7670
0.9591            0.8552
-0.0086           -0.3243
1.9576            1.9080
-0.0077           -0.1041
1.9881            2.0566
2.9223            2.8346
-0.0046           -0.0832
-4.9989           -4.8332
5.8640            5.4598
-0.0207           -0.2141
0.8285            0.8585
1.9310            1.8559
0.0046            0.0413
9.9232            9.4836
-0.0216           0.0303
0.0453            -0.0193
2.9384            3.0004
4.0525            3.9753
4.8723            4.6530
0.0767            0.1192
```

```
0.0132           -0.0203
-5.9750          -5.7537
-7.9764          -7.7594
-0.0054          0.0528
```

Generally, the numbers in the left column, which are more "lasso-like," have clearer zeros and nonzero values closer to those of the weight vector $ww20$ that was used to create the data set. The value of b corresponding to the first call is $b = 5.1372$, and the value of b corresponding to the second call is $b = 5.208$.

We have observed that lasso seems to converge much faster than elastic net when $K < 10^{-3}$. For example, running the above data set with $K = 10^{-3}$ and $\tau = 0.999$ requires 140520 steps to achieve primal and dual residuals less than 10^{-7}, but lasso only takes 86 steps to achieve the same degree of convergence. We observed that the larger K is the faster is the convergence. This could be attributed to the fact that the matrix P becomes more "positive definite." Another factor is that ADMM for lasso solves an $n \times n$ linear system, but ADMM for elastic net solves a $2(n + m) \times 2(n + m)$ linear system. So even though elastic net does not suffer from some of the undesirable properties of ridge regression and lasso, it appears to have a slower convergence rate, in fact much slower when K is small (say $K < 10^{-3}$). It also appears that elastic net may be too expensive a choice if m is much larger than n. Further investigations are required to gain a better understanding of the convergence issue.

19.7 Summary

The main concepts and results of this chapter are listed below:

- Ridge regression.
- Kernel ridge regression.
- Kernel functions.
- Lasso regression.
- Elastic net regression.

19.8 Problems

Problem 19.1. Check the formula

$$(X^\top X + KI_n)^{-1}X^\top = X^\top(XX^\top + KI_m)^{-1},$$

stated in Section 19.1.

Problem 19.2. Implement the ridge regression method described in Section 19.1 in `Matlab`. Also implement ridge regression with intercept and compare solving Program (**DRR3**) and Program (**RR6′**) using centered data.

Problem 19.3. Implement the ridge regression with intercept method (**RR3b**) in `Matlab` and compare it with solving (**RR6′**) using centered data.

Problem 19.4. Verify that (**lasso3**) is equivalent to (**lasso2**) applied to the centered data $\widehat{y} = y - \overline{y}\mathbf{1}$ and $\widehat{X} = X - \overline{X}$.

Problem 19.5. Verify the fomulae obtained for the kernel ridge regression program (**KRR6′**).

Problem 19.6. Implement in `Matlab` and test (**lasso3**) for various values of ρ and τ. Write a program to plot the coordinates of w as a function of τ. Compare the behavior of lasso with ridge regression (**RR6′**), (**RR3**b) (b penalized), and with least squares.

Problem 19.7. Check the details of the derivation of the dual of elastic net.

Problem 19.8. Write a `Matlab` program, solving the dual of elastic net; use inspiration from Section 20.3. Run tests to compare the behavior of ridge regression, lasso, and elastic net.

Problem 19.9. Prove that if an optimal solution exists for the elastic net method, then it is unique.

Problem 19.10. Prove that the matrix

$$P = \begin{pmatrix} I_n & -I_n & -X^\top & X^\top \\ -I_n & I_n & X^\top & -X^\top \\ -X & X & XX^\top + KI_m & -XX^\top - KI_m \\ X & -X & -XX^\top - KI_m & XX^\top + KI_m \end{pmatrix}$$

is almost positive definite, in the sense that

$$\begin{pmatrix} \beta_+^\top & \beta_-^\top & \mu_+^\top & \mu_-^\top \end{pmatrix} P \begin{pmatrix} \beta_+ \\ \beta_- \\ \mu_+ \\ \mu_- \end{pmatrix} = 0$$

if and only if $\beta_+ = \beta_-$ and $\mu_+ = \mu_-$, that is, $\beta = 0$ and $\mu = 0$.

Chapter 20

ν-SV Regression

20.1 ν-SV Regression; Derivation of the Dual

Let $\{(x_1, y_1), \ldots, (x_m, y_m)\}$ be a set of observed data usually called a set of *training data*, with $x_i \in \mathbb{R}^n$ and $y_i \in \mathbb{R}$. As in Chapter 19, we form the $m \times n$ matrix X where the row vectors $x_i^\top$ are the *rows* of X. Our goal is to learn an affine function f of the form $f(x) = x^\top w + b$ that fits the set of training data, but does not penalize errors below some given $\epsilon \geq 0$. Geometrically, we view the pairs (x_i, y_i) are points in $\mathbb{R}^{n+1}$, and we try to fit a hyperplane $H_{w,b}$ of equation

$$(w^\top \;\; -1) \begin{pmatrix} x \\ z \end{pmatrix} + b = w^\top x - z + b = 0$$

that best fits the set of points (x_i, y_i) (where $(x, z) \in \mathbb{R}^{n+1}$). We seek an $\epsilon > 0$ such that most points (x_i, y_i) are inside the slab (or tube) of width 2ϵ bounded by the hyperplane $H_{w,b-\epsilon}$ of equation

$$(w^\top \;\; -1) \begin{pmatrix} x \\ z \end{pmatrix} + b - \epsilon = w^\top x - z + b - \epsilon = 0$$

and the hyperplane $H_{w,b+\epsilon}$ of equation

$$(w^\top \;\; -1) \begin{pmatrix} x \\ z \end{pmatrix} + b + \epsilon = w^\top x - z + b + \epsilon = 0.$$

Observe that the hyperplanes $H_{w,b-\epsilon}, H_{w,b}$ and $H_{w,b+\epsilon}$ intersect the z-axis when $x = 0$ for the values $(b - \epsilon, b, b + \epsilon)$. Since $\epsilon \geq 0$, the hyperplane $H_{w,b-\epsilon}$ is below the hyperplane $H_{w,b}$ which is below the hyperplane $H_{w,b+\epsilon}$. We refer to the lower hyperplane $H_{w,b-\epsilon}$ as the *blue margin*, to the upper hyperplane $H_{w,b+\epsilon}$ as the *red margin*, and to the hyperplane $H_{w,b}$ as the *best fit hyperplane*. Also note that since the term $-z$ appears in the equations of these hyperplanes, points for which $w^\top x - z + b \leq 0$ are *above* the hyperplane $H_{w,b}$, and points for which $w^\top x - z + b \geq 0$ are *below* the hyperplane $H_{w,b}$ (and similarly for $H_{w,b-\epsilon}$ and $H_{b+\epsilon}$). The region bounded by the hyperplanes $H_{w,b-\epsilon}$ and $H_{b+\epsilon}$ (which contains the best fit hyperplane $H_{w,b}$) is called the *ϵ-slab*.

We also allow *errors* by allowing the point (x_i, y_i) to be outside of the ϵ-slab but in the slab between the hyperplane $H_{w,b-\epsilon-\xi_i}$ of equation

$$(w^\top \ \ -1)\begin{pmatrix} x \\ z \end{pmatrix} + b - \epsilon - \xi_i = w^\top x - z + b - \epsilon - \xi_i = 0$$

for some $\xi_i > 0$ (which is below the blue margin hyperplane $H_{w,b-\epsilon}$) and the hyperplane $H_{w,b+\epsilon+\xi_i'}$ of equation

$$(w^\top \ \ -1)\begin{pmatrix} x \\ z \end{pmatrix} + b + \epsilon + \xi_i' = w^\top x_i - z + b + \epsilon + \xi_i' = 0$$

for some $\xi_i' > 0$ (which is above the red margin hyperplane $H_{w,b+\epsilon}$), so that $w^\top x_i - y_i + b - \epsilon - \xi_i \leq 0$ and $w^\top x_i - y_i + b + \epsilon + \xi_i' \geq 0$, that is,

$$\begin{aligned} f(x) - y_i = w^\top x_i + b - y_i &\leq \epsilon + \xi_i, \\ -(f(x) - y_i) = -w^\top x_i - b + y_i &\leq \epsilon + \xi_i'. \end{aligned}$$

Our goal is to minimize ϵ and the errors ξ_i and ξ_i'. See Figure 20.1. The trade off between the size of ϵ and the size of the slack variables ξ_i and ξ_i' is achieved by using two constants $\nu \geq 0$ and $C > 0$. The method of *ν-support vector regression*, for short *ν-SV regression*, is specified by the following minimization problem:

Program ν-SV Regression:

$$\text{minimize} \quad \frac{1}{2}w^\top w + C\left(\nu\epsilon + \frac{1}{m}\sum_{i=1}^m (\xi_i + \xi_i')\right)$$

subject to

$$\begin{aligned} w^\top x_i + b - y_i &\leq \epsilon + \xi_i, \quad && \xi_i \geq 0 \qquad i = 1, \ldots, m \\ -w^\top x_i - b + y_i &\leq \epsilon + \xi_i', \quad && \xi_i' \geq 0 \qquad i = 1, \ldots, m \\ \epsilon &\geq 0, \end{aligned}$$

minimizing over the variables w, b, ϵ, ξ, and ξ'. The constraints are affine. The problem is to minimize ϵ and the errors ξ_i, ξ_i' so that the ℓ^1-error is "squeezed down" to zero as much as possible, in the sense that the right-hand side of the inequality

$$\sum_{i=1}^m |y_i - x_i^\top w - b| \leq m\epsilon + \sum_{i=1}^m \xi_i + \sum_{i=1}^m \xi_i'$$

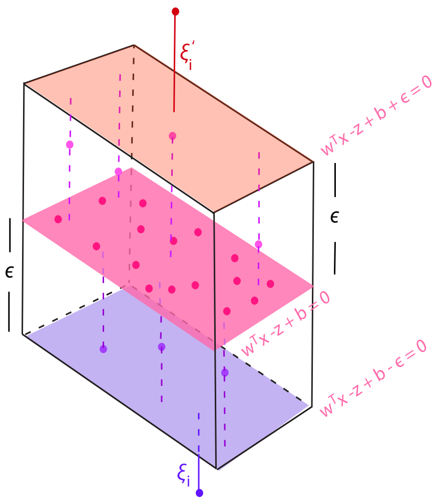


Fig. 20.1 The ϵ-slab around the graph of the best fit affine function $f(x) = x^\top w + b$.

is as small as possible. As shown by Figure 20.2, the region associated with the constraint $w^\top x_i - z + b \leq \epsilon$ contains the ϵ-slab.

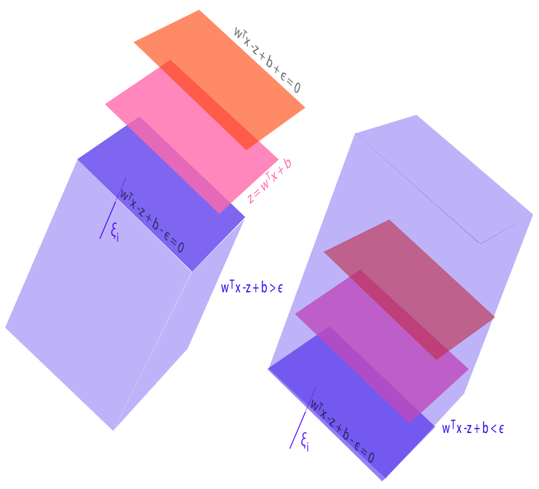


Fig. 20.2 The two blue half spaces associated with the hyperplane $w^\top x_i - z + b = \epsilon$.

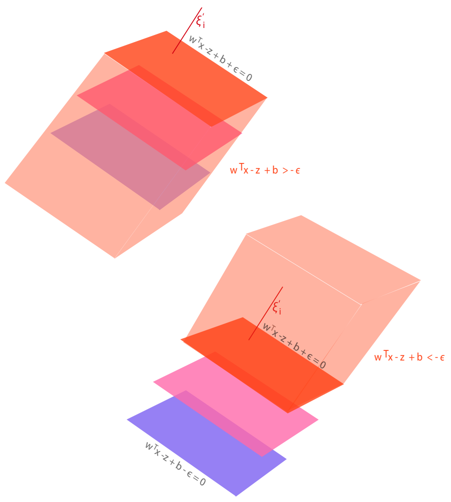

ξ_i'
$w^T x - z + b + \epsilon = 0$
$w^T x - z + b > -\epsilon$
ξ_i'
$w^T x - z + b + \epsilon = 0$
$w^T x - z + b < -\epsilon$
$w^T x - z + b - \epsilon = 0$

Fig. 20.3 The two red half spaces associated with the hyperplane $w^\top x_i - z + b = -\epsilon$.

Similarly, as illustrated by Figure 20.3, the region associated with the constraint $w^\top x_i - z + b \geq -\epsilon$, equivalently $-w^\top x_i + z - b \leq \epsilon$, also contains the ϵ-slab.

Observe that if we require $\epsilon = 0$, then the problem is equivalent to minimizing

$$\|y - Xw - b\mathbf{1}\|_1 + \frac{1}{2} w^\top w.$$

Thus it appears that the above problem is the version of Program (**RR3**) (see Section 19.2) in which the ℓ^2-norm of $y - Xw - b\mathbf{1}$ is replaced by its ℓ^1-norm. This a sort of "dual" of lasso (see Section 19.5) where $(1/2)w^\top w = (1/2)\|w\|_2^2$ is replaced by $\tau\|w\|_1$, and $\|y - Xw - b\mathbf{1}\|_1$ is replaced by $\|y - Xw - b\mathbf{1}\|_2^2$.

Proposition 20.1. *For any optimal solution, the equations*

$$\xi_i \xi_i' = 0, \quad i = 1, \ldots, m \qquad (\xi\xi')$$

hold. If $\epsilon > 0$, then the equations

$$\begin{aligned} w^\top x_i + b - y_i &= \epsilon + \xi_i \\ -w^\top x_i - b + y_i &= \epsilon + \xi_i' \end{aligned}$$

cannot hold simultaneously.

Proof. For an optimal solution we have

$$-\epsilon - \xi_i' \leq w^\top x_i + b - y_i \leq \epsilon + \xi_i.$$

If $w^\top x_i + b - y_i \geq 0$, then $\xi_i' = 0$ since the inequality

$$-\epsilon - \xi_i' \leq w^\top x_i + b - y_i$$

is trivially satisfied (because $\epsilon, \xi_i' \geq 0$), and if $w^\top x_i + b - y_i \leq 0$, then similarly $\xi_i = 0$. See Figure 20.4.

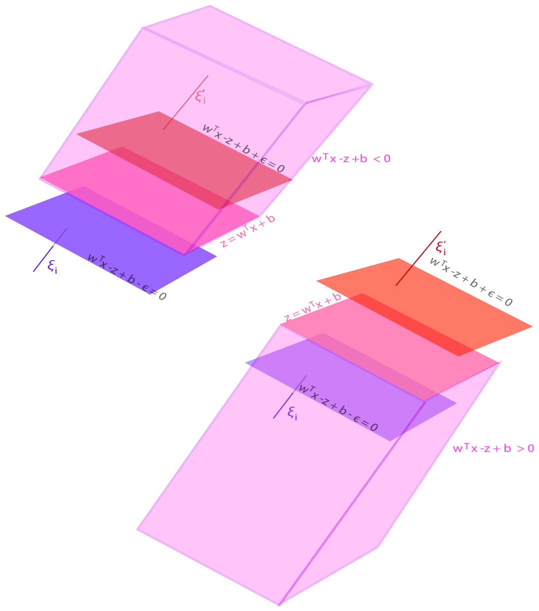


Fig. 20.4 The two pink open half spaces associated with the hyperplane $w^\top x_i - z + b = 0$. Note, $\xi_i > 0$ is outside of the half space $w^\top x_i - z + b - \epsilon < 0$, and $\xi_i' > 0$ is outside of the half space $w^\top x_i - z + b + \epsilon > 0$.

Observe that the equations

$$w^\top x_i + b - y_i = \epsilon + \xi_i$$
$$-w^\top x_i - b + y_i = \epsilon + \xi_i'$$

can only hold simultaneously if

$$\epsilon + \xi_i = -\epsilon - \xi',$$

that is,

$$2\epsilon + \xi_i + \xi_i' = 0,$$

and since $\epsilon, \xi_i, \xi'_i \geq 0$, this can happen only if $\epsilon = \xi_i = \xi'_i = 0$, and then

$$w^\top x_i + b = y_i.$$

In particular, if $\epsilon > 0$, then the equations

$$\begin{aligned} w^\top x_i + b - y_i &= \epsilon + \xi_i \\ -w^\top x_i - b + y_i &= \epsilon + \xi'_i \end{aligned}$$

cannot hold simultaneously. $\square$

Observe that if $\nu > 1$, then an optimal solution of the above program must yield $\epsilon = 0$. Indeed, if $\epsilon > 0$, we can reduce it by a small amount $\delta > 0$ and increase $\xi_i + \xi'_i$ by δ to still satisfy the constraints, but the objective function changes by the amount $-\nu\delta + \delta$, which is negative since $\nu > 1$, so $\epsilon > 0$ is not optimal.

Driving ϵ to zero is not the intended goal, because typically the data is not noise free so very few pairs (x_i, y_i) will satisfy the equation $w^\top x_i + b = y_i$, and then many pairs (x_i, y_i) will correspond to an error ($\xi_i > 0$ or $\xi'_i > 0$). Thus, *typically we assume that* $0 < \nu \leq 1$.

To construct the Lagrangian, we assign Lagrange multipliers $\lambda_i \geq 0$ to the constraints $w^\top x_i + b - y_i \leq \epsilon + \xi_i$, Lagrange multipliers $\mu_i \geq 0$ to the constraints $-w^\top x_i - b + y_i \leq \epsilon + \xi'_i$, Lagrange multipliers $\alpha_i \geq 0$ to the constraints $\xi_i \geq 0$, Lagrange multipliers $\beta_i \geq 0$ to the constraints $\xi'_i \geq 0$, and the Lagrange multiplier $\gamma \geq 0$ to the constraint $\epsilon \geq 0$. The Lagrangian is

$$\begin{aligned} L(w, b, \lambda, \mu, \gamma, \xi, \xi', \epsilon, \alpha, \beta) = \frac{1}{2} w^\top w + C\left(\nu\epsilon + \frac{1}{m}\sum_{i=1}^m (\xi_i + \xi'_i)\right) - \gamma\epsilon \\ - \sum_{i=1}^m (\alpha_i \xi_i + \beta_i \xi'_i) + \sum_{i=1}^m \lambda_i (w^\top x_i + b - y_i - \epsilon - \xi_i) \\ + \sum_{i=1}^m \mu_i (-w^\top x_i - b + y_i - \epsilon - \xi'_i). \end{aligned}$$

The Lagrangian can also be written as

$$\begin{aligned} L(w, b, \lambda, \mu, \gamma, \xi, \xi', \epsilon, \alpha, \beta) = \frac{1}{2} w^\top w + w^\top \left(\sum_{i=1}^m (\lambda_i - \mu_i) x_i\right) \\ + \epsilon\left(C\nu - \gamma - \sum_{i=1}^m (\lambda_i + \mu_i)\right) + \sum_{i=1}^m \xi_i \left(\frac{C}{m} - \lambda_i - \alpha_i\right) \\ + \sum_{i=1}^m \xi'_i \left(\frac{C}{m} - \mu_i - \beta_i\right) + b\left(\sum_{i=1}^m (\lambda_i - \mu_i)\right) - \sum_{i=1}^m (\lambda_i - \mu_i) y_i. \end{aligned}$$

To find the dual function $G(\lambda, \mu, \gamma, \alpha, \beta)$, we minimize $L(w, b, \lambda, \mu, \gamma, \xi, \xi', \epsilon, \alpha, \beta)$ with respect to the primal variables w, ϵ, b, ξ and ξ'. Observe that the Lagrangian is convex, and since $(w, \epsilon, \xi, \xi', b) \in \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^m \times \mathbb{R}^m \times \mathbb{R}$, a convex open set, by Theorem 4.5, the Lagrangian has a minimum iff $\nabla L_{w,\epsilon,b,\xi,\xi'} = 0$, so we compute the gradient $\nabla L_{w,\epsilon,b,\xi,\xi'}$. We obtain

$$\nabla L_{w,\epsilon,b,\xi,\xi'} = \begin{pmatrix} w + \sum_{i=1}^m (\lambda_i - \mu_i) x_i \\ C\nu - \gamma - \sum_{i=1}^m (\lambda_i + \mu_i) \\ \sum_{i=1}^m (\lambda_i - \mu_i) \\ \frac{C}{m} - \lambda - \alpha \\ \frac{C}{m} - \mu - \beta \end{pmatrix},$$

where

$$\left(\frac{C}{m} - \lambda - \alpha\right)_i = \frac{C}{m} - \lambda_i - \alpha_i, \quad \text{and} \quad \left(\frac{C}{m} - \mu - \beta\right)_i = \frac{C}{m} - \mu_i - \beta_i.$$

Consequently, if we set $\nabla L_{w,\epsilon,b,\xi,\xi'} = 0$, we obtain the equations

$$\begin{aligned} w &= \sum_{i=1}^m (\mu_i - \lambda_i) x_i = X^\top (\mu - \lambda), \qquad (*_w) \\ C\nu - \gamma - \sum_{i=1}^m (\lambda_i + \mu_i) &= 0 \\ \sum_{i=1}^m (\lambda_i - \mu_i) &= 0 \\ \frac{C}{m} - \lambda - \alpha = 0, \quad \frac{C}{m} - \mu - \beta &= 0. \end{aligned}$$

Substituting the above equations in the second expression for the Lagrangian, we find that the dual function G is independent of the variables γ, α, β and is given by

$$G(\lambda, \mu) = -\frac{1}{2} \sum_{i,j=1}^m (\lambda_i - \mu_i)(\lambda_j - \mu_j) x_i^\top x_j - \sum_{i=1}^m (\lambda_i - \mu_i) y_i$$

if

$$\begin{aligned} \sum_{i=1}^m \lambda_i - \sum_{i=1}^m \mu_i &= 0 \\ \sum_{i=1}^m \lambda_i + \sum_{i=1}^m \mu_i + \gamma &= C\nu \\ \lambda + \alpha = \frac{C}{m}, \quad \mu + \beta &= \frac{C}{m}, \end{aligned}$$

and $-\infty$ otherwise.

The dual program is obtained by maximizing $G(\alpha, \mu)$ or equivalently by minimizing $-G(\alpha, \mu)$, over $\alpha, \mu \in \mathbb{R}^m_+$. Taking into account the fact that $\alpha, \beta \geq 0$ and $\gamma \geq 0$, we obtain the following dual program:

Dual Program for ν-SV Regression:

$$\begin{aligned}
&\text{minimize} \quad \frac{1}{2}\sum_{i,j=1}^m (\lambda_i - \mu_i)(\lambda_j - \mu_j) x_i^\top x_j + \sum_{i=1}^m (\lambda_i - \mu_i) y_i \\
&\text{subject to} \\
&\qquad \sum_{i=1}^m \lambda_i - \sum_{i=1}^m \mu_i = 0 \\
&\qquad \sum_{i=1}^m \lambda_i + \sum_{i=1}^m \mu_i \leq C\nu \\
&\qquad 0 \leq \lambda_i \leq \frac{C}{m}, \quad 0 \leq \mu_i \leq \frac{C}{m}, \quad i = 1, \ldots, m,
\end{aligned}$$

minimizing over α and μ.

Solving the dual program (for example, using ADMM, see Section 20.3) does not determine b, and for this we use the KKT conditions. The KKT conditions (for the primal program) are

$$\begin{aligned}
\lambda_i(w^\top x_i + b - y_i - \epsilon - \xi_i) &= 0, \quad i = 1, \ldots, m \\
\mu_i(-w^\top x_i - b + y_i - \epsilon - \xi'_i) &= 0, \quad i = 1, \ldots, m \\
\gamma\epsilon &= 0 \\
\alpha_i \xi_i &= 0, \quad i = 1, \ldots, m \\
\beta_i \xi'_i &= 0, \quad i = 1, \ldots, m.
\end{aligned}$$

If $\epsilon > 0$, since the equations

$$\begin{aligned}
w^\top x_i + b - y_i &= \epsilon + \xi_i \\
-w^\top x_i - b + y_i &= \epsilon + \xi'_i
\end{aligned}$$

cannot hold simultaneously, we must have

$$\lambda_i \mu_i = 0, \quad i = 1, \ldots, m. \tag{$\lambda\mu$}$$

From the equations

$$\lambda_i + \alpha_i = \frac{C}{m}, \quad \mu_i + \beta_i = \frac{C}{m}, \quad \alpha_i \xi_i = 0, \quad \beta_i \xi'_i = 0,$$

we get the equations

$$\left(\frac{C}{m} - \lambda_i\right)\xi_i = 0, \quad \left(\frac{C}{m} - \mu_i\right)\xi'_i = 0, \quad i = 1, \ldots, m. \tag{$*$}$$

Suppose we have optimal solution with $\epsilon > 0$. Using the above equations and the fact that $\lambda_i \mu_i = 0$ we obtain the following classification of the points x_i in terms of λ and μ.

(1) $0 < \lambda_i < C/m$. By $(*)$, $\xi_i = 0$, so the equation $w^\top x_i + b - y_i = \epsilon$ holds and x_i is on the blue margin hyperplane $H_{w,b-\epsilon}$. See Figure 20.5.
(2) $0 < \mu_i < C/m$. By $(*)$, $\xi'_i = 0$, so the equation $-w^\top x_i - b + y_i = \epsilon$ holds and x_i is on the red margin hyperplane $H_{w,b+\epsilon}$. See Figure 20.5.
(3) $\lambda_i = C/m$. By $(\lambda\mu)$, $\mu_i = 0$, and by $(*)$, $\xi'_i = 0$. Thus we have

$$\begin{aligned} w^\top x_i + b - y_i &= \epsilon + \xi_i \\ -w^\top x_i - b + y_i &\leq \epsilon. \end{aligned}$$

The second inequality is equivalent to $-\epsilon \leq w^\top x_i + b - y_i$, and since $\epsilon > 0$ and $\xi_i \geq 0$ it is trivially satisfied. If $\xi_i = 0$, then x_i is on the blue margin $H_{w,b-\epsilon}$, else x_i is an error and it lies in the open half-space bounded by the blue margin $H_{w,b-\epsilon}$ and not containing the best fit hyperplane $H_{w,b}$ (it is outside of the ϵ-slab). See Figure 20.5.
(4) $\mu_i = C/m$. By $(\lambda\mu)$, $\lambda_i = 0$, and by $(*)$, $\xi_i = 0$. Thus we have

$$\begin{aligned} w^\top x_i + b - y_i &\leq \epsilon \\ -w^\top x_i - b + y_i &= \epsilon + \xi'_i. \end{aligned}$$

The second equation is equivalent to $w^\top x_i + b - y_i = -\epsilon - \xi'_i$, and since $\epsilon > 0$ and $\xi'_i \geq 0$, the first inequality it is trivially satisfied. If $\xi'_i = 0$, then x_i is on the red margin $H_{w,b+\epsilon}$, else x_i is an error and it lies in the open half-space bounded by the red margin $H_{w,b-\epsilon}$ and not containing the best fit hyperplane $H_{w,b}$ (it is outside of the ϵ-slab). See Figure 20.5.
(5) $\lambda_i = 0$ and $\mu_i = 0$. By $(*)$, $\xi_i = 0$ and $\xi'_i = 0$, so we have

$$\begin{aligned} w^\top x_i + b - y_i &\leq \epsilon \\ -w^\top x_i - b + y_i &\leq \epsilon, \end{aligned}$$

that is

$$-\epsilon \leq w^\top x_i + b - y_i \leq \epsilon.$$

If $w^\top x_i + b - y_i = \epsilon$, then x_i is on the blue margin, and if $w^\top x_i + b - y_i = -\epsilon$, then x_i is on the red margin. If $-\epsilon < w^\top x_i + b - y_i < \epsilon$, then x_i is strictly inside of the ϵ-slab (bounded by the blue margin and the red margin). See Figure 20.6.

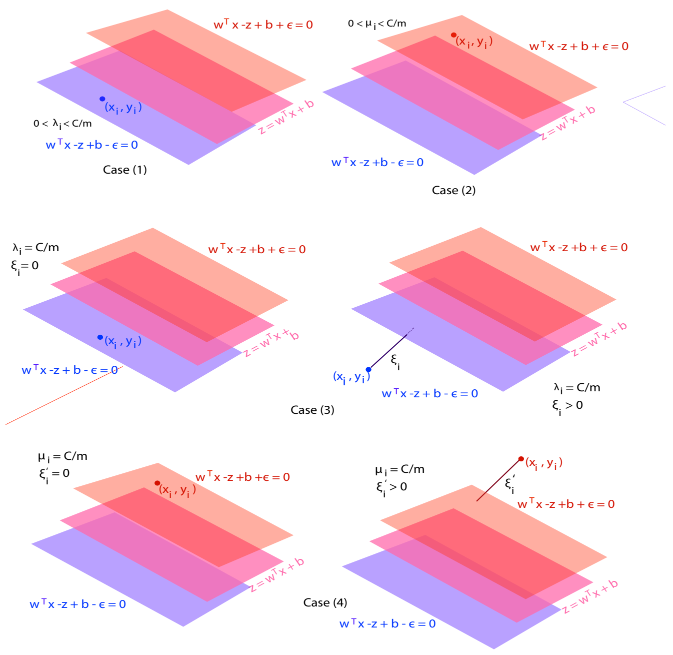


Fig. 20.5 Classifying x_i in terms of nonzero λ and μ.

The above classification shows that the point x_i is an error iff $\lambda_i = C/m$ and $\xi_i > 0$ or or $\mu_i = C/m$ and $\xi'_i > 0$.

As in the case of SVM (see Section 14.6) we define support vectors as follows.

Definition 20.1. A vector x_i such that either $w^\top x_i + b - y_i = \epsilon$ (which implies $\xi_i = 0$) or $-w^\top x_i - b + y_i = \epsilon$ (which implies $\xi'_i = 0$) is called a *support vector*. Support vectors x_i such that $0 < \lambda_i < C/m$ and support vectors x_j such that $0 < \mu_j < C/m$ are *support vectors of type 1*. Support vectors of type 1 play a special role so we denote the sets of indices associated with them by

$$I_\lambda = \{i \in \{1, \ldots, m\} \mid 0 < \lambda_i < C/m\}$$
$$I_\mu = \{j \in \{1, \ldots, m\} \mid 0 < \mu_j < C/m\}.$$

We denote their cardinalities by $numsvl_1 = |I_\lambda|$ and $numsvm_1 = |I_\mu|$. Support vectors x_i such that $\lambda_i = C/m$ and support vectors x_j such that

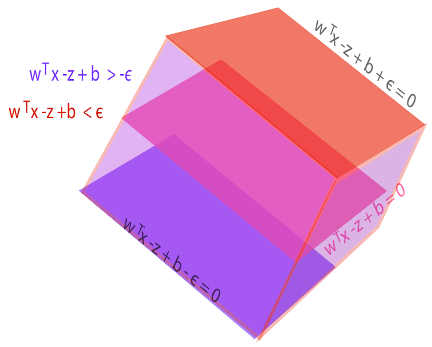


Fig. 20.6 The closed ϵ- tube associated with zero multiplier classification, namely $\lambda_i = 0$ and $\mu_i = 0$.

$\mu_j = C/m$ are *support vectors of type 2*. Support vectors for which $\lambda_i = \mu_i = 0$ are called *exceptional support vectors*.

The following definition also gives a useful classification criterion.

Definition 20.2. A point x_i such that either $\lambda_i = C/m$ or $\mu_i = C/m$ is said to *fail the margin*. The sets of indices associated with the vectors failing the margin are denoted by

$$\begin{aligned} K_\lambda &= \{i \in \{1, \ldots, m\} \mid \lambda_i = C/m\} \\ K_\mu &= \{j \in \{1, \ldots, m\} \mid \mu_j = C/m\}. \end{aligned}$$

We denote their cardinalities by $p_f = |K_\lambda|$ and $q_f = |K_\mu|$.

Vectors u_i such that $\lambda_i > 0$ and vectors v_j such that $\mu_j > 0$ are said to *have margin at most* ϵ. A point x_i such that either $\lambda_i > 0$ or $\mu_i > 0$ is said to have *margin at most* ϵ. The sets of indices associated with these vectors are denoted by

$$\begin{aligned} I_{\lambda>0} &= \{i \in \{1, \ldots, m\} \mid \lambda_i > 0\} \\ I_{\mu>0} &= \{j \in \{1, \ldots, m\} \mid \mu_j > 0\}. \end{aligned}$$

We denote their cardinalities by $p_m = |I_{\lambda>0}|$ and $q_m = |I_{\mu>0}|$.

Points that fail the margin and are not on the boundary of the ϵ-slab lie outside the closed ϵ-slab, so they are errors, also called *outliers*; they correspond to $\xi_i > 0$ or $\xi_i' > 0$.

Observe that we have the equations $I_\lambda \cup K_\lambda = I_{\lambda>0}$ and $I_\mu \cup K_\mu = I_{\mu>0}$, and the inequalities $p_f \leq p_m$ and $q_f \leq q_m$.

We also have the following results showing that p_f, q_f, p_m and q_m have a direct influence on the choice of ν.

Proposition 20.2.

(1) Let p_f be the number of points x_i such that $\lambda_i = C/m$, and let q_f be the number of points x_i such that $\mu_i = C/m$. We have $p_f, q_f \leq (m\nu)/2$.
(2) Let p_m be the number of points x_i such that $\lambda_i > 0$, and let q_m be the number of points x_i such that $\mu_i > 0$. We have $p_m, q_m \geq (m\nu)/2$.
(3) If $p_f \geq 1$ or $q_f \geq 1$, then $\nu \geq 2/m$.

Proof. (1) Recall that for an optimal solution with $w \neq 0$ and $\epsilon > 0$ we have $\gamma = 0$, so we have the equations

$$\sum_{i=1}^m \lambda_i = \frac{C\nu}{2} \quad \text{and} \quad \sum_{j=1}^m \mu_j = \frac{C\nu}{2}.$$

If there are p_f points such that $\lambda_i = C/m$, then

$$\frac{C\nu}{2} = \sum_{i=1}^m \lambda_i \geq p_f \frac{C}{m},$$

so

$$p_f \leq \frac{m\nu}{2}.$$

A similar reasoning applies if there are q_f points such that $\mu_i = C/m$, and we get

$$q_f \leq \frac{m\nu}{2}.$$

(2) If $I_{\lambda>0} = \{i \in \{1, \ldots, m\} \mid \lambda_i > 0\}$ and $p_m = |I_{\lambda>0}|$, then

$$\frac{C\nu}{2} = \sum_{i=1}^m \lambda_i = \sum_{i \in I_{\lambda>0}} \lambda_i,$$

and since $\lambda_i \leq C/m$, we have

$$\frac{C\nu}{2} = \sum_{i \in I_{\lambda>0}} \lambda_i \leq p_m \frac{C}{m},$$

which yields

$$p_m \geq \frac{\nu m}{2}.$$

A similar reasoning applies if $\mu_i > 0$.

(3) This follows immediately from (1). ☐

Proposition 20.2 yields bounds on ν, namely

$$\max\left\{\frac{2p_f}{m}, \frac{2q_f}{m}\right\} \leq \nu \leq \min\left\{\frac{2p_m}{m}, \frac{2q_m}{m}\right\},$$

with $p_f \leq p_m$, $q_f \leq q_m$, $p_f + q_f \leq m$ and $p_m + q_m \leq m$. Also, $p_f = q_f = 0$ means that the ϵ-slab is wide enough so that there are no errors (no points strictly outside the slab).

Observe that a small value of ν keeps p_f and q_f small, which is achieved if the ϵ-slab is wide. A large value of ν allows p_m and q_m to be fairly large, which is achieved if the ϵ-slab is narrow. Thus the smaller ν is, the wider the ϵ-slab is, and the larger ν is, the narrower the ϵ-slab is.

20.2 Existence of Support Vectors

We now consider the issue of the existence of support vectors. We will show that in the generic case, for any optimal solution for which $\epsilon > 0$, there is some support vector on the blue margin *and* some support vector on the red margin. Here generic means that there is an optimal solution for some $\nu < (m-1)/m$.

If the data set (X, y) is well fit by some affine function $f(x) = w^\top x + b$, in the sense that for many pairs (x_i, y_i) we have $y_i = w^\top x_i + b$ and the ℓ^1-error

$$\sum_{i=1}^{m} |w^\top x_i + b - y_i|$$

is small, then an optimal solution may have $\epsilon = 0$. Geometrically, many points (x_i, y_i) belong to the hyperplane $H_{w,b}$. The situation in which $\epsilon = 0$ corresponds to minimizing the ℓ^1-error with a quadratic penalization of w. This is a sort of dual of lasso. The fact that the affine function $f(x) = w^\top x + b$ fits perfectly many points corresponds to the fact that an ℓ^1-minimization tends to encourage sparsity. In this case, if C is chosen too small, it is possible that all points are errors (although "small") and there are no support vectors. But if C is large enough, the solution will be sparse and there will be *many* support vectors on the hyperplane $H_{w,b}$.

Let $E_\lambda = \{i \in \{1, \dots, m\} \mid \xi_i > 0\}$, $E_\mu = \{j \in \{1, \dots, m\} \mid \xi'_j > 0\}$, $p_{sf} = |E_\lambda|$ and $q_{sf} = |E_\mu|$. Obviously, E_λ and E_μ are disjoint.

Given any real numbers u, v, x, y, if $\max\{u, v\} < \min\{x, y\}$, then $u < x$ and $v < y$. This is because $u, v \leq \max\{u, v\} < \min\{x, y\} \leq x, y$.

Proposition 20.3. *If $\nu < (m-1)/m$, then $p_f < \lfloor m/2 \rfloor$ and $q_f < \lfloor m/2 \rfloor$.*

Proof. By Proposition 20.2, $\max\{2p_f/m, 2q_f/m\} \leq \nu$. If m is even, say $m = 2k$, then

$$2p_f/m = 2p_f/(2k) \leq \nu < (m-1)/m = (2k-1)/2k,$$

so $2p_f < 2k - 1$, which implies $p_f < k = \lfloor m/2 \rfloor$. A similar argument shows that $q_f < k = \lfloor m/2 \rfloor$.

If m is odd, say $m = 2k+1$, then

$$2p_f/m = 2p_f/(2k+1) \leq \nu < (m-1)/m = 2k/(2k+1),$$

so $2p_f < 2k$, which implies $p_f < k = \lfloor m/2 \rfloor$. A similar argument shows that $q_f < k = \lfloor m/2 \rfloor$. ☐

Since $p_{sf} \leq p_f$ and $q_{sf} \leq q_f$, we also have $p_{sf} < \lfloor m/2 \rfloor$ and $q_{sf} < \lfloor m/2 \rfloor$. This implies that $\{1, \ldots, m\} - (E_\lambda \cup E_\mu)$ contains at least two elements and there are constraints corresponding to at least two $i \notin (E_\lambda \cup E_\mu)$ (in which case $\xi_i = \xi'_i = 0$), of the form

$$\begin{aligned} w^\top x_i + b - y_i &\leq \epsilon & \quad & i \notin (E_\lambda \cup E_\mu) \\ -w^\top x_i - b + y_i &\leq \epsilon & & i \notin (E_\lambda \cup E_\mu). \end{aligned}$$

If $w^\top x_i + b - y_i = \epsilon$ for some $i \notin (E_\lambda \cup E_\mu)$ and $-w^\top x_j - b + y_j = \epsilon$ for some $j \notin (E_\lambda \cup E_\mu)$ with $i \neq j$, then we have a blue support vector and a red support vector. Otherwise, we show how to modify b and ϵ to obtain an optimal solution with a blue support vector and a red support vector.

Proposition 20.4. *For every optimal solution $(w, b, \epsilon, \xi, \xi')$ with $w \neq 0$ and $\epsilon > 0$, if*

$$\nu < (m-1)/m$$

and if either no x_i is a blue support vector or no x_i is a red support vector, then there is another optimal solution (for the same w) with some i_0 such that $\xi_{i_0} = 0$ and $w^\top x_{i_0} + b - y_{i_0} = \epsilon$, and there is some j_0 such that $\xi'_{j_0} = 0$ and $-w^\top x_{j_0} - b + y_{j_0} = \epsilon$; in other words, some x_{i_0} is a blue support vector and some x_{j_0} is a red support vector (with $i_0 \neq j_0$). If all points (x_i, y_i) that are not errors lie on one of the margin hyperplanes, then there is an optimal solution for which $\epsilon = 0$.

Proof. By Proposition 20.3 if $\nu < (m-1)/m$, then $p_f < \lfloor m/2 \rfloor$ and $q_f < \lfloor m/2 \rfloor$, so the following constraints hold:

$$\begin{aligned} w^\top x_i + b - y_i &= \epsilon + \xi_i & \quad \xi_i &> 0 & \quad & i \in E_\lambda \\ -w^\top x_j - b + y_j &= \epsilon + \xi'_j & \xi'_j &> 0 & & j \in E_\mu \\ w^\top x_i + b - y_i &\leq \epsilon & & & & i \notin (E_\lambda \cup E_\mu) \\ -w^\top x_i - b + y_i &\leq \epsilon & & & & i \notin (E_\lambda \cup E_\mu), \end{aligned}$$

where $|\{1, \ldots, m\} - (E_\lambda \cup E_\mu)| \geq 2$.

If our optimal solution does not have a blue support vector and a red support vector, then either $w^\top x_i + b - y_i < \epsilon$ for all $i \notin (E_\lambda \cup E_\mu)$ or $-w^\top x_i - b + y_i < \epsilon$ for all $i \notin (E_\lambda \cup E_\mu)$.

Case 1. We have

$$\begin{aligned} w^\top x_i + b - y_i &< \epsilon & \qquad & i \notin (E_\lambda \cup E_\mu) \\ -w^\top x_i - b + y_i &\leq \epsilon & \qquad & i \notin (E_\lambda \cup E_\mu). \end{aligned}$$

There are two subcases.

Case 1a. Assume that there is some $j \notin (E_\lambda \cup E_\mu)$ such that $-w^\top x_j - b + y_j = \epsilon$. Our strategy is to decrease ϵ and increase b by a small amount θ in such a way that some inequality $w^\top x_i + b - y_i < \epsilon$ becomes an equation for some $i \notin (E_\lambda \cup E_\mu)$. Geometrically, this amounts to raising the separating hyperplane $H_{w,b}$ and decreasing the width of the slab, keeping the red margin hyperplane unchanged. See Figure 20.7.

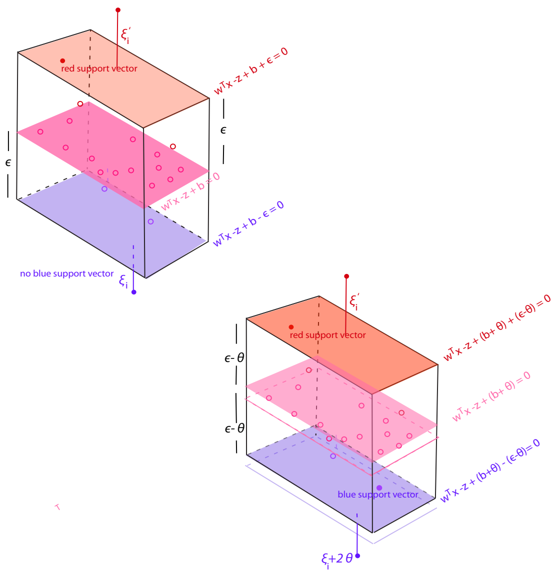


Fig. 20.7 In this illustration points within the ϵ-tube are denoted by open circles. In the original, upper left configuration, there is no blue support vector. By raising the pink separating hyperplane and decreasing the width of the slab, we end up with a blue support vector.

The inequalities imply that

$$-\epsilon \leq w^\top x_i + b - y_i < \epsilon.$$

Let us pick θ such that

$$\theta = (1/2)\min\{\epsilon - w^\top x_i - b + y_i \mid i \notin (E_\lambda \cup E_\mu)\}.$$

Our hypotheses imply that $\theta > 0$, and we have $\theta \leq \epsilon$, because $(1/2)(\epsilon - w^\top x_i - b + y_i) \leq \epsilon$ is equivalent to $\epsilon - w^\top x_i - b + y_i \leq 2\epsilon$ which is equivalent to $-w^\top x_i - b + y_i \leq \epsilon$, which holds for all $i \notin (E_\lambda \cup E_\mu)$ by hypothesis.

We can write

$$\begin{aligned}
w^\top x_i + b + \theta - y_i &= \epsilon - \theta + \xi_i + 2\theta & \xi_i &> 0 & i &\in E_\lambda \\
-w^\top x_j - (b + \theta) + y_j &= \epsilon - \theta + \xi'_j & \xi'_j &> 0 & j &\in E_\mu \\
w^\top x_i + b + \theta - y_i &\leq \epsilon - \theta & & & i &\notin (E_\lambda \cup E_\mu) \\
-w^\top x_i - (b + \theta) + y_i &\leq \epsilon - \theta & & & i &\notin (E_\lambda \cup E_\mu).
\end{aligned}$$

By hypothesis

$$-w^\top x_j - (b + \theta) + y_j = \epsilon - \theta \quad \text{for some } j \notin (E_\lambda \cup E_\mu)$$

and by the choice of θ,

$$w^\top x_i + b + \theta - y_i = \epsilon - \theta \quad \text{for some } i \notin (E_\lambda \cup E_\mu).$$

The value of $C > 0$ is irrelevant in the following argument so we may assume that $C = 1$. The new value of the objective function is

$$\begin{aligned}
\omega(\theta) &= \frac{1}{2}w^\top w + \nu(\epsilon - \theta) + \frac{1}{m}\left(\sum_{i \in E_\lambda}(\xi_i + 2\theta) + \sum_{j \in E_\mu} \xi'_j\right) \\
&= \frac{1}{2}w^\top w + \nu\epsilon + \frac{1}{m}\left(\sum_{i \in E_\lambda}\xi_i + \sum_{j \in E_\mu}\xi'_j\right) - \left(\nu - \frac{2p_{sf}}{m}\right)\theta.
\end{aligned}$$

By Proposition 20.2 we have

$$\max\left\{\frac{2p_f}{m}, \frac{2q_f}{m}\right\} \leq \nu$$

and $p_{sf} \leq p_f$ and $q_{sf} \leq q_f$, which implies that

$$\nu - \frac{2p_{sf}}{m} \geq 0, \tag{$*_1$}$$

and so $\omega(\theta) \leq \omega(0)$. If inequality $(*_1)$ is strict, then this contradicts the optimality of the original solution. Therefore, $\nu = 2p_{sf}/m$, $\omega(\theta) = \omega(0)$ and $(w, b + \theta, \epsilon - \theta, \xi + 2\theta, \xi')$ is an optimal solution such that

$$\begin{aligned}
w^\top x_i + b + \theta - y_i &= \epsilon - \theta \\
-w^\top x_j - (b + \theta) + y_j &= \epsilon - \theta
\end{aligned}$$

for some $i, j \notin (E_\lambda \cup E_\mu)$ with $i \neq j$.

Observe that the exceptional case in which $\theta = \epsilon$ may arise. In this case all points (x_i, y_i) that are not errors (strictly outside the ϵ-slab) are on the red margin hyperplane. This case can only arise if $\nu = 2p_{sf}/m$.

Case 1b. We have $-w^\top x_i - b + y_i < \epsilon$ for all $i \notin (E_\lambda \cup E_\mu)$. Our strategy is to decrease ϵ and increase the errors by a small θ in such a way that some inequality becomes an equation for some $i \notin (E_\lambda \cup E_\mu)$. Geometrically, this corresponds to decreasing the width of the slab, keeping the separating hyperplane unchanged. See Figures 20.8 and 20.9. Then we are reduced to *Case 1a* or *Case 2a*.

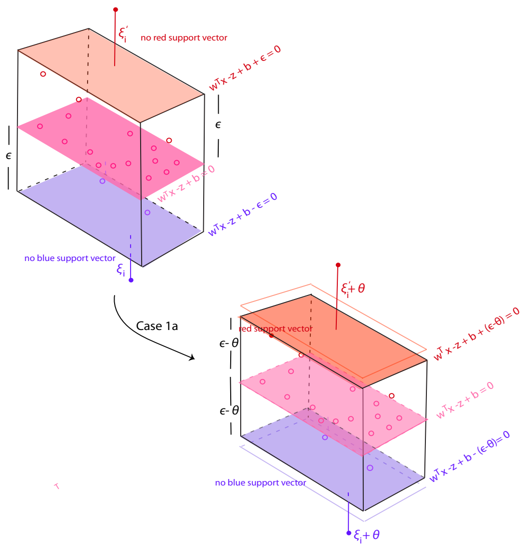


Fig. 20.8 In this illustration points within the ϵ-tube are denoted by open circles. In the original, upper left configuration, there is no blue support vector and no red support vector. By decreasing the width of the slab, we end up with a red support vector and reduce to Case 1a.

We have

$$\begin{aligned}
w^\top x_i + b - y_i &= \epsilon + \xi_i & \xi_i &> 0 & i &\in E_\lambda \\
-w^\top x_j - b + y_j &= \epsilon + \xi_j' & \xi_j' &> 0 & j &\in E_\mu \\
w^\top x_i + b - y_i &< \epsilon & & & i &\notin (E_\lambda \cup E_\mu) \\
-w^\top x_i - b + y_i &< \epsilon & & & i &\notin (E_\lambda \cup E_\mu).
\end{aligned}$$

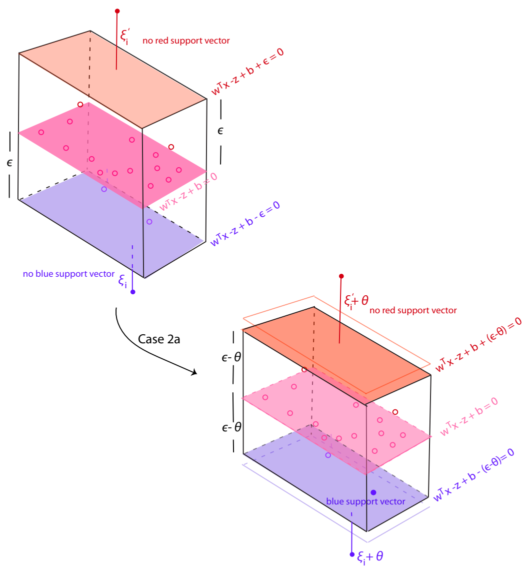


Fig. 20.9 In this illustration points within ϵ-tube are denoted by open circles. In the original, upper left configuration, there is no blue support vector and no red support vector. By decreasing the width of the slab, we end up with a blue support vector and reduce to Case 2a.

Let us pick θ such that

$$\theta = \min\{\epsilon - (w^\top x_i + b - y_i),\ \epsilon + w^\top x_i + b - y_i \mid i \notin (E_\lambda \cup E_\mu)\},$$

Our hypotheses imply that $0 < \theta < \epsilon$. We can write

$$\begin{aligned} w^\top x_i + b - y_i &= \epsilon - \theta + \xi_i + \theta & \xi_i &> 0 & i &\in E_\lambda \\ -w^\top x_j - b + y_j &= \epsilon - \theta + \xi_j' + \theta & \xi_j' &> 0 & j &\in E_\mu \\ w^\top x_i + b - y_i &\leq \epsilon - \theta & & & i &\notin (E_\lambda \cup E_\mu) \\ -w^\top x_i - b + y_i &\leq \epsilon - \theta & & & i &\notin (E_\lambda \cup E_\mu), \end{aligned}$$

and by the choice of θ, either

$$w^\top x_i + b - y_i = \epsilon - \theta \quad \text{for some } i \notin (E_\lambda \cup E_\mu)$$

or

$$-w^\top x_i - b + y_i = \epsilon - \theta \quad \text{for some } i \notin (E_\lambda \cup E_\mu).$$

The new value of the objective function is

$$\begin{aligned}\omega(\theta) &= \frac{1}{2}w^\top w + \nu(\epsilon - \theta) + \frac{1}{m}\biggl(\sum_{i\in E_\lambda}(\xi_i + \theta) + \sum_{j\in E_\mu}(\xi'_j + \theta)\biggr)\\ &= \frac{1}{2}w^\top w + \nu\epsilon + \frac{1}{m}\biggl(\sum_{i\in E_\lambda}\xi_i + \sum_{j\in E_\mu}\xi'_j\biggr) - \biggl(\nu - \frac{p_{sf}+q_{sf}}{m}\biggr)\theta.\end{aligned}$$

Since $\max\{2p_f/m, 2q_f/m\} \leq \nu$ implies that $(p_f+q_f)/m \leq \nu$ and $p_{sf} \leq p_f$, $q_{sf} \leq q_f$, we have

$$\nu - \frac{p_{sf}+q_{sf}}{m} \geq 0, \tag{$*_2$}$$

and so $\omega(\theta) \leq \omega(0)$. If inequality $(*_2)$ is strict, then this contradicts the optimality of the original solution. Therefore, $\nu = (p_{sf}+q_{sf})/m$, $\omega(\theta) = \omega(0)$ and $(w, b, \epsilon - \theta, \xi + \theta, \xi' + \theta)$ is an optimal solution such that either

$$w^\top x_i + b - y_i = \epsilon - \theta \quad \text{for some } i \notin (E_\lambda \cup E_\mu)$$

or

$$-w^\top x_i - b + y_i = \epsilon - \theta \quad \text{for some } i \notin (E_\lambda \cup E_\mu).$$

We are now reduced to *Case 1a* or or *Case 2a*.

Case 2. We have

$$\begin{aligned}w^\top x_i + b - y_i &\leq \epsilon & i &\notin (E_\lambda \cup E_\mu)\\ -w^\top x_i - b + y_i &< \epsilon & i &\notin (E_\lambda \cup E_\mu).\end{aligned}$$

Again there are two subcases.

Case 2a. Assume that there is some $i \notin (E_\lambda \cup E_\mu)$ such that $w^\top x_i + b - y_i = \epsilon$. Our strategy is to decrease ϵ and decrease b by a small amount θ in such a way that some inequality $-w^\top x_j - b + y_j < \epsilon$ becomes an equation for some $j \notin (E_\lambda \cup E_\mu)$. Geometrically, this amounts to lowering the separating hyperplane $H_{w,b}$ and decreasing the width of the slab, keeping the blue margin hyperplane unchanged. See Figure 20.10.

The inequalities imply that

$$-\epsilon < w^\top x_i + b - y_i \leq \epsilon.$$

Let us pick θ such that

$$\theta = (1/2)\min\{\epsilon - (-w^\top x_i - b + y_i) \mid i \notin (E_\lambda \cup E_\mu)\}.$$

Our hypotheses imply that $\theta > 0$, and we have $\theta \leq \epsilon$, because $(1/2)(\epsilon - (-w^\top x_i - b + y_i)) \leq \epsilon$ is equivalent to $\epsilon - (-w^\top x_i - b + y_i) \leq 2\epsilon$ which

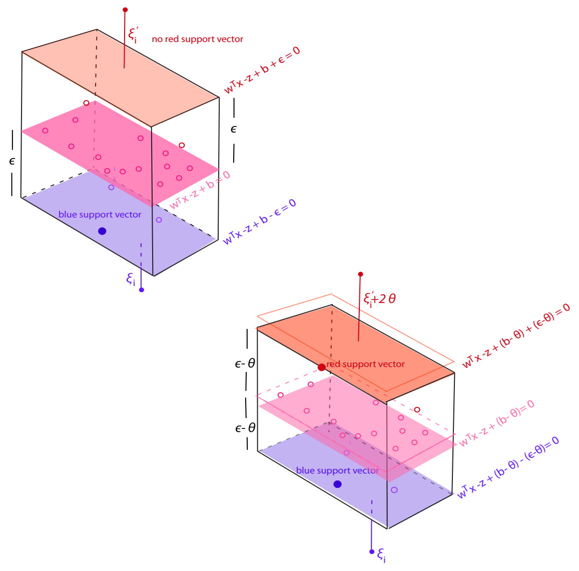


Fig. 20.10 In this illustration points within the ϵ-tube are denoted by open circles. In the original, upper left configuration, there is no red support vector. By lowering the pink separating hyperplane and decreasing the width of the slab, we end up with a red support vector.

is equivalent to $w^\top x_i + b - y_i \leq \epsilon$ which holds for all $i \notin (E_\lambda \cup E_\mu)$ by hypothesis.

We can write

$$\begin{aligned}
w^\top x_i + b - \theta - y_i &= \epsilon - \theta + \xi_i & \xi_i > 0 \quad & i \in E_\lambda \\
-w^\top x_j - (b - \theta) + y_j &= \epsilon - \theta + \xi_j' + 2\theta & \xi_j' > 0 \quad & j \in E_\mu \\
w^\top x_i + b - \theta - y_i &\leq \epsilon - \theta & & i \notin (E_\lambda \cup E_\mu) \\
-w^\top x_i - (b - \theta) + y_i &\leq \epsilon - \theta & & i \notin (E_\lambda \cup E_\mu).
\end{aligned}$$

By hypothesis

$$w^\top x_i + (b - \theta) - y_i = \epsilon - \theta \quad \text{for some } i \notin (E_\lambda \cup E_\mu),$$

and by the choice of θ,

$$-w^\top x_j - (b - \theta) + y_j = \epsilon - \theta \quad \text{for some } j \notin (E_\lambda \cup E_\mu).$$

The new value of the objective function is

$$\begin{aligned}\omega(\theta) &= \frac{1}{2}w^\top w + \nu(\epsilon - \theta) + \frac{1}{m}\Bigg(\sum_{i\in E_\lambda}\xi_i + \sum_{j\in E_\mu}(\xi'_j + 2\theta)\Bigg)\\ &= \frac{1}{2}w^\top w + \nu\epsilon + \frac{1}{m}\Bigg(\sum_{i\in E_\lambda}\xi_i + \sum_{j\in E_\mu}\xi'_j\Bigg) - \bigg(\nu - \frac{2q_{sf}}{m}\bigg)\theta.\end{aligned}$$

The rest of the proof is similar except that $2p_{sf}/m$ is replaced by $2q_{sf}/m$. Observe that the exceptional case in which $\theta = \epsilon$ may arise. In this case all points (x_i, y_i) that are not errors (strictly outside the ϵ-slab) are on the blue margin hyperplane. This case can only arise if $\nu = 2q_{sf}/m$.

Case 2b. We have $w^\top x_i + b - y_i < \epsilon$ for all $i \notin (E_\lambda \cup E_\mu)$. Since we also assumed that $-w^\top x_i - b + y_i < \epsilon$ for all $i \notin (E_\lambda \cup E_\mu)$, Case 2b is identical to Case 1b and we are done. ☐

The proof of Proposition 20.4 reveals that there are three critical values for ν:

$$\frac{2p_{sf}}{m},\ \frac{2q_{sf}}{m},\ \frac{p_{sf}+q_{sf}}{m}.$$

These values can be avoided by requiring the strict inequality

$$\max\left\{\frac{2p_{sf}}{m}, \frac{2q_{sf}}{m}\right\} < \nu.$$

Then the following corollary holds.

Theorem 20.1. *For every optimal solution $(w, b, \epsilon, \xi, \xi')$ with $w \neq 0$ and $\epsilon > 0$, if*

$$\max\left\{\frac{2p_{sf}}{m}, \frac{2q_{sf}}{m}\right\} < \nu < (m-1)/m,$$

then some x_{i_0} is a blue support vector and some x_{j_0} is a red support vector (with $i_0 \neq j_0$).

Proof. We proceed by contradiction. Suppose that for every optimal solution with $w \neq 0$ and $\epsilon > 0$ no x_i is a blue support vector or no x_i is a red support vector. Since $\nu < (m-1)/m$, Proposition 20.4 holds, so there is another optimal solution. But since the critical values of ν are avoided, the proof of Proposition 20.4 shows that the value of the objective function for this new optimal solution is strictly smaller than the original optimal value, a contradiction. ☐

Remark: If an optimal solution has $\epsilon = 0$, then depending on the value of C there may not be any support vectors, or many.

If the primal has an optimal solution with $w \neq 0$ and $\epsilon > 0$, then by $(*_w)$ and since

$$\sum_{i=1}^{m} \lambda_i - \sum_{i=1}^{m} \mu_i = 0 \quad \text{and} \quad \lambda_i \mu_i = 0,$$

there is i_0 such that $\lambda_{i_0} > 0$ and some $j_0 \neq i_0$ such that $\mu_{j_0} > 0$.

Under the mild hypothesis called the **Standard Margin Hypothesis** that there is some i_0 such that $0 < \alpha_{i_0} < \frac{C}{m}$ *and* there is some $j_0 \neq i_0$ such that $0 < \mu_{j_0} < \frac{C}{m}$, in other words there is a blue support vector of type 1 and there is a red support vector of type 1, then by $(*)$ we have $\xi_{i_0} = 0, \xi'_{j_0} = 0$, and we have the two equations

$$\begin{aligned} w^\top x_{i_0} + b - y_{i_0} &= \epsilon \\ -w^\top x_{j_0} - b + y_{j_0} &= \epsilon, \end{aligned}$$

so b and ϵ can be computed. In particular,

$$\begin{aligned} b &= \frac{1}{2}\left(y_{i_0} + y_{j_0} - w^\top(x_{i_0} + x_{j_0})\right) \\ \epsilon &= \frac{1}{2}\left(y_{j_0} - y_{i_0} + w^\top(x_{i_0} - x_{j_0})\right). \end{aligned}$$

The function $f(x) = w^\top x + b$ (often called *regression estimate*) is given by

$$f(x) = \sum_{i=1}^{m} (\mu_i - \lambda_i) x_i^\top x + b.$$

In practice, due to numerical inaccurracy, it is complicated to write a computer program that will select two *distinct indices* as above. It is preferable to compute the list I_λ of indices i such that $0 < \lambda_i < C/m$ and the list I_μ of indices j such that $0 < \mu_j < C/m$. Then it is easy to see that

$$\begin{aligned} b &= \Bigg(\Big(\sum_{i_0 \in I_\lambda} y_{i_0}\Big)/|I_\lambda| + \Big(\sum_{j_0 \in I_\mu} y_{j_0}\Big)/|I_\mu|\Bigg)/2 \\ &\quad - w^\top\Bigg(\Big(\sum_{i_0 \in I_\lambda} x_{i_0}\Big)/|I_\lambda| + \Big(\sum_{j_0 \in I_\mu} x_{j_0}\Big)/|I_\mu|\Bigg)/2 \\ \epsilon &= \Bigg(\Big(\sum_{j_0 \in I_\mu} y_{j_0}\Big)/|I_\mu|\Bigg) - \Big(\sum_{i_0 \in I_\lambda} y_{i_0}\Big)/|I_\lambda\Bigg)/2 \\ &\quad + w^\top\Bigg(\Big(\sum_{i_0 \in I_\lambda} x_{i_0}\Big)/|I_\lambda| - \Big(\sum_{j_0 \in I_\mu} x_{j_0}\Big)/|I_\mu|\Bigg)/2. \end{aligned}$$

These formulae are numerically a lot more stable, but we still have to be cautious to set suitable tolerance factors to decide whether $\lambda_i > 0$ and $\lambda_i < C/m$ (and similarly for μ_i).

The following result gives sufficient conditions for expressing ϵ in terms of a single support vector.

Proposition 20.5. *For every optimal solution $(w, b, \epsilon, \xi, \xi')$ with $w \neq 0$ and $\epsilon > 0$, if*

$$\max\left\{\frac{2p_{sf}}{m}, \frac{2q_{sf}}{m}\right\} < \nu < (m-1)/m,$$

then ϵ and b are determined from a solution (λ, μ) of the dual in terms of a single support vector.

Proof sketch. If we express that the duality gap is zero we obtain the following equation expressing ϵ in terms of b:

$$C\left(\nu - \frac{p_f + q_f}{m}\right)\epsilon = -\begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix} P \begin{pmatrix}\lambda \\ \mu\end{pmatrix} - \begin{pmatrix}y^\top & -y^\top\end{pmatrix}\begin{pmatrix}\lambda \\ \mu\end{pmatrix}$$
$$- \frac{C}{m}\left(w^\top\left(\sum_{i\in K_\lambda} x_i - \sum_{j\in K_\mu} x_j\right) - \sum_{i\in K_\lambda} y_i + \sum_{j\in K_\mu} y_j + (p_f - q_f)b\right).$$

The proof is very similar to the proof of the corresponding formula in Section 20.5. By Theorem 20.1, there is some suppor vector x_i, say

$$w^\top x_{i_0} + b - y_{i_0} = \epsilon \quad \text{or} \quad -w^\top x_{j_0} - b + y_{j_0} = \epsilon.$$

Then we find an equation expressing ϵ in terms of λ, μ and w, provided that $\nu \neq 2p_f/m$ and $\nu \neq 2q_f/m$. The proof is analogous to the proof of Proposition 18.3 and is left as an exercise. $\square$

20.3 Solving ν-Regression Using ADMM

The quadratic functional $F(\lambda, \mu)$ occurring in the dual program given by

$$F(\lambda, \mu) = \frac{1}{2}\sum_{i,j=1}^{m}(\lambda_i - \mu_i)(\lambda_j - \mu_j)x_i^\top x_j + \sum_{i=1}^{m}(\lambda_i - \mu_i)y_i$$

is not of the form $\frac{1}{2}\begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix} P \begin{pmatrix}\lambda \\ \mu\end{pmatrix} + q^\top\begin{pmatrix}\lambda \\ \mu\end{pmatrix}$, but it can be converted in such a form using a trick. First, if we let $\mathbf{K}$ be the $m \times m$ symmetric matrix $\mathbf{K} = XX^\top = (x_i^\top x_j)$, then we have

$$F(\lambda, \mu) = \frac{1}{2}(\lambda^\top - \mu^\top)\mathbf{K}(\lambda - \mu) + y^\top\lambda - y^\top\mu.$$

Consequently, if we define the $2m \times 2m$ symmetric matrix P by

$$P = \begin{pmatrix} XX^\top & -XX^\top \\ -XX^\top & XX^\top \end{pmatrix} = \begin{pmatrix} \mathbf{K} & -\mathbf{K} \\ -\mathbf{K} & \mathbf{K} \end{pmatrix}$$

and the $2m \times 1$ matrix q by

$$q = \begin{pmatrix} y \\ -y \end{pmatrix},$$

it is easy to check that

$$\begin{aligned} F(\lambda, \mu) &= \frac{1}{2} \begin{pmatrix} \lambda^\top & \mu^\top \end{pmatrix} P \begin{pmatrix} \lambda \\ \mu \end{pmatrix} + q^\top \begin{pmatrix} \lambda \\ \mu \end{pmatrix} \\ &= \frac{1}{2}\lambda^\top \mathbf{K}\lambda + \frac{1}{2}\mu^\top \mathbf{K}\mu - \lambda^\top \mathbf{K}\mu + y^\top\lambda - y^\top\mu. \qquad (*_q) \end{aligned}$$

Since

$$\frac{1}{2} \begin{pmatrix} \lambda^\top & \mu^\top \end{pmatrix} P \begin{pmatrix} \lambda \\ \mu \end{pmatrix} = \frac{1}{2}(\lambda^\top - \mu^\top)\mathbf{K}(\lambda - \mu)$$

and the matrix $\mathbf{K} = XX^\top$ is symmetric positive semidefinite, the matrix P is also symmetric positive semidefinite. Thus we are in a position to apply ADMM since the constraints are

$$\begin{gathered} \sum_{i=1}^m \lambda_i - \sum_{i=1}^m \mu_i = 0 \\ \sum_{i=1}^m \lambda_i + \sum_{i=1}^m \mu_i + \gamma = C\nu \\ \lambda + \alpha = \frac{C}{m}, \quad \mu + \beta = \frac{C}{m}, \end{gathered}$$

namely affine. We need to check that the $(2m + 2) \times (4m + 1)$ matrix A corresponding to this system has rank $2m + 2$. Let us clarify this point. The matrix A corresponding to the above equations is

$$A = \begin{pmatrix} \mathbf{1}_m^\top & -\mathbf{1}_m^\top & 0_m^\top & 0_m^\top & 0 \\ \mathbf{1}_m^\top & \mathbf{1}_m^\top & 0_m^\top & 0_m^\top & 1 \\ I_m & 0_{m,m} & I_m & 0_{m,m} & 0_m \\ 0_{m,m} & I_m & 0_{m,m} & I_m & 0_m \end{pmatrix}.$$

For example, for $m = 3$ we have the 8×13 matrix

$$\begin{pmatrix} 1 & 1 & 1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}.$$

We leave it as an exercise to show that A has rank $2m + 2$. Recall that

$$q = \begin{pmatrix} y \\ -y \end{pmatrix}$$

and we also define the vector c (of dimension $2m + 2$) as

$$c = \begin{pmatrix} 0 \\ C\nu \\ \frac{C}{m}\mathbf{1}_{2m} \end{pmatrix}.$$

The constraints are given by the system of affine equations $Ax = c$, where

$$x = \begin{pmatrix} \lambda^\top & \mu^\top & \alpha^\top & \beta^\top & \gamma \end{pmatrix}^\top.$$

Since there are $4m + 1$ Lagrange multipliers $(\lambda, \mu, \alpha, \beta, \gamma)$, we need to pad the $2m \times 2m$ matrix P with zeros to make it into a $(4m+1) \times (4m+1)$ matrix

$$P_a = \begin{pmatrix} P & 0_{2m,2m+1} \\ 0_{2m+1,2m} & 0_{2m+1,2m+1} \end{pmatrix}.$$

Similarly, we pad q with zeros to make it a vector q_a of dimension $4m + 1$,

$$q_a = \begin{pmatrix} q \\ 0_{2m+1} \end{pmatrix}.$$

In order to solve our dual program, we apply ADMM to the quadractic functional

$$\frac{1}{2}x^\top P_a x + q_a^\top x,$$

subject to the constraints

$$Ax = c, \quad x \geq 0,$$

with P_a, q_a, A, b and x, as above.

Since for an optimal solution with $\epsilon > 0$ we must have $\gamma = 0$ (from the KKT conditions), we can solve the dual problem with the following set of constraints only involving the Lagrange multipliers $(\lambda, \mu, \alpha, \beta)$,

$$\begin{aligned}\sum_{i=1}^m \lambda_i - \sum_{i=1}^m \mu_i &= 0\\ \sum_{i=1}^m \lambda_i + \sum_{i=1}^m \mu_i &= C\nu\\ \lambda + \alpha = \frac{C}{m}, \quad \mu + \beta &= \frac{C}{m},\end{aligned}$$

which corresponds to the $(2m+2) \times 4m$ A_2 given by

$$A_2 = \begin{pmatrix} \mathbf{1}_m^\top & -\mathbf{1}_m^\top & 0_m^\top & 0_m^\top \\ \mathbf{1}_m^\top & \mathbf{1}_m^\top & 0_m^\top & 0_m^\top \\ I_m & 0_{m,m} & I_m & 0_{m,m} \\ 0_{m,m} & I_m & 0_{m,m} & I_m \end{pmatrix}.$$

We leave it as an exercise to show that A_2 has rank $2m + 2$. We define the vector c_2 (of dimension $2m + 2$) as

$$c_2 = c = \begin{pmatrix} 0 \\ C\nu \\ \frac{C}{m}\mathbf{1}_{2m} \end{pmatrix}.$$

Since there are $4m$ Lagrange multipliers $(\lambda, \mu, \alpha, \beta)$, we need to pad the $2m \times 2m$ matrix P with zeros to make it into a $4m \times 4m$ matrix

$$P_{2a} = \begin{pmatrix} P & 0_{2m,2m} \\ 0_{2m,2m} & 0_{2m,2m} \end{pmatrix}.$$

Similarly, we pad q with zeros to make it a vector q_{2a} of dimension $4m$,

$$q_{2a} = \begin{pmatrix} q \\ 0_{2m} \end{pmatrix}.$$

We implemented the above methods in `Matlab`; see Appendix B, Section B.4. Choosing $C = m$ is typically a good choice because then the values of λ_i and μ_j are not too small $(C/m = 1)$. If C is chosen too small, we found that numerical instability increases drastically and very poor results are obtained. Increasing C tends to encourage sparsity.

We ran our `Matlab` implementation of the above method on the set of 50 points generated at random by the program shown below with $C = 50$ and various values of ν starting with $\nu = 0.03$:

```
X13 = 15*randn(50,1);
ww13 = 1;
y13 = X13*ww13 + 10*randn(50,1) + 20;
[~,~,~,~,~,~,~,~,w1] = runuregb(rho,0.03,X13,y13,50)
```

Figure 20.11 shows the result of running the program with $\nu = 0.03$. We have $p_f = 0, q_f = 0, p_m = 2$ and $q_m = 1$. There are 47 points strictly inside the slab. The slab is large enough to contain all the data points, so none of them is considered an error.

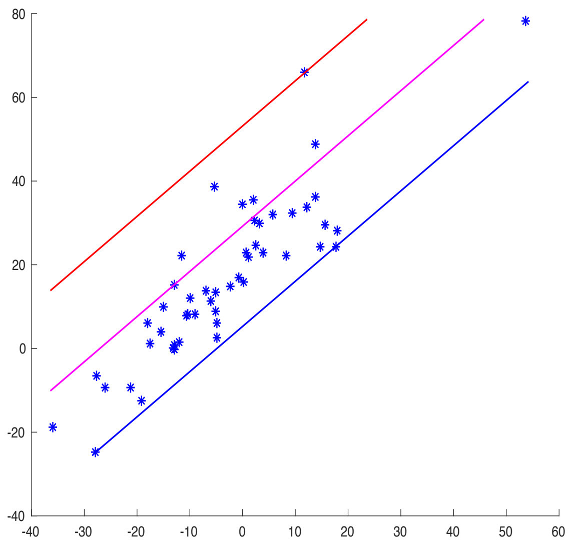


Fig. 20.11 Running ν-SV regression on a set of 50 points; $\nu = 0.03$.

The next value of ν is $\nu = 0.21$, see Figure 20.12. We have $p_f = 4, q_f = 5, p_m = 6$ and $q_m = 6$. There are 38 points strictly inside the slab.

The next value of ν is $\nu = 0.5$, see Figure 20.13. We have $p_f = 12, q_f = 12, p_m = 13$ and $q_m = 14$. There are 23 points strictly inside the slab.

The next value of ν is $\nu = 0.7$, see Figure 20.14. We have $p_f = 17, q_f = 17, p_m = 18$ and $q_m = 19$. There are 13 points strictly inside the slab.

The last value of ν is $\nu = 0.97$, see Figure 20.15. We have $p_f = 23, q_f = 24, p_m = 25$ and $q_m = 25$. There are 0 points strictly inside the slab. The slab is so narrow that it does not contain any of the points x_i in it.

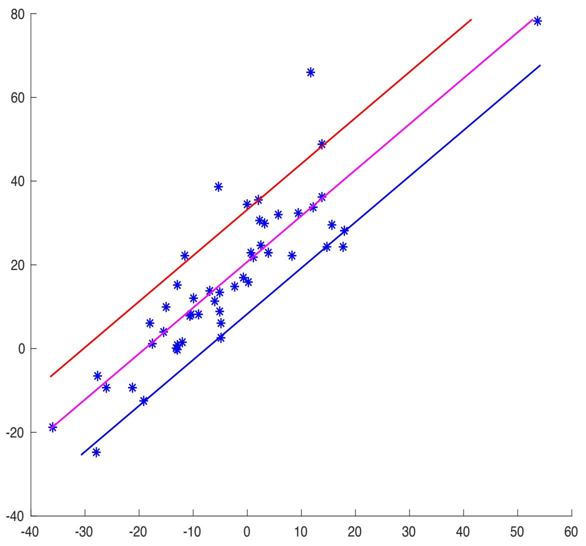


Fig. 20.12 Running ν-SV regression on a set of 50 points; $\nu = 0.21$.

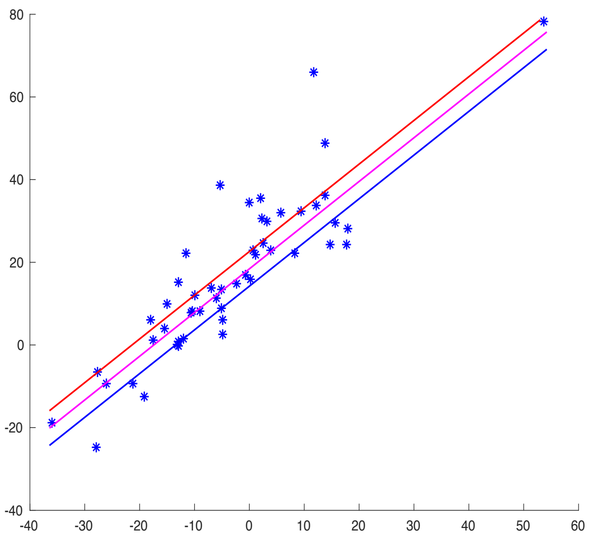


Fig. 20.13 Running ν-SV regression on a set of 50 points; $\nu = 0.5$.

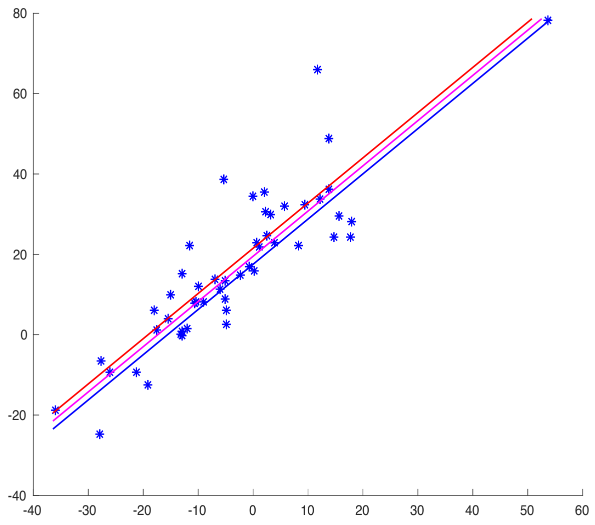


Fig. 20.14 Running ν-SV regression on a set of 50 points; $\nu = 0.7$.

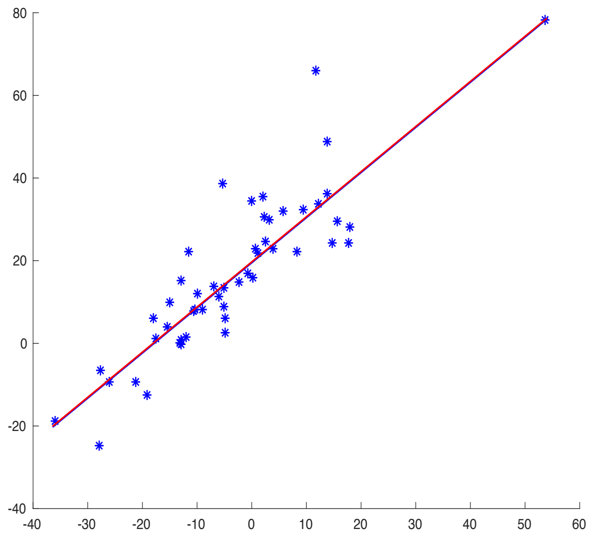


Fig. 20.15 Running ν-SV regression on a set of 50 points; $\nu = 0.97$.

Running the program with any value $\nu > 0.97$ yields $\epsilon = 0$.

20.4 Kernel ν-SV Regression

Since the formulae for w, b, and $f(x)$,

$$\begin{aligned} w &= \sum_{i=1}^{m} (\mu_i - \lambda_i) x_i \\ b &= \frac{1}{2}\left(y_{i_0} + y_{j_0} - w^\top (x_{i_0} + x_{j_0})\right) \\ f(x) &= \sum_{i=1}^{m} (\mu_i - \lambda_i) x_i^\top x + b, \end{aligned}$$

only involve inner products among the data points x_i and x, and since the objective function $-G(\alpha, \mu)$ of the dual program also only involves inner products among the data points x_i, we can kernelize the ν-SV regression method.

As in the previous section, we assume that our data points $\{x_1, \ldots, x_m\}$ belong to a set $\mathcal{X}$ and we pretend that we have feature space $(F, \langle -, - \rangle)$ and a feature embedding map $\varphi \colon \mathcal{X} \to F$, but we only have access to the kernel function $\kappa(x_i, x_j) = \langle \varphi(x_i), \varphi(x_j) \rangle$. We wish to perform ν-SV regression in the feature space F on the data set $\{(\varphi(x_1), y_1), \ldots, (\varphi(x_m), y_m)\}$. Going over the previous computation, we see that the primal program is given by

Program kernel ν-SV Regression:

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2}\langle w, w \rangle + C\left(\nu\epsilon + \frac{1}{m}\sum_{i=1}^{m} (\xi_i + \xi_i')\right) \\ &\text{subject to} \\ &\qquad \langle w, \varphi(x_i) \rangle + b - y_i \le \epsilon + \xi_i, \quad \xi_i \ge 0 \qquad i = 1, \ldots, m \\ &\qquad -\langle w, \varphi(x_i) \rangle - b + y_i \le \epsilon + \xi_i', \quad \xi_i' \ge 0 \qquad i = 1, \ldots, m \\ &\qquad \epsilon \ge 0, \end{aligned}$$

minimizing over the variables w, ϵ, b, ξ, and ξ'.

The Lagrangian is given by

$$\begin{aligned} L(w, b, \lambda, \mu, \gamma, \xi, \xi', \epsilon, \alpha, \beta) &= \frac{1}{2}\langle w, w\rangle + \left\langle w, \sum_{i=1}^m (\lambda_i - \mu_i)\varphi(x_i)\right\rangle \\ &+ \epsilon\left(C\nu - \gamma - \sum_{i=1}^m (\lambda_i + \mu_i)\right) + \sum_{i=1}^m \xi_i\left(\frac{C}{m} - \lambda_i - \alpha_i\right) \\ &+ \sum_{i=1}^m \xi'_i\left(\frac{C}{m} - \mu_i - \beta_i\right) + b\left(\sum_{i=1}^m (\lambda_i - \mu_i)\right) - \sum_{i=1}^m (\lambda_i - \mu_i)y_i. \end{aligned}$$

Setting the gradient $\nabla L_{w,\epsilon,b,\xi,\xi'}$ of the Lagrangian to zero, we also obtain the equations

$$w = \sum_{i=1}^m (\mu_i - \lambda_i)\varphi(x_i), \qquad (*_w)$$

$$\begin{aligned} \sum_{i=1}^m \lambda_i - \sum_{i=1}^m \mu_i &= 0 \\ \sum_{i=1}^m \lambda_i + \sum_{i=1}^m \mu_i + \gamma &= C\nu \\ \lambda + \alpha = \frac{C}{m}, \quad \mu + \beta &= \frac{C}{m}. \end{aligned}$$

Using the above equations, we find that the dual function G is independent of the variables β, α, β, and we obtain the following dual program:

Dual Program kernel ν-SV Regression:

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2}\sum_{i,j=1}^m (\lambda_i - \mu_i)(\lambda_j - \mu_j)\kappa(x_i, x_j) + \sum_{i=1}^m (\lambda_i - \mu_i)y_i \\ &\text{subject to} \\ &\qquad \sum_{i=1}^m \lambda_i - \sum_{i=1}^m \mu_i = 0 \\ &\qquad \sum_{i=1}^m \lambda_i + \sum_{i=1}^m \mu_i \leq C\nu \\ &\qquad 0 \leq \lambda_i \leq \frac{C}{m}, \quad 0 \leq \mu_i \leq \frac{C}{m}, \quad i = 1, \ldots, m, \end{aligned}$$

minimizing over α and μ.

Everything we said before also applies to the kernel ν-SV regression method, except that x_i is replaced by $\varphi(x_i)$ and that the inner product $\langle -, - \rangle$ must be used, and we have the formulae

$$\begin{aligned}
w &= \sum_{i=1}^{m} (\mu_i - \lambda_i)\varphi(x_i) \\
b &= \frac{1}{2}\left(y_{i_0} + y_{j_0} - \sum_{i=1}^{m} (\mu_i - \lambda_i)(\kappa(x_i, x_{i_0}) + \kappa(x_i, x_{j_0})) \right) \\
f(x) &= \sum_{i=1}^{m} (\mu_i - \lambda_i)\kappa(x_i, x) + b,
\end{aligned}$$

expressions that only involve κ.

Remark: There is a variant of ν-SV regression obtained by setting $\nu = 0$ and holding $\epsilon > 0$ fixed. This method is called *ϵ-SV regression* or *(linear) ϵ-insensitive SV regression*. The corresponding optimization program is

Program ϵ-SV Regression:

$$\begin{aligned}
&\text{minimize} \quad \frac{1}{2} w^\top w + \frac{C}{m} \sum_{i=1}^{m} (\xi_i + \xi_i') \\
&\text{subject to} \\
&\qquad w^\top x_i + b - y_i \leq \epsilon + \xi_i, \quad \xi_i \geq 0 \quad i = 1, \ldots, m \\
&\qquad - w^\top x_i - b + y_i \leq \epsilon + \xi_i', \quad \xi_i' \geq 0 \quad i = 1, \ldots, m,
\end{aligned}$$

minimizing over the variables w, b, ξ, and ξ', holding ϵ fixed.

It is easy to see that the dual program is

Dual Program ϵ-SV Regression:

$$\begin{aligned}
&\text{minimize} \quad \frac{1}{2} \sum_{i,j=1}^{m} (\lambda_i - \mu_i)(\lambda_j - \mu_j) x_i^\top x_j + \sum_{i=1}^{m} (\lambda_i - \mu_i) y_i + \epsilon \sum_{i=1}^{m} (\lambda_i + \mu_i) \\
&\text{subject to} \\
&\qquad \sum_{i=1}^{m} \lambda_i - \sum_{i=1}^{m} \mu_i = 0 \\
&\qquad 0 \leq \lambda_i \leq \frac{C}{m}, \quad 0 \leq \mu_i \leq \frac{C}{m}, \quad i = 1, \ldots, m,
\end{aligned}$$

minimizing over α and μ.

The constraint

$$\sum_{i=1}^m \lambda_i + \sum_{i=1}^m \mu_i \leq C\nu$$

is gone but the extra term $\epsilon \sum_{i=1}^m (\lambda_i + \mu_i)$ has been added to the dual function, to prevent λ_i and μ_i from blowing up.

There is an obvious kernelized version of ϵ-SV regression. It is easy to show that ν-SV regression subsumes ϵ-SV regression, in the sense that if ν-SV regression succeeds and yields $w, b, \epsilon > 0$, then ϵ-SV regression with the same C and the same value of ϵ also succeeds and returns the same pair (w, b). For more details on these methods, see Schölkopf, Smola, Williamson, and Bartlett [Schölkopf *et al.* (2000)].

Remark: The linear penalty function $\sum_{i=1}^m (\xi_i + \xi'_i)$ can be replaced by the quadratic penalty function $\sum_{i=1}^m (\xi_i^2 + \xi_i'^2)$; see Shawe–Taylor and Christianini [Shawe-Taylor and Cristianini (2004)] (Chapter 7). In this case, it is easy to see that for an optimal solution we must have $\xi_i \geq 0$ and $\xi'_i \geq 0$, so we may omit the constraints $\xi_i \geq 0$ and $\xi'_i \geq 0$. We must also have $\gamma = 0$ so we omit the variable γ as well. It can be shown that $\xi = (m/2C)\lambda$ and $\xi' = (m/2C)\mu$. This problem is very similar to the Soft Margin SVM (SVM$_{s4}$) discussed in Section 18.13.

20.5 ν-Regression Version 2; Penalizing b

Yet another variant of ν-SV regression is to add the term $\frac{1}{2}b^2$ to the objective function. We will see that solving the dual not only determines w but also b and ϵ (provided a mild condition on ν). We wish to solve the following program:

Program ν-SV Regression Version 2

$$\text{minimize} \quad \frac{1}{2}w^\top w + \frac{1}{2}b^2 + C\left(\nu\epsilon + \frac{1}{m}\sum_{i=1}^m (\xi_i + \xi'_i)\right)$$

subject to

$$\begin{aligned} w^\top x_i + b - y_i &\leq \epsilon + \xi_i, & \xi_i &\geq 0 & i &= 1, \ldots, m \\ -\, w^\top x_i - b + y_i &\leq \epsilon + \xi'_i, & \xi'_i &\geq 0 & i &= 1, \ldots, m, \end{aligned}$$

minimizing over the variables w, b, ϵ, ξ, and ξ'. The constraint $\epsilon \geq 0$ is omitted since the problem has no solution if $\epsilon < 0$.

We leave it as an exercise to show that the new Lagrangian is

$$\begin{aligned}L(w, b, \lambda, \mu, \xi, \xi', \epsilon, \alpha, \beta) &= \frac{1}{2}w^\top w + w^\top \left(\sum_{i=1}^m (\lambda_i - \mu_i)x_i\right)\\ &+ \epsilon \left(C\nu - \sum_{i=1}^m (\lambda_i + \mu_i)\right) + \sum_{i=1}^m \xi_i \left(\frac{C}{m} - \lambda_i - \alpha_i\right)\\ &+ \sum_{i=1}^m \xi'_i \left(\frac{C}{m} - \mu_i - \beta_i\right) + \frac{1}{2}b^2 + b\left(\sum_{i=1}^m (\lambda_i - \mu_i)\right) - \sum_{i=1}^m (\lambda_i - \mu_i)y_i.\end{aligned}$$

If we set the Laplacian $\nabla L_{w,\epsilon,b,\xi,\xi'}$ to zero we obtain the equations

$$\begin{aligned} w &= \sum_{i=1}^m (\mu_i - \lambda_i)x_i = X^\top(\mu - \lambda) \qquad (*_w)\\ C\nu - \sum_{i=1}^m (\lambda_i + \mu_i) &= 0\\ b + \sum_{i=1}^m (\lambda_i - \mu_i) &= 0\\ \frac{C}{m} - \lambda - \alpha = 0, &\quad \frac{C}{m} - \mu - \beta = 0.\end{aligned}$$

We obtain the new equation

$$b = -\sum_{i=1}^m (\lambda_i - \mu_i) = -(\mathbf{1}_m^\top \lambda - \mathbf{1}_m^\top \mu) \qquad (*_b)$$

determining b, which replaces the equation

$$\sum_{i=1}^m \lambda_i - \sum_{i=1}^m \mu_i = 0.$$

Plugging back w from $(*_w)$ and b from $(*_b)$ into the Lagrangian we get

$$\begin{aligned} G(\lambda, \mu, \alpha, \beta) &= -\frac{1}{2}\begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix} P \begin{pmatrix}\lambda\\ \mu\end{pmatrix} - q^\top \begin{pmatrix}\lambda\\ \mu\end{pmatrix} + \frac{1}{2}b^2 - b^2\\ &= -\frac{1}{2}\begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix} P \begin{pmatrix}\lambda\\ \mu\end{pmatrix} - q^\top \begin{pmatrix}\lambda\\ \mu\end{pmatrix} - \frac{1}{2}b^2\\ &= -\frac{1}{2}\begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix} \left(P + \begin{pmatrix}\mathbf{1}_m\mathbf{1}_m^\top & -\mathbf{1}_m\mathbf{1}_m^\top\\ -\mathbf{1}_m\mathbf{1}_m^\top & \mathbf{1}_m\mathbf{1}_m^\top\end{pmatrix}\right) \begin{pmatrix}\lambda\\ \mu\end{pmatrix} - q^\top \begin{pmatrix}\lambda\\ \mu\end{pmatrix},\end{aligned}$$

with

$$P = \begin{pmatrix} XX^\top & -XX^\top\\ -XX^\top & XX^\top\end{pmatrix} = \begin{pmatrix}\mathbf{K} & -\mathbf{K}\\ -\mathbf{K} & \mathbf{K}\end{pmatrix}$$

and

$$q = \begin{pmatrix} y \\ -y \end{pmatrix}.$$

The new dual program is

Dual Program ν-SV Regression Version 2

$$\text{minimize} \quad \frac{1}{2}\begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix}\left(P + \begin{pmatrix} \mathbf{1}_m\mathbf{1}_m^\top & -\mathbf{1}_m\mathbf{1}_m^\top \\ -\mathbf{1}_m\mathbf{1}_m^\top & \mathbf{1}_m\mathbf{1}_m^\top \end{pmatrix}\right)\begin{pmatrix}\lambda \\ \mu\end{pmatrix} + q^\top\begin{pmatrix}\lambda \\ \mu\end{pmatrix}$$

subject to

$$\sum_{i=1}^m \lambda_i + \sum_{i=1}^m \mu_i = C\nu$$
$$0 \leq \lambda_i \leq \frac{C}{m}, \quad 0 \leq \mu_i \leq \frac{C}{m}, \quad i = 1, \ldots, m.$$

Definition 20.1 and Definition 20.2 are unchanged. We have the following version of Proposition 20.2 showing that p_f, q_f, p_m an q_m have direct influence on the choice of ν.

Proposition 20.6.

(1) Let p_f be the number of points x_i such that $\lambda_i = C/m$, and let q_f be the number of points x_i such that $\mu_i = C/m$. We have $p_f + q_f \leq m\nu$.
(2) Let p_m be the number of points x_i such that $\lambda_i > 0$, and let q_m be the number of points x_i such that $\mu_i > 0$. We have $p_m + q_m \geq m\nu$.
(3) If $p_f \geq 1$ or $q_f \geq 1$, then $\nu \geq 1/m$.

Proof. (1) Let K_λ and K_μ be the sets of indices corresponding to points failing the margin,

$$K_\lambda = \{i \in \{1, \ldots, m\} \mid \lambda_i = C/m\}$$
$$K_\mu = \{i \in \{1, \ldots, m\} \mid \mu_i = C/m\}.$$

By definition $p_f = |K_\lambda|, q_f = |K_\mu|$. Since the equation

$$\sum_{i=1}^m \lambda_i + \sum_{j=1}^m \mu_j = C\nu$$

holds, by definition of K_λ and K_μ we have

$$(p_f + q_f)\frac{C}{m} = \sum_{i \in K_\lambda} \lambda_i + \sum_{j \in K_\mu} \mu_j \leq \sum_{i=1}^m \lambda_i + \sum_{j=1}^m \mu_j = C\nu,$$

which implies that

$$p_f + q_f \le m\nu.$$

(2) Let $I_{\lambda>0}$ and $I_{\mu>0}$ be the sets of indices

$$I_{\lambda>0} = \{i \in \{1, \dots, m\} \mid \lambda_i > 0\}$$
$$I_{\mu>0} = \{i \in \{1, \dots, m\} \mid \mu_i > 0\}.$$

By definition $p_m = |I_{\lambda>0}|, q_m = |I_{\mu>0}|$. We have

$$\sum_{i=1}^m \lambda_i + \sum_{j=1}^m \mu_j = \sum_{i\in I_{\lambda>0}} \lambda_i + \sum_{j\in I_{\mu>0}} \mu_j = C\nu.$$

Since $\lambda_i \le C/m$ and $\mu_j \le C/m$, we obtain

$$C\nu \le (p_m + q_m)\frac{C}{m},$$

that is, $p_m + q_m \ge m\nu$.

(3) follows immediately from (1). $\square$

Proposition 20.6 yields the following bounds on ν:

$$\frac{p_f + q_f}{m} \le \nu \le \frac{p_m + q_m}{m}.$$

Again, the smaller ν is, the wider the ϵ-slab is, and the larger ν is, the narrower the ϵ-slab is.

Remark: It can be shown that for any optimal solution with $w \neq 0$ and $\epsilon > 0$, if the inequalities $(p_f + q_f)/m < \nu < 1$ hold, then some point x_i is a support vector. The proof is essentially Case 1b in the proof of Proposition 20.4. We leave the details as an exercise.

The new dual program is solved using ADMM. The $(2m + 1) \times 4m$ matrix A_3 corresponding to the equational constraints

$$\sum_{i=1}^m \lambda_i + \sum_{i=1}^m \mu_i = C\nu$$
$$\lambda + \alpha = \frac{C}{m}, \quad \mu + \beta = \frac{C}{m},$$

is given by

$$A_3 = \begin{pmatrix} \mathbf{1}_m^\top & \mathbf{1}_m^\top & 0_m^\top & 0_m^\top \\ I_m & 0_{m,m} & I_m & 0_{m,m} \\ 0_{m,m} & I_m & 0_{m,m} & I_m \end{pmatrix}.$$

We leave it as an exercise to show that A_3 has rank $2m+1$. We define the vector c_3 (of dimension $2m+1$) as

$$c_3 = \begin{pmatrix} C\nu \\ \frac{C}{m}\mathbf{1}_{2m} \end{pmatrix}.$$

Since there are $4m$ Lagrange multipliers $(\lambda, \mu, \alpha, \beta)$, we need to pad the $2m \times 2m$ matrix $P_3 = P + \begin{pmatrix} \mathbf{1}_m\mathbf{1}_m^\top & -\mathbf{1}_m\mathbf{1}_m^\top \\ -\mathbf{1}_m\mathbf{1}_m^\top & \mathbf{1}_m\mathbf{1}_m^\top \end{pmatrix}$ with zeros to make it into a $4m \times 4m$ matrix

$$P_{3a} = \begin{pmatrix} P_3 & 0_{2m,2m} \\ 0_{2m,2m} & 0_{2m,2m} \end{pmatrix}.$$

Similarly, we pad q with zeros to make it a vector q_{3a} of dimension $4m$,

$$q_{3a} = \begin{pmatrix} q \\ 0_{2m} \end{pmatrix}.$$

It remains to compute ϵ. Ther are two methods to do this.

The first method assumes the **Standard Margin Hypothesis**, which is that there is some i_0 such that $0 < \lambda_{i_0} < C/m$ *or* there is some j_0 such that $0 < \mu_{j_0} < C/m$; in other words, there is some support vector of type 1. By the complementary slackness conditions, $\xi_{i_0} = 0$ or $\xi'_{j_0} = 0$, so we have either $w^\top x_{i_0} + b - y_{i_0} = \epsilon$ or $-w^\top x_{j_0} - b + y_{j_0} = \epsilon$, which determines ϵ.

Due to numerical instability, when writing a computer program it is preferable to compute the lists of indices I_λ and I_μ given by

$$\begin{aligned} I_\lambda &= \{i \in \{1, \ldots, m\} \mid 0 < \lambda_i < C/m\} \\ I_\mu &= \{j \in \{1, \ldots, m\} \mid 0 < \mu_j < C/m\}. \end{aligned}$$

Then it is easy to see that we can compute ϵ using the following averaging formulae: if $I_\lambda \neq \emptyset$, then

$$\epsilon = w^\top \Big(\sum_{i \in I_\lambda} x_i\Big)/|I_\lambda| + b - \Big(\sum_{i \in I_\lambda} y_i\Big)/|I_\lambda|,$$

and if $I_\mu \neq \emptyset$, then

$$\epsilon = -w^\top \Big(\sum_{j \in I_\mu} x_j\Big)/|I_\mu| - b + \Big(\sum_{i \in I_\mu} y_i\Big)/|I_\mu|.$$

The second method uses duality. Under a mild condition, expressing that the duality gap is zero, we can determine ϵ in terms of λ, μ and b. This is because points x_i that fail the margin, which means that $\lambda_i = C/m$

or $\mu_i = C/m$, are the only points for which $\xi_i > 0$ or $\xi'_i > 0$. But in this case we have an active constraint

$$w^\top x_i + b - y_i = \epsilon + \xi_i \tag{$*_\xi$}$$

or

$$-w^\top x_i - b + y_i = \epsilon + \xi'_i, \tag{$*_{\xi'}$}$$

so ξ_i and ξ'_i can be expressed in terms of w and b. Since the duality gap is zero for an optimal solution, the optimal value of the primal is equal to the optimal value of the dual. Using the fact that

$$\begin{aligned} w &= X^\top(\mu - \lambda) \\ b &= -(\mathbf{1}_m^\top \lambda - \mathbf{1}_m^\top \mu) = \begin{pmatrix} \lambda^\top & \mu^\top \end{pmatrix} \begin{pmatrix} -\mathbf{1}_m \\ \mathbf{1}_m \end{pmatrix} \end{aligned}$$

we obtain an expression for the optimal value of the primal. First we have

$$\begin{aligned} \frac{1}{2} w^\top w + \frac{1}{2} b^2 &= \frac{1}{2}(\lambda^\top - \mu^\top) X X^\top (\lambda - \mu) \\ &\quad + \frac{1}{2} \begin{pmatrix} \lambda^\top & \mu^\top \end{pmatrix} \begin{pmatrix} \mathbf{1}_m \mathbf{1}_m^\top & -\mathbf{1}_m \mathbf{1}_m^\top \\ -\mathbf{1}_m \mathbf{1}_m^\top & \mathbf{1}_m \mathbf{1}_m^\top \end{pmatrix} \begin{pmatrix} \lambda \\ \mu \end{pmatrix} \\ &= \frac{1}{2} \begin{pmatrix} \lambda^\top & \mu^\top \end{pmatrix} \left(P + \begin{pmatrix} \mathbf{1}_m \mathbf{1}_m^\top & -\mathbf{1}_m \mathbf{1}_m^\top \\ -\mathbf{1}_m \mathbf{1}_m^\top & \mathbf{1}_m \mathbf{1}_m^\top \end{pmatrix} \right) \begin{pmatrix} \lambda \\ \mu \end{pmatrix}, \end{aligned}$$

with

$$P = \begin{pmatrix} XX^\top & -XX^\top \\ -XX^\top & XX^\top \end{pmatrix}.$$

Let K_λ and K_μ be the sets of indices corresponding to points failing the margin,

$$\begin{aligned} K_\lambda &= \{i \in \{1, \ldots, m\} \mid \lambda_i = C/m\} \\ K_\mu &= \{i \in \{1, \ldots, m\} \mid \mu_i = C/m\}. \end{aligned}$$

Because $\lambda_i \mu_i = 0$, the sets K_λ and K_μ are disjoint. Observe that from Definition 20.2 we have $p_f = |K_\lambda|$ and $q_f = |K_\mu|$. Then by $(*_\xi)$ and $(*_{\xi'})$, we have

$$\begin{aligned} \sum_{i=1}^m (\xi_i + \xi'_i) &= \sum_{i \in K_\lambda} \xi_i + \sum_{j \in K_\mu} \xi'_j \\ &= \sum_{i \in K_\lambda} (w^\top x_i + b - y_i - \epsilon) + \sum_{j \in K_\mu} (-w^\top x_j - b + y_j - \epsilon) \\ &= w^\top \Bigg(\sum_{i \in K_\lambda} x_i - \sum_{j \in K_\mu} x_j \Bigg) - \sum_{i \in K_\lambda} y_i + \sum_{j \in K_\mu} y_j \\ &\quad + (p_f - q_f) b - (p_f + q_f)\epsilon. \end{aligned}$$

The optimal value of the dual is given by

$$-\frac{1}{2}\begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix}\left(P + \begin{pmatrix}\mathbf{1}_m\mathbf{1}_m^\top & -\mathbf{1}_m\mathbf{1}_m^\top \\ -\mathbf{1}_m\mathbf{1}_m^\top & \mathbf{1}_m\mathbf{1}_m^\top\end{pmatrix}\right)\begin{pmatrix}\lambda \\ \mu\end{pmatrix} - q^\top\begin{pmatrix}\lambda \\ \mu\end{pmatrix},$$

with

$$q = \begin{pmatrix} y \\ -y\end{pmatrix}.$$

Expressing that the duality gap is zero we have

$$\begin{aligned}&\frac{1}{2}w^\top w + \frac{1}{2}b^2 + C\nu\epsilon + \frac{C}{m}\sum_{i=1}^m(\xi_i + \xi_i') \\ &\quad = -\frac{1}{2}\begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix}\left(P + \begin{pmatrix}\mathbf{1}_m\mathbf{1}_m^\top & -\mathbf{1}_m\mathbf{1}_m^\top \\ -\mathbf{1}_m\mathbf{1}_m^\top & \mathbf{1}_m\mathbf{1}_m^\top\end{pmatrix}\right)\begin{pmatrix}\lambda \\ \mu\end{pmatrix} - q^\top\begin{pmatrix}\lambda \\ \mu\end{pmatrix},\end{aligned}$$

that is,

$$\begin{aligned}&\frac{1}{2}\begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix}\left(P + \begin{pmatrix}\mathbf{1}_m\mathbf{1}_m^\top & -\mathbf{1}_m\mathbf{1}_m^\top \\ -\mathbf{1}_m\mathbf{1}_m^\top & \mathbf{1}_m\mathbf{1}_m^\top\end{pmatrix}\right)\begin{pmatrix}\lambda \\ \mu\end{pmatrix} + C\nu\epsilon \\ &\quad + \frac{C}{m}\Bigg(w^\top\Bigg(\sum_{i\in K_\lambda} x_i - \sum_{j\in K_\mu} x_j\Bigg) - \sum_{i\in K_\lambda} y_i + \sum_{j\in K_\mu} y_j \\ &\qquad\qquad + (p_f - q_f)b - (p_f + q_f)\epsilon\Bigg) \\ &\quad = -\frac{1}{2}\begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix}\left(P + \begin{pmatrix}\mathbf{1}_m\mathbf{1}_m^\top & -\mathbf{1}_m\mathbf{1}_m^\top \\ -\mathbf{1}_m\mathbf{1}_m^\top & \mathbf{1}_m\mathbf{1}_m^\top\end{pmatrix}\right)\begin{pmatrix}\lambda \\ \mu\end{pmatrix} - q^\top\begin{pmatrix}\lambda \\ \mu\end{pmatrix}.\end{aligned}$$

Solving for ϵ we get

$$\begin{aligned}C\left(\nu - \frac{p_f + q_f}{m}\right)\epsilon &= -\begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix}\left(P + \begin{pmatrix}\mathbf{1}_m\mathbf{1}_m^\top & -\mathbf{1}_m\mathbf{1}_m^\top \\ -\mathbf{1}_m\mathbf{1}_m^\top & \mathbf{1}_m\mathbf{1}_m^\top\end{pmatrix}\right)\begin{pmatrix}\lambda \\ \mu\end{pmatrix} \\ &\quad - \begin{pmatrix}y^\top & -y^\top\end{pmatrix}\begin{pmatrix}\lambda \\ \mu\end{pmatrix} - \frac{C}{m}\Bigg(w^\top\Bigg(\sum_{i\in K_\lambda} x_i - \sum_{j\in K_\mu} x_j\Bigg) \\ &\quad - \sum_{i\in K_\lambda} y_i + \sum_{j\in K_\mu} y_j + (p_f - q_f)b\Bigg),\end{aligned}$$

so we get

$$\begin{aligned}&(m\nu - p_f - q_f)\epsilon = \\ &-\Bigg(\frac{m}{C}\Bigg(\begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix}\left(P + \begin{pmatrix}\mathbf{1}_m\mathbf{1}_m^\top & -\mathbf{1}_m\mathbf{1}_m^\top \\ -\mathbf{1}_m\mathbf{1}_m^\top & \mathbf{1}_m\mathbf{1}_m^\top\end{pmatrix}\right)\begin{pmatrix}\lambda \\ \mu\end{pmatrix} + \begin{pmatrix}y^\top & -y^\top\end{pmatrix}\begin{pmatrix}\lambda \\ \mu\end{pmatrix}\Bigg) \\ &\qquad + w^\top\Bigg(\sum_{i\in K_\lambda} x_i - \sum_{j\in K_\mu} x_j\Bigg) - \sum_{i\in K_\lambda} y_i + \sum_{j\in K_\mu} y_j + (p_f - q_f)b\Bigg).\end{aligned}$$

Using the equations

$$w = X^{\top}(\mu - \lambda)$$

$$b = -(\mathbf{1}_m^{\top}\lambda - \mathbf{1}_m^{\top}\mu) = \begin{pmatrix}\lambda^{\top} & \mu^{\top}\end{pmatrix}\begin{pmatrix}-\mathbf{1}_m \\ \mathbf{1}_m\end{pmatrix},$$

we see that ϵ is determined by λ and μ provided that $(p_f + q_f)/m < \nu$.

By Proposition 20.6(1),

$$\frac{p_f + q_f}{m} \leq \nu,$$

therefore the condition $(p_f + q_f)/m < \nu$ is very natural.

We have implemented this method in `Matlab`, and we have observed that for some examples the choice of ν caused the equation $\nu(p_f + q_f) = m$ to hold. In such cases, running the program again with a slightly perturbed value of ν always succeeded.

The other observation we made is that b tends to be smaller and ϵ tends to be bigger in ν-SV Regression Version 2, so the fit is actually not as good as in ν-SV Regression without penalizing b. Figure 20.16 shows the result of running our program on the data set of Section 20.3. Compare with Figure 20.13.

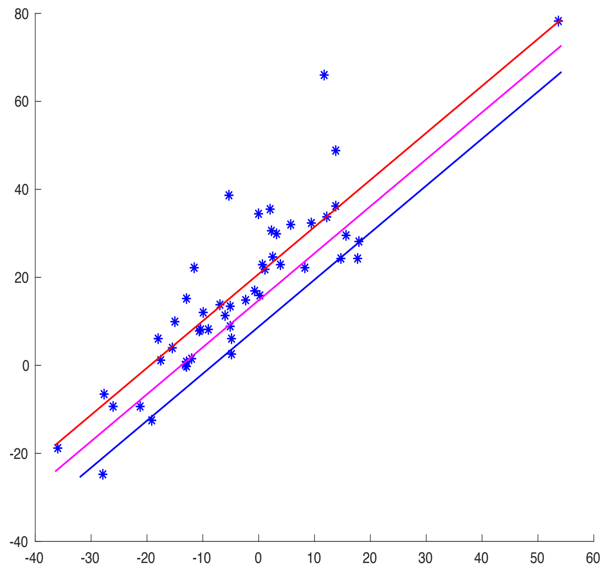


Fig. 20.16 Running ν-SV regression version 2 on a set of 50 points; $\nu = 0.5$.

20.6 Summary

The main concepts and results of this chapter are listed below:

- ν-support vector regression (ν-SV regression).
- Regression estimate.
- Kernel ν-SV regression.
- ϵ-SV regression, ϵ-insensitive SV regression,
- ν-SV regression Version 2; penalizing b.

20.7 Problems

Problem 20.1. Prove that if ν-SV regression succeeds and yields $w, b, \epsilon > 0$, then ϵ-SV regression with the same C and the same value of ϵ also succeeds and returns the same pair (w, b).

Problem 20.2. Prove the formulae

$$\begin{aligned} b &= \left(\left(\sum_{i_0\in I_\lambda} y_{i_0}\right)/|I_\lambda| + \left(\sum_{j_0\in I_\mu} y_{j_0}\right)/|I_\mu|\right)/2 \\ &\quad - w^\top\left(\left(\sum_{i_0\in I_\lambda} x_{i_0}\right)/|I_\lambda| + \left(\sum_{j_0\in I_\mu} x_{j_0}\right)/|I_\mu|\right)/2 \\ \epsilon &= \left(\left(\sum_{j_0\in I_\mu} y_{j_0}\right)/|I_\mu|\right) - \left(\sum_{i_0\in I_\lambda} y_{i_0}\right)/|I_\lambda\Big)/2 \\ &\quad + w^\top\left(\left(\sum_{i_0\in I_\lambda} x_{i_0}\right)/|I_\lambda| - \left(\sum_{j_0\in I_\mu} x_{j_0}\right)/|I_\mu|\right)/2 \end{aligned}$$

stated just before Proposition 20.5.

Problem 20.3. Give the details of the proof of Proposition 20.5. In particular, prove that

$$\begin{aligned} C\left(\nu - \frac{p_f + q_f}{m}\right)\epsilon &= -\begin{pmatrix}\lambda^\top & \mu^\top\end{pmatrix} P \begin{pmatrix}\lambda \\ \mu\end{pmatrix} - \begin{pmatrix}y^\top & -y^\top\end{pmatrix}\begin{pmatrix}\lambda \\ \mu\end{pmatrix} \\ &\quad - \frac{C}{m}\left(w^\top\left(\sum_{i\in K_\lambda} x_i - \sum_{j\in K_\mu} x_j\right) - \sum_{i\in K_\lambda} y_i + \sum_{j\in K_\mu} y_j + (p_f - q_f)b\right). \end{aligned}$$

Problem 20.4. Prove that the matrices

$$A = \begin{pmatrix} \mathbf{1}_m^\top & -\mathbf{1}_m^\top & 0_m^\top & 0_m^\top & 0 \\ \mathbf{1}_m^\top & \mathbf{1}_m^\top & 0_m^\top & 0_m^\top & 1 \\ I_m & 0_{m,m} & I_m & 0_{m,m} & 0_m \\ 0_{m,m} & I_m & 0_{m,m} & I_m & 0_m \end{pmatrix}, \quad A_2 = \begin{pmatrix} \mathbf{1}_m^\top & -\mathbf{1}_m^\top & 0_m^\top & 0_m^\top \\ \mathbf{1}_m^\top & \mathbf{1}_m^\top & 0_m^\top & 0_m^\top \\ I_m & 0_{m,m} & I_m & 0_{m,m} \\ 0_{m,m} & I_m & 0_{m,m} & I_m \end{pmatrix}$$

have rank $2m + 2$.

Problem 20.5. Derive the version of ν-SV regression in which the linear penalty function $\sum_{i=1}^m (\xi_i + \xi_i')$ is replaced by the quadratic penalty function $\sum_{i=1}^m (\xi_i^2 + \xi_i'^2)$. Derive the dual program.

Problem 20.6. The linear penalty function $\sum_{i=1}^m (\xi_i + \xi_i')$ can be replaced by the quadratic penalty function $\sum_{i=1}^m (\xi_i^2 + \xi_i'^2)$. Prove that for an optimal solution we must have $\xi_i \geq 0$ and $\xi_i' \geq 0$, so we may omit the constraints $\xi_i \geq 0$ and $\xi_i' \geq 0$. We must also have $\gamma = 0$ so we may omit the variable γ as well. Prove that $\xi = (m/2C)\lambda$ and $\xi' = (m/2C)\mu$. This problem is very similar to the Soft Margin SVM (SVM$_{s4}$) discussed in Section 18.13.

Problem 20.7. Consider the version of ν-SV regression in Section 20.5. Prove that for any optimal solution with $w \neq 0$ and $\epsilon > 0$, if the inequalities $(p_f + q_f)/m < \nu < 1$ hold, then some point x_i is a support vector.

Problem 20.8. Prove that the matrix

$$A_3 = \begin{pmatrix} \mathbf{1}_m^\top & \mathbf{1}_m^\top & 0_m^\top & 0_m^\top \\ I_m & 0_{m,m} & I_m & 0_{m,m} \\ 0_{m,m} & I_m & 0_{m,m} & I_m \end{pmatrix}$$

has rank $2m + 1$.

Problem 20.9. Consider the version of ν-SV regression in Section 20.5. Prove the following formulae: If $I_\lambda \neq \emptyset$, then

$$\epsilon = w^\top \left(\sum_{i \in I_\lambda} x_i \right) / |I_\lambda| + b - \left(\sum_{i \in I_\lambda} y_i \right) / |I_\lambda|,$$

and if $I_\mu \neq \emptyset$, then

$$\epsilon = -w^\top \left(\sum_{j \in I_\mu} x_j \right) / |I_\mu| - b + \left(\sum_{i \in I_\mu} y_i \right) / |I_\mu|.$$

Problem 20.10. Implement ν-Regression Version 2 described in Section 20.5. Run examples using both the original version and version 2 and compare the results.

Appendix A

Total Orthogonal Families in Hilbert Spaces

A.1 Total Orthogonal Families (Hilbert Bases), Fourier Coefficients

We conclude our quick tour of Hilbert spaces by showing that the notion of orthogonal basis can be generalized to Hilbert spaces. However, the useful notion is not the usual notion of a basis, but a notion which is an abstraction of the concept of Fourier series. Every element of a Hilbert space is the "sum" of its Fourier series.

Definition A.1. Given a Hilbert space E, a family $(u_k)_{k\in K}$ of nonnull vectors is an *orthogonal family* iff the u_k are pairwise orthogonal, i.e., $\langle u_i, u_j\rangle = 0$ for all $i \neq j$ $(i, j \in K)$, and an *orthonormal family* iff $\langle u_i, u_j\rangle = \delta_{i,j}$, for all $i, j \in K$. A *total orthogonal family (or system)* or *Hilbert basis* is an orthogonal family that is dense in E. This means that for every $v \in E$, for every $\epsilon > 0$, there is some finite subset $I \subseteq K$ and some family $(\lambda_i)_{i\in I}$ of complex numbers, such that

$$\left\| v - \sum_{i\in I} \lambda_i u_i \right\| < \epsilon.$$

Given an orthogonal family $(u_k)_{k\in K}$, for every $v \in E$, for every $k \in K$, the scalar $c_k = \langle v, u_k\rangle / \|u_k\|^2$ is called the *k-th Fourier coefficient of v over $(u_k)_{k\in K}$*.

Remark. The terminology Hilbert basis is misleading because a Hilbert basis $(u_k)_{k\in K}$ is not necessarily a basis in the algebraic sense. Indeed, in general, $(u_k)_{k\in K}$ does not span E. Intuitively, it takes linear combinations of the u_k's with infinitely many nonnull coefficients to span E. Technically, this is achieved in terms of limits. In order to avoid the confusion between bases in the algebraic sense and Hilbert bases, some authors refer to algebraic bases as *Hamel bases* and to total orthogonal families (or Hilbert bases) as *Schauder bases*.

Given an orthogonal family $(u_k)_{k \in K}$, for any finite subset I of K, we often call sums of the form $\sum_{i \in I} \lambda_i u_i$ *partial sums of Fourier series*, and if these partial sums converge to a limit denoted as $\sum_{k \in K} c_k u_k$, we call $\sum_{k \in K} c_k u_k$ a *Fourier series*.

However, we have to make sense of such sums! Indeed, when K is unordered or uncountable, the notion of limit or sum has not been defined. This can be done as follows (for more details, see Dixmier [Dixmier (1984)]):

Definition A.2. Given a normed vector space E (say, a Hilbert space), for any nonempty index set K, we say that a family $(u_k)_{k \in K}$ of vectors in E is *summable with sum* $v \in E$ iff for every $\epsilon > 0$, there is some finite subset I of K, such that,

$$\left\| v - \sum_{j \in J} u_j \right\| < \epsilon$$

for every finite subset J with $I \subseteq J \subseteq K$. We say that the family $(u_k)_{k \in K}$ is *summable* iff there is some $v \in E$ such that $(u_k)_{k \in K}$ is summable with sum v. A family $(u_k)_{k \in K}$ is a *Cauchy family* iff for every $\epsilon > 0$, there is a finite subset I of K, such that,

$$\left\| \sum_{j \in J} u_j \right\| < \epsilon$$

for every finite subset J of K with $I \cap J = \emptyset$,

If $(u_k)_{k \in K}$ is summable with sum v, we usually denote v as $\sum_{k \in K} u_k$. The following technical proposition will be needed:

Proposition A.1. *Let E be a complete normed vector space (say, a Hilbert space).*

(1) For any nonempty index set K, a family $(u_k)_{k \in K}$ is summable iff it is a Cauchy family.
(2) Given a family $(r_k)_{k \in K}$ of nonnegative reals $r_k \geq 0$, if there is some real number $B > 0$ such that $\sum_{i \in I} r_i < B$ for every finite subset I of K, then $(r_k)_{k \in K}$ is summable and $\sum_{k \in K} r_k = r$, where r is least upper bound of the set of finite sums $\sum_{i \in I} r_i$ ($I \subseteq K$).

Proof. (1) If $(u_k)_{k \in K}$ is summable, for every finite subset I of K, let

$$u_I = \sum_{i \in I} u_i \quad \text{and} \quad u = \sum_{k \in K} u_k$$

For every $\epsilon > 0$, there is some finite subset I of K such that

$$\|u - u_L\| < \epsilon/2$$

for all finite subsets L such that $I \subseteq L \subseteq K$. For every finite subset J of K such that $I \cap J = \emptyset$, since $I \subseteq I \cup J \subseteq K$ and $I \cup J$ is finite, we have

$$\|u - u_{I \cup J}\| < \epsilon/2 \quad \text{and} \quad \|u - u_I\| < \epsilon/2,$$

and since

$$\|u_{I \cup J} - u_I\| \leq \|u_{I \cup J} - u\| + \|u - u_I\|$$

and $u_{I \cup J} - u_I = u_J$ since $I \cap J = \emptyset$, we get

$$\|u_J\| = \|u_{I \cup J} - u_I\| < \epsilon,$$

which is the condition for $(u_k)_{k \in K}$ to be a Cauchy family.

Conversely, assume that $(u_k)_{k \in K}$ is a Cauchy family. We define inductively a decreasing sequence (X_n) of subsets of E, each of diameter at most $1/n$, as follows: For $n = 1$, since $(u_k)_{k \in K}$ is a Cauchy family, there is some finite subset J_1 of K such that

$$\|u_J\| < 1/2$$

for every finite subset J of K with $J_1 \cap J = \emptyset$. We pick some finite subset J_1 with the above property, and we let $I_1 = J_1$ and

$$X_1 = \{u_I \mid I_1 \subseteq I \subseteq K, \ I \text{ finite}\}.$$

For $n \geq 1$, there is some finite subset J_{n+1} of K such that

$$\|u_J\| < 1/(2n + 2)$$

for every finite subset J of K with $J_{n+1} \cap J = \emptyset$. We pick some finite subset J_{n+1} with the above property, and we let $I_{n+1} = I_n \cup J_{n+1}$ and

$$X_{n+1} = \{u_I \mid I_{n+1} \subseteq I \subseteq K, \ I \text{ finite}\}.$$

Since $I_n \subseteq I_{n+1}$, it is obvious that $X_{n+1} \subseteq X_n$ for all $n \geq 1$. We need to prove that each X_n has diameter at most $1/n$. Since J_n was chosen such that

$$\|u_J\| < 1/(2n)$$

for every finite subset J of K with $J_n \cap J = \emptyset$, and since $J_n \subseteq I_n$, it is also true that

$$\|u_J\| < 1/(2n)$$

for every finite subset J of K with $I_n \cap J = \emptyset$ (since $I_n \cap J = \emptyset$ and $J_n \subseteq I_n$ implies that $J_n \cap J = \emptyset$). Then for every two finite subsets J, L such that $I_n \subseteq J, L \subseteq K$, we have

$$\|u_{J-I_n}\| < 1/(2n) \quad \text{and} \quad \|u_{L-I_n}\| < 1/(2n),$$

and since

$$\|u_J - u_L\| \leq \|u_J - u_{I_n}\| + \|u_{I_n} - u_L\| = \|u_{J-I_n}\| + \|u_{L-I_n}\|,$$

we get

$$\|u_J - u_L\| < 1/n,$$

which proves that $\delta(X_n) \leq 1/n$. Now if we consider the sequence of closed sets $(\overline{X_n})$, we still have $\overline{X_{n+1}} \subseteq \overline{X_n}$, and by Proposition 12.3, $\delta(\overline{X_n}) = \delta(X_n) \leq 1/n$, which means that $\lim_{n\to\infty} \delta(\overline{X_n}) = 0$, and by Proposition 12.3, $\bigcap_{n=1}^{\infty} \overline{X_n}$ consists of a single element u. We claim that u is the sum of the family $(u_k)_{k\in K}$.

For every $\epsilon > 0$, there is some $n \geq 1$ such that $n > 2/\epsilon$, and since $u \in \overline{X_m}$ for all $m \geq 1$, there is some finite subset J_0 of K such that $I_n \subseteq J_0$ and

$$\|u - u_{J_0}\| < \epsilon/2,$$

where I_n is the finite subset of K involved in the definition of X_n. However, since $\delta(X_n) \leq 1/n$, for every finite subset J of K such that $I_n \subseteq J$, we have

$$\|u_J - u_{J_0}\| \leq 1/n < \epsilon/2,$$

and since

$$\|u - u_J\| \leq \|u - u_{J_0}\| + \|u_{J_0} - u_J\|,$$

we get

$$\|u - u_J\| < \epsilon$$

for every finite subset J of K with $I_n \subseteq J$, which proves that u is the sum of the family $(u_k)_{k\in K}$.

(2) Since every finite sum $\sum_{i\in I} r_i$ is bounded by the uniform bound B, the set of these finite sums has a least upper bound $r \leq B$. For every $\epsilon > 0$, since r is the least upper bound of the finite sums $\sum_{i\in I} r_i$ (where I finite, $I \subseteq K$), there is some finite $I \subseteq K$ such that

$$\left| r - \sum_{i\in I} r_i \right| < \epsilon,$$

and since $r_k \geq 0$ for all $k \in K$, we have

$$\sum_{i\in I} r_i \leq \sum_{j\in J} r_j$$

whenever $I \subseteq J$, which shows that

$$\left| r - \sum_{j\in J} r_j \right| \leq \left| r - \sum_{i\in I} r_i \right| < \epsilon$$

for every finite subset J such that $I \subseteq J \subseteq K$, proving that $(r_k)_{k\in K}$ is summable with sum $\sum_{k\in K} r_k = r$. $\square$

Remark. The notion of summability implies that the sum of a family $(u_k)_{k\in K}$ is independent of any order on K. In this sense it is a kind of "commutative summability." More precisely, it is easy to show that for every bijection $\varphi\colon K \to K$ (intuitively, a reordering of K), the family $(u_k)_{k\in K}$ is summable iff the family $(u_l)_{l\in\varphi(K)}$ is summable, and if so, they have the same sum.

The following proposition gives some of the main properties of Fourier coefficients. Among other things, at most countably many of the Fourier coefficient may be nonnull, and the partial sums of a Fourier series converge. Given an orthogonal family $(u_k)_{k\in K}$, we let $U_k = \mathbb{C}u_k$, and $p_{U_k}\colon E \to U_k$ is the projection of E onto U_k.

Proposition A.2. *Let E be a Hilbert space, $(u_k)_{k\in K}$ an orthogonal family in E, and V the closure of the subspace generated by $(u_k)_{k\in K}$. The following properties hold:*

(1) For every $v \in E$, for every finite subset $I \subseteq K$, we have

$$\sum_{i\in I} |c_i|^2 \leq \|v\|^2,$$

where the c_k are the Fourier coefficients of v.

(2) For every vector $v \in E$, if $(c_k)_{k\in K}$ are the Fourier coefficients of v, the following conditions are equivalent:

(2a) $v \in V$

(2b) The family $(c_k u_k)_{k\in K}$ is summable and $v = \sum_{k\in K} c_k u_k$.

(2c) The family $(|c_k|^2)_{k\in K}$ is summable and $\|v\|^2 = \sum_{k\in K} |c_k|^2$;

(3) The family $(|c_k|^2)_{k\in K}$ is summable, and we have the Bessel inequality:

$$\sum_{k\in K} |c_k|^2 \leq \|v\|^2.$$

As a consequence, at most countably many of the c_k may be nonzero. The family $(c_k u_k)_{k\in K}$ forms a Cauchy family, and thus, the Fourier series $\sum_{k\in K} c_k u_k$ converges in E to some vector $u = \sum_{k\in K} c_k u_k$. Furthermore, $u = p_V(v)$.

See Figure A.1.

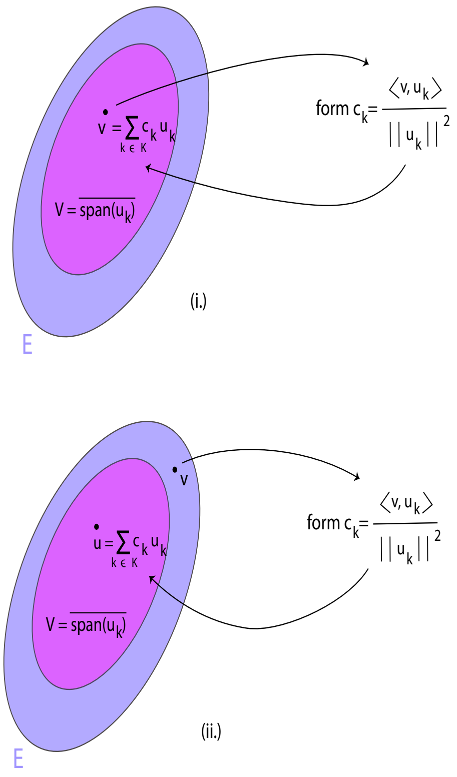

$v = \sum_{k\in K} c_k u_k$
form $c_k = \frac{\langle v, u_k\rangle}{||u_k||^2}$
$V = \overline{\text{span}(u_k)}$
(i.)
E
v
$u = \sum_{k\in K} c_k u_k$
form $c_k = \frac{\langle v, u_k\rangle}{||u_k||^2}$
$V = \overline{\text{span}(u_k)}$
(ii.)
E

Fig. A.1 A schematic illustration of Proposition A.2. Figure (i.) illustrates Condition (2b), while Figure (ii.) illustrates Condition (3). Note E is the purple oval and V is the magenta oval. In both cases, take a vector of E, form the Fourier coefficients c_k, then form the Fourier series $\sum_{k\in K} c_k u_k$. Condition (2b) ensures v equals its Fourier series since $v \in V$. However, if $v \notin V$, the Fourier series does not equal v. Eventually, we will discover that $V = E$, which implies that that Fourier series converges to its vector v.

Proof. (1) Let

$$u_I = \sum_{i \in I} c_i u_i$$

for any finite subset I of K. We claim that $v - u_I$ is orthogonal to u_i for every $i \in I$. Indeed,

$$\begin{aligned}\langle v - u_I, u_i \rangle &= \left\langle v - \sum_{j \in I} c_j u_j, u_i \right\rangle \\ &= \langle v, u_i \rangle - \sum_{j \in I} c_j \langle u_j, u_i \rangle \\ &= \langle v, u_i \rangle - c_i \|u_i\|^2 \\ &= \langle v, u_i \rangle - \langle v, u_i \rangle = 0,\end{aligned}$$

since $\langle u_j, u_i \rangle = 0$ for all $i \neq j$ and $c_i = \langle v, u_i \rangle / \|u_i\|^2$. As a consequence, we have

$$\begin{aligned}\|v\|^2 &= \left\| v - \sum_{i \in I} c_i u_i + \sum_{i \in I} c_i u_i \right\|^2 \\ &= \left\| v - \sum_{i \in I} c_i u_i \right\|^2 + \left\| \sum_{i \in I} c_i u_i \right\|^2 \\ &= \left\| v - \sum_{i \in I} c_i u_i \right\|^2 + \sum_{i \in I} |c_i|^2,\end{aligned}$$

since the u_i are pairwise orthogonal, that is,

$$\|v\|^2 = \left\| v - \sum_{i \in I} c_i u_i \right\|^2 + \sum_{i \in I} |c_i|^2.$$

Thus,

$$\sum_{i \in I} |c_i|^2 \leq \|v\|^2,$$

as claimed.

(2) We prove the chain of implications $(a) \Rightarrow (b) \Rightarrow (c) \Rightarrow (a)$.

$(a) \Rightarrow (b)$: If $v \in V$, since V is the closure of the subspace spanned by $(u_k)_{k \in K}$, for every $\epsilon > 0$, there is some finite subset I of K and some family $(\lambda_i)_{i \in I}$ of complex numbers, such that

$$\left\| v - \sum_{i \in I} \lambda_i u_i \right\| < \epsilon.$$

Now for every finite subset J of K such that $I \subseteq J$, we have

$$\begin{aligned}\Big\|v - \sum_{i\in I}\lambda_i u_i\Big\|^2 &= \Big\|v - \sum_{j\in J} c_j u_j + \sum_{j\in J} c_j u_j - \sum_{i\in I}\lambda_i u_i\Big\|^2 \\ &= \Big\|v - \sum_{j\in J} c_j u_j\Big\|^2 + \Big\|\sum_{j\in J} c_j u_j - \sum_{i\in I}\lambda_i u_i\Big\|^2,\end{aligned}$$

since $I \subseteq J$ and the u_j (with $j \in J$) are orthogonal to $v - \sum_{j\in J} c_j u_j$ by the argument in (1), which shows that

$$\Big\|v - \sum_{j\in J} c_j u_j\Big\| \leq \Big\|v - \sum_{i\in I}\lambda_i u_i\Big\| < \epsilon,$$

and thus, that the family $(c_k u_k)_{k\in K}$ is summable with sum v, so that

$$v = \sum_{k\in K} c_k u_k.$$

$(b) \Rightarrow (c)$: If $v = \sum_{k\in K} c_k u_k$, then for every $\epsilon > 0$, there some finite subset I of K, such that

$$\Big\|v - \sum_{j\in J} c_j u_j\Big\| < \sqrt{\epsilon},$$

for every finite subset J of K such that $I \subseteq J$, and since we proved in (1) that

$$\|v\|^2 = \Big\|v - \sum_{j\in J} c_j u_j\Big\|^2 + \sum_{j\in J}|c_j|^2,$$

we get

$$\|v\|^2 - \sum_{j\in J}|c_j|^2 < \epsilon,$$

which proves that $(|c_k|^2)_{k\in K}$ is summable with sum $\|v\|^2$.

$(c) \Rightarrow (a)$: Finally, if $(|c_k|^2)_{k\in K}$ is summable with sum $\|v\|^2$, for every $\epsilon > 0$, there is some finite subset I of K such that

$$\|v\|^2 - \sum_{j\in J}|c_j|^2 < \epsilon^2$$

for every finite subset J of K such that $I \subseteq J$, and again, using the fact that

$$\|v\|^2 = \Big\|v - \sum_{j\in J} c_j u_j\Big\|^2 + \sum_{j\in J}|c_j|^2,$$

we get

$$\Big\| v - \sum_{j\in J} c_j u_j \Big\| < \epsilon,$$

which proves that $(c_k u_k)_{k\in K}$ is summable with sum $\sum_{k\in K} c_k u_k = v$, and $v \in V$.

(3) Since $\sum_{i\in I} |c_i|^2 \leq \|v\|^2$ for every finite subset I of K, by Proposition A.1(2), the family $(|c_k|^2)_{k\in K}$ is summable. The Bessel inequality

$$\sum_{k\in K} |c_k|^2 \leq \|v\|^2$$

is an obvious consequence of the inequality $\sum_{i\in I} |c_i|^2 \leq \|v\|^2$ (for every finite $I \subseteq K$). Now for every natural number $n \geq 1$, if K_n is the subset of K consisting of all c_k such that $|c_k| \geq 1/n$, the number of elements in K_n is at most

$$\sum_{k\in K_n} |nc_k|^2 \leq n^2 \sum_{k\in K} |c_k|^2 \leq n^2\|v\|^2,$$

which is finite, and thus, at most a countable number of the c_k may be nonzero.

Since $(|c_k|^2)_{k\in K}$ is summable with sum c, by Proposition A.1(1) we know that for every $\epsilon > 0$, there is some finite subset I of K such that

$$\sum_{j\in J} |c_j|^2 < \epsilon^2$$

for every finite subset J of K such that $I \cap J = \emptyset$. Since

$$\Big\| \sum_{j\in J} c_j u_j \Big\|^2 = \sum_{j\in J} |c_j|^2,$$

we get

$$\Big\| \sum_{j\in J} c_j u_j \Big\| < \epsilon.$$

This proves that $(c_k u_k)_{k\in K}$ is a Cauchy family, which, by Proposition A.1(1), implies that $(c_k u_k)_{k\in K}$ is summable since E is complete. Thus, the Fourier series $\sum_{k\in K} c_k u_k$ is summable, with its sum denoted $u \in V$.

Since $\sum_{k\in K} c_k u_k$ is summable with sum u, for every $\epsilon > 0$, there is some finite subset I_1 of K such that

$$\Big\| u - \sum_{j\in J} c_j u_j \Big\| < \epsilon$$

for every finite subset J of K such that $I_1 \subseteq J$. By the triangle inequality, for every finite subset I of K,

$$\Big\| u - v \Big\| \leq \Big\| u - \sum_{i \in I} c_i u_i \Big\| + \Big\| \sum_{i \in I} c_i u_i - v \Big\|.$$

By (2), every $w \in V$ is the sum of its Fourier series $\sum_{k \in K} \lambda_k u_k$, and for every $\epsilon > 0$, there is some finite subset I_2 of K such that

$$\Big\| w - \sum_{j \in J} \lambda_j u_j \Big\| < \epsilon$$

for every finite subset J of K such that $I_2 \subseteq J$. By the triangle inequality, for every finite subset I of K,

$$\Big\| v - \sum_{i \in I} \lambda_i u_i \Big\| \leq \|v - w\| + \Big\| w - \sum_{i \in I} \lambda_i u_i \Big\|.$$

Letting $I = I_1 \cup I_2$, since we showed in (2) that

$$\Big\| v - \sum_{i \in I} c_i u_i \Big\| \leq \Big\| v - \sum_{i \in I} \lambda_i u_i \Big\|$$

for every finite subset I of K, we get

$$\begin{aligned} \|u - v\| &\leq \Big\| u - \sum_{i \in I} c_i u_i \Big\| + \Big\| \sum_{i \in I} c_i u_i - v \Big\| \\ &\leq \Big\| u - \sum_{i \in I} c_i u_i \Big\| + \Big\| \sum_{i \in I} \lambda_i u_i - v \Big\| \\ &\leq \Big\| u - \sum_{i \in I} c_i u_i \Big\| + \|v - w\| + \Big\| w - \sum_{i \in I} \lambda_i u_i \Big\|, \end{aligned}$$

and thus

$$\|u - v\| \leq \|v - w\| + 2\epsilon.$$

Since this holds for every $\epsilon > 0$, we have

$$\|u - v\| \leq \|v - w\|$$

for all $w \in V$, i.e. $\|v - u\| = d(v, V)$, with $u \in V$, which proves that $u = p_V(v)$. ☐

A.2 The Hilbert Space $\ell^2(K)$ and the Riesz–Fischer Theorem

Proposition A.2 suggests looking at the space of sequences $(z_k)_{k\in K}$ (where $z_k \in \mathbb{C}$) such that $(|z_k|^2)_{k\in K}$ is summable. Indeed, such spaces are Hilbert spaces, and it turns out that every Hilbert space is isomorphic to one of those. Such spaces are the infinite-dimensional version of the spaces $\mathbb{C}^n$ under the usual Euclidean norm.

Definition A.3. Given any nonempty index set K, the space $\ell^2(K)$ is the set of all sequences $(z_k)_{k\in K}$, where $z_k \in \mathbb{C}$, such that $(|z_k|^2)_{k\in K}$ is summable, i.e., $\sum_{k\in K} |z_k|^2 < \infty$.

Remarks.

(1) When K is a finite set of cardinality n, $\ell^2(K)$ is isomorphic to $\mathbb{C}^n$.
(2) When $K = \mathbb{N}$, the space $\ell^2(\mathbb{N})$ corresponds to the space ℓ^2 of Example 2 in Section 13.1 (Vol. I). In that example, we claimed that ℓ^2 was a Hermitian space, and in fact, a Hilbert space. We now prove this fact for any index set K.

Proposition A.3. *Given any nonempty index set K, the space $\ell^2(K)$ is a Hilbert space under the Hermitian product*

$$\langle (x_k)_{k\in K}, (y_k)_{k\in K}\rangle = \sum_{k\in K} x_k\overline{y_k}.$$

The subspace consisting of sequences $(z_k)_{k\in K}$ such that $z_k = 0$, except perhaps for finitely many k, is a dense subspace of $\ell^2(K)$.

Proof. First we need to prove that $\ell^2(K)$ is a vector space. Assume that $(x_k)_{k\in K}$ and $(y_k)_{k\in K}$ are in $\ell^2(K)$. This means that $(|x_k|^2)_{k\in K}$ and $(|y_k|^2)_{k\in K}$ are summable, which, in view of Proposition A.1(2), is equivalent to the existence of some positive bounds A and B such that $\sum_{i\in I} |x_i|^2 < A$ and $\sum_{i\in I} |y_i|^2 < B$, for every finite subset I of K. To prove that $(|x_k + y_k|^2)_{k\in K}$ is summable, it is sufficient to prove that there is some $C > 0$ such that $\sum_{i\in I} |x_i + y_i|^2 < C$ for every finite subset I of K. However, the parallelogram inequality implies that

$$\sum_{i\in I} |x_i + y_i|^2 \leq \sum_{i\in I} 2(|x_i|^2 + |y_i|^2) \leq 2(A + B),$$

for every finite subset I of K, and we conclude by Proposition A.1(2). Similarly, for every $\lambda \in \mathbb{C}$,

$$\sum_{i\in I} |\lambda x_i|^2 \leq \sum_{i\in I} |\lambda|^2|x_i|^2 \leq |\lambda|^2 A,$$

and $(\lambda_k x_k)_{k\in K}$ is summable. Therefore, $\ell^2(K)$ is a vector space.

By the Cauchy-Schwarz inequality,

$$\begin{aligned}\sum_{i\in I}|x_i\overline{y_i}| &\leq \sum_{i\in I}|x_i||y_i| \leq \big(\sum_{i\in I}|x_i|^2\big)^{1/2}\big(\sum_{i\in I}|xy_i|^2\big)^{1/2}\\ &\leq \sum_{i\in I}(|x_i|^2+|y_i|^2)/2 \leq (A+B)/2,\end{aligned}$$

for every finite subset I of K. For the third inequality we used the fact that

$$4CD \leq (C+D)^2,$$

(with $C = \sum_{i\in I}|x_i|^2$ and $D = \sum_{i\in I}|y_i|^2$) which is equivalent to

$$(C-D)^2 \geq 0.$$

By Proposition A.1(2), $(|x_k\overline{y_k}|)_{k\in K}$ is summable. The customary language is that $(x_k\overline{y_k})_{k\in K}$ is absolutely summable. However, it is a standard fact that this implies that $(x_k\overline{y_k})_{k\in K}$ is summable (For every $\epsilon > 0$, there is some finite subset I of K such that

$$\sum_{j\in J}|x_j\overline{y_j}| < \epsilon$$

for every finite subset J of K such that $I \cap J = \emptyset$, and thus

$$|\sum_{j\in J}x_j\overline{y_j}| \leq \sum_{i\in J}|x_j\overline{y_j}| < \epsilon,$$

proving that $(x_k\overline{y_k})_{k\in K}$ is a Cauchy family, and thus summable). We still have to prove that $\ell^2(K)$ is complete.

Consider a sequence $((\lambda_k^n)_{k\in K})_{n\geq 1}$ of sequences $(\lambda_k^n)_{k\in K} \in \ell^2(K)$, and assume that it is a Cauchy sequence. This means that for every $\epsilon > 0$, there is some $N \geq 1$ such that

$$\sum_{k\in K}|\lambda_k^m - \lambda_k^n|^2 < \epsilon^2$$

for all $m, n \geq N$. For every fixed $k \in K$, this implies that

$$|\lambda_k^m - \lambda_k^n| < \epsilon$$

for all $m, n \geq N$, which shows that $(\lambda_k^n)_{n\geq 1}$ is a Cauchy sequence in $\mathbb{C}$. Since $\mathbb{C}$ is complete, the sequence $(\lambda_k^n)_{n\geq 1}$ has a limit $\lambda_k \in \mathbb{C}$. We claim that $(\lambda_k)_{k\in K} \in \ell^2(K)$ and that this is the limit of $((\lambda_k^n)_{k\in K})_{n\geq 1}$.

Given any $\epsilon > 0$, the fact that $((\lambda_k^n)_{k\in K})_{n\geq 1}$ is a Cauchy sequence implies that there is some $N \geq 1$ such that for every finite subset I of K, we have

$$\sum_{i\in I} |\lambda_i^m - \lambda_i^n|^2 < \epsilon/4$$

for all $m, n \geq N$. Let $p = |I|$. Then

$$|\lambda_i^m - \lambda_i^n| < \frac{\sqrt{\epsilon}}{2\sqrt{p}}$$

for every $i \in I$. Since λ_i is the limit of $(\lambda_i^n)_{n\geq 1}$, we can find some n large enough so that

$$|\lambda_i^n - \lambda_i| < \frac{\sqrt{\epsilon}}{2\sqrt{p}}$$

for every $i \in I$. Since

$$|\lambda_i^m - \lambda_i| \leq |\lambda_i^m - \lambda_i^n| + |\lambda_i^n - \lambda_i|,$$

we get

$$|\lambda_i^m - \lambda_i| < \frac{\sqrt{\epsilon}}{\sqrt{p}},$$

and thus,

$$\sum_{i\in I} |\lambda_i^m - \lambda_i|^2 < \epsilon,$$

for all $m \geq N$. Since the above holds for every finite subset I of K, by Proposition A.1(2), we get

$$\sum_{k\in K} |\lambda_k^m - \lambda_k|^2 < \epsilon,$$

for all $m \geq N$. This proves that $(\lambda_k^m - \lambda_k)_{k\in K} \in \ell^2(K)$ for all $m \geq N$, and since $\ell^2(K)$ is a vector space and $(\lambda_k^m)_{k\in K} \in \ell^2(K)$ for all $m \geq 1$, we get $(\lambda_k)_{k\in K} \in \ell^2(K)$. However,

$$\sum_{k\in K} |\lambda_k^m - \lambda_k|^2 < \epsilon$$

for all $m \geq N$, means that the sequence $(\lambda_k^m)_{k\in K}$ converges to $(\lambda_k)_{k\in K} \in \ell^2(K)$. The fact that the subspace consisting of sequences $(z_k)_{k\in K}$ such that $z_k = 0$ except perhaps for finitely many k is a dense subspace of $\ell^2(K)$ is left as an easy exercise. $\square$

Remark: The subspace consisting of all sequences $(z_k)_{k\in K}$ such that $z_k = 0$, except perhaps for finitely many k, provides an example of a subspace which is not closed in $\ell^2(K)$. Indeed, this space is strictly contained in $\ell^2(K)$, since there are countable sequences of nonnull elements in $\ell^2(K)$ (why?).

We just need two more propositions before being able to prove that every Hilbert space is isomorphic to some $\ell^2(K)$.

Proposition A.4. *Let E be a Hilbert space, and $(u_k)_{k\in K}$ an orthogonal family in E. The following properties hold:*

(1) For every family $(\lambda_k)_{k\in K} \in \ell^2(K)$, the family $(\lambda_k u_k)_{k\in K}$ is summable. Furthermore, $v = \sum_{k\in K} \lambda_k u_k$ is the only vector such that $c_k = \lambda_k$ for all $k \in K$, where the c_k are the Fourier coefficients of v.

(2) For any two families $(\lambda_k)_{k\in K} \in \ell^2(K)$ and $(\mu_k)_{k\in K} \in \ell^2(K)$, if $v = \sum_{k\in K} \lambda_k u_k$ and $w = \sum_{k\in K} \mu_k u_k$, we have the following equation, also called Parseval identity:

$$\langle v, w\rangle = \sum_{k\in K} \lambda_k \overline{\mu_k}.$$

Proof. (1) The fact that $(\lambda_k)_{k\in K} \in \ell^2(K)$ means that $(|\lambda_k|^2)_{k\in K}$ is summable. The proof given in Proposition A.2 (3) applies to the family $(|\lambda_k|^2)_{k\in K}$ (instead of $(|c_k|^2)_{k\in K}$), and yields the fact that $(\lambda_k u_k)_{k\in K}$ is summable. Letting $v = \sum_{k\in K} \lambda_k u_k$, recall that $c_k = \langle v, u_k\rangle / \|u_k\|^2$. Pick some $k \in K$. Since $\langle -, -\rangle$ is continuous, for every $\epsilon > 0$, there is some $\eta > 0$ such that

$$|\langle v, u_k\rangle - \langle w, u_k\rangle| < \epsilon \|u_k\|^2$$

whenever

$$\|v - w\| < \eta.$$

However, since for every $\eta > 0$, there is some finite subset I of K such that

$$\left\| v - \sum_{j\in J} \lambda_j u_j \right\| < \eta$$

for every finite subset J of K such that $I \subseteq J$, we can pick $J = I \cup \{k\}$ and letting $w = \sum_{j\in J} \lambda_j u_j$ we get

$$\left| \langle v, u_k\rangle - \left\langle \sum_{j\in J} \lambda_j u_j, u_k \right\rangle \right| < \epsilon \|u_k\|^2.$$

However,

$$\langle v, u_k \rangle = c_k \|u_k\|^2 \quad \text{and} \quad \left\langle \sum_{j \in J} \lambda_j u_j, u_k \right\rangle = \lambda_k \|u_k\|^2,$$

and thus, the above proves that $|c_k - \lambda_k| < \epsilon$ for every $\epsilon > 0$, and thus, that $c_k = \lambda_k$.

(2) Since $\langle -, - \rangle$ is continuous, for every $\epsilon > 0$, there are some $\eta_1 > 0$ and $\eta_2 > 0$, such that

$$|\langle x, y \rangle| < \epsilon$$

whenever $\|x\| < \eta_1$ and $\|y\| < \eta_2$. Since $v = \sum_{k \in K} \lambda_k u_k$ and $w = \sum_{k \in K} \mu_k u_k$, there is some finite subset I_1 of K such that

$$\left\| v - \sum_{j \in J} \lambda_j u_j \right\| < \eta_1$$

for every finite subset J of K such that $I_1 \subseteq J$, and there is some finite subset I_2 of K such that

$$\left\| w - \sum_{j \in J} \mu_j u_j \right\| < \eta_2$$

for every finite subset J of K such that $I_2 \subseteq J$. Letting $I = I_1 \cup I_2$, we get

$$\left| \left\langle v - \sum_{i \in I} \lambda_i u_i, w - \sum_{i \in I} \mu_i u_i \right\rangle \right| < \epsilon.$$

Furthermore,

$$\begin{aligned}
\langle v, w \rangle &= \left\langle v - \sum_{i \in I} \lambda_i u_i + \sum_{i \in I} \lambda_i u_i, w - \sum_{i \in I} \mu_i u_i + \sum_{i \in I} \mu_i u_i \right\rangle \\
&= \left\langle v - \sum_{i \in I} \lambda_i u_i, w - \sum_{i \in I} \mu_i u_i \right\rangle + \sum_{i \in I} \lambda_i \overline{\mu_i},
\end{aligned}$$

since the u_i are orthogonal to $v - \sum_{i \in I} \lambda_i u_i$ and $w - \sum_{i \in I} \mu_i u_i$ for all $i \in I$. This proves that for every $\epsilon > 0$, there is some finite subset I of K such that

$$\left| \langle v, w \rangle - \sum_{i \in I} \lambda_i \overline{\mu_i} \right| < \epsilon.$$

We already know from Proposition A.3 that $(\lambda_k \overline{\mu_k})_{k \in K}$ is summable, and since $\epsilon > 0$ is arbitrary we get

$$\langle v, w \rangle = \sum_{k \in K} \lambda_k \overline{\mu_k}.$$

$\square$

The next proposition states properties characterizing Hilbert bases (total orthogonal families).

Proposition A.5. *Let E be a Hilbert space, and let $(u_k)_{k\in K}$ be an orthogonal family in E. The following properties are equivalent:*

(1) The family $(u_k)_{k\in K}$ is a total orthogonal family.
(2) For every vector $v \in E$, if $(c_k)_{k\in K}$ are the Fourier coefficients of v, then the family $(c_k u_k)_{k\in K}$ is summable and $v = \sum_{k\in K} c_k u_k$.
(3) For every vector $v \in E$, we have the Parseval identity:

$$\|v\|^2 = \sum_{k\in K} |c_k|^2.$$

(4) For every vector $u \in E$, if $\langle u, u_k\rangle = 0$ for all $k \in K$, then $u = 0$.

See Figure A.2.

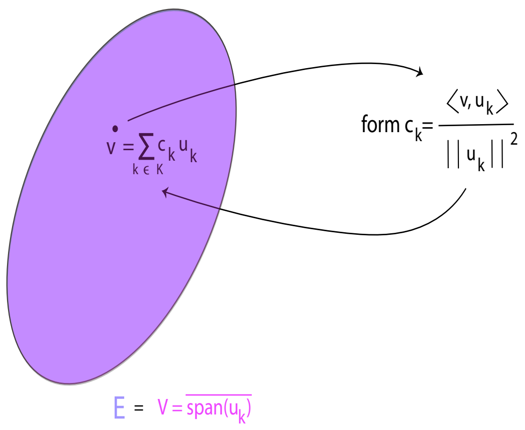

$v = \sum_{k \in K} c_k u_k$
form $c_k = \frac{\langle v, u_k \rangle}{\|u_k\|^2}$
$E = V = \overline{\text{span}(u_k)}$

Fig. A.2 A schematic illustration of Proposition A.5. Since $(u_k)_{k\in K}$ is a Hilbert basis, $V = E$. Then given a vector of E, if we form the Fourier coefficients c_k, then form the Fourier series $\sum_{k\in K} c_k u_k$, we are ensured that v is equal to its Fourier series.

Proof. The equivalence of (1), (2), and (3) is an immediate consequence of Proposition A.2 and Proposition A.4.

(4) If $(u_k)_{k\in K}$ is a total orthogonal family and $\langle u, u_k\rangle = 0$ for all $k \in K$, since $u = \sum_{k\in K} c_k u_k$ where $c_k = \langle u, u_k\rangle/\|u_k\|^2$, we have $c_k = 0$ for all $k \in K$, and $u = 0$.

Conversely, assume that the closure V of $(u_k)_{k\in K}$ is different from E. Then by Proposition 12.6, we have $E = V \oplus V^{\perp}$, where $V^{\perp}$ is the orthogonal complement of V, and $V^{\perp}$ is nontrivial since $V \neq E$. As a consequence, there is some nonnull vector $u \in V^{\perp}$. But then u is orthogonal to every vector in V, and in particular,

$$\langle u, u_k \rangle = 0$$

for all $k \in K$, which, by assumption, implies that $u = 0$, contradicting the fact that $u \neq 0$. $\square$

Remarks:

(1) If E is a Hilbert space and $(u_k)_{k\in K}$ is a total orthogonal family in E, there is a simpler argument to prove that $u = 0$ if $\langle u, u_k \rangle = 0$ for all $k \in K$ based on the continuity of $\langle -, - \rangle$. The argument is to prove that the assumption implies that $\langle v, u \rangle = 0$ for all $v \in E$. Since $\langle -, - \rangle$ is positive definite, this implies that $u = 0$. By continuity of $\langle -, - \rangle$, for every $\epsilon > 0$, there is some $\eta > 0$ such that for every finite subset I of K, for every family $(\lambda_i)_{i\in I}$, for every $v \in E$,

$$\left| \langle v, u \rangle - \left\langle \sum_{i\in I} \lambda_i u_i, u \right\rangle \right| < \epsilon$$

whenever

$$\left\| v - \sum_{i\in I} \lambda_i u_i \right\| < \eta.$$

Since $(u_k)_{k\in K}$ is dense in E, for every $v \in E$, there is some finite subset I of K and some family $(\lambda_i)_{i\in I}$ such that

$$\left\| v - \sum_{i\in I} \lambda_i u_i \right\| < \eta,$$

and since by assumption, $\left\langle \sum_{i\in I} \lambda_i u_i, u \right\rangle = 0$, we get

$$|\langle v, u \rangle| < \epsilon.$$

Since this holds for every $\epsilon > 0$, we must have $\langle v, u \rangle = 0$

(2) If V is any nonempty subset of E, the kind of argument used in the previous remark can be used to prove that $V^{\perp}$ is closed (even if V is not), and that $V^{\perp\perp}$ is the closure of V.

We will now prove that every Hilbert space has some Hilbert basis. This requires using a fundamental theorem from set theory known as *Zorn's lemma*, which we quickly review.

Given any set X with a partial ordering $\leq$, recall that a nonempty subset C of X is a *chain* if it is totally ordered (i.e., for all $x, y \in C$, either $x \leq y$ or $y \leq x$). A nonempty subset Y of X is *bounded* iff there is some $b \in X$ such that $y \leq b$ for all $y \in Y$. Some $m \in X$ is *maximal* iff for every $x \in X$, $m \leq x$ implies that $x = m$. We can now state Zorn's lemma. For more details, see Rudin [Rudin (1987)], Lang [Lang (1993)], or Artin [Artin (1991)].

Proposition A.6. *(Zorn's lemma) Given any nonempty partially ordered set X, if every (nonempty) chain in X is bounded, then X has some maximal element.*

We can now prove the existence of Hilbert bases. We define a partial order on families $(u_k)_{k\in K}$ as follows: for any two families $(u_k)_{k\in K_1}$ and $(v_k)_{k\in K_2}$, we say that

$$(u_k)_{k\in K_1} \leq (v_k)_{k\in K_2}$$

iff $K_1 \subseteq K_2$ and $u_k = v_k$ for all $k \in K_1$. This is clearly a partial order.

Proposition A.7. *Let E be a Hilbert space. Given any orthogonal family $(u_k)_{k\in K}$ in E, there is a total orthogonal family $(u_l)_{l\in L}$ containing $(u_k)_{k\in K}$.*

Proof. Consider the set $\mathcal{S}$ of all orthogonal families greater than or equal to the family $B = (u_k)_{k\in K}$. We claim that every chain in $\mathcal{S}$ is bounded. Indeed, if $C = (C_l)_{l\in L}$ is a chain in $\mathcal{S}$, where $C_l = (u_{k,l})_{k\in K_l}$, the union family

$$(u_k)_{k\in \bigcup_{l\in L} K_l}, \text{ where } u_k = u_{k,l} \text{ whenever } k \in K_l,$$

is clearly an upper bound for C, and it is immediately verified that it is an orthogonal family. By Zorn's Lemma A.6, there is a maximal family $(u_l)_{l\in L}$ containing $(u_k)_{k\in K}$. If $(u_l)_{l\in L}$ is not dense in E, then its closure V is strictly contained in E, and by Proposition 12.6, the orthogonal complement $V^\perp$ of V is nontrivial since $V \neq E$. As a consequence, there is some nonnull vector $u \in V^\perp$. But then u is orthogonal to every vector in $(u_l)_{l\in L}$, and we can form an orthogonal family strictly greater than $(u_l)_{l\in L}$ by adding u to this family, contradicting the maximality of $(u_l)_{l\in L}$. Therefore, $(u_l)_{l\in L}$ is dense in E, and thus it is a Hilbert basis. $\square$

Remark: It is possible to prove that all Hilbert bases for a Hilbert space E have index sets K of the same cardinality. For a proof, see Bourbaki [Bourbaki (1981)].

Definition A.4. A Hilbert space E is *separable* if its Hilbert bases are countable.

At last, we can prove that every Hilbert space is isomorphic to some Hilbert space $\ell^2(K)$ for some suitable K.

Theorem A.1. *(Riesz–Fischer) For every Hilbert space E, there is some nonempty set K such that E is isomorphic to the Hilbert space $\ell^2(K)$. More specifically, for any Hilbert basis $(u_k)_{k\in K}$ of E, the maps $f\colon \ell^2(K) \to E$ and $g\colon E \to \ell^2(K)$ defined such that*

$$f\left((\lambda_k)_{k\in K}\right) = \sum_{k\in K} \lambda_k u_k \quad \textit{and} \quad g(u) = \left(\langle u, u_k\rangle / \|u_k\|^2\right)_{k\in K} = (c_k)_{k\in K},$$

are bijective linear isometries such that $g \circ f = \mathrm{id}$ and $f \circ g = \mathrm{id}$.

Proof. By Proposition A.4 (1), the map f is well defined, and it is clearly linear. By Proposition A.2 (3), the map g is well defined, and it is also clearly linear. By Proposition A.2 (2b), we have

$$f(g(u)) = u = \sum_{k\in K} c_k u_k,$$

and by Proposition A.4 (1), we have

$$g(f\left((\lambda_k)_{k\in K}\right)) = (\lambda_k)_{k\in K},$$

and thus $g \circ f = \mathrm{id}$ and $f \circ g = \mathrm{id}$. By Proposition A.4 (2), the linear map g is an isometry. Therefore, f is a linear bijection and an isometry between $\ell^2(K)$ and E, with inverse g. ☐

Remark: The surjectivity of the map $g\colon E \to \ell^2(K)$ is known as the *Riesz–Fischer* theorem.

Having done all this hard work, we sketch how these results apply to Fourier series. Again we refer the readers to Rudin [Rudin (1987)] or Lang [Lang (1996, 1997)] for a comprehensive exposition.

Let $\mathcal{C}(T)$ denote the set of all periodic continuous functions $f\colon [-\pi, \pi] \to \mathbb{C}$ with period 2π. There is a Hilbert space $\mathrm{L}^2(T)$ containing $\mathcal{C}(T)$ and such that $\mathcal{C}(T)$ is dense in $\mathrm{L}^2(T)$, whose inner product is given by

$$\langle f, g\rangle = \int_{-\pi}^{\pi} f(x)\overline{g(x)}dx.$$

The Hilbert space $\mathrm{L}^2(T)$ is the space of *Lebesgue square-integrable periodic functions* (of period 2π).

It turns out that the family $(e^{ikx})_{k\in\mathbb{Z}}$ is a total orthogonal family in $\mathrm{L}^2(T)$, because it is already dense in $\mathcal{C}(T)$ (for instance, see Rudin [Rudin (1987)]). Then the Riesz–Fischer theorem says that for every family $(c_k)_{k\in\mathbb{Z}}$ of complex numbers such that

$$\sum_{k\in\mathbb{Z}} |c_k|^2 < \infty,$$

there is a unique function $f \in \mathrm{L}^2(T)$ such that f is equal to its Fourier series

$$f(x) = \sum_{k\in\mathbb{Z}} c_k e^{ikx},$$

where the Fourier coefficients c_k of f are given by the formula

$$c_k = \frac{1}{2\pi}\int_{-\pi}^{\pi} f(t)e^{-ikt}dt.$$

The Parseval theorem says that

$$\sum_{k=-\infty}^{+\infty} c_k\overline{d_k} = \frac{1}{2\pi}\int_{-\pi}^{\pi} f(t)\overline{g(t)}dt$$

for all $f, g \in \mathrm{L}^2(T)$, where c_k and d_k are the Fourier coefficients of f and g.

Thus, there is an isomorphism between the two Hilbert spaces $\mathrm{L}^2(T)$ and $\ell^2(\mathbb{Z})$, which is the deep reason why the Fourier coefficients "work." Theorem A.1 implies that the Fourier series $\sum_{k\in\mathbb{Z}} c_k e^{ikx}$ of a function $f \in \mathrm{L}^2(T)$ converges to f in the L^2-sense, i.e., in the mean-square sense. This does not necessarily imply that the Fourier series converges to f pointwise! This is a subtle issue, and for more on this subject, the reader is referred to Lang [Lang (1996, 1997)] or Schwartz [Schwartz (1993a,b)].

We can also consider the set $\mathcal{C}([-1,1])$ of continuous functions $f\colon [-1,1] \to \mathbb{C}$. There is a Hilbert space $\mathrm{L}^2([-1,1])$ containing $\mathcal{C}([-1,1])$ and such that $\mathcal{C}([-1,1])$ is dense in $\mathrm{L}^2([-1,1])$, whose inner product is given by

$$\langle f, g\rangle = \int_{-1}^{1} f(x)\overline{g(x)}dx.$$

The Hilbert space $\mathrm{L}^2([-1,1])$ is the space of *Lebesgue square-integrable functions* over $[-1,1]$. The Legendre polynomials $P_n(x)$ defined in Example

5 of Section 11.2 (Chapter 11, Vol. I) form a Hilbert basis of $\mathrm{L}^2([-1,1])$. Recall that if we let f_n be the function

$$f_n(x) = (x^2 - 1)^n,$$

$P_n(x)$ is defined as follows:

$$P_0(x) = 1, \quad \text{and} \quad P_n(x) = \frac{1}{2^n n!} f_n^{(n)}(x),$$

where $f_n^{(n)}$ is the nth derivative of f_n. The reason for the leading coefficient is to get $P_n(1) = 1$. It can be shown with much efforts that

$$P_n(x) = \sum_{0 \leq k \leq n/2} (-1)^k \frac{(2(n-k))!}{2^n (n-k)! k! (n-2k)!} x^{n-2k}.$$

A.3 Summary

The main concepts and results of this chapter are listed below:

- Hilbert space
- Orthogonal family, total orthogonal family.
- Hilbert basis.
- Fourier coefficients.
- Hamel bases, Schauder bases.
- Fourier series.
- Cauchy family, summable family.
- Bessel inequality.
- The Hilbert space $\ell^2(K)$.
- Parseval identity.
- Zorn's lemma.
- Riesz–Fischer theorem.
- Legendre polynomials.

A.4 Problems

Problem A.1. Prove that the subspace consisting of sequences $(z_k)_{k \in K}$ such that $z_k = 0$ except perhaps for finitely many k is a dense suspace of $\ell^2(K)$.

Problem A.2. If V is any nonempty subset of E, prove that $V^{\perp}$ is closed (even if V is not) and that $V^{\perp\perp}$ is the closure of V (see the remarks following Proposition A.5).

Appendix B

Matlab Programs

B.1 Hard Margin ($\mathbf{SVM}_{h2}$)

The following `Matlab` programs implement the method described in Section 16.7.

The first program is the heart of the method; it implements ADMM for quadratic programming.

```
function [x,u,nr,ns,k] = qsolve1(P, q, A, b, rho, tolr, tols,
                                 iternum)
%  Solve a quadratic programming problem
%  min (1/2) x^T P x + x^T q + r
%  subject to Ax = b, x >= 0  using ADMM
%  P n x n,   q, r, in R^n, A m x n,  b in R^m
%  A of rank m

m = size(A,1); fprintf('m =  %d ',m)
n = size(P,1); fprintf('    n =  %d \n',n)
u = ones(n,1); u(1,1) = 0; % to initialize u
z = ones(n,1);             % to initialize z
% iternum = maximum number of iterations;
% iternum = 80000 works well
k = 0; nr= 1; ns = 1;
%  typically tolr = 10^(-10); tols = 10^(-10);
%  Convergence is controlled by the norm nr of the primal
%  residual r
%  and the norm ns of the dual residual s

while (k <= iternum) && (ns > tols || nr > tolr)
```

```
    z0 = z;
    k = k+1;
    %  Makes KKT matrix
    KK = [P + rho* eye(n) A'; A zeros(m,m)];
    % Makes right hand side of KKT equation
    bb = [-q + rho*(z - u); b];
    % Solves KKT equation
    xx = KK\bb;
    %  update x, z, u (ADMM update steps)
    x = xx(1:n);
    z = poslin(x + u);
    u = u + x - z;
    %  to test stopping criterion
    r = x - z;                      % primal residual
    nr = sqrt(r'*r);                % norm of primal residual
    s = rho*(z - z0);               % dual residual
    ns = sqrt(s'*s);                % norm of dual residual
end
end
```

The second program SBVMhard2 implements hard margin SVM (v2).

```
function [lamb,mu,w] = SVMhard2(rho,u,v)
%
%   Runs hard margin SVM version 2
%
%   p green vectors u_1, ..., u_p in n x p array u
%   q red   vectors v_1, ..., v_q in n x q array v
%
%   First builds the matrices for the dual program
%
p = size(u,2); q = size(v,2); n = size(u,1);
[A,c,X,Pa,qa] = buildhardSVM2(u,v);
%
%  Runs quadratic solver
%
tolr = 10^(-10); tols = 10^(-10); iternum = 80000;
[lam,U,nr,ns,kk] = qsolve1(Pa, qa, A, c, rho, tolr, tols,
                           iternum);
```

```
fprintf('nr =  %d ',nr)
fprintf('   ns =  %d \n',ns)
fprintf('kk =  %d \n',kk)
if kk > iternum
   fprintf('** qsolve did not converge. Problem
           not solvable ** \n')
end
w = -X*lam;
nw = sqrt(w'*w);   % norm of w
fprintf('nw =  %.15f \n',nw)
delta = 1/nw;
fprintf('delta =  %.15f \n',delta)
if   delta < 10^(-9)
     fprintf('** Warning, delta too small, program
              does not converge ** \n')
end
%
lamb = lam(1:p,1);
mu = lam(p+1:p+q,1);
b = 0;
tols = 10^(-10);
% tols < lambda_i; finds the nonzero lambda_i
[lambnz,numsvl1] = countmlu2(lamb,tols);
% tols < mu_i; finds the nonzero mu_j
[munz,numsvm1] = countmlv2(mu,tols);
fprintf('numsvl1 = %d ',numsvl1)
fprintf('  numsvm1 = %d \n',numsvm1)

if numsvl1 > 0 && numsvm1 > 0
   sx1 = zeros(n,1);  num1 = 0;
   sx2 = zeros(n,1);  num2 = 0;
   for i = 1:p
       if lambnz(i) > 0
          sx1 = sx1 + u(:,i);
          num1 = num1 + 1;
       end
   end
   for j = 1:q
       if munz(j) > 0
```

```
            sx2 = sx2 + v(:,j);
            num2 = num2 + 1;
         end
   end
   b =  (w'*(sx1/num1 + sx2/num2))/2;
   fprintf('b =  %.15f \n',b)
else
   fprintf('** Not enough support vectors ** \n')
end

if n == 2
   [ll,mm] = showdata(u,v);
   if numsvl1 > 0 && numsvm1 > 0
      showSVMs2(w,b,1,ll,mm,nw)
   end
end
end
```

The function buildhardSVM2 builds the constraint matrix and the matrices defining the quadratic functional.

```
function [A,c,X,Xa,q] = buildhardSVM2(u,v)
%   builds the matrix of constraints A for
%   hard SVM h2, and the right hand side c
%   Aso builds X and Xa = X'*X, and the vector q = -1_{p+q}
%   for the linear part of the quadratic function
%   The right-hand side is c = 0 (Ax = 0).
p = size(u,2); q = size(v,2);
A = [ones(1,p) -ones(1,q)];
c = 0;
X = [-u v];
Xa = X'*X;
q = -ones(p+q,1);
end
```

The function countmlu2 returns a vector consisting of those λ_i such that $\lambda_i > 0$, and the number of such λ_i.

```
function [lambnz, mlu] = countmlu2(lambda,tols)
%  Counts the number of points u_i (in u)
%  such that lambda_i > 0 and returns a vector
%  of these lambda_i

% tols = 10^(-11);
p = size(lambda,1); lambnz = zeros(p,1);
mlu = 0;
for i = 1:p
    if lambda(i) > tols
        mlu = mlu + 1;
        lambnz(i) = lambda(i);
    end
end
end
```

The function `countmlv2` returns a vector consisting of those μ_j such that $\mu_j > 0$, and the number of such μ_j. It is similar to `countmlu2`. Here a judicious choice of `tols` is crucial and one has to experiment with various values.

The function `showdata` displays the data points (the u_i and the v_j) and the function `showSVMs2` displays the separating line and the two margin lines.

```
function  showSVMs2(w,b,eta,ll,mm,nw)
%
%  Function to display the result of running SVM
%  on p blue points u_1, ..., u_p in u
%  and q red points v_1, ..., v_q in v

l = makeline(w,b,ll,mm,nw);        %  makes separating line
lm1 = makeline(w,b+eta,ll,mm,nw);  %  makes blue margin line
lm2 = makeline(w,b-eta,ll,mm,nw);  %  makes red margin line

%  plots separating line
plot(l(1,:),l(2,:),'-m','LineWidth',1.2)
%  plots blue margin line
plot(lm1(1,:),lm1(2,:),'-b','LineWidth',1.2)
%  plots red margin line
plot(lm2(1,:),lm2(2,:),'-r','LineWidth',1.2)
hold off
```

```
end
```

Actually, implementing the above function is not entirely trivial. It is necessary to write a function `makeline` to plot the line segment which is part of the line of equation $w_1x + w_2y = b$ inside a box containing the data points. We leave the details an exercises.

B.2 Soft Margin SVM (SVM$_{s2'}$)

The following `Matlab` programs implement the method described in Section 18.8.

The function `doSVMs2pbv3` calls the function `solve1` given in Section 16.7.

```
function [lamb,mu,alpha,beta,lambnz,munz,numsvl1,numsvm1,badnu,
w,nw,b,eta] = doSVMs2pbv3(nu,rho,u,v,K)
%
%   Best version
%   Uses the duality gap to compute eta
%   In principle, needs a single support vector of type 1
%
%   Soft margin nu-SVM version s2'
%   with the constraint
%   \sum_{i = 1}^p + \sum_{j = 1}^q mu_j  = K_m
%   (without the variable gamma)
%
%   p green vectors u_1, ..., u_p in n x p array u
%   q red   vectors v_1, ..., v_q in n x q array v
%
%   First builds the matrices for the dual program
%    K is a scale factor
%
p = size(u,2); q = size(v,2); n = size(u,1);
[A,c,X,Pa,qa] = buildSVMs2pb(nu,u,v,K);
%
%  Runs quadratic solver
%
tolr = 10^(-10); tols = 10^(-10); iternum = 80000;
[x,U,nr,ns,kk] = qsolve1(Pa, qa, A, c, rho, tolr, tols, iternum);
```

```
fprintf('nr =  %d ',nr)
fprintf('   ns =  %d \n',ns)
fprintf('kk =  %d \n',kk)
noconv = 0;
if kk > iternum
   noconv = 1;
   fprintf('** qsolve did not converge. Problem
           not solvable ** \n')
end
lam = x(1:(p+q),1);
alpha = x((p+q+1):2*p+q,1);
beta = x(2*p+q+1:2*(p+q),1);
w = -X*lam;
nw = sqrt(w'*w);   % norm of w
fprintf('nw =  %d \n',nw)
%
lamb = x(1:p,1);
mu = x(p+1:p+q,1);
tols = 10^(-10); tolh = 10^(-9);
% tols < lambda_i < K - tolh
[lambnz,numsvl1] = findpsv2(lamb,K,tols,tolh);
% tols < mu_i < K - tolh
[munz,numsvm1] = findpsv2(mu,K,tols,tolh);
fprintf('numsvl1 = %d ',numsvl1)
fprintf('   numsvm1 = %d \n',numsvm1)
%  lambda_i >= K - tolh
% number of blue margin failures
[lamK,pf] = countumf2(lamb,K,tolh);
%  mu_j >= K - tolh
% number of red margin failures
[muK,qf] = countvmf2(mu,K,tolh);
fprintf('pf =  %d ',pf)
fprintf('   qf =  %d \n',qf)
% number of  points such that lambda_i > tols
[~,pm] = countmlu2(lamb,tols);
% number of  points such that mu_i > tols
[~,qm] = countmlv2(mu,tols);
fprintf('pm =  %d ',pm)
fprintf('   qm =  %d \n',qm)
fprintf('p - pm =  %d ',p - pm)
fprintf('   q - qm =  %d \n',q - qm)
lnu = max(2*pf/(p+q),2*qf/(p+q));
unu = min(2*pm/(p+q),2*qm/(p+q));
fprintf('lnu =  %d ',lnu)
fprintf('    unu =  %d \n',unu)
if nu < lnu
      fprintf('** Warning; nu is too small ** \n')
else
      if nu > unu
        fprintf('** Warning; nu is too big ** \n')
      end
end

sx1 = zeros(n,1);  num1 = 0;
sKu = zeros(n,1);  Knum1 = 0;
for i = 1:p
       if lambnz(i) > 0
          sx1 = sx1 + u(:,i);
          num1 = num1 + 1;
       end
       if lamK(i) > 0
          sKu = sKu + u(:,i);
          Knum1 = Knum1 + 1;
       end
end
% Knum1
sx2 = zeros(n,1);  num2 = 0;
sKv = zeros(n,1);  Knum2 = 0;
for j = 1:q
       if munz(j) > 0
          sx2 = sx2 + v(:,j);
          num2 = num2 + 1;
       end
       if muK(j) > 0
          sKv = sKv + v(:,j);
          Knum2 = Knum2 + 1;
       end
end
```

```
% Knum2

b = 0; eta = 0;
epsilon = 0; xi = 0;
P2 = X'*X;
badnu = 0;
if numsvl1 > 0
   if numsvm1 > 0
      b =  (w'*(sx1/num1 + sx2/num2))/2;
      fprintf('b =  %.15f \n',b)
      eta =  (w'*(sx1/num1 - sx2/num2))/2;
      fprintf('eta =  %.15f \n',eta)
   else
       errterm = w'*(sKv - sKu) + (pf - qf)*w'*(sx1/num1);
       Pterm = (1/K)*(lam'*P2*lam);
       denomqf = (p+q)*nu  -2*qf;
       fprintf('denomqf =  %.15f \n',denomqf)
       if denomqf > 0
          eta = (errterm + Pterm)/denomqf;
          fprintf('eta =  %.15f \n',eta)
          b = -eta + w'*sx1/num1;
       else
           badnu = 1;
           fprintf('** Warning: numsvl1 > 0,
                   numsvm1 = 0 and nu = 2*qf/(p+q) **  \n')
       end
   end
else
    if numsvm1 > 0
       errterm = w'*(sKv - sKu) + (pf - qf)*w'*(sx2/num2);
       Pterm = (1/K)*(lam'*P2*lam);
       denompf = (p+q)*nu  -2*pf;
       fprintf('denompf =  %.15f \n',denompf)
       if denompf > 0
          eta = (errterm + Pterm)/denompf;
          fprintf('eta =  %.15f \n',eta)
          b = eta + w'*sx2/num2;
       else
           badnu = 1;
           fprintf('** Warning: numsvm1 > 0,
                    numsvl1 = 0 and nu = 2*pf/(p+q) **  \n')
        end
     else
        fprintf('** Not enough support vectors ** \n')
     end
end

Km = (p+q)*nu*K;
fprintf('K  =  %.15f ',K)
fprintf('    (p+q)*nu*Ks/2 =  %.15f \n',Km/2)
fprintf('sum(lambda) =  %.15f ',sum(lamb))
fprintf('    sum(mu) =  %.15f \n',sum(mu))

if (numsvl1 > 0 || numsvm1 > 0) && badnu == 0
   if eta < 10^(-9)
      fprintf('** Warning, eta too small or negative ** \n')
      eta = 0;
   end
   delta = eta/nw;
   fprintf('delta =  %.15f \n',delta)
   tolxi = 10^(-10);
   % tols < lambda_i < K - tolh or K - tolh <= lambda_i
   %  and epsilon_i < tolxi
   [lamsv,psf,epsilon] = findsvl2(lamb,w,b,u,eta,K,tols,
    tolh,tolxi);
   % tols < mu_i < K - tolh or K - tolh <= mu_i
   % and xi_i < tolxi
   [musv,qsf,xi] = findsvm2(mu,w,b,v,eta,K,tols,tolh,tolxi);
   fprintf('psf =  %d ',psf)
   fprintf('   qsf =  %d \n',qsf)
   fprintf('pf - psf =  %d ',pf - psf)
   fprintf('   qf - qsf =  %d \n',qf - qsf)

   % computes eta from the duality gap
   errterm = w'*(sKv - sKu) + (pf - qf)*b;
   Pterm = (1/K)*(lam'*P2*lam);
   denom = (p+q)*nu - pf -qf;
   fprintf('denom =  %.15f \n',denom)
   if denom > 0
      eta1 = (errterm + Pterm)/denom;
      fprintf('eta1 =  %.15f \n',eta1)
   end
end
end
```

The constraint matrix and the matrices defining the quadratic program are constructed by the function `buildSVMs2pb`.

```
function [A,c,X,Pa,q] = buildSVMs2pb(nu,u,v,K)
%  builds the matrix of constraints A for
%  soft margin nu-SVM s2'
%  with the constraint
%  \sum_{i = 1}^p + \sum_{j = 1}^q mu_j  = K_m
%  (without the  variable gamma) and the right-hand side c
%  u: vector of p blue points (each an n-dim vector)
%  v: vector of q red points (each an n-dim vector)
%  builds the matrix X = [-u_1 ... -u_p v1 .... v_q]
%  and the matrix Pa as 2(p+q) matrix obtained
%  by augmenting X'*X with zeros
%  K is a scale factor (K = Ks)

p = size(u,2); q = size(v,2);
% Ks = 1/(p+q);
Ks = K; Km = (p+q)*K*nu;
A = [ones(1,p) -ones(1,q) zeros(1,p+q);
     ones(1,p) ones(1,q)  zeros(1,p+q) ;
     eye(p) zeros(p,q) eye(p) zeros(p,q);
     zeros(q,p) eye(q)  zeros(q,p) eye(q) ];
c = [0; Km; Ks*ones(p+q,1)];
X = [-u v];
XX = X'*X;
Pa = [XX zeros(p+q,p+q); zeros(p+q, 2*(p+q))];
q = zeros(2*(p+q),1);
end
```

The function `findpsv2` makes a vector of λ_i (and μ_j) corresponding to support vectors of type 1.

```
function [lampsv,num] = findpsv2(lambda,K,tols,tolh)
%
%   This function find the vector of
%   lambda_i's such that 0 < lambda_i < K
%   and the number of such  lambda_i.
%
% tols = 10^(-11);  %  the smaller this is, the larger
% the number of points on the margin
% tolh = 10^(-9);  %
m = size(lambda,1); lampsv = zeros(m,1);
num = 0;
for i = 1:m
    if lambda(i) > tols && lambda(i) < K - tolh
       lampsv(i) = lambda(i);
       num = num + 1;
    end
end
end
```

The function `countumf2` finds those λ_i such that $\lambda_i = K$.

```
function [lamK,mf] = countumf2(lambda,K,tolh)
%  Counts the number of margin failures, that is,
%  points u_i (in u) such that lambda_i = K

p = size(lambda,1);
mf = 0; lamK = zeros(p,1);
for i = 1:p
    if lambda(i) >= K - tolh
       mf = mf + 1;
       lamK(i) = lambda(i);
    end
end
end
```

Similarly, the function `countvmf2` finds those μ_j such that $\mu_j = K$.
The function `countmlu2` finds those λ_i such that $\lambda_i > 0$.

```
function [lambnz, mlu] = countmlu2(lambda,tols)
%  Counts the number of points u_i (in u)
%  such that lambda_i > 0 and returns a vector
%  of these lambda_i

% tols = 10^(-11);
p = size(lambda,1); lambnz = zeros(p,1);
mlu = 0;
for i = 1:p
    if lambda(i) > tols
        mlu = mlu + 1;
        lambnz(i) = lambda(i);
    end
end
end
```

Similarly, the function `countmlv2` finds those μ_j such that $\mu_j > 0$. The function `findsvl2` finds the λ_i corresponding to blue support vectors of type 1 and 2 and the error vector ϵ. The number of blue errors is *psf* (the u_i for which $\epsilon_i > 0$). Similarly the function `findsvm2` finds the μ_j corresponding to red support vectors of type 1 and 2 and the error vector ξ. The number of red errors is *qsf* (the v_j for which $\xi_j > 0$).

The main function `runSVMs2pbv3` calls `doSVMs2pbv3` and displays the separating line (or plane) and the two margin lines (or planes).

```
function [lamb,mu,alpha,beta,lambnz,munz,w]
         = runSVMs2pbv3(nu,rho,u,v,K)
%
%   Best version
%   Uses the duality gap to compute eta
%   In principle, needs a single support vector of type 1
%
%   Runs soft margin nu-SVM version s2'
%   with the constraint
%   \sum_{i = 1}^p + \sum_{j = 1}^q mu_j  = K_m
%   (without the variable gamma)
%
%   p green vectors u_1, ..., u_p in n x p array u
%   q red   vectors v_1, ..., v_q in n x q array v
%
%   First builds the matrices for the dual program
%    K is a scale factor
%
p = size(u,2); q = size(v,2); n = size(u,1);

[lamb,mu,alpha,beta,lambnz,munz,numsvl1,numsvm1,badnu,w,
          nw,b,eta] = doSVMs2pbv3(nu,rho,u,v,K);

   if n == 2
       [ll,mm] = showdata(u,v);
       if (numsvl1 > 0 || numsvm1 > 0) && badnu == 0
          showSVMs2(w,b,eta,ll,mm,nw)
       end
   else
      if n == 3
         showpointsSVM(u,v)
         if (numsvl1 > 0 || numsvm1 > 0) && badnu == 0
            offset = 10;
            C1 = [1 0 1];  % magenta
            plotplaneSVM(u,v,w,b,offset,C1)
            C2 = [0 0 1];  % blue
            plotplaneSVM(u,v,w,b+eta,offset,C2)
            C3 = [1,0,0];  % red
            plotplaneSVM(u,v,w,b-eta,offset,C3)
         end
         axis equal
         view([-1 -1 1]);
         xlabel('X','fontsize',14);ylabel('Y','fontsize',14);
               zlabel('Z','fontsize',14);
         hold off
      end
   end
end
```

B.3 Soft Margin SVM (SVM$_{s3}$)

The following `Matlab` programs implement the method described in Section 18.12. The main function `doSVMs3b` is given below.

```
function [lamb,mu,alpha,beta,lambnz,munz,lamK,muK,w,b,eta,
          nw,fail] = doSVMs3b (nu,rho,u,v,K)
%
%  Soft margin nu-SVM version s3
%
%  Computes eta using the duality gap
%  Needs a single support vector of type 1
%
%   p green vectors u_1, ..., u_p in n x p array u
%   q red   vectors v_1, ..., v_q in n x q array v
%
%   First builds the matrices for the dual program
%    K is a scale factor
%
p = size(u,2); q = size(v,2); n = size(u,1);
[A,c,X,P2,Pa,qa] = buildSVMs3b (nu,u,v,K);
%
%  Runs quadratic solver
%
tolr = 10^(-10); tols = 10^(-10); iternum = 80000;
[x,U,nr,ns,kk] = qsolve1(Pa, qa, A, c, rho, tolr, tols,
                         iternum);
fprintf('nr =  %d ',nr)
fprintf('    ns =  %d ',ns)
fprintf('    kk =  %d \n',kk)
noconv = 0;
if kk > iternum
   noconv = 1;
   fprintf('** qsolve did not converge. Problem
            not solvable ** \n')
end
lam = x(1:(p+q),1);
alpha = x((p+q+1):2*p+q,1);
beta = x(2*p+q+1:2*(p+q),1);
w = -X*lam;
nw = sqrt(w'*w);   % norm of w
fprintf('nw =  %d \n',nw)
lamb = x(1:p,1);
mu = x(p+1:p+q,1);
b = -(sum(lamb) - sum(mu));
```

```
fprintf('b =  %.15f \n',b)
%

tols = 10^(-10); tolh = 10^(-9);
% tols < lambda_i < K - tolh
[lambnz,numsvl1] = findpsv2(lamb,K,tols,tolh);
% tols < mu_i < K - tolh
[munz,numsvm1] = findpsv2(mu,K,tols,tolh);
fprintf('numsvl1 = %d ',numsvl1)
fprintf('    numsvm1 = %d \n',numsvm1)
%  lambda_i >= K - tolh
% number of blue margin failures
[lamK,pf] = countumf2(lamb,K,tolh);
%  mu_j >= K - tolh
% number of red margin failures
[muK,qf] = countvmf2(mu,K,tolh);
fprintf('pf =  %d ',pf)
fprintf('   qf =  %d \n',qf)
% number of  points such that lambda_i > tols
[~,pm] = countmlu2(lamb,tols);
% number of  points such that mu_i > tols
[~,qm] = countmlv2(mu,tols);
fprintf('pm =  %d ',pm)
fprintf('   qm =  %d \n',qm)
fprintf('p - pm =  %d ',p - pm)
fprintf('   q - qm =  %d \n',q - qm)
lnu =  (pf + qf)/(p+q); unu = (pm + qm)/(p+q);
fprintf('lnu =  %d ',lnu)
   fprintf('    unu =  %d \n',unu)
if nu < lnu
      fprintf('** Warning; nu is too small ** \n')
else
      if nu > unu
         fprintf('** Warning; nu is too big ** \n')
      end
end

sx1 = zeros(n,1);  num1 = 0;
sKu = zeros(n,1);  Knum1 = 0;
```

```
for i = 1:p
       if lambnz(i) > 0
          sx1 = sx1 + u(:,i);
          num1 = num1 + 1;
       end
       if lamK(i) > 0
          sKu = sKu + u(:,i);
          Knum1 = Knum1 + 1;
       end
end
% Knum1
sx2 = zeros(n,1);  num2 = 0;
sKv = zeros(n,1);  Knum2 = 0;
for j = 1:q
       if munz(j) > 0
          sx2 = sx2 + v(:,j);
          num2 = num2 + 1;
       end
       if muK(j) > 0
          sKv = sKv + v(:,j);
          Knum2 = Knum2 + 1;
       end
end
% Knum2
% computes eta from the duality gap
errterm = w'*(sKv - sKu) + (pf - qf)*b;
Pterm = (1/K)*(lam'*P2*lam);
denom = (p+q)*nu - pf -qf;
fprintf('denom =  %.15f \n',denom)
epsilon = 0; xi = 0;
if denom > 0
   eta = (errterm + Pterm)/denom;
   fprintf('eta =  %.15f \n',eta)
   if eta < 10^(-10)
      fprintf('** Warning; eta is too small
               or negative ** \n')
   end

   tolxi = 10^(-10);
   % tols < lambda_i < K - tolh or K - tolh <= lambda_i
   % and epsilon_i < tolxi
   [lamsv,psf,epsilon] = findsvl2(lamb,w,b,u,eta,K,tols,
                                  tolh,tolxi);
    % tols < mu_i < K - tolh or K - tolh <= mu_i
    % and xi_i < tolxi
   [musv,qsf,xi] = findsvm2(mu,w,b,v,eta,K,tols,tolh,tolxi);
   fprintf('psf =  %d ',psf)
   fprintf('   qsf =  %d \n',qsf)
   fprintf('pf - psf =  %d ',pf - psf)
   fprintf('   qf - qsf =  %d \n',qf - qsf)
else
    eta = 0;
    denom = 0;
    fprintf('** Warning, nu = (pf + qf)/(p+q) **  \n')
end

Km = (p+q)*nu*K;
fprintf('K  =  %.15f ',K)
fprintf('    (p+q)*nu*Ks =  %.15f \n',Km)
fprintf('sum(lambda) + sum(mu)=  %.15f \n',sum(lamb) + sum(mu))

eta1 = 0;
if numsvl1 > 0 || numsvm1 > 0
   if numsvl1 > numsvm1
      eta1 =  w'*sx1/num1 - b;
   else
      eta1 =  b - w'*sx2/num2;
   end
   fprintf('eta1 =  %.15f \n',eta1)
else
      fprintf('** Warning: not enough support vectors ** \n')
end
if denom == 0
   if numsvl1 > 0 || numsvm1 > 0
      eta = eta1;
      fail = 0;
   else
      fail = 1;
      fprintf('** Warning, denom = 0 and
              not enough support vectors **  \n')
   end
else
   fail = 0;
end
end
```

The main function `doSVMs3b` is executed by the following function:

```
function [lamb,mu,alpha,beta,lambnz,munz,w]
          = runSVMs3b(nu,rho,u,v,K)
%
%   Runs soft margin nu-SVM version s3
%
%  Computes eta using the duality gap
%  Needs a single support vector of type 1
%
%   p green vectors u_1, ..., u_p in n x p array u
%   q red   vectors v_1, ..., v_q in n x q array v
%
%   First builds the matrices for the dual program
%    K is a scale factor
%
p = size(u,2); q = size(v,2); n = size(u,1);

[lamb,mu,alpha,beta,lambnz,munz,lamK,muK,w,b,eta,nw,fail]
   = doSVMs3b(nu,rho,u,v,K);

   if n == 2
       [ll,mm] = showdata(u,v);
       if fail == 0
          showSVMs2(w,b,eta,ll,mm,nw)
       end
   else
      if n == 3
         showpointsSVM(u,v)
         if fail == 0
            offset = 10;
            C1 = [1 0 1];  % magenta
            plotplaneSVM(u,v,w,b,offset,C1)
            C2 = [0 0 1];  % blue
            plotplaneSVM(u,v,w,b+eta,offset,C2)
            C3 = [1,0,0];  % red
            plotplaneSVM(u,v,w,b-eta,offset,C3)
         end
         axis equal
  %      axis([ll(1) mm(1) ll(2) mm(2)]);
         view([-1 -1 1]);
         xlabel('X','fontsize',14);ylabel('Y','fontsize',14);
          zlabel('Z','fontsize',14);
         hold off
      end
   end
end
```

The function buildSVMs3b builds the constraint matrix and the matrices defining the quadratic program.

```
function [A,c,X,P2,Pa,q] = buildSVMs3b(nu,u,v,K)
%  builds the matrix of constraints A for
%  soft margin nu-SVM s3 and the right-hand side c
%  u: vector of p blue points (each an n-dim vector)
%  v: vector of q red points (each an n-dim vector)
%  builds the matrix X = [-u_1 ... -u_p v1 .... v_q]
%  and the matrix Xa as 2(p+q)  matrix obtained
%  by augmenting X'*X with zeros
%  K is a scale factor (K = Ks)

p = size(u,2); q = size(v,2);
% Ks = 1/(p+q);
Ks = K; Km = (p+q)*K*nu;
A = [ones(1,p) ones(1,q)  zeros(1,p+q) ;
     eye(p) zeros(p,q) eye(p) zeros(p,q);
     zeros(q,p) eye(q)  zeros(q,p) eye(q) ];
c = [Km; Ks*ones(p+q,1)];
X = [-u v];
XX1 = X'*X;
XX2 = [ones(p,1)*ones(p,1)' -ones(p,1)*ones(q,1)';
       -ones(q,1)*ones(p,1)' ones(q,1)*ones(q,1)'];
```

```
P2 = XX1 + XX2;
Pa = [P2 zeros(p+q,p+q); zeros(p+q, 2*(p+q))];
q = zeros(2*(p+q),1);
end
```

B.4 ν-SV Regression

g The main function `donuregb` is given below.

```
function
[lamb,mu,alpha,beta,lambnz,munz,lamK,muK,numsvl1,numsvm1,w,
        epsilon,b] = donuregb (rho,nu,X,y,C)
%
%   Soft margin nu-regression
%   with the constraint
%   \sum_{i = 1}^m + \sum_{j = 1}^m mu_j = C nu
%   (Without the variable gamma)
%
%  Input: an m x n matrix of data points represented as
%  as the rows of X, and y a vector in R^n
%
%   First builds the matrices for the dual program
%    C is a scale factor
%
m = size(X,1); n = size(X,2);
[A,c,P,Pa,qa] = buildnuregb(nu,X,y,C);
%
%  Runs quadratic solver
%
tolr = 10^(-10); tols = 10^(-10); iternum = 80000;
[x,U,nr,ns,kk] = qsolve1(Pa, qa, A, c, rho, tolr, tols,
                         iternum);
% fprintf('nr =  %d ',nr)
% fprintf('   ns =  %d \n',ns)
fprintf('nr =  %d',nr)
fprintf('  ns =  %d',ns)
fprintf('  kk =  %d \n',kk)
noconv = 0;
if kk > iternum
```

```
   noconv = 1;
   fprintf('** qsolve did not converge. Problem
           not solvable ** \n')
end
lamb = x(1:m,1);
mu = x(m+1:2*m,1);
alpha = x((2*m+1):3*m,1);
beta = x(3*m+1:4*m,1);
w = X'*(mu - lamb);
%
b = 0; epsilon = 0;
tols = 10^(-10); tolh = 10^(-9);
% tols < lambda_i < C/m - tolh
[lambnz,numsvl1] = findpsv2(lamb,C/m,tols,tolh);
% tols < mu_i < C/m - tolh
[munz,numsvm1] = findpsv2(mu,C/m,tols,tolh);
fprintf('numsvl1 = %d',numsvl1)
fprintf('   numsvm1 = %d \n',numsvm1)
%  lambda_i >= C/m - tolh
% number of blue margin failures
[lamK,pf] = countumf2(lamb,C/m,tolh);
%  mu_j >= C/m - tolh
% number of red margin failures
[muK,qf] = countvmf2(mu,C/m,tolh);
fprintf('pf =  %d',pf)
fprintf('    qf =  %d \n',qf)
% number of  points such that lambda_i > tols
[~,pm] = countmlu2(lamb,tols);
% number of  points such that mu_i > tols
[~,qm] = countmlv2(mu,tols);
fprintf('pm =  %d',pm)
fprintf('    qm =  %d \n',qm)
%  lambda_i <= tols
[lmz,nz] = countLzero(lamb,mu,tols);
pm2 = numsvl1 + pf; qm2 = numsvm1 + qf;
fprintf('pm2 =  %d',pm2)
fprintf('   qm2 =  %d \n',qm2)
lnu = max(2*pf/m,2*qf/m); unu = min(2*pm/m,2*qm/m);
fprintf('lnu =  %d',lnu)
fprintf('  unu =  %d \n',unu)
fprintf('nz =  %d \n',nz)
if nu < lnu
      fprintf('** Warning; nu is too small ** \n')
else
      if nu > unu
           fprintf('** Warning; nu is too big ** \n')
      end
end

fprintf('C/m  =  %.15f ',C/m)
fprintf('    (C nu)/2 =  %.15f \n',(C*nu)/2)
fprintf('sum(lambda) =  %.15f ',sum(lamb))
fprintf('    sum(mu) =  %.15f \n',sum(mu))
lamsv = 0; musv = 0; xi = 0; xip = 0;
if numsvl1 > 0 && numsvm1 > 0
   sx1 = zeros(n,1); sy1 = 0; num1 = 0;
   sx2 = zeros(n,1); sy2 = 0; num2 = 0;
   for i = 1:m
       if lambnz(i) > 0
          sx1 = sx1 + X(i,:)'; sy1 = sy1 + y(i);
          num1 = num1 + 1;
       end
       if munz(i) > 0
          sx2 = sx2 + X(i,:)'; sy2 = sy2 + y(i);
          num2 = num2 + 1;
       end
   end
   % num1
   % num2
   b =  (sy1/num1 + sy2/num2 - w'*(sx1/num1 + sx2/num2))/2;
   fprintf('b =  %.15f \n',b)
   epsilon =  (w'*(sx1/num1 - sx2/num2)
              + sy2/num2 - sy1/num1)/2;
   fprintf('epsilon =  %.15f \n',epsilon)
   if epsilon < 10^(-10)
      fprintf('** Warning; epsilon is too small
              or negative ** \n')
   end
   nw = sqrt(w'*w);   % norm of w
   fprintf('nw =  %.15f \n',nw)
   %
   tolxi = 10^(-10);
   % tols < lambda_i < C/m - tolh or C/m - tolh <= lambda_i
   % and xi_i < tolxi
   [lamsv,psf,xi] = findnuregsvl2(lamb,w,b,X,y,epsilon,C/m,
                    tols,tolh,tolxi);
    % tols < mu_i < C/m - tolh or C/m - tolh <= mu_i
   % and xi_i' < tolxi
   [musv,qsf,xip] = findnuregsvm2(mu,w,b,X,y,epsilon,C/m,tols,
                    tolh,tolxi);
   fprintf('psf =  %d ',psf)
   fprintf('   qsf =  %d \n',qsf)
else
   fprintf('** Not enough support vectors ** \n')
end
end
```

To run `donuregb` use the function `runuregb` listed below.

```
function [lamb,mu,alpha,beta,lambnz,munz,lamK,muK,w]
         = runuregb (rho,nu,X,y,C)
%
%   Runs soft margin nu-regression
%   with the constraint
%   \sum_{i = 1}^m + \sum_{j = 1}^m mu_j = C nu
%  (Without the variable gamma)
%
%  Input: an m x n matrix of data points represented as
%  as the rows of X, and y a vector in R^n
%
%   First builds the matrices for the dual program
%    C is a scale factor
%

m = size(X,1); n = size(X,2);
[lamb,mu,alpha,beta,lambnz,munz,lamK,muK,numsvl1,numsvm1,w,
           epsilon,b] = donuregb(rho,nu,X,y,C);
```

```
   if n == 1
      [ll,mm] = showgraph(X,y);
      ww = [w;-1]; n1 = sqrt(ww'*ww);
      if numsvl1 > 0 && numsvm1 > 0
         showSVMs2(ww,-b,epsilon,ll,mm,n1)
      end
   else
       if n == 2
          offset = 10;
          [ll,mm] = showpoints(X,y,offset);
          if numsvl1 > 0 && numsvm1 > 0
             showplanes(w,b,ll,mm,epsilon)
          end
          axis equal
          axis([ll(1) mm(1) ll(2) mm(2)]);
          view([-1 -1 1]);
          xlabel('X','fontsize',14);ylabel('Y','fontsize',14);
          zlabel('Z','fontsize',14);
       end
   end
end
```

The function buildnuregb creates the constraint matrix and the matrices defining the quadratic functional.

```
function [A,c,P,Pa,qa] = buildnuregb (nu,X,y,C)
%  builds the matrix of constraints A for
%  soft margin nu-regression
%   with the constraint
%   \sum_{i = 1}^m + \sum_{j = 1}^m mu_j = C nu
%  (without the  variable gamma)
%  and the right-hand side c.
%  Input: an m x n matrix X of data points represented as
%  as the rows of X, and y a vector in R^n.
%  builds the m x m matrix X*X^T, the 2m x 2m matrix
%  P = [X*X^T -X*X^T;  -X*X^T X*X^T],
%  and the matrix Pa as the 4m  x 4m   matrix obtained
%  by augmenting  with zeros.
%  Also builds the vector q_a (q augmented with zeros).
%  C is a scale factor.

m = size(X,1); n = size(X,2);
% Ks = 1/(p+q);
Ks = C; Km = C*nu;
A = [ones(1,m) -ones(1,m) zeros(1,2*m);
     ones(1,m) ones(1,m)  zeros(1,2*m) ;
     eye(m) zeros(m,m) eye(m) zeros(m,m);
     zeros(m,m) eye(m)  zeros(m,m) eye(m)];
c = [0; Km; (Ks/m)*ones(2*m,1)];
XX1 = X*X';
P = [XX1 -XX1; -XX1 XX1];
Pa = [P zeros(2*m,2*m); zeros(2*m, 4*m )];
qa = [y; -y; zeros(2*m,1)];
end
```

Bibliography

Abraham, R. and Marsden, J. E. (1978). *Foundations of Mechanics*, 2nd edn. (Addison Wesley).

Apostol, T. (1974). *Analysis*, 2nd edn. (Addison Wesley).

Arnold, V. (1989). *Mathematical Methods of Classical Mechanics*, 2nd edn., GTM No. 102 (Springer Verlag).

Artin, M. (1991). *Algebra*, 1st edn. (Prentice Hall).

Avez, A. (1991). *Calcul Différentiel*, 1st edn. (Masson).

Berger, M. (1990a). *Géométrie 1* (Nathan), english edition: Geometry 1, Universitext, Springer Verlag.

Berger, M. (1990b). *Géométrie 2* (Nathan), english edition: Geometry 2, Universitext, Springer Verlag.

Berger, M. and Gostiaux, B. (1992). *Géométrie différentielle: variétés, courbes et surfaces*, 2nd edn., Collection Mathématiques (Puf), english edition: Differential geometry, manifolds, curves, and surfaces, GTM No. 115, Springer Verlag.

Bertsekas, D. P. (2009). *Convex Optimization Theory*, 1st edn. (Athena Scientific).

Bertsekas, D. P. (2015). *Convex Optimization Algorithms*, 1st edn. (Athena Scientific).

Bertsekas, D. P. (2016). *Nonlinear Programming*, 3rd edn. (Athena Scientific).

Bertsekas, D. P., Nedić, A., and Ozdaglar, A. E. (2003). *Convex Analysis and Optimization*, 1st edn. (Athena Scientific).

Bertsekas, D. P. and Tsitsiklis, J. N. (1997). *Parallel and Distributed Computation: Numerical Methods*, 1st edn. (Athena Scientific).

Bertsimas, D. and Tsitsiklis, J. N. (1997). *Introduction to Linear Optimization*, 3rd edn. (Athena Scientific).

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*, 1st edn., Information Science and Statistics (Springer).

Bourbaki,
N. (1981). *Espaces Vectoriels Topologiques*, Eléments de Mathématiques (Masson).

Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2010). Distributed optimization and statistical learning via the alternating direction method of multiplier, *Foundations and Trends in Machine Learning* **3(1)**, pp. 1–122.

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*, 1st edn. (Cambridge University Press).

Bredon, G. E. (1993). *Topology and Geometry*, 1st edn., GTM No. 139 (Springer Verlag).

Brezis, H. (2011). *Functional Analysis, Sobolev Spaces and Partial Differential Equations*, 1st edn., Universitext (Springer-Verlag).

Cartan, H. (1990). *Cours de Calcul Différentiel*, Collection Méthodes (Hermann).

Chang, C.-C. and Chih-Jen, L. (2001). Training ν-support vector classifiers: Theory and algorithms, *Neural Computation* **13**, pp. 2119–2147.

Choquet-Bruhat, Y., DeWitt-Morette, C., and Dillard-Bleick, M. (1982). *Analysis, Manifolds, and Physics, Part I: Basics*, 1st edn. (North-Holland).

Chvatal, V. (1983). *Linear Programming*, 1st edn. (W.H. Freeman).

Ciarlet, P. (1989). *Introduction to Numerical Matrix Analysis and Optimization*, 1st edn. (Cambridge University Press), french edition: Masson, 1994.

Cour, T. and Shi, J. (2007). Solving markov random fields with spectral relaxation, in M. Meila and X. Shen (eds.), *Artifical Intelligence and Statistics* (Society for Artificial Intelligence and Statistics).

Demmel, J. W. (1997). *Applied Numerical Linear Algebra*, 1st edn. (SIAM Publications).

Dixmier, J. (1984). *General Topology*, 1st edn., UTM (Springer Verlag).

do Carmo, M. P. (1976). *Differential Geometry of Curves and Surfaces* (Prentice Hall).

do Carmo, M. P. (1992). *Riemannian Geometry*, 2nd edn. (Birkhäuser).

Faugeras, O. (1996). *Three-Dimensional Computer Vision, A geometric Viewpoint*, 1st edn. (the MIT Press).

Foley, J., van Dam, A., Feiner, S., and Hughes, J. (1993). *Computer Graphics. Principles and Practice*, 2nd edn. (Addison-Wesley).

Gabay, D. (1983). Applications of the method of multipliers to variational inequalities, *Studies in Mathematics and Applications* **15(C)**, pp. 299–331.

Gallier, J. H. (2011). *Geometric Methods and Applications, For Computer Science and Engineering*, 2nd edn., TAM, Vol. 38 (Springer).

Gallier, J. H. (2016). Notes on Convex Sets, Polytopes, Polyhedra, Combinatorial Topology, Voronoi Diagrams, and Delaunay Triangulations, Tech. rep., University of Pennsylvania, CIS Department, Philadelphia, PA 19104, book in Preparation.

Gander, W., Golub, G. H., and von Matt, U. (1989). A constrained eigenvalue problem, *Linear Algebra and its Applications* **114/115**, pp. 815–839.

Golub, G. H. (1973). Some modified eigenvalue problems, *SIAM Review* **15(2)**, pp. 318–334.

Gray, A. (1997). *Modern Differential Geometry of Curves and Surfaces*, 2nd edn. (CRC Press).

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd edn. (Springer).

Hastie, T., Tibshirani, R., and Wainwright, M. (2015). *Statistical Learning with Sparsity. The Lasso and Generalizations*, 1st edn. (CRC Press).

Helgason, S. (2000). *Groups and Geometric Analysis. Integral Geometry, Invariant Differential Operators and Spherical Functions*, 1st edn., MSM, Vol. 83 (AMS).

Higham, N. J. (2008). *Functions of Matrices. Theory and Computation*, 1st edn. (SIAM).

Horn, R. A. and Johnson, C. R. (1990). *Matrix Analysis*, 1st edn. (Cambridge University Press).

Jain, R., Katsuri, R., and Schunck, B. G. (1995). *Machine Vision*, 1st edn. (McGraw-Hill).

Kolmogorov, A. and Fomin, S. (1975). *Introductory Real Analysis*, 1st edn. (Dover).

Kreyszig, E. (1991). *Differential Geometry*, 1st edn. (Dover).

Lang, S. (1993). *Algebra*, 3rd edn. (Addison Wesley).

Lang, S. (1995). *Differential and Riemannian Manifolds*, 3rd edn., GTM No. 160 (Springer Verlag).

Lang, S. (1996). *Real and Functional Analysis*, 3rd edn., GTM 142 (Springer Verlag).

Lang, S. (1997). *Undergraduate Analysis*, 2nd edn., UTM (Springer Verlag).

Lax, P. (2007). *Linear Algebra and Its Applications*, 2nd edn. (Wiley).

Luenberger, D. G. (1997). *Optimization by Vector Space Methods*, 1st edn. (Wiley).

Luenberger, D. G. and Ye, Y. (2016). *Linear and Nonlinear Programming*, 4th edn. (Verlag).

Matousek, J. and Gartner, B. (2007). *Understanding and Using Linear Programming*, 1st edn., Universitext (Springer Verlag).

Metaxas, D. N. (1997). *Physics-Based Deformable Models*, 1st edn. (Kluwer Academic Publishers).

Milnor, J. W. (1969). *Topology from the Differentiable Viewpoint*, 2nd edn. (The University Press of Virginia).

Molla, T. (2015). Class notes, math 588 example 5, Tech. rep., http://myweb.usf.edu/molla/2015_spring_math588/example5.pdf.

Munkres, J. R. (1991). *Analysis on Manifolds* (Addison Wesley).

Munkres, J. R. (2000). *Topology*, 2nd edn. (Prentice Hall).

Papadimitriou, C. H. and Steiglitz, K. (1998). *Combinatorial Optimization. Algorithms and Complexity*, 1st edn. (Dover).

Rockafellar, R. T. (1970). *Convex Analysis*, Princeton Landmarks in Mathematics (Princeton University Press).

Rudin, W. (1987). *Real and Complex Analysis*, 3rd edn. (McGraw Hill).

Rudin, W. (1991). *Functional Analysis*, 2nd edn. (McGraw Hill).

Schölkopf, B., Platt, J. C., Shawe-Taylor, J., and Smola, A. J. (2001). Estimating the support of a high-dimensional distribution, *Neural Computation* **13**, pp. 1443–1471.

Schölkopf, B. and Smola, A. J. (2002). *Learning with Kernels*, 1st edn. (MIT Press).

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

Schrijver, A. (1999). *Theory of Linear and Integer Programming*, 1st edn. (Wiley).

Schwartz, L. (1980). *Topologie Générale et Analyse Fonctionnelle*, Collection Enseignement des Sciences (Hermann).

Schwartz, L. (1991). *Analyse I. Théorie des Ensembles et Topologie*, Collection Enseignement des Sciences (Hermann).

Schwartz, L. (1992). *Analyse II. Calcul Différentiel et Equations Différentielles*, Collection Enseignement des Sciences (Hermann).

Schwartz, L. (1993a). *Analyse III. Calcul Intégral*, Collection Enseignement des Sciences (Hermann).

Schwartz, L. (1993b). *Analyse IV. Applications à la Théorie de la Mesure*, Collection Enseignement des Sciences (Hermann).

Seifert, H. and Threlfall, W. (1980). *A Textbook of Topology*, 1st edn. (Academic Press).

Shawe-Taylor, J. and Cristianini, N. (2004). *Kernel methods for Pattern Analysis*, 1st edn. (Cambridge University Press).

Stoker, J. (1989). *Differential Geometry*, 1st edn., Wiley Classics (Wiley-Interscience).

Strang, G. (1986). *Introduction to Applied Mathematics*, 1st edn. (Wellesley-Cambridge Press).

Strang, G. (2019). *Linear Algebra and Learning from Data*, 1st edn. (Wellesley-Cambridge Press).

Trefethen, L. and Bau III, D. (1997). *Numerical Linear Algebra*, 1st edn. (SIAM Publications).

Trucco, E. and Verri, A. (1998). *Introductory Techniques for 3D Computer Vision*, 1st edn. (Prentice-Hall).

Vanderbei, R. J. (2014). *Linear Programming: Foundations and Extensions*, 4th edn. (Springer).

Vapnik, V. N. (1998). *Statistical Learning Theory*, 1st edn. (Wiley).

Warner, F. (1983). *Foundations of Differentiable Manifolds and Lie Groups*, 1st edn., GTM No. 94 (Springer Verlag).

Yu, S. X. and Shi, J. (2001). Grouping with bias, in T. G. Dietterich, S. Becker, and Z. Ghahramani (eds.), *Neural Information Processing Systems, Vancouver, Canada, 3-8 Dec. 2001* (MIT Press).

Ziegler, G. (1997). *Lectures on Polytopes*, 1st edn., GTM No. 152 (Springer Verlag).

Index