A Study on Sigmoid Kernels for SVM and the Training of non-PSD Kernels by SMO-type Methods

Hsuan-Tien Lin and Chih-Jen Lin
Department of Computer Science and
Information Engineering
National Taiwan University
Taipei 106, Taiwan
{b6506054, cjlin}@csie.ntu.edu.tw

Abstract

The sigmoid kernel was quite popular for support vector machines due to its origin from neural networks. However, as the kernel matrix may not be positive semidefinite (PSD), it is not widely used and the behavior is unknown. In this paper, we analyze such non-PSD kernels through the point of view of separability. Based on the investigation of parameters in different ranges, we show that for some parameters, the kernel matrix is conditionally positive definite (CPD), a property which explains its practical viability. Experiments are given to illustrate our analysis. Finally, we discuss how to solve the non-convex dual problems by SMO-type decomposition methods. Suitable modifications for any symmetric non-PSD kernel matrices are proposed with convergence proofs.

1 Introduction

Given training vectors $x_i \in R^n, i = 1, \dots, l$ in two classes, labeled by the vector $y \in R^l$ such that $y_i \in \{1, -1\}$, the support vector machine (SVM) (Boser, Guyon, and Vapnik 1992; Cortes and Vapnik 1995) tries to separate the training vectors in a ϕ-mapped (and possibly infinite dimensional) space, with an error cost $C > 0$:

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2}w^T w + C\sum_{i=1}^{l}\xi_i \\ \text{subject to} \quad & y_i(w^T\phi(x_i)+b) \geq 1-\xi_i, \\ & \xi_i \geq 0, i = 1, \dots, l. \end{aligned} \tag{1.1}$$

Due to the high dimensionality of the vector variable w, we usually solve (1.1) through its Lagrangian dual problem:

$$\begin{aligned} \min_{\alpha} \quad & F(\alpha) = \frac{1}{2}\alpha^T Q \alpha - e^T \alpha \\ \text{subject to} \quad & 0 \leq \alpha_i \leq C, i = 1, \ldots, l, \\ & y^T \alpha = 0, \end{aligned} \tag{1.2}$$

where $Q_{ij} \equiv y_i y_j \phi(x_i)^T \phi(x_j)$ and e is the vector of all ones. Here,

$$K(x_i, x_j) \equiv \phi(x_i)^T \phi(x_j) \tag{1.3}$$

is called the kernel function where some popular ones are, for example, the polynomial kernel $K(x_i, x_j) = (ax_i^T x_j + r)^d$, and the RBF (Gaussian) kernel $K(x_i, x_j) = e^{-\gamma\|x_i - x_j\|^2}$. We can see that by the definition (1.3), the matrix Q is symmetric and positive semi-definite (PSD). After (1.2) is solved, $w = \sum_{i=1}^{l} y_i \alpha_i \phi(x_i)$ so the decision function for any test vector x is

$$\text{sgn}(\sum_{i=1}^{l} \alpha_i y_i K(x_i, x) + b), \tag{1.4}$$

where b is calculated through the primal-dual relationship.

In practice, some non-PSD matrices are used in (1.2). An important one is the sigmoid kernel $K(x_i, x_j) = \tanh(ax_i^T x_j + r)$ which is related to neural networks. It was first pointed out in (Vapnik 1995) that its kernel matrix might not be PSD for certain values of the parameters a and r. More discussions are in, for instance, (Burges 1998; Schölkopf and Smola 2002). Without $K(x_i, x_j)$ being the inner product of two vectors, there is no problem (1.1) so it is unclear what kind of classification problems we are solving. Surprisingly, the sigmoid kernel has been successfully used in some practical cases. Some explanations are in (Schölkopf 1997).

Recently, quite a few kernels specific to different applications are proposed. However, similar to the sigmoid kernel, some of them are not PSD either (e.g. kernel jittering in (DeCoste and Schölkopf 2002) and tangent distance kernels in (Haasdonk and Keysers 2002)). Thus, it is essential to analyze such non-PSD kernels. In Section 2, we discuss them by considering the separability of training data. Then in Section 3, we explain the practical viability of the sigmoid kernel by showing that for parameters in certain ranges, it is conditionally positive definite (CPD). We also discuss in Section 4 that for some parameters, the sigmoid kernel behaves like the RBF kernel.

Section 5 presents experiments showing that the linear constraint $y^T\alpha = 0$ in the dual problem is essential for the sigmoid kernel matrix to work for SVM.

In addition to unknown behaviors, the non-PSD kernels also cause difficulties on solving the dual problem. Due to the high density of the Hessian matrix Q of (1.2), currently a special approach called the decomposition method is the major way to solve it. However, this method was designed for convex problems where Q is PSD. If the sigmoid or other non-PSD kernels are used, it may get into serious troubles. In Section 6, we propose modifications for SMO-type decomposition methods which guarantee the convergence to a local minimum for non-PSD kernels. Finally, some discussions are in Section 7.

2 The Separability of Using non-PSD Kernel Matrices

When using non-PSD kernels such as the sigmoid, $K(x_i, x_j)$ cannot be separated as the inner product form in (1.3). Thus, the relationship between (1.1) and (1.2) does not hold. In other words, after obtaining α from (1.2), it is not clear how the training data are classified since (1.1) may not exist any more. To analyze what we actually obtained when using a non-PSD Q, we consider a new problem:

$$\begin{aligned} \min_{\alpha,b,\xi} \quad & \frac{1}{2}\alpha^T Q\alpha + C\sum_{i=1}^{l}\xi_i \\ \text{subject to} \quad & Q\alpha + by \geq e - \xi, \\ & \xi_i \geq 0, i = 1,\ldots,l. \end{aligned} \tag{2.1}$$

It is from substituting $w = \sum_{i=1}^{l} y_i\alpha_i\phi(x_i)$ into (1.1) so that $w^T w = \alpha^T Q\alpha$ and $y_i w^T\phi(x_i) = (Q\alpha)_i$. Note that in (2.1), α_i may be negative. This problem was used in (Osuna and Girosi 1998) and some subsequent work. (Lin and Lin 2003) shows that if Q is symmetric PSD, the optimal solution α of the dual problem (1.2) is also optimal for (2.1). However, the opposite may not be true unless Q is symmetric positive definite (PD).

From now on, we consider that Q (or K) may not be PSD. However, we still assume that it is symmetric. The next theorem tries to address the relation between (1.2) and (2.1).

Theorem 1 *Any local optimal solution $\hat{\alpha}$ of (1.2) is a feasible point of (2.1).*

Proof.

As (1.2) is a linearly constrained problem, any local optimum $\hat{\alpha}$ of (1.2) satisfies the Karash-Kunh-Tucker (KKT) condition. As Q is symmetric, the KKT condition of (1.2) is that there are scalar p, and non-negative vectors λ and μ such that

$$\begin{aligned} Q\hat{\alpha} - e - \mu + \lambda - py &= 0, \\ \mu_i \geq 0, \mu_i \hat{\alpha}_i &= 0, \\ \lambda_i \geq 0, \lambda_i (C - \hat{\alpha}_i) &= 0, \quad i = 1, \ldots, l. \end{aligned}$$

If we consider $\alpha_i = \hat{\alpha}_i$, $b = -p$, and $\xi_i = \lambda_i$, then $\mu_i \geq 0$ implies that $(\hat{\alpha}, -p, \lambda)$ is feasible for (2.1). □

An immediate implication is that if $\hat{\alpha}$, a local optimum of (1.2), has enough zero components, the training error is not large as many $\hat{\alpha}_i$'s corresponding inequalities $(Q\hat{\alpha})_i + by_i \geq 1$ are satisfied. Thus, even if Q is not PSD, it is still possible that the training error is small. Next, we give a more formal analysis on the separability of training data:

Theorem 2 *Consider the problem (1.2) without C:*

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2}\alpha^T Q \alpha - e^T \alpha \\ \text{subject to} \quad & 0 \leq \alpha_i, i = 1, \ldots, l, \\ & y^T \alpha = 0. \end{aligned} \tag{2.2}$$

If it attains a global optimal solution at $\hat{\alpha}$, then

1. *Its optimal objective value is not $-\infty$.*
2. *(2.1) has a feasible solution with $\xi_i = 0$, for $i = 1, \ldots, l$.*
3. *After C is large enough, $\hat{\alpha}$ is also a global optimal solution of (1.2).*

The proof is directly from Theorem 1 which shows that $\hat{\alpha}$ is feasible for (2.1) with $\xi_i = 0$. The third property comes from the fact that when $C \geq \max_i \hat{\alpha}_i$, $\hat{\alpha}$ is also optimal for (1.2).

Theorem 2 suggests that even if Q is not PSD, if the optimal solution of (2.2) is attained, the kernel matrix has the ability to fully separate the training data. This somehow gives an explanation why sometimes non-PSD kernels work. The next issue is to see if any conditions on a kernel matrix imply this property. This will help the analysis of the sigmoid kernel.

Several earlier work have given useful results for the analysis here. In particular, it has been shown that a conditionally PSD (CPSD) kernel is good enough for SVM. A matrix K is CPSD (CPD) if for all $v \neq 0$ with $\sum_{i=1}^{l} v_i = 0$, $v^T K v \geq 0$ (> 0). Note that some earlier work use different names: conditionally PD (strictly PD) for the case of ≥ 0 (> 0). More properties can be seen in, for example, (Berg, Christensen, and Ressel 1984). Then, the use of a CPSD kernel is equivalent to the use of a PSD one as $y^T \alpha = 0$ in (1.2) plays a similar role of $\sum_{i=1}^{l} v_i = 0$ in the definition of CPSD (Schölkopf 2000). Note that here for easier analyses, we will work only on the kernel matrices but not the kernel functions. Therefore, results will be more restricted. The following theorem gives properties which imply the existence of optimal solutions of (2.2).

Theorem 3

1. *A kernel matrix K is CPD if and only if there is Δ such that $K + \Delta ee^T$ is PD.*

2. *If K is CPD, then the solution of (2.2) is attained and its optimal objective value is greater than $-\infty$.*

Proof.

The "if" part of the first result is very simple by definition. For any $v \neq 0$ with $e^T v = 0$,

$$v^T K v = v^T (K + \Delta ee^T) v > 0,$$

so K is CPD.

On the other hand, if K is CPD but there is no Δ such that $K + \Delta ee^T$ is PD, there are infinite $\{v_i, \Delta_i\}$ with $\|v_i\| = 1, \forall i$ and $\Delta_i \to \infty$ as $i \to \infty$ such that

$$v_i^T (K + \Delta_i ee^T) v_i \leq 0, \forall i. \tag{2.3}$$

As $\{v_i\}$ is in a compact region, there is a subsequence $\{v_i\}, i \in \mathcal{K}$ which converges to v^*. Since $v_i^T K v_i \to (v^*)^T K v^*$ and $e^T v_i \to e^T v^*$,

$$\lim_{i \to \infty, i \in \mathcal{K}} \frac{v_i^T (K + \Delta_i ee^T) v_i}{\Delta_i} = (e^T v^*)^2 \leq 0.$$

Therefore, $e^T v^* = 0$. By the CPD of K, $(v^*)^T K v^* > 0$ so

$$v_i^T (K + \Delta_i ee^T) v_i > 0 \text{ after } i \text{ is larger enough,}$$

which contradicts (2.3).

For the second result of this theorem, if K is CPD, we have shown that $K + \Delta ee^T$ is PD. Hence (2.2) is equivalent to

$$\begin{array}{ll} \min\limits_{\alpha} & \frac{1}{2}\alpha^T(Q + \Delta yy^T)\alpha - e^T\alpha \\ \text{subject to} & 0 \leq \alpha_i, i = 1, \ldots, l, \\ & y^T\alpha = 0, \end{array} \tag{2.4}$$

which is a strict convex programming problem. Hence (2.4) attains a unique global minimum and so does (2.2). □

Unfortunately, the property that (2.2) has a finite objective value is not equivalent to the CPD of a matrix. The main reason is that (2.2) has additional constraints $\alpha_i \geq 0, i = 1, \ldots, l$. We illustrate this by a simple example: If

$$K = \begin{bmatrix} 1 & 2 & -1 \\ 2 & 1 & -1 \\ -1 & -1 & 0 \end{bmatrix} \text{ and } y = \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix},$$

we can get that

$$\begin{aligned} \frac{1}{2}\alpha^T Q\alpha - e^T\alpha &= \frac{1}{2}[3(\alpha_1 - \frac{2}{3})^2 + 3(\alpha_2 - \frac{2}{3})^2 + 8\alpha_1\alpha_2 - \frac{8}{3}] \\ &\geq -\frac{4}{3} \text{ if } \alpha_1 \geq 0 \text{ and } \alpha_2 \geq 0. \end{aligned}$$

However, K is not CPD as we can easily set $\alpha_1 = -\alpha_2 = 1, \alpha_3 = 0$ which satisfy $e^T\alpha = 0$ but $\alpha^T K\alpha = -2 < 0$.

Moreover, the first result of the above theorem may not hold if K is only CPSD. For example, $K = [\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}]$ is CPSD as for any $\alpha_1 + \alpha_2 = 0$, $\alpha^T K\alpha = 0$. However, for any $\Delta \neq 0$, $K + \Delta ee^T$ has an eigenvalue $\Delta - \sqrt{\Delta^2 + 1} < 0$. Therefore, there is no Δ such that $K + \Delta ee^T$ is PSD. On the other hand, even though $K + \Delta ee^T$ PSD implies its CPSD, they both may not guarantee the optimal objective value of (2.2) is finite. For example, if $K = [\begin{smallmatrix} 0 & 0 \\ 0 & 0 \end{smallmatrix}]$, it satisfies both properties but the objective value of (2.2) can be $-\infty$.

Next we use concepts given in this section to analyze the sigmoid kernel.

3 The Behavior of the Sigmoid Kernel

In this section, we consider the sigmoid kernel $K(x_i, x_j) = \tanh(ax_i^T x_j + r)$. The kernel takes two parameters: a and r. For $a > 0$, we can view a as a scaling parameter of the input data, and r as a shifting parameter that controls the threshold of mapping. For $a < 0$, the dot-product of the input data is not only scaled but reversed. In the following table we summarize the behaviors in different parameter combinations, which will be discussed in the rest of this section. It concludes that the first case, $a > 0$ and $r < 0$, is more suitable for the sigmoid kernel.

a	r	results
$+$	$-$	K is CPD after r is small; similar to RBF for small a
$+$	$+$	in general not as good as the $(+,-)$ case
$-$	$+$	objective value of (2.2) $-\infty$ after r large enough
$-$	$-$	easily the objective value of (2.2) $-\infty$

Case 1: $a > 0$ and $r < 0$

We analyze the limiting case of this region and show that when r is small enough, the matrix K is CPD. To prove this, we begin with a lemma about the sigmoid function:

Lemma 1 *Given any δ,*

$$\lim_{x \to -\infty} \frac{1+\tanh(x+\delta)}{1+\tanh(x)} = e^{2\delta}.$$

Proof.

Since $\tanh(x) = (e^x - e^{-x})/(e^x + e^{-x})$, $1 + \tanh(x) = 2e^x/(e^x + e^{-x})$. Then,

$$\frac{1+\tanh(x+\delta)}{1+\tanh(x)} = \frac{e^x + e^{-x}}{2e^x}\frac{2e^{x+\delta}}{e^{x+\delta}+e^{-x-\delta}} = \frac{e^{2x}+1}{e^{2x+2\delta}+1}e^{2\delta}.$$

Therefore,

$$\lim_{x \to -\infty} \frac{1+\tanh(x+\delta)}{1+\tanh(x)} = e^{2\delta}.$$

□

With this lemma, we can prove that the sigmoid kernel matrices are CPD when r is small enough:

Theorem 4 *Given any training set, if $x_i \neq x_j$, for $i \neq j$ and $a > 0$, there exists $\hat{r}$ such that for all $r \leq \hat{r}$, $K + ee^T$ is PD.*

Proof.

Let $H^r \equiv (K + ee^T)/(1 + \tanh(r))$, where $K_{ij} = \tanh(ax_i^T x_j + r)$. From Lemma 1,

$$\begin{aligned} & \lim_{r \to -\infty} H_{ij}^r \\ = & \lim_{r \to -\infty} \frac{1 + \tanh(ax_i^T x_j + r)}{1 + \tanh(r)} \\ = & e^{2ax_i^T x_j}. \end{aligned}$$

Let $\bar{H} = \lim_{r \to -\infty} H^r$. Thus, $\bar{H}_{ij} = e^{2ax_i^T x_j} = e^{a\|x_i\|^2} e^{-a\|x_i - x_j\|^2} e^{a\|x_j\|^2}$. If written in matrix products, we can see that the first and last terms form the same diagonal matrices with positive elements. And the middle one is in the form of an RBF kernel matrix. From (Micchelli 1986), if $x_i \neq x_j$, for $i \neq j$, the RBF kernel matrix is PD. Therefore, $\bar{H}$ is PD.

If H^r is not PD after r is small enough, there is an infinite sequence $\{r_i\}$ with $\lim_{i \to \infty} r_i = -\infty$ and $H^{r_i}, \forall i$ are not PD. Thus, for each r_i, there exists $\|v_i\| = 1$ such that $v_i^T H^{r_i} v_i \leq 0$.

Since v_i is a bounded infinite sequence, there is a subsequence which converges to $\bar{v} \neq 0$. Therefore, $\bar{v}^T \bar{H} \bar{v} \leq 0$ which contradicts the fact that $\bar{H}$ is PD. So there is $\hat{r}$ such that for all $r \leq \hat{r}$, H^r is PD. By the definition of H^r, $K + ee^T$ is PD as well. □

With Theorems 3 and 4, K is CPD after r is small enough. In addition, we can define a new kernel:

$$\tilde{K}(x_i, x_j) = \tanh(ax_i^T x_j + r) + 1 = 2/(1 + e^{-2(ax_i^T x_j + r)}).$$

It is in the form of a logistic function commonly used in neural networks.

Corollary 1 *For $a > 0$ and any training set with $x_i \neq x_j$ for $i \neq j$, there exists $\hat{r} < 0$ such that for all $r \leq \hat{r}$, the logistic function $\tilde{K}(x_i, x_j) = 2/(1 + e^{-2(ax_i^T x_j + r)})$ can be used to form a PD kernel matrix.*

Note that Theorem 4 provides a connection between the sigmoid and a special PD kernel related to the RBF kernel when a is fixed and r gets small enough. In Section 4, we will discuss more about the relation between the sigmoid and the RBF kernels.

Case 2: $a > 0$ and $r \geq 0$

It was stated in (Burges 1999) that if $\tanh(ax_i^T x_j + r)$ is PD, then $r \geq 0$ and $a \geq 0$. However, the inverse does not hold so for this case, kernels may not be PD and the practical viability is not clear. As Section 2 has shown that useful kernels are in a broader set than PD ones, we discuss this case by checking the separability of training data.

Comparing to Case 1, we show that it is more possible that the objective value of (2.2) goes to $-\infty$. Therefore, with experiments in Section 4, we conclude that in general using $a > 0$ and $r \geq 0$ is not as good as $a > 0$ and $r < 0$.

The following theorem discusses possible situations that (2.2) has the objective value $-\infty$:

Theorem 5

1. *If there are i and j such that $y_i \neq y_j$ and $K_{ii} + K_{jj} - 2K_{ij} \leq 0$, (2.2) has the optimal objective value $-\infty$.*

2. *For the sigmoid kernel, if*

$$\max_i (a\|x_i\|^2 + r) \leq 0, \tag{3.1}$$

then $K_{ii} + K_{jj} - 2K_{ij} > 0$ for any $x_i \neq x_j$.

Proof.

For the first result, let $\alpha_i = \alpha_j = \Delta$ and $\alpha_k = 0$ for $k \neq i, j$. Then, the objective value of (2.2) is

$$\begin{aligned} & \frac{1}{2}\alpha^T Q \alpha - e^T \alpha \\ = \; & \Delta^2 (K_{ii} - 2K_{ij} + K_{jj}) - 2\Delta. \end{aligned}$$

Thus, $\Delta \rightarrow \infty$ leads to a feasible solution of (2.2) with objective value $-\infty$.

For the second result, now

$$\begin{aligned} & K_{ii} - 2K_{ij} + K_{jj} \\ = \; & \tanh(a\|x_i\|^2 + r) - 2\tanh(a\|x_i^T x_j\| + r) + \tanh(a\|x_j\|^2 + r). \end{aligned} \tag{3.2}$$

Since $\max_i (a\|x_i\|^2 + r) \leq 0$, by the monotonicity of $\tanh(x)$ and its strict convexity when $x \leq 0$,

$$\begin{aligned} & \frac{\tanh(a\|x_i\|^2 + r) + \tanh(a\|x_j\|^2 + r)}{2} \\ \geq \; & \tanh(\frac{(a\|x_i\|^2 + r) + (a\|x_j\|^2 + r)}{2}) && (3.3) \\ = \; & \tanh(a\frac{\|x_i\|^2 + \|x_j\|^2}{2} + r) \\ > \; & \tanh(a x_i^T x_j + r). && (3.4) \end{aligned}$$

Note that the last inequality uses the property that $x_i \neq x_j$.

Then, by (3.2) and (3.4), $K_{ii} - 2K_{ij} + K_{jj} > 0$, so the proof is complete. □

The requirement that $x_i \neq x_j$ is in general true if there are no duplicated training instances. Apparently, (3.1) must happen (for $a > 0$) when r is negative. If (3.1) is wrong, it is possible that $a\|x_i\|^2+r \geq 0$ and $a\|x_j\|^2+r \geq 0$. Then due to the concavity of $\tanh(x)$ at the positive side, "$\geq$" in (3.3) becomes "$\leq$." Thus, $K_{ii} - 2K_{ij} + K_{jj}$ may be ≤ 0 and (2.2) has the optimal objective value $-\infty$.

Case 3: $a < 0$ and $r > 0$

The following theorem tells us that the parameters $a < 0$ and large $r > 0$ may not be a good choice.

Theorem 6 *For any given training set, if $a < 0$ and each class has at least one data point, there exists $\bar{r} > 0$ such that for all $r \geq \bar{r}$, (2.2) has optimal objective value $-\infty$.*

Proof.

Since $K_{ij} = \tanh(ax_i^T x_j + r) = -\tanh(-ax_i^T x_j - r)$, by Theorem 4, there is $-\bar{r} < 0$ such that for all $-r \leq -\bar{r}$, $-K + ee^T$ is PD. That is, there exist $\bar{r} > 0$ such that for all $r \geq \bar{r}$, any α with $y^T\alpha = 0$ and $\alpha \neq 0$ satisfies $\alpha^T Q\alpha < 0$.

Since there is at least one data point in each class, we can find $y_i = +1$ and $y_j = -1$. Let $\alpha_i = \alpha_j = \Delta$, and $\alpha_k = 0$ for $k \neq i, j$ be a feasible solution of (2.2). The objective value decreases to $-\infty$ as $\Delta \to \infty$. Therefore, for all $r \geq \bar{r}$, (2.2) has optimal objective value $-\infty$. □

Case 4: $a < 0$ and $r \leq 0$

The following theorem provides evidence that the optimal objective value of (2.2) easily goes to $-\infty$ in this case:

Theorem 7 *For any given training set, if $a < 0$, $r \leq 0$, and there are x_i, x_j such that*

$$x_i^T x_j \leq \min(\|x_i\|^2, \|x_j\|^2)$$

and $y_i \neq y_j$, (2.2) has optimal objective value $-\infty$.

Proof.

By $x_i^T x_j \leq \min(\|x_i\|^2, \|x_j\|^2)$, (3.2), and the monotonicity of $\tanh(x)$,

$$
\begin{aligned}
& K_{ii} - 2K_{ij} + K_{jj} \\
\leq\ & \tanh(a\|x_i\|^2 + r) + \tanh(a\|x_j\|^2 + r) - 2\tanh(a\min(\|x_i\|^2, \|x_j\|^2) + r) \\
\leq\ & 0.
\end{aligned}
$$

Then the proof follows from Theorem 5. □

Note that the situation $x_i^T x_j < \min(\|x_i\|^2, \|x_j\|^2)$ and $y_i \neq y_j$ easily happens if the two classes of training data are not close in the input space. Thus, $a < 0$ and $r \leq 0$ is generally not a good choice of parameters.

4 Relation with the RBF Kernel

In this section we extend Case 1 (i.e. $a > 0$, $r < 0$) in Section 3 to show that the sigmoid kernel behaves like the RBF kernel when (a, r) are in a certain range.

Lemma 1 implies that when $r < 0$ is small enough,

$$
1 + \tanh(a x_i^T x_j + r) \approx (1 + \tanh(r))(e^{2a x_i^T x_j}). \tag{4.1}
$$

If we further make a close to 0, $e^{a\|x\|^2} \approx 1$ so

$$
e^{2a x_i^T x_j} = e^{a\|x_i\|^2} e^{-a\|x_i - x_j\|^2} e^{a\|x_j\|^2} \approx e^{-a\|x_i - x_j\|^2}.
$$

Therefore, when $r < 0$ is small enough and a is close to 0,

$$
1 + \tanh(a x_i^T x_j + r) \approx (1 + \tanh(r))(e^{-a\|x_i - x_j\|^2}), \tag{4.2}
$$

a form of the RBF kernel.

However, the closeness of kernel elements does not directly imply similar generalization performance. Hence, we need to show that they have nearly the same decision functions. Note that the objective function of (1.2) is the same as:

$$
\begin{aligned}
\frac{1}{2}\alpha^T Q \alpha - e^T \alpha &= \frac{1}{2}\alpha^T (Q + yy^T)\alpha - e^T \alpha \qquad (4.3) \\
&= \frac{1}{1 + \tanh(r)} \left(\frac{1}{2}\tilde{\alpha}^T \frac{Q + yy^T}{1 + \tanh(r)} \tilde{\alpha} - e^T \tilde{\alpha}\right),
\end{aligned}
$$

where $\tilde{\alpha} \equiv (1 + \tanh(r))\alpha$, and (4.3) comes from the equality constraint in (1.2). Multiplying the objective function of (1.2) by $(1 + \tanh(r))$, and setting $\tilde{C} = (1 +$

$\tanh(r))C$, solving (1.2) is the same as solving

$$\begin{aligned}\min_{\tilde{\alpha}} \quad & F_r(\tilde{\alpha}) = \frac{1}{2}\tilde{\alpha}^T \frac{Q + yy^T}{1 + \tanh(r)} \tilde{\alpha} - e^T \tilde{\alpha} \\ \text{subject to} \quad & 0 \leq \tilde{\alpha}_i \leq \tilde{C}, i = 1, \dots, l, \\ & y^T \tilde{\alpha} = 0.\end{aligned} \tag{4.4}$$

Given a fixed $\tilde{C}$, as $r \to -\infty$, since $(Q+yy^T)_{ij} = y_i y_j (K_{ij}+1)$, the problem approaches

$$\begin{aligned}\min_{\tilde{\alpha}} \quad & F_T(\tilde{\alpha}) = \frac{1}{2}\tilde{\alpha}^T \bar{Q} \tilde{\alpha} - e^T \tilde{\alpha} \\ \text{subject to} \quad & 0 \leq \tilde{\alpha}_i \leq \tilde{C}, i = 1, \dots, l, \\ & y^T \tilde{\alpha} = 0,\end{aligned} \tag{4.5}$$

where $\bar{Q}_{ij} = y_i y_j e^{2ax_i^T x_j}$ is a PD kernel matrix when $x_i \neq x_j$ for all $i \neq j$. Then, we can prove the following theorem:

Theorem 8 *Given fixed a and $\tilde{C}$, assume that $x_i \neq x_j$ for all $i \neq j$, and the optimal b of the decision function from (4.5) is unique. Then for any data point x,*

$$\begin{aligned}& \lim_{r \to -\infty} && \text{decision value at } x \text{ using the sigmoid kernel in (1.2)} \\ = & && \text{decision value at } x \text{ using (4.5).}\end{aligned}$$

We leave the proof in Appendix A. Theorem 8 tells us that when $r < 0$ is small enough, the separating hyperplanes of (1.2) and (4.5) are almost the same. Similar cross-validation accuracy will be shown in the later experiments.

(Keerthi and Lin 2003, Theorem 2) shows that when $a \to 0$, for any given $\bar{C}$, the decision value by the SVM using the RBF kernel $e^{-a\|x_i - x_j\|^2}$ with the error cost $\frac{\bar{C}}{2a}$ approaches the decision value of the following linear SVM:

$$\begin{aligned}\min_{\bar{\alpha}} \quad & \frac{1}{2}\sum_i \sum_j \bar{\alpha}_i \bar{\alpha}_j y_i y_j x_i^T x_j - \sum_i \bar{\alpha}_i \\ \text{subject to} \quad & 0 \leq \bar{\alpha}_i \leq \bar{C}, i = 1, \dots, l, \\ & y^T \bar{\alpha} = 0.\end{aligned} \tag{4.6}$$

The same result can be proved for the SVM in (4.5). Therefore, under the assumption that the optimal b of the decision function from (4.6) is unique, for any data point x,

$$\begin{aligned}& \lim_{a \to 0} && \text{decision value at } x \text{ using the RBF kernel with } \tilde{C} = \tfrac{\bar{C}}{2a} \\ = & && \text{decision value at } x \text{ using (4.6) with } \bar{C} \\ = & \lim_{a \to 0} && \text{decision value at } x \text{ using (4.5) with } \tilde{C} = \tfrac{\bar{C}}{2a}.\end{aligned}$$

Then we can get the similarity between the sigmoid and the RBF kernels as follows:

Theorem 9 *Given a fixed $\bar{C}$, assume that $x_i \neq x_j$ for all $i \neq j$, and each of* (4.5) *after a is close to* 0 *and* (4.6) *has a unique b. Then for any data point x,*

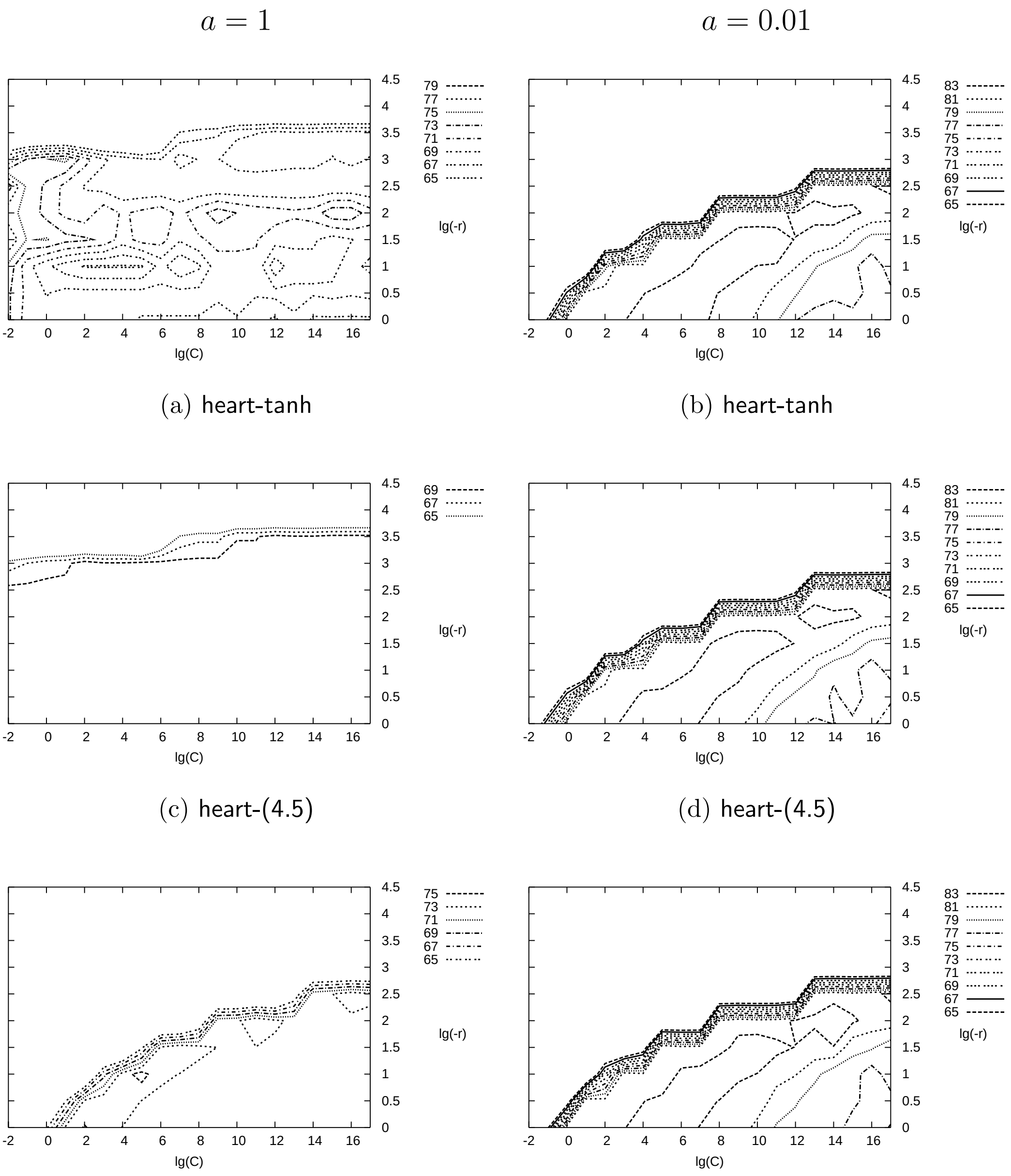
$$\begin{aligned}
& \lim_{a\to 0}\ \lim_{r\to-\infty} && \text{decision value at } x \text{ using the sigmoid kernel with } C = \tfrac{\bar{C}}{1+\tanh(r)} \\
= & \lim_{a\to 0} && \text{decision value at } x \text{ using (4.5) with } \tilde{C} = \tfrac{\bar{C}}{2a} \\
= & \lim_{a\to 0} && \text{decision value at } x \text{ using the RBF kernel with } \tilde{C} = \tfrac{\bar{C}}{2a} \\
= & && \text{decision value at } x \text{ using the linear kernel with } \bar{C}.
\end{aligned}$$

From Theorems 8 and 9, the sigmoid SVM with an error cost C would perform similarly to (4.5) with $\tilde{C} = (1 + \tanh(r))C$ when r is small. If we further make a close to 0, the sigmoid SVM with C would also perform similarly to the RBF SVM with $\tilde{C}$. We can observe these from Figure 1. The contours show five-fold cross-validation accuracy of the data set heart in different r and C. The contours with $a = 1$ are on the left-hand-side, while those with $a = 0.01$ are on the right-hand-side. Other parameters considered here are $\log_2 C$ from -2 to 13, with grid space 1, and $\log_2(-r)$ from 0 to 4.5, with grid space 0.5. Detailed description of data sets as well as the tool used to draw the contours will be shown later in this section.

From both sides of Figure 1, we can see that the middle contour (using (4.5)) is similar to the top one (using tanh) when r gets small. This verifies our approximation in (4.1) as well as Theorem 8. However, on the left-hand-side, since a is not small enough, the data-dependent scaling term $e^{a\|x_i\|^2}$ between (4.1) and (4.2) is large and causes a difference between the middle and bottom contours. When a is reduced to 0.01 on the right-hand-side, the top, middle, and bottom contours all become similar when r is small. This observation corresponds to Theorem 9.

We observe this on other data sets, too. Note that for some problems, unlike Figure 1, the performance using small a values may not be better than that using large a values. That is, Figure 1 and Theorem 9 shows only a connection between the sigmoid and the RBF kernels when (a, r) are in a limited range. Thus, we try to compare the two kernels using parameters in other ranges. In Table 1 we compare the best five-fold CV rates using the sigmoid and RBF kernels.

We fix a to $1/n$, where n is the number of features, and change the value of r. Cross-validation on different r (-2 to 2 with grid space 0.2) and $\log_2 C$ (-2 to 13 with grid space 1) is conducted. For the RBF kernel, $K(x_i, x_j) = e^{-a\|x_i - x_j\|^2}$ so the two



(a) heart-tanh (b) heart-tanh

(c) heart-(4.5) (d) heart-(4.5)

(e) heart-RBF-$\tilde{C}$ (f) heart-RBF-$\tilde{C}$

Figure 1: Performance of different kernels

parameters are a and C. The grid search is then on $\log_2 a$ (-11 to -2 with grid space 1) and $\log_2 C$ (-2 to 13 with grid space 1).

Four problems are tested: heart, german, diabete, and a1a. They are from (Michie, Spiegelhalter, and Taylor 1994) and (Blake and Merz 1998). The first three data sets are linearly scaled so values of each attribute are in [-1, 1]. For a1a, its values of each attribute are in [0, 1] so we do not scale it. We solve (1.2) using LIBSVM (Chang and Lin 2001), with its model selection tool for grid search and contour drawing. Note that LIBSVM, an SMO-type decomposition implementation, uses techniques in Section 6 for solving non-convex optimization problems. A local optimum of (1.2) is obtained for constructing the decision function.

Table 1: Comparison of cross-validation rates between kernels

data set	Sigmoid kernel: $\tanh(x_i^T x_j/n + r)$		RBF kernel: $e^{-a\|x_i-x_j\|^2}$	
	best $(r, \log_2 C)$	best CV rate	best $(\log_2 a, \log_2 C)$	best CV rate
heart	$(1.6, -1)$	84.1%	$(-11, 7)$	84.1%
german	$(-1.4, 4)$	77.8%	$(-5, 3)$	77.5%
diabete	$(-0.4, 5)$	77.7%	$(-9, 13)$	77.6%
a1a	$(0, 8)$	83.9%	$(-8, 6)$	83.8%

The resulting contours are also shown on the left-hand-side of Figure 2 of the next Section. We can see that the performance of the sigmoid kernel is comparable to that of the RBF kernel. From the limited experiments here, on one hand, for appropriate parameters, the sigmoid kernel performs well in practice. On the other hand, it is not better than RBF. As RBF has properties of being PD and having fewer parameters, somehow there is no strong reason to use the sigmoid.

5 The Importance of the Linear Constraint $y^T\alpha = 0$

In Section 3 we show that for certain parameters, the kernel matrix using the sigmoid kernel is CPD. This is strongly related to the linear constraint $y^T\alpha = 0$ in the dual problem (1.2). In this section, we would like to investigate the effect with and without this linear constraint.

Recall that $y^T\alpha = 0$ of (1.2) is originally derived from the bias term b of (1.1). It has been known that if the kernel function is PD and $x_i \neq x_j$ for all $i \neq j$, Q will be PD and the problem (2.2) attains an optimal solution. In other words, training data is guaranteed to be fully separated. Therefore, for PD kernels such as the RBF, in many cases, the performance is not affected much if the biased term b is not used.

By doing so, the dual problem becomes

$$\begin{array}{rl} \min\limits_{\alpha} & \frac{1}{2}\alpha^T Q\alpha - e^T\alpha \\ \text{subject to} & 0 \leq \alpha_i \leq C, i = 1, \ldots, l. \end{array} \tag{5.1}$$

For the sigmoid kernel, we may think that (5.1) is also acceptable. It turns out that without $y^T\alpha = 0$, in more cases, (5.1) without the upper bound C, has the objective value $-\infty$. Thus, training data are not properly separated. The following theorem gives an example on such cases:

Theorem 10 *If there is one $K_{ii} < 0$ and there is no upper bound C of α, (5.1) has optimal objective value $-\infty$.*

Proof.

Let $\alpha_i = \Delta$ and $\alpha_k = 0$ for $k \neq i$. We can easily see that $\Delta \to \infty$ leads to an optimal objective value $-\infty$. □

Note that for sigmoid kernel matrices, this situation happens when $\min_i(a\|x_i\|^2 + r) < 0$. Thus, when $a > 0$ but r is small, unlike our analysis in Case 1 of Section 3, solving (5.1) may lead to very different results. We can further show that the training error may be large. If α is a global optimum of (5.1),

$$\begin{aligned} & \min\Big(\frac{1}{2}\sum_{y_i=y_j=1} K_{ij}C^2 - \sum_{y_i=1} C, \frac{1}{2}\sum_{y_i=y_j=-1} K_{ij}C^2 - \sum_{y_i=-1} C\Big) && (5.2) \\ \geq\ & \text{optimal objective value of (5.1)} \\ \geq\ & \frac{1}{2}\sum_{\alpha_i>0,\alpha_j>0} K_{ij}C^2 - \sum_{\alpha_i>0} C. && (5.3) \end{aligned}$$

The first inequality is by setting $\alpha_i = C$ for all $y_i = 1$ or -1 as a feasible solution of (5.1), and using the property $K_{ij} = Q_{ij}$ if $y_i = y_j$. The second inequality comes from $K_{ij} \leq 0$ when r is small, and $\alpha_i \leq C$ by (5.1). Without considering the linear term when C is not small, we can clearly see that (5.3) may be larger than (5.2) if there are too many zero α_i. Therefore, the optimal solution should not contain too many zero α_i so the training error may be large, a situation which makes the sigmoid kernel perform poorly. This will be shown in the following experiments.

We compare the five-fold cross-validation accuracy using problems (1.2) and (5.1). The same four problems as in Table 1 are used, with the same ranges of parameters. We use LIBSVM for solving (1.2), and a modification of BSVM (Hsu and Lin 2002) for (5.1). Results of CV accuracy are presented in Figure 2. Contours of (1.2) are on the left column, and those of (5.1) are on the right. For each contour, the horizontal axis is $\log_2 C$, while the vertical axis is r. The internal optimization solver of BSVM can handle non-convex problems, so its decomposition procedure guarantees the strict decrease of function values throughout all iterations. However, unlike LIBSVM which always obtains a local minimum of (1.2) using the analysis in Section 6, for BSVM, we do not know whether its convergent point is a local minimum of (5.1) or not.

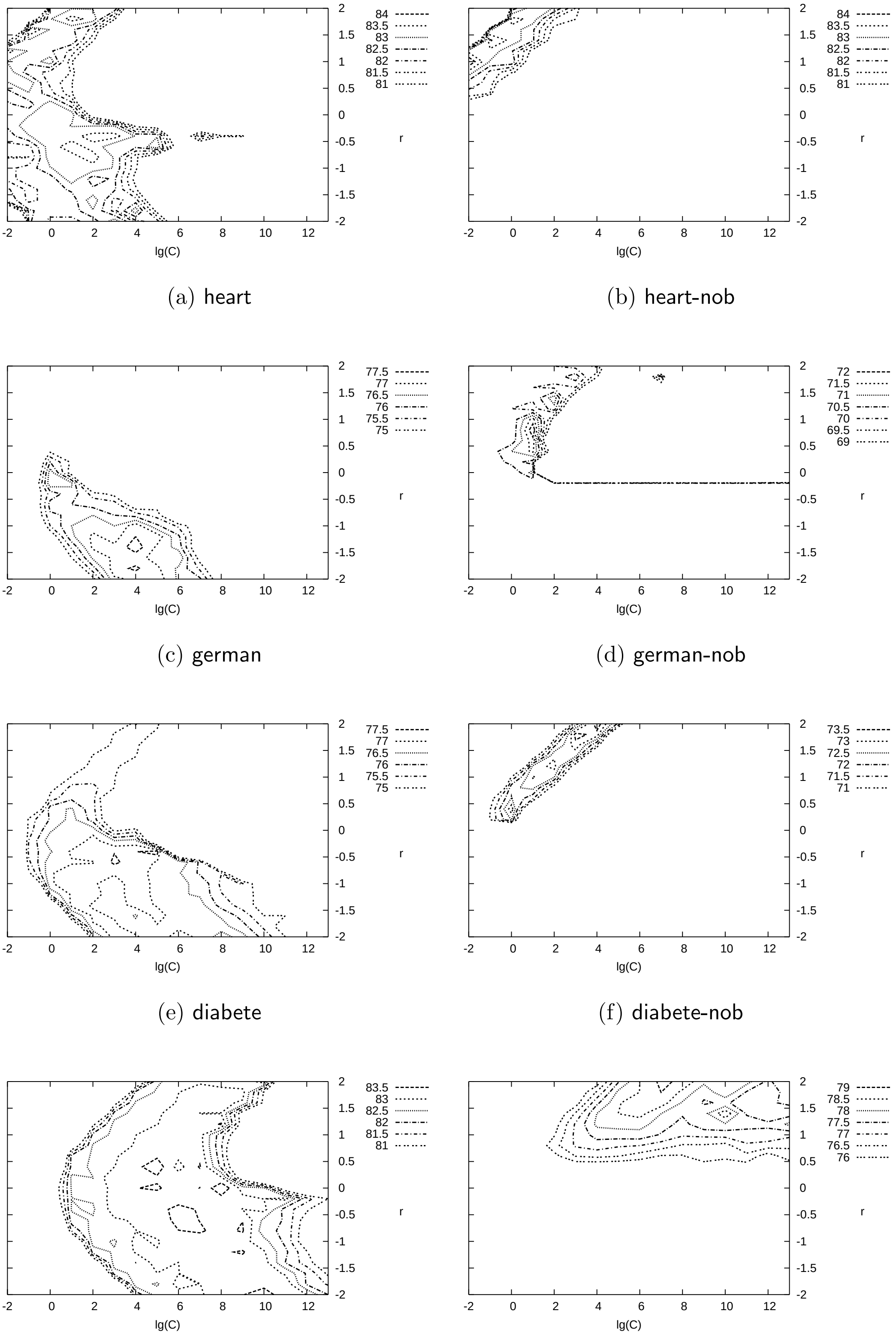
When (1.2) is solved, from Figure 2, higher accuracy generally happens when $r < 0$ (especially german and diabete). This corresponds to our analysis about the CPD of K when $a > 0$ and r small enough. However, sometimes the CV accuracy is also high when $r > 0$. We have also tried the cases of $a < 0$, results are worse.

In addition, by comparing the left and right columns, solving (5.1) gives much worse performance. Also, the good regions shift to $r \geq 0$. This confirms our analysis in Theorem 10 as when $r < 0$, (5.1) without C tends to have the objective value $-\infty$. In other words, without $y^T\alpha = 0$, CPD of K for small r is not useful.

The experiments fully demonstrate the importance of incorporating constraints of the dual problem into the analysis of the kernel. An earlier work (Sellathurai and Haykin 1999) on the sigmoid kernel explains that each kernel element (i.e., K_{ij}) is from a hyperbolic inner product. Thus, a special type of maximal margin still exists. However, as shown in Figure 2, without $y^T\alpha = 0$, the performance is very bad. Thus, the separability of the sigmoid kernel does not come from its own properties. Therefore, a direct analysis on the non-PSD kernel matrix itself may not be very useful.

6 SMO-type Implementation for non-PSD Kernel Matrices

First we discuss how decomposition methods work for PSD kernels and the difficulties for non-PSD cases. The decomposition method (e.g. (Osuna, Freund, and Girosi 1997; Joachims 1998; Platt 1998; Chang and Lin 2001)) is an iterative process. In each step, the index set of variables is partitioned to two sets B and N, where B is the working set. Then in that iteration variables corresponding to N are fixed while a sub-problem on variables corresponding to B is minimized. Thus, if α^k is the current



(a) heart

(b) heart-nob

(c) german

(d) german-nob

(e) diabete

(f) diabete-nob

(g) a1a

(h) a1a-nob

Figure 2: Comparison of cross validation rates between problems with the linear constraint (left) and without it (right)

solution, the following sub-problem is solved:

$$\begin{array}{ll} \min\limits_{\alpha_B} & \frac{1}{2}\begin{bmatrix}\alpha_B^T & (\alpha_N^k)^T\end{bmatrix}\begin{bmatrix}Q_{BB} & Q_{BN}\\ Q_{NB} & Q_{NN}\end{bmatrix}\begin{bmatrix}\alpha_B\\ \alpha_N^k\end{bmatrix} - \begin{bmatrix}e_B^T & (e_N^k)^T\end{bmatrix}\begin{bmatrix}\alpha_B\\ \alpha_N^k\end{bmatrix} \\ \text{subject to} & y_B^T\alpha_B = -y_N^T\alpha_N^k, \\ & 0 \leq \alpha_i \leq C, i \in B. \end{array} \tag{6.1}$$

The objective function of (6.1) can be simplified to

$$\min_{\alpha_B} \quad \frac{1}{2}\alpha_B^T Q_{BB}\alpha_B + (Q_{BN}\alpha_N^k - e_B)^T\alpha_B$$

after removing constant terms.

The extreme of the decomposition method is the Sequential Minimal Optimization (SMO) algorithm (Platt 1998) whose working sets are restricted to two elements. The advantage of SMO is that (6.1) can be easily solved without an optimization package. A simple and common way to select the two variables is through the following form of optimal conditions (Keerthi, Shevade, Bhattacharyya, and Murthy 2001; Chang and Lin 2001): α is a local optimum of (1.2) if and only if α is feasible and

$$\max_{t \in I_{up}(\alpha, C)} -y_t \nabla F(\alpha)_t \leq \min_{t \in I_{low}(\alpha, C)} -y_t \nabla F(\alpha)_t, \tag{6.2}$$

where

$$\begin{aligned} I_{up}(\alpha, C) &\equiv \{t \mid \alpha_t < C, y_t = 1 \text{ or } \alpha_t > 0, y_t = -1\}, \\ I_{low}(\alpha, C) &\equiv \{t \mid \alpha_t < C, y_t = -1 \text{ or } \alpha_t > 0, y_t = 1\}. \end{aligned}$$

Thus, when α^k is feasible but not optimal for (1.2), (6.2) does not hold so a simple selection of $B = \{i, j\}$ is

$$i \equiv \operatorname*{argmax}_{t \in I_{up}(\alpha^k, C)} -y_t \nabla F(\alpha^k)_t \text{ and } j \equiv \operatorname*{argmin}_{t \in I_{low}(\alpha^k, C)} -y_t \nabla F(\alpha^k)_t. \tag{6.3}$$

By considering the variable $\alpha_B = \alpha_B^k + d$, and defining

$$\hat{d}_i \equiv y_i d_i \text{ and } \hat{d}_j \equiv y_j d_j,$$

the two-variable sub-problem is

$$\begin{array}{ll} \min\limits_{\hat{d}_i, \hat{d}_j} & \frac{1}{2}\begin{bmatrix}\hat{d}_i & \hat{d}_j\end{bmatrix}\begin{bmatrix}K_{ii} & K_{ij}\\ K_{ji} & K_{jj}\end{bmatrix}\begin{bmatrix}\hat{d}_i\\ \hat{d}_j\end{bmatrix} + \begin{bmatrix}y_i\nabla F(\alpha^k)_i & y_j\nabla F(\alpha^k)_j\end{bmatrix}\begin{bmatrix}\hat{d}_i\\ \hat{d}_j\end{bmatrix} \\ \text{subject to} & \hat{d}_i + \hat{d}_j = 0, \\ & 0 \leq \alpha_i^k + y_i\hat{d}_i, \alpha_j^k + y_j\hat{d}_j \leq C. \end{array} \tag{6.4}$$

To solve (6.4), we can substitute $\hat{d}_i = -\hat{d}_j$ into its objective function:

$$\begin{aligned} \min_{\hat{d}_j} \quad & \frac{1}{2}(K_{ii} - 2K_{ij} + K_{jj})\hat{d}_j^2 + (-y_i \nabla F(\alpha^k)_i + y_j \nabla F(\alpha^k)_j)\hat{d}_j. && (6.5a) \\ \text{subject to} \quad & L \le \hat{d}_j \le H, && (6.5b) \end{aligned}$$

where L and H are upper and lower bounds of $\hat{d}_j$ after including information on $\hat{d}_i$: $\hat{d}_i = -\hat{d}_j$ and $0 \le \alpha_i^k + y_i \hat{d}_i \le C$. For example, if $y_i = y_j = 1$,

$$L = \max(-\alpha_j^k, \alpha_i^k - C) \text{ and } H = \min(C - \alpha_j^k, \alpha_i^k).$$

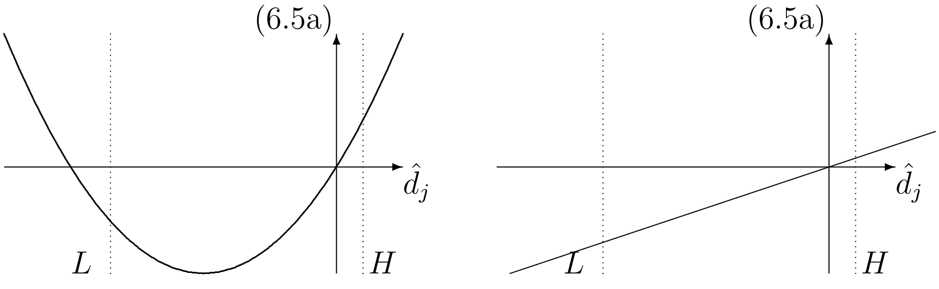
Since $i \in I_{up}(\alpha^k, C)$ and $j \in I_{low}(\alpha^k, C)$, we can clearly see $L < 0$ but H only ≥ 0. If Q is PSD, $K_{ii} + K_{jj} - 2K_{ij} \ge 0$ so (6.5) is a convex parabola or a straight line. In addition, from the working set selection strategy in (6.3), $-y_i \nabla F(\alpha^k)_i + y_j \nabla F(\alpha^k)_j > 0$, so (6.5) is like Figure 3. Thus, there exists $\hat{d}_j < 0$ such that the objective value of (6.5) is strictly decreasing. In addition, $\hat{d}_j < 0$ also shows the direction toward the minimum of the function.

If $K_{ii} + K_{jj} - 2K_{ij} > 0$, the way to solve (6.5) is by calculating the minimum of (6.5a) first:

$$-\frac{-y_i \nabla F(\alpha^k)_i + y_j \nabla F(\alpha^k)_j}{K_{ii} - 2K_{ij} + K_{jj}} < 0. \tag{6.6}$$

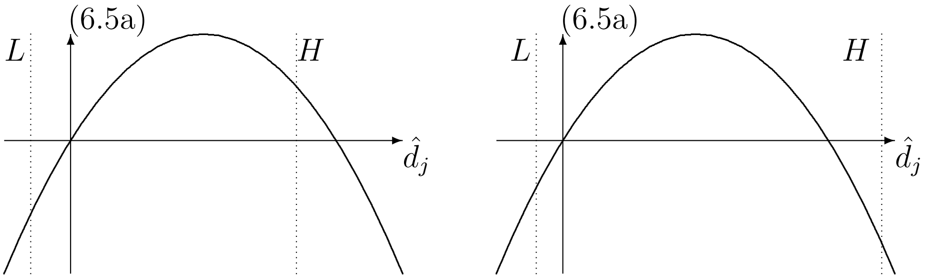
Then, if $\hat{d}_j$ defined by the above is less than L, we reduce $\hat{d}_j$ to the lower bound. If the kernel matrix is only PSD, it is possible that $K_{ii} - 2K_{ij} + K_{jj} = 0$, as shown in Figure 3(b). In this case, using the trick under IEEE floating point standard (Goldberg 1991), we can make sure that (6.6) to be $-\infty$ which is still defined. Then, a comparison with L still reduce $\hat{d}_j$ to the lower bound. Thus, a direct (but careful) use of (6.6) does not cause any problem. More details are in (Chang and Lin 2001). The above procedure explains how we solve (6.5) in an SMO-type software.

If $K_{ii} - 2K_{ij} + K_{jj} < 0$, which may happen if the kernel is not PSD, (6.6) is positive. That is, the quadratic function (6.5a) is concave (see Figure 4) and a direct use of (6.6) move the solution toward (6.5a)'s maximum. Therefore, the decomposition method may not have the objective value strictly decreasing, a property usually required for an optimization algorithm. Moreover, it may not be feasible to move along a positive direction $\hat{d}_j$. For example, if $\alpha_i^k = 0, y_i = 1$ and $\alpha_j^k = 0, y_j = -1$, $H = 0$ in (6.5) so we can neither decrease α_i nor α_j. Thus, under the current setting for PSD kernels, it is possible that the next solution stays at the same point so the program never ends. In the following we propose different approaches to handle this difficulty.



(a) $K_{ii} + K_{jj} - 2K_{ij} > 0$ (b) $K_{ii} + K_{jj} - 2K_{ij} = 0$

Figure 3: Solving the convex sub-problem (6.5)



(a) L is the minimum (b) H is the minimum

Figure 4: Solving the concave sub-problem (6.5)

6.1 Restricting the Range of Parameters

The first approach is to restrict the parameter space. In other words, users are allowed to specify only certain kernel parameters. Then the sub-problem is guaranteed to be convex so the original procedure for solving sub-problems works without modification.

Lemma 2 *If $a > 0$ and*

$$\max_i (a\|x_i\|^2 + r) \leq 0, \tag{6.7}$$

any two-variable sub-problem of an SMO algorithm is convex.

We have explained that the sub-problem can be reformulated as (6.5) so the proof is reduced to show that $K_{ii} - 2K_{ij} + K_{jj} \geq 0$. This, in fact, is nearly the same as the proof of Theorem 5. The only change is that without assuming $x_i \neq x_j$, "> 0" becomes "≥ 0."

Therefore, if we require that a and r satisfy (6.7), we will never have an endless loop staying at one α^k.

6.2 An SMO-type Method for General non-PSD Kernels

Results in Section 6.1 depend on properties of the sigmoid kernel. Here we will propose an SMO-type method which is able to handle all kernel matrices no matter they are PSD or not. To have such a method, the key is on solving the sub-problem when $K_{ii} - 2K_{ij} + K_{jj} < 0$. In this case, (6.5a) is a concave quadratic function like that in Figure 4. The two sub-figures clearly show that the global optimal solution of (6.5) can be obtained by checking the objective values at two bounds L and H.

A disadvantage is that this procedure of checking two points is different from the solution procedure of $K_{ii} - 2K_{ij} + K_{jj} \geq 0$. Thus, we propose to consider only the lower bound L which, as $L < 0$, always ensures the strict decrease of the objective function. Therefore, the algorithm is as follows:

$$\begin{aligned}&\text{If } K_{ii} - 2K_{ij} + K_{jj} > 0, \text{then } \hat{d}_j \text{ is the maximum of (6.6) and } L,\\ &\text{Else } \hat{d}_j = L.\end{aligned} \tag{6.8}$$

Practically the change of the code may be only from (6.6) to

$$-\frac{-y_i \nabla F(\alpha^k)_i + y_j \nabla F(\alpha^k)_j}{\max(K_{ii} - 2K_{ij} + K_{jj}, 0)}. \tag{6.9}$$

When $K_{ii} + K_{jj} - 2K_{ij} < 0$, (6.9) is $-\infty$. Then the same as the situation of $K_{ii} + K_{jj} - 2K_{ij} = 0$, $\hat{d}_j = L$ is taken.

An advantage of this strategy is that we do not have to exactly solve (6.5). (6.9) also shows that a very simple modification from the PSD-kernel version is possible. Moreover, it is easier to prove the asymptotic convergence. The reason will be discussed after Lemma 3. In the following we prove that any limit point of the decomposition procedure discussed above is a local minimum of (1.2). In earlier convergence results, Q is PSD so a local minimum is already a global one.

If the working set selection is via (6.3), existing convergence proofs for PSD kernels (Lin 2001; Lin 2002) require the following important lemma which is also needed here:

Lemma 3 *There exists $\sigma > 0$ such that for any k,*

$$F(\alpha^{k+1}) \leq F(\alpha^k) - \frac{\sigma}{2}\|\alpha^{k+1} - \alpha^k\|^2. \tag{6.10}$$

Proof.

If $K_{ii} + K_{jj} - 2K_{ij} \geq 0$ in the current iteration, (Lin 2002) shows that by selecting σ as the following number

$$\min\{\frac{2}{C}, \min_{t,r}\{\frac{K_{tt} + K_{rr} - 2K_{tr}}{2} \mid K_{tt} + K_{rr} - 2K_{tr} > 0\}\}, \tag{6.11}$$

(6.10) holds.

If $K_{ii} + K_{jj} - 2K_{ij} < 0$, $\hat{d}_r = L < 0$ is the step chosen so $(-y_i \nabla F(\alpha^k)_i + y_j \nabla F(\alpha^k)_j)\hat{d}_j < 0$. As $\|\alpha^{k+1} - \alpha^k\|^2 = 2\hat{d}_j^2$ from $\hat{d}_i = -\hat{d}_j$, (6.5a) implies that

$$\begin{aligned} F(\alpha^{k+1}) - F(\alpha^k) &< \frac{1}{2}(K_{ii} + K_{jj} - 2K_{ij})\hat{d}_j^2 \\ &= \frac{1}{4}(K_{ii} + K_{jj} - 2K_{ij})\|\alpha^{k+1} - \alpha^k\|^2 \\ &\leq -\frac{\sigma'}{2}\|\alpha^{k+1} - \alpha^k\|^2, \end{aligned} \tag{6.12}$$

where

$$\sigma' \equiv -\max_{t,r}\{\frac{K_{tt} + K_{rr} - 2K_{tr}}{2} \mid K_{tt} + K_{rr} - 2K_{tr} < 0\}. \tag{6.13}$$

Therefore, by defining σ as the minimum of (6.11) and (6.13), the proof is complete. □

Interestingly, if we exactly solve (6.5), so far we have not been able to establish Lemma 3. The reason is that if $\hat{d}_j = H$ is taken, $(-y_i \nabla F(\alpha^k)_i + y_j \nabla F(\alpha^k)_j)\hat{d}_j > 0$ so (6.12) may not be true. A possible way to have it is by modifying the sub-problem (6.1) as shown in (Palagi and Sciandrone 2002). Then the sub-problem is less obvious but Lemma 3 is easily obtained.

Next we give the main convergence result:

Theorem 11 *For the decomposition method using (6.3) for the working set selection and (6.8) for solving the sub-problem, any limit point of $\{\alpha^k\}$ is a local minimum of (1.2).*

Proof.

If we carefully check the proof in (Lin 2001; Lin 2002), it can be extended to non-PSD Q if (1) (6.10) holds and (2) a local minimum of the sub-problem is obtained in each iteration. Now we have (6.10) from Lemma 3. In addition, $\hat{d}_j = L$ is essentially one of the two local minima of problem (6.5) as clearly seen from Figure 4. Thus, the same proof follows. □

7 Discussions

From the results in Sections 3 and 5, we clearly see the importance of the CPD-ness which is directly related to the linear constraint $y^T\alpha = 0$. We suspect that for many non-PSD kernels used so far, their viability is based on it as well as inequality constraints $0 \leq \alpha_i \leq C, i = 1, \ldots, l$ of the dual problem. It is known that some non-PSD kernels are not CPD. For example, the tangent distance kernel matrix in (Haasdonk and Keysers 2002) may contain more than one negative eigenvalue, a property that indicates the matrix is not CPD. Further investigation on such non-PSD kernels and the effect of inequality constraints $0 \leq \alpha_i \leq C$ will be interesting research directions.

Our analysis indicates that for certain parameters the sigmoid kernel behaves like the RBF kernel. Experiments also show that their performance are similar. Therefore, with the result in (Keerthi and Lin 2003) showing that the linear kernel is essentially a special case of the RBF kernel, among existing kernels, RBF should be the first choice for general users.

Acknowledgments

This work was supported in part by the National Science Council of Taiwan via the grant NSC 90-2213-E-002-111. We thank users of LIBSVM (in particular, Carl Staelin) who somewhat force us to study this issue. We also thank Bernhard Schölkopf and Bernhard Haasdonk for some helpful discussions.

References

Berg, C., J. P. R. Christensen, and P. Ressel (1984). *Harmonic Analysis on Semigroups*. New York: Springer-Verlag.

Blake, C. L. and C. J. Merz (1998). UCI repository of machine learning databases. Technical report, University of California, Department of Information and Computer Science, Irvine, CA. Available at `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

Boser, B., I. Guyon, and V. Vapnik (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*.

Burges, C. J. C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery 2*(2), 121–167.

Burges, C. J. C. (1999). Geometry and invariance in kernel based methods. In B. Schölkopf, C. Burges, and A. Smola (Eds.), *Advances in Kernel Methods: Support Vector Learning*, pp. 89–116. MIT Press.

Chang, C.-C. and C.-J. Lin (2001). *LIBSVM: a library for support vector machines*. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

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

DeCoste, D. and B. Schölkopf (2002). Training invariant support vector machines. *Machine Learning 46*, 161–190.

Goldberg, D. (1991). What every computer scientist should know about floating-point arithmetic. *ACM Computing Surveys 23*(1), 5–48.

Haasdonk, B. and D. Keysers (2002). Tangent distance kernels for support vector machines. In *Proceedings of the 16th ICPR*, pp. 864–868.

Hsu, C.-W. and C.-J. Lin (2002). A simple decomposition method for support vector machines. *Machine Learning 46*, 291–314.

Joachims, T. (1998). Making large-scale SVM learning practical. In B. Schölkopf, C. J. C. Burges, and A. J. Smola (Eds.), *Advances in Kernel Methods - Support Vector Learning*, Cambridge, MA. MIT Press.

Keerthi, S. S. and C.-J. Lin (2003). Asymptotic behaviors of support vector machines with Gaussian kernel. *Neural Computation 15*(7). To appear.

Keerthi, S. S., S. K. Shevade, C. Bhattacharyya, and K. R. K. Murthy (2001). Improvements to Platt's SMO algorithm for SVM classifier design. *Neural Computation 13*, 637–649.

Lin, C.-J. (2001). On the convergence of the decomposition method for support vector machines. *IEEE Transactions on Neural Networks 12*(6), 1288–1298.

Lin, C.-J. (2002). Asymptotic convergence of an SMO algorithm without any assumptions. *IEEE Transactions on Neural Networks 13*(1), 248–250.

Lin, K.-M. and C.-J. Lin (2003). A study on reduced support vector machines. *IEEE Transactions on Neural Networks*. To appear.

Micchelli, C. A. (1986). Interpolation of scattered data: distance matrices and conditionally positive definite functions. *Constructive Approximation 2*, 11–22.

Michie, D., D. J. Spiegelhalter, and C. C. Taylor (1994). *Machine Learning, Neural and Statistical Classification.* Englewood Cliffs, N.J.: Prentice Hall. Data available at `http://www.ncc.up.pt/liacc/ML/statlog/datasets.html`.

Osuna, E., R. Freund, and F. Girosi (1997). Training support vector machines: An application to face detection. In *Proceedings of CVPR'97*, New York, NY, pp. 130–136. IEEE.

Osuna, E. and F. Girosi (1998). Reducing the run-time complexity of support vector machines. In *Proceedings of International Conference on Pattern Recognition.*

Palagi, L. and M. Sciandrone (2002). On the convergence of a modified version of SVM^{light} algorithm. Technical report.

Platt, J. C. (1998). Fast training of support vector machines using sequential minimal optimization. In B. Schölkopf, C. J. C. Burges, and A. J. Smola (Eds.), *Advances in Kernel Methods - Support Vector Learning*, Cambridge, MA. MIT Press.

Schölkopf, B. (1997). *Support Vector Learning.* Ph. D. thesis.

Schölkopf, B. (2000). The kernel trick for distances. In *NIPS*, pp. 301–307.

Schölkopf, B. and A. J. Smola (2002). *Learning with kernels.* MIT Press.

Sellathurai, M. and S. Haykin (1999). The separability theory of hyperbolic tangent kernels and support vector machines for pattern classification. In *Proceedings of ICASSP99.*

Vapnik, V. (1995). *The Nature of Statistical Learning Theory.* New York, NY: Springer-Verlag.

A Proof of Theorem 8

The proof of Theorem 8 contains three parts: the convergence of the optimal solution, the convergence of the decision value without the bias term, and the convergence of the bias term. Before entering the proof, we first need to know that (4.5) has a PD kernel under our assumption $x_i \neq x_j$ for all $i \neq j$. Therefore, the optimal solution $\hat{\alpha}^*$ of (4.5) is unique. From now on we denote $\hat{\alpha}^r$ as a local optimal solution of (1.2), and b^r as the associated optimal b value. For (4.5), b^* denotes its optimal b.

1. The convergence of optimal solution:

$$\lim_{r \to -\infty} \theta_r \hat{\alpha}^r = \hat{\alpha}^*, \text{where } \theta_r \equiv 1 + \tanh(r). \tag{A.1}$$

Proof.

By the equivalence between (1.2) and (4.4), $\theta_r \hat{\alpha}^r$ is the optimal solution of (4.4). The convergence to $\hat{\alpha}^*$ comes from (Keerthi and Lin 2003, Lemma 2) since $\bar{Q}$ is PD and the kernel of (4.4) approaches $\bar{Q}$ by Lemma 1. □

2. The convergence of the decision value without the bias term: For any x,

$$\lim_{r \to -\infty} \sum_{i=1}^{l} y_i \hat{\alpha}_i^r \tanh(a x_i^T x + r) = \sum_{i=1}^{l} y_i \hat{\alpha}_i^* e^{2a x_i^T x_j}. \tag{A.2}$$

Proof.

$$\begin{aligned} & \lim_{r \to -\infty} \sum_{i=1}^{l} y_i \hat{\alpha}_i^r \tanh(a x_i^T x + r) \\ = \; & \lim_{r \to -\infty} \sum_{i=1}^{l} y_i \hat{\alpha}_i^r (1 + \tanh(a x_i^T x + r)) \qquad \text{(A.3)} \\ = \; & \lim_{r \to -\infty} \sum_{i=1}^{l} y_i \theta_r \hat{\alpha}_i^r \frac{1 + \tanh(a x_i^T x + r)}{\theta_r} \\ = \; & \sum_{i=1}^{l} y_i \lim_{r \to -\infty} \theta_r \hat{\alpha}_i^r \lim_{r \to -\infty} \frac{1 + \tanh(a x_i^T x + r)}{\theta_r} \\ = \; & \sum_{i=1}^{l} y_i \hat{\alpha}_i^* e^{2a x_i^T x}. \qquad \text{(A.4)} \end{aligned}$$

(A.3) comes from the equality constraint in (1.2) and (A.4) comes from (A.1) and Lemma 1. □

3. The convergence of the bias term:

$$\lim_{r \to -\infty} b^r = b^*. \tag{A.5}$$

Proof.

By the KKT condition that b^r must satisfy,

$$\max_{i \in I_{up}(\hat{\alpha}^r, C)} -y_i \nabla F(\hat{\alpha}^r)_i \leq b^r \leq \min_{i \in I_{low}(\hat{\alpha}^r, C)} -y_i \nabla F(\hat{\alpha}^r)_i,$$

where I_{up} and I_{low} are defined in (6.2). In addition, because b^* is unique,

$$\max_{i \in I_{up}(\hat{\alpha}^*, \tilde{C})} -y_i \nabla F_T(\hat{\alpha}^*)_i = b^* = \min_{i \in I_{low}(\hat{\alpha}^*, \tilde{C})} -y_i \nabla F_T(\hat{\alpha}^*)_i.$$

Note that the equivalence between (1.2) and (4.4) implies $\nabla F(\hat{\alpha}^r)_i = \nabla F_r(\theta_r \hat{\alpha}^r)_i$. Thus,

$$\max_{i \in I_{up}(\theta_r \hat{\alpha}^r, \tilde{C})} -y_i \nabla F_r(\theta_r \hat{\alpha}^r)_i \leq b^r \leq \min_{i \in I_{low}(\theta_r \hat{\alpha}^r, \tilde{C})} -y_i \nabla F_r(\theta_r \hat{\alpha}^r)_i.$$

By the convergence of $\theta_r \hat{\alpha}^r$ when $r \to -\infty$, after r is small enough, all index i's satisfying $\hat{\alpha}_i^* < \tilde{C}$ would have $\theta_r \hat{\alpha}_i^r < \tilde{C}$. That is, $I_{up}(\hat{\alpha}^*, \tilde{C}) \subseteq I_{up}(\theta_r \hat{\alpha}^r, \tilde{C})$. Therefore, when r is small enough,

$$\max_{i \in I_{up}(\hat{\alpha}^*, \tilde{C})} -y_i \nabla F_r(\theta_r \hat{\alpha}^r)_i \leq \max_{i \in I_{up}(\theta_r \hat{\alpha}^r, \tilde{C})} -y_i \nabla F_r(\theta_r \hat{\alpha}^r)_i.$$

Similarly,

$$\min_{i \in I_{low}(\hat{\alpha}^*, \tilde{C})} -y_i \nabla F_r(\theta_r \hat{\alpha}^r)_i \geq \min_{i \in I_{low}(\theta_r \hat{\alpha}^r, \tilde{C})} -y_i \nabla F_r(\theta_r \hat{\alpha}^r)_i.$$

Thus, for $r < 0$ small enough,

$$\max_{i \in I_{up}(\hat{\alpha}^*, \tilde{C})} -y_i \nabla F_r(\theta_r \hat{\alpha}^r)_i \leq b^r \leq \min_{i \in I_{low}(\hat{\alpha}^*, \tilde{C})} -y_i \nabla F_r(\theta_r \hat{\alpha}^r)_i.$$

Taking $\lim_{r \to -\infty}$ on both sides, using Lemma 1 and (A.1),

$$\lim_{r \to -\infty} b^r = \max_{i \in I_{up}(\hat{\alpha}^*, \tilde{C})} -y_i \nabla F_T(\hat{\alpha}^*)_i = \min_{i \in I_{low}(\hat{\alpha}^*, \tilde{C})} -y_i \nabla F_T(\hat{\alpha}^*)_i = b^*. \tag{A.6}$$

□

Therefore, with (A.4) and (A.6), our proof of Theorem 8 is complete.