Support Vector Machines, Data Reduction, and Approximate Kernel Matrices

XuanLong Nguyen[1], Ling Huang[2], and Anthony D. Joseph[3]

[1] SAMSI & Duke University
xuanlong.nguyen@stat.duke.edu
[2] Intel Research
ling.huang@intel.com
[3] UC Berkeley
adj@eecs.berkeley.edu

Abstract. The computational and/or communication constraints associated with processing large-scale data sets using support vector machines (SVM) in contexts such as distributed networking systems are often prohibitively high, resulting in practitioners of SVM learning algorithms having to apply the algorithm on approximate versions of the kernel matrix induced by a certain degree of data reduction. In this paper, we study the tradeoffs between data reduction and the loss in an algorithm's classification performance. We introduce and analyze a consistent estimator of the SVM's achieved classification error, and then derive approximate upper bounds on the perturbation on our estimator. The bound is shown to be empirically tight in a wide range of domains, making it practical for the practitioner to determine the amount of data reduction given a permissible loss in the classification performance.[1]

Keywords: Support vector machines, kernel methods, approximate kernel matrices, matrix perturbation, classification.

1 Introduction

The popularity of using support vector machines (SVM) for classification has led to their application in a growing number of problem domains and to increasingly larger data sets [1,2,3,4]. An appealing key feature of the SVM is that the only interface of the learning algorithm to the data is through its kernel matrix. In many applications, the communication-theoretic constraints imposed by limitations in the underlying distributed data collection infrastructure, or the computational bottleneck associated with a large-scale kernel matrix, naturally requires some degree of data reduction. This means that practitioners usually do not have the resources to train the SVM algorithm on the original kernel matrix. Instead, they must rely on an approximate, often simplified, version of the kernel matrix induced by data reduction.

Consider, for instance, the application of an SVM to a detection task in a distributed networking system. Each dimension of the covariate X represents the data captured by a

[1] The authors would like to thank Michael I. Jordan, Noureddine El Karoui and Ali Rahimi for helpful discussions.

W. Daelemans et al. (Eds.): ECML PKDD 2008, Part II, LNAI 5212, pp. 137–153, 2008.

monitoring device (e.g., network node or sensor), which continuously ships its data to a coordinator for an aggregation analysis using the SVM algorithm. Due to the communication constraints between nodes within the network and the power constraints of each node (e.g., for battery-powered sensors), the monitoring devices do not ship all of their observations to the coordinator; rather, they must appropriately down-sample the data. From the coordinator's point of view, the data analysis (via the SVM or any other algorithm) is not applied to the original data collected by the monitoring devices, but rather to an approximate version. This type of in-network distributed processing protocol has become increasingly popular in various fields, including systems and databases [5,6], as well as in signal processing and machine learning [7,8,9]. In the case where the coordinator uses an SVM for classification analysis, the SVM has access not to the original data set, but rather to only an approximate version, which thus yields an approximate kernel matrix. The amount of kernel approximation is dictated by the amount of data reduction applied by the monitoring devices.

Within the machine learning field, the need for training with an approximate kernel matrix has long been recognized, primarily due to the computational constraints associated with large kernel matrices. As such, there are various methods that have been developed for replacing an original kernel matrix K with a simplified version $\tilde{K}$: matrices with favorable properties such as sparsity, low-rank, etc [10,11,12,13,14].

To our knowledge, there has been very little work focusing on the tradeoffs between the amount of data reduction and the classification accuracy. This issue has only been recently explored in the machine learning community; see [15] for a general theoretical framework. Understanding this issue is important for learning algorithms in general, and especially for SVM algorithms, as it will enable their application in distributed systems, where large streams of data are generated in distributed devices, but not all data can be centrally collected. Furthermore, the tradeoff analysis has to be achieved in simple terms if it is to have impact on practitioners in applied fields.

The primary contribution of this paper is an analysis of the tradeoff between data reduction and the SVM classification error. In particular, we aim to produce simple and practically useful upper bounds that specify the amount of loss of classification accuracy for a given amount of data reduction (to be defined formally). To this end, the contributions are two-fold: (i) First, we introduce a novel estimate, called the *classification error coefficient* C, for the classification error produced by the SVM, and prove that it is a consistent estimate under appropriate conditions. The derivation of this estimator is drawn from the relationship between the hinge loss (used by the SVM) and the 0-1 loss [16]. (ii) Second, using the classification error coefficient C as a surrogate for the classification accuracy, we introduce upper bounds on the change in C given an amount of data reduction. Specifically, let K be the kernel matrix on the original data that we don't have access to, $\tilde{K}$ the kernel matrix induced by data reduction, and suppose that each element of $\Delta = \tilde{K} - K$ has variance bounded by σ^2. Let $\tilde{C}$ be the classification error coefficient associate to $\tilde{K}$. We express an upper bound of $\tilde{C} - C$ in terms of σ and matrix $\tilde{K}$. The bound is empirically shown to be remarkably tight for a wide range of data domains, making it practical for the practitioner of the SVM to determine the amount of data reduction given a permissible loss in the classification performance.

The remainder of the paper is organized as follows: in Section 2, we provide background information about the SVM algorithm, and describe the contexts that motivate the need for data reduction and approximate kernel matrices; in Section 3, we describe the main results of this paper, starting with a derivation and consistency analysis of the classification error coefficient C, and then presenting upper bounds on the change of C due to kernel approximation; in Section 4, we present an empirical evaluation of our analyses; and in Section 5, we discuss our conclusions.

2 SVM, Data Reduction and Kernel Matrix Approximation

2.1 SVM Background

In a classification algorithm, we are given as our training data m i.i.d. samples $(x_i, y_i)_{i=1}^m$ in $\mathcal{X} \times \{\pm 1\}$, where $\mathcal{X}$ denotes a bounded subset of R^d. A classification algorithm involves finding a discriminant function $y = \mathrm{sign}(f(x))$ that minimizes the classification error $P(Y \neq \mathrm{sign}(f(X)))$.

Central to a kernel-based SVM classification algorithm is the notion of a kernel function $K(x, x')$ that provides a measure of similarity between two data points x and x' in $\mathcal{X}$. Technically, K is required to be a symmetric positive semidefinite kernel. For such a function, Mercer's theorem implies that there must exist a reproducing kernel Hilbert space $\mathcal{H} = \mathrm{span}\{\Phi(x) | x \in \mathcal{X}\}$ in which K acts as an inner product, i.e., $K(x, x') = \langle \Phi(x), \Phi(x') \rangle$. The SVM algorithm chooses a linear function in this feature space $f(x) = \langle \mathrm{w}, \Phi(x) \rangle$ for some w that minimizes the regularized training error:

$$\min_{\mathrm{w} \in \mathcal{H}} \frac{1}{m} \sum_{i=1}^{m} \phi(y_i f(x_i)) + \lambda_m \|\mathrm{w}\|^2 / 2. \tag{1}$$

Here λ_m denotes a regularization parameter, and ϕ denotes an appropriate loss function that is a convex surrogate to the 0-1 loss $\mathbb{I}(y \neq \mathrm{sign}(f(x)))$. In particular, the SVM uses hinge loss $\phi(yf(x)) = (1 - yf(x))_+$ [3]. It turns out that the above optimization has the following dual formulation in quadratic programming:

$$\max_{0 \leq \alpha \leq 1} \frac{1}{m} \sum_i \alpha_i - \frac{1}{2m^2 \lambda_m} \sum_{i,j} \alpha_i \alpha_j y_i y_j K(x_i, x_j). \tag{2}$$

For notational convenience, we define matrix Q such that $Q_{ij} = K(x_i, x_j) y_i y_j$. The solution α of the above dual formulation defines the optimal f and w of the primal formulation via the following:

$$\mathrm{w} = \frac{1}{m \lambda_m} \sum_{i=1}^{m} \alpha_i \Phi(x_i) \tag{3}$$

$$f(x) = \frac{1}{m \lambda_m} \sum_{i=1}^{m} \alpha_i K(x_i, x). \tag{4}$$

2.2 In-Network Data Reduction and Approximate Kernel Matrices

As seen from the dual formulation (2), the kernel matrix $K = \{K(x_i, x_j)_{i,j}\}$ and the label vector $y = [y_1, \ldots, y_m]$ form sufficient statistics of the SVM. However, there is substantial previous work that focuses on the application of an SVM to an approximate version $\tilde{K}$ of the kernel matrix from data reduction. We extend this work to focus in particular on the application of SVM in distributed system environments.

Suppression of data streams and data quantization in distributed systems. A primary motivation regarding this work is the application of SVM-based classification analysis to distributed settings in a number of fields, including databases, distributed systems, and sensor networks [5,6,9]. In a distributed system setting, there are d monitoring devices which receive streams of raw data represented by a d-dimensional covariate X and send the data to a central coordinator for classification analysis. Because of communication constraints, each monitoring devices cannot send all its received data; instead, they must send as little data as possible. An ϵ-suppression algorithm is frequently used: each monitoring devices j, $j = 1, \ldots, d$, send the i-th data point to the coordinator only if: $|X_i^j - X_{i-1}^j| > \epsilon$. Using these values, the coordinator reconstructs an approximate view $\tilde{X}$ of the true data X, such that $\|X - \tilde{X}\|_\infty \leq \epsilon$. A key question in the design of such systems is how to determine the data reduction parameter ϵ, given a permissible level of loss in the classification accuracy.

In signal processing, data reduction is achieved by quantization or binning: each dimension of X is discretized into a given number of bins before being sent to the central coordinator [7,8]. The bin size is determined by the number of bits available for transmission: for bins of equal size ϵ, the number of bins is proportional to $1/\epsilon$, corresponding to using $\log(1/\epsilon)$ number of bits. As before, the coordinator receives an approximate version $\tilde{X}$, such that $\|X - \tilde{X}\|_\infty \leq \epsilon$. Once $\tilde{X}$ is received by the coordinator, one obtains an approximate kernel matrix by applying the kernel function K to $\tilde{X}$. Suppose that a Gaussian kernel with width parameter $\omega > 0$ is used, then we obtain the approximate kernel $\tilde{K}$ as $\tilde{K}(\tilde{X}_i, \tilde{X}_j) = \exp\left(-\frac{\|\tilde{X}_i - \tilde{X}_j\|^2}{2\omega^2}\right)$.

Kernel matrix sparsification and approximation. Beside applications in in-network and distributed data processing, a variety of methods have been devised to approximate a large kernel matrix by a more simplified version with desirable properties, such as sparsity and low-rank (e.g., [10,11,12,13,14]). For instance, [17] proposes a simple method to approximate K by randomly zeroing out its entries:

$$\tilde{K}_{ij} = \tilde{K}_{ji} = \begin{cases} 0 & \text{with probability } 1 - 1/\delta, \\ \delta K_{ij} & \text{with probability } 1/\delta, \end{cases}$$

where $\delta \geq 1$ controls the degree of sparsification on the kernel.[2] This sparsification was shown to greatly speed up the construction and significantly reduce the space required

[2] This method may not retain the positive definiteness of the kernel matrix, in which case positive values have to be added to the matrix diagonal.

to store the matrix. Our analysis can also be applied to analyze the tradeoff of kernel approximation error and the change in classification error.

3 Classification Error Coefficient and Effects of Data Reduction

We begin by describing the set-up of our analysis. Let $\tilde{K}$ be a (random) kernel matrix that is an approximate version of kernel matrix K induced by a data reduction scheme described above (e.g., quantization or suppression). Let C_0 and $\tilde{C}_0$ be the (population) classification error associated with the SVM classifier trained with kernel matrix K and $\tilde{K}$, respectively. We wish to bound $|\tilde{C}_0 - C_0|$ in terms of the "magnitude" of the error matrix $\Delta = \tilde{K} - K$, which we now define. For a simplified analysis, we make the following assumption about the error matrix Δ:

A0. Conditioned on $\tilde{K}$ and y, all elements e_{ij} $(i, j = 1, \ldots, m; i \neq j)$ of Δ are uncorrelated, have zero mean, and the variance bounded by σ^2.

We use σ to control the degree of our kernel matrix approximation scheme, abstracting away from further detail. It is worth noting that certain kernel matrix approximation schemes may not satisfy the independence assumption. On the one hand, it is possible to incorporate the correlation of elements of Δ into our analysis. On the other hand, we find that the correlation is typically small, such that elaboration does not significantly improve our bounds in most cases.

Our ultimate goal is to produce practically useful bounds on $\tilde{C}_0 - C_0$ in terms of σ and kernel matrix $\tilde{K}$. This is a highly nontrivial task, especially since we have access only to approximate data (through $\tilde{K}$, but not K).

3.1 Classification Error Coefficient

In order to quantify the effect on the population SVM classification error C_0, we first introduce a simple estimate of C_0 from empirical data. In a nutshell, our estimator relies on the following intuitions:

1. The SVM algorithm involves minimizing over a surrogate loss (the hinge loss), while we are interested in the performance in terms of 0-1 loss. Thus, we need to be able to compare between these two losses.
2. We are given only empirical data, and we replace the risk (population expectation of a loss function) by its empirical version.
3. We avoid terms that are "nonstable" for the choice of learning parameters, which is important for our subsequent perturbation analysis.

The first key observation comes from the fact that the optimal expected ϕ-risk using the hinge loss is shown to be twice the optimal Bayes error (i.e., using 0-1 loss) (cf. [16], Sec. 2.1):

$$\min_{f \in \mathcal{F}} P(Y \neq f(X)) = \frac{1}{2} \min_{f \in \mathcal{F}} \mathrm{E}\phi(Y f(X)), \tag{5}$$

where $\mathcal{F}$ denotes an arbitrary class of measurable functions that contains the optimal Bayes classifier.

Note that we can estimate the optimal expected ϕ-risk by its empirical version defined in Eqn. (1), which equals its dual formulation (2). Let $\hat{\mathrm{w}}$ be the solution of (1). As shown in the proof of Theorem 1, if $\lambda_m \to 0$ sufficiently slowly as $m \to \infty$, the penalty term $\lambda_m \|\hat{\mathrm{w}}\|^2$ vanishes as $m \to \infty$. Due to (3), the second quantity in the dual formulation (2) satisfies

$$\frac{1}{2m^2\lambda_m} \sum_{i,j} \alpha_i \alpha_j y_i y_j K(x_i, x_j) = \lambda_m \|\hat{\mathrm{w}}\|^2/2 \to 0.$$

As a result, we have:

$$\inf_{\mathrm{w}\in\mathcal{H}} \hat{\mathrm{E}}\phi(Yf(X)) + \lambda_m \|\mathrm{w}\|^2/2 = \frac{1}{m}\sum_{i=1}^{m} \alpha_i - \lambda_m \|\hat{\mathrm{w}}\|^2/2. \tag{6}$$

Approximating the optimal ϕ-risk in (5) by its empirical version over $\mathcal{H}$, and dropping off the vanishing term $\lambda_m \|\hat{\mathrm{w}}\|^2$ from Eqn. (6), we obtain the following estimate:

Definition 1. *Let α be the solution of the SVM's dual formulation* (2)*, the following quantity is called the* **classification error coefficient:**

$$C = \frac{1}{2m}\sum_{i=1}^{m} \alpha_i. \tag{7}$$

An appealing feature of C is that $C \in [0, 1/2]$. Furthermore, it is a simple function of α. As we show in the next section, this simplicity significantly facilitates our analysis of the effect of kernel approximation error. Applying consistency results of SVM classifiers (e.g., [18]) we can show that C is also a universally consistent estimate for the optimal classification error under appropriate assumptions. These assumptions are:

A1. K is a universal kernel on $\mathcal{X}$, i.e., the function class $\{\langle \mathrm{w}, \Phi(\cdot)\rangle | \mathrm{w} \in \mathcal{H}\rangle\}$ is dense in the space of continuous functions on $\mathcal{X}$ with respect to the sup-norm (see [18] for more details). Examples of such kernels include the Gaussian kernel $K(x, x') = \exp\left(-\frac{\|x-x'\|^2}{2\omega^2}\right)$, among others.

A2. $\lambda_m \to 0$ such that $m\lambda_m \to \infty$.

Theorem 1. *Suppose that $(X_i, Y_i)_{i=1}^m$ are drawn i.i.d. from a Borel probability measure P. Under assumptions* A1 *and* A2*, there holds as $m \to \infty$:*

$$C - \inf_{f\in\mathcal{F}} P(Y \neq f(X)) \to 0 \ \textit{ in probability.}$$

See the appendix for a proof. It is worth noting that this result is kernel-independent.

Let $\tilde{K}, \tilde{\alpha}, \tilde{f}, \tilde{C}$ denote the corresponding counterparts for kernel matrix K, the dual formulation's solutions α, classifier f, and the classification coefficient C, respectively. For the data suppression and quantization setting described in Section 2, suppose that a universal kernel (such as Gaussian kernel) is applied to both original and approximate data. By Theorem 1, both C and $\tilde{C}$ are consistent estimates of the classification error

applied on original and approximate data, respectively. Thus, the difference $\tilde{C} - C$ can be used to evaluate the loss of classification accuracy of the SVM. This is the focus of the next section. [3]

3.2 Effects of Data Reduction on Classification Error Coefficient

In this section, we analyze the effects of the approximation of the kernel matrix $K - \tilde{K}$ on the classification error coefficient difference $C - \tilde{C}$.

Let $r = \#\{i : \alpha_i \neq \tilde{\alpha}_i\}$. From Eqn. (7), the difference of the classification coefficients is bounded via Cauchy-Schwarz inequality:

$$|\tilde{C} - C| \leq \frac{1}{2m}\|\tilde{\alpha} - \alpha\|_1 \leq \frac{1}{2m}\sqrt{r}\|\tilde{\alpha} - \alpha\|, \tag{8}$$

from which we can see the key point lies in deriving a tight bound on the L_2 norm $\|\tilde{\alpha} - \alpha\|$. Define two quantities:

$$R_1 = \frac{\|\tilde{\alpha} - \alpha\|^2}{(\tilde{\alpha} - \alpha)^T Q(\tilde{\alpha} - \alpha)}, \quad R_2 = \frac{(\tilde{\alpha} - \alpha)^T(Q - \tilde{Q})\tilde{\alpha}}{\|\tilde{\alpha} - \alpha\|}.$$

Proposition 1. *If α and $\tilde{\alpha}$ are the optimal solution of the program* (2) *using kernel matrix K and $\tilde{K}$ respectively, then:*

$$|\tilde{C} - C| \leq \frac{\sqrt{r}}{2m}\|\tilde{\alpha} - \alpha\| \leq \frac{\sqrt{r}}{2m}R_1R_2.$$

For a proof, see the Appendix. Although it is simple to derive rigorous absolute bounds on R_1 and R_2, such bounds are not practically useful. Indeed, R_1 is upper bounded by the inverse of the smallest eigenvalue of Q, which tends to be very large. An alternative solution is to obtain probabilistic bounds that hold with high probability, using Prop. 1 as a starting point. Note that given a data reduction scheme, there is an induced joint distribution generating kernel matrix K, its approximate version $\tilde{K}$, as well as the label vector y. Matrix $Q = K \circ yy^T$ determines the value of vector α through an optimization problem (2). Likewise, $\tilde{Q} = \tilde{K} \circ yy^T$ determines $\tilde{\alpha}$. Thus, α and $\tilde{\alpha}$ are random under the distribution that marginalizes over random matrices Q and $\tilde{Q}$, respectively.

The difficult aspect of our analysis lies in the fact that we do not have closed forms of either α or $\tilde{\alpha}$, which are solutions of quadratic programs parameterized by Q and

[3] We make several remarks: (i) The rates at which C and $\tilde{C}$ converge to the respective misclassification rate may not be the same. To understand this issue one has to take into account additional assumptions on both the kernel function, and the underlying distribution P. (ii) Although quantization of data does not affect the consistency of the classification error coefficient since one can apply the same universal kernel function to quantized data, quantizing/approximating *directly* the kernel matrix (such as those proposed in [17] and described in Sec. 2) may affect both consistency and convergence rates in a nontrivial manner. An investigation of approximation rates of the *quantized/sparsified* kernel function class is an interesting open direction.

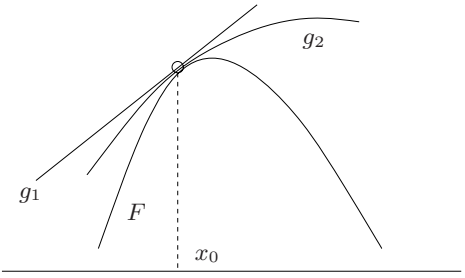

Fig. 1. Illustration of upper bounds via perturbation analysis: Linear approximation g_1 and upper bound g_2 via second-order perturbation analysis of a concave function F around a point x_0 in the domain. The bounds continue to hold for large perturbation around x_0.

$\tilde{Q}$, respectively. We know a useful fact, however, regarding the distributions of vector α and $\tilde{\alpha}$. Since the training data are i.i.d., the roles of α_i and $\tilde{\alpha}_i$ for $i = 1, \ldots, m$ are equivalent. Thus $(\alpha_i, \tilde{\alpha}_i)$ have marginally identical distributions for $i = 1, \ldots, m$.

We first motivate our subsequent perturbation analysis by an observation that the optimal classification error defined by Eq. (5), for which C is an estimate, is a concave function with respect jointly to the class probability distributions $(P(X|Y = 1), P(X|Y = -1))$ (cf. [16], Sec. 2). When the data is perturbed (e.g., via quantization/suppression) the joint distribution $(P(X|Y = 1), P(X|Y = -1))$ is also perturbed. Intuitively, upper bounds for a concave functional via either linear or second-order approximation under a small perturbation on its variables should also hold under larger perturbations, even if such bounds tend to be less tight in the latter situation. See Fig. 3.2 for an illustration. Thus, to obtain useful probabilistic bounds on $\tilde{C} - C$, we restrict our analysis to the situation where $\tilde{K}$ is a small perturbation from the original matrix K. Under a small perturbation, the following assumptions can be made:

B1. The random variables $\tilde{\alpha}_i - \alpha_i$ for $i = 1, \ldots, n$ are non-correlated.
B2. The random variables $\tilde{\alpha}_i - \alpha_i$ have zero means.

Given Assumption B1, coupled with the fact that $(\tilde{\alpha}_i - \alpha_i)$ have identical distributions for $i = 1, \ldots, m$, by the central limit theorem, as m gets large, a rescaled version of $\tilde{C} - C$ behaves like a standard normal distribution. Using a result for standard normal random variables, for any constant $t > 0$, we obtain that with probability at least $1 - \frac{1}{\sqrt{2\pi}t} e^{-t^2/2}$:

$$\tilde{C} - C \lesssim t\sqrt{\mathrm{Var}(\tilde{C} - C)} + \mathrm{E}(\tilde{C} - C) \overset{Ass.(B1)}{=} \frac{t}{2m}\sqrt{\sum_{i=1}^{m} \mathrm{Var}(\tilde{\alpha}_i - \alpha_i)} + \mathrm{E}(\tilde{C} - C)$$

$$\leq \frac{t}{2m}\sqrt{\mathrm{E}\|\alpha - \tilde{\alpha}\|^2} + \mathrm{E}(\tilde{C} - C) \overset{Prop.\ 1}{\leq} \frac{t}{2m}\sqrt{\mathrm{E}R_1^2 R_2^2} + \mathrm{E}(\tilde{C} - C).$$

Our next step involves an observation that under certain assumptions to be described below, random variable R_1 is tightly concentrated around a constant, and that $\mathrm{E}R_2^2$ can be easily bounded.

$$R_1 \approx \frac{m}{\mathrm{tr}(K)} \quad \text{by Lemma 2} \tag{9}$$

$$\mathrm{E}R_2^2 \leq \sigma^2 m \mathrm{E}\|\tilde{\alpha}\|^2 \quad \text{by Lemma 1.} \tag{10}$$

As a result, we obtain the following approximate bound:

$$\tilde{C} - C \lesssim t\frac{\sigma\sqrt{m\mathrm{E}\|\tilde{\alpha}\|^2}}{2\mathrm{tr}(K)} + \mathrm{E}(\tilde{C} - C) \overset{Ass.B2}{=} t\frac{\sigma\sqrt{m\mathrm{E}\|\tilde{\alpha}\|^2}}{2\mathrm{tr}(K)}, \tag{11}$$

where Eqn. (11) is obtained by invoking Assumption B2.

Suppose that in practice we do not have access to K, then $\mathrm{tr}(K)$ can be approximated by $\mathrm{tr}(\tilde{K})$. In fact, for a Gaussian kernel $\mathrm{tr}(K) = \mathrm{tr}(\tilde{K}) = m$. One slight complication is estimating $\mathrm{E}\|\tilde{\alpha}\|^2$. Since we have only one training sample for $\tilde{K}$, which induces a single sample for $\tilde{\alpha}$, this expectation is simply estimated by $\|\tilde{\alpha}\|^2$.

When we choose $t = 1$ in bound (11), the probability that the bound is correct is approximately $1 - e^{-1/2}/\sqrt{2\pi} = 75\%$. For $t = 2$, the probability improves to $1 - e^{-2}/2\sqrt{2\pi} = 97\%$. While $t = 1$ yields relatively tighter bound, we choose $t = 2$ in practice. In summary, we have obtained an approximate bound:

$$\text{classif. coeff. (approx. data)} \leq \text{classif. coeff. (original data)} + \frac{\sigma\sqrt{m\|\tilde{\alpha}\|^2}}{\mathrm{tr}(\tilde{K})} \tag{12}$$

Remark. (i) Even though our analysis is motivated by the context of small perturbations to the kernel matrix, bound (12) appears to hold up well in practice when σ is large. This agrees with our intuition on the concavity of (5) discussed earlier. (ii) Our analysis is essentially that of second-order matrix perturbation which requires the perturbation be small so that both Assumptions B1 and B2 hold. Regarding B1, for $i = 1, \ldots, m$, each pair $(\alpha_i, \tilde{\alpha}_i)$ corresponds to the i-th training data point, which is drawn i.i.d. As a result, $(\alpha_i - \tilde{\alpha}_i)$ are very weakly correlated with each other. We show that this is empirically true through a large number of simulations. (ii) Assumption B2 is much more stringent by comparison. When $\tilde{K}$ is only a small perturbation of matrix K, we have also found through simulations that this assumption is very reasonable, especially in the contexts of the data quantization and kernel sparsification methods described earlier.

3.3 Technical Issues

Probabilistic bounds of R_1 and R_2. Here we elaborate on the assumptions under which the probabilistic bounds for R_1 and R_2 are obtained, which motivate the approximation method given above. Starting with R_2, it is simple to obtain:

Lemma 1. *Under Assumption* A0, $\mathrm{E}R_2 \leq \sqrt{\mathrm{E}R_2^2} \leq \sigma\sqrt{m\mathrm{E}[\|\tilde{\alpha}\|^2|]}$.

See the Appendix for a proof. Turning to the inverse of Raleigh quotient term R_1, our approximation is motivated by the following fact, which is a direct consequence of Thm 2.2. of [19]:

Lemma 2. *Let A be fixed $m \times m$ symmetric positive definite matrix with* bounded *eigenvalues $\lambda_1, \ldots, \lambda_m$, and z be an m-dim random vector drawn from any spherically symmetric distribution,*

$$\mathrm{E}[z^T Az/\|z\|^2] = \mathrm{tr}(A)/m$$
$$\mathrm{Var}[z^T Az/\|z\|^2] = \frac{2}{m+2}\left(\sum_{i=1}^{m} \lambda_i^2/m - (\sum_{i=1}^{m} \lambda_i/m)^2\right).$$

By this result, $z^T Az/\|z\|^2$ has vanishing variance as $m \to \infty$. Thus, this quantity is tightly concentrated around its mean. Note that if $\mathrm{tr}(A)/m$ is bounded away from 0, we can also approximate $1/z^T Az$ by $m/\mathrm{tr}(A)$. This is indeed the situation with most kernels in practice: As m becomes large, $\mathrm{tr}(\tilde{K})/m \to \mathrm{E}\tilde{K}(X, X) > 0$. As a result, we obtain approximation (9).

It is worth noting that the "goodness" of this heuristic approximation relies on the assumption that $\alpha - \tilde{\alpha}$ follows an approximately spectrally symmetric distribution. On the other hand, the concentration of the Raleigh quotient term also holds under more general conditions (cf. [20]). An in-depth analysis of such conditions on α and $\tilde{\alpha}$ is beyond the scope of this paper.

3.4 Practical Issues

The bound we derived in Eqn. (12) is readily applicable to practical applications. Recall from Section 2 the example of the detection task in a distributed networking system using a SVM. Each monitoring device independently applies a quantization scheme on their data before sending to the coordinator. The size of the quantized bin is ϵ. Equivalently, one could use an ϵ-suppression scheme similar to [9]. The coordinator (e.g., network operation center) has access only to approximate data $\tilde{X}$, based on which it can compute $\tilde{C}, \tilde{K}, \tilde{\alpha}$ by applying a SVM on $\tilde{X}$. Given ϵ, one can estimate the amount of kernel matrix approximation error σ and vice versa (see, e.g., [9]). Thus, Eqn. (12) gives the maximum possible loss in the classification accuracy due to data reduction. The tightness of bound (12) is crucial: it allows the practitioner to tune the data reduction with good a confidence on the detection performance of the system. Conversely, suppose that the practitioner is willing to incur a loss of classification accuracy due to data reduction by an amount at most δ. Then, the appropriate amount of kernel approximation due to data reduction is:

$$\sigma^* = \frac{\delta \cdot \mathrm{tr}(\tilde{K})}{\sqrt{m\|\tilde{\alpha}\|^2}}. \tag{13}$$

4 Evaluation

In this section, we present an empirical evaluation of our analysis on both synthetic and real-life data sets. For exhaustive evaluation of the behavior of the classification error coefficient C and the tradeoff analysis captured by bound (12), we replicate our experiments on a large number of of synthetic data sets of different types in moderate dimensions; for illustration in two dimensions, see Fig 2. To demonstrate the practical

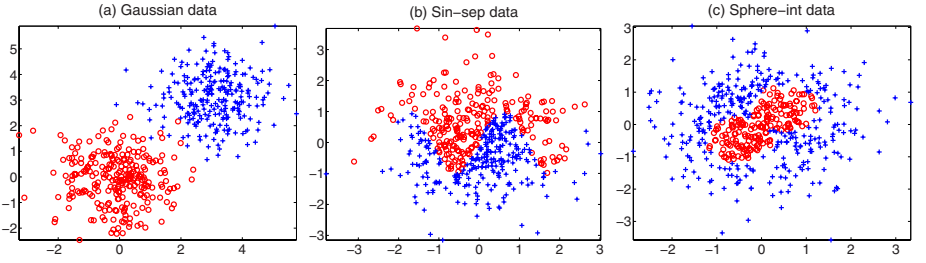

Fig. 2. Synthetic data sets illustrated in two dimensions

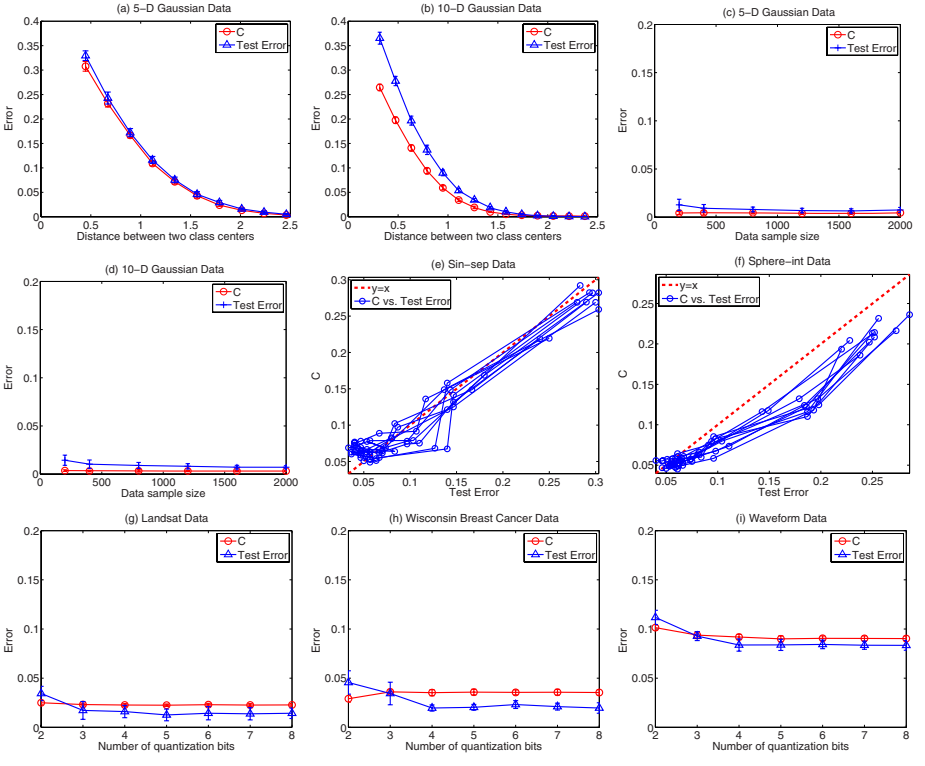

Fig. 3. Comparison between C and the test error under varying conditions: (a–b) varying amount of overlap between two classes (both training and test data sets have 2,000 sample points. Error bars are derived from 25 replications); (c–d) varying sample sizes; (e–f) varying amount of data reduction via scatter plots (each path in the scatter plots connects points corresponding to varying number of quantization bits ranging from 8 in low-left corner to 2 bits in upper-right corner); (g–i) varying amount of data reduction via error bar plots. All plots show C remains a good estimate of the test error even with data reduction. We use Gaussian kernels for all experiments.

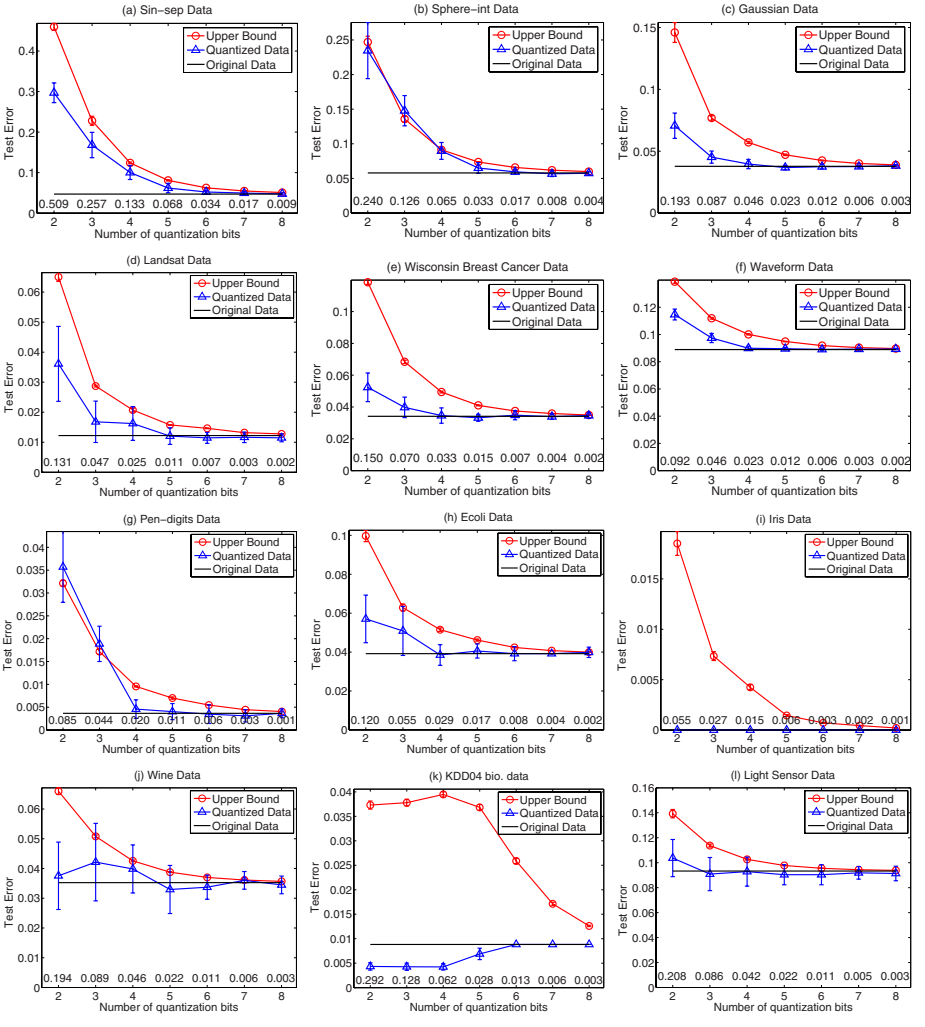

Fig. 4. Upper bounds of test error on approximate data due to quantization using bound (12). (a–c) Simulated data sets with 2, 5, 10 features, respectively; (d) Landsat satellite data (6435 sample size, 36 features); (e) Wisconsin breast cancer data (569 sample size, 30 features); (f) Waveform data (5000 sample size, 21 features); (g) Pen-Based recognition of digits data (10992 sample size, 16 features); (h) Ecoli data (336 sample size, 8 features). (i) Iris data (150 sample size, 4 features); (j) Wine data (178 sample size, 13 features); (k) KDD04 Bio data (145K sample size, 74 features); (l) Intel Lab light sensor data (81 sample size, 25 features). We use Gaussian kernels for (a–i), and linear kernels for (j–l). The x-axis shows increased bit numbers and the correspondingly decreasing matrix error σ.

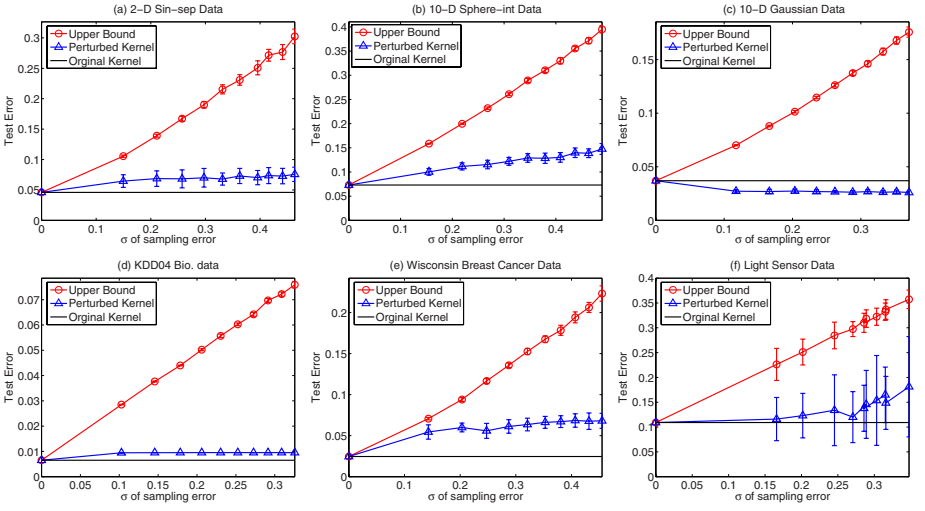

Fig. 5. Upper bounds of test error based on bound (12) using approximate Gaussian kernel matrices obtained from kernel sparsification sampling. (a–c) Simulated data sets; (d) KDD04 Bio data; (e) Wisconsin breast cancer data; (f) Intel Lab light sensor data. We use Gaussian kernels for all experiments. The x-axis shows the increasing matrix error σ due to down sampling on the kernel matrix.

usefulness of our analysis, we have tested (12) on nine real-life data sets (from the UCI repository [21] and one light sensor data set from the IRB Laboratory [7]), which are subject to varying degrees of data reduction (quantization bits). The data domains are diverse, including satellite images, medicine, biology, agriculture, handwritten digits, and sensor network data, demonstrating the wide applicability of our analysis.

Evaluation of estimate C. The first set of results in Fig. 3 verify the relationship between the classification error coefficient C and test error on held-out data under varying conditions on: (i) overlap between classification classes (subfigs (a–b)), (ii) sample sizes (subfigs (c–d)) and (iii) amount of data reduction (subfigs (e–i)). It is observed that C estimates the test error very well in all such situations for both simulated and real data sets, and even when the misclassification rate is high (i.e. noisy data). In particular, Fig. 3 (e)(f) show scatter plots comparing C against test error. Each path connects points corresponding to varying amount of data reduction on the same data set. They are very closely parallel to the $y = x$ line, with the points in the upper-right corner corresponding to the most severe data reduction.

Effects of data reduction on test error. Next, we evaluate the effect of data reduction via quantization (suppression). Fig. 4 plots the misclassification rate for data sets subject to varying degree of quantization, and the upper bound developed in this paper. Our bound is defined as a sum of test error on original (non-quantized) data set plus the upper bound of $\tilde{C} - C$ provided by (12). As expected, the misclassification rate increases as one decreases the number of quantization bits. What is remarkable is that our upper

bound on the approximate data set is very tight in most cases. The effectiveness of our bound should allow the practitioner to determine the right amount of quantization bits given a desired loss in classification accuracy.

It is worth highlighting that although our bound was derived using the viewpoint of (small) stochastic perturbation analysis (i.e., σ is small, and number of quantization bits is large), in most cases the bound continues to hold up for large σ (and small number of bits), even if it becomes less tight. This strengthens our intuition based on the concavity of the optimal Bayes error. Note also that under small perturbation (small σ) the mean of difference of test error in original data and approximate data is very close to 0. This provides a strong empirical evidence for the validity of Assumption B2.

We also applied our analysis to study the tradeoff between kernel approximation and classification error in the context of kernel sparsification sampling described in Section 2. The bounds are still quite good, although they are not as tight as in data quantization (see Fig. 5). Note that in one case (subfig (c)), the classification error actually decreases as the kernel becomes sparser, but our upper bound fails to capture such phenomenon. This is because in contrast to data reduction methods, direct approximation schemes on the kernel matrix may influence the approximation error rate of the induced kernel function class in a nontrivial manner. This aspect is not accounted for by our classification error coefficient C (see remarks following Theorem 1).

5 Conclusion

In this paper, we studied the tradeoff of data reduction and classification error in the context of the SVM algorithm. We introduced and analyzed an estimate of the test error for the SVM, and by adopting a viewpoint of stochastic matrix perturbation theory, we derived approximate upper bounds on the test error for the SVM in the presence of data reduction. The bound's effectiveness is demonstrated in a large number of synthetic and real-world data sets, and thus can be used to determine the right amount of data reduction given a permissible loss in classification accuracy in applications. Our present analysis focuses mainly on the effect of data reduction on the classification error estimate C while ignoring the its effect on approximability and the approximation rate of the quantized (or sparsified) kernel function class. Accounting for the latter is likely to improve the analysis further, and is an interesting open research direction.

References

1. Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning 20(3), 273–297 (1995)
2. Joachims, T.: Making large-scale SVM learning practical. In: Schölkopf, B., Burges, C., Smola, A. (eds.) Advances in Kernel Methods – Support Vector Learning. MIT Press, Cambridge (1999)
3. Schölkopf, B., Smola, A.: Learning with Kernels. MIT Press, Cambridge (2002)
4. Shawe-Taylor, J., Cristianini, N.: Kernel methods for pattern analysis. Cambridge University Press, Cambridge (2004)
5. Keralapura, R., Cormode, G., Ramamirtham, J.: Communication-efficient distributed monitoring of thresholded counts. In: Proceedings of ACM SIGMOD (2006)

6. Silberstein, A., Gelfand, G.P.A., Munagala, K., Yang, J.: Suppression and failures in sensor networks: A bayesian approach. In: Proceedings of VLDB (2007)
7. Nguyen, X., Wainwright, M.J., Jordan, M.I.: Nonparametric decentralized detection using kernel methods. IEEE Transactions on Signal Processing 53(11), 4053–4066 (2005)
8. Shi, T., Yu, B.: Binning in Gaussian kernel regularization. Statist. Sinic. 16, 541–567 (2005)
9. Huang, L., Nguyen, X., Garofalakis, M., Joseph, A.D., Jordan, M.I., Taft, N.: In-network PCA and anomaly detection. In: Advances in Neural Information Processing Systems (NIPS) (2006)
10. Fine, S., Scheinberg, K.: Efficient SVM training using low-rank kernel representations. Journal of Machine Learning Research 2, 243–264 (2001)
11. Smola, A., Schölkopf, B.: Sparse greedy matrix approximation for machine learning. In: Proceedings of the 17th International Conference on Machine Learning (2000)
12. Williams, C., Seeger, M.: Using the Nyström method to speed up kernel machines. In: Advances in Neural Information Processing Systems (NIPS) (2001)
13. Yang, C., Duraiswami, R., Davis, L.: Efficient kernel machines using the improved fast gauss transform. In: Advances in Neural Information Processing Systems (NIPS) (2004)
14. Rahimi, A., Recht, B.: Random features for large-scale kernel machines. In: Advances Neural Information Processing Systems (NIPS) (2007)
15. Bottou, L., Bousquet, O.: The tradeoffs of large scale learning. In: Advances in Neural Information Processing Systems (NIPS) (2007)
16. Nguyen, X., Wainwright, M.J., Jordan, M.I.: On surrogate loss functions and f-divergences. Annals of Statistics (to appear, 2008)
17. Achlioptas, D., McSherry, F., Schölkopf, B.: Sampling techniques for kernel methods. In: Advances in Neural Information Processing Systems (NIPS) (2001)
18. Steinwart, I.: Consistency of support vector machines and other regularized kernel classifiers. IEEE Trans. on Information Theory 51, 128–142 (2005)
19. Böttcher, A., Grudsky, S.: The norm of the product of a large matrix and a random vector. Electronic Journal of Probability 8, 1–29 (2003)
20. Ledoux, M.: The concentration of measure phenomenon. AMS Society (2001)
21. Asuncion, A., Newman, D.: UCI Machine Learning Repository, Department of Information and Computer Science (2007), http://www.ics.uci.edu/ mlearn/MLRepository.html
22. Bartlett, P., Mendelson, S.: Gaussian and Rademacher complexities: Risk bounds and structural results. Journal of Machine Learning Research 3, 463–482 (2002)

Appendix

Proof sketch of Theorem 1: Let $R_m(f) := \frac{1}{m}\sum_{i=1}^{m}\phi(Yf(X))$, $R(f) := \mathrm{E}\phi(Yf(X))$, and let $I(f) = \|f\|_{\mathcal{H}}$ for any $f \in \mathcal{H}$. To signify the dependence on sample size m we shall use f_m in this proof to denote the SVM classifier defined by (4). The primal form (1) can be re-written as

$$f_m = \mathrm{argmin}_{f\in\mathcal{H}} R_m(f) + \lambda_m I(f)^2/2.$$

The classification error coefficient can be expressed by:

$$C = \frac{1}{2}(R_m(f) + \lambda_m I(f_m)^2).$$

K being a universal kernel implies that $\inf_{\mathrm{w}\in\mathcal{H}} R(f) = \min_{f\in\mathcal{F}} R(f)$ (cf. [18], Prop. 3.2). For arbitrary $\epsilon > 0$, let $f_0 \in \mathcal{H}$ such that $R(f_0) \leq \min_{f\in\mathcal{F}} R(f) + \epsilon$.

By the definition of f_m, we obtain that $R_m(f_m) + \lambda_m I(f_m)^2/2 \leq R_m(0) + \lambda_m I(0)^2/2 = R_m(0) = \phi(0)$, implying that $I(f_m) = O(1/\sqrt{\lambda_m})$. We also have:

$$R_m(f_m) + \lambda_m I(f_m)^2/2 \leq R_m(f_0) + \lambda_m I(f_0)^2/2.$$

Rearranging gives:

$$R(f_m) - R(f_0) \leq (R(f_m) - R_m(f_m)) + (R_m(f_0) - R(f_0)) + \lambda_m (I(f_0)^2 - I(f_m)^2/2.$$

Now, note that for any $B > 0$, if $I(f) \leq B$, then $f(x) = \langle \mathrm{w}, \Phi(x)\rangle \leq \|w\|\sqrt{K(x,x)} \leq M \cdot B$, where $M := \sup_{x\in\mathcal{X}} \sqrt{K(x,x)}$. Note also that the hinge loss ϕ is a Lipschitz function with unit constant. We can now apply a result on the concentration of the supremum of empirical processes to bound $R(\cdot) - R_m(\cdot)$. Indeed, applying Thm. 8 of [22] to function class $\{\frac{1}{MB} f | I(f) \leq B\}$ (using their Thm. 12 to bound the Rademacher complexity of the kernel function class), we obtain that for any $\delta > 0$, with probability at least $1 - \delta$:

$$R(f_m) - R_m(f_m) \leq \frac{4MI(f_m)}{\sqrt{m}} + MI(f_m)\sqrt{\frac{8\ln(2/\delta)}{m}}.$$

We obtain with probability at least $1 - 2\delta$:

$$R(f_m) + \lambda_m I(f_m)^2/2 \leq R(f_0) + \frac{4M(I(f_m) + I(f_0))}{\sqrt{m}} + M(I(f_m) + I(f_0))\sqrt{\frac{8\ln(2/\delta)}{m}} +$$

$$\lambda_m I(f_0)^2/2 \leq \min_{f\in\mathcal{F}} R(f) + \epsilon + \frac{4M(I(f_m) + I(f_0))}{\sqrt{m}} + M(I(f_m) + I(f_0))\sqrt{\frac{8\ln(2/\delta)}{m}} +$$

$$\lambda_m (I(f_0)^2/2.$$

Combining Assumption A2 with the fact that $I(f_m) = O(1/\sqrt{\lambda_m})$, the RHS tends to $\min_{f\in\mathcal{F}} R(f) + \epsilon$ as $m \to \infty$. But $R(f_m) \geq \min_{f\in\mathcal{F}} R(f)$ by definition, so $R(f_m) - \min_{f\in\mathcal{F}} R(f) \to 0$ and $\lambda_m I(f_m)^2/2 \to 0$ in probability.

Thus we obtain that

$$C = \frac{1}{2}(R_m(f_m) + \lambda_m I(f_m)^2) \xrightarrow{p} \frac{1}{2} \min_{f\in\mathcal{F}} R(f) = \min_{f\in\mathcal{F}} P(Y \neq f(X)),$$

where the last equality is due to (5).

Before completing the proof, it is worth noting that to the rate of convergence also depends on the rate that $\lambda_m I(f_0)^2 \to 0$ as $\epsilon \to 0$. This requires additional knowledge of the approximating kernel class $\mathcal{H}$ driven by kernel function K, and additional properties of the optimal Bayes classifier that f_0 tends to.

Proof sketch of Proposition 1: If x_0 is a minimizer of a differentiable function $F : \mathrm{R}^d \to \mathrm{R}$ over a convex domain, then for any z in the domain, $(z - x_0)^T \nabla F(x_0) \geq 0$.

Applying this fact to both α and $\tilde{\alpha}$ which are the optimizers of Eqn. (2) using Q and $\tilde{Q}$, respectively:

$$(\tilde{\alpha}-\alpha)^T(\frac{1}{2m^2\lambda_m}Q\alpha-\frac{1}{m}\mathbb{I}_m)\geq 0$$
$$(\alpha-\tilde{\alpha})^T(\frac{1}{2m^2\lambda_m}\tilde{Q}\tilde{\alpha}-\frac{1}{m}\mathbb{I}_m)\geq 0,$$

where $\mathbb{I}_m=[1\ldots 1]^T$. Adding up the two inequalities yields

$$(\alpha-\tilde{\alpha})^T(\tilde{Q}\tilde{\alpha}-Q\alpha)\geq 0.$$

A minor rearrangement yields

$$(\tilde{\alpha}-\alpha)^T(Q-\tilde{Q})\tilde{\alpha}\geq(\tilde{\alpha}-\alpha)^TQ(\tilde{\alpha}-\alpha),$$

from which the proposition follows immediately.

Proof of Lemma 1: By Cauchy-Schwarz, $R_2\leq\|(\tilde{Q}-Q)\tilde{\alpha}\|$. The i-th element of the vector inside $\|.\|$ in the RHS is $a_i=y_i\sum_{j=1}^m e_{ij}y_j\tilde{\alpha}_j$. Note that $\tilde{K},y$ determines the value of $\tilde{\alpha}$. Thus, by Assumption A0, we have:

$$\mathrm{E}[a_i^2|\tilde{K},y]=\sum_{j=1}^m\mathrm{E}[e_{ij}^2|\tilde{K},y]\mathrm{E}[\tilde{\alpha}_j^2|\tilde{K},y]\leq\sigma^2\mathrm{E}[\|\tilde{\alpha}\|^2|\tilde{K},y].$$

Marginalizing over $(\tilde{K},y)$ gives $\mathrm{E}a_i^2\leq\sigma^2\mathrm{E}\|\tilde{\alpha}\|^2$. Thus, $\mathrm{E}R_2\leq(\mathrm{E}R_2^2)^{1/2}\leq\sigma\sqrt{m\mathrm{E}\|\tilde{\alpha}\|^2}$.