Kernel Matrix Evaluation

Canh Hao Nguyen and Tu Bao Ho
School of Knowledge Science
Japan Advanced Institute of Science and Technology
1-1 Asahidai, Nomi, Ishikawa, 923-1292, Japan
{canhhao,bao}@jaist.ac.jp

Abstract

We study the problem of evaluating the goodness of a kernel matrix for a classification task. As kernel matrix evaluation is usually used in other expensive procedures like feature and model selections, the goodness measure must be calculated efficiently. Most previous approaches are not efficient, except for Kernel Target Alignment (KTA) that can be calculated in $O(n^2)$ time complexity. Although KTA is widely used, we show that it has some serious drawbacks. We propose an efficient surrogate measure to evaluate the goodness of a kernel matrix based on the data distributions of classes in the feature space. The measure not only overcomes the limitations of KTA, but also possesses other properties like invariance, efficiency and error bound guarantee. Comparative experiments show that the measure is a good indication of the goodness of a kernel matrix.

1 Introduction

Kernel methods, such as Support Vector Machines, Gaussian Processes, etc., have delivered extremely high performance in a wide variety of supervised and non-supervised learning tasks [Schölkopf and Smola, 2002]. The key to success is that kernel methods can be modularized into two modules: the first is to map data into a (usually higher dimensional) feature space, which is a powerful framework for many different data types; and the second is to use a linear algorithm in the feature space, which is efficient and has theoretical guarantees [Shawe-Taylor and Cristianini, 2004]. While many linear algorithms can be kernelized, the major effort in kernel methods is put into designing good mappings defined by kernel functions ϕ:

$$\phi : X \to H. \tag{1}$$

X is the original data space and *H* is the feature space, which is a Reproducing Kernel Hilbert Space (RKHS). While *H* may be a many-dimensional space, the algorithm takes advantage of the kernel trick to operate solely on the kernel matrix induced from data:

$$K = \{\langle \phi(x_i), \phi(x_j) \rangle\}_{i=1..n, j=1..n}. \tag{2}$$

The goodness of kernel function can only be seen from the goodness of the kernel matrix K; therefore, measuring the goodness of K is of primary interest in various contexts.

There are many ways to measure the goodness of a kernel matrix (kernel function) for a classification task, differently reflecting the expected quality of the kernel function. Commonly used measures are regularized risk [Schölkopf and Smola, 2002], negative log-posterior [Fine and Scheinberg, 2002] and hyperkernels [Ong *et al.*, 2005]. These measures do not give a specific value, but only assert certain criteria in form of regularities in certain spaces, for example, RKHS or hyper-RKHS. All of these kernel measures require an optimization procedure for evaluation. Therefore, they are prohibitively expensive to be incorporated into other expensive procedures like model and feature selections. Other techniques like cross validation, leave-one-out estimators or radius-margin bound can also be considered as expected quality functionals. Like the previously mentioned works, they require the whole learning process.

To be used in feature and model selection processes, goodness measures of kernel matrices must be efficiently calculated. To the best of our knowledge, the only efficient kernel goodness measure is Kernel Target Alignment [Cristianini *et al.*, 2002]. Due to its simplicity and efficiency, KTA has been used in many methods for feature and model selections. Some notable methods are learning conic combinations of kernels [Lanckriet *et al.*, 2004], learning idealized kernels [Kwok and Tsang, 2003], feature selection [Neumann *et al.*, 2005] and kernel design using boosting [Crammer *et al.*, 2002].

In this work, we first analyze KTA to show that having a high KTA is only a *sufficient* condition to be a good kernel matrix, but not a *necessary* condition. It is possible for a kernel matrix to have a very good performance even though its KTA is still low. We then propose an efficient surrogate measure based on the data distributions in the feature space to relax the strict conditions imposed by KTA. The measure is invariant to linear operators in the feature space and retains several properties of KTA, such as efficiency and an error bound guarantee. The measure is closely related to some other works on worst-case error bounds and distance metric learning. We show experimentally that the new measure is more closely correlated to the goodness of a kernel matrix, and conclude finally with some future works.

2 Kernel Target Alignment

Kernel target alignment is used to measure how well a kernel matrix aligns to a target (or another kernel) [Cristianini *et al.*, 2002]. Alignment to the target is defined as the (normalized) Frobenius inner product between the kernel matrix and the covariance matrix of the target vector. It is interpreted as cosine distance between these two bi-dimensional vectors. We first introduce some notations.

Training example set $\{x_i\}_{i=1..n} \subset X$ with the corresponding target vector $y = \{y_i\}_{i=1..n}^T \subset \{-1, 1\}^n$. Suppose that $y_1 = .. = y_{n_+} = 1$ and $y_{n_++1} = .. = y_{n_++n_-} = -1$; n_+ examples belong to class 1, n_- examples belong to class -1, $n_+ + n_- = n$. Under a feature map ϕ, the kernel matrix is defined as:

$$K = \{k_{ij} = \langle \phi(x_i), \phi(x_j) \rangle\}_{i=1..n, j=1..n}. \quad (3)$$

Frobenius inner product is defined as:

$$\langle K, K^* \rangle_F = \sum_{i=1}^{n} \sum_{j=1}^{n} k_{ij} k_{ij}^*. \quad (4)$$

The target matrix is defined to be $y.y^T$. Kernel target alignment of *K* is defined as the normalized Frobenius inner product:

$$A(K, y) = \frac{\langle K, y.y^T \rangle_F}{\sqrt{\langle K, K \rangle_F \langle y.y^T, y.y^T \rangle_F}}. \quad (5)$$

The advantage, which makes KTA popular in many feature and model selection methods, is that it satisfies several properties. It is efficient as its computational time is $O(n^2)$. It is highly concentrated around its expected value, giving a high probability that a value is close to the true alignment value. It also gives some error bounds, meaning that when KTA is high, one can expect that the error rate using the kernel must be limited to a certain amount.

We claim that having a very high value *A(K,y)* is only a sufficient condition, but not a necessary condition, for *K* to be a good kernel matrix for a given task specified by the target y. As $0 \leqslant A(K, y) \leqslant 1$[1], when $A(K, y) = 1$, the two bi-dimensional vectors K and $y.y^T$ are linear. Up to a constant, the optimal alignment happens when $k_{ij} = y_i.y_j$. This is equivalent to two conditions:

1. All examples of the same class are mapped into the same vector in the feature space ($k_{ij} = 1$ for x_i, x_j in the same class).
2. The mapped vectors of different classes in the feature space are additive inverse ($k_{ij} = -1$ for x_i, x_j in different classes).

It is a sufficient condition that having a high *K*, one can expect good performance, as classes are collapsed into a point in the feature space. Violating any of the conditions would mean that KTA is penalized. However, it is not a necessary condition as we analyze below.

The first condition implies that the within-class variance penalizes KTA. However, there is a gap between the condition and the concept of margin in Support Vector Machines (SVMs). Varying data (in the feature space) along the separating hyperplane will not affect the margin. If data varies along the perpendicular direction of the separating hyperplane, the margin can be changed. However, KTA does not take into account variances in different directions.

The second condition is too strict, and not applicable in many cases, because it requires the mapped vectors of different classes to be additive inverses of each other. Ideally, if the kernel function maps all examples of a class to one vector in the feature space, the kernel matrix should be evaluated as optimal. This is the reason why having high KTA is only a sufficient condition, but not a necessary condition. We use some examples to show this limitation quantitatively.

Example 1 (best case): The kernel function ϕ maps all examples of class 1 into vector $\phi_+ \in H$ and all examples of class -1 into vector $\phi_- \in H$. Assume that $\phi_+.\phi_+ = \phi_-.\phi_- = 1$ and $\phi_+.\phi_- = \alpha$, $-1 \leqslant \alpha < 1$. For any α, the kernel matrix *K* should be evaluated as optimal, as it is induced from an optimal feature map. However, its KTA value is:

$$A(K, y) = \frac{n_+^2 + n_-^2 - 2n_+n_-\alpha}{\sqrt{n_+^2 + n_-^2 + 2n_+n_-\alpha^2} * n}. \quad (6)$$

KTA values of these kernel matrices change as α varies from 1 to -1, and any value in that range can be the KTA value of a kernel matrix of an optimal feature function. As $\lim_{\alpha \to 1} A(K, y) = \frac{(n_+ - n_-)^2}{n^2}$, KTA ranges from $\frac{(n_+ - n_-)^2}{n^2}$ to 1 in this case.

Example 2 (worst case): The kernel function ϕ maps half of the examples of each class into vector $\phi_1 \in H$ and the other half into vector $\phi_2 \in H$. Assume that $\phi_1.\phi_1 = \phi_2.\phi_2 = 1$ and $\phi_1.\phi_2 = \alpha$, $-1 \leqslant \alpha \leqslant 1$. For any α, the kernel matrix *K* should be evaluated very low, as it is induced from the worst kernel function, which fuses the two classes together and has no generalization ability. However, its KTA value is:

$$A(K, y) = \frac{(n_+ - n_-)^2 \frac{1+\alpha}{2}}{n^2 * \sqrt{(\frac{1+\alpha^2}{2})}}. \quad (7)$$

As shown above, $\lim_{\alpha \to 1} A(K, y) = \frac{(n_+ - n_-)^2}{n^2}$, which is the same limit as the best case. We can see that KTA of this case ranges from 0 to $\frac{(n_+ - n_-)^2}{n^2}$. There is a similar caution in [Meila, 2003]. As the best and the worst cases cover the whole range of KTA values, any other case would coincide with one of them. Therefore, KTA may mistake any case to be either the best or the worst.

Example 3 (popular case): Consider the case when a kernel function maps all examples into an orthant in the feature space, inducing a kernel matrix with non-negative entries, i.e. $k_{ij} \geqslant 0$. In this case:

$$A(K, y) \leqslant \frac{\sqrt{n_+^2 + n_-^2}}{n}. \quad (8)$$

Proof can be derived by using the fact that $k_{ij} \geqslant 0$ and the Cauchy-Schwarz inequality [Gradshteyn and Ryzhik, 2000]. Take a special case when $n_+ = n_-$, $A(K, y) \leqslant \frac{1}{\sqrt{2}}$. Recall that KTA of a kernel matrix ranges from 0 to 1, and the

[1] In [Cristianini *et al.*, 2002], it is $-1 \leqslant A(K, y)$. The inequality here comes from the fact that K is positive semidefinite.

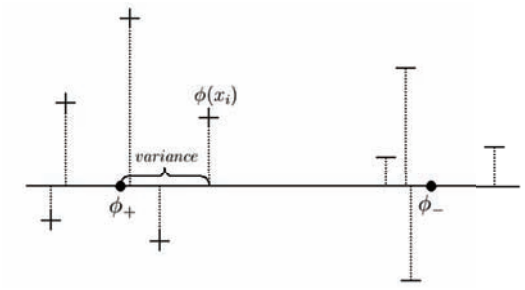

Figure 1: Data variance in the direction between class centers.

higher the better. This example means that these types of kernel functions will have bounded KTA values, no matter how good the kernel functions are. This is a drawback of KTA. Unfortunately, this type of kernel function is extensively used in practice. Gaussian kernels [Shawe-Taylor and Cristianini, 2004], and many other kernels defined over discrete structures [Gartner *et al.*, 2004] or distributions [Jebara *et al.*, 2004], etc. fall into this category.

The above examples show that KTA can mistake any kernel matrix to be the best or the worst. KTA is always bound to some numbers for many popular kernel matrices with positive entries (e.g., Gaussian, and many kernels for discrete structures or distributions). KTA will disadvantage the use of such kernels.

3 Feature Space-based Kernel Matrix Evaluation Measure

In this section, we introduce a new goodness measure of a kernel matrix for a given (binary classification) task, named Feature Space-based Kernel Matrix Evaluation Measure (FSM). This measure should be computed on the kernel matrix efficiently, and overcome the limitations of KTA. Our idea is to use the data distributions in the feature space. Specifically, two factors are taken into account: (1) *the within-class variance* in the direction between class centers; and (2) *relative positions of the class centers*. These factors are depicted in Figure 1. The first factor improves the first condition of KTA, by allowing data to vary in certain directions. The second factor solves the problem imposed by the second condition of KTA. Let var be the total within-class variance (for both classes) in the direction between class centers. Denote center of a class as the mean of the class data in the feature space: $\phi_+ = \frac{\sum_{i=1}^{n_+}\phi(x_i)}{n_+}$ and $\phi_- = \frac{\sum_{i=n_++1}^{n}\phi(x_i)}{n_-}$. Our evaluation measure FSM is defined to be *the ratio of the total within-class variance in the direction between the class centers to the distance between the class centers*:

$$FSM(K,y) := \frac{var}{\parallel \phi_- - \phi_+ \parallel}. \tag{9}$$

3.1 FSM Calculation

We show that the evaluation measure can be calculated using the kernel matrix efficiently with simple formulas. We first calculate the within-class variance (of both classes) in the direction between class centers. Denote $e = \frac{\phi_- - \phi_+}{\|\phi_- - \phi_+\|}$ as the unit vector in this direction. The total within-class variance of two classes in the direction is:

$$\begin{aligned} var = &[\frac{\sum_{i=1}^{n_+}\langle\phi(x_i)-\phi_+, e\rangle^2}{n_+-1}]^{\frac{1}{2}} \\ &+[\frac{\sum_{i=n_++1}^{n}\langle\phi(x_i)-\phi_-, e\rangle^2}{n_--1}]^{\frac{1}{2}}. \end{aligned} \tag{10}$$

We calculate the first term in equation (10), the total within-class variance of the class $+1$ in the direction between ϕ_+ and ϕ_-, denoted as var_+.

$$\begin{aligned} (n_+-1)var_+^2 &= \sum_{i=1}^{n_+}\langle\phi(x_i)-\phi_+, e\rangle^2 \\ &= \frac{\sum_{i=1}^{n_+}\langle\phi(x_i)-\phi_+, \phi_--\phi_+\rangle^2}{(\phi_--\phi_+)^2} \\ &= \frac{\sum_{i=1}^{n_+}(\phi(x_i)\phi_- + \phi_+^2 - \phi(x_i)\phi_+ - \phi_+\phi_-)^2}{(\phi_--\phi_+)^2}. \end{aligned} \tag{11}$$

We substitute $\phi_+ = \frac{\sum_{j=1}^{n_+}\phi(x_j)}{n_+}$ and $\phi_- = \frac{\sum_{j=n_++1}^{n}\phi(x_j)}{n_-}$ into equation (11), and then we define some auxiliary variables as follows. For $i = 1..n_+$:

- $a_i = \phi(x_i)\phi_+ = \frac{\sum_{j=1}^{n_+}\phi(x_i)\phi(x_j)}{n_+} = \frac{\sum_{j=1}^{n_+}k_{ij}}{n_+}$,
- $b_i = \phi(x_i)\phi_- = \frac{\sum_{j=n_++1}^{n}\phi(x_i)\phi(x_j)}{n_-} = \frac{\sum_{j=n_++1}^{n}k_{ij}}{n_-}$.

For $i = n_+ + 1..n$:

- $c_i = \phi(x_i)\phi_+ = \frac{\sum_{j=1}^{n_+}\phi(x_i)\phi(x_j)}{n_+} = \frac{\sum_{j=1}^{n_+}k_{ij}}{n_+}$,
- $d_i = \phi(x_i)\phi_- = \frac{\sum_{j=n_++1}^{n}\phi(x_i)\phi(x_j)}{n_-} = \frac{\sum_{j=n_++1}^{n}k_{ij}}{n_-}$.

Denote $A = \frac{\sum_{i=1}^{n_+}a_i}{n_+}$, $B = \frac{\sum_{i=1}^{n_+}b_i}{n_+}$, $C = \frac{\sum_{i=n_++1}^{n}c_i}{n_-}$ and $D = \frac{\sum_{n_++1}^{n}d_i}{n_-}$. Hence, $A = \phi_+\phi_+$, $B = C = \phi_+\phi_-$ and $D = \phi_-\phi_-$. Therefore, $(\phi_- - \phi_+)^2 = A + D - B - C$. Plugging them into equation (11), then:

$$(n_+-1)var_+^2 = \frac{\sum_{i=1}^{n_+}(b_i - a_i + A - B)^2}{A+D-B-C}. \tag{12}$$

We can use a similar calculation for the second term in equation (10):

$$(n_--1)var_-^2 = \frac{\sum_{i=n_++1}^{n}(c_i - d_i + D - C)^2}{A+D-B-C}. \tag{13}$$

The proposed kernel matrix evaluation measure FSM, as defined in formula (9), is calculated as:

$$FSM(K,y) = \frac{var_+ + var_-}{\sqrt{A+D-B-C}}. \tag{14}$$

$FSM(K,y) \geqslant 0$, and the smaller $FSM(K,y)$ is, the better K is.

One can easily see that a_i, b_i, c_i, d_i, A, B, C and D can be all calculated together in $O(n^2)$ time complexity (one pass through the kernel matrix). Therefore, the evaluation measure is also efficiently calculated in $O(n^2)$ time complexity.

3.2 Invariance

There are some properties of $FSM(K, y)$ regarding linear operations that make it closer to SVMs than KTA is.

Theorem 1: *$FSM(K, y)$ is invariant under translation, rotation and scale in the feature space.*

Sketch of proof: $FSM(K, y)$ is scale invariant owing to its normalization factor $A + D - B - C$. $FSM(K, y)$ is rotation and translation invariant, as it is built on the building blocks of $a_i - b_i$ and $d_i - c_i$.

The performance of SVMs and other kernel methods are unchanged under rotation, translation and scale in the feature space. Therefore, it is reasonable to ask for measures to have these properties. However, KTA is neither rotation invariant nor translation invariant.

3.3 Error Bound

We show that for any kernel matrix, it is possible to obtain a training error rate, which is bounded by some amount proportionate to $FSM(K, y)$. This means that a low $FSM(K, y)$ value can guarantee a low training error rate. In this case, we can expect a low generalization error rate.

Theorem 2: *There exists a separating hyperplane such that its training error is bounded by:*

$$FSMerr := \frac{FSM(K, y)^2}{1 + FSM(K, y)^2}. \tag{15}$$

Proof: We use a one-tailed version of Chebyshev's inequality [Feller, 1971]. Consider the data distribution in the direction between class centers. For each class, the data distribution has a mean of either ϕ_+ or ϕ_-, and the corresponding variance var_+ or var_-. The following inequalities can be derived:

$$\begin{aligned} Pr(\langle \phi - \phi_-, -e \rangle \geqslant k.var_- | \phi \in class(-)) &\leqslant \frac{1}{1 + k^2} \\ Pr(\langle \phi - \phi_+, e \rangle \geqslant k.var_+ | \phi \in class(+)) &\leqslant \frac{1}{1 + k^2}. \end{aligned} \tag{16}$$

The separating hyperplane, which takes e as its norm vector and intersects the line segment between ϕ_+ and ϕ_- at the point h such that $\frac{|h-\phi_+|}{|h-\phi_-|} = \frac{var_+}{var_-}$, has the formula:

$$f(x) = e.x - e.\frac{var_-\phi_+ + var_+\phi_-}{var_+ + var_-} = 0. \tag{17}$$

The error rate of this hyperplane for each class is bounded using the inequalities in equation (16) with $k = \frac{|\phi_- - \phi_+|}{var_+ + var_-} = \frac{1}{FSM(K,y)}$. Therefore, the total training error rate on both classes is also bounded by:

$$\frac{1}{1 + k^2} = \frac{FSM(K, y)^2}{1 + FSM(K, y)^2}. \tag{18}$$

3.4 Discussion

The measure can be interpreted as the ratio of within-class variance to between-class variance. It indicates how well the two classes are separated. It is advantageous over KTA because it takes into account the within-class variances at a finer scale, and relaxes the strict conditions of KTA by considering relative positions of classes in the feature space. For the examples in Section 2, FSM values of the best cases are always 0, those of the worst cases are always ∞, and for the popular case, there are no bounds. This measure also has some similarities with other works.

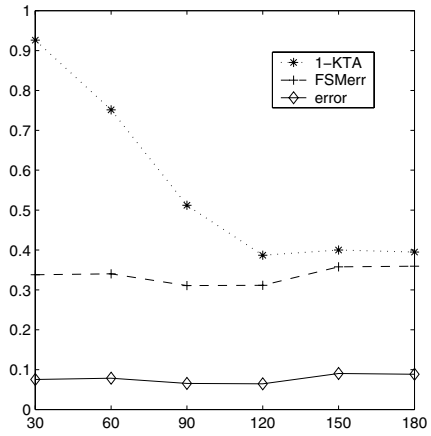

Figure 2: Results on synthetic data with different β values.

The minimax probability machine (MPM) [Lanckriet *et al.*, 2002] controls misclassification probability in the worst-case setting. The worst-case misclassification probability is minimized using Semidefinite Programming. The worst case is determined by making no assumption rather than the mean and the covariance matrix of each class. MPM is similar to our approach that our measure also shows the worst-case misclassification error. The difference is that MPM considers the data variance in all directions (by using the covariance matrix), while our measure takes only the variance in the direction between the class centers. That is the reason why our measure is a lightweight version of MPM and can be calculated efficiently. A special case of MPM is when the direction between class centers contains eigenvectors of both covariance matrices; the objective function that MPM optimizes is equivalent to our measure.

In the context of nearest neighbor classification, one of the criteria of learning a good distance function is to transform data such that examples from the same class are collapsed into one point [Globerson and Roweis, 2006]. Our measure also shows quantitatively how much a class collapses. However, the key difference is that in their method, the objective is the whole class collapses to one point, while our measure shows how the whole class collapses to a hyperplane. This difference explains why their method is applied to nearest neighbor classification, while our measure is for kernel methods.

4 Experiments

As the purpose of these measures is to predict efficiently how good a kernel matrix is for a given task using kernel methods, we compare the measures to the de facto standard of cross validation error rates of SVMs. We mimicked the model selection process by choosing different kernels. We monitored

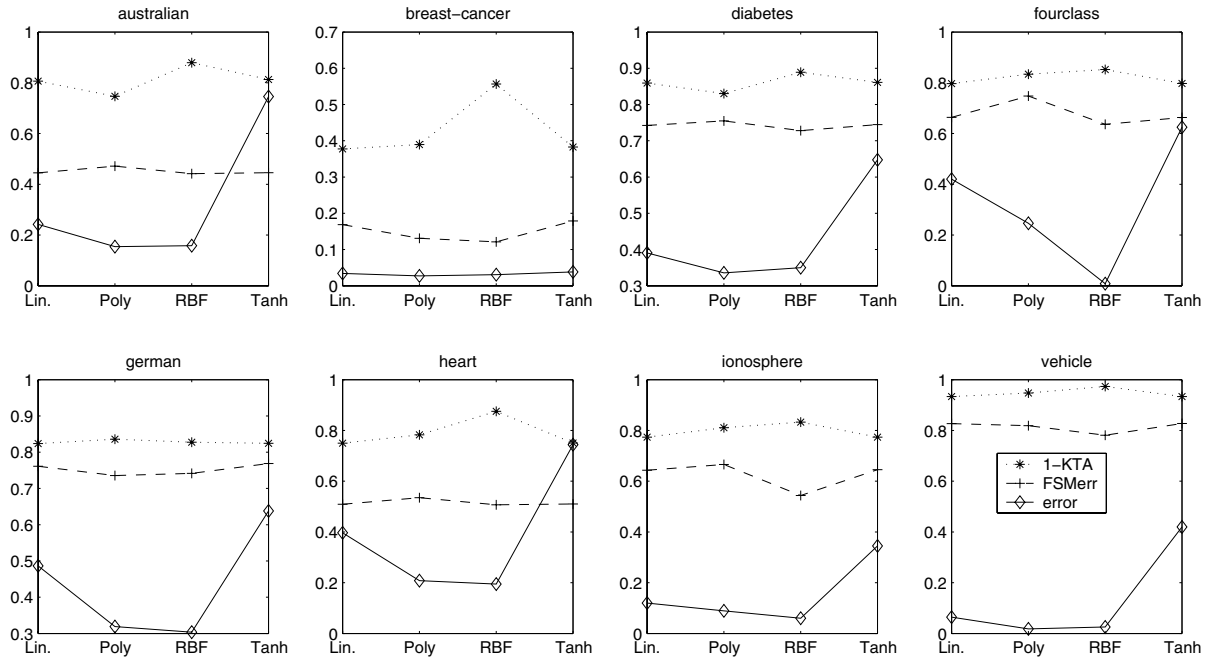

Figure 3: Results on different UCI data sets.

the error rates (as baselines), KTA and FSM, to see if how they reflect the baselines. We used 10-times 5-fold stratified cross validations to estimate the error rates.

To facilitate visual inspection of the results, we instead showed the following quantities: (a) $1-KTA$, (b) $FSMerr$, defined as error bound based on FSM measure in formula (15), and (c) *error*, cross validation error rates. The reason is that all of these quantities range from 0 to 1 and relate to the expected error rates of the kernel function in some sense. The smaller their values, the better the kernel. The first two quantities are expected to be correlated to the third one.

4.1 Synthetic Data

We generated synthetic data in R^2, and used linear kernels to simulate different data distributions by different kernels in a feature space. For each data set parameterized by an angle β, we used two Gaussian distributions (fixed variance in all directions) for two classes centered at $\phi_+ = (1,0) \in R^2$ for class $+1$ and at $\phi_- = (cos(\beta), sin(\beta)) \in R^2$ for class -1. Each class contains 500 training examples. The variances of the Gaussian distributions are determined to be $var_+ = var_- = \frac{1}{2} \parallel \phi_- - \phi_+ \parallel$. This ensures that for any β, any data set can be images of another after a linear operation. Therefore, these data sets with linear kernels are equivalent to one data set with different kernel functions. For any $\beta < 1$, the problems should have the same level of error rates (or other measures) when using linear kernels, i.e. linear kernels should be evaluated at the same level of goodness. We run experiments with different β values of 30, 60, 90, 120, 150 and 180 (degrees) in turn. The results of *1-KTA*, $FSMerr$ and $error$ are shown in Figure 2.

From Figure 2, we can observe that $error$ is stable across different β, as we expected. $FSMerr$ is also rather stable, varying similarly to $error$. *1-KTA* changes dramatically from 0.936 down to 0.395. It can be concluded that KTA is too sensitive to absolute positions of data in the feature space. This confirms our claim about the limitations of KTA, and shows that our measure is more robust.

4.2 Real Data

We showed the advantage of our proposed measure over KTA in real situations by selecting several popular data sets from the UCI collection for experiments. Data names and experimental results are displayed in Figure 3. Data was normalized to $[-1, 1]$. Classes were grouped to make binary classification problems. We chose four types of kernel for model selection: linear kernels, polynomial kernels (degree 3 with scale 1), RBF (Gaussian) kernels (default), and tan-hyperbolic kernels (default)[2]. As described above, we ran cross validations and displayed the results of different kernels.

The following conclusions can be observed from the graphs:

- RBF kernels give the (approximately) lowest cross validation error rates in 7 out of 8 data sets. This shows that RBF kernels are usually a good choice. However, KTA makes RBF kernels the worst one as values of *1-KTA* are the highest ones in 7 out of 8 cases. Our measure agrees with the error rates in most cases, making it more reliable than KTA is.
- Similarly, Tanh kernels perform worst in most cases, while KTA values are among the best ones.

[2]Default parameter values are set by LIBSVM environment at *http://www.csie.ntu.edu.tw/~cjlin/libsvm/*

- In case of very low error rates as in *breast-cancer* data set, FSMerr is highly correlated to error rates, but KTA is not.

We ranked the kernels according to *error*, KTA and FSM separately from 1 to 4. We collected the rank of the best kernel (according to $error$) for each dataset. On average, KTA ranks the best kernel at 2.63 ($\pm$1.061) while FSM ranks it at 1.63 ($\pm$0.518). A student t-test (with 95% level of confidence) shows that FSM ranks the best kernel closer to 1 than KTA. This also confirms the advantage of FSM.

In summary, we used synthetic and real data sets and compared measures against the *de facto* standard of cross validation error rates. We showed that our measure correlates more closely to the error rates than KTA does. When ranking the best kernel according to the error rates, FSM shows a significantly lower rank, on average. This suggests that our measure is more reliable. This confirms our analysis of the limitations of KTA and that our measure can overcome themse limitations. It was also observed that there is still a gap between kernel matrix evaluation measures and error rates, as error rates are collected from many training rounds while measures are calculated with one pass through the kernel matrices. This is the trade off necessary for efficiency.

5 Conclusion

The paper shows that KTA, an efficient kernel matrix measure, which is popular in many contexts, has fundamental limitations, as it is only a sufficient, not a necessary condition to evaluate the goodness of a kernel matrix. A new measure is proposed to overcome those limitations by using data distributions in the feature space. This new measure follows the conventional wisdom of measuring within-class and between-class variances in an appropriate way. The measure provides the same efficiency as KTA, as evaluated in $O(n^2)$ time complexity, and other properties. It also has links with other methods. This measure is more correlated to the error rates of SVMs than KTA is.

The implication of this work is vast. One can take into account finer data distribution models in the feature space, to improve the current work. Also, there are a large number of applications of this measure for other methods. We can directly apply this measure (similarly to KTA) to many feature and model selection problems, such as boosting kernel matrices, multiple kernel learning, feature selection and so on. In general, with efficient kernel evaluation, we can leverage the work of kernel matrix and kernel function learning, which is of central interest in kernel methods.

Acknowledgements

We would like to thank Japan MEXT for the first author's financial support, and anonymous reviewers for helpful comments.

References

[Crammer *et al.*, 2002] Koby Crammer, Joseph Keshet, and Yoram Singer. Kernel design using boosting. In *NIPS*, pages 537–544, 2002.

[Cristianini *et al.*, 2002] Nello Cristianini, John Shawe-Taylor, Andre Elisseeff, and Jaz Kandola. On kernel-target alignment. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, Cambridge, MA, 2002. MIT Press.

[Feller, 1971] William Feller. *An Introduction to Probability Theory and Its Applications*, volume 2. John Wiley & Sons, 1971.

[Fine and Scheinberg, 2002] Shai Fine and Katya Scheinberg. Efficient svm training using low-rank kernel representations. *Journal of Machine Learning Research*, 2:243–264, 2002.

[Gartner *et al.*, 2004] Thomas Gartner, John W. Lloyd, and Peter A. Flach. Kernels and distances for structured data. *Machine Learning Journal*, 57(3):205–232, 2004.

[Globerson and Roweis, 2006] Amir Globerson and Sam Roweis. Metric learning by collapsing classes. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 451–458, 2006.

[Gradshteyn and Ryzhik, 2000] I. S. Gradshteyn and I. M. Ryzhik. *Table of Integrals, Series, and Products*. Academic Press, San Diego, CA, 2000.

[Jebara *et al.*, 2004] Tony Jebara, Risi Kondor, and Andrew Howard. Probability product kernels. *Journal of Machine Learning Research*, 5:819–844, 2004.

[Kwok and Tsang, 2003] James T. Kwok and Ivor W. Tsang. Learning with idealized kernels. In *ICML*, pages 400–407, 2003.

[Lanckriet *et al.*, 2002] Gert R. G. Lanckriet, Laurent El Ghaoui, Chiranjib Bhattacharyya, and Michael I. Jordan. A robust minimax approach to classification. *Journal of Machine Learning Research*, 3:555–582, 2002.

[Lanckriet *et al.*, 2004] Gert R. G. Lanckriet, Nello Cristianini, Peter Bartlett, Laurent El Ghaoui, and Michael I. Jordan. Learning the kernel matrix with semidefinite programming. *J. Mach. Learn. Res.*, 5:27–72, 2004.

[Meila, 2003] Marina Meila. Data centering in feature space. In Christopher M. Bishop and Brendan J. Frey, editors, *Ninth International Workshop on Artificial Intelligence and Statistics*, 2003.

[Neumann *et al.*, 2005] Julia Neumann, Christoph Schnorr, and Gabriele Steidl. Combined svm-based feature selection and classification. *Machine Learning*, 61(1-3):129–150, 2005.

[Ong *et al.*, 2005] Cheng Soon Ong, Alexander J. Smola, and Robert C. Williamson. Learning the kernel with hyperkernels. *Journal of Machine Learning Research*, 6:1043–1071, 2005.

[Schölkopf and Smola, 2002] Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

[Shawe-Taylor and Cristianini, 2004] John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, New York, NY, USA, 2004.