# Kernel Selection in Support Vector Machines Using Gram-Matrix Properties

**Roberto Valerio**
Department of Computer Science
University Houston
Houston, TX 77004
rvalerio@cs.uh.edu

**Ricardo Vilalta**
Department of Computer Science
University Houston
Houston, TX 77004
vilalta@cs.uh.edu

## Abstract

We describe an approach to kernel selection in Support Vector Machines (SVMs) driven by the Gram matrix. Our study extracts properties from this matrix (e.g., Fisher's discriminant, Bregman's divergence) using different kernel functions (linear, polynomial, Gaussian, Laplacian, Bessel and ANOVARBF), and incorporates such properties as meta-features within a meta-learning framework. The goal is to predict the best kernel in SVMs. Results show how introducing a new meta-feature, Distance Ratio, capturing inter-class and intra-class distances in the feature space, yields substantial improvements during kernel selection.

## 1 Introduction

Kernel methods have gained increased popularity in the machine learning community in recent years; one key strength underlying these methods is the ability to map the original training points into a higher dimensional feature space, thereby facilitating the job of a low capacity (linear) learning machine. Mapping points into high dimensional spaces has proved an effective strategy for classification in learning algorithms like Support Vector Machines (SVMs) [6], mainly because the mapping does not imply an increase in the complexity of the classifier, and because it is possible to rely on an artifice known as the *kernel trick*, that obviates a precise definition of the mapping functions.

Despite the popularity of kernel methods, there is not yet a mechanism in place that can serve to guide the selection of the kernel function. The goal of this study is to analyze the behavior of different kernels in SVMs, and determine the kernel that seems best suited for a specific task. To achieve this goal, we analyze and characterize data sets using Gram-Matrix properties [7, 5]. Our approach is guided by the performance of SVMs on several real world dataset, by metrics capturing class integration and separability (Fisher's discriminant and Bregman's divergence) and class distribution (Homoscedasticity), and by introducing a new estimate of the inter-class and intra-class distances (Distance Ratio) in the *feature space* induced by the kernel function.

We validate the results of our methodology on different real world data sets by assessing the performance of SVMs when invoked with our predicted kernel. We compare the generalization accuracy of each metric alone, and the performance of a meta-learner [1] that aims to predict the best kernel using the aforementioned properties or meta-features.

## 2 Preliminaries

We begin by describing our notation. A classifier receives as input a set of training examples, $T = \{(\mathbf{x_i}, y_i)\}_{i=1}^{n}$, where $\mathbf{x} = (x_1, x_2, \cdots, x_n)$ is a vector in the input space $\mathcal{X}$, and $y_i$ is a value in the (discrete) output space $\mathcal{Y}$. We assume the training sample $T$ consists of independently and identically distributed (i.i.d.) examples obtained according to a fixed but unknown joint probability distribution, $P(\mathbf{x}, y)$, in the input-output space $\mathcal{X} \times \mathcal{Y}$. The outcome of the classifier is a hypothesis or function $f(\mathbf{x}|\theta)$ (parameterized by $\theta$) mapping the input space to the output space, $f : \mathcal{X} \to \mathcal{Y}$.

Support Vector Machines work by mapping the original attribute space $\mathcal{X}$ into a high dimensional space $\Phi$ (i.e., a feature space), and by searching for a hyperplane that maximizes the margin[6]. The final model $f(\mathbf{x})$ is a weighted sum:

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

where $b$ is the bias component, $\{\alpha_i\}$ are weights ($\alpha_i > 0$ indicates $\mathbf{x}_i$ is a support vector), and we assume $K(\mathbf{x}, \mathbf{z}) = \langle \phi(\mathbf{x}) \cdot \phi(\mathbf{z}) \rangle$; the kernel implicitly computes the dot product of two feature vectors on a high dimensional space $\phi(\mathbf{x})$, an operation known as the *kernel trick*.

The Gram matrix $K$ stores the values of the kernel function $K(\mathbf{x}, \mathbf{z})$; $K$ is a matrix of size $n \times n$, where $n$ represents the number of examples in dataset $T$. In order for $K(\mathbf{x}, \mathbf{z})$ to be considered a kernel function, it must meet Mercer's conditions: the matrix must be symmetric positive semi-definite with no negative eigenvalues.

### 2.1 Gram Matrix Properties or Meta-features

Three main approaches have been explored to extract information from the Gram matrix to study the effect of the kernel function in Support Vector Machines [7][5]: Fisher's Discriminant $F$, Bregman's Divergence $B$, and Homoscedasticity $Q$:

$$F = \frac{tr(\Sigma_b^\Phi)}{tr(\Sigma_w^\Phi)} \tag{2}$$

$$B(\Sigma_1^\Phi, \Sigma_2^\Phi) = tr(\Sigma_1^\Phi) + tr(\Sigma_2^\Phi) - 2tr(\Sigma_1^\Phi \Sigma_2^\Phi) \tag{3}$$

$$Q = Q_1 Q_2 \qquad Q_1 = \frac{tr(\Sigma_1^\Phi \Sigma_2^\Phi)}{tr(\Sigma_1^\Phi) + tr(\Sigma_2^\Phi)} \qquad Q_2 = tr(\Sigma_B^\Phi) \tag{4}$$

where the superscript $\Phi$ indicates we are operating in the high dimensional space $\Phi$ (as opposed to the input space $\mathcal{X}$). $tr(\Sigma^\Phi)$ is the trace of covariance matrix $\Sigma^\Phi$. $\Sigma_1^\Phi$ and $\Sigma_2^\Phi$ are the covariance matrices of class 1 and class 2 respectively. Since the Gram matrix meets Mercer's conditions, we can use it directly on the three metrics above (i.e, we can substitute $\Sigma^\Phi$ for $K$).

Specifically, Fisher's discriminant measures the ratio of the separation between classes and the cohesiveness of elements in each class. This is computed as the ratio of the inter-class variance $\Sigma_b^\Phi$ and the intra-class variance $\Sigma_w^\Phi$. The best kernel is the one that maximizes $F$ (equation 2).

Bregman's divergence is used to measure the similarity between two convex sets and is widely used in optimization. Following the approach suggested by [7], the optimal kernel is found by minimizing $B$ (equation 3)

In Homoscedasticity (equal distribution of clusters; equation 4), the desired kernel is the one that produces clusters with equal distribution in $\Phi$ (i.e., in the feature space); this is equivalent to maximizing $Q_1$. However $Q_1$ by itself does not provides enough information, since clusters can overlap while being homoscedastic. $Q_2$ measures class separability and needs to be maximized. $Q$ is a metric that combines homoscedasticity and class separability; by maximizing $Q_1$ and $Q_2$ we also maximize $Q$ [5].

## 3 Distance Ratio

We introduce a new distance metric as a relevant property of the Gram matrix. The rationale to our approach is to identify the degree of class cohesiveness and class separability directly in space $\Phi$ based on geometric distances, and not covariance calculations, a property that remains unaccounted for in previous work. This enables us to capture the degree of locality (or lack of) embedded in each kernel function. To begin we generate a distance matrix $D$ over feature space $\Phi$. This is readily available from the Gram matrix using equation 5.

$$D(\mathbf{x}_i, \mathbf{x}_j) = D_{ij} = K_{ii} + K_{jj} - 2K_{ij} \tag{5}$$

where $K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$. One can then compute the intra-class (equation 6) and inter-class (equation 8) distances directly in feature space $\Phi$:

$$\begin{aligned} D_{intra} &= \Sigma_{i=1}^{n} \Sigma_{j=1}^{n} h(i,j) & (6) \\ h(i,j) &= D_{ij}, \textit{if } y_i = y_j\textit{; 0, otherwise.} & (7) \end{aligned}$$

$$\begin{aligned} D_{inter} &= \Sigma_{i=1}^{n} \Sigma_{j=1}^{n} g(i,j) & (8) \\ g(i,j) &= D_{ij}, \textit{if } y_i \neq y_j\textit{; 0, otherwise.} & (9) \end{aligned}$$

where $g(i,j)$ and $h(i,j)$ retrieve the distance between examples $\mathbf{x}_i$ and $\mathbf{x}_j$ across classes and within classes respectively. We define Distance Ratio 10 as the ratio of class cohesiveness (intra-class distance) and class separability (inter-class distance). We search for a kernel that minimizes *Distance Ratio* as a form of kernel selection.

$$\textit{Distance Ratio} = \frac{D_{intra}}{D_{inter}} \tag{10}$$

## 4 A Meta-Learning Approach to Kernel Selection in SVM

In this section we use the metrics introduced in Section 2 for kernel prediction. For each dataset, we first use a single metric to predict the kernel using SVMs. Performance is assessed by computing a form of accuracy loss, defined as the difference between the accuracy obtained by an SVM model built with the (known) best kernel, and the accuracy obtained by an SVM model built with the kernel selected by the current metric. The best kernel is the the one that minimizes such loss.

We report results using real-world data sets from UCI repository [4] characterized by two classes, numeric attributes, and no missing data. We use package R to train SVM models using the following kernel functions: RBF, Linear, Polynomial, Laplacian, Bessel and ANOVA RBF. Table 1 compares the performance of our kernel predictors ($F$, $B$, $Q_1$, $Q_2$, $Q$)) with *Distance Ratio*, by displaying accuracy loss as described above. The last row shows mean scores over all datasets; *Distance Ratio* shows the lowest accuracy loss.

We also report on a meta-learning approach using a neural network as the meta-model. Each dataset is represented by the prediction of Gram matrix properties and the class label is the kernel function with best performance. Table 2 compares three different meta-learners where the aim is to minimize *Accuracy Loss* and *Error Rate* (0 if predicted correctly, 1 if predicted incorrectly). For both *Accuracy Loss* and *Error Rate* we compare performance using all (traditional) properties or kernel predictors, using top properties ($F$, $B$, $Q_2$, and $Q$ are selected after performing feature selection using Relieff), and combining top properties with *Distance Ratio*. Best mean scores are attained when *Distance Ratio* is added to the set of top properties.

We conclude that the proposed new metric *Distance Ratio* is an effective way to characterize the Gram matrix; it is an accurate kernel predictor on its own, and improves the performance of a meta-learner when combined with traditional kernel properties.

Table 1: *Accuracy Loss* obtained by kernel selection using Gram-Matrix properties.

| Dataset | $F$ | $B$ | $Q_1$ | $Q_2$ | $Q$ | *Distance Ratio* |
|---|---|---|---|---|---|---|
| Arcene | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| Bank Marketing | 1.71 | 0.81 | 1.71 | 1.71 | 1.71 | 0.81 |
| Census Income | 0.84 | 0.85 | 0.84 | 7.85 | 7.85 | 32.17 |
| Voting Records | 0.45 | 1.41 | 0.45 | 0.45 | 0.45 | 0.45 |
| Credit Approval | 0.13 | 0.11 | 0.13 | 0.13 | 0.13 | 0.11 |
| Bands | 4.08 | 4.12 | 4.08 | 4.08 | 4.08 | 4.08 |
| Echocardiogram | 0.00 | 2.40 | 0.00 | 0.00 | 0.00 | 1.50 |
| Fertility | 2.55 | 0.55 | 2.55 | 2.55 | 2.55 | 0.55 |
| Haberman | 1.40 | 1.43 | 1.40 | 1.40 | 1.40 | 1.40 |
| Hepatitis | 2.00 | 2.00 | 2.00 | 0.42 | 0.42 | 0.42 |
| Hill Valley | 0.00 | 0.00 | 0.00 | 6.54 | 0.00 | 9.54 |
| Ionosphere | 54.24 | 7.80 | 54.24 | 54.24 | 54.24 | 7.80 |
| Magic | 0.00 | 0.00 | 0.00 | 0.20 | 0.00 | 0.00 |
| Musk ver 1 | 6.37 | 6.41 | 6.37 | 6.37 | 6.37 | 0.00 |
| Musk ver 2 | 4.08 | 4.08 | 4.08 | 4.08 | 4.08 | 0.00 |
| Diabetes | 0.32 | 0.32 | 0.32 | 12.13 | 12.13 | 0.32 |
| Sonar | 0.00 | 9.35 | 0.00 | 0.00 | 0.00 | 0.00 |
| Spambase | 32.70 | 0.42 | 32.70 | 32.70 | 32.70 | 0.42 |
| Blood | 3.16 | 3.16 | 3.16 | 3.05 | 3.16 | 3.16 |
| Breast Cancer | 0.21 | 0.21 | 0.21 | 1.48 | 1.48 | 0.21 |
| Average | 5.71 | 2.17 | 5.71 | 6.96 | 6.63 | 3.14 |

Table 2: *Accuracy Loss* and *Error Rate* by kernel selection using meta-learning.

| | *Accuracy Loss* | | | *Error Rate* | | |
|---|---|---|---|---|---|---|
| Dataset | All Properties | Top Properties | Top Properties + *Distance Ratio* | All Properties | Top Properties | Top Properties + *Distance Ratio* |
| Arcene | 0.00 | 0.00 | 0.00 | 0 | 0 | 0 |
| Bank Marketing | 0.00 | 0.00 | 0.00 | 0 | 0 | 0 |
| Census Income | 0.00 | 0.00 | 0.00 | 0 | 0 | 0 |
| Voting Records | 0.00 | 0.00 | 0.45 | 0 | 0 | 1 |
| Credit Approval | 0.00 | 0.00 | 0.00 | 0 | 0 | 0 |
| Bands | 0.00 | 0.00 | 0.00 | 0 | 0 | 0 |
| Echocardiogram | 8.40 | 7.50 | 7.50 | 1 | 1 | 1 |
| Fertility | 0.00 | 0.00 | 0.00 | 0 | 0 | 0 |
| Haberman | 0.00 | 0.00 | 0.00 | 0 | 0 | 0 |
| Hepatitis | 0.00 | 0.00 | 0.00 | 0 | 0 | 0 |
| Hill Valley | 0.00 | 0.00 | 0.00 | 0 | 0 | 0 |
| Ionosphere | 0.00 | 0.00 | 0.00 | 0 | 0 | 0 |
| Magic | 0.00 | 0.00 | 0.00 | 0 | 0 | 0 |
| Musk ver 1 | 0.00 | 0.00 | 0.00 | 0 | 0 | 0 |
| Musk ver 2 | 0.00 | 0.00 | 0.00 | 0 | 0 | 0 |
| Diabetes | 1.56 | 1.56 | 0.00 | 1 | 1 | 0 |
| Sonar | 1.75 | 1.75 | 0.00 | 1 | 1 | 0 |
| Spambase | 0.00 | 0.00 | 0.00 | 0 | 0 | 0 |
| Blood | 3.16 | 3.16 | 3.16 | 1 | 1 | 1 |
| Breast Cancer | 0.00 | 0.00 | 0.00 | 0 | 0 | 0 |
| Average | 0.7435 | 0.6985 | 0.5555 | 0.20 | 0.20 | 0.15 |

## References

[1] Brazdil, P., Giraud-Carrier, C., Soares, C. and Vilalta, R.: Metalearning: applications to data mining. Springer Verlag. ISBN: 978-3-540-73262-4. (2008).

[2] Bregman, L.: The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. USSR Computational Math and Math. Physics 3, vol. 7, pp. 200-217, (1967).

[3] Fisher, R.A.: The use of multiple measurements in taxonomic problems. Annals of Eugenics 2, vol. 7, pp. 179-188, (1936).

[4] Bache K., Lichman, M.: UCI Machine Learning Repository. (2010).

[5] Wang, L., Chan, K.L., Xue, P., Zhou, L.: A kernel-induced space selection approach to model selection in KLDA. IEEE Trans. Neural Networks 12, vol. 19, pp. 2116-2131, (2008).

[6] Cortes C., Vapnik, V.: Support-Vector networks. Machine Learning 3, vol. 20, pp. 273-297, (1995).

[7] You, D., Hamsici, O.C., Martinez, A.M.: Kernel optimization in discriminant analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence 3, vol. 33, pp. 631-638, (2011).