To appear in: Proceedings of the Seventeenth International Conference on Machine Learning. Morgan Kaufmann (2000).

The Effect of the Input Density Distribution on Kernel-based Classifiers

Christopher K. I. Williams C.K.I.WILLIAMS@ED.AC.UK
Matthias Seeger SEEGER@DAI.ED.AC.UK
Institute for Adaptive and Neural Computation, Division of Informatics, University of Edinburgh 5 Forrest Hill, Edinburgh EH1 2QL, Scotland, UK

Abstract

The eigenfunction expansion of a kernel function $K(\mathbf{x}, \mathbf{y})$ as used in support vector machines or Gaussian process predictors is studied when the input data is drawn from a distribution $p(\mathbf{x})$. In this case it is shown that the eigenfunctions $\{\phi_i\}$ obey the equation $\int K(\mathbf{x}, \mathbf{y}) p(\mathbf{x}) \phi_i(\mathbf{x}) d\mathbf{x} = \lambda_i \phi_i(\mathbf{y})$. This has a number of consequences including (i) the eigenvalues/vectors of the $n \times n$ Gram matrix K obtained by evaluating the kernel at all pairs of training points $K(\mathbf{x}_i, \mathbf{x}_j)$ converge to the eigenvalues and eigenfunctions of the integral equation above as $n \to \infty$ and (ii) the dependence of the eigenfunctions on $p(\mathbf{x})$ may be useful for the class-discrimination task. We show that on a number of datasets using the RBF kernel the eigenvalue spectrum of the Gram matrix decays rapidly, and discuss how this property might be used to speed up kernel-based predictors.

1. Introduction

In recent years kernel-based classifiers such as support vector machines (SVMs) (Vapnik, 1995), Gaussian process classifiers (e.g. see Williams & Barber, 1998) and spline methods (Wahba, 1990) have become popular. This paper studies kernel methods from the perspective of an eigenfunction expansion which is dependent on the input data density $p(\mathbf{x})$ as well as the kernel function $K(\mathbf{x}, \mathbf{y})$. In Section 1 we explain the relevant eigenfunction theory and demonstrate the importance of large-eigenvalue functions. In Section 2.1 the effect of a multimodal input density is studied, leading to general ideas in Section 2.2 about how the density effectively helps overcome the curse of dimensionality by creating basis functions tuned to the density. A further observation concerns the rate of decay of the eigenspectrum; this is demonstrated on some datasets in Section 2.3. Section 3 then discusses how this eigenstructure might be used to speed up computation.

2. Eigenfunction Expansions

In the theory of kernel machines we consider covariance kernels $K(\mathbf{x}, \mathbf{y})$. These can be related to an expansion into a feature space (typically of higher dimension than the input space $\mathbf{x}$ which has dimension N) so that

$$K(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{N_F} \lambda_i \phi_i(\mathbf{x}) \phi_i(\mathbf{y}), \qquad (1)$$

where $N_F \leq \infty$. If $N_F < \infty$ then we can construct a kernel using arbitrary suitably-smooth functions as the ϕs. However, if we want to interpret a covariance function such as the RBF kernel with $N_F = \infty$ then the functions ϕ_i are the eigenfunctions of the kernel via Mercer's theorem (e.g., see Vapnik, 1995).

The common textbook definition of the eigenfunctions relates them to a compact subset $\mathcal{C}$ of $\mathbb{R}^N$, so that

$$\int_{\mathcal{C}} K(\mathbf{x}, \mathbf{y}) \phi_i(\mathbf{x}) \, d\mathbf{x} = \lambda_i \phi_i(\mathbf{y}). \qquad (2)$$

If we wish to use a lower-dimensional linear model to approximate the solution of a kernel method (so that we use an approximator of the form $g(\mathbf{x}) = \sum_{i=1}^{M} w_i \psi_i(\mathbf{x})$) it is well known that the optimal solution is to use the eigenfunctions of the kernel corresponding to the M largest eigenfunctions as the basis functions $\psi_i(\mathbf{x})$. This choice minimizes the error $\int_{\mathcal{C}} (f(\mathbf{x}) - g(\mathbf{x}))^2 \, d\mathbf{x}$ between the true function $f(\mathbf{x})$ and the finite dimensional prediction $g(\mathbf{x})$, when averaged over functions drawn from the prior which corresponds to the covariance kernel K. This construction is an infinite-dimensional analogue of PCA.

Often in the kernel-methods literature not much attention has been paid to exactly which region $\mathcal{C}$ this integral is defined over, as explicit calculation of the eigenfunctions is unnecessary in order to use kernel methods. Indeed, this is what makes the algorithms tractable when dealing with high- or infinite-dimensional feature spaces. However, in most machine learning problems it is more likely that there is a probability density in input space $p(\mathbf{x})$ which is smoothly varying, rather than being constant within $\mathcal{C}$ an zero outside. In this case we can generalize the eigenproblem to include the weight function $p(\mathbf{x})$, to obtain

$$\int K(\mathbf{x},\mathbf{y})p(\mathbf{x})\phi_i(\mathbf{x})\; d\mathbf{x} = \lambda_i \phi_i(\mathbf{y}). \qquad (3)$$

Clearly equation 2 is a special case of equation 3 when $p(\mathbf{x})$ is uniform in $\mathcal{C}$ and zero outside.[1] It can be shown (see Appendix A) that these eigenfunctions minimize the average of the error $\int (f(\mathbf{x}) - g(\mathbf{x}))^2 p(\mathbf{x})\; d\mathbf{x}$ between a finite-dimensional model $g(\mathbf{x})$ (as defined above) and the true function f.

We note the following properties of this eigenproblem:

1. The eigenfunctions are orthonormal with respect to $p(\mathbf{x})$, i.e. $\int \phi_i(\mathbf{x})p(\mathbf{x})\phi_j(\mathbf{x}) = \delta_{ij}$.
2. The eigenvalues are the same for equation 3 and for the unweighted problem of the symmetric kernel $\tilde{K}(\mathbf{x},\mathbf{y})$ defined by

$$\tilde{K}(\mathbf{x},\mathbf{y}) = p^{1/2}(\mathbf{x})K(\mathbf{x},\mathbf{y})p^{1/2}(\mathbf{y}). \qquad (4)$$

   See Appendix A for the proof.

Equation 3 sheds light on the eigenvalue spectrum of the matrix K whose elements are the covariance function evaluated at the data points $\mathbf{x}_i$, $i = 1, \ldots, n$. First we observe that

$$\int K(\mathbf{x},\mathbf{y})p(\mathbf{x})\phi_i(\mathbf{x})d\mathbf{x} \simeq \frac{1}{n}\sum_{k=1}^{n} K(\mathbf{x}_k,\mathbf{y})\phi_i(\mathbf{x}_k) \qquad (5)$$

when the $\mathbf{x}_k$'s are sampled from $p(\mathbf{x})$. The standard numerical method (see, e.g., Baker, 1977, chapter 3) for approximating the eigenfunctions and eigenvalues of equation 3 is to use a numerical routine such as equation 5 to approximate the integral, and then plug in $\mathbf{y} = \mathbf{x}_k$ for $k = 1, \ldots, n$ to obtain a matrix eigenproblem. Thus we see that if $\lambda_1^{mat}, \lambda_2^{mat}, \ldots, \lambda_n^{mat}$ is the eigenvalue spectrum of the matrix K, then $1/n$ times this spectrum is an obvious estimator for the n largest eigenvalues of the continuous problem. One would expect that the larger eigenvalues would be better estimated than the smaller ones, and that the $\frac{1}{n}\lambda_n^{mat}$ will be a very poor estimator of λ_n. See the numerical example in section 2.1 for an example of this. The theory of the numerical solution of eigenvalue problems (Baker, 1977, Theorem 3.4) shows that for $k = 1,\ 2, \ldots,\ \frac{1}{n}\lambda_k^{mat}$ will converge to λ_k in the limit that $n \to \infty$.

For a stationary kernel (so that $K(\mathbf{x},\mathbf{y}) = K(\mathbf{x}-\mathbf{y})$) more can be said about the relationship of the eigenvalues of the matrix and eigenfunction problems. Let $K(\mathbf{0}) = C$, then it follows that $\int K(\mathbf{x},\mathbf{x})p(\mathbf{x})d\mathbf{x} = C$. From the eigenfunction representation of the kernel and the orthonormality of the eigenfunctions wrt $p(\mathbf{x})$ we get $\int K(\mathbf{x},\mathbf{x})p(\mathbf{x})d\mathbf{x} = \sum_{i=1}^{\infty}\lambda_i$. For the matrix problem we have $\mathrm{tr}(K) = nC = \sum_{i=1}^{n}\lambda_i^{mat}$. Hence we see that $\frac{1}{n}\sum_{i=1}^{n}\lambda_i^{mat} = \sum_{i=1}^{\infty}\lambda_i$.

The Nyström approximation to the ith eigenfunction (see Baker, 1977) is given by

$$\phi_i(\mathbf{y}) = \frac{1}{n\lambda_i}\sum_{k=1}^{n} K(\mathbf{x}_k,\mathbf{y})\phi_i(\mathbf{x}_k) \qquad (6)$$

where the $\phi_i(\mathbf{x}_k)$'s are known from the solution of the matrix problem.

There is an interesting relationship between the kernel PCA method of Schölkopf et al. (1998) and the eigenfunction expansion discussed above. The eigenfunction expansion has (at least potentially) an infinite number of non-zero eigenvalues. In contrast, the kernel PCA algorithm operates on the $n \times n$ matrix K and yields n eigenvalues and eigenvectors. Equation 5 clarifies the relationship between the two. However, note that equation 6 is identical (up to scaling factors) to equation 4.1 in Schölkopf et al. (1998) which describes the projection of a new point $\mathbf{x}$ onto the ith eigenvector in feature space.

2.1 An Analytic Example

For the case that $p(x)$ is a Gaussian and $K(x,y) = \exp -b(x-y)^2$, the RBF kernel with lengthscale $b^{-1/2}$, there are analytic results for the eigenvalues and eigenfunctions, as given in section 4 of Zhu et al. (1998). Putting $p(x) = \sqrt{\frac{2a}{\pi}}\exp -2ax^2$ we find that the eigenvalues λ_k and eigenfunctions ϕ_k (for convenience let $k = 0, 1, \ldots$) are given by

$$\lambda_k = \sqrt{\frac{2a}{A}}B^k \qquad (7)$$

$$\phi_k(x) = \exp(-(c-a)x^2)H_k(\sqrt{2c}x), \qquad (8)$$

[1] In this case the eigenvalues of equations 2 and 3 are related by volume($\mathcal{C}$).

where H_k is the kth order Hermite polynomial, and

$$c = \sqrt{a^2 + 2ab}, \quad A = a + b + c, \quad B = b/A. \qquad (9)$$

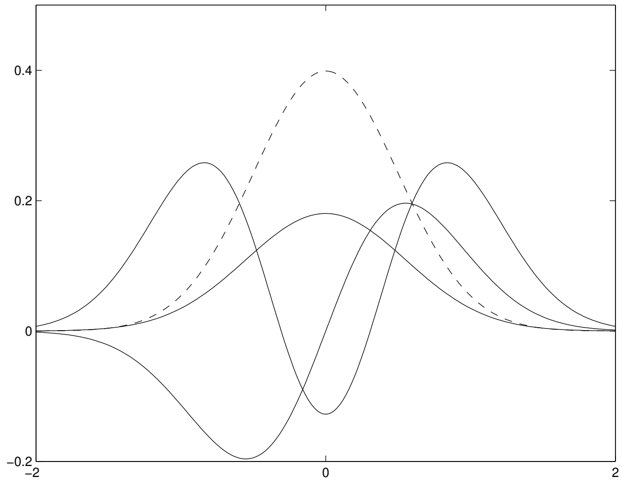

Figure 1. The first 3 eigenfunctions of the RBF kernel wrt a Gaussian density. The value of k is equal to the number of zero-crossings of the function. The dashed line is proportional to the density $p(x)$.

A plot of the first three eigenfunctions for $a = 1$ and $b = 3$ is shown in Figure 1. Note that the geometric decay of the eigenvalues in equation 7 is due to the "smooth" nature of the RBF kernel, which gives rise to a stochastic process which is infinitely mean-square differentiable. For processes which are r-times mean-square differentiable Ritter et al. (1995) showed that asymptotically $\lambda_k \propto k^{-(2r+2)}$.

Keeping a fixed, we can study the effect of making b very large or very small. As $b \to 0$ so that the kernel varies slowly as the arguments change, then the eigenvalues decay very rapidly, and the Gaussian corresponding to the $k = 0$ eigenfunction dominates in the limit. On the other hand, for $b \to \infty$ so that $B \to 1$, very many eigenfunctions are needed to well-approximate the process.

The result for the eigenvalues and eigenfunctions is readily generalized to the multivariate case when the kernel and Gaussian density are products of the univariate expressions, as the eigenfunctions and eigenvalues will simply be products too. For the case that a and b are equal on all N dimensions, the degeneracy of the of the eigenvalue $\left(\frac{2a}{A}\right)^{N/2} B^k$ is $\binom{k+N-1}{N-1}$ which is $O(k^{N-1})$. As $\sum_{j=0}^{k} \binom{j+N-1}{N-1} = \binom{k+N}{N}$ we see that the $\binom{k+N}{N}$th eigenvalue has the value $\left(\frac{2a}{A}\right)^{N/2} B^k$, and this can be used to determine the rate of decay of the spectrum.

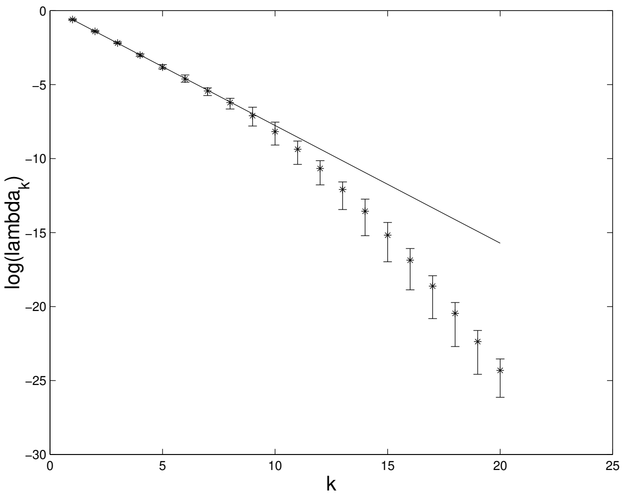

Figure 2. A plot of the log eigenvalue against the index of the eigenvalue. The straight line is the theoretical relationship $\log \lambda_k = k \log B + \frac{1}{2}\log(2a/A)$ derived from equation 7. The centre point (marked with a star) in the "error bar" is the log of the average value of λ_k^{mat}/n. The upper and lower ends of the error bars are the maximum and minimum values of $\log(\lambda_k^{mat}/n)$ respectively taken over the ten repetitions.

The approximation result for the convergence of the eigenvalues of the matrix to those of the underlying process has been checked numerically for the case that $p(x)$ is Gaussian and $K(x, y)$ is the RBF kernel. 500 samples were drawn from the Gaussian density $p(x)$ with $a = 1$. The Gram matrix was then calculated (with $b = 3$) and the eigenvalue spectrum computed. As the result depends upon the random points chosen for the sample, this process was repeated ten times. Figure 2 shows a plot of the log of the average value of λ_k^{mat}/n against the index k of the eigenvalue ($k = 1, 2, \ldots$). Also shown are maximum and minimum values of $\log(\lambda_k^{mat}/n)$ taken over the ten runs, and the theoretical straight line relationship. We observe that the agreement is good for small k, but deteriorates for larger k.

2.2 The Importance of Large-λ Eigenfunctions

The argument we wish to make now is that it is basis functions with large eigenvalue which are important. To see this, consider first a related regression problem. We consider a covariance function with basis functions ϕ_i and eigenvalues λ_i. Say that the target function is of the form $f(\mathbf{x}) = \sum_i \alpha_i \phi_i(\mathbf{x})$ for some fixed α_is, and that we observe n noisy samples from this function (with noise variance σ^2). Then the posterior mean $\hat{\alpha}_i$ estimated for each α_i is roughly

$$\hat{\alpha}_i \sim \frac{\lambda_i}{\lambda_i + \frac{\sigma^2}{n}} \alpha_i \tag{10}$$

(e.g., see Ferrari Trecate et al., 1999). Notice that this is a *shrinkage* of the true α_is. Hence eigenfunctions with $\lambda_i \ll \sigma^2/n$ are effectively "zeroed-out" of the calculation.

Suppose there is only one non-zero α (say the j-th). Due to sampling of the n datapoints, the other $\hat{\alpha}_i$s will not be identically zero. Hence if λ_j is small relative to σ^2/n, the signal component j will be drowned out relative to the noise in the other eigenfunctions for which $\lambda_i > \sigma^2/n$.

In the corresponding classification problem we let $f(\mathbf{x}) = \sum_i \alpha_i \phi_i(\mathbf{x})$ and $\pi(\mathbf{x}) = \sigma(f(\mathbf{x}))$ where $\sigma(z) = (1 + e^{-z})^{-1}$, the logistic function. A single data point is generated by sampling an input value $\mathbf{x}_k$, and then generating a label $t_k \in \{0, 1\}$ from a Bernoulli distribution with parameter $\pi(\mathbf{x})$. A data set of n points is generated by applying this method independently n times. If we consider the problem as above where all α's are set to zero except for α_j, then it turns out that the estimated parameter $\hat{\alpha}_j$ will be severely shrunken if λ_j is smaller than some critical value. For small α_j, so that we can linearize the sigmoid function around input 0, we obtain shrinkage if $\lambda_j < \frac{4}{n}$, where the factor 4 arises from the slope of the logistic function (which is 1/4) at input 0 (see Appendix B for details). For larger α_j, numerical experiments indicate that shrinkage occurs if $\lambda_j n < C$ with $C > 4$. Overall we conclude that eigenfunctions will not be useful for defining decision boundaries in the classification problem unless $\lambda_j n > 4$.

Note also that this condition is on the eigenvalues of the process. Using the relationship between the eigenvalues of the Gram matrix and the process, we expect eigenfunctions will be relevant if $\lambda_j^{mat} > 4$.

3. Consequences of the Density Dependence of the Eigenproblem

Above we have seen how the input density affects the eigenfunctions of the RBF kernel when $p(\mathbf{x})$ is Gaussian. In this section we first study the effect of multimodality on the eigenfunctions, and then draw out some general consequences of the density dependence. Also, eigenvalue spectra are obtained for some test problems.

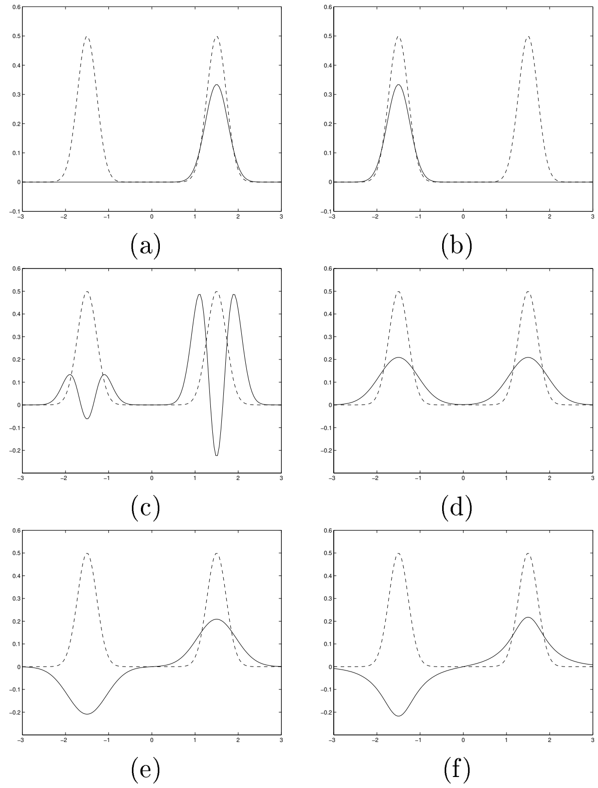

Figure 3. In each figure the dashed line is proportional to the density $p(x)$ from a mixture of two Gaussians. Plots (a), (b) and (c) show the first, second and fifth eigenfunctions for a RBF covariance kernel and $l = 0.2$. Plots (d) and (e) show the first and second eigenfunctions for a RBF kernel and $l = 0.4$. Plot (f) shows the second eigenfunction for an Ornstein-Uhlenbeck kernel with $l = 0.2$.

3.1 The Effect of Density Dependence on a Model Problem

To study further the effects of density dependence, we consider a one-dimensional example where $p(x)$ is a mixture of Gaussians with means at ± 1.5 and variances of 0.05. The covariance function is of RBF-type, i.e. $K(x, y) = \exp -(x - y)^2/2l^2$.

For a lengthscale of $l = 0.2$, the first two eigenfunctions are shown in Figures 3(a) and (b). Notice how each of the eigenfunctions is localized within one peak of the density. However, this behaviour does not continue for the later eigenfunctions; for example the fifth eigenfunction (shown in Figure 3(c)) is non-zero under both peaks. For a longer lengthscale ($l = 0.4$) a rather different behaviour is observed, with the first and second eigenfunctions as shown in Figures 3(d) and (e). Notice that the pair of basis functions (a) and (b) (for $l = 0.2$) or the basis function (e) (for $l = 0.4$) are very useful for discriminating under which mode of the density a x-point lies.

These findings are not restricted to the RBF covariance function. For example, Figure 3(f) shows the second eigenfunction belonging to the Ornstein-Ulhenbeck covariance function $K_{OU}(x,y) = \exp -|x - y|/l$. The first eigenfunction is similar to Figure 3(d). The OU covariance function corresponds to a very rough process (which is mean-squared continuous but not differentiable), which contrasts with the RBF (or squared exponential) process which is infinitely mean-square differentiable.

The effect of the density dependence can be observed not only on the eigenfunctions, but also on the eigenvalues. If we compare the eigenvalues for the bimodal distribution with those for the same kernel function, but for a single Gaussian with the same mean and variance as the bimodal distribution, we observe that the eigenvalues of the bimodal distribution decay more rapidly, and thus that fewer basis functions are needed to approximate the process so the same precision (i.e. so that the fraction of variance explained by M basis functions $\sum_{i=1}^{M} \lambda_i / \sum_{i=1}^{\infty} \lambda_i$ is greater for the bimodal problem).

3.2 General Consequences of the Density Dependence of the Eigenproblem

Above we have seen that (i) the basis functions derived from an eigenfunction expansion of the kernel have density dependence, and (ii) that it is only functions that have a relatively large eigenvalue that we can hope to extract from finite data samples.

In a two-class classification task we assume that the density function $p(\mathbf{x})$ is at least bimodal (it may be more if each class is made up of sub-classes). In order to obtain a good separation of the classes then the basis functions need to be able to be linearly combined to produce a good decision boundary between the classes.

In both examples in Figure 3, the large-eigenvalue basis functions depend strongly on the density and create useful basis functions for the class-discrimination task. Although the exact form of the basis functions will depend on the chosen kernel function, we believe that this density dependence is ubiquitous, and helps to explain the success of kernel-based classifiers. This effect may help to explain the observations by Schölkopf et al. (1995) that kernel methods using different kernels (polynomial, RBF and MLP kernels; this last one has the form $K(\mathbf{x}, \mathbf{y}) = \tanh(\kappa \mathbf{x}.\mathbf{y} - \theta)$) obtain similar classification results.

One major problem with kernel methods is that the computational complexity if the desired solution of the problem is typically of $O(n^3)$. This is true of the matrix inversion needed in Gaussian process prediction (e.g. in the regression case), and also for the quadratic programming problem problem for Support Vector Machines. In (Ferrari Trecate et al., 1999) the authors used a finite dimensional basis to approximate the Gaussian process, but this is only possible if the density $p(\mathbf{x})$ is known and for special cases where the eigenfunctions are available analytically. However, from equation 5 we know that the eigenvalues of the covariance matrix K should decay at a similar rate to those of the continuous problem. Hence the behaviour of the eigenvalues mentioned in section 3.1, where they decrease more rapidly than we would expect based on a single Gaussian model of the data, suggests that a low-rank approximation of the covariance matrix K could be used, and that this would give the potential speedups for kernel-based predictors. We investigate this idea in the next section.

Table 1. Table showing the values of M_{95} and M_{99} for each of the data sets. n denotes the number of training examples and d denotes the number of input features. $\%_{95}$ and $\%_{99}$ give M_{95} and M_{99} as a percentage of n.

dataset	n	d	M_{95}	$\%_{95}$	M_{99}	$\%_{99}$
crabs	80	5	3	3.8	5	6.3
pima	200	7	3	1.5	6	3.0
ringnorm	400	20	1	0.3	60	15.0
twonorm	300	20	10	3.3	28	9.3
wdbc	300	30	5	1.7	13	4.3
waveform	800	21	30	3.8	112	14.0
USPS	7291	256	268	3.7	1508	20.7

3.3 The Eigenspectrum of K for Some Test Problems

The eigenspectrum of the K matrix was calculated for the six datasets that were studied in (Seeger, 2000) and for the USPS handwritten digit database. This calculation is similar to those carried out in the kernel PCA paper of Schölkopf et al. (1998) but here we focus on the fraction of variance explained by a subset of the eigenvectors, rather than on visualization of the data or use as a preprocessing step for classification.

The six problems are the *Leptograpsus crabs, Pima Indian diabetes, Wisconsin Breast Cancer, Ringnorm, Twonorm* and *Waveform* datasets, available from `http://www.cs.utoronto.ca/~delve` and `http://www.ics.uci.edu/~mlearn/MLRepository.html`. For each of the six problems the whole dataset was normalized to have zero mean and unit variance on each input dimension and a training set was picked at random. The squared-exponential or RBF kernel

$$K(\mathbf{x}, \mathbf{y}) = C \exp\left(-\frac{1}{2d}\sum_i w_i (x_i - y_i)^2\right) + v \quad (11)$$

was used, with the parameters C, v and the w_i's set to values determined in Seeger (2000) for each data set. Here d denotes the number of input features, w_i denotes the inverse squared lengthscale for the ith input feature, C is a scaling constant and v is the prior variance associated with the constant basis function $\phi_0 \equiv 1$.

For the USPS dataset, we used the same kernel parameters as in Schölkopf et al. (1999a), namely $K(\mathbf{x}, \mathbf{y}) = \exp(- \parallel \mathbf{x} - \mathbf{y} \parallel^2 /(0.5 \cdot 16^2))$ where the width 0.5 equals twice the average of the data variance on each dimension.

In Table 1 we show the number of datapoints used (n) and the number of eigenvalues needed (M_{95} and M_{99}) to explain 95% and 99% of the variance.[2] It is noticeable that in all cases M_{95} and M_{99} are much smaller than n, and indeed M_{95} is always less than 4% of n. M_{99} is a small fraction of n for the *crabs*, *pima* and *wdbc* data sets. All of these three are real problems, while the *ringnorm*, *twonorm* and *waveform* are synthetic problems that all involve spherically symmetric Gaussian distributions in around 20 dimensions. It is likely that the features in the real problems are correlated so that the $\mathbf{x}$ points lie on a lower dimensional manifold in input space, and that this gives a faster decay for the eigenvalues as the intrinsic dimensionality of the data will be less than the number of features. In contrast, all of the synthetic problems have intrinsic dimensionality equal to the number of features.

4. Exploiting the Eigenstructure in Practice

Perhaps the most obvious way to make use of a rapidly decaying eigenspectrum is to approximate the eigendecomposition of the matrix K. Let K have eigenvalues $\lambda_1 \geq \lambda_2 \ldots \geq \lambda_n \geq 0$ and corresponding eigenvectors denoted by $\mathbf{v}_i$. Then $K = \sum_{i=1}^n \lambda_i \mathbf{v}_i \mathbf{v}_i^T$. However, if the eigenvalue spectrum decays rapidly this suggest we can approximate K with $K_M \stackrel{def}{=} \sum_{i=1}^M \lambda_i \mathbf{v}_i \mathbf{v}_i^T$ for $M < n$. This approximation was used with the Gaussian Process classifier described in (Seeger, 2000) using $M = M_{99}$. The results are shown in Table 2, and show that the results are generally very close (and sometimes slightly better) than using the full matrix.

Table 2. Table showing the number of test errors on each of the six problems using either the full covariance matrix (column 4), or a low-rank approximation (column 5). n denotes the number of training examples and d denotes the number of input features.

data set	n	d	full K	reduced K
crabs	80	5	4	4
pima	200	7	68	67
ringnorm	400	20	184	177
twonorm	300	20	297	302
wdbc	300	30	8	8
waveform	800	21	221	212

A similar experiment has also been carried out for the USPS data set on the task of discriminating the examples of “4”s from the rest of the digits. The results were that using both 256 and 1024 eigenvectors (less than M_{95} and M_{99} respectively) with the kernel as specified in section 3.3 gave the same error rate as the full Gaussian process classifier.

If the eigenvectors/values are available, then the computations for the Gaussian process become $O(M^2 n)$ rather than $O(n^3)$. This complexity is obtained by using the Woodbury formula to invert the matrix $(K + W^{-1})$, where W is a diagonal matrix (see Williams and Barber (1998) for further details). In the Gaussian process regression case the complexity is $O(Mn)$ if the eigendecomposition is available. Of course in general the eigenvectors are not available unless they have been computed, and this has $O(n^3)$ complexity. However, note that there are routines which return the M largest eigenvalues of a matrix. As $\sum_{i=1}^n \lambda_i$ can be computed as $\mathrm{tr}(K)$, the fraction of variance explained is readily calculated.

An alternative to using the eigenexpansion explicitly is to use iterative methods to solve the optimization problem that needs to be solved in the Gaussian process classifier. For example biconjugate gradient methods can be used; this mirrors the use of conjugate gradient methods for Gaussian process regression as described in Gibbs and MacKay (1997). Such methods will tend to converge quickly (i.e. with a number of iterations much less than n) when the eigenspectrum is rapidly decaying. By such methods we can obtain the advantages of a low-rank approximation without explicitly computing the eigenvectors. An example of the kinds of result available is the following theorem from Luenberger (1984; section 8.5) concerning

[2] In fact to obtain good conditioning of the eigenvalue problem we computed the eigenvalues of $K + \sigma_J^2 I$, where σ_J^2 is a “jitter” variance, and then subtracted off σ_J^2. Values of 10^{-2} and 10^{-6} were used, obtaining similar results.

the conjugate-gradient method for solving the linear system $Q\mathbf{x} = \mathbf{b}$.

Suppose the symmetric positive definite matrix Q has $n - M$ eigenvalues in the interval $[a, b]$, $a > 0$, and the remaining M eigenvalues are greater than b. Then after $M + 1$ steps, the error $E(\mathbf{x}) = (\mathbf{x} - \mathbf{x}^*)^T Q(\mathbf{x} - \mathbf{x}^*)$ satisfies

$$E(\mathbf{x}_{M+1}) \leq \left(\frac{b-a}{b+a}\right)^2 E(\mathbf{x}_0), \tag{12}$$

where $\mathbf{x}^*$ is minimum of $E(\mathbf{x})$ and $\mathbf{x}_0$ is the starting value of $\mathbf{x}$. In a Gaussian process classifier one usually adds a "jitter" term $\sigma_J^2 I$ to K to give better conditioning of the covariance matrix (Neal, 1998). If there are many small eigenvalues of the unjittered matrix then b is close to a and the factor $(b - a/b + a)$ will be small.

5. Discussion

This paper makes a number of contributions to the study of kernel machines, including (i) a clarification of the relationship between the eigenvalues of the process and those of the Gram matrix in the presence of an input density $p(\mathbf{x})$, (ii) a discussion of how this density dependence can give rise to basis functions tuned for classification problems and how it may tend to lessen differences between kernels, and (iii) observations on the eigenspectrum of the matrix K and consequences for the optimization problem associated with constructing a classifier.

Schölkopf et al. (1999b) have discussed generalization bounds based on the eigenvalues of the Gram matrix. Although their results are based on entropy numbers of operators, the underlying idea appears to be concerned with the importance of large-λ eigenvalues; we look forward to exploring this connection further.

Appendix A

This derivation is based on that of Zhu et al. (1998) and is included for completeness. Consider approximating a fixed function $f(\mathbf{x})$ with a linear approximator $g(\mathbf{x}) = \sum_i w_i \psi_i(\mathbf{x})$. If we choose the basis functions to be orthonormal[3] wrt $p(\mathbf{x})$ so that $\int \psi_i(\mathbf{x})\psi_j(\mathbf{x})p(\mathbf{x})\, d\mathbf{x} = \delta_{ij}$, then when minimizing the error

$$J_f = \int (f(\mathbf{x}) - g(\mathbf{x}))^2 p(\mathbf{x})\, d\mathbf{x} \tag{13}$$

[3]This can always be achieved using a Gram-Schmidt procedure.

with respect to the w_i's, we obtain $\hat{w}_i = \int f(\mathbf{x})p(\mathbf{x})\psi_i(\mathbf{x})\, d\mathbf{x}$ and

$$J_f^{min} = \int f^2(\mathbf{x})p(\mathbf{x})\, d\mathbf{x} - \sum_i \hat{w}_i^2 \tag{14}$$

We now take the expectation over f using a zero-mean Gaussian process prior over functions, having covariance function $K(\mathbf{x}, \mathbf{y}) = E[f(\mathbf{x})f(\mathbf{y})]$ to obtain

$$E[J_f^{min}] \stackrel{def}{=} J = \int K(\mathbf{x}, \mathbf{x})p(\mathbf{x})d\mathbf{x} - \sum_{i,j} \int \int \psi_i(\mathbf{x})\psi_j(\mathbf{y})K(\mathbf{x}, \mathbf{y})p(\mathbf{x})p(\mathbf{y})d\mathbf{x}d\mathbf{y}. \tag{15}$$

To minimize $J \stackrel{def}{=} J_1 + J_2$ we need to maximize the second term J_2 of equation 15. We transform the problem by setting $\tilde{\psi}_i(\mathbf{x}) = p^{1/2}(\mathbf{x})\psi(\mathbf{x})$ and $\tilde{K}(\mathbf{x}, \mathbf{y})$ as in equation 4 to obtain

$$J_2 = \sum_{i,j} \int \int \tilde{\psi}_i(\mathbf{x})\tilde{K}(\mathbf{x}, \mathbf{y})\tilde{\psi}(\mathbf{y})d\mathbf{x}d\mathbf{y}. \tag{16}$$

This maximization (with constraints $\int \tilde{\psi}_i(\mathbf{x})\tilde{\psi}_j(\mathbf{x})d\mathbf{x} = \delta_{ij}$) is a standard problem whose solution is that the optimal choice for $\{\tilde{\psi}_i\}$ is the M leading eigenfunctions of $\tilde{K}(\mathbf{x}, \mathbf{y})$, i.e.

$$\int \tilde{K}(\mathbf{x}, \mathbf{y})\tilde{\phi}_i(\mathbf{x}) = \lambda_i \tilde{\phi}_i(\mathbf{y}). \tag{17}$$

Substituting for $\tilde{K}(\mathbf{x}, \mathbf{y})$ we obtain

$$\int p^{1/2}(\mathbf{x})K(\mathbf{x}, \mathbf{y})p^{1/2}(\mathbf{y})\tilde{\phi}_i(\mathbf{x})d\mathbf{x} = \lambda_i \tilde{\phi}_i(\mathbf{y}) \tag{18}$$

which can be re-arranged using $\tilde{\phi}_i(\mathbf{x}) = p^{1/2}(\mathbf{x})\phi_i(\mathbf{x})$ to obtain equation 3.

Appendix B

In this appendix we show that for the binary classification case, shrinkage occurs unless $\lambda_j n \gtrsim 4$.

Consider binary data generated from an underlying function $y(\mathbf{x}) = \alpha\phi(\mathbf{x})$ using the logistic function so that $P(\text{class } 1|\mathbf{x}) \stackrel{def}{=} \pi(\mathbf{x}) = \sigma(y(\mathbf{x}))$ and $P(\text{class } 0|\mathbf{x}) = 1 - \sigma(y(\mathbf{x}))$. $\phi(\cdot)$ is normalized so that $\int \phi^2(\mathbf{x})p(\mathbf{x})d\mathbf{x} = 1$. We place a Gaussian prior on α with variance λ. Denoting the label corresponding to input $\mathbf{x}_i$ as $t_i \in \{0, 1\}$ the penalized log likelihood is given by

$$J = \sum_{i=1}^n t_i y_i - \sum_{i=1}^n \log(1 + \exp y_i) - \frac{1}{2}\frac{\alpha^2}{\lambda} \tag{19}$$

where $y_i = y(\mathbf{x}_i)$. To obtain the MAP solution, we differentiate, and equate to zero to obtain

$$\sum_i (t_i - \hat{\pi}_i)\phi(\mathbf{x}_i) - \frac{\hat{\alpha}}{\lambda} = 0, \tag{20}$$

where $\hat{\alpha}$ denotes the MAP value of α and $\hat{\pi}_i = \sigma(\hat{\alpha}\phi(\mathbf{x}))$. We wish to compare $\hat{\alpha}$ with α^*, the value of alpha used to generate the data.

If α is small, $\sigma(\alpha\phi(\mathbf{x})) \simeq \frac{1}{2} + \frac{1}{4}\alpha\phi(\mathbf{x})$ using a Taylor expansion about 0. Using this and approximating a sum by an integral we find

$$\begin{aligned}\sum_i t_i\phi(\mathbf{x}_i) &\simeq n\int \sigma(\alpha^*\phi(\mathbf{x}))\phi(\mathbf{x})p(\mathbf{x})d\mathbf{x} \\ &\simeq \frac{n}{2}\int \phi(\mathbf{x})p(\mathbf{x})d\mathbf{x} + \frac{n}{4}\alpha^*.\end{aligned}$$

Similarly we find that

$$\sum_i \hat{\pi}_i\phi(\mathbf{x}_i) \simeq \frac{n}{2}\int \phi(\mathbf{x})p(\mathbf{x})d\mathbf{x} + \frac{n}{4}\hat{\alpha}, \tag{21}$$

and hence

$$\sum_i (t_i - \hat{\pi}_i)\phi(\mathbf{x}_i) \simeq \frac{n}{4}(\alpha^* - \hat{\alpha}). \tag{22}$$

Substituting this into equation 20 we find

$$\frac{\hat{\alpha}}{\alpha^*} = \frac{\lambda}{\lambda + \frac{4}{n}} \tag{23}$$

as claimed.

Acknowledgements

We thank Chris Bishop, Manfred Opper, Peter Sollich, Mike Tipping and the anonymous ICML referees for their comments on an earlier draft of this paper. MS gratefully acknowledges support through a research studentship from Microsoft Research Ltd.

References

Baker, C. T. H. (1977). *The numerical treatment of integral equations*. Oxford: Clarendon Press.

Ferrari Trecate, G., Williams, C. K. I., & Opper, M. (1999). Finite-dimensional approximation of Gaussian processes. In M. S. Kearns, S. A. Solla and D. A. Cohn (Eds.), *Advances in Neural Information Processing Systems 11*, 218–224. MIT Press.

Gibbs, M., & MacKay, D. J. C. (1997). Efficient Implementation of Gaussian Processes. Unpublished manuscript. Cavendish Laboratory, Cambridge, UK. Also available from `http://wol.ra.phy.cam.ac.uk/mackay/homepage.html`.

Luenberger, D. G. (1984). *Linear and nonlinear programming*. Reading, MA: Addison-Wesley. 2nd ed.

Neal, R. M. (1998). Regression and classification using Gaussian process priors (with discussion). In J. M. Bernardo et al. (Eds.), *Bayesian statistics 6*, 475–501. Oxford University Press.

Ritter, K., Wasilkowski, G. W., & Woźniakowski, H. (1995). Multivariate integration and approximation of random fields satisfying Sacks-Ylvisaker conditions. *Annals of Applied Probability*, *5*, 518–540.

Schölkopf, B., Burges, C., & Vapnik, V. (1995). Extracting support data for a given task. In *Proceedings of the First International Conference on Knowledge Discovery and Data Mining*. Menlo Park, CA: AAAI Press.

Schölkopf, B., Mika, S., Burges, C. J. C., et al. (1999a). Input space vs feature space in kernel-based methods. *IEEE Transactions on Neural Networks*, *10(5)*, 1000–1017.

Schölkopf, B., Shawe-Taylor, J., Smola, A. J., & Williamson, R. C. (1999b). *Generalization bounds via eigenvalues of the Gram matrix* (Technical Report NC2-TR-1999-035). Available from `http://www.neurocolt.com`.

Schölkopf, B., Smola, A., & Müller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, *10*, 1299–1319.

Seeger, M. (2000). Bayesian model selection for support vector machines, Gaussian Processes and other kernel classifiers. In S. A. Solla, T. K. Leen and K.-R. Müller (Eds.), *Advances in Neural Information Processing Systems 12*. Cambridge, MA: MIT Press.

Vapnik, V. N. (1995). *The nature of statistical learning theory*. New York: Springer Verlag.

Wahba, G. (1990). *Spline models for observational data*. Philadelphia, PA: Society for Industrial and Applied Mathematics. CBMS-NSF Regional Conference series in applied mathematics.

Williams, C. K. I., & Barber, D. (1998). Bayesian classification with Gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20(12)*, 1342–1351.

Zhu, H., Williams, C. K. I., Rohwer, R. J., & Morciniec, M. (1998). Gaussian regression and optimal finite dimensional linear models. In C. M. Bishop (Ed.), *Neural networks and machine learning*. Berlin: Springer-Verlag.