

Kernel extrapolation

S.V.N. Vishwanathan^{a,b,*}, Karsten M. Borgwardt^{c,1}, Omri Guttman^{a,b}, Alex Smola^{a,b}

^aStatistical Machine Learning Program, National ICT Australia

^bRSISE, Australian National University, Canberra, 0200 ACT, Australia

^cInstitute for Computer Science, Ludwig-Maximilians-University Munich, Oettingenstr. 67, 80538 Munich, Germany

Abstract

We present a framework for efficient extrapolation of reduced rank approximations, graph kernels, and locally linear embeddings (LLE) to unseen data. We also present a principled method to combine many of these kernels and then extrapolate them. Central to our method is a theorem for matrix approximation, and an extension of the representer theorem to handle multiple joint regularization constraints. Experiments in protein classification demonstrate the feasibility of our approach.

© 2006 Elsevier B.V. All rights reserved.

Keywords: Kernel methods; Regularization; Graph kernels; Protein classification

1. Introduction

The form of the kernel is critical for achieving good generalization in many machine learning problems employing kernel methods [12]. Kernel design is typically guided by three criteria. Firstly, the kernel should reflect prior knowledge relevant to the particular problem at hand. Secondly, it should be easy to evaluate the kernel for prediction purposes. Finally, computation of the kernel matrix on unseen data should be possible without limitations.

The first two goals can lead to conflicting requirements: for instance, we may wish to limit ourselves to a small set of functions (Fourier basis, Fisher scores, nearest neighbours, a small set of kernel functions, etc.) for the sake of efficiency. On the other hand, we may want to enforce an estimate with bounded Sobolev norm (as in the case of the Laplacian kernel), a pseudo-differential operator (as for

the Gaussian kernel), a discrete flatness functional (as for graph kernels), or locally weighted smoothness functionals.

The practitioner has one of two unsatisfactory choices: either choose a kernel suggested by practical considerations or use only a small subset of the basis functions.

Sometimes, information about the data can only be effectively captured by evaluating two different kernel functions. For instance, if the data has both discrete and continuous valued attributes, a graph kernel might capture interactions among the discrete variables while a Fisher kernel might be better suited to model the continuous variables. A practitioner is then forced to either employ a simple combination of kernels, with no control over the joint regularization properties, or to choose one kernel over the other.

Extension to unseen data is a problematic issue in the context of kernels on graphs [8], or when computing a kernel matrix via semidefinite optimization [14]. In this paper, we suggest a strategy for efficiently extending many well known kernels to unseen data. Central to this is a notion of matrix approximation under a semi-definite constraint. We then discuss a principled way of combining such kernels which imposes a smoothness constraint on the estimator with respect to each kernel, and proceed to address the practitioner's dilemma in a principled way.

*Corresponding author.

E-mail addresses: SVN.Vishwanathan@nicta.com.au (S.V.N. Vishwanathan), kb@dbis.ifi.lmu.de (K.M. Borgwardt), Omri.Guttman@nicta.com.au (O. Guttman), Smola@nicta.com.au (A. Smola).

¹Part of this work was done while visiting NICTA.

1.1. Notation

For a matrix A we use A^\dagger to denote its Moore–Penrose pseudo inverse, $\sigma_i(A)$ to denote its i th largest singular value, and $\lambda_i(A)$ to denote its i th largest eigenvalue. We use $A \succeq 0$ to indicate that A is positive semi-definite and $A \succ 0$ to denote that it is positive definite. Analogously, we use $A \succeq \bar{A}$ ($A \succ \bar{A}$) to indicate that $A - \bar{A} \succeq 0$ ($A - \bar{A} \succ 0$). The Von-Neumann Schatten p -norms are defined as ([3])

$$\|A\|_p := \left(\sum_i |\sigma_i(A)|^p \right)^{1/p} \quad \text{for } p \geq 1. \quad (1)$$

It is easy to see that $\|A\|_2$ is the Frobenius norm; $\|A\|_\infty$ is the operator norm or spectral norm; and $\|A\|_1 = \text{tr } A$ for positive semidefinite A . We use \mathbf{I} to denote an identity matrix and $\mathbf{1}$ to denote the vector of all ones.

We denote by \mathcal{X} the space of observations, and \mathcal{Y} be the space of labels or targets, which we wish to predict. Let $X := \{x_1, \dots, x_m\} \subset \mathcal{X}^m$ be the set of observations, and let $Y := \{y_1, \dots, y_m\} \subset \mathcal{Y}^m$ be the corresponding labels. We use \tilde{X} to denote the matrix whose i th row corresponds to $x_i \in X$. The matrix \tilde{Y} is defined analogously.

A function $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ will denote a Mercer kernel with a corresponding reproducing kernel Hilbert space (RKHS) \mathcal{H}_k . The kernel function k evaluated on $X \times X$ gives rise to the kernel matrix K . Moreover, let

$$\phi : \mathcal{X} \rightarrow \mathbb{R}^n, \quad (2)$$

for n finite be a feature map, and let $Q \in \mathbb{R}^{n \times n}$ with $Q \succeq 0$. Then a kernel k_Q is defined by ϕ and Q as

$$k_Q(x, x') := \phi(x)^\top Q \phi(x'). \quad (3)$$

The kernel matrix associated with k_Q is denoted by K_Q . With some abuse of notation we will use \mathcal{H}_Q to denote the RKHS corresponding to k_Q .

Functions $f : \mathcal{X} \rightarrow \mathbb{R}$ are understood to be members of the corresponding RKHS \mathcal{H}_k . In the finite dimensional cases it will be convenient to denote them by

$$f(x) = \langle \phi(x), w \rangle \quad \text{with } w \in \mathbb{R}^n. \quad (4)$$

1.2. Setting

Quite often we may find ourselves in the situation where we are given a kernel matrix K which is defined only on a small subset of the input domain, say only on X rather than \mathcal{X} . Moreover, we may be given a feature map ϕ as in Eq. (2). The problem arising in this context is to find a matrix Q such that K_Q most closely resembles K while allowing one to *extrapolate* the behaviour of K to novel data drawn from \mathcal{X} via the kernel k_Q .

For instance, K may be the kernel matrix arising from a nearest neighbour graph kernel on X that we wish to extend to additional data without the need to recompute the entire kernel matrix [1]. Likewise, K may be given by semidefinite optimization for the low-dimensional embed-

ding of data [14] and we may wish to extend this projection to additional data. Details and further examples will be given in Section 4.

When $n = m$ and the design matrix

$$\Phi \in \mathbb{R}^{n \times m} \quad \text{with } \Phi_{ij} = [\phi(x_j)]_i$$

has full rank, it is easy to see that the linear system

$$\underset{Q \succeq 0}{\text{argmin}} \|K - K_Q\|_p = \underset{Q \succeq 0}{\text{argmin}} \|K - \Phi^\top Q \Phi\|_p$$

has an exact solution $Q = (\Phi^{-1})^\top K \Phi^{-1}$, regardless of the matrix norm used, as the residual between K and $\Phi^\top Q \Phi$ vanishes.

However, for $n \neq m$ or for cases where Φ is rank deficient, it is far from clear how to best determine Q . Should one minimize the 2-norm, the Frobenius norm, or another matrix norm? What further constraints should be imposed on Q ? How does this kernel align with our three criteria for kernel design? These are some of the questions we will address in the following sections.

Sometimes we might be given two kernel matrices K_1 and K_2 and their corresponding feature maps ϕ_1 and ϕ_2 . The problem then is to combine these two kernels and estimate a function f which has a small norm in both \mathcal{H}_{k_1} and \mathcal{H}_{k_2} . We address this problem by extending the representer theorem to handle multiple RKHS norm constraints.

1.3. Paper outline

In Section 2 we state the main algorithmic result of the paper regarding the approximation of matrices with respect to subspace constraints. Section 3 contains the extended representer theorem and its use for joint regularization. Section 4 contains a list of applications of the obtained results to various problems in kernel methods. Some of these applications are backed up by experiments in Section 5. We conclude with a discussion and outlook in Section 6.

2. Matrix approximation

To judge the proximity between K and K_Q we need to establish a criterion of optimality. For this purpose we propose the use of the Von-Neumann Schatten p -norms. All $\|\cdot\|_p$ norms are monotonic in the magnitude of the singular values and hence the eigenvalues. Consequently, this set of matrix norms will give us a wide array of criteria to measure the proximity between the two matrices K and K_Q .

Additionally, we impose a projection constraint on Q : clearly, Q needs to be positive semidefinite for k_Q to be a kernel. However, this need not be true for the residual $K - K_Q$. Since we know that \mathcal{H}_K is a RKHS and \mathcal{H}_Q is a RKHS, it is natural to demand that the RKHS \mathcal{H}_K be decomposable into two orthogonal subspaces, one given by \mathcal{H}_Q and the other orthogonal to it. This is equivalent to

demanding that $K - K_Q \geq 0$. We call this constraint the shear constraint.

It turns out that it is sufficient to impose the constraint $K - K_Q \geq 0$ since this automatically ensures that $Q \geq 0$.

2.1. Key lemma

Lemma 1 (Matrix approximation). Let K, K_Q , and Φ be as defined above, and $\Phi = UDV^T$ denote the singular value decomposition (SVD) of Φ . Write $V = [V_1, V_2]$, for $V_1 \in \mathbb{R}^{m \times n}$ and $V_2 \in \mathbb{R}^{m \times m-n}$. For all Von-Neumann Schatten p -norms we have

$$\operatorname{argmin}_{Q \geq 0, K - K_Q \geq 0} \|K - K_Q\|_p = (\Phi^\dagger)^\top [K - P]\Phi^\dagger, \tag{5}$$

where

$$P = KV_2(V_2^\top KV_2)^{-1}V_2^\top K. \tag{6}$$

The lemma looks daunting but has a rather intuitive interpretation. If we did not enforce the shear constraint $K - K_Q \geq 0$, then the minimizer of $\|K - K_Q\|_p$ would simply be $(\Phi^\dagger)^\top K \Phi^\dagger$. This can be easily verified by using a SVD argument. In order to enforce the shear constraint, we have to correct the vanilla projection by the matrix P , which can be seen as a projection of K on the space spanned by KV_2 . This ensures that we do not distort the residual space $K - K_Q$ orthogonal to the span of Φ .

Surprisingly, Lemma 1 is independent of the specific Von-Neumann Schatten p -norm used for measuring proximity. One way to understand this result is to realize that these norms are monotonic in the singular values of the matrix. Thus, minimizing with respect to the Frobenius norm (i.e., $p = 2$) is essentially equivalent to minimizing with respect to any value of p . But, minimizing the norm of the residual under the Frobenius norm can be achieved by using the SVD. Therefore, it is not surprising that the lemma makes heavy use of the SVD in order to express the projecting matrix P .

2.2. Proof of lemma

We begin by stating a well known result for the Von-Neumann Schatten p -norms of positive semi-definite matrices. In fact, a similar result holds for any monotonic function of the eigenvalues of a semi-definite matrix.

Lemma 2 (Increasing eigenvalues [3]). If $M_1 \geq M_2 \geq 0$ then

- (1) $\lambda_i(M_1) \geq \lambda_i(M_2)$ for all i .
- (2) $\|M_1\|_p \geq \|M_2\|_p$ for all $p \geq 1$.

The next lemma states a few other properties of positive semi-definite matrices which follow directly from the Schur complement lemma [3].

Lemma 3 (Positive semi-definite matrices). Let $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{(m-n) \times n}$ and $C \in \mathbb{R}^{(m-n) \times (m-n)}$ and

$$M = \begin{bmatrix} A & B \\ B^\top & C \end{bmatrix}.$$

Then the following holds:

- (1) $M \geq 0$ iff $A \geq BC^{-1}B^\top \geq 0$.
- (2) Let $S \in \mathbb{R}^{m \times n}$ be any matrix. If $M \geq 0$ then $S^\top MS \geq 0$.
- (3) If $C \geq 0$ then subject to the constraints $M \geq 0$ we have $\operatorname{argmin}_A \|M\|_p = BC^{-1}B^\top$.

Proof. The first part is simply the Schur complement lemma and the second part is a well known fact about positive semi-definite matrices (see e.g. [3]).

Use the Schur complement lemma to observe that if $M \geq 0$ then

$$\begin{bmatrix} A & B \\ B^\top & C \end{bmatrix} \succeq \begin{bmatrix} BC^{-1}B^\top & B \\ B^\top & C \end{bmatrix} \succeq 0.$$

The third part now follows directly by applying Lemma 2 to the above observation. \square

We now have all the tools required to prove our main lemma.

Proof (Lemma 1). Since orthogonal transformations leave the spectrum unchanged we have

$$\|K - \Phi^\top Q \Phi\|_p = \|V^\top KV - D^\top U^\top QUD\|_p.$$

The matrix Φ has rank n , therefore it is easy to see that

$$D^\top U^\top QUD = \begin{bmatrix} \tilde{Q} & 0 \\ 0 & 0 \end{bmatrix},$$

where $\tilde{Q} \in \mathbb{R}^{n \times n}$ can be written as $\tilde{Q} = D_1 U^\top QUD_1$. The diagonal matrix D_1 is obtained by writing

$$D = \begin{bmatrix} D_1 \\ 0 \end{bmatrix}.$$

Observe that Q is similar to \tilde{Q} since UD_1 is invertible. If we write

$$V^\top KV = \begin{bmatrix} A & B \\ B^\top & C \end{bmatrix}$$

for $A \in \mathbb{R}^{n \times n}$, then we have

$$\|K - \Phi^\top Q \Phi\|_p = \left\| \begin{bmatrix} A - \tilde{Q} & B \\ B^\top & C \end{bmatrix} \right\|_p.$$

Enforcing the constraint $K - K_Q \geq 0$, and using Lemma 3, the norm $\|K - K_Q\|_p$ is minimized when $\tilde{Q} = A - BC^{-1}B^\top \geq 0$. Since Q is similar to \tilde{Q} this implies $Q \geq 0$.

Note that minimizing the norm subject to the constraints $K - K_Q \geq 0$ is sufficient since the solution satisfies $Q \geq 0$.

Solving for \tilde{Q} and substituting terms back into the definition of Q proves the claim. \square

3. Joint regularization

When combining different feature spaces, it may be desirable to find an estimate which is smooth with respect to one regularization operator, while satisfying the constraint of being small with respect to a few other regularizers (e.g., by requiring that the estimate has small variance). This section shows that such optimization problems lead to kernel expansions. It lays the theoretical groundwork for combining kernels on various domains, in our case kernels on graphs and the Fisher kernel.

3.1. Extended representer theorem

Theorem 4 (Joint regularization). Denote by \mathcal{H}_{k_i} with $i \in \{1, \dots, l\}$ a RKHS and let $R_{\text{emp}}[f]$ be a convex empirical risk functional, depending on the function $f : \mathcal{X} \rightarrow \mathbb{R}$ only via its evaluations on the set $X := \{x_1, \dots, x_m\}$. Consider a convex constrained optimization problem

$$\begin{aligned} & \underset{f}{\text{minimize}} \quad R_{\text{emp}}[f] \\ & \text{s.t.} \quad \frac{1}{2} \|f\|_{\mathcal{H}_i}^2 \leq c_i \quad \forall i, \end{aligned} \tag{7}$$

for some $c_i > 0$. Then there exists a RKHS \mathcal{H} with kernel k and scalar product

$$\langle f, g \rangle_{\mathcal{H}} = \sum_{i=1}^m \beta_i \langle f, g \rangle_{\mathcal{H}_i} \quad \text{for some } \beta_i \geq 0 \tag{8}$$

such that the minimizer f^* of Eq. (7) can be written as $f^*(x) = \sum_{i=1}^m \alpha_i k(x_i, x)$, and hence $f^* \in \mathcal{H}$.

Proof. Eq. (7) describes a convex optimization problem. Hence its minimum is unique. Furthermore, we can compute the Lagrange function

$$L(f, \lambda) = R_{\text{emp}}[f] + \sum_{i=1}^l \lambda_i \left(\frac{1}{2} \|f\|_{\mathcal{H}_{k_i}}^2 - c_i \right), \tag{9}$$

with nonnegative Lagrange multipliers λ_i . Since L has a saddle point at optimality, there exists a set of λ_i^* for which the unconstrained minimizer of $L(f, \lambda^*)$ with respect to f coincides with the solution of Eq. (7). Ignoring terms independent of f in L yields

$$R_{\text{emp}}[f] + \sum_{i=1}^n \frac{\lambda_i^*}{2} \|f\|_{\mathcal{H}_{k_i}}^2. \tag{10}$$

Combining the regularization terms in f into one Hilbert space with $\beta_i = \lambda_i^*$ and subsequently appealing to the representer theorem ([13]) concludes the proof. \square

Note that the condition of convexity is necessary: without this requirement on $R_{\text{emp}}[f]$ we would still be able to obtain a local optimum with suitable Lagrange multipliers, but we cannot guarantee that the local optimum is the unique global solution of Eq. (10). Also observe that some of the λ_i in Eq. (10) could vanish, corresponding to inactive constraints in Eq. (7).

It is also easy to see that the above theorem can be extended, in a straightforward manner, to handle norm constraints of the form $\omega_i(\|f\|_{\mathcal{H}_{k_i}}) \leq c_i$ where $\omega_i : [0, \infty) \rightarrow \mathbb{R}$ are strictly monotonic increasing functions.

The consequence of the extended representer theorem is that we can take convex combinations of regularization functionals in order to obtain joint regularizers.

3.2. Kernels and metrics

It is well known ([12]) that for f defined as in Eq. (4) one can exploit linearity in the Hilbert space \mathcal{H} and compute

$$\|f\|_{\mathcal{H}}^2 = w^T M w \quad \text{where } M_{ij} := \langle \phi(x_i), \phi(x_j) \rangle_{\mathcal{H}}. \tag{11}$$

It can be easily verified that using the inverse of M as the metric will yield a kernel with equivalent regularization properties on the subspace spanned by $\phi(\cdot)$.

Lemma 5 (Equivalent kernel [12]). The kernel k arising from $\|f\|_{\mathcal{H}}^2$ on the space spanned by $\phi(\cdot)$ is given by $k(x, x') = \phi(x)^T M^{-1} \phi(x')$, where $M_{ij} = \langle \phi(x_i), \phi(x_j) \rangle_{\mathcal{H}}$.

The importance of this lemma is that it allows us to establish a relation between the matrix Q defining the kernel function k_Q (see Eq. (3)) and the function norm in the space \mathcal{H}_Q . This when combined with the extended representer theorem provides a powerful method for combining various kernels.

3.3. Combining kernels

We consider two matrices $Q_1 \geq 0$ and $Q_2 \geq 0$ defining kernel functions k_{Q_1} and k_{Q_2} via Eq. (3). With slight abuse of notation we use $\|f\|_{Q_i}$ to denote the function norm in \mathcal{H}_{Q_i} . Let $c > 0$ be a constant and, $\lambda \in [0, 1]$ denote a confidence parameter which specifies the amount of regularization we wish to impose on the estimator in \mathcal{H}_{Q_1} and \mathcal{H}_{Q_2} . The following lemma asserts that there is a principled way of obtaining a joint regularizer by combining kernels k_{Q_1} and k_{Q_2} .

Lemma 6 (Joint kernel). Let Q_1, Q_2, c and λ as above. The joint regularization induced by requiring $\|f\|_{Q_1} \leq c/\lambda$ and $\|f\|_{Q_2} \leq c/(1-\lambda)$ is equivalent to requiring $\|f\|_Q \leq c$ where $Q := (\lambda Q_1^{-1} + (1-\lambda)Q_2^{-1})^{-1} \geq 0$ and k_Q is defined via Eq. (3).

Proof. The proof is straightforward. We require that $\|f\|_{Q_1} = w^T Q_1^{-1} w \leq c/\lambda$ and $\|f\|_{Q_2} = w^T Q_2^{-1} w \leq c/(1-\lambda)$. By Theorem 4 this is equivalent to requiring that $w^T (\lambda Q_1^{-1} + (1-\lambda)Q_2^{-1}) w \leq c$. By Lemma 5 the corresponding kernel is induced by $Q := (\lambda Q_1^{-1} + (1-\lambda)Q_2^{-1})^{-1} \geq 0$. \square

3.4. Putting things together

We have discussed two different methods for kernel extrapolation. The first one allows the approximation of a kernel matrix by using only a fixed number of basis

functions. The second method allows for kernels to be combined in order to satisfy joint regularization properties. We can now combine the two results to obtain joint kernels. We first approximate the individual kernel matrices using a common feature map and then combine them using Lemma 6 to obtain a joint kernel.

4. Applications

In this section, we cast a few existing algorithms as special cases of our framework. We then show how the approximation lemma can be used to extrapolate some kernels. Finally, we also show how side information from various sources can be combined using our joint regularization result.

4.1. Reduced set methods

In reduced set methods, we are given the kernel function k and the corresponding kernel matrix K and we are interested in approximating K by using a finite set of basis functions. Without loss of generality, we pick $\{x_1, \dots, x_n\}$ the first n points in the dataset and let $\phi(x) := (k(x_1, x), \dots, k(x_n, x))$. Let $K^{mm} \in \mathbb{R}^{m \times n}$ denote the left sub-matrix of K , while $K^{nn} \in \mathbb{R}^{n \times n}$ be the upper left sub-matrix of K . It is easy to see that $K^{mm} = \Phi^\top$. Some straightforward but tedious algebra, which we omit for reasons of brevity, shows that for P defined by Eq. (6) we have $(\Phi^\dagger)^\top P \Phi^\dagger = 0$. By Lemma 1 it follows that the best approximation to K is given by K_Q with $Q := (K^{mm})^\dagger K ((K^{mm})^\dagger)^\top$. Using the Schur complement lemma ([3]) we can show that $Q \succeq (K^{nn})^{-1}$. Therefore, a good conservative estimate is to use the kernel $K_{\bar{Q}}$ with $\bar{Q} = (K^{nn})^{-1}$, in order to approximate K . This is exactly the approximation chosen for instance in [12]. An attractive feature of this choice is that the metric imposed by $K_{\bar{Q}}$ preserves the regularization imposed by the RKHS \mathcal{H}_k . In other words, from Lemma 5, we have that

$$\|f\|_{\mathcal{H}_{\bar{Q}}}^2 = w^\top M w,$$

where $M \in \mathbb{R}^{n \times n}$ and

$$M_{ij} = [K^{mm}]_{ij} = \langle \phi(x_i), \phi(x_j) \rangle_{\mathcal{H}_k} = k(x_i, x_j).$$

4.2. Extension of graph kernels

An undirected, unweighted graph G consists of an ordered vertex set V and an edge set $E \subseteq V \times V$. If $|V| = m$, then the adjacency matrix of G is given by $W \in \mathbb{R}^{m \times m}$, where $W_{ij} = 1$ if vertex v_i is connected to vertex v_j and $W_{ij} = 0$ otherwise.

The degree of a vertex $v_i \in V$, denoted by d_i , is the number of edges emanating from v_i . The graph Laplacian is given by $H = D - W$ where D is a diagonal matrix with $D_{ii} = \sum_j W_{ij}$. The normalized graph Laplacian is defined as $L = D^{-1/2} H D^{-1/2}$. It is easy to see that the normalized

graph Laplacian is positive semi-definite and its eigenvalues are bounded by 2. Given L and a $\tau > 0$ we can write a kernel on graph G as

$$K = \exp(-\tau L). \tag{12}$$

Such kernels have the property of assigning similarities to the vertices v_i and v_j of G according to the similarity of the diffusion processes starting from them ([8]).

While this setting has been successful in dealing with transductive problems where K could be jointly computed on training and test set, or where the matrix exponential could be approximated efficiently, special methods are required to extend K to novel observations. The main difficulty here is that an enlarged K would immediately lead to a changed matrix exponential for all the terms.

We propose a straightforward method to extend K to novel points. Using a suitable set of basis functions we approximate the kernel matrix K by a matrix K_Q (see Lemma 1). Given a novel point we simply compute the basis functions corresponding to that point and use them to compute the kernel k_Q .

More concretely, we use the Fisher score map as basis functions. This choice is well motivated since the features of the Fisher score map are the sufficient statistics of the underlying density model. Recall that if $p_\theta(x)$ denotes a family of log-differentiable densities parameterized by $\theta \in \mathbb{R}^n$, the Fisher score map of a point x is given by ([4])

$$u_\theta(x) = -\partial_\theta \log p_\theta(x). \tag{13}$$

An exponential family of distributions parameterized by θ can be written as $p_\theta(x) = \exp(\langle t(x), \theta \rangle - g(\theta))$, where $g(\theta)$ is called the log-partition function and $t(x)$ denotes the sufficient statistics of the distribution. The Fisher scores are now given by $-\partial_\theta \log p_\theta(x) = -t(x) + \partial_\theta g(\theta)$. It is well known that $\partial_\theta g(\theta) = \mathbb{E}_\theta[t(x)]$ ([5]). We can see that in the case of the exponential family the Fisher score map is simply the sufficient statistics standardized to zero mean and unit variance, and hence easily computable.

Given a matrix $Q \succeq 0$ we can use the Fisher score map to define a kernel

$$k_Q(x, x') = u_\theta(x)^\top Q u_\theta(x'). \tag{14}$$

If U_θ denotes the Fisher score map for the set of vertices in graph G , then it follows from Eq. (5) that $Q = (U_\theta^\dagger)^\top [K - P] U_\theta^\dagger$, where P is the orthogonal projection defined in Lemma 1. As before, if we do not require $K - K_Q \succeq 0$, then it suffices to set $Q = (U_\theta^\dagger)^\top K U_\theta^\dagger$.

4.3. Locally linear embedding kernels

The basic idea behind locally linear embedding (LLE) ([11]) is straightforward. Given $X = \{x_1, x_2, \dots, x_m\}$, we first construct a weight matrix W such that the i th row of W contains the optimal coefficients, in terms of square loss, required to reconstruct the i th data point x_i as a convex

combination of its k nearest neighbours. The weight matrix is then subsequently used to embed the data points into a lower dimensional space.

Recall that \tilde{X} denotes the points of X stacked up as a matrix. The LLE algorithm can now be summarized as follows:

- (1) compute $\gamma_k(x_i)$, the set of indices of the k nearest neighbours of each data point x_i ;
- (2) subject to $W\mathbf{1} = \mathbf{1}$ and $W_{ij} \neq 0$ only if $j \in \gamma_k(i)$ solve

$$\underset{W}{\text{Minimize}} \operatorname{tr}[(\mathbf{I} - W)\tilde{X}][(\mathbf{I} - W)\tilde{X}]^\top;$$

- (3) find an embedding $F \in \mathbb{R}^{m \times l}$ which solves

$$\underset{F}{\text{Minimize}} \operatorname{tr}[(\mathbf{I} - W)F][(\mathbf{I} - W)F]^\top,$$

$$\text{subject to } FF^\top = \mathbf{I}.$$

It can be shown that the LLE embedding is given by the $m - p, \dots, m - 1$ eigenvectors of $N := (\mathbf{I} - W^\top)(\mathbf{I} - W)$ ([11]). In fact, LLE can be recovered by diagonalizing and using the p smallest eigenvectors of the centered kernel matrix

$$K = (\mathbf{I} - \mathbf{1}\mathbf{1}^\top)(A \cdot \mathbf{I} - N)(\mathbf{I} - \mathbf{1}\mathbf{1}^\top),$$

where A denotes the maximum eigenvalue of N ([2]).

To extend LLE to unseen data, we need to choose a relevant feature map. Let $\alpha \in \mathbb{R}^m$ denote a vector of coefficients such that $\alpha^\top \mathbf{1} = 1$ and $\alpha_i \neq 0$ only if $i \in \gamma_k(x)$. We let \mathcal{P} denote the set of all α 's satisfying the above constraints and define our feature map as

$$\phi : x \rightarrow \underset{\alpha \in \mathcal{P}}{\operatorname{argmin}} \left\| x - \sum_i \alpha_i x_i \right\|_2. \quad (15)$$

Note that on the training data, $\phi(x_i)$ is exactly the i th row of W , i.e., $\Phi = W$. To extend K to unknown data points, we follow essentially the same strategy as before: compute a matrix Q such that the kernel matrix K_Q best approximates K and use this to extend K to unseen data points. As before, the best Q is obtained from Eq. (5) as

$$Q = (W^\dagger)^\top (K - P)W^\dagger,$$

where P is as defined in Eq. (6). Hence, we can use the eigenvectors obtained by diagonalizing K and extrapolate LLE to test data, by virtue of $k_Q(x, x') = \phi(x)^\top Q \phi(x')$.

Note that our method is fundamentally different from the approach taken by [1]. Roughly speaking, they approximate a new kernel function \tilde{k} for every new point x . Using our notation, barring a normalization, they use $\tilde{k}(x, x_i) = \alpha_i$ and $\tilde{k}(x, x) = 0$. They then project this kernel onto the eigen-vectors of the matrix K in order to obtain an embedding for a new point.

5. Experiments

5.1. Experimental setting

To evaluate the performance of our kernel extrapolation technique, we chose the following problem from bioinformatics ([7]). The goal is to classify proteins correctly within the SCOP hierarchy (structural classification of proteins) ([9]). The SCOP hierarchically classifies protein structures into the following categories: class, fold, superfamily and family. More precisely, we were looking at 206 proteins from 9 distinct superfamilies within the triose phosphate isomerase (TIM) beta/alpha-barrel protein fold class. We wanted to predict the superfamily class label of these 206 proteins in two-fold cross-validation (Table 1).

For this task of classifying proteins into the SCOP hierarchy, structural information is most useful, as SCOP is a structure-based hierarchy of classes. Sequence information is of secondary importance for this purpose, as structure is derived from sequence.

In this setting, we tackled the following problem: given a set of proteins, we know all sequences, but only a subset of structures of these proteins. The challenge is to approximate the kernel values for the missing structures and then to combine the sequence and structure information into one joint kernel that leads to good prediction accuracy on our protein classification task. We propose to use our techniques of kernel approximation and joint regularization for this purpose.

5.2. Sequence and structure kernel matrices

For all pairs of protein sequences, Smith–Waterman scores were determined. Analogously, for all pairs of structures, similarity scores were computed using MATRAS ([6]). A normalized similarity measure for inter-residual distances was employed. As both resulting 206×206 matrices are not positive definite, they are turned into positive definite kernel matrices by cutting off non-positive eigenvectors ([10]): If λ_i and v_i , respectively, denote the i th eigenvalue and eigenvector of a non-positive definite symmetric matrix S , then a positive definite kernel matrix is obtained via

$$K = \sum_{i:\lambda_i > 0} \lambda_i v_i v_i^\top.$$

5.3. Kernel approximation and joint regularization

To simulate the situation of not knowing the structures of 10% of our proteins, we randomly removed 10% of the structure matrix' rows (and columns). We repeated the same experiment, removing 25% and 50% of the structure matrix' rows. Each of these experiments was repeated five times to mitigate random effects.

We then approximated the structure kernel matrix from the sequence kernel matrix using kernel approximation.

Table 1

Classification accuracy of structure, sequence and joint regularization kernel on 206 proteins from 9 SCOP TIM superfamilies (str, structure kernel; seq, sequence kernel; best jr, joint regularization kernel with best parameterization; st. dev., standard deviation across classes (K_{str} and K_{seq}) and across repetitions of same experiment ($K_{best jr}$))

Kernel	K_{str}	K_{seq}	$K_{best jr}$	$K_{best jr}$	$K_{best jr}$
% Missing entries	None	None	10	25	50
Accuracy	99.1 ± 0.8	95.9 ± 2.0	98.5 ± 1.3	97.6 ± 2.4	95.5 ± 3.8
Yates' p -value	–	–	0.0005	0.0402	0.7345

Yates' p -value; Yates' p -value from chi-square test which tests hypothesis that best jr kernel has the same accuracy as the sequence kernel.

Table 2

Effect of λ on prediction accuracy, in experiments with 10%, 25% and 50% of missing structural data

Extrapolation λ	0.10	0.25	0.50	0.75	0.90
10% Missing	98.5 ± 1.3	98.0 ± 1.4	97.8 ± 1.4	97.1 ± 1.8	96.3 ± 2.2
20% Missing	97.6 ± 2.4	97.3 ± 2.6	96.4 ± 2.3	95.6 ± 2.2	95.4 ± 2.0
50% Missing	95.5 ± 3.8	95.3 ± 3.6	94.7 ± 3.4	94.3 ± 3.3	93.1 ± 3.0

Afterwards we combined the approximated structure kernel matrix and the sequence kernel matrix via joint regularization.

In detail, the sequence kernel on the subset of the proteins with known structure was used as a common feature map to expand both kernels (see Section 4.1). The normalized structure kernel was approximated by using this feature map (see Section 2.1). Joint regularization is used to combine these two kernels using a previously chosen value of λ (see Section 3.3).

We then performed two-fold cross-validation on the joint regularization kernel matrix. We repeated this for all nine classes, classifying “one class vs rest” using C-support vector machines ([12]). We repeated the experiment for values of $\lambda \in \{0.1, 0.25, 0.5, 0.75, 0.9\}$ (see Table 2). We report results as averages over all classes and all five repetitions in Table 1. Furthermore, as a control experiment, we ran the same classification experiment on all 206 proteins using the sequence kernel and the structure kernel matrix, respectively (Fig. 1).

Classifying the TIM proteins via the complete structure kernel matrix is almost optimal, with a classification accuracy of 99.1%. If 10%, 25% or 50% of the structural information is missing, i.e. if one does not have enough structural data, to classify the proteins via structure, one can resort to sequence information and still reach 95.9% classification accuracy. When 10% or 20% of structural data are missing, classification accuracy can even be significantly increased to 98.5% and 97.6%, respectively, by combining the complete sequence information and the partial structure information via kernel matrix approximation and joint regularization. If 50% of data are missing, our extrapolated and joint kernel does not yield a better result than the sequence kernel matrix alone.

For all levels of missing data, our joint regularization kernel performs progressively better as the value of λ is decreased (see Table 2). Using lower values for λ is

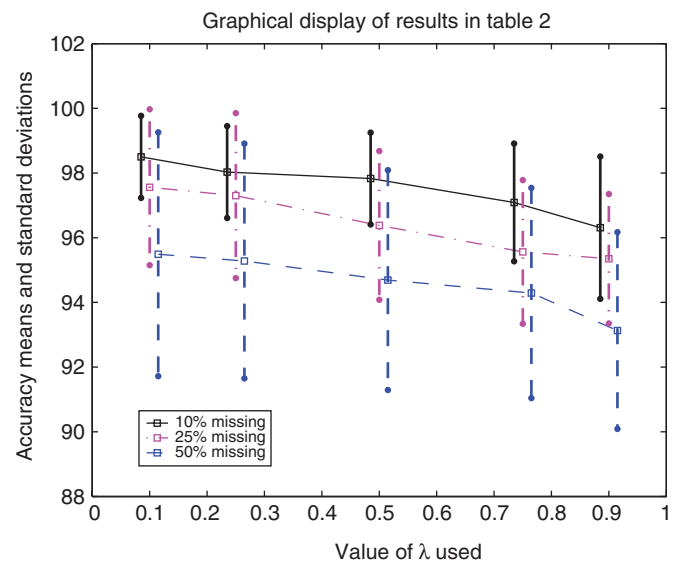


Fig. 1. Graphical display of the results in Table 2. All values of λ are in $\{0.1, 0.25, 0.5, 0.75, 0.9\}$. The means and standard deviation bars are slightly shifted along the x -axis w.r.t. each other in order to enhance visibility. Solid lines correspond to 10% missing entries, dash-dotted ones to 25%, and dashed lines to 50% missing entries.

equivalent to imposing heavy regularization constraints on the structure kernel matrix, while imposing lighter constraints on the sequence kernel. As the structure kernel matrix is only partially known and estimated from the sequence kernel, this fact does not come as a surprise.

6. Discussion and outlook

In this article, we presented a principled method for extending various kernels to unseen data. Our method relies on using an appropriate feature map to approximate the original kernel matrix and using this approximation to extend the kernel to unseen points. We formulated an

optimization problem with semi-definite constraints for this approximation. We showed that an SVD based method can be used to obtain the solution for all Von-Neumann Schatten p -norms.

We also showed how different kernels can be combined in a principled way by using joint regularization. Many well known methods including reduced set methods can be viewed as a special case of our method. We also showed out-of-sample extensions to graph kernels and LLE.

In our experiments, kernel approximation and joint regularization proved capable of combining a fully known protein sequence kernel matrix and an incomplete protein structure kernel matrix, such that the joint kernel achieves higher classification results than the individual kernels. We are currently testing the performance of our methods on larger datasets from various applications including bio-informatics. Future work will be focused on applying our methods to develop out-of-sample extensions to the method proposed by [14].

Acknowledgements

The authors are greatly indebted to Koji Tsuda, Taishin Kin and Tsuyoshi Kato for providing us with the datasets. We thank Gunnar Rätsch, Bernhard Schölkopf, Bob Williamson, and Risi Kondor for helpful discussions. We also thank Stefan Schönauer and Hans-Peter Kriegel for fostering the cooperation between NICTA and the Ludwig-Maximilians-Universität, Munich, Germany. National ICT Australia is funded through the Australian Government's *Backing Australia's Ability* initiative, in part through the Australian Research Council. This work was supported by grants of the ARC and by the IST Program of the European Community, under the Pascal Network of Excellence, IST-2002-506778.

References

- [1] Y. Bengio, J.F. Paiement, P. Vincent, O. Delalleau, N.L. Roux, M. Ouimet, Out of sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering, in: S. Thrun, L. Saul, B. Schölkopf (Eds.), *Advances in Neural Information Processing Systems*, vol. 16, MIT Press, Cambridge, 2003, pp. 177–184.
- [2] J. Ham, D. Lee, S. Mika, B. Schölkopf, Kernel view of dimensionality reduction of manifolds, in: *Proceedings of the 21st International Conference on Machine Learning*, Banff, Canada, 2004.
- [3] R.A. Horn, C.R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, 1985.
- [4] T.S. Jaakkola, D. Haussler, Exploiting generative models in discriminative classifiers, in: M.S. Kearns, S.A. Solla, D.A. Cohn (Eds.), *Advances in Neural Information Processing Systems*, vol. 11, MIT Press, Cambridge, 1999, pp. 487–493.
- [5] R.E. Kass, P.W. Vos, *Geometrical Foundations of Asymptotic Inference*, Wiley series in Probability and Statistics. Wiley Interscience, New York, 1997.
- [6] T. Kawabata, K. Nishikawa, Protein tertiary structure comparison using the Markov transition model of evolution, *Proteins* 41 (2000) 108–122.
- [7] T. Kin, T. Kato, K. Tsuda, Protein classification via kernel matrix completion, in: K. Tsuda, B. Schoelkopf, J.P. Vert (Eds.), *Kernel Methods in Computational Biology*, MIT Press, Cambridge, MA, USA, 2004, pp. 261–274.
- [8] I.R. Kondor, J.D. Lafferty, Diffusion kernels on graphs and other discrete structures, in: *Proceedings of the ICML*, 2002.
- [9] A.G. Murzin, S.E. Brenner, T. Hubbard, C. Chothia, Scop: a structural classification of proteins database for the investigation of sequences and structures, *J. Mol. Biol.* 247 (1995) 536–540.
- [10] V. Roth, J. Laub, J.M. Buhmann, K.-R. Müller, Going metric: denoising pairwise data, in: S. Becker, S. Thrun, K. Obermayer (Eds.), *Advances in Neural Information Processing Systems*, vol. 15, MIT Press, Cambridge, MA, 2003, pp. 817–824.
- [11] S. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, *Science* 290 (2000) 2323–2326.
- [12] B. Schölkopf, A. Smola, *Learning with Kernels*, MIT Press, Cambridge, MA, 2002.
- [13] B. Schölkopf, R. Herbrich, A.J. Smola, A generalized representer theorem, in: *Proceedings of the Annual Conference on Computational Learning Theory*, 2001, pp. 416–426.
- [14] K.Q. Weinberger, F. Sha, L.K. Saul, Learning a kernel matrix for nonlinear dimensionality reduction, in: *Proceedings of the 21st International Conference on Machine Learning*, Banff, Canada, 2004.



S.V.N. Vishwanathan received his masters and Ph.D. in Computer Science from the Indian Institute of Science in 2000 and 2002, respectively. He has been with the Statistical Machine Learning Program, National ICT, Australia, since 2002, first as a researcher and then as a senior researcher. His research interests include machine learning, exponential families, kernel methods, and optimization.



Karsten M. Borgwardt studied computer science and biology from 1999 to 2004 in Munich and Oxford. In 2003, he earned a M.Sc. degree in biology at the University of Oxford, and in 2004, a M.Sc. in computer science at the University of Munich. After finishing his master thesis in computer science at NICTA, Canberra in 2004, he is now a Ph.D. student and scientific assistant at the University of Munich since January 2005.



Omri Guttman holds a Bachelor of Science degree in Physics and Mathematics from the Hebrew University in Jerusalem and an M.A. in Electrical Engineering from the Technion, Israeli Institute of Technology. Since 2003, he is a Ph.D. student at the Statistical Machine Learning (SML) group of the Research School of Information Sciences and Engineering (RSISE)/NICTA at the Australian National University. His research on machine learning currently focuses on learning distributions alphabets using ideas from probabilistic formal

language models.



Alex Smola studied physics in Munich at the University of Technology, Munich, Università degli Studi di Pavia, and AT&T Research in Holmdel. During this time he was at the Maximilianeum in Munich and the Collegio Ghislieri in Pavia. In 1996, he received his Masters degree at the University of Technology, Munich and in 1998 his Doctoral Degree in Computer Science at the University of Technology Berlin. Until 1999, he was a researcher at the IDA Group of the

GMD Institute for Software Engineering and Computer Architecture in Berlin (now part of the Fraunhofer Gesellschaft). After that he joined the Australian National University and worked from 2002 to 2004 as leader of the Machine Learning group. Since 2004, he has been program leader of

the Statistical Machine Learning Program of National ICT Australia. His research interests are nonparametric methods for estimation, such as kernels, inference on discrete objects, structured estimation, optimization and numerical analysis, and learning theory.