Journal of Machine Learning Research 5 (2004) 1127–1141 Submitted 12/03; Revised 7/04; Published 9/04

Knowledge-Based Kernel Approximation

Olvi L. Mangasarian OLVI@CS.WISC.EDU
Jude W. Shavlik SHAVLIK@CS.WISC.EDU
Edward W. Wild WILDT@CS.WISC.EDU
Computer Sciences Department
University of Wisconsin
1210 West Dayton Street
Madison, WI 53706, USA

Editor: John Shawe-Taylor

Abstract

Prior knowledge, in the form of linear inequalities that need to be satisfied over multiple polyhedral sets, is incorporated into a function approximation generated by a linear combination of linear or nonlinear kernels. In addition, the approximation needs to satisfy conventional conditions such as having given exact or inexact function values at certain points. Determining such an approximation leads to a linear programming formulation. By using nonlinear kernels and mapping the prior polyhedral knowledge in the input space to one defined by the kernels, the prior knowledge translates into nonlinear inequalities in the original input space. Through a number of computational examples, including a real world breast cancer prognosis dataset, it is shown that prior knowledge can significantly improve function approximation.

Keywords: function approximation, regression, prior knowledge, support vector machines, linear programming

1. Introduction

Support vector machines (SVMs) play a major role in classification problems (Vapnik, 2000, Cherkassky and Mulier, 1998, Mangasarian, 2000). More recently, prior knowledge has been incorporated into SVM classifiers, both to improve the classification task and to handle problems where conventional data may be few or not available (Schölkopf et al., 1998, Fung et al., 2003b,a). Although SVMs have also been extensively used for regression (Drucker et al., 1997, Smola and Schölkopf, 1998, Evgeniou et al., 2000, Mangasarian and Musicant, 2002), prior knowledge on properties of the function to be approximated has not been incorporated into the SVM function approximation as has been done for an SVM classifier (Fung et al., 2003b,a). In this work, we introduce prior knowledge in the form of linear inequalities to be satisfied by the function on polyhedral regions of the input space for linear kernels, and on similar regions of the feature space for nonlinear kernels. These inequalities, unlike point-wise inequalities or general convex constraints that have already been treated in approximation theory (Mangasarian and Schumaker, 1969, 1971, Micchelli and Utreras, 1988, Deutsch, 2001), are inequalities that need to be satisfied over specific polyhedral sets. Such "prior knowledge" does not seem to have been treated in the extensive approximation theory literature.

We outline the contents of the paper now. In Section 2 we define the prior knowledge formulation for a linear kernel approximation in the input space of the problem which leads to a linear programming formulation in that space. In Section 3 we approximate the function by a linear combination of nonlinear kernel functions and explicitly map the polyhedral prior knowledge in the input space to one defined by the kernel functions. This leads to a linear programming formulation in that space. In Section 4 we demonstrate the utility of our results on a number of synthetic approximation problems as well as a real world breast cancer prognosis dataset where we show that prior knowledge can improve the approximation. Section 5 concludes the paper with a brief summary and some possible extensions and applications of the present work.

We describe our notation now. All vectors will be column vectors unless transposed to a row vector by a prime $'$. The scalar (inner) product of two vectors x and y in the n-dimensional real space R^n will be denoted by $x'y$. For $x \in R^n$, $\|x\|_1$ denotes the 1-norm: $\sum_{i=1}^{n} |x_i|$. The notation $A \in R^{m\times n}$ will signify a real $m \times n$ matrix. For such a matrix, A' will denote the transpose of A, A_i will denote the i-th row of A and $A_{\cdot j}$ the j-th column of A. A vector of ones in a real space of arbitrary dimension will be denoted by e. Thus for $e \in R^m$ and $y \in R^m$ the notation $e'y$ will denote the sum of the components of y. A vector of zeros in a real space of arbitrary dimension will be denoted by 0. For $A \in R^{m\times n}$ and $B \in R^{n\times k}$, a *kernel* $K(A,B)$ maps $R^{m\times n} \times R^{n\times k}$ into $R^{m\times k}$. In particular, if x and y are column vectors in R^n then, $K(x',y)$ is a real number, $K(x',A')$ is a row vector in R^m and $K(A,A')$ is an $m \times m$ matrix. We shall make no assumptions on our kernels other than symmetry, that is $K(x',y)' = K(y',x)$, and in particular we shall not assume or make use of Mercer's positive semidefiniteness condition (Vapnik, 2000, Schölkopf and Smola, 2002). The base of the natural logarithm will be denoted by ε. A frequently used kernel in nonlinear classification is the Gaussian kernel (Vapnik, 2000, Cherkassky and Mulier, 1998, Mangasarian, 2000) whose ijth element, $i = 1\ldots,m,\ j = 1\ldots,k$, is given by: $(K(A,B))_{ij} = \varepsilon^{-\mu\|A_i{}'-B_{\cdot j}\|^2}$, where $A \in R^{m\times n}$, $B \in R^{n\times k}$ and μ is a positive constant. Approximate equality is denoted by $\approx$, while the abbreviation "s.t." stands for "subject to". The symbol $\wedge$ denotes the logical "and" while $\vee$ denotes the logical "or".

2. Prior Knowledge for a Linear Kernel Approximation

We begin with a linear kernel model and show how to introduce prior knowledge into such an approximation. We consider an unknown function f from R^n to R for which approximate or exact function values are given on a dataset of m points in R^n denoted by the matrix $A \in R^{m\times n}$. Thus, corresponding to each point A_i we are given an exact or inexact value of f, denoted by a real number y_i, $i = 1,\ldots,m$. We wish to approximate f by some linear or nonlinear function of the matrix A with unknown linear parameters. We first consider the simple linear approximation

$$f(x) \approx w'x + b, \tag{1}$$

for some unknown weight vector $w \in R^n$ and constant $b \in R$ which is determined by minimizing some error criterion that leads to

$$Aw + be - y \approx 0. \tag{2}$$

If we consider w to be a linear combination of the rows of A, i.e. $w = A'\alpha$, $\alpha \in R^m$, which is similar to the dual representation in a linear support vector machine for the weight w (Mangasarian, 2000, Schölkopf and Smola, 2002), we then have

$$AA'\alpha + be - y \approx 0. \tag{3}$$

This immediately suggests the much more general idea of replacing the linear kernel AA' by some arbitrary nonlinear kernel $K(A,A') : R^{m\times n} \times R^{n\times m} \longrightarrow R^{m\times m}$ that leads to the following approximation, which is nonlinear in A but linear in α:

$$K(A,A')\alpha + be - y \approx 0. \tag{4}$$

We will measure the error in (4) componentwise by a vector $s \in R^m$ defined by

$$-s \leq K(A,A')\alpha + be - y \leq s. \tag{5}$$

We now drive this error down by minimizing the 1-norm of the error s together with the 1-norm of α for complexity reduction or stabilization. This leads to the following constrained optimization problem with positive parameter C that determines the relative weight of exact data fitting to complexity reduction:

$$\begin{array}{rcccl} & & \displaystyle\min_{(\alpha,b,s)} & \|\alpha\|_1 + C\|s\|_1 & \\ \text{s.t.} \quad -s & \leq & K(A,A')\alpha + be - y & \leq & s, \end{array} \tag{6}$$

which can be represented as the following linear program:

$$\begin{array}{rcccl} & & \displaystyle\min_{(\alpha,b,s,a)} & e'a + Ce's & \\ \text{s.t.} \quad -s & \leq & K(A,A')\alpha + be - y & \leq & s, \\ -a & \leq & \alpha & \leq & a. \end{array} \tag{7}$$

We note that the 1-norm formulation employed here leads to a linear programming formulation without regard to whether the kernel $K(A,A')$ is positive semidefinite or not. This would not be the case if we used a kernel-induced norm on α that would lead to a quadratic program. This quadratic program would be more difficult to solve than our linear program especially when it is nonconvex, which would be an NP-hard problem, as is the case when the kernel employed is not positive semidefinite.

We now introduce prior knowledge for a linear kernel as follows. Suppose that it is known that the function f represented by (1) satisfies the following condition. For all points $x \in R^n$, not necessarily in the training set but lying in the nonempty polyhedral set determined by the linear inequalities

$$Bx \leq d, \tag{8}$$

for some $B \in R^{k\times n}$, the function f, and hence its linear approximation $w'x + b$, must dominate a given linear function $h'x + \beta$, for some user-provided $(h,\beta) \in R^{n+1}$. That is, for a *fixed* (w,b) we have the implication

$$Bx \leq d \implies w'x + b \geq h'x + \beta, \tag{9}$$

or equivalently in terms of α, where $w = A'\alpha$:

$$Bx \leq d \implies \alpha' Ax + b \geq h'x + \beta. \tag{10}$$

Thus, the implication (10) needs to be added to the constraints of the linear program (7). To do that we make use of the following equivalence relationship that converts the implication (10) to a set of linear constraints that can be appended to the linear program (7). A similar technique was used in (Fung et al., 2003b, Proposition 2.1) to incorporate prior knowledge into linear classifiers.

Proposition 2.1 Prior Knowledge Equivalence. *Let the set $\{x \mid Bx \le d\}$ be nonempty. Then for a fixed (α, b, h, β), the implication (10) is equivalent to the following system of linear inequalities having a solution $u \in R^k$:*

$$B'u + A'\alpha - h = 0,\ -d'u + b - \beta \ge 0, u \ge 0. \tag{11}$$

Proof The implication (10) is equivalent to the following system having no solution $(x, \zeta) \in R^{n+1}$:

$$Bx - d\zeta \le 0,\ (\alpha' A - h')x + (b - \beta)\zeta < 0,\ -\zeta < 0. \tag{12}$$

By the Motzkin theorem of the alternative (Mangasarian, 1994, Theorem 2.4.2) we have that (12) is equivalent to the following system of inequalities having a solution (u, η, τ):

$$B'u + (A'\alpha - h)\eta = 0,\ -d'u + (b - \beta)\eta - \tau = 0,\ u \ge 0,\ 0 \ne (\eta, \tau) \ge 0. \tag{13}$$

If $\eta = 0$ in (13), then we contradict the nonemptiness of the knowledge set $\{x \mid Bx \le d\}$. Because, for $x \in \{x \mid Bx \le d\}$ and (u, τ) that solve (13) with $\eta = 0$, we obtain the contradiction

$$0 \ge u'(Bx - d) = x'B'u - d'u = -d'u = \tau > 0. \tag{14}$$

Hence $\eta > 0$ in (13). Dividing (13) by η and redefining (u, α, τ) as $(\frac{u}{\eta}, \frac{\alpha}{\eta}, \frac{\tau}{\eta})$ we obtain (11). □

Adding the constraints (11) to the linear programming formulation (7) with a linear kernel $K(A, A') = AA'$, we obtain our desired linear program that incorporates the prior knowledge implication (10) into our approximation problem:

$$\begin{array}{rrcccl} & & & \min\limits_{(\alpha,b,s,a,u \ge 0)} & e'a + Ce's & \\ \text{s.t.} & -s & \le & AA'\alpha + be - y & \le & s, \\ & -a & \le & \alpha & \le & a, \\ & & & A'\alpha + B'u & = & h, \\ & & & -d'u & \ge & \beta - b. \end{array} \tag{15}$$

Note that in this linear programming formulation with a linear kernel approximation, both the approximation $w'x + b = \alpha' Ax + b$ to the unknown function f as well as the prior knowledge are linear in the input data A of the problem. This is somewhat restrictive, and therefore we turn now to our principal concern in this work, which is the incorporation of prior knowledge into a *nonlinear* kernel approximation.

3. Knowledge-Based Nonlinear Kernel Approximation

In this part of the paper we will incorporate prior knowledge by using a nonlinear kernel in *both* the linear programming formulation (7) as well as in the prior knowledge implication (10). We begin with the latter, the linear prior knowledge implication (10). If we again consider x as a linear combination of the rows of A, that is

$$x = A't, \tag{16}$$

then the implication (10) becomes

$$BA't \le d \implies \alpha' AA't + b \ge h'A't + \beta, \tag{17}$$

for a given fixed (α,b). The assumption (16) is not restrictive for the many problems where a sufficiently large number of training data points are available so that any vector in input space can be represented as a linear combination of the training data points.

If we now "kernelize" the various matrix products in the above implication, we have the implication

$$K(B,A')t \leq d \implies \alpha' K(A,A')t + b \geq h'A't + \beta. \tag{18}$$

We note that the two kernels appearing in (18) need not be the same and neither needs to satisfy Mercer's positive semidefiniteness condition. In particular, the first kernel of (18) could be a linear kernel which renders the left side of the implication of (18) the same as that of (17). We note that for a nonlinear kernel, implication (18) is nonlinear in the input space data, but is linear in the implication variable t. We have thus mapped the polyhedral implication (9) into a nonlinear one (18) in the input space data. Assuming for simplicity that the kernel K is symmetric, that is $K(B,A')' = K(A,B')$, it follows directly by Proposition 2.1 that the following equivalence relation holds for implication (18).

Proposition 3.1 Nonlinear Kernel Prior Knowledge Equivalence. *Let the set $\{t \mid K(B,A')t \leq d\}$ be nonempty. Then for a given (α,b,h,β), the implication (18) is equivalent to the following system of linear inequalities having a solution $u \in R^k$:*

$$K(A,B')u + K(A,A')\alpha - Ah = 0,\ -d'u + b - \beta \geq 0, u \geq 0. \tag{19}$$

We now append the constraints (19), which are equivalent to the nonlinear kernel implication (18), to the linear programming formulation (7). This gives the following linear program for approximating a given function with prior knowledge using a nonlinear kernel:

$$\begin{array}{rrcccl} & & & \displaystyle\min_{(\alpha,b,s,a,u\geq 0)} & e'a + Ce's & \\ \text{s.t.} & -s & \leq & K(A,A')\alpha + be - y & \leq & s, \\ & -a & \leq & \alpha & \leq & a, \\ & & & K(A,B')u + K(A,A')\alpha & = & Ah, \\ & & & -d'u & \geq & \beta - b. \end{array} \tag{20}$$

Since we are not certain that the prior knowledge implication (18) is satisfiable, and since we wish to balance the influence of prior knowledge with that of fitting conventional data points, we need to introduce error variables z and ζ associated with the last two constraints of the linear program (20). These error variables are then driven down by a modified objective function as follows:

$$\begin{array}{rrcccl} & & & \displaystyle\min_{(\alpha,b,s,a,z,(u,\zeta)\geq 0)} & e'a + Ce's + \mu_1 e'z + \mu_2\zeta & \\ \text{s.t.} & -s & \leq & K(A,A')\alpha + be - y & \leq & s, \\ & -a & \leq & \alpha & \leq & a, \\ & -z & \leq & K(A,B')u + K(A,A')\alpha - Ah & \leq & z, \\ & & & -d'u + \zeta & \geq & \beta - b, \end{array} \tag{21}$$

where (μ_1,μ_2) are some positive parameters. This is our final linear program for a single prior knowledge implication. If we have more than one such implication, then the last two sets of constraints are repeated for each implication. For the sake of simplicity we omit these details. The values of the parameters C, μ_1, and μ_2 are chosen so as to balance fitting conventional numerical data versus the given prior knowledge. One way to choose these parameters is to set aside a "tuning set" of data points and then choose the parameters so as to give a best fit of the tuning set. We also note that all three kernels appearing in (21) could possibly be distinct kernels from each other and none needs to be positive semidefinite. In fact, the kernel $K(A,B')$ could be the linear kernel AB' which was was actually tried in some of our numerical experiments without a noticeable change from using a Gaussian kernel.

We now turn to our numerical experiments.

4. Numerical Experiments

The focus of this paper is mainly theoretical. However, in order to illustrate the power of the proposed formulation, we tested our algorithm on three synthetic examples and one real world example with and without prior knowledge. Two of the synthetic examples are based on the "sinc" function which has been extensively used for kernel approximation testing (Vapnik et al., 1997, Baudat and Anouar, 2001), while the third synthetic example is a two-dimensional hyperboloid. All our results indicate significant improvement due to prior knowledge. The parameters for the synthetic examples were selected using a combination of exhaustive search and a simple variation on the Nelder-Mead simplex algorithm (Nelder and Mead, 1965) that uses only reflection, with average error as the criterion. The chosen parameter values are given in the captions of relevant figures.

4.1 One-Dimensional Sinc Function

We consider the one-dimensional sinc function

$$f(x) = sinc(x) = \frac{sin\pi x}{\pi x}. \tag{22}$$

Given data for the sinc function includes approximate function values for 52 points on the intervals $-3 \leq x \leq -1.4303$ and $1.4303 \leq x \leq 3$. The endpoints ± 1.4303 are approximate local minima of the sinc function. The given approximate function values for $sinc(x)$ are normally perturbed around the true values, with mean 0 and standard deviation 0.5. In addition, there are also three given values at $x = 0$. One of these values is 1, which is the

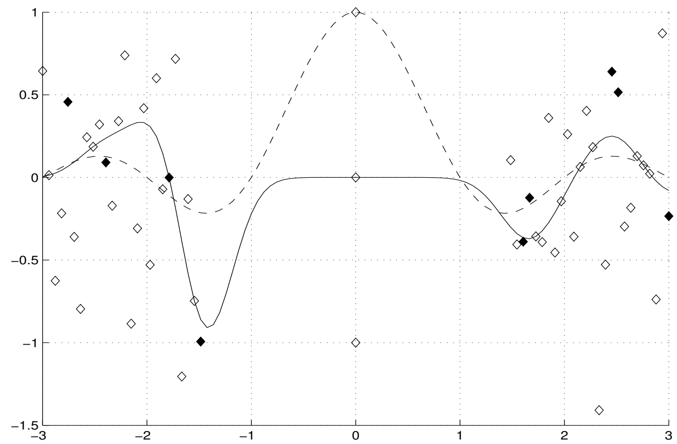

Figure 1: The one-dimensional sinc function $sinc(x) = \frac{sin\pi x}{\pi x}$ (dashed curve) and its Gaussian kernel approximation *without* prior knowledge based on the 55 points shown by diamonds. The nine solid diamonds depict the "support" points used by the nonlinear Gaussian kernel in generating the approximation of $sinc(x)$. That is, they are the rows A_i of A for which $\alpha_i \neq 0$ in the solution of the nonlinear Gaussian kernel approximation of (7) for $f(x)$: $f(x) \approx K(x', A')\alpha + b$. The approximation has an average error of 0.3113 over a grid of 100 points in the interval $[-3, 3]$. Parameter values used: $\mu = 7, C = 5$.

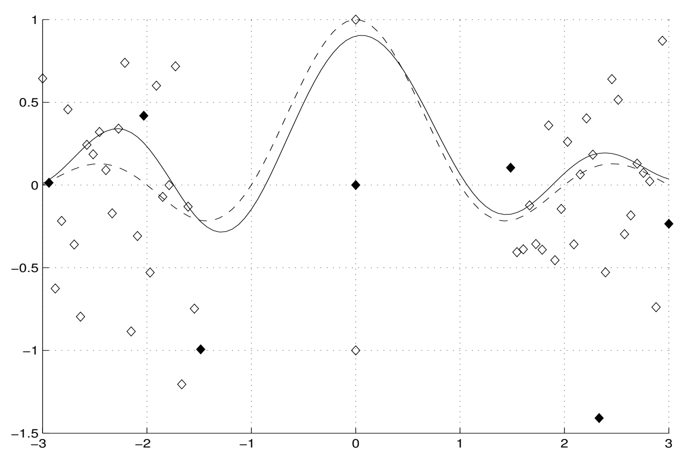

Figure 2: The one-dimensional sinc function $sinc(x) = \frac{sin\pi x}{\pi x}$ (dashed curve) and its Gaussian kernel approximation *with* prior knowledge based on 55 points, shown by diamonds, which are the same as those of Figure 1. The seven solid diamonds depict the "support" points used by the nonlinear Gaussian kernel in generating the approximation of $sinc(x)$. The prior knowledge consists of the implication $-\frac{1}{4} \leq x \leq \frac{1}{4} \Rightarrow f(x) \geq \frac{\sin(\pi/4)}{\pi/4}$, which is implemented by replacing $f(x)$ by its nonlinear kernel approximation (23). The approximation has an average error of 0.0901 over a grid of 100 points in the interval $[-3, 3]$, which is less than $\frac{1}{3.4}$ times the error of Figure 1. Parameter values used: $\mu = 1, C = 13, \mu_1 = 5, \mu_2 = 450$.

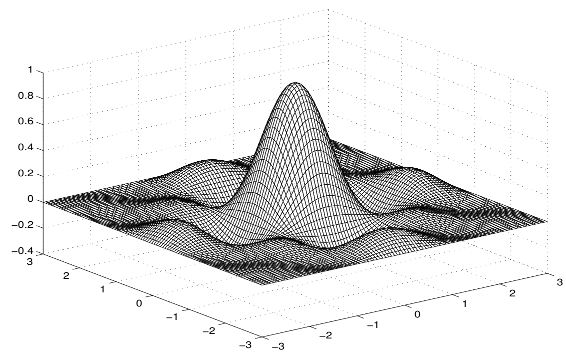

Figure 3: The exact product sinc function $f(x_1,x_2) = \frac{sin\pi x_1}{\pi x_1}\frac{sin\pi x_2}{\pi x_2}$.

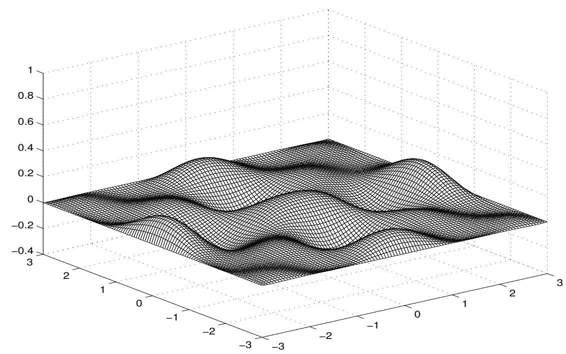

Figure 4: Gaussian kernel approximation of the product sinc function $f(x_1,x_2) = \frac{sin\pi x_1}{\pi x_1}\frac{sin\pi x_2}{\pi x_2}$ based on 211 exact function values plus 2 incorrect function values, but *without* prior knowledge. The approximation has an average error of 0.0501 over a grid of 2500 points in the set $\{[-3,3]\times[-3,3]\}$. Parameter values used: $\mu = 0.2, C = 10^6$.

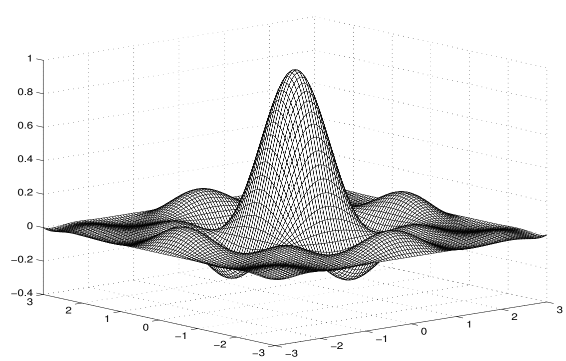

Figure 5: Gaussian kernel approximation of the product sinc function based on the same 213 function values as Figure 4 *plus* prior knowledge consisting of $(x_1,x_2) \in \{[-0.1,0.1]\times[-0.1,0.1]\}\} \Rightarrow f(x_1,x_2) \geq (\frac{sin(\pi/10)}{\pi/10})^2$. The approximation has an average error of 0.0045 over a grid of 2500 points in the set $\{[-3,3]\times[-3,3]\}$, which is less than $\frac{1}{11.1}$ times the error of Figure 4. Parameters are $\mu = 1, C = 16000, \mu_1 = 15000, \mu_2 = 5\cdot 10^6$.

actual limit of the sinc function at 0. The other values at $x = 0$ are 0 and -1 which are intended to be misleading to the approximation.

Figure 1 depicts $sinc(x)$ by a dashed curve and its approximation *without* prior knowledge by a solid curve based on the 55 points shown by diamonds. The nine solid diamonds depict "support" points, that is rows A_i of A for which $\alpha_i \neq 0$ in the solution of the nonlinear Gaussian kernel approximation of (7) for $f(x)$:

$$f(x) \approx K(x', A')\alpha + b. \tag{23}$$

The approximation in Figure 1 has an average error of 0.3113. This error is computed by averaging over a grid of 100 equally spaced points in the interval $[-3, 3]$.

Figure 2 depicts $sinc(x)$ by a dashed curve and its much better approximation *with* prior knowledge by a solid curve based on the 55 points shown, which are the same as those of Figure 1. The seven solid diamond points are "support" points, that is rows A_i of A for which $\alpha_i \neq 0$ in the solution of the nonlinear Gaussian kernel approximation (23) of (21) for $f(x)$. The approximation in Figure 2 has an average error of 0.0901 computed over a grid of 100 equally spaced points on $[-3, 3]$. The prior knowledge used to approximate the one-dimensional sinc function is $-\frac{1}{4} \leq x \leq \frac{1}{4} \Rightarrow f(x) \geq \frac{sin(\pi/4)}{\pi/4}$. The value $\frac{sin(\pi/4)}{\pi/4}$ is the minimum of $sinc(x)$ on the knowledge interval $[-\frac{1}{4}, \frac{1}{4}]$. This prior knowledge is implemented by replacing $f(x)$ by its nonlinear kernel approximation (23) and then using the implication (18) as follows:

$$K(I, A')t \leq \frac{1}{4} \wedge K(-I, A')t \leq \frac{1}{4} \Longrightarrow \alpha' K(A, A')t + b \geq \frac{sin(\pi/4)}{\pi/4}. \tag{24}$$

4.2 Two-Dimensional Sinc Function

Our second example is the two-dimensional $sinc(x)$ function for $x \in R^2$:

$$f(x_1, x_2) = sinc(x_1)sinc(x_2) = \frac{sin\pi x_1}{\pi x_1}\frac{sin\pi x_2}{\pi x_2}. \tag{25}$$

The given data for the two-dimensional sinc function includes 210 points in the region $\{(x_1, x_2) | (-3 \leq x_1 \leq -1.4303 \vee 1.4303 \leq x_1 \leq 3) \wedge (-3 \leq x_2 \leq -1 \vee 1 \leq x_2 \leq 3)\}$. This region excludes the largest bump in the function centered at $(x_1, x_2) = (0, 0)$. The given values are exact function values. There are also three values given at $(x_1, x_2) = (0, 0)$, similar to the previous example with the one dimensional sinc. The first value is the actual limit of the function at $(0, 0)$, which is 1. The other two values are 0 and -1. These last two values are intended to mislead the approximation.

Figure 3 depicts the two-dimensional sinc function of (25). Figure 4 depicts an approximation of $sinc(x_1)sinc(x_2)$ *without* prior knowledge by a surface based on the 213 points described above. The approximation in Figure 4 has an average error of 0.0501. This value is computed by averaging over a grid of 2500 equally spaced points in $\{[-3, 3] \times [-3, 3]\}$.

Figure 5 depicts a much better approximation of $sinc(x_1)sinc(x_2)$ *with* prior knowledge by a surface based on the same 213 points. The approximation in Figure 5 has an average error of 0.0045. This value is computed by averaging over 2500 equally spaced points in

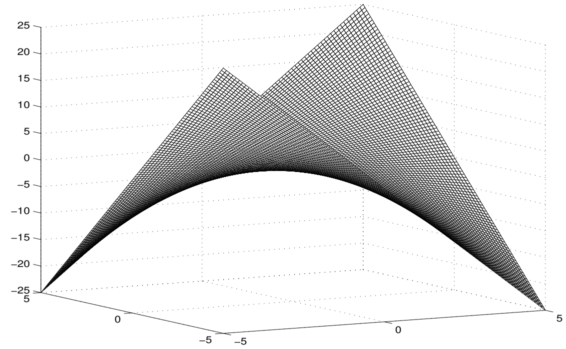

Figure 6: The exact hyperboloid function $f(x_1,x_2) = x_1x_2$.

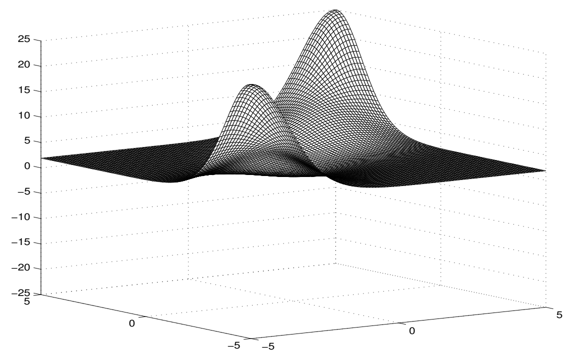

Figure 7: Gaussian kernel approximation of the hyperboloid function $f(x_1,x_2) = x_1x_2$ based on 11 exact function values along the line $x_2 = x_1, x_1 \in \{-5,-4,\ldots,4,5\}$, but *without* prior knowledge. The approximation has an average error of 4.8351 over 2500 points in the set $\{[-5,5]\times[-5,5]\}$. Parameter values used: $\mu = 0.361, C = 145110$.

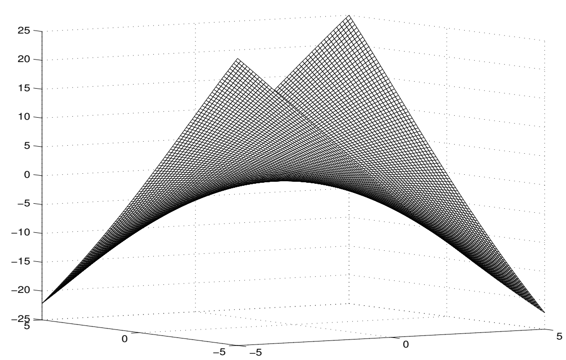

Figure 8: Gaussian kernel approximation of the hyperboloid function $f(x_1,x_2) = x_1x_2$ based on the same 11 function values as of Figure 7 *plus* prior knowledge consisting of the implications (27) and (28). The approximation has an average error of 0.2023 over 2500 points in the set $\{[-5,5]\times[-5,5]\}$, which is less than $\frac{1}{23.9}$ times the error of Figure 7. Parameter values used: $\mu = 0.0052, C = 5356, \mu_1 = 685, \mu_2 = 670613$.

$\{[-3,3]\times[-3,3]\}$. The prior knowledge consists of the implication

$$(x_1,x_2)\in\{[-0.1,0.1]\times[-0.1,0.1]\}\Rightarrow f(x_1,x_2)\geq(\frac{sin(\pi/10)}{\pi/10})^2.$$

The value $(\frac{sin(\pi/10)}{\pi/10})^2$ is equal to the minimum value of $sinc(x_1)sinc(x_2)$ on the knowledge set $\{[-0.1,0.1]\times[-0.1,0.1]\}$. This prior knowledge is implemented by replacing $f(x_1,x_2)$ by its nonlinear kernel approximation (23) and then using the implication (18).

4.3 Two-Dimensional Hyperboloid Function

Our third example is the two-dimensional hyperboloid function

$$f(x_1,x_2)=x_1x_2. \tag{26}$$

For the two-dimensional hyperboloid function, the given data consists of 11 points along the line $x_2=x_1, x_1\in\{-5,-4,\ldots,4,5\}$. The given values at these points are the actual function values.

Figure 6 depicts the two-dimensional hyperboloid function of (26). Figure 7 depicts an approximation of the hyperboloid function, *without* prior knowledge, by a surface based on the 11 points described above. The approximation in Figure 7 has an average error of 4.8351 computed over a grid of 2500 equally spaced points in $\{[-5,5]\times[-5,5]\}$.

Figure 8 depicts a much better approximation of the hyperboloid function by a nonlinear surface based on the same 11 points above *plus* prior knowledge. The approximation in Figure 8 has an average error of 0.2023 computed over a grid of 2500 equally spaced points in $\{[-5,5]\times[-5,5]\}$. The prior knowledge consists of the following two implications:

$$(x_1,x_2)\in\{(x_1,x_2)|-\frac{1}{3}x_1\leq x_2\leq-\frac{2}{3}x_1\}\Rightarrow f(x_1,x_2)\leq 10x_1 \tag{27}$$

and

$$(x_1,x_2)\in\{(x_1,x_2)|-\frac{2}{3}x_1\leq x_2\leq-\frac{1}{3}x_1\}\Rightarrow f(x_1,x_2)\leq 10x_2. \tag{28}$$

These implications are implemented by replacing $f(x_1,x_2)$ by its nonlinear kernel approximation (23) and then using the implication (18). The regions on which the knowledge is given are cones on which x_1x_2 is negative. Since the two implications are analogous, we explain (27) only. This implication is justified on the basis that $x_1x_2\leq 10x_1$ over the knowledge cone $\{(x_1,x_2)|-\frac{1}{3}x_1\leq x_2\leq -\frac{2}{3}x_1\}$ for sufficiently large x_2, that is $x_2\geq 10$. This is intended to capture coarsely the global shape of $f(x_1,x_2)$ and succeeds in generating a more accurate overall approximation of the function.

4.4 Predicting Lymph Node Metastasis

We conclude our numerical results with a potentially useful application of knowledge-based approximation to breast cancer prognosis (Mangasarian et al., 1995, Wolberg et al., 1995, Lee et al., 2001). An important prognostic indicator for breast cancer recurrence is the number of metastasized lymph nodes under a patient's armpit, which could be as many as 30. To determine this number, a patient must undergo optional surgery in addition to the removal of the breast tumor. If the predicted number of metastasized lymph nodes is sufficiently small, then the oncological surgeon may decide not to perform the additional surgery. Thus, it is useful to approximate the number of metastasized lymph nodes as a function of thirty available cytological features and one histological feature. The cytological features are obtained from a fine needle aspirate during the diagnostic procedure while the histological feature is obtained during surgery. Our proposed knowledge-based approximation can be used to improve the determination of such a function, $f : R^{31} \longrightarrow R$, that predicts the number of metastasized lymph nodes. For example, in certain polyhedral regions of R^{31}, past training data indicate the existence of a substantial number of metastasized lymph nodes, whereas certain other regions indicate the unlikely presence of any metastasis. This knowledge can be applied to obtain a hopefully more accurate lymph node function f than that based on numerical function approximation alone.

We have performed preliminary experiments with the Wisconsin Prognostic Breast Cancer (WPBC) data available from (Murphy and Aha, 1992). In our experiments we reduced R^{31} to R^4 and predicted the number of metastasized lymph nodes based on three cytological features: mean cell texture, worst cell smoothness, and worst cell area, as well as the histological feature tumor size. The tumor size is an obvious histological feature to include, while the three other cytological features were the same as those selected for breast cancer diagnosis in (Mangasarian, 2001). Thus, we are approximating a function $f : R^4 \longrightarrow R$. Note that the online version of the WPBC data contains four entries with no lymph node information which were removed for our experiments. After removing these entries, we were left with 194 examples in our dataset.

To simulate the procedure of an expert obtaining prior knowledge from past data we used the following procedure. First we took a random 20% of the dataset to analyze as "past data". Inspecting this past data, we choose the following background knowledge:

$$x_1 \geq 22.4 \wedge x_2 \geq 0.1 \wedge x_3 \geq 1458.9 \wedge x_4 \geq 3.1 \implies f(x_1,x_2,x_3,x_4) \geq 1, \tag{29}$$

where x_1, x_2, x_3, and x_4 denote mean texture, worst smoothness, worst area, and tumor size respectively. This prior knowledge is based on a typical oncological surgeon's advice that larger values of the variables are likely to result in more metastasized lymph nodes. The constants in (29) were chosen by taking the average values of $x_1, \ldots, x_4$ for the entries in the past data with at least one metastasized lymph node.

We used ten-fold cross validation to compare the average absolute error between an approximation without prior knowledge and an approximation with the prior knowledge of Equation (29) on the 80% of the data that was not used as "past data" to generate the constants in (29). Parameters in (21) using a Gaussian kernel were chosen using the Nelder-Mead algorithm on a tuning set taken from the training data for each fold. The average absolute error of the function approximation with no prior knowledge was 3.75 while the average absolute error with prior knowledge was 3.35, a 10.5% reduction. The mean function value of the data used in the ten-fold cross validation experiments is 3.30, so neither approximation is accurate. However, these results indicate that adding prior knowledge does indeed improve the function approximation substantially. Hopefully more sophisticated prior knowledge, based on a more detailed analysis of the data and consultation with domain experts, will further reduce the error.

We close this section with a potential application to a reinforcement learning task (Sutton and Barto, 1998), where the goal is to predict the value of taking an action at a given state. Thus, the domain of the function to be approximated is the Cartesian product of the set of states and the set of actions. In particular, we plan to use the *Keep-Away* subtask of the soccer game developed in (Stone and Sutton, 2001). The state description includes measurements such as distance to each of the opposing players, distance to the soccer ball, distances to the edges of the field, etc. Actions include holding the ball and attempting a pass to a teammate. It has been demonstrated that providing prior knowledge can improve the choice of actions significantly (Kuhlmann et al., 2004, Maclin and Shavlik, 1996). One example of advice (that is, prior knowledge) that has been successfully used in this domain is the simple advice that "if no opponent is within 8 meters, holding the ball is a good idea." In our approach we approximate a value function v as a function of states and actions. Advice can be stated as the following implication, assuming two opponents:

$$d_1 \geq 8 \wedge d_2 \geq 8 \wedge a = h \implies v \geq c, \tag{30}$$

where d_1 denotes the distance to Opponent 1, d_2 the distance to Opponent 2, $a = h$ the action of holding the ball, v the predicted value, and c is some constant. It is hoped that this "advice" can help in generating an improved value function v based on the current description of the state of the soccer game.

5. Conclusion and Outlook

We have presented a knowledge-based formulation of a nonlinear kernel SVM approximation. The approximation is obtained using a linear programming formulation with any nonlinear symmetric kernel and with no positive semidefiniteness (Mercer) condition assumed. The issues associated with sampling the knowledge sets in order to generate function values (that is, a matrix A and a corresponding vector y) in situations where there are no conventional data points constitute an interesting topic for future research. Additional future work includes refinement of prior knowledge and applications to medical problems, computer vision, microarray gene classification, and efficacy of drug treatment, all of which have prior knowledge available.

Acknowledgments

We are grateful to our colleagues Rich Maclin and Dave Musicant for constructive comments. Research described in this UW Data Mining Institute Report 03-05, October 2003, was supported by NSF Grants CCR-0138308, and IRI-9502990, by NLM Grant 1 R01 LM07050-01, by DARPA ISTO Grant HR0011-04-0007, by PHS Grant 5 T15 LM07359-02 and by Microsoft.

References

G. Baudat and F. Anouar. Kernel-based methods and function approximation. In *International Joint Conference on Neural Networks*, pages 1244–1249, Washington, D.C., 2001.

V. Cherkassky and F. Mulier. *Learning from Data - Concepts, Theory and Methods*. John Wiley & Sons, New York, 1998.

F. Deutsch. *Best Approximation in Inner Product Spaces*. Springer-Verlag, Berlin, 2001.

H. Drucker, C. J. C. Burges, L. Kaufman, A. Smola, and V. Vapnik. Support vector regression machines. In M. C. Mozer, M. I. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems -9-*, pages 155–161, Cambridge, MA, 1997. MIT Press.

T. Evgeniou, M. Pontil, and T. Poggio. Regularization networks and support vector machines. In A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 171–203, Cambridge, MA, 2000. MIT Press.

G. Fung, O. L. Mangasarian, and J. Shavlik. Knowledge-based nonlinear kernel classifiers. Technical Report 03-02, Data Mining Institute, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, March 2003a. ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/02-03.ps. *Conference on Learning Theory (COLT 03) and Workshop on Kernel Machines*, Washington D.C., August 24-27, 2003. Proceedings edited by M. Warmuth and B. Schölkopf, Springer Verlag, Berlin, 2003, 102-113.

G. Fung, O. L. Mangasarian, and J. Shavlik. Knowledge-based support vector machine classifiers. In Suzanna Becker, Sebastian Thrun, and Klaus Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 521–528. MIT Press, Cambridge, MA, October 2003b. ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/01-09.ps.

G. Kuhlmann, P. Stone, R. Mooney, and J. Shavlik. Guiding a reinforcement learner with natural language advice: Initial results in robocup soccer. In *Proceedings of the AAAI Workshop on Supervisory Control of Learning and Adaptive Systems*, San Jose, CA, 2004.

Y.-J. Lee, O. L. Mangasarian, and W. H. Wolberg. Survival-time classification of breast cancer patients. Technical Report 01-03, Data Mining Institute, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, March 2001. ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/01-03.ps. *Computational Optimization and Applications* 25, 2003, 151-166.

R. Maclin and J. Shavlik. Creating advice-taking reinforcement learners. *Machine Learning*, 22, 1996.

O. L. Mangasarian. *Nonlinear Programming*. SIAM, Philadelphia, PA, 1994.

O. L. Mangasarian. Generalized support vector machines. In A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 135–146, Cambridge, MA, 2000. MIT Press. ftp://ftp.cs.wisc.edu/math-prog/tech-reports/98-14.ps.

O. L. Mangasarian. Data mining via support vector machines, July 23-27, 2001. http://ftp.cs.wisc.edu/math-prog/talks/ifip3tt.ppt.

O. L. Mangasarian and D. R. Musicant. Large scale kernel regression via linear programming. *Machine Learning*, 46:255–269, 2002. ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/99-02.ps.

O. L. Mangasarian and L. L. Schumaker. Splines via optimal control. In I. J. Schoenberg, editor, *Approximations with Special Emphasis on Splines*, pages 119–156, New York, 1969. Academic Press.

O. L. Mangasarian and L. L. Schumaker. Discrete splines via mathematical programming. *SIAM Journal on Control*, 9:174–183, May 1971.

O. L. Mangasarian, W. N. Street, and W. H. Wolberg. Breast cancer diagnosis and prognosis via linear programming. *Operations Research*, 43(4):570–577, July-August 1995.

C. A. Micchelli and F. I. Utreras. Smoothing and interpolation in a convex subset of a hilbert space. *SIAM Journal of Statistical Computing*, 9:728–746, 1988.

P. M. Murphy and D. W. Aha. UCI machine learning repository, 1992. www.ics.uci.edu/∼mlearn/MLRepository.html.

J. A. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7: 308–313, 1965.

B. Schölkopf, P. Simard, A. Smola, and V. Vapnik. Prior knowledge in support vector kernels. In M. Jordan, M. Kearns, and S. Solla, editors, *Advances in Neural Information Processing Systems 10*, pages 640 – 646, Cambridge, MA, 1998. MIT Press.

B. Schölkopf and A. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

A. Smola and B. Schölkopf. On a kernel-based method for pattern recognition, regression, approximation and operator inversion. *Algorithmica*, 22:211–231, 1998.

P. Stone and R. Sutton. Scaling reinforcement learning toward robocup soccer. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML'01)*, Williams, MA, 2001.

R. Sutton and A. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.

V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, second edition, 2000.

V. N. Vapnik, S. E. Golowich, and A. Smola. Support vector method for function approximation, regression estimation and signal processing. In *Neural Information Processing Systems Volume 9*, pages 281–287, Cambridge, MA, 1997. MIT Press.

W. H. Wolberg, W. N. Street, D. N. Heisey, and O. L. Mangasarian. Computerized breast cancer diagnosis and prognosis from fine-needle aspirates. *Archives of Surgery*, 130:511–516, 1995.