Nonlinear Knowledge-Based Classification

Olvi L. Mangasarian[*] Edward W. Wild[†]

Abstract

Prior knowledge over general nonlinear sets is incorporated into nonlinear kernel classification problems as linear constraints in a linear program. The key tool in this incorporation is a theorem of the alternative for convex functions that converts nonlinear prior knowledge implications into linear inequalities without the need to kernelize these implications. Effectiveness of the proposed formulation is demonstrated on three publicly available classification datasets, including a cancer prognosis dataset. Nonlinear kernel classifiers for these datasets exhibit marked improvements upon the introduction of nonlinear prior knowledge compared to nonlinear kernel classifiers that do not utilize such knowledge.

Keywords: prior knowledge, kernel classification, linear programming, theorem of the alternative

1 INTRODUCTION

Prior knowledge has been used effectively in improving classification both for linear [6] and nonlinear [5] kernel classifiers as well as for nonlinear kernel approximation [19, 14]. In all these applications prior knowledge was converted to linear inequalities that were imposed on a linear program. The linear program generated a linear or nonlinear classifier, or a linear or nonlinear function approximation, all of which were more accurate than the corresponding results that did not utilize prior knowledge. However, whenever a nonlinear kernel was utilized in these applications, kernelization of the prior knowledge was not a transparent procedure that could be easily related to the original sets over which prior knowledge was given. In contrast, in [20] no kernelization of the prior knowledge sets was used in order to incorporate that knowledge into a nonlinear function approximation. We shall use a similar approach here to incorporate prior knowledge into a nonlinear classifier without the need to kernelize the prior knowledge. Furthermore, the region in the input space on which the prior knowledge is given is completely arbitrary in the present work, whereas in all previous classification work prior knowledge had to be restricted to convex polyhedral sets. The present approach is possible through the use of a fundamental theorem of the alternative for convex functions that we describe in Section 2 of the paper, whereas previous work utilized such a theorem for linear inequalities *only*. An interesting, novel approach to knowledge-based support vector machines that modifies the hypothesis space rather than the optimization problem is given in [11]. In another recent approach, prior knowledge is incorporated by adding additional points labeled based on the prior knowledge to the dataset [15].

In Section 3 we describe our linear programming formulation that incorporates nonlinear prior knowledge into a nonlinear kernel, while Section 4 gives numerical examples that show prior knowledge can improve a nonlinear kernel classification significantly. Section 5 concludes the paper.

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

[*] Computer Sciences Department, University of Wisconsin, Madison, WI 53706 and Department of Mathematics, University of California at San Diego, La Jolla, CA 92093.*olvi@cs.wisc.edu.*

[†] Computer Sciences Department, University of Wisconsin, Madison, WI 53706. *wildt@cs.wisc.edu.*

will be denoted by $x'y$. For $x \in R^n$,$\|x\|_1$ denotes the 1-norm: $(\sum_{i=1}^{n}|x_i|)$ while $\|x\|$ denotes the 2-norm: $(\sum_{i=1}^{n}(x_i)^2)^{\frac{1}{2}}$. 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, $K(x', B')$ is a row vector in R^m and $K(A, B')$ is an $m \times m$ matrix. We shall make no assumptions whatsoever 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 definiteness condition [23, 22, 3]. The base of the natural logarithm will be denoted by ε. A frequently used kernel in nonlinear classification is the Gaussian kernel [23, 2, 17] whose ij-th 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. The abbreviation "s.t." stands for "subject to".

2 CONVERSION OF NONLINEAR PRIOR KNOWLEDGE INTO LINEAR CONSTRAINTS

The problem that we wish to impart prior knowledge to consists of classifying a dataset in R^n represented by the m rows of the matrix $A \in R^{m\times n}$ that are labeled as belonging to the class $+1$ or -1 by a corresponding diagonal matrix $D \in R^{m\times m}$ of ±1's. The nonlinear kernel classifier to be generated based on this data as well as prior knowledge will be:

$$K(x', B')u - \gamma = 0, \tag{2.1}$$

where $B \in R^{k\times n}$ and $K(x', B') : R^{1\times n} \times R^{n\times k} \longrightarrow R^{1\times k}$ is an arbitrary kernel function. The variables $u \in R^k$ and $\gamma \in R$ are parameters to be determined by an optimization problem such that the labeled data A satisfy, to the extent possible, the separation condition:

$$D(K(A, B')u - e\gamma) \geq 0. \tag{2.2}$$

This condition (2.2) places the $+1$ and -1 points represented by A on opposite sides of the nonlinear separating surface (2.1). In general the matrix B is set equal to A [17]. However, in reduced support vector machines [12, 9] $B = \bar{A}$, where $\bar{A}$ is a submatrix of A whose rows are a small subset of the rows of A. In fact B can be an arbitrary matrix in $R^{k\times n}$. We now impose prior knowledge on the construction of our classifier function $K(x', B')u - \gamma$ through the following implication:

$$g(x) \leq 0 \implies K(x', B')u - \gamma \geq \alpha, \ \ \forall x \in \Gamma. \tag{2.3}$$

Here, $g(x) : \Gamma \subset R^n \longrightarrow R^k$ is a k-dimensional function defined on a subset Γ of R^n that determines the region in the input space where prior knowledge requires that the classifier function $K(x', B')u - \gamma$ be larger or equal to α, some nonnegative number, in order to classify the points $x \in \{x \mid g(x) \leq 0\}$ as $+1$. Typically α is set to 0 or the margin value 1. A similar implication to (2.3), which we will introduce later in Section 3, classifies points as -1. In previous classification work [5, 19] prior knowledge implications such as (2.3) could not be handled as we shall do here by using Theorem 2.1 below. Instead, in [5, 19], the inequality $g(x) \leq 0$ was kernelized. This led to an inequality not easily related to the original constraint $g(x) \leq 0$. In addition, all previous classification work [5, 19] could handle only linear $g(x)$, which is a significant restriction. The implication (2.3) can be written in the following equivalent logical form:

$$\begin{array}{c} g(x) \leq 0,\ K(x', B')u - \gamma - \alpha < 0, \\ \text{has no solution } x \in \Gamma. \end{array} \tag{2.4}$$

It is precisely implication (2.3), through its equivalent form (2.4), that we shall try to convert to a system of inequalities which is linear in the classification function parameters (u, γ) by means of the following theorem of the alternative for convex functions. The alternatives here are that either the negation of (2.4) holds, or (2.5) below holds, but not both.

THEOREM 2.1. **Prior Knowledge as System of Linear Inequalities** *For a fixed $u \in R^m, \gamma \in R$, the following are equivalent:*

(i) The implication (2.3) or equivalently (2.4) holds.

(ii) There exists $v \in R^k$, $v \geq 0$ such that:

$$K(x', B')u - \gamma - \alpha + v'g(x) \geq 0,\ \forall x \in \Gamma, \tag{2.5}$$

where it is assumed for the implication (i)$\Longrightarrow$(ii) **only**, *that $g(x)$ and $K(x', B')$ are convex on Γ, Γ is a convex subset of R^n, $u \geq 0$ and that $g(x) < 0$ for some $x \in \Gamma$.*

Proof (i)$\Longrightarrow$(ii): This implication follows by starting with (2.4) and utilizing [16, Corollary 4.2.2] together with the fact that the functions $g(x)$ and $K(x', B')u - \gamma - \alpha$ of (2.4) are convex functions of x on the convex set Γ and that $g(x) < 0$ for some $x \in \Gamma$.

(i)$\Longleftarrow$(ii): If (i) did not hold then, by the negation of (2.4), there exists an $x \in \Gamma$ such that $g(x) \leq 0,\ K(x', B')u + \gamma - \alpha < 0$, which would result in the contradiction:

$$0 > K(x', B')u - \gamma - \alpha + v'g(x) \geq 0, \tag{2.6}$$

where the first strict inequality follows from the negation of (2.4), and the last inequality from (2.5). □

We note immediately that in the proposed application in Section 3 of converting prior knowledge to linear inequalities in the parameters (u, γ), all we need is the implication (i)$\Longleftarrow$(ii). This requires no assumptions whatsoever on the functions $g(x)$, $K(x', B')$ or on the parameter u. However, it is important to show that under certain conditions, as we have done above, that (i)$\Longrightarrow$(ii). This ensures that the sufficient condition (ii) for (i) to hold is not a vacuous condition.

We turn now to our linear programming formulation of the knowledge-based nonlinear kernel classification by utilizing Theorem 2.1 above.

3 NONLINEAR PRIOR KNOWLEDGE CLASSIFICATION VIA LINEAR PROGRAMMING

We first formulate the classification problem (2.2) without knowledge in the usual way [17, 22] by allowing a minimal amount of error in data fitting and a minimal number of kernel functions. We measure the error in (2.2) by a nonnegative slack variable $y \in R^m$ as follows:

$$D(K(A, B')u - \gamma e) + y \geq e,\ y \geq 0, \tag{3.7}$$

where, as usual, a margin of width $\frac{2}{\|u\|}$ between the $+1$ and -1 classes in the u space is introduced. We now drive down the slack variable y by minimizing its 1-norm together with the 1-norm of u for kernel function and complexity reduction. This leads to the following constrained optimization problem with positive parameter ν that determines the relative weight of data fitting to complexity reduction:

$$\begin{array}{rrcl} & \min\limits_{(u,\gamma,y)} & \nu\|y\|_1 + \|u\|_1 & \\ \text{s.t.} & D(K(A,B')u - \gamma e) + y & \geq & e, \\ & y & \geq & 0. \end{array} \tag{3.8}$$

This optimization problem is equivalent to the following linear program:

$$\begin{array}{rrcl} & & \min\limits_{(u,\gamma,y,s)} \ \nu e'y + e's & \\ \text{s.t.} & & D(K(A,B')u - \gamma e) + y & \geq & e, \\ -s & \leq & u & \leq & s, \\ & & y & \geq & 0. \end{array} \tag{3.9}$$

We now introduce prior knowledge contained in the implication (2.3) by making use of Theorem 2.1 and converting it to the linear constraints (2.5) and incorporating it into the linear program (3.9) as follows:

$$\begin{array}{rl} \min\limits_{(u,\gamma,y,s,v,z_1,\ldots,z_\ell)} & \nu e'y + e's + \sigma\sum\limits_{i=1}^{\ell} z_i \\ \text{s.t.} & D(K(A,B')u - \gamma e) + y \geq e, \\ & -s \leq u \leq s, \\ & y \geq 0, \\ & K({x^i}', B')u - \gamma - \alpha + v'g(x^i) + z_i \geq 0, \\ & v \geq 0,\ z_i \geq 0,\ i = 1, \ldots, \ell. \end{array} \tag{3.10}$$

We note that we have discretized the variable $x \in \Gamma$ in the next to the last constraint above to a mesh of points $x^1, x^2, \ldots, x^\ell$ in order to convert a semi-infinite linear program [7], that is a linear program with an infinite number of constraints, to a linear program with a finite number of constraints. We have also added nonnegative slack error variables z_i, $i = 1, \ldots, \ell$, to allow small deviations in satisfying the prior knowledge. The sum of these nonnegative slack variables $z_1, z_2, \ldots, z_\ell$ for the prior knowledge inequalities are minimized with weight $\sigma > 0$ in the objective function in order to drive them to zero to the extent possible.

To complete the prior knowledge formulation we include prior knowledge that implies that points in a given set are in the class -1. Thus instead, of the implication (2.3) we have the implication:

$$h(x) \leq 0 \implies K(x', B')u - \gamma \leq -\alpha, \ \ \forall x \in \Lambda. \tag{3.11}$$

Here, $h(x) : \Lambda \subset R^n \longrightarrow R^r$ is an r-dimensional function defined on a subset Λ of R^n that determines the region in the input space where prior knowledge requires that the classifier $K(x', B') - \gamma$ be less or equal to $-\alpha$ in order to classify the points $x \in \{x \mid h(x) \leq 0\}$ as belonging to the class -1. This implication is equivalent to:

$$\begin{array}{c} h(x) \leq 0,\ -K(x', B')u + \gamma - \alpha < 0, \\ \text{has no solution } x \in \Lambda. \end{array} \tag{3.12}$$

Upon invoking Theorem 2.1, the statement (3.12) is implied by the existence of a nonnegative $p \in R^r$ such that:

$$-K(x', B')u + \gamma - \alpha + p'h(x) \geq 0,\ \forall x \in \Lambda. \tag{3.13}$$

Discretizing this constraint over $x \in \Lambda$ and incorporating it into the linear programming formulation (3.10) results in our final linear program that incorporates prior knowledge for both classes $+1$ and -1 as follows:

$$\begin{array}{rl} \min\limits_{(u,\gamma,y,s,v,p,z_1,\ldots,z_\ell,q_1,\ldots,q_t)} & \nu e'y + e's + \sigma(\sum\limits_{i=1}^{\ell} z_i + \sum\limits_{j=1}^{t} q_j) \\ \text{s.t.} & D(K(A,B')u - \gamma e) + y \geq e, \\ & -s \leq u \leq s, \\ & y \geq 0, \\ & K(x^{i'},B')u - \gamma - \alpha + v'g(x^i) + z_i \geq 0, \\ & v \geq 0,\ z_i \geq 0,\ i = 1,\ldots,\ell, \\ & -K(x^{j'},B')u + \gamma - \alpha + p'h(x^j) + q_j \geq 0, \\ & p \geq 0,\ q_j \geq 0,\ j = 1,\ldots,t. \end{array} \tag{3.14}$$

We turn now to computational results and test examples of the proposed approach for incorporating nonlinear knowledge into kernel classification problems.

4 COMPUTATIONAL RESULTS

To illustrate the effectiveness of our proposed formulation, we report results on three publicly available datasets: The Checkerboard dataset [8], the Spiral dataset [24], and the Wisconsin Prognostic Breast Cancer (WPBC) dataset [21]. It is important to point out that the present formulation is very different in nature from that presented by Fung et al. in [5]. Our primary concern here is to incorporate prior knowledge in an end explicit and transparent manner without having to kernelize it as was done in [5]. In particular, we are able to directly incorporate general implications involving nonlinear functions as linear inequalities in a linear program by utilizing Theorem 2.1. Although the given prior knowledge is strong, the examples illustrate the simplicity and effectiveness of our approach of incorporating it into a nonlinear support vector machine classifier.

4.1 CHECKERBOARD PROBLEM

Our first example is based on the frequently utilized checkerboard dataset [8, 10, 18, 12, 5]. This synthetic dataset contains two-dimensional points in $[-1,1] \times [-1,1]$ labeled so that they form a checkerboard. For this example, we use a dataset consisting of only the sixteen points at the center of each square in the checkerboard to generate a classifier without knowledge. We set the rows of both matrices A and B of (3.14) equal to the coordinates of the sixteen points, which are the standard values. Figure 1 shows a classifier trained on these sixteen points *without* any additional prior knowledge.

Figure 2 shows a much more accurate classifier trained on the same sixteen points as used in Figure 1, *plus* prior knowledge representing only the leftmost two squares in the bottom row of the checkerboard. This knowledge was imposed via the following implications:

$$\begin{array}{c} -1 \leq x_1 \leq -0.5 \wedge -1 \leq x_2 \leq -0.5 \Longrightarrow f(x_1,x_2) \geq 0, \\ -0.5 \leq x_1 \leq 0 \wedge -1 \leq x_2 \leq -0.5 \Longrightarrow f(x_1,x_2) \leq 0. \end{array} \tag{4.15}$$

The implication on the first line was imposed at 100 uniformly spaced points in $[-1,-0.5] \times [-1,-0.5]$, and the implication on the second line were imposed at 100 uniformly spaced points in $[-0.5,0] \times [-1,-0.5]$. No prior knowledge was given for the remaining squares of the checkerboard. We note that this knowledge is very similar to that used in [5], although our classifier is more accurate here. This example demonstrates that knowledge of the form used in [5] can be easily applied with our proposed approach without kernelizing the prior knowledge.

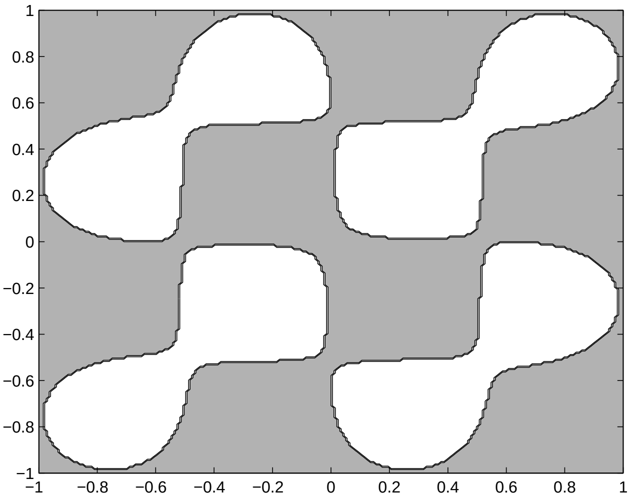

Figure 1: **A classifier for the checkerboard dataset trained using only the sixteen points at the center of each square *without* prior knowledge. The white regions denote areas where the classifier returns a value greater than zero, and the gray regions denote areas where the classifier returns a value less than zero. A uniform grid consisting of $40,000$ points was used to create the plot utilizing the obtained classifier.**

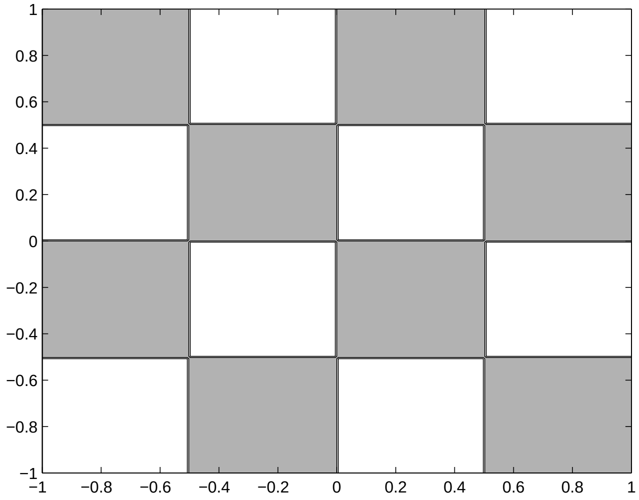

Figure 2: **A classifier for the checkerboard dataset trained using the sixteen points at the center of each square *with* prior knowledge representing the two leftmost squares in the bottom row given in (4.15). The white regions denote areas where the classifier returns a value greater than zero, and the gray regions denote areas where the classifier returns a value less than zero. A uniform grid consisting of $40,000$ points was used to create the plot utilizing the knowledge-based classifier.**

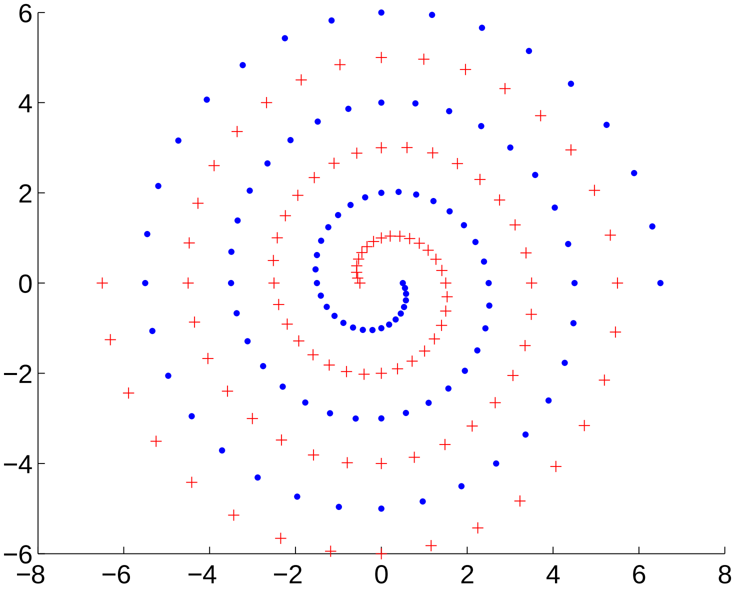

Figure 3: **The spiral dataset. The filled circles represent points with label $+1$ while the crosses represent points with label -1.**

4.2 SPIRAL PROBLEM

The spiral dataset [24, 4] is used for our second synthetic example. This dataset consists of the two concentric spirals shown in Figure 3.

In order to illustrate the effectiveness of our approach on this dataset, we randomly chose to provide labels for only a subset of the points in Figure 3. For this dataset, the matrix B of Equation (2.1) consists of all the points in the dataset. Figure 4 shows a classifier trained using ten-fold cross validation on the points with given labels and no prior knowledge. The points for which labels were given during training are circled. Note that the classifier incorrectly classifies many of the points with label $+1$ for which no label was provided during training.

Figure 5 shows a much more accurate classifier trained on the same labeled points *plus* prior knowledge based on the construction of the spiral dataset. This knowledge can be represented as follows:

$$\begin{gathered} g(x) \leq 0 \Longrightarrow f(x) \geq \quad 1 \\ g(x) \geq 0 \Longrightarrow f(x) \leq \ -1 \quad \text{where} \\ g(x) = \left\| \begin{pmatrix} \|x\| \cos\left(\frac{\pi(6.5-x)104}{(16)(6.5)}\right) \\ \|x\| \sin\left(\frac{\pi(6.5-x)104}{(16)(6.5)}\right) \end{pmatrix} - x \right\| - \left\| \begin{pmatrix} \|x\| \cos\left(\frac{\pi(6.5-x)104}{(16)(6.5)} + \pi\right) \\ \|x\| \sin\left(\frac{\pi(6.5-x)104}{(16)(6.5)} + \pi\right) \end{pmatrix} - x \right\|. \end{gathered} \tag{4.16}$$

Though complicated in appearance, the derivation of this expression is actually quite straightforward given the source code that generates the spiral dataset [24]. To impose the prior knowledge, each implication was imposed at the points defined by the rows of the matrix B for which the left-hand side of the implication held, as well as two additional points near that point. Recall that for this dataset, B contains every point in the dataset as shown in Figure 3. For example, the first implication was imposed on the points x and $x \pm \binom{0.2}{0.2}$ where x is a row of the matrix B and $g(x) \leq 0$.

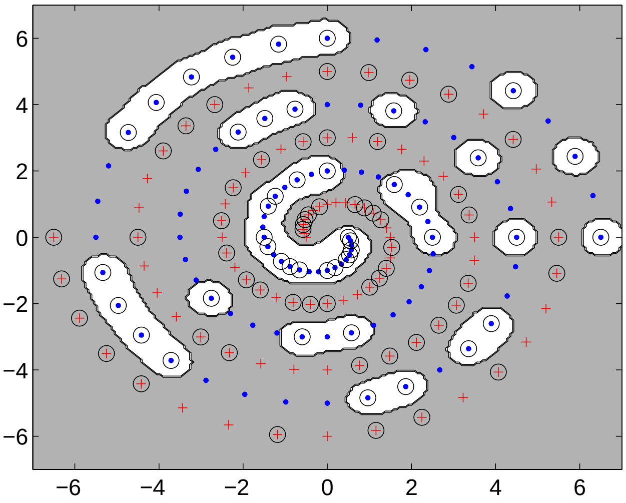

Figure 4: **A classifier for the spiral dataset trained using only a subset of given labels *without* prior knowledge. The circled points represent the labeled points constituting the dataset represented by the matrix A of Equation (3.14). The matrix B consists of all points shown. The white regions denote areas where the classifier returns a value greater than zero, thus classifying the points therein as $+1$, i.e. white ⇒ dots. Gray regions denote areas where the classifier returns a value less than zero, thus classifying points as -1, i.e. gray ⇒ crosses. Note the many points (dots) incorrectly classified as -1.**

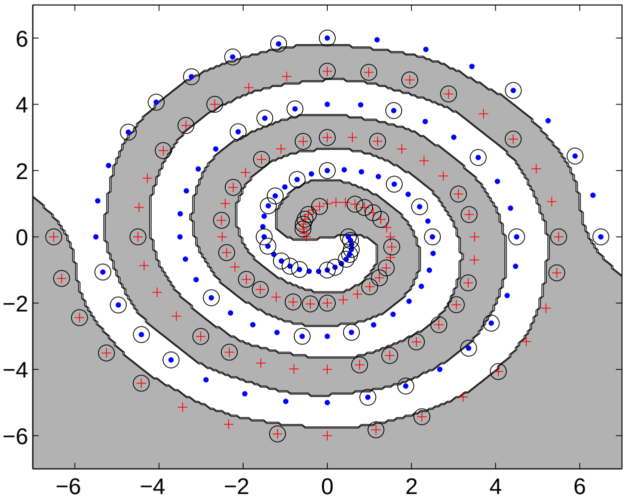

Figure 5: **A classifier for the spiral dataset trained using only the subset represented by circled points with given labels *plus* the prior knowledge given in (4.16). The white regions denote areas where the classifier returns a value greater than zero and should contain only dots. The gray regions denote areas where the classifier returns a value less than zero and should contain only crosses. Note that there are no misclassified points.**

4.3 PREDICTING BREAST CANCER SURVIVAL TIME

We conclude our experimental results with a potentially useful application of the Wisconsin Prognostic Breast Cancer (WPBC) dataset [21, 13]. This dataset contains thirty cytological features obtained from a fine needle aspirate and two histological features, tumor size and the number of metastasized lymph nodes, obtained during surgery for breast cancer patients. The dataset also contains the amount of time before each patient experienced a recurrence of the cancer, if any. Here, we shall consider the task of predicting whether a patient will remain cancer free for at least 24 months. Past experience with this dataset has shown that an accurate classifier for this task is difficult to obtain. In this dataset, 81.9% of patients are cancer free after 24 months. To our knowledge, the best result on this dataset is 86.3% correctness obtained by Bennett in [1]. It is possible that incorporating expert information about this task is necessary to obtain higher accuracy on this dataset. We demonstrate that with sufficient prior knowledge, our approach can achieve 91.0% correctness.

To obtain prior knowledge for this dataset, we plotted the number of metastasized lymph nodes against the tumor size, along with the class label, for each patient. We then simulated an oncological surgeon's advice by selecting regions containing patients who experienced a recurrence withing 24 months. In a typical machine learning task, not all of the class labels would be available. However, our purpose here is to demonstrate that if an expert is able to provide useful prior knowledge, our approach can effectively apply that knowledge to learn a more accurate classifier. We leave studies on this dataset in which an expert provides knowledge without all of the labels available to future work. In such studies, the expert would be given information regarding the class of only data points in a training set that is a subset of all the data, and then give advice on the class of points in the entire dataset. The prior knowledge we constructed for this dataset is depicted in Figure 6 and consists of the following three implications:

$$\begin{aligned}
\left\|\binom{(5.5)x_1}{x_2} - \binom{(5.5)7}{9}\right\| + \left\|\binom{(5.5)x_1}{x_2} - \binom{(5.5)4.5}{27}\right\| - 23.0509 \quad & \leq 0 \Longrightarrow f(x) \geq 1 \\
\begin{pmatrix} -x_2 + 5.7143x_1 - 5.75 \\ x_2 - 2.8571x_1 - 4.25 \\ -x_2 + 6.75 \end{pmatrix} \quad & \leq 0 \Longrightarrow f(x) \geq 1 \\
\tfrac{1}{2}(x_1 - 3.35)^2 + (x_2 - 4)^2 - 1 \quad & \leq 0 \Longrightarrow f(x) \geq 1.
\end{aligned} \tag{4.17}$$

The class +1 represents patients who experienced a recurrence in less than 24 months. Here, x_1 is the tumor size, and x_2 is the number of metastasized lymph nodes. Each implication is enforced at the points in the dataset for which the left-hand side of the implication is true. These regions are shown in Figure 6. The first implication corresponds to the region closest to the upper right-hand corner. The triangular region corresponds to the second implication, and the small elliptical region closest to the x_1 axis corresponds to the third implication. Although these implications seem complicated, it would not be difficult to construct a more intuitive interface similar to standard graphics programs to allow a user to create arbitrary regions. Applying these regions with our approach would be straightforward.

In order to evaluate our proposed approach, we compared the misclassification rates of two classifiers on this dataset. One classifier is learned without prior knowledge, while the second classifier is learned using the prior knowledge given in (4.17). For both cases the rows of the matrices A and B of (3.14) were set to the usual values, that is to the coordinates of the points of the training set. The misclassification rates are computed using leave-one-out cross validation. For each fold, the parameter ν and the kernel parameter μ were chosen from the set $\{2^i | i \in \{-7, \ldots, 7\}\}$ by using ten-fold cross validation on the training set of the fold. In the classifier with prior knowledge, the parameter σ was set to 10^6, which corresponds to very strict adherence to the prior knowledge. The results are summarized in Table 1. The reduction in misclassification rate indicates that our approach can use appropriate prior knowledge to obtain a classifier on this difficult dataset with 50% improvement.

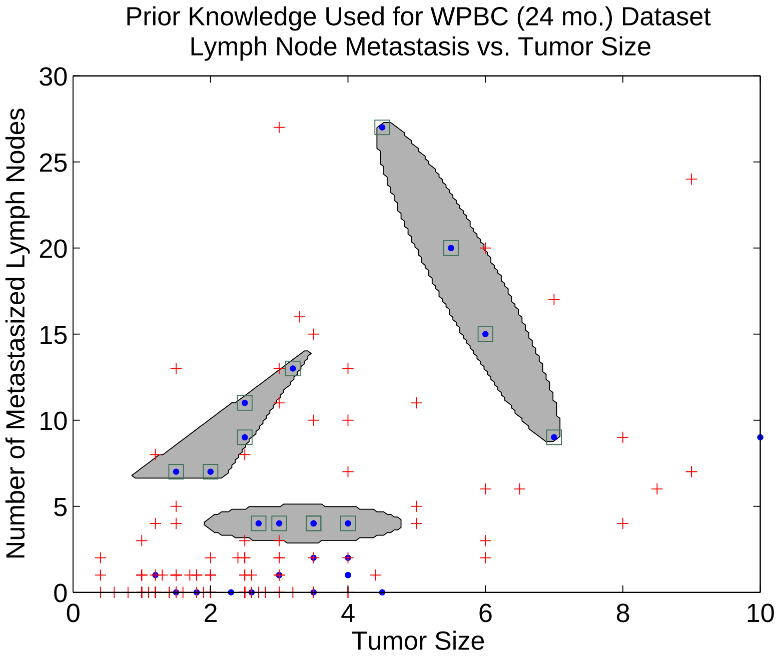

Figure 6: **Number of metastasized lymph nodes versus tumor size for the WPBC (24 mo.) dataset. The solid dots represent patients who experienced a recurrence within 24 months of surgery, while the crosses represent the cancer free patients. The shaded regions which correspond to the areas in which the left-hand side of one of the three implications in Equation (4.17) is true simulate an oncological surgeon's prior knowledge regarding patients that are likely to have a recurrence. Prior knowledge was enforced at the points enclosed in squares.**

Classifier	Misclassification Rate
Without knowledge	0.1806
With knowledge	**0.0903**
Improvement due to knowledge	50.0%

Table 1: **Leave-one-out misclassification rate of classifiers with and without knowledge on the WPBC (24 mo.) dataset. Best result is in bold.**

5 CONCLUSION AND OUTLOOK

We have proposed a computationally effective framework for handling general nonlinear prior knowledge in kernel classification problems. We have reduced such prior knowledge to easily implemented linear constraints in a linear programming formulation. We have demonstrated the effectiveness of our approach on two synthetic problems and an important real world problem arising in breast cancer prognosis. Possible future extensions are to even more general prior knowledge, such as that where the right hand side of the implications (2.3) and (3.11) are replaced by very general nonlinear inequalities involving the classification function (2.1). Another important avenue of future work is to construct an interface which allows users to easily specify arbitrary regions to be used as prior knowledge.

Acknowledgments The research described in this Data Mining Institute Report 06-04, August 2006, was supported by National Science Foundation Grants CCR-0138308 and IIS-0511905, the Microsoft Corporation and ExxonMobil.

References

[1] K. P. Bennett. Decision tree construction via linear programming. In M. Evans, editor, *Proceedings of the 4th Midwest Artificial Intelligence and Cognitive Science Society Conference*, pages 97–101, Utica, Illinois, 1992.

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

[3] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, Cambridge, 2000.

[4] G. Fung and O. L. Mangasarian. Proximal support vector machine classifiers. In F. Provost and R. Srikant, editors, *Proceedings KDD-2001: Knowledge Discovery and Data Mining, August 26-29, 2001, San Francisco, CA*, pages 77–86, New York, 2001. Association for Computing Machinery. ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/01-02.ps.

[5] 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 2003. ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/03-02.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.

[6] 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 2003. ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/01-09.ps.

[7] M. A. Goberna and M. A. López. *Linear Semi-Infinite Optimization*. John Wiley, New York, 1998.

[8] T. K. Ho and E. M. Kleinberg. Checkerboard dataset, 1996. http://www.cs.wisc.edu/math-prog/mpml.html.

[9] S.Y. Huang and Y.-J. Lee. Theoretical study on reduced support vector machines. Technical report, National Taiwan University of Science and Technology, Taipei, Taiwan, 2004. yuh-jye@mail.ntust.edu.tw.

[10] L. Kaufman. Solving the quadratic programming problem arising in support vector classification. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, pages 147–167. MIT Press, 1999.

[11] Q. V. Le, A. J. Smola, and T. Gärtner. Simpler knowledge-based support vector machines. In *Proceedings of the 23^{rd} International Conference on Machine Learning, Pittsburgh, PA, 2006*, 2006. http://www.icml2006.org/icml2006/technical/accepted.html.

[12] Y.-J. Lee and O. L. Mangasarian. RSVM: Reduced support vector machines. In *Proceedings of the First SIAM International Conference on Data Mining, Chicago, April 5-7, 2001, CD-ROM*, 2001. ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/00-07.ps.

[13] Y.-J. Lee, O. L. Mangasarian, and W. H. Wolberg. Breast cancer survival and chemotherapy: a support vector machine analysis. Technical Report 99-10, Data Mining Institute, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, December 1999. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, American Mathematical Society, Volume 55, 2000, 1-10. ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/99-10.ps.

[14] R. Maclin, J. Shavlik, L. Torrey, T. Walker, and E. Wild. Giving advice about preferred actions to reinforcement learners via knowledge-based kernel regression. In *Proceedings of the 20th National Conference on Artificial Intelligence*, 2005.

[15] R. Maclin, J. Shavlik, T. Walker, and L. Torrey. A simple and effective method for incorporating advice into kernel methods. In *Proceedings of the 21st National Conference on Artificial Intelligence*, 2006.

[16] O. L. Mangasarian. *Nonlinear Programming*. McGraw–Hill, New York, 1969. Reprint: SIAM Classic in Applied Mathematics 10, 1994, Philadelphia.

[17] 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.

[18] O. L. Mangasarian and D. R. Musicant. Lagrangian support vector machines. *Journal of Machine Learning Research*, 1:161–177, 2001. ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/00-06.ps.

[19] O. L. Mangasarian, J. W. Shavlik, and E. W. Wild. Knowledge-based kernel approximation. *Journal of Machine Learning Research*, 5:1127–1141, 2004. ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/03-05.ps.

[20] O. L. Mangasarian and E. W. Wild. Nonlinear knowledge in kernel approximation. Technical Report 05-05, Data Mining Institute, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, October 2005. ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/05-05.ps.

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

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

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

[24] A. Wieland. Twin spiral dataset. http://www-cgi.cs.cmu.edu/afs/cs.cmu.edu/project/ai-repository/ai/areas/neural/bench/cmu/0.html.