

Proximal Knowledge-Based Classification

O. L. Mangasarian* E. W. Wild† G. M. Fung ‡

Abstract

Prior knowledge over general nonlinear sets is incorporated into nonlinear kernel proximal classification problems as linear equalities. The key tool in this incorporation is the conversion of general nonlinear prior knowledge implications into linear equalities in the classification variables without the need to kernelize these implications. These equalities are then included into a proximal nonlinear kernel classification formulation [6] that is solvable as a system of linear equations. Effectiveness of the proposed formulation is demonstrated on a number of publicly available classification datasets. 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, proximal support vector machines

1 INTRODUCTION

Prior knowledge has been used effectively in improving classification both for linear [10] and nonlinear [9] kernel classifiers as well as for nonlinear kernel approximation [23, 18]. 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, 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. However, in [24] no kernelization of the prior knowledge sets was used in order to incorporate that knowledge into a nonlinear kernel approximation and in [25] into nonlinear kernel classification. In all these approaches a linear program needs to be solved which may be costly time-wise and requires the availability of an efficient linear programming package. In contrast, proximal nonlinear classifiers [6, 8, 7] require the solution of a considerably simpler system of linear equations which we shall employ here together with the imposition of nonlinear prior knowledge as linear equalities. The fundamental tool in converting prior knowledge implications to linear equalities is motivated by a theorem of the alternative for convex functions [25, Theorem 2.1] which is employed in Section 2 in a very simple manner as described in Proposition 2.1. Another interesting approach to knowledge-based support vector machines modifies the hypothesis space rather than the optimization problem is given in [16]. In another recent work, prior knowledge is incorporated by adding additional points labeled based on the prior knowledge to the dataset [19].

In Section 3 we describe our nonlinear kernel classification formulation that incorporates nonlinear prior knowledge as linear equalities in a proximal support vector machine formulation which leads

*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*.

‡Siemens Medical Solutions, Inc., 51 Valley Stream Parkway, Malvern, PA 19355. *glenn.fung@siemens.com*.

to a system of linear equations with a symmetric positive definite matrix. Section 4 gives numerical examples using publicly available datasets which show that 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 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:

$(\sum_{i=1}^n (x_i)^2)^{\frac{1}{2}}$ and x_+ denotes the vector $\max\{x, 0\}$. 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_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 [30, 28, 4]. The base of the natural logarithm will be denoted by ε . A frequently used kernel in nonlinear classification is the Gaussian kernel [30, 2, 21] whose ij -th element, $i = 1, \dots, m$, $j = 1, \dots, k$, is given by: $(K(A, B))_{ij} = \varepsilon^{-\mu \|A_i' - B_j\|^2}$, where $A \in R^{m \times n}$, $B \in R^{n \times k}$ and μ is a positive constant. We will use this kernel exclusively in this paper.

2 CONVERSION OF NONLINEAR PRIOR KNOWLEDGE INTO A LINEAR EQUALITY

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:

$$K(x', B')u - \gamma = 0, \quad (2.1)$$

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 variables whose values will be determined by an optimization problem such that the labeled data A satisfy, to the extent possible, the separation condition:

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

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 [21]. However, in reduced support vector machines [17, 13] $B = \bar{A}$, 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 $K(x', B')u - \gamma$ to ensure that a certain set of points lies on the $+1$ side of the classifier (2.1). We achieve this through the following implication:

$$g(x) \leq 0 \implies K(x', B')u - \gamma = 1, \quad \forall x \in \Gamma_1. \quad (2.3)$$

Here, $g(x) : \Gamma_1 \subset R^n \longrightarrow R^r$ is an r -dimensional function defined on a subset Γ_1 of R^n that determines the region in the input space where prior knowledge requires that for a given point x in that region,

the classifier function $K(x', B')u - \gamma$ return a value of +1 in order for the classifier to be proximal to the surface $K(x', B')u = \gamma + 1$. This would classify a point $x \in \{x \mid g(x) \leq 0\}$ as being in class +1. A similar implication to (2.3), which we will introduce later in Section 3, classifies points as being in class -1. The implication (2.3) can be written in the following equivalent form:

$$g(x)_+ = 0 \implies K(x', B')u - \gamma = 1, \quad \forall x \in \Gamma_1, \quad (2.4)$$

where, as defined in the Introduction, $g(x)_+ = \max\{g(x), 0\}$. The use of $g(x)_+ = 0$ in (2.4) in place of $g(x) \leq 0$ is a key observation which allows us to have a multiplier in Proposition 2.1 which is not required to be nonnegative. Hence, we can utilize a proximal point support vector machine formulation. We immediately state a simple proposition, motivated by the fundamental theorem of the alternative for convex functions [25, Theorem 2.1] which ensures the satisfaction of the implication (2.4) and hence the implication (2.3) once a certain linear equality is satisfied.

PROPOSITION 2.1. Prior Knowledge as a Linear Equality *The implication (2.4), or equivalently the implication (2.3), is satisfied if $\exists v \in R^r$ such the following linear equality in v is satisfied:*

$$K(x', B')u - \gamma - 1 + v'g(x)_+ = 0, \quad \forall x \in \Gamma_1. \quad (2.5)$$

Proof If the implication (2.4) does not hold then for some $x \in \Gamma_1$ such that $g(x)_+ = 0$ it follows that $K(x', B')u - \gamma \neq 1$. However this leads to the following contradiction for that x :

$$0 = K(x', B')u - \gamma - 1 + v'g(x)_+ = K(x', B')u - \gamma - 1 \neq 0, \quad (2.6)$$

where the first equality follows from (2.5), the second equality from $g(x)_+ = 0$ and the inequality follows from $K(x', B')u - \gamma \neq 1$. \square

We note that the motivation for this proposition comes from the theorem for the alternative for convex functions [25, Theorem 2.1], [20, Corollary 4.2.2], for which the implication $f(x) \leq 0 \implies \theta(x) > 0, \forall x \in \Gamma$ follows from $w'f(x) + \theta(x) \geq 0, \forall x \in \Gamma$ for some $w \geq 0$, where $f : \Gamma \subset R^n \longrightarrow R^r, \theta : \Gamma \longrightarrow R$ and $w \in R^r$.

We turn now to our proximal classification formulation of the knowledge-based nonlinear kernel classification by utilizing Proposition 2.1 above.

3 NONLINEAR PRIOR KNOWLEDGE CLASSIFICATION VIA PROXIMAL SUPPORT VECTOR MACHINES

We first formulate the classification problem (2.2) without knowledge using a proximal support vector machine approach [6] by allowing a minimal amount of error in data fitting and a minimal number of kernel functions. The error in a proximal formulation is measured by closeness to the two following bounding surfaces that are parallel, in the u -space, to the classifier (2.1):

$$\begin{aligned} K(x', B')u - \gamma &= +1 \\ K(x', B')u - \gamma &= -1 \end{aligned} \quad (3.7)$$

We measure the error in satisfying (2.2) by how close the points in the class +1 are to the first surface of (3.7) and the points in the class -1 are to the second surface of (3.7). This error can be succinctly written as $\|D(K(A, B')u - e\gamma) - e\|$. Minimizing the square of this error with parameter weight $\nu/2$ and the square of the space variables (u, γ) for model simplicity, we obtain our proximal support vector machine classification formulation without prior knowledge:

$$\min_{(u,\gamma)} \frac{\nu}{2} \|D(K(A, B') - e\gamma) - e\|^2 + \frac{1}{2} \left\| \begin{bmatrix} u \\ \gamma \end{bmatrix} \right\|^2. \quad (3.8)$$

This unconstrained quadratic optimization problem can be solved by setting its gradient, a system of linear equations in (u, γ) , equal to zero. However before we do that we shall introduce prior knowledge in the form of the linear equality (2.5) with weight σ to be satisfied in a least square sense at ℓ discrete points in the set Γ_1 as follows:

$$\min_{(u,\gamma,v)} \frac{\nu}{2} \|D(K(A, B') - e\gamma) - e\|^2 + \frac{\sigma}{2} \sum_{i=1}^{\ell} (K(x^{i'}, B')u - \gamma - 1 + v'g(x^i)_+)^2 + \frac{1}{2} \left\| \begin{bmatrix} u \\ \gamma \\ v \end{bmatrix} \right\|^2. \quad (3.9)$$

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 (2.4) we have the implication:

$$h(x)_+ = 0 \implies K(x', B')u - \gamma = -1, \quad \forall x \in \Gamma_2. \quad (3.10)$$

Here, $h(x) : \Gamma_2 \subset R^n \rightarrow R^s$ is an s -dimensional function defined on a subset Γ_2 of R^n that determines the region in the input space where prior knowledge requires that the classifier $K(x', B') - \gamma$ be equal to -1 in order to classify the points $x \in \{x \mid h(x) \leq 0\}$ as belonging to the class -1 . By Proposition 2.1 this implication is satisfied if $\exists p \in R^s$ such that:

$$K(x', B')u - \gamma + 1 + p'h(x)_+ = 0, \quad \forall x \in \Gamma_2. \quad (3.11)$$

Discretizing this condition over t points in Γ_2 and incorporating it into the minimization problem (3.9) results in our final unconstrained minimization problem that incorporates prior knowledge for both classes $+1$ and -1 as follows:

$$\begin{aligned} \min_{(u,\gamma,v,p)} \frac{\nu}{2} \|D(K(A, B') - e\gamma) - e\|^2 + \frac{\sigma}{2} \sum_{i=1}^{\ell} (K(x^{i'}, B')u - \gamma - 1 + v'g(x^i)_+)^2 \\ + \frac{\sigma}{2} \sum_{j=1}^t (K(x^{j'}, B')u - \gamma - 1 + p'h(x^j)_+)^2 + \frac{1}{2} \left\| \begin{bmatrix} u \\ \gamma \\ v \\ p \end{bmatrix} \right\|^2. \end{aligned} \quad (3.12)$$

We note immediately that the objective function of (3.12) is strongly convex with a positive definite symmetric Hessian matrix. Hence (3.12) has a unique solution obtained by setting its gradient, which consists of a system of $k + 1 + r + s$ nonsingular linear equations in as many unknowns (u, γ, v, p) , equal to zero as follows:

$$\begin{aligned}
& \nu K(B, A') D(D(K(A, B')u - e\gamma) - e) + \sigma \sum_{i=1}^{\ell} K(B, x^i) (K(x^{i'}, B')u - \gamma - 1 + v'g(x^i)_+) \\
& + \sigma \sum_{j=1}^t K(B, x^j) (K(x^{j'}, B')u - \gamma + 1 + p'h(x^j)_+) + u = 0 \\
& -\nu e' D(D(K(A, B')u - e\gamma) - e) + \sigma \sum_{i=1}^{\ell} - (K(x^{i'}, B')u - \gamma - 1 + v'g(x^i)_+) \\
& + \sigma \sum_{j=1}^t - (K(x^{j'}, B')u - \gamma + 1 + p'h(x^j)_+) + \gamma = 0 \\
& \sigma \sum_{i=1}^{\ell} g(x^i)_+ (K(x^{i'}, B')u - \gamma - 1 + v'g(x^i)_+) + v = 0 \\
& \sigma \sum_{j=1}^t h(x^j)_+ (K(x^{j'}, B')u - \gamma + 1 + p'h(x^j)_+) + p = 0
\end{aligned} \tag{3.13}$$

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 the tests performed in [25] where an altogether different approach using linear *inequalities* was utilized in a linear programming formulation for *nonlinear* kernel classification. We also include new experiments in which prior knowledge is generated by ordinary classifiers of labeled datasets. In [25], prior knowledge was incorporated into three publicly available datasets: the Checkerboard dataset [12], the Spiral dataset [31], and the Wisconsin Prognostic Breast Cancer (WPBC) dataset [27]. We utilize here the same prior knowledge as in [25], however we employ our new proposed formulation instead. All results exhibit improvement as a consequence of the incorporation of prior knowledge.

4.1 CHECKERBOARD AND SPIRAL PROBLEMS

The frequently used checkerboard [12, 15, 22, 17, 9] and spiral [31, 6] datasets are synthetic datasets for which prior knowledge can be easily constructed [25, 9]. The checkerboard dataset consists of points with labels “black” and “white” arranged in the shape of a checkerboard, while the spiral dataset consists of points from two concentric spirals. Table 1 shows both the accuracy and CPU time needed to run the experiments of [25] using both the linear programming formulation originally used in [25] and our proposed proximal formulation. Both experiments were carried out using the procedures and prior knowledge described in [25]. For the checkerboard experiment, the knowledge consists only of the two leftmost squares of the bottom row, while for the spiral dataset, the knowledge was constructed by inspecting the source code used to generate the spiral. In the checkerboard experiment, the matrix B has 16 rows, and the prior knowledge is imposed at 200 points. In the spiral dataset, the matrix B has 194 rows, and the knowledge is imposed at 194 points. The experiments were performed using MATLAB 7.2 [26] under CentOS Linux 4.4 on an Intel Pentium IV 3 GHz processor with 1 gigabyte of RAM. The running times were calculated by the MATLAB profiler, and represent the total time

Dataset	Linear Programming SVM [25]	Proximal SVM	Time
	Accuracy CPU Time in Seconds	Accuracy CPU Time in Seconds	Ratio
Checkerboard without Knowledge	89.2% 2.3	94.2% 0.2	11.5
Checkerboard with Knowledge	100% 26.4	98.0% 3.2	8.3
Spiral without Knowledge	79.9% 21.3	80.4% 4.3	5.0
Spiral with Knowledge	100% 300.2	100% 19.0	15.8

Table 1: Accuracy and CPU time in seconds for the linear programming formulation [25] and the proposed proximal formulation. Each running time result is the total time needed to set up and solve the optimization problem, either as a linear program or a linear system of equations, 225 times. The time ratio is the time for the linear programming formulation divided by the time for the proximal formulation.

during the experiment consumed in setting up and solving the optimization problem. Linear programs were solved using CPLEX 9.0 [14], and the linear equations were solved using the `chol` routine of MATLAB. On both datasets, 225 optimization problems were solved. For the spiral dataset, flush-to-zero mode was enabled to speed the multiplication of numbers with very small magnitude. This change had a significant impact on the running time of our proximal formulation, and negligible impact on the linear programming formulation. Note that the proximal formulation has similar accuracy to the linear programming formulation, while being *approximately an order of magnitude faster* to solve. We further note that considering only the time taken to solve the linear program or linear system of equations gives a similar result, thus we do not believe that the difference in computation time can be attributed to the setup procedure.

4.2 PREDICTING BREAST CANCER SURVIVAL TIME

We have also tested our proposed proximal formulation on the Wisconsin Prognostic Breast Cancer (WPBC) dataset [27]. 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. In [25] prior knowledge was used to achieve 91% correctness on this task. In this dataset, 81.9% of patients are cancer free after 24 months. To our knowledge, the best result on this dataset without the knowledge used in [25] is 86.3% correctness obtained by Bennett in [1]. We have repeated the experiment of [25], which we describe below for the sake of completeness.

Prior knowledge for this dataset was obtained by plotting the number of metastasized lymph nodes against the tumor size, along with the class label, for each patient. An oncological surgeon’s advice was simulated 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

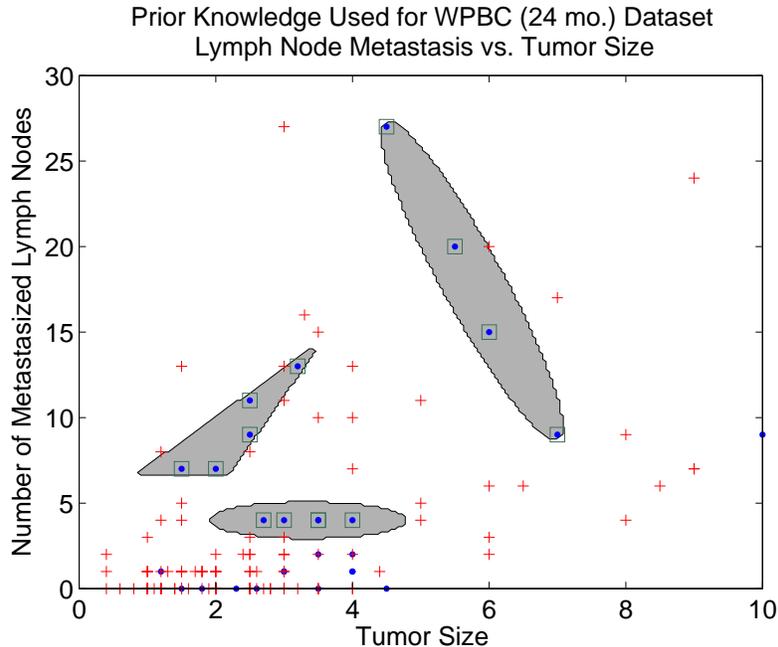


Figure 1: Number of metastasized lymph nodes versus tumor size for the WPBC (24 mo.) dataset. The solid dots represent patients who experienced a recurrence within 24 months of surgery, while the crosses represent the cancer free patients. The shaded regions which correspond to the areas in which the left-hand side of one of the three implications in the form of Equation (2.3) 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.

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 constructed for the WPBC dataset used in [25] consists of three implications on the three regions shown in Figure 1. The shaded regions indicate areas in which the left-hand side of an implication of the form (2.3) is true and simulate an oncological surgeon’s prior knowledge regarding patients likely to experience a recurrence within 2 years. The prior knowledge was imposed at the dataset points within the three regions, marked with squares in Figure 1. Details of these regions are given in [25].

In order to evaluate our proposed proximal 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 from [25]. For both cases the rows of the matrices A and B of (3.12) were set to the usual values, that is to the coordinates of the points of the training set. The

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

Table 2: Leave-one-out misclassification rate of classifiers with and without knowledge on the WPBC (24 mo.) dataset. Best result is in bold.

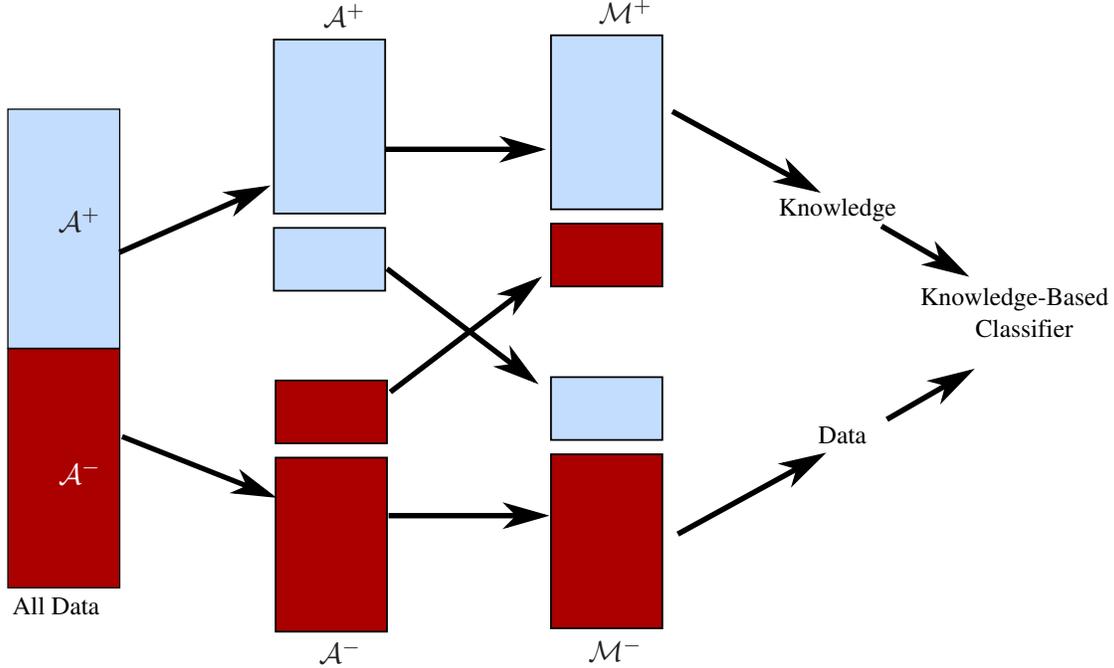


Figure 2: Generation of prior knowledge from a standard dataset. The dataset is first separated into the datasets \mathcal{A}^+ which consists of all +1 points, and \mathcal{A}^- which consists of all -1 points. Then the mostly +1 dataset \mathcal{M}^+ is formed by replacing a small fraction of +1 points in \mathcal{A}^+ with an equal number of -1 points from \mathcal{A}^- . The mostly -1 dataset \mathcal{M}^- is formed from the points not used in \mathcal{M}^+ . We use \mathcal{M}^+ to produce prior knowledge, and \mathcal{M}^- as ordinary data. Combining the knowledge from \mathcal{M}^+ and the data from \mathcal{M}^- leads to a knowledge-based classifier which is superior to a classifier formed using either \mathcal{M}^+ as pure knowledge or \mathcal{M}^- as pure data alone.

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 $\{2^i | i \in \{-7, \dots, 7\}\}$ 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 for simplicity, which corresponds to very strict adherence to the prior knowledge. The results are summarized in Table 2. The reduction in misclassification rate indicates that our proximal approach can achieve the same 50% improvement in classification accuracy using prior knowledge as the linear programming formulation [25].

4.3 GENERATING PRIOR KNOWLEDGE FROM ORDINARY CLASSIFICATION DATASETS

In order to further demonstrate the effectiveness of our proposed formulation, we have developed a procedure by which prior knowledge is generated from a subset of an ordinary classification dataset. In order for prior knowledge to improve classification accuracy when combined with ordinary data, the prior knowledge and the data must contain different information about the “true” dataset. Thus, we simulate a situation in which the use of either the data or prior knowledge alone will give poor results whereas a knowledge-based classifier using both prior knowledge and data will be superior. In our scenario, the set \mathcal{M}^+ will consist mostly of points from the class +1 and will be used only to generate prior knowledge, while the set \mathcal{M}^- will consist mostly of points from the class -1 and will be used only as ordinary data. The motivation for this scenario is a situation in which prior knowledge is available

about data in the set \mathcal{M}^+ which contains mostly +1 points, while available data has mostly label -1, and is available in the set \mathcal{M}^- . Thus, the learning algorithm will need to incorporate both prior knowledge about \mathcal{M}^+ and the conventional data in \mathcal{M}^- in order to generalize well to new points. Our results show that our approach can effectively incorporate prior knowledge and enhance classification accuracy.

Construction of the sets \mathcal{M}^+ and \mathcal{M}^- is illustrated in Figure 2. We first take the set \mathcal{A}^+ as that consisting of all the points with label +1, and \mathcal{A}^- consisting of all the points with label -1. Then, to form the mostly +1 dataset \mathcal{M}^+ , we took all but a small percentage of the points in \mathcal{A}^+ , and a few points in \mathcal{A}^- so that \mathcal{M}^+ and \mathcal{A}^+ had equal cardinality. The points in \mathcal{A}^+ and \mathcal{A}^- not used to form \mathcal{M}^+ were all used to form \mathcal{M}^- . Thus, \mathcal{M}^+ contains mostly points with label +1 and a small number of points with label -1. Similarly, \mathcal{M}^- contains mostly points with label -1 and a small number of points with label +1. The points from \mathcal{A}^- that were used in \mathcal{M}^+ and the points in \mathcal{A}^+ that were used in \mathcal{M}^- were randomly selected. To explore the behavior of our approach, we varied the percentage of negative points in \mathcal{M}^+ . For a sufficiently large dataset, as this percentage approaches fifty percent one expects that \mathcal{M}^+ and \mathcal{M}^- will contain the same information, and the gain due to incorporating prior knowledge will be minimal.

One can imagine many methods of automatically generating prior knowledge from \mathcal{M}^+ , such as [11, 5, 3]. However, we used the simple approach of learning a proximal support vector machine on the points in \mathcal{M}^+ . The knowledge we used was the following:

$$(-\phi(x))_+ = 0 \implies K(x', B')u - \gamma = 1, \forall x \in \Gamma_1, \quad (4.14)$$

where $\phi(x)$ is the classifier function (2.1) learned on the set \mathcal{M}^+ . This knowledge simply states that if the proximal support vector machine represented by $\phi(x)$ labels the point as +1, then the point should be labeled +1 by the classifier which combines both data and knowledge. In addition to the proximal support vector machine we fit a multivariate normal distribution to the points in \mathcal{M}^+ , and impose the prior knowledge of (4.14) at a random sample drawn from this distribution. Although we chose the multivariate normal distribution for simplicity, one can easily imagine a more sophisticated density estimate being used. However, we leave the investigation of different methods of generating prior knowledge to future research.

Figure 3 shows the result of applying the above procedure to Thompson’s Normally Distributed Clusters on Cubes (NDCC) dataset [29]. This dataset generates points according to multivariate normal distributions centered at the vertices of two concentric 1-norm cubes. Points are mostly labeled according to the cube they were generated from, with some specified fraction of noisy labels. We generated a dataset of 20000 points in R^{50} , with ten percent label noise. We used 300 points as a training set, 2000 separate points as a tuning set to choose parameters, and the remaining 17700 points as a testing set to evaluate the classifiers. In Figure 3, we compare an approach using only the data in the set \mathcal{M}^- and no prior knowledge to an approach based on the same data *plus* prior knowledge obtained from the points in \mathcal{M}^+ . The knowledge was imposed on $|\mathcal{M}^+|$ randomly sampled points as described above, where $|\mathcal{M}^+|$ is the cardinality of \mathcal{M}^+ . In our experience on this dataset, reducing the number of sampled points to less than half the cardinality of \mathcal{M}^+ had very little impact on accuracy. Determining the appropriate number of points to sample for a given dataset is left to future work. We note that the approach using prior knowledge is able to approach ten percent misclassification error even when relatively few points in \mathcal{M}^- have label +1.

Figure 4 shows the result of applying the above procedure to the publicly available Wisconsin Diagnostic Breast Cancer (WDBC) dataset [27]. In this dataset, the task is to classify tumors as either malignant or benign based on the 30 features given. We chose to label malignant tumors +1, to simulate the scenario in which most information about malignant tumors is available only through prior

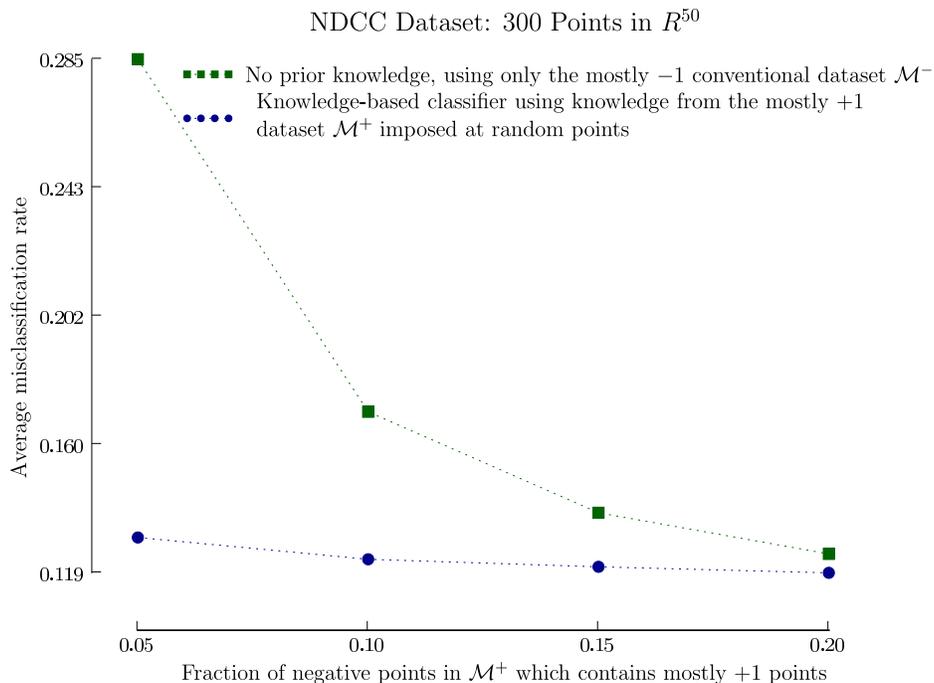


Figure 3: Prior knowledge experiment on the WDBC dataset.

knowledge, while information about benign tumors is more readily gathered. To assess the generalization performance of our approach, we computed ten-fold cross validation misclassification rates. We chose all parameters from the set $\{2^i | i = -7, \dots, 7\}$ using a random ten percent of the training set as a tuning set. When using prior knowledge, we set the value of σ to be equal to ν . In carrying out the cross validation experiment, \mathcal{M}^+ and \mathcal{M}^- were formed from the training set for each fold. In Figure 4, three different approaches are compared. In the first approach, represented by squares, the classifier is learned using only the data in \mathcal{M}^- with *no* prior knowledge. This classifier performs poorly until a sufficient number of +1 points are present in \mathcal{M}^- . The second approach, represented by circles, learns a classifier using the data in \mathcal{M}^- *plus* the prior knowledge from \mathcal{M}^+ described by (4.14). The knowledge was imposed at a set with the same cardinality as \mathcal{M}^+ , but containing randomly generated points as described above. We note that the use of prior knowledge results in considerable improvement, especially when there are few points in \mathcal{M}^+ with class -1. Finally, we include an approach represented by triangles which uses no prior knowledge, but *all* the data as a reference. Note that this classifier has the same misclassification rate regardless of the fraction of negative points in \mathcal{M}^+ . We have included this approach as a reference to illustrate that our approach is able to use the prior knowledge generated from \mathcal{M}^+ to recover most of the information in \mathcal{M}^+ . Recall that we are simulating a situation in which \mathcal{M}^+ is only available as prior knowledge.

Figure 5 shows the results of the same procedure as used for the WDBC dataset on the publicly available Ionosphere dataset [27]. Note that this dataset exhibits similar behavior to the WDBC dataset. The classifier using only the points in \mathcal{M}^- and *no* prior knowledge has a high error rate until a sufficiently large number of positive points are included in \mathcal{M}^- . The classifier which makes use of the prior knowledge from \mathcal{M}^+ is again much closer to the error rate attained by a classifier which uses all data.

WDBC Dataset: 569 Points in R^{30}
 212 Malignant Tumors Labeled +1

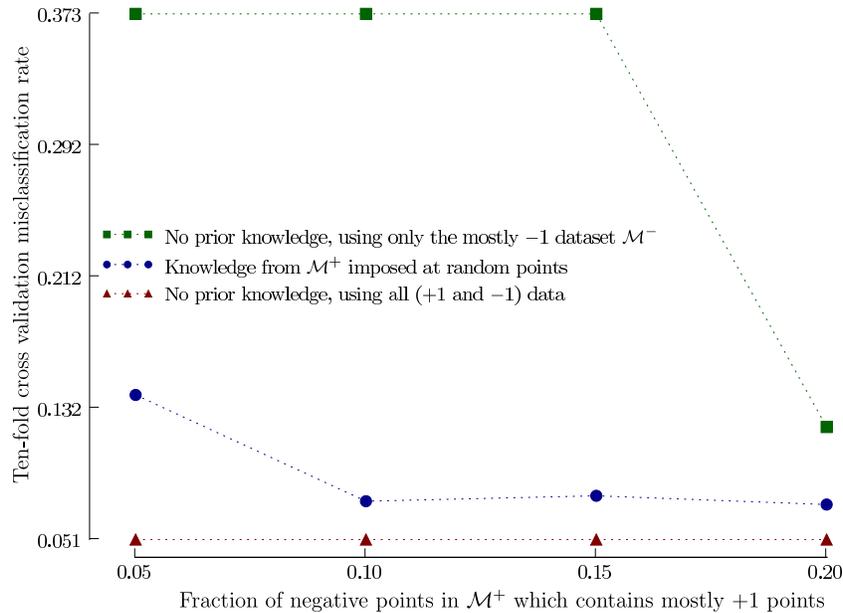


Figure 4: Prior knowledge experiment on the WDBC dataset.

5 CONCLUSION AND OUTLOOK

We have proposed a computationally effective framework for handling general nonlinear prior knowledge in proximal kernel classification problems. We have reduced such prior knowledge to an easily implemented linear equation that can be incorporated into an unconstrained strongly convex quadratic programming problem. We have demonstrated the effectiveness of our approach on a number of publicly available datasets. Possible future extensions are to even more general prior knowledge, such as that where the right hand side of the implications (2.3) and (3.10) are replaced by very general nonlinear inequalities involving the classification function (2.1). Proximal knowledge-based approximation would be an interesting piece of future work as well as the construction of 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 06-05, November 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.

Ionosphere Dataset: 351 Points in R^{34}
 126 Bad Radar Returns Labeled +1

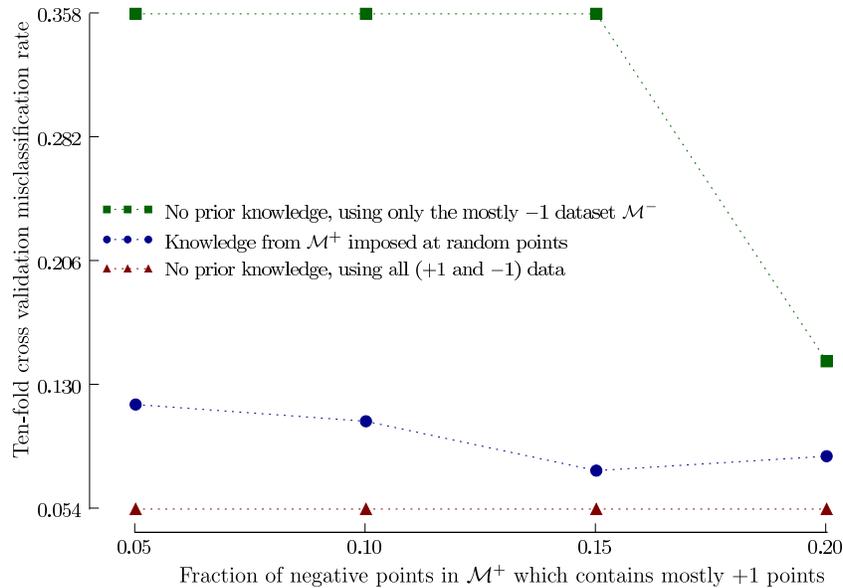


Figure 5: Prior knowledge experiment on the Ionosphere dataset.

- [3] M. Craven and J. Shavlik. Learning symbolic rules using artificial neural networks. In *Proceedings of the 10th International Conference on Machine Learning*, pages 73–80, Amherst, MA, 1993. Morgan Kaufmann.
- [4] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, Cambridge, 2000.
- [5] W. Duch, R. Adamczak, and K. Grąbczewski. A new methodology of extraction, optimization and application of crisp and fuzzy logical rules. *IEEE Transactions on Neural Networks*, 2001.
- [6] 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>.
- [7] G. Fung and O. L. Mangasarian. Incremental support vector machine classification. In H. Manilla R. Grossman and R. Motwani, editors, *Proceedings of the Second SIAM International Conference on Data Mining, Arlington, Virginia, April 11-13, 2002*, pages 247–260, Philadelphia, 2002. SIAM. <ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/01-08.ps>.
- [8] G. Fung and O. L. Mangasarian. Multicategory proximal support vector machine classifiers. *Machine Learning*, pages 77–97, 2005. University of Wisconsin Data Mining Institute Technical Report 01-06, July 2001, <ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/01-06.ps>.
- [9] 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.
- [10] 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.
- [11] G. Fung, S. Sandilya, and B. Rao. Rule extraction for linear support vector machines. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 32–40, 2005.
 - [12] T. K. Ho and E. M. Kleinberg. Checkerboard dataset, 1996. <http://www.cs.wisc.edu/math-prog/mpml.html>.
 - [13] 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.
 - [14] ILOG, Incline Village, Nevada. *ILOG CPLEX 9.0 User's Manual*, 2003. <http://www.ilog.com/products/cplex/>.
 - [15] 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.
 - [16] Q. V. Le, A. J. Smola, and T. Gärtner. Simpler knowledge-based support vector machines. In *Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, PA, 2006*, 2006. <http://www.icml2006.org/icml2006/technical/accepted.html>.
 - [17] 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>.
 - [18] 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*, pages 819–824, 2005.
 - [19] 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.
 - [20] O. L. Mangasarian. *Nonlinear Programming*. SIAM, Philadelphia, PA, 1994.
 - [21] 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>.
 - [22] 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>.
 - [23] 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>.
 - [24] 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.pdf>. IEEE Transactions on Neural Networks, to appear.
 - [25] O. L. Mangasarian and E. W. Wild. Nonlinear knowledge-based classification. Technical Report 06-04, Data Mining Institute, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, August 2006. <ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/06-04.pdf>.
 - [26] MATLAB. *User's Guide*. The MathWorks, Inc., Natick, MA 01760, 1994-2006. <http://www.mathworks.com>.
 - [27] P. M. Murphy and D. W. Aha. UCI machine learning repository, 1992. www.ics.uci.edu/~mllearn/MLRepository.html.
 - [28] B. Schölkopf and A. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.
 - [29] M. E. Thompson. NDCC: normally distributed clustered datasets on cubes, 2006. www.cs.wisc.edu/dmi/svm/ndcc/.
 - [30] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, second edition, 2000.
 - [31] 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>.