Knowledge-Based Nonlinear Kernel Classifiers

Glenn M. Fung, Olvi L. Mangasarian *, and Jude W. Shavlik

Computer Sciences Department, University of Wisconsin
Madison, WI 53706
{gfung,olvi,shavlik}@cs.wisc.edu

Abstract. Prior knowledge in the form of multiple polyhedral sets, each belonging to one of two categories, is introduced into a reformulation of a nonlinear kernel support vector machine (SVM) classifier. The resulting formulation leads to a linear program that can be solved efficiently. This extends, in a rather unobvious fashion, previous work [3] that incorporated similar prior knowledge into a *linear* SVM classifier. Numerical tests on standard-type test problems, such as exclusive-or prior knowledge sets and a checkerboard with 16 points and prior knowledge instead of the usual 1000 points, show the effectiveness of the proposed approach in generating sharp nonlinear classifiers based mostly or totally on prior knowledge.
Keywords: ***prior knowledge, support vector machines, linear programming***

1 Introduction

Support vector machines (SVMs) have played a major role in classification problems [15, 2, 10]. However unlike other classification tools such as knowledge-based neural networks [13, 14, 4], little work [11, 3] has gone into incorporating prior knowledge into support vector machines. In this work we extend the previous work [3] of incorporating multiple polyhedral sets as prior knowledge for a linear classifier to nonlinear kernel-based classifiers. This extension is not an obvious one, since it depends critically on the theory of *linear* inequalities and cannot be incorporated directly into a nonlinear kernel classifier. However, if the "kernel trick" is employed *after* one uses a theorem of the alternative for linear inequalities [9, Chapter 2], then incorporation of polyhedral knowledge sets into a nonlinear kernel classifier can be achieved. We show this in Section 2 of the paper. In Section 3 we derive a linear programming formulation that generates a nonlinear kernel SVM classifier that is based on conventional data as well as two groups of polyhedral sets, each group of which belongs to one of two classes. We note that conventional datasets are not essential to our formulation and can be surrogated by samples taken from the knowledge sets. In Section 4 we test our formulation on standard-type test problems. The first test problem is the exclusive-or (XOR) problem consisting of four points in 2-dimensional input space plus four

* Also Department of Mathematics, University of California at San Diego, La Jolla, CA 92093.

polyhedral knowledge sets, all of which get classified perfectly by a Gaussian kernel knowledge-based classifier. The second test problem is the checkerboard problem consisting of 16 two-colored squares. Typically this problem is classified based on 1000 points. Here, by using only 16 points plus prior knowledge, our knowledge-based nonlinear kernel classifier, generates a sharp classifier that is as good as that obtained by using 1000 points. Section 5 concludes the paper.

We now describe our notation. 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 $x \in R^n$, $\|x\|_p$ denotes the p-norm, $p = 1, 2, \infty$. 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 and A_i will denote the i-th row 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. The identity matrix of arbitrary dimension will be denoted by I. A *separating plane*, with respect to two given point sets $\mathcal{A}$ and $\mathcal{B}$ in R^n, is a plane that attempts to separate R^n into two halfspaces such that each open halfspace contains points mostly of $\mathcal{A}$ or $\mathcal{B}$. A *bounding plane* to the set $\mathcal{A}$ is a plane that places $\mathcal{A}$ in one of the two closed halfspaces that the plane generates. The abbreviation "s.t." stands for "such that". 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 definiteness condition [15, 12]. The base of the natural logarithm will be denoted by ε. A frequently used kernel in nonlinear classification is the Gaussian kernel [15, 2, 10] whose ijth element, $i = 1 \dots, m$, $j = 1 \dots, k$, 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.

2 Prior Knowledge in a Nonlinear Kernel Classifier

We begin with a brief description of support vector machines (SVMs). SVMs are used principally for classification [10, 15, 12]. The simplest classifier is a *linear separating surface*, a plane in R^n:

$$x'w = \gamma, \tag{1}$$

where $w \in R^n$ determines the orientation of the plane (1), in fact it is the normal to the plane, and γ determines the location of the plane relative to the origin. The separating plane (1) lies midway between two parallel *bounding planes*:

$$\begin{aligned} x'w &= \gamma + 1, \\ x'w &= \gamma - 1, \end{aligned} \tag{2}$$

each of which attempts to place each class of points in one of the two halfspaces:

$$\begin{aligned} &\{x \mid x'w \geq \gamma + 1\}, \\ &\{x \mid x'w \leq \gamma - 1\}. \end{aligned} \tag{3}$$

In addition, these bounding planes are pushed apart as far as possible. To obtain a more complex classifier, one resorts to a nonlinear separating surface in R^n instead of the linear separating surface (1) defined as follows [10]:

$$K(x', A')Du = \gamma, \tag{4}$$

where K is an arbitrary nonlinear kernel as defined in the Introduction, and (u, γ) are determined by solving the linear program (19). Here, $A \in R^{m\times n}$ represents a set of m points in R^n each of which belonging to class A^+ or A^- depending on whether the corresponding element of a given $m \times m$ diagonal matrix D is $+1$ or -1, and $u \in R^m$ is a dual variable. The linear separating surface (1) becomes a special case of the nonlinear surface (4) if we use the linear kernel $K(A, A') = AA'$ and set $w = A'Du$ [10, 8].

We turn now to the incorporation of prior knowledge in the form of a polyhedral set into a nonlinear kernel classifier. But first, we show that a routine incorporation of such knowledge leads to a nonlinear system of nonconvex inequalities that are not very useful.

Suppose that the polyhedral $\{x \mid Bx \le b\}$ where $B \in R^{\ell\times n}$ and $b \in R^\ell$, must lie in the halfspace $\{x \mid x'w \ge \gamma + 1\}$ for some given $w \in R^n$ and $\gamma \in R$. We thus have the implication:

$$Bx \le b \implies x'w \ge \gamma + 1. \tag{5}$$

By letting w take on its dual representation $w = A'Du$ [10, 8], the implication (5) becomes:

$$Bx \le b \implies x'A'Du \ge \gamma + 1. \tag{6}$$

If we now "kernelize" this implication by letting $x'A' \longrightarrow K(x', A')$, where K is some nonlinear kernel as defined in the Introduction, we then have the implication, for a given A, D, u and γ, that:

$$Bx \le b \implies K(x', A')Du \ge \gamma + 1. \tag{7}$$

This is equivalent to the following nonlinear, and generally nonconvex, system of inequalities *not* having a solution x for a given A, D, u and γ:

$$Bx \le b,\ K(x', A')Du < \gamma + 1. \tag{8}$$

Unfortunately, the nonlinearity and nonconvexity of the system (8) precludes the use of any theorem of the alternative for either linear or convex inequalities [9]. We thus have to backtrack to the implication (6) and rewrite it equivalently as the following system of homogeneous linear inequalities not having a solution $(x, \zeta) \in R^{n+1}$ for a given fixed u and γ:

$$\begin{array}{rr} Bx & -b\zeta \le 0, \\ u'DAx & -(\gamma+1)\zeta < 0, \\ & -\zeta < 0. \end{array} \tag{9}$$

Here, the positive variable ζ is introduced in order to make the inequalities (9) homogeneous in (x, ζ), thus enabling us to use a desired theorem of the alternative [9] for such linear inequalities. It follows by Motzkin's Theorem of the Alternative [9], that (9) is equivalent to the following system of linear inequalities having a solution in $(v, \eta, \tau) \in R^{\ell+1+1}$ for a given fixed u and γ:

$$\begin{array}{rlr} B'v & +(A'Du)\eta & = 0, \\ -b'v & -(\gamma+1)\eta \quad -\tau & = 0, \\ v & & \geq 0, \\ & 0 \neq (\eta, \tau) & \geq 0. \end{array} \tag{10}$$

Here, the last constraint signifies that at most one of the two nonnegative variables η and τ can be zero. Hence, if $\eta = 0$, then $\tau > 0$. It follows then from (10) that there exists a v such that: $B'v = 0,\ -b'v > 0,\ v \geq 0$, which contradicts the natural assumption that the knowledge set $\{x \mid Bx \leq b\}$ is nonempty. Otherwise, we have the contradiction:

$$0 = v'Bx \leq b'v < 0. \tag{11}$$

Hence $\eta > 0$ and $\tau \geq 0$. Dividing the inequalities of (10) by η and redefining v as $\frac{v}{\eta}$, we have from (10) that the following system of linear equalities has a solution v for a given u and γ:

$$\begin{array}{rlr} B'v & +A'Du & = 0, \\ b'v & +\gamma+1 & \leq 0, \\ v & & \geq 0. \end{array} \tag{12}$$

Under the rather natural assumption that A has linearly independent columns, this in turn is equivalent to following system of linear equalities having a solution v for a given u and γ:

$$\begin{array}{rlr} AB'v & +AA'Du & = 0, \\ b'v & +\gamma+1 & \leq 0, \\ v & & \geq 0. \end{array} \tag{13}$$

Note that the linear independence is needed only for (13) to imply (12). Replacing the the linear kernels AB' and AA' by the general nonlinear kernels $K(A, B')$ and $K(A, A')$, we obtain that the following system of linear equalities has a solution v for a given u and γ:

$$\begin{array}{rlr} K(A, B')v & +K(A, A')Du & = 0, \\ b'v & +\gamma+1 & \leq 0, \\ v & & \geq 0. \end{array} \tag{14}$$

This is the set of constraints that we shall impose on our nonlinear classification formulation as a surrogate for the implication (7). Since the derivation of the conditions were not directly obtained from (7), it is useful to state precisely what the conditions (14) are equivalent to. By using a similar reasoning that employs theorems of the alternative as we did above, we can derive the following equivalence result which we state without giving its explicit proof. The proof is very similar to the arguments used above.

Proposition 21 Knowledge Set Classification *Let*

$$\{y \mid K(B, A')y \le b\} \ne \emptyset. \tag{15}$$

Then the system (14) having a solution v, for a given u and γ, is equivalent to the implication:

$$K(B, A')y \le b \implies u'DK(A, A')y \ge \gamma + 1. \tag{16}$$

We note that the implication is not precisely the implication (7) that we started with, but can be thought of as a kernelized version of it. To see this we state a corollary to the above proposition which shows what the implication means for a linear kernel AA'.

Corollary 22 Linear Knowledge Set Classification *Let*

$$\{y \mid Bx \le b,\ x = A'y\} \ne \emptyset. \tag{17}$$

For a linear kernel $K(A, A') = AA'$, the system (14) having a solution v, for a given u and γ, is equivalent to the implication:

$$Bx \le b,\ x = A'y \implies w'x \ge \gamma + 1,\ w = A'Du,\ x = A'y. \tag{18}$$

We immediately note that the implication (18) is equivalent to the desired implication (5) for linear knowledge sets, under the rather unrestrictive assumption that A has linearly independent columns. That the columns of A are linearly independent is equivalent to assuming that in the input space, features are not linearly dependent on each other. If they were, then linearly dependent features could be easily removed from the problem.

We turn now to a linear programming formulation of a nonlinear kernel classifier that incorporates prior knowledge in the form of multiple polyhedral sets.

3 Knowledge-Based Linear Programming Formulation of Nonlinear Kernel Classifiers

A standard [10, 1] linear programming formulation of a nonlinear kernel classifier is given by:

$$\begin{array}{rr} \displaystyle\min_{u,\gamma,r,y} & \nu e'y + e'r \\ \text{s.t.} & D(K(A, A')Du - e\gamma) + y \ge e, \\ & -r \le u \le r, \\ & y \ge 0. \end{array} \tag{19}$$

The (u, γ) taken from a solution (u, γ, r, y) of (19) generates the nonlinear separating surface (4). Suppose now that we are given the following *knowledge sets*:

$$\begin{array}{l} p \text{ sets belonging to } A+ : \{x \mid B^i x \le b^i\},\ i = 1, \ldots, p, \\ q \text{ sets belonging to } A- : \{x \mid C^i x \le c^i\},\ i = 1, \ldots, q. \end{array} \tag{20}$$

It follows from the implication (7) for $B = B^i$ and $b = b^i$ for $i = 1, \ldots, p$ and its consequence, the existence of a solution to (14), and a similar implication for the sets $\{x \mid C^i x \le c^i\},\ i = 1, \ldots, q$, that the following holds:

$$\begin{array}{l} \text{There exist } s^i,\ i = 1, \ldots, p,\ t^j,\ j = 1, \ldots, q, \text{ such that:} \\ K(A, B^{i'})s^i + K(A, A')Du = 0,\ b^{i'}s^i + \gamma + 1 \le 0,\ s^i \ge 0,\ i = 1, \ldots, p, \\ K(A, C^{j'})t^j - K(A, A')Du = 0,\ c^{j'}t^j - \gamma + 1 \le 0,\ t^j \ge 0,\ j = 1, \ldots, q. \end{array} \tag{21}$$

We now incorporate the knowledge sets (20) into the nonlinear kernel classifier linear program (19) by adding conditions (21) as constraints to (19) as follows:

$$\begin{array}{rr} \min\limits_{u,\gamma,r,(y,s^i,t^j)\ge 0} & \nu e'y + e'r \\ \text{s.t.} & D(K(A, A')Du - e\gamma) + y \ge e, \\ & -r \le u \le r, \\ & K(A, B^{i'})s^i + K(A, A')Du = 0, \\ & b^{i'}s^i + \gamma + 1 \le 0,\ i = 1, \ldots, p, \\ & K(A, C^{j'})t^j - K(A, A')Du = 0, \\ & c^{j'}t^j - \gamma + 1 \le 0,\ j = 1, \ldots, q. \end{array} \tag{22}$$

This linear programming formulation incorporates the knowledge sets (20) into the appropriate halfspaces in the higher dimensional feature space. However since there is no guarantee that we are able to place each knowledge set in the appropriate halfspace, we need to introduce error variables $z_1^i, \zeta_1^i, i = 1, \ldots, p, z_2^j, \zeta_2^j, j = 1, \ldots, q$, just like the error variable y of the SVM formulation (19), and attempt to drive these error variables to zero by modifying our last formulation above as follows:

$$\begin{array}{rr} \min\limits_{u,\gamma,r,z_1^i,z_2^j,(y,s^i,t^j,\zeta_1^i,\zeta_2^j)\ge 0} & \nu e'y + e'r + \mu(\sum\limits_{i=1}^{p}(e'z_1^i + \zeta_1^i) \quad + \sum\limits_{j=1}^{q}(e'z_2^j + \zeta^j)) \\ \text{s.t.} & D(K(A, A')Du - e\gamma) + y \ge e, \\ & -r \le u \le r, \\ & -z_1^i \le K(A, B^{i'})s^i + K(A, A')Du \le z_1^i, \\ & b^{i'}s^i + \gamma + 1 \le \zeta_1^i,\ i = 1, \ldots, p, \\ & -z_2^j \le K(A, C^{j'})t^j - K(A, A')Du \le z_2^j, \\ & c^{j'}t^j - \gamma + 1 \le \zeta_2^j,\ j = 1, \ldots, q. \end{array} \tag{23}$$

This is our final knowledge-based linear programming formulation which incorporates the knowledge sets (20) into the linear classifier with weight μ, while the (empirical) error term $e'y$ is given weight ν. As usual, the value of these two parameters, ν, μ, are chosen by means of a tuning set extracted from the training set.

Remark 31 Data-Based and Knowledge-Based Classifiers *If we set* $\mu = 0$*, then the linear program (23) degenerates to (19), the linear program associated with an ordinary data-based nonlinear kernel SVM. We can also make the linear program (23), which generates a nonlinear classifier, to be* **only knowledge-based** *and not dependent on any specific training data* **if** *we replace the matrix* A *appearing everywhere in (23) by a random sample of points taken from the knowledge sets (20) together with the associated diagonal matrix* D*. This might be a useful paradigm for situations where training datasets are not easily available, but expert knowledge, such as doctors' experience in diagnosing certain diseases, is readily available. In fact, using this idea of making* A *and* D *random samples drawn from the knowledge sets (20), the linear programming formulation (23)* **as is** *can be made* **totally** *dependent on prior knowledge only.*

We turn now to our numerical experiments.

4 Numerical Experience

The focus of this paper is rather theoretical. However, in order to illustrate the power of the proposed formulation, we tested our algorithm on two synthetic examples for which most or all the data is constituted of knowledge sets. Experiments involving real world knowledge sets will be utilized in future work.

Exclusive-Or (XOR) Knowledge Sets

This example generalizes the well known XOR example which consists of the four vertices of a rectangle in 2-dimensions, with the pair of vertices on the end of one diagonal belonging to one class (crosses) while the other pair belongs to another class (stars). Figure 1 depicts two such pairs of vertices symmetrically placed around the origin. It also depicts two pairs of knowledge sets with each pair belonging to one class (two triangles and two parallelograms respectively).

The given points in this XOR example can be considered in two different ways. We note that in line with Remark 31, this classifier can be considered either partially or totally dependent on prior knowledge, depending on whether the four data points are given independently of the knowledge sets or as points contained in them. Our knowledge-based linear programming classifier (23) with a Gaussian kernel yielded the depicted nonlinear separating surface that classified all given points and sets correctly.

Another realization of the XOR example is depicted in Figure 2. Here the data points are not positioned symmetrically with respect to the origin and only one of them is contained in a knowledge set. The resulting nonlinear separating surface for this XOR example is constrained by one of the knowledge sets, in fact it is tangent to one of the diamond-shaped knowledge sets.

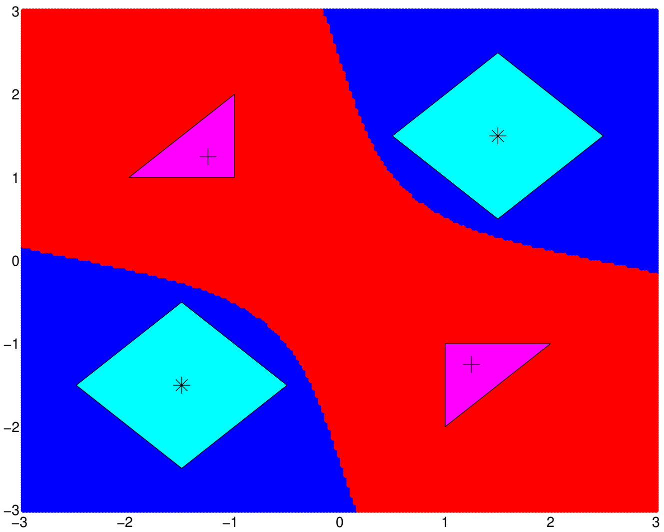

Fig. 1. **Totally or partially knowledge-based XOR classification problem. The nonlinear classifier obtained by using a Gaussian kernel in our linear programming formulation (23), completely separates the two pairs of prior knowledge sets as well the two pairs of points. The points can be treated as samples taken from the knowledge sets or given independently of them.**

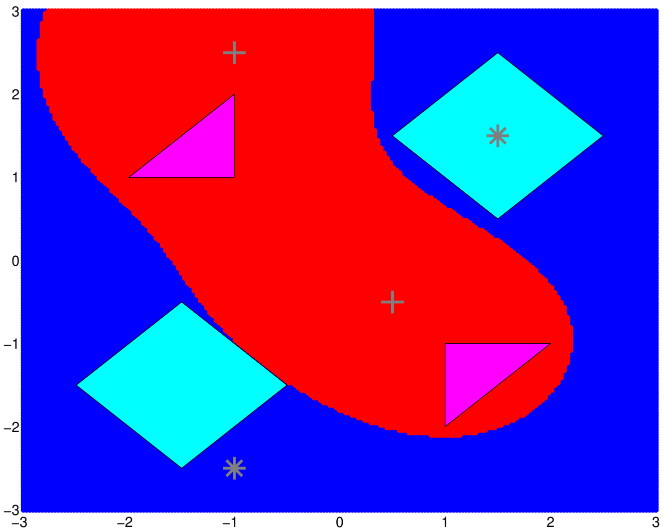

Fig. 2. **Another XOR classification problem where only one of the points is contained in a knowledge set. The nonlinear classifier obtained by using a Gaussian kernel in our linear programming formulation (23), completely separates the two pairs of prior knowledge sets as well the two pairs of points. Note the strong influence of the knowledge sets on the separating surface which is tangent to one of the knowledge sets.**

Checkerboard

Our second example is the classical checkerboard dataset [5, 6, 8, 7] which consists of 1000 points taken from a 16-square checkerboard. The following experiment on this dataset shows the strong influence of knowledge sets on the separating surface.

We first took a subset of 16 points only, each one is the “center” of one of the 16 squares. Since we are using a nonlinear Gaussian kernel to model the separating surface, this particular choice of the training set is very appropriate for the checkerboard dataset. However, due to the nature of the Gaussian function it is hard for it to learn the “sharpness” of the checkerboard by using only a 16-point Gaussian kernel basis. We thus obtain a fairly poor Gaussian-based representation of the checkerboard depicted in Figure 3 with correctness of only 89.66% on a testing set of uniformly randomly generated 39,601 points labeled according to the true checkerboard pattern.

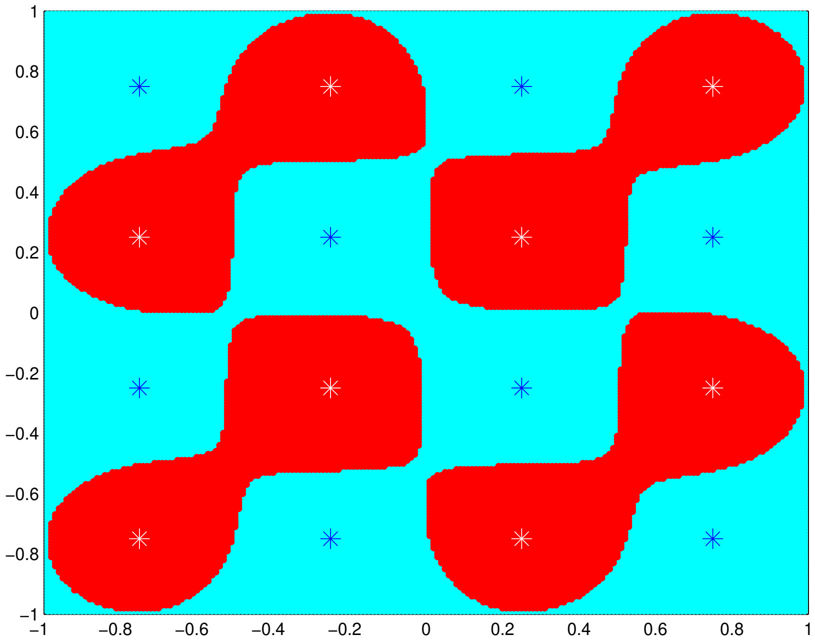

Fig. 3. A poor nonlinear classifier based on 16 points taken from each of the 16 squares of a checkerboard. The nonlinear classifier was obtained by using a Gaussian kernel in a conventional linear programming formulation (19).

On the other hand, if we use these same 16 points in the linear program (23) as a distinct dataset in conjunction with prior knowledge in the form of 8 linear inequalities characterizing *only two* subsquares fully contained in the leftmost two squares of the bottom row of squares, we obtain the very sharply defined checkerboard depicted in Figure 4, with a correctness of 98.5% on a $39,601$-point testing set. We note that, it does not matter which two subsquares are taken as prior knowledge, as long as there is one from each class. Also the size of the subsquare is not critical either. Figure 4 was obtained with subsquares with sides equal to 0.75 times the the original sides and centered around the given data points. Squares down to size 0.25 of the original squares, gave similar results. We emphasize here that the prior knowledge given to the linear program (23) here is truly a partial knowledge set in the sense that it gives the linear program information on only 2 of 16 squares.

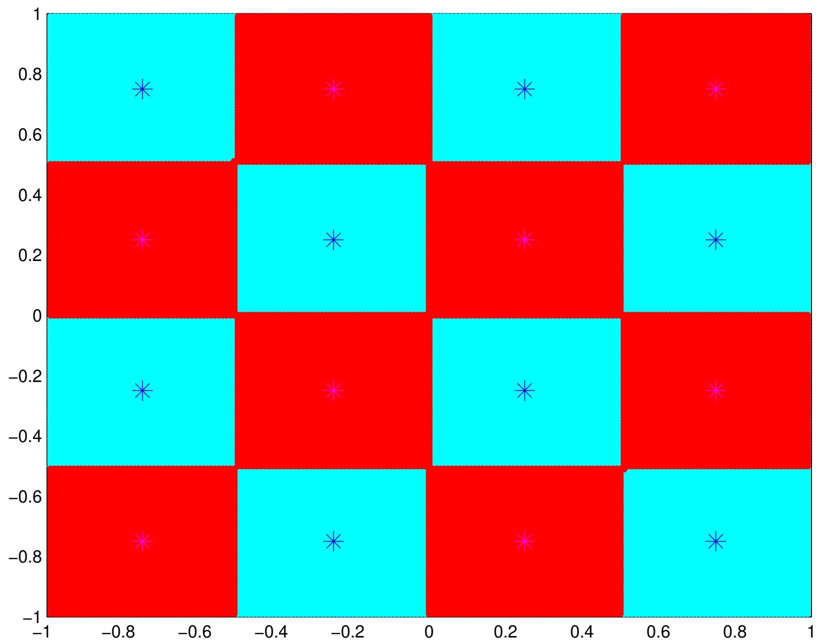

Fig. 4. Knowledge-based checkerboard classification problem. The nonlinear classifier was obtained by using a Gaussian kernel in our linear programming formulation (23) with the 16 depicted points as a given dataset together with prior knowledge consisting of 8 linear inequalities characterizing *only two* subsquares contained in the leftmost two squares of the bottom row of squares. The sharply defined checkerboard has a correctness of 98.5% on a $39,601$-point testing set.

It is interesting to note that by using prior knowledge from *only two* squares, our linear programming formulation (23) is capable of transforming a complex boundary between the union of eight squares in each class, into a *single* nonlinear classifier equation given by (4).

5 Conclusion

We have presented a knowledge-based formulation of a nonlinear kernel SVM classifier. The classifier is obtained using a linear programming formulation with any nonlinear symmetric kernel with no positive definiteness (Mercer) condition assumed. The formulation works equally well with or without conventional datasets. We note that unlike the linear kernel case with prior knowledge [3], where the absence of conventional datasets was handled by deleting some constraints from a linear programming formulation, here arbitrary representative points from the knowledge sets are utilized to play the role of such datasets, that is A and D in (23). The issues associated with sampling the knowledge sets, in situations where there are no conventional data points, constitute an interesting topic for future research.

Future application to medical problems, computer vision, microarray gene classification, and efficacy of drug treatment, all of which have prior knowledge available, are planned.

Acknowledgments

Research described in this UW Data Mining Institute Report 03-02, March 2003, was supported by NSF Grant CCR-138308, by NLM Grant 1 R01 LM07050-01, and by Microsoft.

References

1. P. S. Bradley and O. L. Mangasarian. Feature selection via concave minimization and support vector machines. In J. Shavlik, editor, *Machine Learning Proceedings of the Fifteenth International Conference(ICML '98)*, pages 82–90, San Francisco, California, 1998. Morgan Kaufmann. ftp://ftp.cs.wisc.edu/math-prog/tech-reports/98-03.ps.
2. V. Cherkassky and F. Mulier. *Learning from Data - Concepts, Theory and Methods*. John Wiley & Sons, New York, 1998.
3. G. Fung, O. L. Mangasarian, and J. Shavlik. Knowledge-based support vector machine classifiers. Technical Report 01-09, Data Mining Institute, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, November 2001. ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/01-09.ps, NIPS 2002 Proceedings, to appear.
4. F. Girosi and N. Chan. Prior knowledge and the creation of "virtual" examples for RBF networks. In *Neural networks for signal processing, Proceedings of the 1995 IEEE-SP Workshop*, pages 201–210, New York, 1995. IEEE Signal Processing Society.

5. T. K. Ho and E. M. Kleinberg. Building projectable classifiers of arbitrary complexity. In *Proceedings of the 13th International Conference on Pattern Recognition*, pages 880–885, Vienna, Austria, 1996. http://cm.bell-labs.com/who/tkh/pubs.html. Checker dataset at: ftp://ftp.cs.wisc.edu/math-prog/cpo-dataset/machine-learn/checker.
6. T. K. Ho and E. M. Kleinberg. Checkerboard dataset, 1996. http://www.cs.wisc.edu/math-prog/mpml.html.
7. Y.-J. Lee and O. L. Mangasarian. RSVM: Reduced support vector machines. Technical Report 00-07, Data Mining Institute, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, July 2000. Proceedings of the First SIAM International Conference on Data Mining, Chicago, April 5-7, 2001, CD-ROM Proceedings. ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/00-07.ps.
8. Y.-J. Lee and O. L. Mangasarian. SSVM: A smooth support vector machine. *Computational Optimization and Applications*, 20:5–22, 2001. Data Mining Institute, University of Wisconsin, Technical Report 99-03. ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/99-03.ps.
9. O. L. Mangasarian. *Nonlinear Programming.* SIAM, Philadelphia, PA, 1994.
10. 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.
11. 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.
12. A. Smola and B. Schölkopf. *Learning with Kernels.* MIT Press, Cambridge, MA, 2002.
13. G. G. Towell and J. W. Shavlik. Knowledge-based artificial neural networks. *Artificial Intelligence*, 70:119–165, 1994.
14. G. G. Towell, J. W. Shavlik, and M. Noordewier. Refinement of approximate domain theories by knowledge-based artificial neural networks. In *Proceedings of the Eighth National Conference on Artificial Intelligence (AAAI-90)*, pages 861–866, 1990.
15. V. N. Vapnik. *The Nature of Statistical Learning Theory.* Springer, New York, second edition, 2000.