Journal of Classification 20:3-23 (2003)
DOI: 10.1007/s00357-003-0003-7

Classification Using Class Cover Catch Digraphs

Carey E. Priebe
Johns Hopkins University

David J. Marchette
Naval Surface Warfare Center, Virginia

Jason G. DeVinney
Center for Computing Sciences, Maryland

Diego A. Socolinsky
Equinox Corporation, Maryland

Abstract: We present a semiparametric mixture methodology for classification which involves modelling the class-conditional discriminant regions via collections of balls. The number, location, and size of the balls are determined adaptively through consideration of dominating sets for *class cover catch digraphs* based on proximity between training observations. Performance comparisons are presented on synthetic and real examples versus k-nearest neighbors, Fisher's linear discriminant and support vector machines. We demonstrate that the proposed semiparametric classifier has performance approaching that of the optimal parametric classifier in cases for which the optimal is available for comparison.

Keywords: Classification; Random graph; Class cover; Prototype selection.

CEP and JGD partially supported by Office of Naval Research Grant N00014-01-1-0011. CEP and DAS partially supported by Defense Advanced Research Projects Agency Grant F49620-01-1-0395. DJM partially supported by the Office of Naval Research In-House Laboratory Independent Research Program. Minefield data was provided by NSWC Coastal Systems Station, Dahlgren Division, Panama City, Florida, with support from United States Marine Corps Amphibious Warfare Technology; special thanks to R.R. Muise. The authors thank Willem Heiser and three anonymous referees for helpful editorial suggestions.

Authors' Addresses: Carey E. Priebe, Department of Mathematical Sciences, Johns Hopkins University, Baltimore, MD 21218-2682, USA; David J. Marchette, Naval Surface Warfare Center, B10, Dahlgren, VA 22448, USA; Jason G. DeVinney, Center for Computing Sciences, 17100 Science Dr., Bowie, MD 20715, USA; Diego A. Socolinsky, Equinox Corporation, 207 East Redwood St., Suite 205, Baltimore, MD 21202, USA.

1. Introduction

Throughout this paper, we focus on classification in the two-class case. We are given as training data $\mathcal{X}_0$ and $\mathcal{X}_1$, two finite non-empty sets of class-conditional observations with elements from a set $\mathcal{X}$. The set $\mathcal{X}$ may be any such that it admits a dissimilarity measure as defined in Section 1.1. In particular, $\mathcal{X}$ may be a vector space. For $j = 0, 1$, we denote the class-conditional distributions by F_j and the class-conditional sample sizes by $n_j = |\mathcal{X}_j|$. Our goal is to design a classifier $g : \mathcal{X} \times \mathcal{X}^{n_0} \times \mathcal{X}^{n_1} \rightarrow \{0, 1\}$ such that, when presented with training data $\mathcal{X}_0$,$\mathcal{X}_1$ and an unlabelled $\mathcal{X}$-valued observation Z with true but unobserved class label Y in $\{0, 1\}$, the (conditional) probability of misclassification $L(g) := P[g(Z; \mathcal{X}_0, \mathcal{X}_1) \neq Y | \mathcal{X}_0, \mathcal{X}_1]$ is close to Bayes optimal L^*. That is, we want the probability of misclassification to be as low as possible. See for instance Devroye, Gyorfi and Lugosi (1996) or Kulkarni, Lugosi and Venkatesh (1998).

Class cover catch digraphs, introduced in Priebe, DeVinney and Marchette (2001) and studied further in DeVinney and Priebe (2001+), are proximity graphs defined via the relationship between class-labeled observations. Each class j gives rise to a digraph D_j. The vertices of D_j are the class-conditional observations $\mathcal{X}_j$ and a directed edge between two vertices exists if the proximity of the two vertices is small compared to the proximity of the vertices to the observations from class $1 - j$. We use these digraphs to determine a classifier g.

The classifier described in this paper utilizes the proximity digraphs to construct a low-complexity representation of each class, which can then be used as a set of prototypes in a reduced nearest-neighbor classifier. The representation modelling each class takes the form of a union of balls — appropriately defined in Section 1.1 — whose radii depend on the distribution of $\mathcal{X}_0$ and $\mathcal{X}_1$ about the center of the ball. The radii are such that any given ball will cover mostly same-class observations. While we may have a large amount of training data for each class, we seek a compact representation, both in order to improve generalization ability of the classifier and to lower the computational burden of classification. We achieve a low-complexity model by selecting a small subset of the training data in each class such that the union of balls about that subset covers all but a small fraction of the training data. By considering the proximity digraphs for each class in Section 2.2, we cast subset selection as the problem of finding a minimum dominating set for the digraphs. A greedy algorithm is used to produce an approximate solution to this NP-hard graph-theoretic problem in polynomial time.

There are several methods in the literature that have a similar flavor to the class cover catch digraph approach. Perhaps most closely related is the reduced Coulomb energy network (see Duda, Hart and Stork 2001) which in its simplest form corresponds to a class cover catch digraph classifier without our complexity-reducing subset optimization. Our selection of the covering subset of balls is a kind of training set reduction, reminiscent of nearest neighbor editing. For example, Dasrathy and Sánchez (2000) provides a methodology for nearest neighbor editing using proximity graphs, where the criterion for selecting the prototypes is classification directly, rather than coverage as in the class cover catch digraphs. See also Skalak (1997) for related work on prototype selection for nearest neighbor classifiers. Ho and Basu (2002) describes a method of Lebourgeois and Emptoz (1996) in which the classes are covered by balls and the decision boundary is described in terms of a small number of the balls which are close to the boundary. Radial basis function neural networks (see Jain, Mao and Hohluddin 1996) for an introduction to these and other neural network models) relax the requirement that the balls be centered at observations, and seek to find a small number of balls that provide good classification. Thus there is a relationship between class cover catch digraph classification and k-means clustering (Duda, Hart and Stork 2001) as well as support vector machines (Joachims 1999; Vapnik 1995) using radial kernels.

We present results from applying our proposed classifier in synthetic and real-world classification examples. For the synthetic examples, we have full knowldege of the class-conditional densities, and thus can compare the results obtained with our classifier to the Bayes optimal bound. In these cases, we show that the new semi-parametric method can perform nearly as well as the optimal parametric classifier, and significantly better than nearest neighbor and support vector machine classifiers. As a real-world example, we use the well-known COBRA dataset, widely used for mine and minefield detection research (Smith 1995; Witherspoon et al. 1995). For this challenging dataset, we demonstrate performance superior to optimized k-nearest neighbors, as well as support vector machines with linear, polynomial and radial kernels.

1.1 Notation

Let $\mathcal{X}$ be a set, and let $\rho : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$ be a dissimilarity measure (e.g., a distance function) on $\mathcal{X}$; that is, for all $x, y \in \mathcal{X}$ we require $0 \leq \rho(x, y) = \rho(y, x) < \infty$ and $\rho(x, y) = 0$ if and only if $x = y$. Note that we do *not* require ρ to be a distance, in particular the triangle inequality need not be satisfied.

For $x \in \mathcal{X}$ and $r \geq 0$ we define the open ball of radius r centered at x to be $B(x, r) = \{x' \in \mathcal{X} : \rho(x, x') < r\}$. The balls of radius zero are the empty set; $B(x, 0) = \emptyset$. If ρ is a distance function, then our open balls agree with the open spheres defined by that distance.

For $x \in \mathcal{X}$ and a finite subset $S \subset \mathcal{X}$ we define

$$\rho(x, S) := \min_{x' \in S} \rho(x, x').$$

More generally, for $k = 1, \cdots, |S|$ we define the order statistic $\rho_{(k)}(x, S)$ to be k^{th} smallest of the $\rho(x, x')$, $x' \in S$. Here and hereafter, $|S|$ denotes the cardinality of the set S. For notational purposes we set $\rho_{(0)}(x, S) := 0$ and $\rho_{(|S|+1)}(x, S) := \infty$.

2. Class Cover Catch Digraphs

The proposed method involves the construction of data-random digraphs D_0, D_1 for the supervised two-class classification problem, followed by complexity reduction via dominating sets S_0, S_1 for these digraphs. Classification is then approached via mixtures based on these sets. The present section introduces the necessary graph-theoretic background and terminology.

2.1 Digraphs

A digraph $D = (V, A)$ consists of a vertex set V and an edge (or arc) set A. The set A is a collection of *ordered* pairs $(x, x') \in V \times V$ indicating an edge from vertex x to vertex x'. We identify the vertex set with the set of target class observations, so that $V_j = \mathcal{X}_j$. The *class cover catch digraph (cccd)* D_j for $\mathcal{X}_j$ against $\mathcal{X}_{1-j}$ is defined by first specifying $\mathcal{X}_j$ as the *target class* and $\mathcal{X}_{1-j}$ as the non-target class. Thus the definition of the *cccd*s D_0, D_1 are identical, except that the roles of $\mathcal{X}_0, \mathcal{X}_1$ are swapped.

For the *cccd* D_j, we consider open balls of radius $r(x)$ centered at each target class observation $x \in \mathcal{X}_j$. For distinct $x, x' \in \mathcal{X}_j$ the digraph has an edge from x to x' if and only if x' is in the ball about x; $(x, x') \in A \iff x' \in B(x, r(x))$. In order to fully define D_j, it remains to specify $r(x)$ for each $x \in \mathcal{X}_j$.

We specify a "robustness to contamination" parameter $\beta_j \in \{0, \cdots, n_{1-j} - 1\}$ which determines how many non-target class observations each ball may contain. The proposed classification strategy leverages the property that each ball cover as many target observations as possible while covering at most β_j non-target class observations; $r(x; \beta_j) = \rho_{(\beta_j+1)}(x, \mathcal{X}_{1-j})$ satisfies these covering requirements. Note that $\beta_j = -1 \implies D_j$ is the empty digraph — no edges — and $\beta_j = n_{1-j} \implies D_j$ is the complete digraph — all edges. It is not the case that each ball will necessarily cover β_j non-target class observations, however. In Section 3. we show these radii may be reduced appropriately so that no more non-target observations are covered than is necessary

In addition to depending on the data $\mathcal{X}_0, \mathcal{X}_1$ and the dissimilarity ρ, the digraph pair D_0, D_1 depends on the (possibly different) choices of β_0, β_1.

The *cccd*s defined above are a special case of Maehara's sphere digraphs (Maehara 1984). Because D_0, D_1 are defined by identifying the *random* sets of points $\mathcal{X}_0, \mathcal{X}_1$ with the vertex sets V_0, V_1, the digraphs are themselves random, of a type termed *vertex random* digraphs (Karonski et al. 1999).

2.2 Dominating Sets

The support for each class-conditional distribution can now be modeled as a mixture of balls. We have two collections of class-conditional observations, and a radius associated with each observation. Our estimate for support(F_j) is given by $\cup_{x \in \mathcal{X}_j} B(x, r(x; \beta_j))$. Note that that this may be interpreted as a kernel density estimate, where the kernel is spherical — with respect to the dissimilarity measure — and its bandwidth varies adaptively with the training data. We now wish to reduce the complexity of the model by selecting an appropriate subset of the balls in the class cover. Such a complexity reduction is accomplished by choosing appropriate subsets — small *dominating sets* — $S_j \subset \mathcal{X}_j$ of the vertices of *cccd* digraphs.

For a digraph $D = (V, A)$, the open neighborhood of a vertex $v \in V$, denoted $N(v)$, is the collection of vertices $w \in V$ such that $(v, w) \in A$. The closed neighborhood is defined by $\bar{N}(v) = N(v) \cup \{v\}$. For a set of vertices $S \subset V$, we have $N(S) = \cup_{v \in S} N(v)$ and $\bar{N}(S) = \cup_{v \in S} \bar{N}(v)$. A *dominating set* S for the digraph $D = (V, A)$ is a set $S \subset V$ such that, for all $w \in V$, either $w \in S$ or $(v, w) \in A$ for some $v \in S$. That is, S is a dominating set for D if and only if $\bar{N}(S) = V$. The graph invariant $\gamma(D)$ is defined as the cardinality of the smallest dominating set(s) of D. Clearly, $1 \leq \gamma(D) \leq |V|$. A minimum dominating set for D is defined as a dominating set with cardinality $\gamma(D)$.

Finding a minimum dominating set is, in general, an NP-Hard optimization problem (Karp 1972; Arora and Lund 1997). An approximately minimum dominating set $\hat{S}$ can be obtained in $O(|V|^2)$ using a well-known greedy algorithm (Chvatal 1979; Parekh 1991). The greedy algorithm begins by selecting the vertex with the largest cardinality neighborhood: $S^1 = \{v^1\}$, where $v^1 \in \arg\max_{v \in V} |\bar{N}(v)|$. Throughout this procedure, cardinality ties are broken by randomization. Then, for iteration $t \geq 2$ and until $\cup_{v \in S^{t-1}} \bar{N}(v) = V$ the algorithm selects $v^t \in \arg\max_{v \in V \setminus S^{t-1}} |\bar{N}(v) \setminus \cup_{v' \in S^{t-1}} \bar{N}(v')|$ and sets $S^t = S^{t-1} \cup \{v^t\}$. This process is guaranteed to terminate after at most $|V|$ iterations. If the algorithm terminates after t^* iterations, the set $\hat{S} = S^{t^*}$ is a dominating set. The approximation for the domination number γ of the digraph D is $\hat{\gamma} = |\hat{S}|$; $\hat{\gamma} \geq \gamma$.

When applied to the class digraphs defined above, at each iteration the greedy algorithm selects from among the vertices whose ball covers the most as yet unaccounted for target-class observations. We use the greedy algorithm to identify (hopefully small) dominating sets $\hat{S}_j \subset \mathcal{X}_j$ for *cccd*s $D_j = (\mathcal{X}_j, A_j)$, $j = 0, 1$. This provides a reduced complexity mixture of balls which still model the class-conditional distribution supports. Our estimate for support(F_j) is given by $\cup_{x \in \hat{S}_j} B(x, r(x; \beta_j))$.

For the purposes of classification, and to further reduce model complexity, we wish to obtain an estimate of the *high probability* region of the class-conditional distribution supports (Scholkopf, et al. 2001). Toward this end, we define parameters $0 \leq \alpha_j \leq n_j - 1$ and halt the greedy algorithm when $|\bar{N}(\hat{S}_j)| \geq n_j - \alpha$. (Note that for $\alpha_j \geq n_j$, $\hat{S}_j = \emptyset$.) That is, we do not require a proper dominating set, but are willing to settle for an approximate dominating set in order to reduce model complexity. The parameters α_j allow for "robustness to outliers", as measured by the "improperness" $n_j - |\bar{N}(\hat{S}_j)|$. For example, setting $\alpha_j > 0$ may leave singletons — target class j observations x for which $\mathcal{X}_j \cap B(x, r(x; \beta_j)) = \{x\}$ — uncovered.

Figure 1 shows how the number of balls in each class-cover grows as the number of training samples per class increases. Data points for each class are shown along with the respective log-linear fit, the correlation coefficient of which is above 0.97 for both classes. We see how the complexity of the class-cover — and as seen below, of the resulting classifier — grows roughly with the logarithm of the training sample size. The data used to generate these results lies on the unit sphere in $\mathbb{R}^{441}$, and comes from vectorized images of faces (class 0) and non-faces (class 1). (See Socolinsky et al. 2003 for further details regarding this application.)

3. Classification

The approximate dominating sets $\hat{S}_0$,$\hat{S}_1$ depend on the training data $\mathcal{X}_0$,$\mathcal{X}_1$, on the dissimilarity ρ, and on the choice of α_j,β_j. Selecting $\alpha_j > 0$ allows the model $\cup_{x \in \hat{S}_j} B(x, r(x; \beta_j))$ to neglect a few "outlying" target class training observations, while $\beta_j > 0$ allows the model to ignore a few "contaminating" non-target class training observations. These are precisely the types of "robustness" that classification models require. Furthermore, non-zero choices for the parameters α_j,β_j serve to reduce the complexity of the model used to define class-conditional discriminant regions in $\mathcal{X}$ for the classification problem.

The requirement that the ball about class j training observation x cover as many target class observations as possible while covering at most β_j non-target class observations allows — unless ties occur — some flexibility in the choice of ball radius. Clearly, $r(x)$ must be at least large enough so that

$$\mathcal{X}_j \cap B(x, \rho_{(\beta_j+1)}(x, \mathcal{X}_{1-j})) \subset B(x, r(x)).$$

Let $r_j(x; \beta_j) := \max_{x' \in \mathcal{X}_j \cap B(x, \rho_{(\beta_j+1)}(x, \mathcal{X}_{1-j}))} \rho(x, x')$. Then, for $\tau_j \in (0, 1]$,

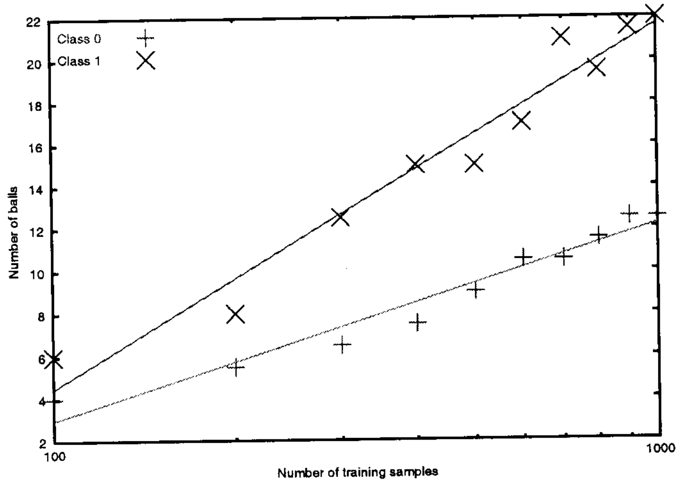

Figure 1. Number of balls in class cover versus logarithm of training sample size for 441-dimensional data.

the radii $r(x;\beta_j,\tau_j) := r_j(x;\beta_j) + \tau_j \cdot (\rho_{(\beta_j+1)}(x,\mathcal{X}_{1-j}) - r_j(x;\beta_j))$ satisfy the covering requirements. For $\tau_j = 0$ we define $r(x;\beta_j,0) := r_j(x;\beta_j) + \epsilon$ for $0 < \epsilon < \min_j \min_{x \in \mathcal{X}_{1-j}} \rho_{(\beta_j+1)}(x,\mathcal{X}_{1-j})$ to avoid the possibility of vanishing radius.

Note that the digraphs are independent of the choice of τ_0, τ_1, so that the selection of $\hat{S}_j$ does not require prior selection of τ_j. However, the classifier constructed based on the digraphs will depend on the τ_j.

3.1 Preclassifier

The sets $\hat{S}_j$ and the radii depend on the choice of $\alpha_j, \beta_j, \tau_j$; we suppress this dependency for notational clarity. Elements of $\hat{S}_j$ are selected prototypes for the problem of modelling the class-conditional discriminant regions via collections of balls. The associated radii represent an estimate of the domain of influence, or region in which a given prototype should influence class labelling. This is the region where the density of class j observations is high relative to the density of class $1-j$ observations. Let $C_j := \cup_{x \in \hat{S}_j} B(x, r(x))$ be the *class cover* for class j. We define the *preclassifier* m to be $m(z) = j$ if $z \in C_j \backslash C_{1-j}$, $m(z) = -1$ if $z \in C_j \cap C_{1-j}$ and $m(z) = -2$ if $z \in (C_j \cup C_{1-j})^c$. That is, the preclassifier m allows for two types of "no decision"; $m(z) = -1$ means z may well be from either class 0 or class 1, while $m(z) = -2$ means the answer may be neither.

Such "no decision" decisions can be quite valuable in practical classification applications. In particular, it may be the case that a well-founded "no decision" will compel the investigator to collect additional data. To investigate the relative performance of such preclassifiers, one may resort to decision-theoretic considerations and define a loss function which prescribes appropriate relative costs for the various misclassifications and no decisions. However, there are cases in which we wish to demand that a decision be made for all $z \in \mathcal{X}$. Furthermore, in such cases and with the additional assumption that all classification errors be treated equally, it can be easier to compare competing procedures.

3.2 A Reduced Nearest Neighbor Classifier

A *reduced* (or "edited") nearest neighbor classifier (Devroye, Gyorfi and Lugosi 1996) based on the prototypes $\hat{S}_j$ is given by

$$g_{RNN}(z) := I\{\min_{x \in \hat{S}_1} \rho(x,z) < \min_{x \in \hat{S}_0} \rho(x,z)\}.$$

With $\hat{S}_j = \mathcal{X}_j$, g_{RNN} is in fact the standard one-nearest neighbor classifier g_{1NN} (Cover and Hart 1967). Thus our methodology can be used for the selection of a reduced set of exemplars for nearest neighbor classification. This is a canonical problem in classification. However, our $\hat{S}_j$ are not chosen so as to be good sets of reduced nearest neighbor exemplars. On the contrary, our $\hat{S}_j$ are chosen to provide a good "class cover" estimate of the discriminant regions, in the sense of the preclassifier m described above. The classifier g_{cccd} proposed in the next subsection, rather than g_{RNN}, is the appropriate classifier for investigation based on our $\hat{S}_j$.

3.3 The Class Cover Catch Digraph Classifier

We propose a classifier g_{cccd} which is consistent with the class covers C_j. That is, we require that g_{cccd} agree with the preclassifier m whenever m makes a decision (whenever $m(z) \in \{0,1\}$). For $z \in (C_0 \cap C_1) \cup (C_0^c \cap C_1^c)$ — the cases in which m chooses not to decide — we choose a class label based on the *locally scaled* distances defined by the ball radii r. The classifer g_{cccd} is given by

$$g_{cccd}(z) := I\{\min_{x \in \hat{S}_1} \rho(x,z)/r(x) < \min_{x \in \hat{S}_0} \rho(x,z)/r(x)\}.$$

The distances to the observations in the prototype sets $\hat{S}_j$ are scaled by the associated ball radius. Note that if $\hat{S}_j = \mathcal{X}_j$ and all the radii are identical, the classifier g_{cccd} reduces to g_{1NN}. Also note that if the dissimilarity measure is the standard L^2 norm, the class cover may be interpreted as a mixture of Gaussians with spherical covariance with the classifier determined by the smallest Mahalanobis distance.

For sets of observations $\mathcal{X}_0$ and $\mathcal{X}_1$, we define their dissimilarity by $\rho(\mathcal{X}_0, \mathcal{X}_1) = \min\{\rho(x_i, x_j) \mid x_i \in \mathcal{X}_0, x_j \in \mathcal{X}_1\}$. By construction, we have the following result regarding the empirical, or resubstitution, error rate $\hat{L}^{(R)}(g_{cccd}) = (n_0+n_1)^{-1}\left[\sum_{x\in\mathcal{X}_0} I\{g_{cccd}(x) \neq 0\}+\sum_{x\in\mathcal{X}_1} I\{g_{cccd}(x) \neq 1\}\right]$.

Theorem 1 *Let $\hat{S}_j$ be dominating sets for cccds D_j with $\alpha_j = \beta_j = 0$ and $\tau_j \in (0, 1]$. If $\rho(\mathcal{X}_0, \mathcal{X}_1) > 0$ then $\hat{L}^{(R)}(g_{cccd}) = 0$. That is, the resubstitution error rate for the cccd classifier is zero if the dissimilarity between the training sets is strictly positive.*

Theorem 1 holds for any dissimilarity ρ. This result follows from the facts that $\beta_j = 0$ implies that the covers C_j are pure (contain no class $1 - j$ training observations), $\alpha_j = 0$ implies that the covers C_j are proper (contain all class j training observations), and g_{cccd} agrees with the preclassifier m on the training data.

Recall that a classifier is *consistent* if its error rate approches that of the Bayes optimal classifier as the size of the training set grows to infinity (Debroye, Gyorfi and Lugosi 1996). For consistency, a general dissimilarity is not sufficient. A simple consistency result assumes that ρ is a distance function satisfying the *continuity condition*

$$P_F[\rho(X, z) < \epsilon] > 0 \text{ for any } z \in s(F) \text{ and } \epsilon > 0, \qquad \text{(CC)}$$

where $s(F)$ denotes the *support* of distribution F. This condition rules out the discrete metric ($\rho(x, y) = 1$ for $x \neq y$, under which g_{cccd} is not consistent) but allows for instance the L_p distances. We say class-conditional distributions F_0, F_1 are *strictly separable* (Devroye, Gyorfi and Lugosi 1996) if

$$\inf_{x_0,x_1:x_j\in s(F_j)} \rho(x_0, x_1) = \delta > 0.$$

Weak consistency for strictly separable class-conditional distributions F_j is established for *i.i.d.* training data by demonstrating that, for a random variable Z distributed according to F_j, $\lim\min_{x\in\hat{S}_j} \rho(x, z)/r(x) \to c_j < 1$ a.s. and $\lim\min_{x\in\hat{S}_{1-j}} \rho(x, z)/r(x) \to c_{1-j} \geq 1$ a.s.

Theorem 2 *Assume that the training data are i.i.d. $F = \pi_0 F_0 + (1-\pi_0)F_1$ with $0 \leq \pi_0 \leq 1$, and that the class-conditional distributions F_j are continuous, finite dimensional, compactly supported and strictly separable. Assume further that ρ is a distance function which satisfies the continuity condition (CC). Let $\hat{S}_j$ be dominating sets for cccds D_j with $\alpha_j = \beta_j = 0$ and $\tau_j \in (0, 1]$. Then*

g_{cccd} is consistent. That is, $L(g_{cccd}) \to L^* = 0$ as $|\mathcal{X}_0|, |\mathcal{X}_1| \to \infty$, where L^* is the Bayes optimal probability of misclassification.

Proof: Consider $Z \sim F_j$ for $j \in \{0,1\}$. The proof for Theorem 2 begins by noting that the continuity condition (CC) implies that $\mathcal{X}_j^* := \lim_{n\to\infty} \mathcal{X}_j$ is dense in $s(F_j)$ almost surely. Let $C(n)$ be such that $\mathcal{X}_j \subset C(n)$; e.g., $C(n) = C_j$ or $C(n) = C_{1-j}^c$ for any sequence of covers C_j and C_{1-j}. ($\mathcal{X}_j$, C_j, and C_{1-j} are implicitly indexed by n.) Define $C^* := \liminf C(n)$, and note that $\mathcal{X}_j \subset C^*$ and hence C^* is dense in $s(F_j)$. Then $Z \in \overline{C^*}$, the closure of C^*, for with probability one there exists a subsequence of $\mathcal{X}_j^*$, and hence a subsequence of C^*, converging to Z. Taking $C^* = \liminf C_{1-j}^c$ yields $P[\lim \min_{s\in\hat{S}_{1-j}} \rho(Z,s)/r_s \geq 1] = 1$. The proof is completed by taking $C^* = \liminf C_j$ and showing that, for the boundary ∂C^*, $P_j[\partial C^*] = 0$. This implies that $Z \in C^{*o}$, the interior of C^*, and $P[\lim \min_{s\in\hat{S}_j} \rho(Z,s)/r_s < 1] = 1$. Thus $g_n(Z) = j$ eventually and evermore.

The restrictions of continuous distributions, finite dimensionality, and compact support can be relaxed. For instance, for any atom a of F_j, $g_n(a) = j$ eventually and evermore with probability one so long as there are no shared atoms. However, the strictly separable assumption is key; for overlapping densities it may be the case that $P_j[\partial C^*] > 0$. Theorem 2 relies on the fact that strictly separable class-conditional densities imply $P_j[\partial C^*] = 0$. A more general result, such as universal consistency, requires a data-adaptive selection of the $\alpha_j = \alpha_j(x)$ and $\beta_j = \beta_j(x)$ and will be pursued elsewhere.

We investigate the performance of g_{cccd} through simulations and experiments in Section 4.

3.4 Multi-Class Problems

To address the multi-class classification problem, in which the class labels take values $j \in \{1, \cdots, J\}$ for $J > 2$, one can simply adapt the methodology presented above by building *cccd*s $D_1, \cdots, D_J$ by considering each class j in turn as the target class and $\cup_{j'\neq j}\mathcal{X}_{j'}$ as the non-target class. The class j cover is once again given by $C_j := \cup_{x\in\hat{S}_j} B(x, r(x;\beta_j,\tau_j))$. Note that, unless we add additional parameters, this model treats all non-target class observations the same. It is possible that when considering target class j, one would wish to treat non-target subclass j' different than non-target subclass j''. This could be accomplished by having different values for $\beta_{j,j'}$ and $\beta_{j,j''}$. This will not be pursued further here.

The preclassifier m in the multi-class problem is defined via

$$m(z) = [I\{z \in C_1\}, \cdots, I\{z \in C_J\}]'.$$

That is, $m(z)$ is a binary vector of length J, the j^{th} component of which indicates whether z is in the j^{th} cover; $||z||_1 = 1$ implies that a classification is made, while $||z||_1 = 0$ suggests that z be labelled as coming from none of the target classes and $||z||_1 > 1$ suggests that z be labelled as coming from more than one of the target classes.

Given the multi-class prototype sets $\hat{S}_j$ and the associated radii $r(x; \beta_j, \tau_j)$, the two-class classifiers g defined above can be applied, *mutatis mutandis*, to the multi-class case:

$$g_{cccd}(z) := \arg\min_{j} \min_{x \in \hat{S}_j} \rho(x, z)/r(x).$$

There are two canonical methods for addressing the multi-class classification problem in terms of (multiple) two-class problems: J two-class problems class j against class "not j", or $J\,(J-1)/2$ two-class problems class j against class j' (Friedman 1996; Hastie and Tibshirani 1998). The former can be approached as J versions of the multi-class approach described above, while the latter can be approached as $J\,(J-1)/2$ versions of the two-class approach described above. In either case, results from the multiple classifiers must be combined after the fact.

4. Simulations and Experiments

4.1 Example 1

In this first simulation example, we consider two-dimensional ($\mathcal{X} = \mathbb{R}^2$) two-class data with the Euclidean distance function as dissimilarity. The class-conditional training data $\mathcal{X}_j$ are *i.i.d.* bivariate normal, $N(\mu_j, I)$ where I is the 2×2 identity matrix. $\mathcal{X}_0$ and $\mathcal{X}_1$ are mutually independent. We set $\mu_0 = [0,0]'$ and $\mu_1 = [2,0]'$. We assume that the unlabelled observations Z come from a mixture of these two normals with equal priors: $Z \sim (1/2)N(\mu_0, I) + (1/2)N(\mu_1, I)$. The Bayes optimal classifier for this case is linear, with $L^* = 1 - \Phi(1) \approx 0.159$.

We vary the class-conditional training sample sizes, considering $n_0 = n_1 \in \{5, 10, 20, 50, 100\}$. Independent test samples of size 100 are used to estimate the probability of misclassification $L_{n_j}(g)$ for each Monte Carlo replication for two versions of the *cccd* classifier g_{cccd}, the asymptotically optimal Fisher's linear discriminant g_{LC}, and the nearest neighbor classifier g_{1NN}. While the nearest neighbor classifier is not consistent for this case, we include it as a benchmark for comparison because it is so commonly used and because its limiting performance is guaranteed to be less than $2L^*$. At each stage, 100 Monte Carlo replications are performed.

The performance for the four classifiers is plotted in Figure 2 as probability of misclassification against class-conditional training sample size. Standard errors for the curves are small compared to the separation between curves; i.e. performance differentials are statistically significant. L^* and $2L^*$ are plotted as horizontal lines. The two Xs curves are the *cccd* classifiers — "vanilla" g_{cccd} (with $\alpha_j = \beta_j = 0$ and $\tau = 1$) is the top Xs curve, and "optimized" g_{cccd} (with parameters chosen based on a separate training set) is the bottom Xs curve; g_{LC} is denoted in the figure by triangles; g_{1NN} is denoted in the figure by pluses. We see that, as expected, the linear classifier is best. The *cccd* classifiers out-perform the nearest neighbor classifier. The vanilla *cccd* classifier is slightly better than the nearest neighbor classifier, while optimizing over *cccd* parameters greatly improves the performance of g_{cccd}.

Figure 3 presents the relative complexity $c = (|\hat{S}_0| + |\hat{S}_1|)/(n_0 + n_1)$ of the two *cccd* classifiers applied to this example. While both reduce complexity as compared to the one nearest neighbor classifier ($c = 1$), the classifier which performs best in terms of probability of misclassification is the classifier which reduces complexity the most. Namely, the g_{cccd} with $\alpha_j, \beta_j > 0$. At each class-conditional sample size $n_j \in \{5, 10, 20, 50, 100\}$, the leftmost boxplot in each pair is for 100 Monte Carlo replications of "vanilla" g_{cccd} (top Xs curve in Figure 2) and the rightmost boxplot in each pair is for 100 Monte Carlo replications of "optimized" g_{cccd} (bottom Xs curve in Figure 2). Note that the linear classifier can be viewed as a particular ball classifier with $|\hat{S}_j| = 1$. Thus, for this simple example wherein the optimal complexity is minimal (the optimal discriminant is linear) the performance results (Figure 2) follow precisely the complexity ranking.

Figure 4 displays the two *cccd* models for a representative example with $n_j = 50$. The second row in the figure is for vanilla g_{cccd} with $\alpha_j = \beta_j = 0$, yielding $\hat{L} \approx 0.21$, and the bottom row is for optimized g_{cccd} yielding $\hat{L} \approx 0.16$. Here, it is apparent that the robustness to outliers and contamination provided by choosing $\alpha_j, \beta_j > 0$ is desirable; $\alpha_j = \beta_j = 0$ yields overfitting and an overly complex model.

Our conclusion is that, for this example, the semiparametric *cccd* approach performs significantly better than the simple nonparametric nearest neighbor classifier and can perform nearly as well as the optimal parametric classifier.

4.2 Example 2

In Example 2, we again consider two-dimensional ($\mathcal{X} = \mathbb{R}^2$) two class data with the Euclidean distance function. The class 0 data $\mathcal{X}_0$ are

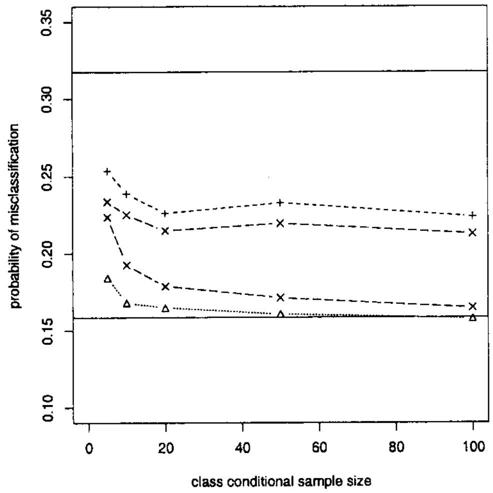

Figure 2. Classification performance curves for two versions of the *cccd* classifier for Example 1.

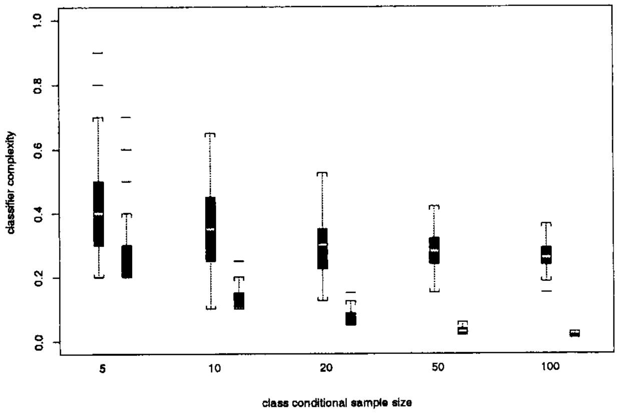

Figure 3. The relative complexity of the two *cccd* classifiers applied to Example 1.

$i.i.d.\ \mathcal{U}([0,1]^2)$ while the class 1 data $\mathcal{X}_1$ are $i.i.d.$

$$f(\cdot) = (1/2)\,\mathcal{U}(\cdot; B([0.25, 0.25]', r) + (1/2)\,\mathcal{U}(\cdot; B([0.6, 0.6]', 2r)$$

where $r \approx 0.113$ is chosen so that $L^* = 0.1$. (Again, $\mathcal{X}_0$ and $\mathcal{X}_1$ are mutually independent.)

An abridged version of the results for this case is given by considering class-conditional training sample sizes $n_0 = n_1 = 1000$: $\hat{L}_{1NN} \approx 0.169$, while $\hat{L}_{cccd} \approx 0.166$ for the "vanilla" version and $\hat{L}_{cccd} \approx 0.130$ for the "optimized" version. The "optimized" *cccd* eliminates more than 50% of the avoidable 1NN error: $\hat{L}_{cccd} \approx L^* + (3/7)\ (\hat{L}_{1NN} - L^*)$. In addition, the superiority of optimized g_{cccd} compared to both k-nearest neighbors optimized over k (Devroye, Gyorfi, and Lugosi 1996) and support vector machines with linear, polynomial, or radial kernels (Joachims 1999; Vapnik 1995) is statistically significant. The linear classifier, of course, does not perform well.

The relative complexity for optimized g_{cccd} for this case is $c \approx 1/10$. That is, $|\hat{S}_0| + |\hat{S}_1| \approx 200$. However, $|\hat{S}_0| \approx 12|\hat{S}_1|$. This is as expected, since class 1 is easy to model with a mixture of balls, while class 0 is not.

Higher dimensional versions of Example 2 are investigated in DeVinney (2003) with the result that the class cover catch digraph classifier continues to be competitive.

4.3 Example 3: Mine Classification

Minefield detection and localization is an important problem receiving much attention in the engineering and scientific literature. The Coastal Battlefield Reconnaissance and Analysis (COBRA) Program has as its goal the development of detection technologies for mines and minefields (Smith 1995; Witherspoon et al. 1995). Data collected under this program has been made available by NSWC Coastal Systems Station, Dahlgren Division, Panama City, Florida, to aid in the analysis of algorithms and approaches. The observations are detections of mines and minelike targets obtained from an unmanned aerial vehicle via multispectral sensors. An operational imperative imposed on the detector to find nearly all true mines implies that the number of false detections may be relatively high.

We denote true mines as class 0 and false detections as class 1. For the data set considered here, $n = n_0 + n_1 = 39$ with $n_0 = 12$, $n_1 = 27$. The original multispectral data set consists of 6-dimensional imagery. Priebe, Pang and Olson (1999) and Olson, Pang and Priebe (2001+) demonstrate that dimensions 3 and 5 are the most valuable for the classification task at hand. For this example, therefore, we consider each observation $x_i \in \mathbb{R}^2$.

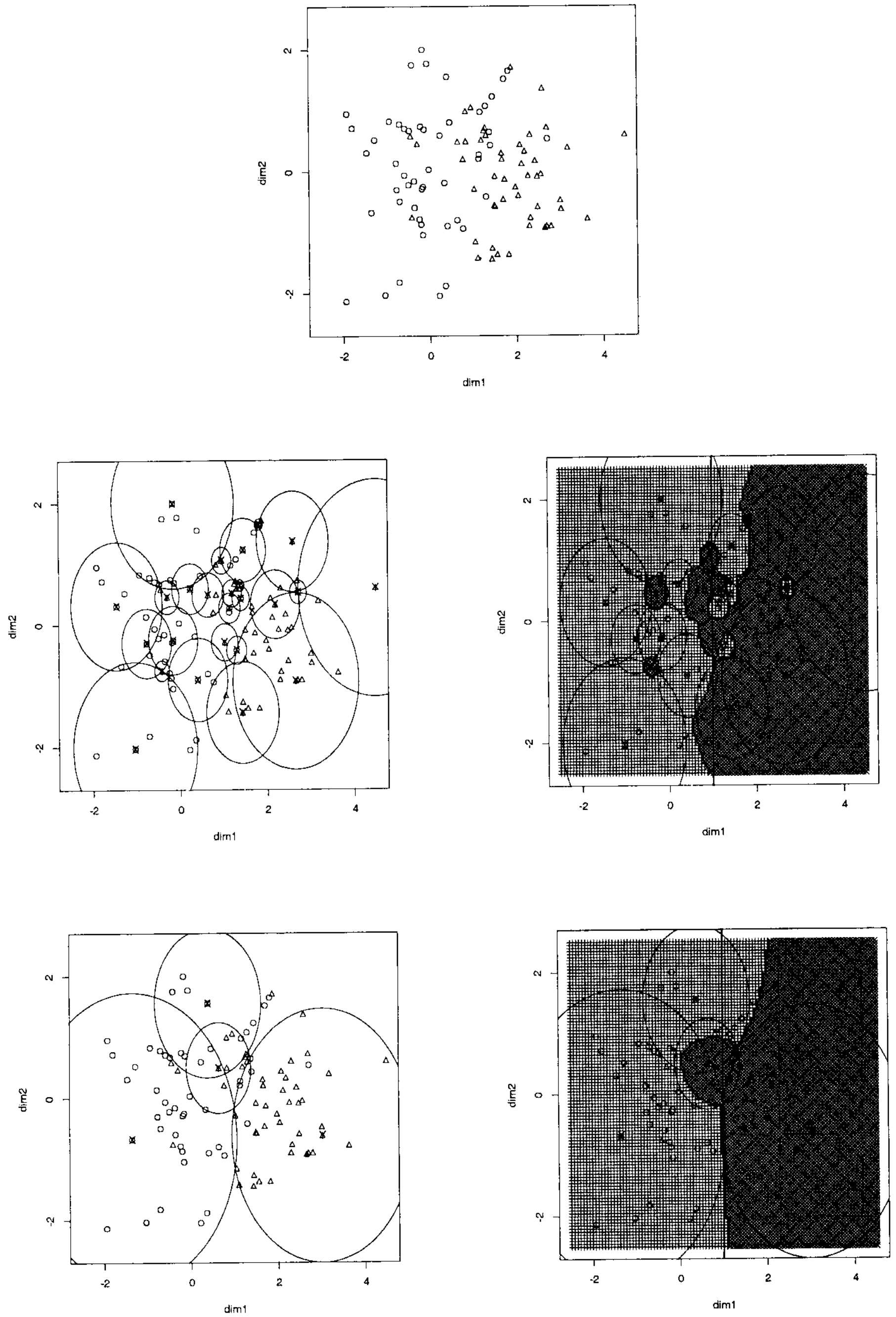

Figure 4. Two *cccd* models for a representative example from Example 1 with $n_j = 50$. The second row is for vanilla g_{cccd} with $\alpha_j = \beta_j = 0$, and the bottom row is for optimized g_{cccd}. The horizontal line at $dim1 = 1$ is the Bayes optimal discriminant surface

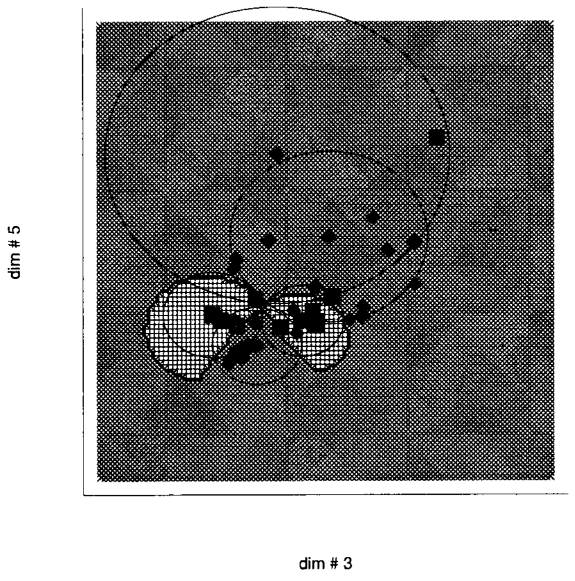

Figure 5. Discriminant regions produced by the *cccd* classifier applied to the COBRA mine data. True mines are black filled squares, false detections are gray filled diamonds. The true mine discriminant region is the lightly shaded region.

Figure 5 displays the discriminant regions produced by g_{cccd} with $(\alpha_0, \beta_0) = (4, 1)$, $(\alpha_1, \beta_1) = (2, 2)$, and $\tau_j = 1/2$ for both classes. The complexity for this *cccd* model is $|\hat{S}_0| = 2$ and $|\hat{S}_1| = 3$ and classification performance is $\hat{L}^{(R)}(g_{cccd}) = 7/39 \approx 0.179$ with $\hat{L}^{(D)}(g_{cccd}) = 8/39 \approx 0.205$. ($\hat{L}^{(D)}$ is the *deleted*, or leave-one-out, error rate estimate (Devroye, Gyorfi and Lugosi 1996). This performance is in fact superior to competing approaches. For instance, k-nearest neighbors optimized over k (Devroye, Gyorfi and Lugosi 1996) yields $\hat{L}^{(D)}(g_{knn}) = 10/39$, and support vector machines with linear, polynomial, or radial kernels (Joachims 1999; Vapnik 1995) yield $\hat{L}^{(D)}(g_{svm}) \geq 12/39$. (None of the classifiers perform as well in the original six-dimensional space; the dimensionality reduction improves performance for this example.)

Figure 6 displays the ROC curve—$\hat{P}[g(Z) = 0|Z \sim F_0]$ vs. $\hat{P}[g(Z) = 1|Z \sim F_1]$—for g_{cccd} on this data set. To obtain this ROC curve we replace the radii $\{r(x) : x \in \hat{S}_j\}$ with $r'(x) = r(x) \cdot t$ for $x \in \hat{S}_0$ and $r'(x) = r(x)/t$ for $x \in \hat{S}_1$, and allow t to range in $(0, \infty)$. This allows us to sweep out a

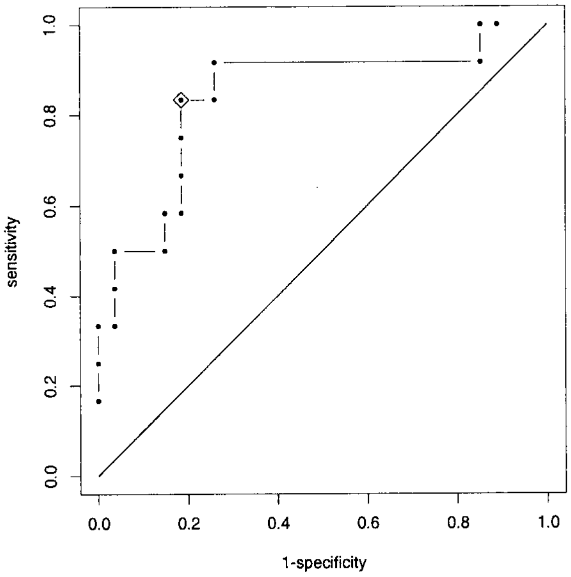

Figure 6. ROC curve for the *cccd* classifier applied to the COBRA mine data. Sensitivity $:= P[g(Z) = 0|Z \sim F_0]$ and specificity $:= P[g(Z) = 1|Z \sim F_1]$, where class 0 is true mines. The original *cccd* ($t = 1$) is marked with a diamond.

performance curve ranging from $P[g(Z) = 0|Z \sim F_0] = 1$ for $t \uparrow \infty$ to $P[g(Z) = 1|Z \sim F_1] = 1$ for $t \downarrow 0$. Choosing $t = 1$ yields the result reported above, $\hat{L}^{(R)}(g_{cccd}) = 7/39$, in which $P[g(Z) = 0|Z \sim F_0] = 10/12$ and $P[g(Z) = 1|Z \sim F_1] = 22/27$. We see from the ROC curve that correctly classifying one of the two misclassified mines can be accomplished with a relatively modest increase (from 5 of 27 to 7 of 27) in the number of false detections incorrectly classified as mines, while correctly classifying the second of the two misclassified mines is accomplished only at the expense of an enormous additional increase (from 7 of 27 to 23 of 27) in the number of false detections incorrectly classified as mines. We conclude that *cccd* classification can reduce the false detection rate from 27 to 7, while eliminating just one true mine. This can be of significant value as a preprocessor prior to spatial point pattern analysis aimed at minefield detection and localization (Priebe, Naiman and Cope 2001).

The approach described above is not the only — nor even necessarily the best — way to obtain ROC curves for *cccd* classifiers. This is a simple approach which both satisfies ROC requirements and does not require building a separate *cccd* model for each choice of ROC parameter.

5. Discussion

The classification methodology we propose herein is semiparametric. The data-adaptive choice of the sets $\hat{S}_j$ determines the complexity of the mixture-of-balls class covers, and hence the complexity of the classifier. This complexity is unbounded as sample size increases, but grows slowly with sample size when appropriate.

A major stumbling block to the use of g_{cccd} is the choice of parameters. In addition to providing a reduction in model complexity, the interpretation of α_j and β_j as robustness parameters provides some guidance. As with, for instance, the trimmed mean or the minimum volume ellipse, choosing $\alpha_j > 0$ provides benefit in terms of breakdown point by allowing the model to ignore outliers which degrade performance. Another approach to building in robustness to outliers is to consider α_j a threshold for individual iterations in the greedy dominating set selection algorithm rather than a total number of target class observations to leave uncovered. We may choose to halt the greedy algorithm when $\arg\max_{v \in V_j \setminus S_j^{t-1}} |\bar{N}(v) \setminus \cup_{v' \in S_j^{t-1}} \bar{N}(v')| \leq \alpha_j$. While this implementation of α_j provides no bound on the total number of uncovered target class observations — in particular, we may have $\hat{S}_j = \emptyset$ — it does have the effect of eliminating balls covering no more than α_j otherwise-uncovered observations — singletons in the case $\alpha_j = 1$ — from the model.

The parameter β_j is more problematic. *Each* target class observation's ball will cover β_j non-target class observation (when $\tau_j = 1$). It is clear that this is undesirable behavior, and that the number of non-target class observations covered should be a function of how valuable it is to cover them; i.e., how many additional target class observations are covered as a result of covering the non-target class observations. We can achieve this effect to some degree by letting $\tau_j < 1$. More generally, the desired behavior can be modeled for target class observation $x \in \mathcal{X}_j$ by considering $|B(x,r) \cap \mathcal{X}_j| - |B(x,r) \cap \mathcal{X}_{1-j}|$ as a function of the radius r. This is pursued in detail in DeVinney et al. (2002).

The selection of $\tau = 1/2$ can be seen as analogous to "maximizing the margin" in large-margin classifiers, however, $\tau \approx 0$ appears to provide superior results in practice. In a slightly different context, Marchette and Priebe (2003) provides guidance for automating the selection of these parameters. A major focus of our continuing efforts involves developing a methodology for the data-adaptive selection of these parameters. This is the subject of DeVinney et al. (2002).

Another approach to improving the basic model is to recognize that "not all balls are created equal." Consider, for example, the *fitness statistics* T:

$$T(x) = \max\{|\mathcal{X}_j \cap B(x, r(x))| - |\mathcal{X}_{1-j} \cap B(x, r(x))|, 1\}$$

for $x \in \hat{S}_j$, which specifies the fitness of a ball to be the difference of the number of target class observations and the number of non-target class observations in the ball, thresholded so that $T \geq 1$. We place more stock in balls for which T is larger. This extension yields the classifer given by $g(z) := I\{\min_{x \in \hat{S}_1}(\rho(x,z)/r(x))^{T(x)} < \min_{x \in \hat{S}_0}(\rho(x,z)/r(x))^{T(x)}\}$, which remains consistent with the preclassifier m (and Theorems 1 and 2 hold) as long as $T(x) \geq 1$ for all x. For instance, we find that the error for the "optimized" version of g_{cccd} decreases from 0.130 to 0.122 for the case presented as Example 2 in Section 4.2 above, when using the classifier which takes into account the value of T. This improvement is statistically significant.

A note about priors. The *cccd* model as presented assumes that the training sample sizes are representative of the true prior class membership probabilities. If this is not the case, knowledge of the priors must be assumed. Our methodology can be adapted to this latter case by altering the counting of non-target class observations in the specification of the proximity regions (the ball radii). For instance, if the non-target class $1 - j$ prior is smaller than the empirical prior $n_{1-j}/(n_{1-j} + n_j)$, each non-target class observation should count as less than one when choosing the radius for target class observation x to cover "as many target class observations as possible while covering at most β_j non-target class observations" (see Section 2.1).

Finally, we note that our methodology is quite general. The classifier g_{cccd} requires only a dissimilarity matrix and class labels for J-class training data. Therefore, any reasonable approach to determining dissimilarities for categorical and/or missing data will allow classification using class cover catch digraphs.

References

ARORA, S., and LUND, C. (1997), *Hardness of approximations.* PWS Publishing Company.

CHVATAL, V. (1979), "A greedy heuristic for the set-covering problem," *Mathematics of Operations Research*, 4:233-235.

COVER, T.M., and HART, P.E. (1967), "Nearest Neighbor Pattern Classification," *IEEE Transactions on Information Theory*, 13, 21-27.

DASRATHY, B.V., and SÁNCHEZ, J.S. (2000), "Tandem fusion of nearest neighbor editing and condensing algorithms – data dimensionality effects," *Proceedings of the 15th International Conference on Pattern Recognition*, Volume 2, 692-695.

DEVINNEY, J.G., PRIEBE, C.E., MARCHETTE, D.J., and SOCOLINSKY, D.A. (2002), "Random Walks and Catch Digraphs in Classification," *Computing Science and Statistics*, 34, to appear.

DEVINNEY, J.G., and PRIEBE, C.E. (2001+), "Class cover catch digraphs," submitted for publication.

DEVINNEY, J.G. (2003), "The Class Cover Problem and its Applications in Pattern Recognition," Ph.D. Thesis, Department of Mathematical Sciences, Johns Hopkins University, Baltimore, MD.

DEVROYE, L., GYORFI, L., and LUGOSI, G. (1996), *A Probabilistic Theory of Pattern Recognition.* Springer.

DUDA, R.O., HART, P.E., and STORK, D.G. (2001), *Pattern Classification*, 2nd Edition, Wiley.

FRIEDMAN, J.H. (1996), "Another Approach to Polychotomous Classification," Stanford University Technical Report (unpublished).

HARTIGAN, J.A. (1975), *Clustering Algorithms.* Wiley.

HARTIGAN, J.A., and WONG, M.A. (1979), "A k-means clustering algorithm," *Applied Statistics*, 28, 100-108.

HASTIE, T., and TIBSHIRANI, R. (1998), "Classification by Pairwise Coupling," *Annals of Statistics*, 26, 2, 451-471.

HO, T.K., and BASU, M. (2002), "Complexity Measures of Supervised Classification Problems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24, 3, 289-300.

JAIN, A.K., Mao, J., and HOHLUDDIN, K.M. (1996), "Artificial Neural Networks: A Tutorial," *IEEE Computer*, March, 31-44.

JOACHIMS, T. (1999) in: Making large-Scale SVM Learning Practical. Advances in Kernel Methods – Support Vector Learning, B. Schlkopf and C. Burges and A. Smola (ed.), MIT Press.

KARP, R. (1972), *Reducibility among combinatorial problems.* Plenum Press.

KAROŃSKI, M. SCHEINERMAN, E.R., and SINGER-COHEN, K.B. (1999), "On random intersection graphs: the subgraph problem," *Combinatorics, Probability and Computing*, 8, 131-159.

KULKARNI, S.R., LUGOSI, G., and VENKATESH, S. (1998), "Learning Pattern Classification – A Survey," *IEEE Transactions on Information Theory*, 44, 6, 2178-2206.

LEBOURGEOIS, F., and EMPTOZ, H. (1996), "Pretopological Approach for Supervised Learning," *Proceedings of the 13th International Conference on Pattern Recognition*, 256-260.

MAA, J.-F., PEARL, D.K., and BARTOSZYNSKI, R. (1996), "Reducing multidimensional two-sample data to one-dimensional interpoint comparisons," *Annals of Statistics* 24, 1069-1074.

MAEHARA, H. (1984), "A digraph represented by a family of boxes or spheres," *Journal of Graph Theory* 8, 431-439.

MARCHETTE, D.J., and PRIEBE, C.E. (2003) "Characterizing the Scale Dimension of a high dimensional classification problem," *Pattern Recognition*, 36, 1, 45-60.

OLSON, T., PANG, J.S., and PRIEBE, C.E. (2001+), "A Likelihood-MPEC Approach to Target Classification," *Mathematical Programming*, to appear.

PAREKH, A.K. (1991), "Analysis of a greedy heuristic for finding small dominating sets in graphs," *Information Processing Letters*, 39:237-240.

PRIEBE, C.E., DEVINNEY, J.G., and MARCHETTE, D.J. (2001), "On the distribution of the domination number of random class cover catch digraphs," *Statistics and Probability Letters*, 55, 3, 239-246.

PRIEBE, C.E., NAIMAN, D.Q., and COPE, L. (2001), "Importance Sampling for Spatial Scan Analysis: Computing Scan Statistic p-Values for Marked Point Processes," *Computational Statistics and Data Analysis*, 35, 4, 475-485.

PRIEBE, C.E., PANG, J.S., and OLSON, T. (1999), "Optimizing Sensor Fusion for Classification Performance," *Proceedings of the International Conference on Imaging Science, Systems, and Technology: CISST '99*, 397-403.

SCHOLKOPF, B., PLATT, J., SHAWE-TAYLOR, J., SMOLA, A.J., and WILLIAMSON, R.C. (2001), "Estimating the support of a high-dimensional distribution," *Neural Computation*, 13, 7.

SKALAK, D.B. (1997), "Prototype Selection for Composite Nearest Neighbor Classifiers," Ph.D. Thesis, Department of Computer Science, University of Massachusetts, Amherst, MA.

SMITH, D.L. (1995), "Detection Technologies for Mines and Minelike Targets," *SPIE Volume 2496: Detection Technologies for Mines and Minelike Targets*, 404-408.

SOCOLINSKY, D.A., NEUHEISEL, J.D., PRIEBE, C.E., MARCHETTE, D.J., and DEVINNEY, J.G. (2003), "Fast Face Detection with a Boosted CCCD Classifier," submitted for publication. (Available as Technical Report No. 634, Department of Mathematical Sciences, Johns Hopkins University, Baltimore, MD 21218-2682.)

VAPNIK, V.N. (1995), *The Nature of Statistical Learning Theory.* Springer.

WITHERSPOON, N.H., HOLLOWAY, J.H., DAVIS, K.S., MILLER, R.W., and DUBEY, A.C. (1995), "The Coastal Battlefield Reconnaissance and Analysis (COBRA) Program for Minefield Detection," *SPIE Volume 2496: Detection Technologies for Mines and Minelike Targets*, 500-508.