Random Walks and Catch Digraphs in Classification

Jason DeVinney[*], Carey Priebe[†], Dave Marchette[‡], Diego Socolinsky[§]

July 1, 2002

Abstract

We present a new application of the class cover problem to statistical pattern classification. The goal of the class cover problem is to find a set of covering balls to cover one class of points while not covering another class of points. We present an adaptive method for choosing the radii of the covering balls. The purpose of the method is toward improved classifier performance.

1 Introduction

Throughout this paper, we will consider classification in the two-class case. We are given as *training data* two finite, non-empty sets of class conditional $\mathcal{X}$ valued observations, $\mathcal{X}_0$ and $\mathcal{X}_1$. We assume that our data come from a dissimilarity space, with base set $\mathcal{X}$ and dissimilarity measure $\rho : \mathcal{X} \times \mathcal{X} \to \Re^+$. Recall that a dissimilarity measure, ρ must obey $0 = \rho(x,x) < \rho(x,y) = \rho(y,x)$ for $x \neq y$ [2]. Our goal is to design a classifier $g_{(\mathcal{X}_0,\mathcal{X}_1)} : \mathcal{X} \to \{0,1\}$, such that given an unlabeled observation Z with true but unknown class label Y in $\{0,1\}$, the probability of misclassification $L(g) = P[g_{(\mathcal{X}_0,\mathcal{X}_1)}(Z) \neq Y]$ is close to Bayes optimal L^*. See for instance [5, 11].

Our methodology stems from a variation of the class cover problem (CCP). We begin with a dissimilarity space $(\mathcal{X}, \rho)$ and two finite, non-empty sets, $\mathcal{X}_0, \mathcal{X}_1 \subset \mathcal{X}$. We refer to $\mathcal{X}_0$ as the *target* class and $\mathcal{X}_1$ as the *non-target* class. The most general CCP is to find a minimum cardinality set of covering balls B_i, with center c_i and radius r_i ($B_i = \{x \in \mathcal{X} : \rho(x, c_i) < r_i\}$), whose union contains all of the target class and does not contain any of the non-target class. Restrictions on c_i and r_i give rise to an interesting family of class cover problems. Cannon and Cowen introduced the CCP [1] and study a version where the center of each ball must be an element in the target class ($c_i \in \mathcal{X}_0$) and the radii, r_i, must be the same for all values of i.

The basis for our classifier is a CCP with a simple relation to directed graphs [3, 6, 8, 9, 10]. This *constrained* CCP allows the radii to differ, but the centers must belong to the target class. At each point $x \in \mathcal{X}_0$ we define a *covering ball for* x, $B_x = \{z \in \mathcal{X} : \rho(z,x) < r_x\}$ where $r_x = \rho(x, \mathcal{X}_1) = \min_{z \in \mathcal{X}_1} \rho(x,z)$. We define a *cover*, C, for $\mathcal{X}_0$ as a collection of covering balls such that $\mathcal{X}_0 \subseteq \cup_{B \in C} B$. The goal of the constrained CCP is to find a minimum cardinality cover.

[*]Mathematical Sciences Department, Johns Hopkins University, Baltimore, MD 21218

[†]Mathematical Sciences Department, Johns Hopkins University, Baltimore, MD 21218

[‡]Naval Surface Warfare Center, B10, Dahlgren, VA 22448

[§]Equinox Corporation, 207 East Redwood St., Suite 1005, Baltimore, MD 21202

A more general CCP can be achieved by relaxing the constraint that all target class points must be in the cover and no non-target class points may be covered. A cover which does not contain any non-target class points is called *pure* and a cover which contains all target class points is called *proper*. A constrained CCP with allowances for impure and improper covers was first studied in [9] and is called the α, β CCP. We can further generalize the CCP by minimizing some function of the cover rather than the size of the cover. We will apply both of these ideas to create a new CCP classifier.

When applying the CCP to classification, we use a generalization of the reduced nearest neighbor classifier [12] as a framework. The idea is to find a cover (independently) for both classes by choosing one to be the target class, solving the CCP, and then switching the roles of target and non target class and solving the new CCP. We must also choose a *cover-dissimilarity* function d between new observations and a cover. The classifier is then

$$g(z) = \begin{cases} 0 & \text{if } d(z, C_0) < d(z, C_1), \\ 1 & \text{if } d(z, C_1) < d(z, C_0), \\ -1 & \text{otherwise} \end{cases} \tag{1}$$

where C_0, C_1 are the covers of class zero and one respectively and an output of -1 represents no decision.

The goal of any classifier is to correctly identify the regions (discriminant regions) where the class conditional density function for one class is larger than that of the other class. The goal of our cover is to estimate these regions as accurately and efficiently as possible with the given training data. It is therefore important that the function d we use to measure distance to a cover has the following property for two covers C_0, C_1

$$z \in C_0 \cap C_1^c \Rightarrow d(z, C_0) < d(z, C_1). \tag{2}$$

Priebe et al. [9] investigate applications of the constrained CCP using impure and improper covers. We will review this method in sections 2.1 and 2.2. The focus of this paper extends that work with data adaptive allowances for impureness and improperness. This is done by observing the local neighborhood while choosing the radius for the potential covering ball. This method is explained in detail in section 2.3. We also modify the way in which covers are chosen from previous methods. We will finally present some results and an analysis of performance.

2 Classifiers

In this section we review the details for applying some variations of the constrained CCP to classification. We also present our new adaptive methodology. As before, we are given training sets $\mathcal{X}_0, \mathcal{X}_1 \subseteq \mathcal{X}$.

2.1 Preclassifier

A naive classifier is built using a pure and proper cover from each class. By switching the role of target class between $\mathcal{X}_0$ and $\mathcal{X}_1$, two different instances of the CCP can be solved, resulting in two covers C_0 and C_1. For a cover C we define a simple cover-dissimilarity function as

$$d_N(z, C) = \begin{cases} 0 & \text{if } z \in \bigcup_{B \in C} B \\ 1 & \text{otherwise} \end{cases} \tag{3}$$

Given two covers C_0, C_1, the above cover-dissimilarity function in the nearest neighbor framework of (1) creates the following simple classifier $g : \mathcal{X} \rightarrow \{-1, 0, 1\}$ where,

$$g_{pre}(z) = \begin{cases} 0 & z \in C_0 \cap C_1^c \\ 1 & z \in C_1 \cap C_0^c \\ -1 & \text{otherwise} \end{cases} \tag{4}$$

One drawback of this classifier is that it makes no decision for some points in $\mathcal{X}$. We can remedy this with a scaled cover-dissimilarity function. For a cover C made up of balls $\{B_i\}$ with centers c_i and radius r_i, we define a new cover-dissimilarity function as

$$d_S(z, C) = \min_{\{i : B_i \in C\}} \frac{\rho(z, c_i)}{r_i} \tag{5}$$

Both dissimilarity functions in (3) and (5) have the property in (2).

Another drawback of the pre-classifier is its tendency to overfit. When trying to approximate the discriminant region for one class (finding a cover), it would be better to allow our covering balls to contain a few "contaminating" points from the other class. Presumably, these points should be outliers of that class. We might also allow our cover to miss a few target class points. Ideally, these points would also be outliers and we would not want to consider their covering ball as part of the cover. We begin to address these observations with the α, β CCP.

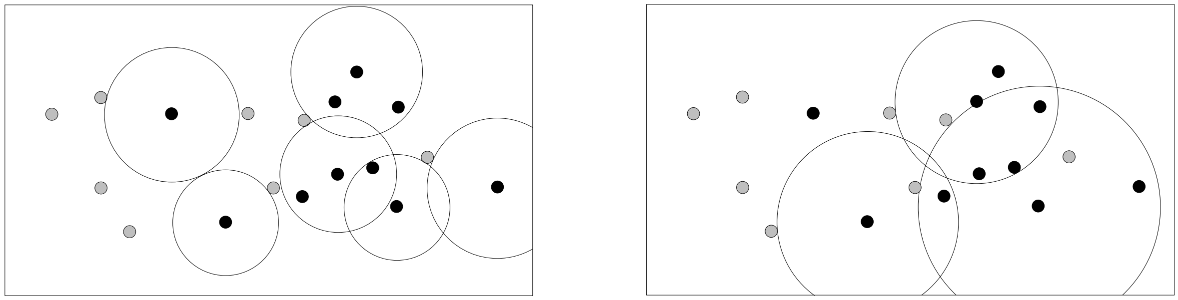
2.2 α, β CCP

In this section we review the CCP in [9] which involves impure and improper covers. Let α and β be nonnegative integers. Without loss of generality, in this section we will assume that $\mathcal{X}_0$ is the target class. Ideally we would like to require our cover for $\mathcal{X}_0$ to miss at most α points from $\mathcal{X}_0$ and to contain at most β points from $\mathcal{X}_1$. This turns out to be an extremely hard decision problem if $\beta > 0$ because now we must consider $\beta + 1$ balls centered at each target class point and find a smallest subset of balls (a cover) so that their union contains at least $|\mathcal{X}_0 - \alpha|$ target class points and at most β non-target class points. We simplify the situation by redefining β to be the number of non-target class points in each covering ball. We define the covering ball B_i^β at each target class point $x_i \in \mathcal{X}_0$ as $\{x \in \mathcal{X} : \rho(x_i, x) < \rho_\beta(x_i, \mathcal{X}_1)\}$ where $\rho_\beta(x_i, \mathcal{X}_1)$ is the $\beta + 1$th smallest distance from x_i to elements in $\mathcal{X}_1$. Now a cover for the target class is the smallest collection of balls B_i^β such that at least $|\mathcal{X}_0| - \alpha$ target class points are covered. Clearly we can achieve a pure and proper cover by setting $\alpha = \beta = 0$. Below (Figure 1) are illustrations of covers with various values of α and β.

While this technique greatly improves on the pre-classifier, its assumption that every covering ball should contain β non-target class points is suboptimal. What we would like is some way to let each covering ball determine its own radius based on its local neighborhood. For example, if we increase the radius of a covering ball and the result is to capture one new non-target class point and ten new target class points, then we might say that change in radius is worthwhile. We attempt to formalize this idea in the next section.

2.3 Random Walk CCP

We propose a new adaptive strategy for choosing the radii for covering balls. Our intent is to have the same effect as the α and β parameters (sensitivity to contamination and outliers),



(a) $\alpha = 0$ $\beta = 0$

(b) $\alpha = 1$ $\beta = 1$

Figure 1: Figure 1(a) shows a pure and proper cover of the black points. Figure 1(b) shows a cover with $\alpha = 1$ $\beta = 1$.

while behaving in a more local manner. Instead of choosing global parameters α and β, we allow each ball to choose its own radius based on the local density of target and non target class points. We will also choose our covers in a slightly different way. Instead of choosing a minimum cardinality cover, we will choose a cover which minimizes some property of the cover.

2.3.1 Choosing Radii

For each point x_i in the target class $\mathcal{X}_0$, we will examine a random walk R_{x_i} which is defined as follows. For any nonnegative $r \in \mathbb{R}$ let

$$R_{x_i}(r) = |\{x \in \mathcal{X}_0 : \rho(x_i, x) \leq r\}| - |\{x \in \mathcal{X}_1 : \rho(x_i, x) \leq r\}|.$$

A way of visualizing this as a random walk is to think of a ball of radius r centered at x_i. As r increases from zero, the ball will encounter points from $\mathcal{X}_0$ and $\mathcal{X}_1$. Each time the ball encounters a target class point or non-target class point, the random walk goes up by one or down by one respectively. See Figure 2 for an illustration of this. A large positive value of R_{x_i} indicates a high local density of target class points in some ball around x_i. If there are an unequal number of target class and non-target class points (consider unequal priors on the data) then we change the definition of the random walk. Suppose $|\mathcal{X}_0| = n_0$ and $|\mathcal{X}_1| = n_1$. A more general definition for the random walk is

$$R_{x_i}(r) = \frac{n_1}{n_0}|\{x \in \mathcal{X}_0 : \rho(x_i, x) \leq r\}| - |\{x \in \mathcal{X}_1 : \rho(x_i, x) \leq r\}|.$$

Once we have the random walk for some target class point x_i, we will use it to choose a radius for the ball B_i. But how shall we do this? We will argue for one possible way. Before we can do so, we must understand the goal or purpose of each individual covering ball. What follows here is a rather casual discussion designed to understand the intuition behind our methodology. Let us suppose that the training data are independent observations drawn from the class conditional distributions F_0 and F_1 (with densities f_0 and f_1). Let $D_i = \{x \in \mathcal{X} : f_i(x) > f_{1-i}(x)\}$ be the discriminant region for class $i \in \{0, 1\}$. To approximate D_i, each covering ball B_j should be the largest ball centered at x_j such that $B_j \cap D_{1-i} = \emptyset$. Because

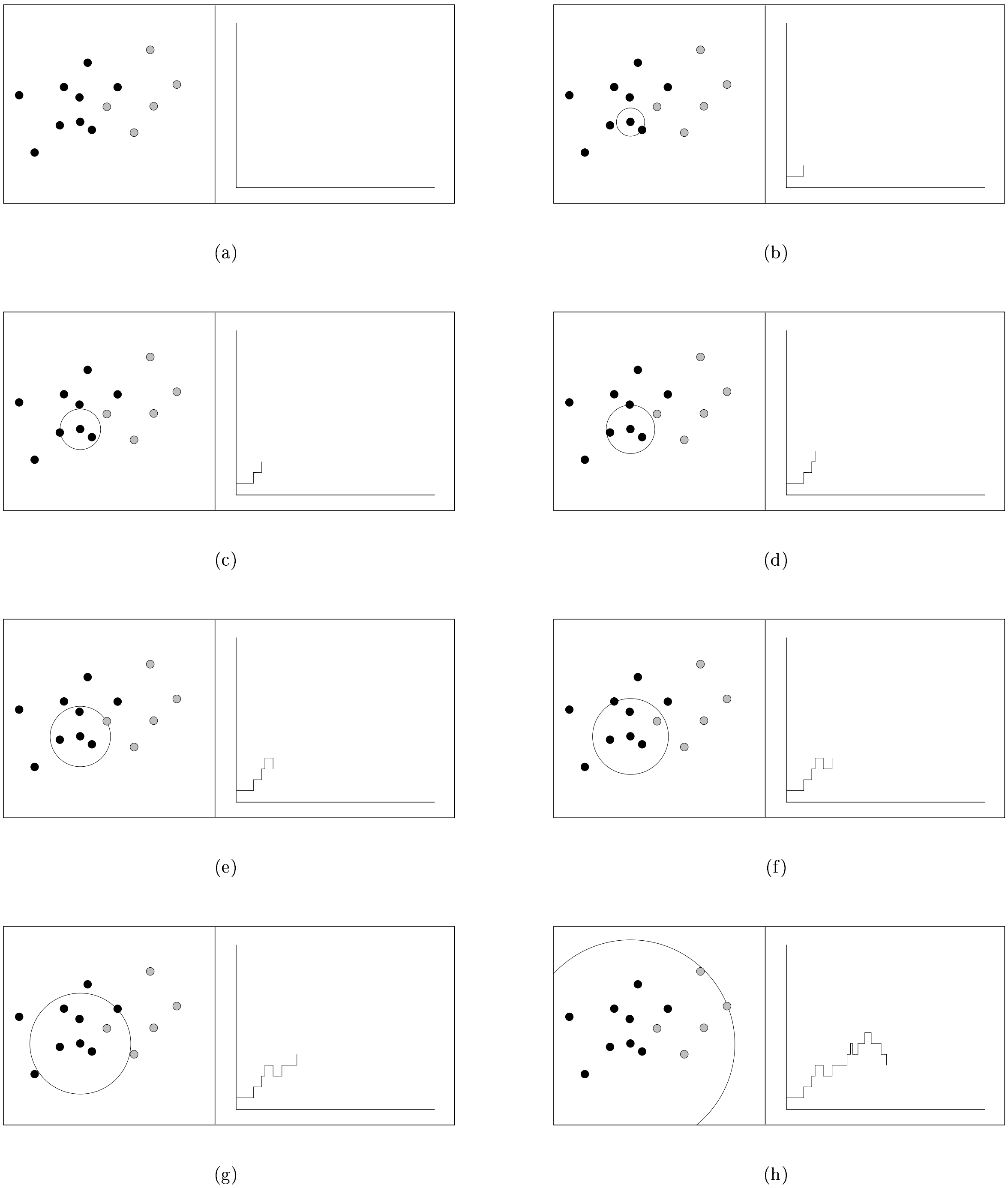


Figure 2: Snapshots of a random walk.

our training samples are finite, it is impossible to determine the exact largest radius for B_j so that $B_j \cap D_{1-i} = \emptyset$. We will instead attempt to find a radius $r^*_{x_j}$ for a point x_j that is as large as possible and has high probability that $B_j \cap D_{1-i} = \emptyset$. In this way, our cover will become an approximation of the discriminant region D_i for each class.

Consider a point $x \in \mathcal{X}_0$. We will use $f_1(z) \geq f_0(z)$ as the null hypothesis for all $z \in \mathcal{X}$ in a small region around x. If we reject the null hypothesis, that is, we see evidence that $f_1(z) < f_0(z)$ in some region around x, then we would like to put a covering ball of positive radius around x. Suppose x is in a region where $f_1(z) < f_0(z)$ locally. Then, with high probability $R_x(r)$ will increase for $r \in (0, r_m)$ for some $r_m > 0$. We determine the radius, r^*_x, for a covering ball centered at x with the following formula

$$r^*_x = \arg\max_r R_x(r) - P(r)$$

where $P(r)$ is an increasing penalty function that biases toward choosing smaller radii. The choice of smaller radii as opposed to larger radii has two advantages; we can more accurately approximate the discriminant region with smaller balls, and our balls have a higher probability of lying completely in the target class' discriminant region.

2.3.2 Finding the Cover

Once the radii are chosen, we must find a cover. Ideally we would like to find the cover which maximizes the performance of our classifier. Because of the combinatorial explosion of possibilities we will greedily choose our cover. That is, we will choose our cover one ball at a time, each ball improving the current classifier as much as possible.

Instead of checking the performance of our classifier at each stage we will instead use a closely related surrogate test. To determine which ball to add next to the cover we will find the ball which most improves our preclassifier. That is we will favor balls with a high number of (as yet uncovered) target class points and a low number of non target class points. Because of the local nature of this methodology, we will impose a penalty function $p(r)$ that increases with radius. For the ball of radius r^*_x centered at x we assign a score $T_x = R_x(r^*_x) - p(r^*_x)$ and we choose the ball with maximum score. After a ball is added to a cover, any points covered by that ball are disregarded and we recompute radii for each uncovered point and choose a new ball to add to the cover based on newly computed scores. We continue adding balls in this way until all target class points are covered.

2.3.3 The Classifier

Once we have the cover for both classes, we define a distance function ρ to describe distance to a cover and then use the classifier as defined in equation (1) to perform the classification.

3 Results

3.1 Simulation Data

In this simulation we have F_0 =U([0,1]×[0,1]) and $F_1 = \frac{1}{2}$ U([0.1,0.55]×[0.1,0.55]) + $\frac{1}{2}$U([0.6,0.8]×[0.6,0.8]). Figure 3 shows 200 points drawn from F_0 as empty circles and 200 points drawn from F_1 as the black filled circles. We compared the performance of our CCP classifiers to the nearest neighbor classifier, the k-nearest neighbor (with k optimized for each value of n) classifier and support vector machines. We used the radial basis function kernel included in the SVM-light package to implement our support vector machines [4]. We created training sets of n observations from each class ($n \in \{50, 100, 200, 500\}$) and then performed Monte Carlo replicates on test sets of 100 observations from each class. For each replicate, the performance of a classifier was measured by the fraction of observations misclassified (as an approximation of the misclassification rate $L(g)$). We performed Monte Carlo replicates until the standard deviation for the average performance became less than 0.003. The experimental approximation of the misclassification rate for each classifier is shown in Table 1. The Bayes optimal error rate for this model is approximately 0.121. Notice that the CCP-based classifiers outperform the nearest neighbor and the optimized k-nearest neighbor classifiers. The two CCP classifiers have about the same performance on this data set, however one advantage of the random walk CCP classifier over the α, β classifier is reduced classifier complexity. Table 2 shows the average number of balls in a cover for each class for each value of n.

Also shown in Figure 3 is an illustration of the classification regions calculated by the top four performing classifiers for $n = 200$. The optimal classification regions are outlined in black.

Training Size	NN	k-NN	SVM	α, β CCP	RW-CCP
50	0.242	0.240	0.221	0.212	0.212
100	0.224	0.212	0.195	0.190	0.183
200	0.210	0.188	0.181	0.171	0.165
500	0.199	0.166	0.159	0.154	0.153

Table 1: Classifier performance on simulation data.

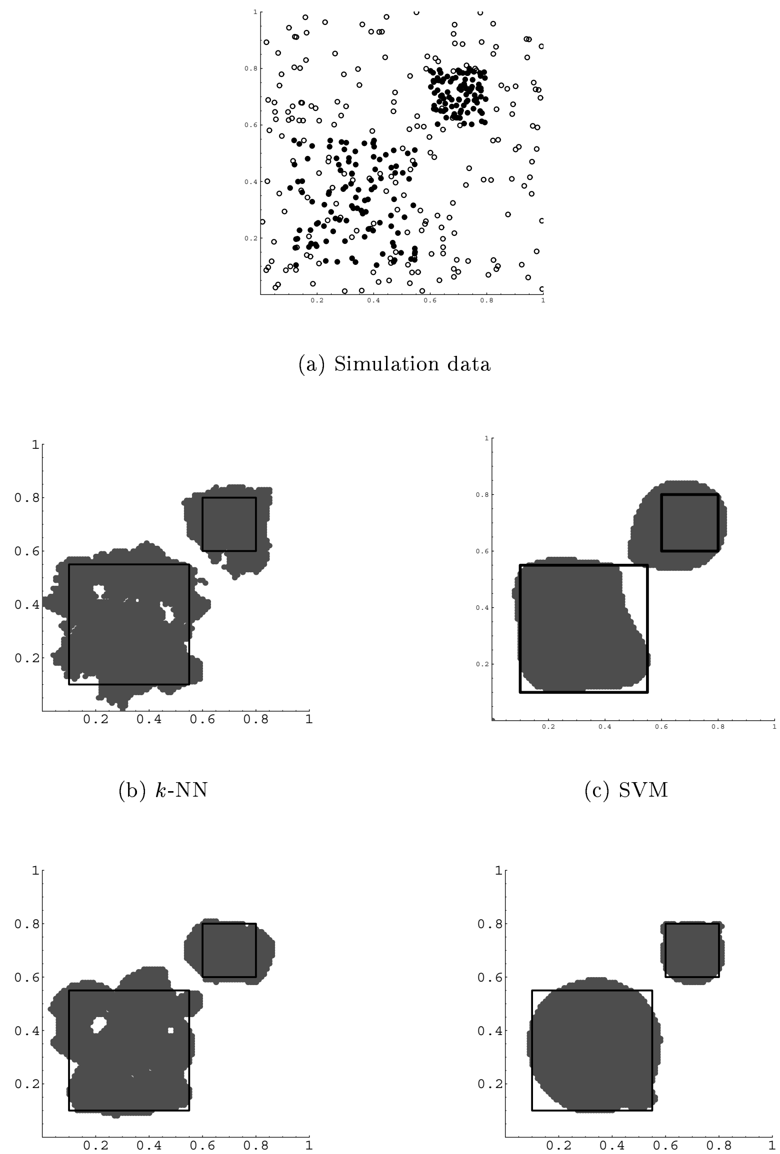
Training Size	α, β CCP		RW-CCP	
	class 1	class 0	class 1	class 0
50	6.7	16.3	2.8	5.3
100	9.3	28.2	3.4	8.4
200	13.4	49.6	4.6	13.0
500	27.3	110.1	5.9	19.2

Table 2: Average number of balls per cover.

3.2 Experimental Data

The experimental data set is multispectral data observations of minelike objects taken by an unmanned aerial vehicle as part of the Coastal Battlefield Reconnaissance and Analysis (COBRA) Program. There are 39 observations, of which 12 are actual mines and 27 are false alarms. The raw data is six dimensional (six spectra), but we consider the two dimensions most valuable to classification based on the work of Olson, Pang and Priebe [7].

Figure 4 shows the classification regions produced by the top four performing classifiers. Using the leave-one-out error rate estimate we observe that the random walk CCP and α, β CCP classifiers have the best performance of 9/39 and 8/39 incorrect respectively. The nearest neighbor, k-nearest neighbor and SVM classifiers classify 10/39 incorrectly or worse.



(a) Simulation data

(b) k-NN

(c) SVM

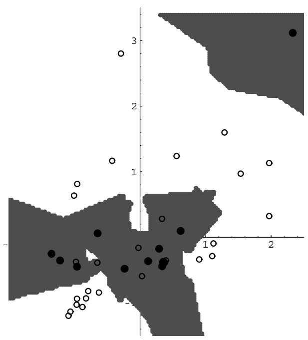
(d) α, β CCP

(e) RW-CCP

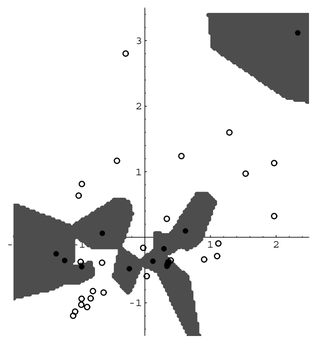
Figure 3: Comparison of classification regions for simulation data.

4 Conclusion

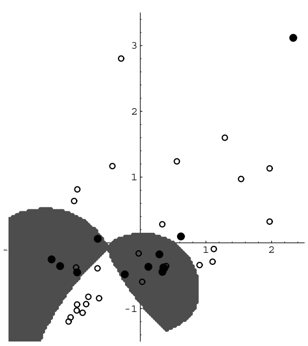
We have presented a new methodology for applying the class cover problem to classification. The classifier based on the random walk CCP was designed to improve on some of the weaknesses of previous CCP-based classifiers. The adaptive manner in which the radii of covering balls are chosen is the main strength of the new classifier. This classifier is being presented here as a proof of concept. Our strategy for determining the radius of a ball is a good first attempt, but we believe that a more robust method exists. Also, our method of finding a cover could be improved on by considering other factors such as statistical depth.



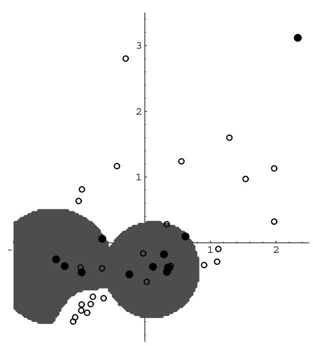
(a) NN



(b) k-NN



(c) α, β CCP



(d) RW-CCP

Figure 4: Comparison of classification regions for minefield data.

5 Acknowledgments

This work was partially supported by the Office of Naval Research Grant N00014-01-1-0395 and Defense Advanced Research Projects Agency Grant F49620-01-1-0395. Minefield data was provided by NSWC Coastal Systems Station, Dahlgren Division, Panama City, Florida with support from the United States Marine Corps Amphibious Warfare Technology. Special thanks to R.R. Muise.

References

[1] A. Cannon and L. Cowen. Approximations algorithms for the class cover problem. In *6th International Symposium on Artificial Intelligence and Mathematics, 2000*, 2000.

[2] F. Critchley and B. Fichet. *Lecture Notes in Statistics: Classification and Dissimilarity Analysis*, volume 93, chapter 2. Springer-Verlag, 1994.

[3] J. DeVinney and C. Priebe. Class cover catch digraphs. 2002. Submitted for Publication. Available as Technical Report No. 633, Department of Mathematical Sciences, Johns Hopkins University, Baltimore, MD 21218-2682.

[4] T. Joachims. *Making large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning.* MIT-Press, 1999.

[5] S. Kulkarni, G. Lugosi, and S. Venkatesh. Learning pattern classification - a survey. *IEEE Transactions on Information Theory*, 44(6):2178–2206, October 1998.

[6] D.J. Marchette and C.E. Priebe. Characterizing the scale dimension of a high dimensional classification problem. *Pattern Recognition*, to appear. Available as Technical Report No. 614, Department of Mathematical Sciences, Johns Hopkins University, Baltimore, MD 21218-2682.

[7] T. Olson, J.S. Pang, and C.E. Priebe. A likelihood–mpec approach to target classification. *Mathematical Programming*, to appear. Available as Technical Report No. 590, Department of Mathematical Sciences, Johns Hopkins University, Baltimore, MD 21218-2682.

[8] C. Priebe, J. DeVinney, and D. Marchette. On the distribution of the domination number for random class cover catch digraphs. *Stat. Probab. Lett.*, (55), 2002.

[9] C. Priebe, D. Marchette, J. DeVinney, and D. Socolinsky. Classification using class cover catch digraphs. 2002. Submitted for publication. Available as Technical Report No. 628, Department of Mathematical Sciences, Johns Hopkins University, Baltimore, MD 21218-2682.

[10] C.E. Priebe, J.L. Solka, D.J. Marchette, and B.T. Clark. Class cover catch digraphs for latent class discovery in gene expression monitoring by dna microarrays. *Computational Statistics and Data Analysis.* to appear.

[11] B.D. Ripley. *Pattern Recognition and Neural Networks*, chapter 6.3. Cambridge, 1996.

[12] D.B. Skalak. Prototype selection for composite nearest neighbor classifiers. Technical report, University of Massachusetts, 1995.