An Assessment of CCP Approach in Statistical Learning

Artür Manukyan*
Koç University, Istanbul, Turkey - amanukyan13@ku.edu.tr

Elvan Ceyhan
Koç University, Istanbul, Turkey - elceyhan@ku.edu.tr

Abstract

In statistical learning, numerous methods use proximity graphs to model the structure of the data. Among proximity graphs, class cover catch digraphs (CCCDs) have been introduced primarily to investigate the class cover problem (CCP) but also were employed in classification and clustering. However, similar to all supervised learning algorithms, CCCDs have their own advantages and drawbacks. In this study, we apply CCCD classifiers on various types of artificial datasets in order to further evaluate their classification performance. We consider various scenarios: (i) one class is embedded inside of the other; (ii) classes have overlapping support where the level of overlap is controlled by a parameter; (iii) classes are from a mixture of uniform distributions with circular support.

Keywords: class cover problem; proximity graphs; classification

1. Introduction

The *k-nearest neighbor* (k-NN) classification is considered to be one of the oldest and the most commonly used classification methods (Cover and Hart, 1967; Fix and Hodges, 1989). Moreover, condensed and reduced NN algorithms (CNN and RNN), have also been introduced to represent the discriminant regions based on the entire data set of k-NN by fewer number of points when $k = 1$ (Hart, 1968; Gates, 1972). Class cover problem (CCP) uses a similar approach for classification (Cannon and Cowen, 2004). The goal in CCP is to define a set of points which are the center of a set of balls. Each ball covers neighbouring class points, and the main objective is to define a region to cover all the points in a given class with a small (preferably minimum) number of balls.

In this study, we use class cover catch digraphs (CCCD) introduced by DeVinney et al. (2002) which is also a graph theoretic representation of CCP. In CCCD, the term 'catch' refers to covering neighbouring points by a relative neighborhood rule, and 'digraph' refers to the one-sided (i.e. non-symmetric) relationship between the 'catching' and 'caught' points. Here, catching points are the points used to construct the cover, and caught points are the points which are covered. Since a pair of points do not necessarily cover each other, the relation could be one sided. This relative neighbourhood measure is given by a set of covering balls, in which radii are determined by some fashion. The method will be explained briefly and employed in classification along with k-NN classifier.

2. Class Cover Catch Digraphs

CCP was first introduced by (Cannon and Cowen, 2004) to define covers of classes, called *class covers*, in $\mathbb{R}^d$ where all points from the class of interest lie in the corresponding cover. CCCDs are graph theoretic representations of the CCP (Priebe et al., 2001, 2003). For example in a two class setting, let $\mathcal{X}_n = (x_1, x_2, ..., x_n)$ and $\mathcal{Y}_m = (y_1, y_2, ..., y_m)$ be two realizations of points representing two classes. Assume, without loss of generality, that target class is $\mathcal{X}_n$. In a CCCD, $x_i \in \mathbb{R}^d$ is the center of a ball with radius r_i. Each ball is represented by $B_i = B(x_i, r_i)$, where $x_j \in B_i$ if and only if $d(x_j, x_i) < r_i$, meaning x_i catches (or covers) x_j. Here, $d(.,.)$ can be any distance measure but we take the Euclidean distance. The radius r_i, with a relaxation parameter τ, is defined as, given $0 \leq \tau \leq 1$:

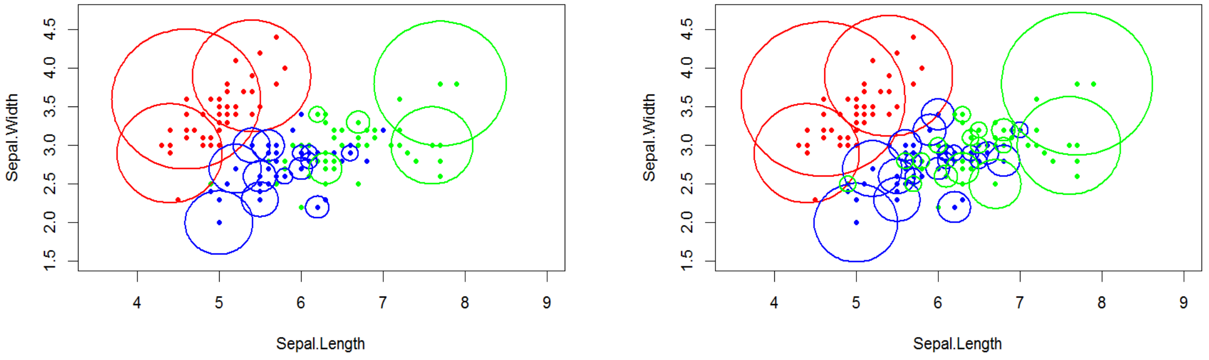

Figure 1: Two class cover scenerios: covers with $\tau = 1$ (left), covers with $\tau = 0$ (right). When $\tau = 0$, some points cover only themselves since their corresponding radii are 0

$$r_i = (1 - \tau) d(x_i, x^*) + \tau d(x_i, y^*), \tag{1}$$

where

$$y^* = \arg \min_{y \in \mathcal{Y}_m} d(x, y) \tag{2}$$

and

$$x^* = \arg \max_{z \in \mathcal{X}_n} \{d(x, z) : d(x, z) < d(x, y^*)\}. \tag{3}$$

The goal is to find a subset of balls, $C \subseteq \mathcal{B} = (B_1, B_2, ..., B_n)$, catching all the points of the target class $\mathcal{X}_n$, $x \in \cup_{B \in C} B$ for all $x \in \mathcal{X}_n$. CCCD is the digraph $D = D(V, A)$ with vertex set $V(D) = \mathcal{X}_n$ and the arc set $A(D)$ where $(x_i, x_j) \in A(D)$ if and only if $x_j \in B_i$. Hence in the CCCD, the term 'catch' refers to arc (x_i, x_j), and 'digraph' refers to the one-sided (i.e. asymmetric) relationship between the 'catching' and 'caught' points x_i and x_j, respectively. Thus, finding an appropriate cover $C \subseteq \mathcal{B}$ corresponds to find the dominating set $S \subseteq V(D)$ with approximate minimum cardinality. An illustration has been given employing the first two dimensions of the well-known Iris dataset in Figure 1 (Bache and Lichman, 2013).

After establishing each class cover $C_k = \cup_1^{n_k} B_i$, for $C_k \in \mathcal{C}$, given points can be easily classified in R^d. Here, n_k is the number of chosen balls for k'th class cover. Let C_{0k} be the k'th class cover, and C_{1k} be the union of the remaining class covers, $C_{1k} = \cup_{k \neq j} C_j$, respectively. Here, there are three cases according to the position of a given point to be classified: (i) a new point z is in a single cover, (ii) the point is in multiple balls where some balls are of different class covers (iii) the point is in non of the covers. For all the cases the minimizing the function:

$$\min_{C \in \mathcal{C}} \left[\min_{i: B_i \in C} \frac{d(z, x_i)}{r_i} \right] \tag{4}$$

is sufficient, since $\frac{d(z, x_i)}{r_i} < 1$ for $z \in B_i$ where $B_i \in C_k$ if z is in the cover C_k. The minimizing function simply considers points with distance ≤ 1 as points inside balls.

4. Simulations

We will consider three simulation models: (i) one class is embedded inside of the other (i.e. support of one class lies entirely in the interior of the support of the other); (ii) classes have overlapping support where the level of overlap is controlled by a parameter δ; (iii) classes are from a mixture of uniform distributions with circular support. Here, we will compare the performance of seven classifiers: CCCD with $\tau = 0$ (referred as CCCD0), with $\tau = 1$ (CCCD1), with overall best τ (CCCD-opt) and with τ producing the maximum performance; k-NN with $k = 1$ (NN), with overall best k (k-NN-opt), with k producing the maximum performance (k-NN-max). We evaluate the overall best k and τ by conducting a preliminary simulation study. In this pilot simulation, we take the mode k^* and τ^* of the set of k's and τ's producing the maximum classification performance of each Monte Carlo replication. Finally, we set modes k^* and τ^* as the fixed parameters used in k-NN-opt and CCCD-opt. After the preliminary simulation, we conduct a second simulation. On both simulations, we train the data with number of observations $n_x, n_y \in \{50, 100, 200\}$ for each class and find the classification rate as the number of correctly classified points in the test data with size $n_t = 100$ with same number of observations from each class.

We will start with model (i). A similar version of the this model has been investigated before where CCCD showed relatively good performance compared to k-NN (DeVinney et al., 2002). Here, we will replicate this experiment by investigating the two class setting $\mathcal{X}_n \sim U(0,1)^d$ and $\mathcal{Y}_m \sim U(0.3, 0.7)^d$. Thus, support of one class is completely embedded into the support of other. The classification rates of seven classifiers have been given in Figure 2. In all observation and dimensionality settings, CCCD classifiers beat the NN classifier. CCCD-max achieves up to %1 better correct classification rate compared to other CCCD versions. Although, CCCD shows slightly less performance than k-NN in $d = 2$, CCCD performs better for $d = 3$ and especially for $d = 5$. However, the relation between rates of the classifiers drastically change with the number of observations in each class of the training set. In the next experiment, we set the $n_x = 200$ and $n_y = 50$. Hence, the class with bigger support has more points to represent it. The results have been given in Figure 3. Apparently, the ratio n_x/n_y effects the solutions where k-NN beats CCCD classifiers but CCCDs are catching up to k-NN with increasing dimensionaliy d. This effect is merely a result of the differences between areas of support of each class. In fact, this result pertains to the influence of intensity or density (number of points/per unit area) of classes. In particular, our simulation results suggest that CCCD-based classifiers are more robust to the differences in class intensity levels. On the other hand, differences in class intensity highly confound the performance of k-NN classifiers.

In the second simulation model, we will first assign the same support for both classes, but slightly move the support of the second class away in the diagonal direction. Here, $\mathcal{X}_n \sim U(0,1)^d$ and $\mathcal{Y}_m \sim U(\delta, 1+\delta)^d$ for $\delta \in \{0, 0.1, 0.2, \cdots, 1\}$. The classification rates of seven classifiers for $d \in \{2, 3, 5\}$ have been given in Figure 4. Unlike model (i) with embedded classes, increasing dimensionality does not help CCCD to outperform the k-NN. However, similar to the embedded case, maximizing the classification performance of CCCD over τ, CCCD-max, allows CCCD to catch the performance of k-NN-max.

Finally, for the model (iii), we define classes as mixtures of uniform distributions with circular support. Let $\mathcal{X}_n \sim \sum_{i=1}^{k_1} \frac{1}{k_1} CU(x_i, r_i)$ and $\mathcal{Y}_m \sim \sum_{i=1}^{k_2} \frac{1}{k_2} CU(y_i, r_i)$ where $CU(x, r)$ is a 2 dimensional uniform distribution with a circular support for the center x and radius r:

$$f_{CU}(z|x, r) = \frac{1}{\pi r^2} I\{d(x, z) \leq r\}. \tag{5}$$

Here, centers x_i, y_i are drawn from $U(0,1)^2$ and radii r are drawn from $U(0.2, 0.4)^2$. In this fashion, we have generated two models with two classes of mixtures. Supports of these two models have been illustrated in Figure 5 and results of the classifiers for these two models have been given in Table 1. The results are similar to simulation models (i) and (ii): CCCD-max behaves similar to k-NN-max and CCCD-max allows correct classification rates of CCCD classifiers to increase up to %3. However, k-NN-max outperforms CCCD-max in this setting.

5. Conclusions

In this study, we have investigated the relationship between four versions of CCCD and three versions of k-NN classifiers within three simulation models. In simulation model (i), CCCD performs better than k-NN with increasing dimensionality d. However, this is not the case with the simulation model (ii): k-NN outperforms CCCD with all δ and $d \in \{2, 3, 5\}$ cases. Similarly, although the first part of the simulation

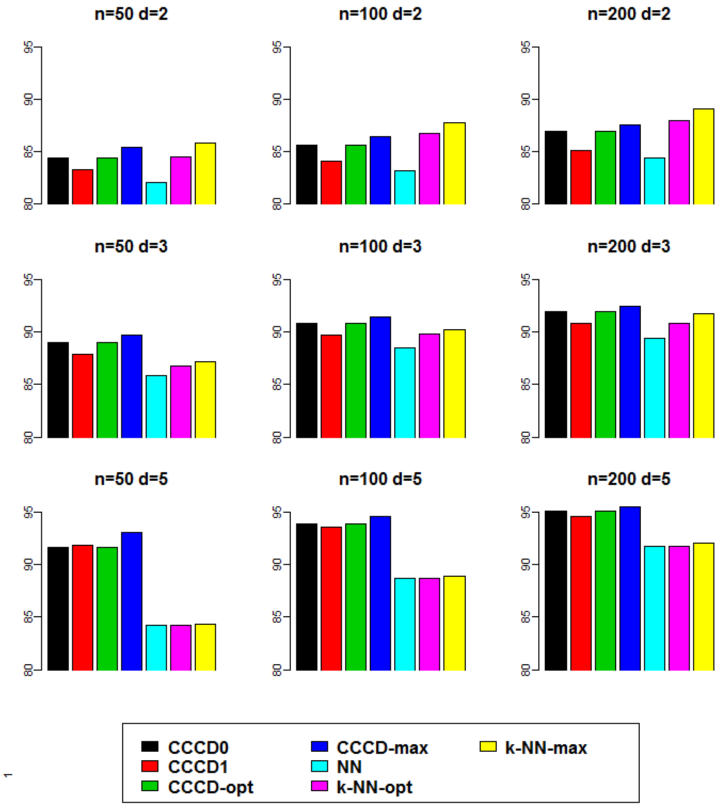

Figure 2: Correct classification rates (in percentage) of seven classifiers in nine different settings, where $n = n_x = n_y$ and d refers to $\mathbb{R}^d$.

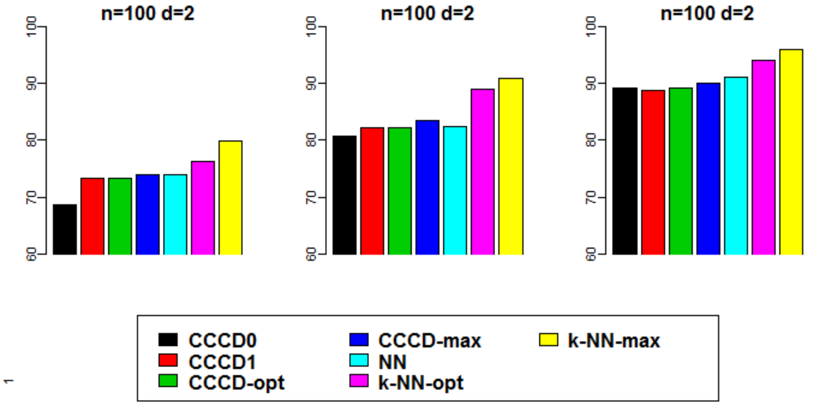

Figure 3: Correct classification rates (in percentage) of seven classifiers, where $n = n_x = n_y = 100$ and $d \in \{2, 3, 5\}$

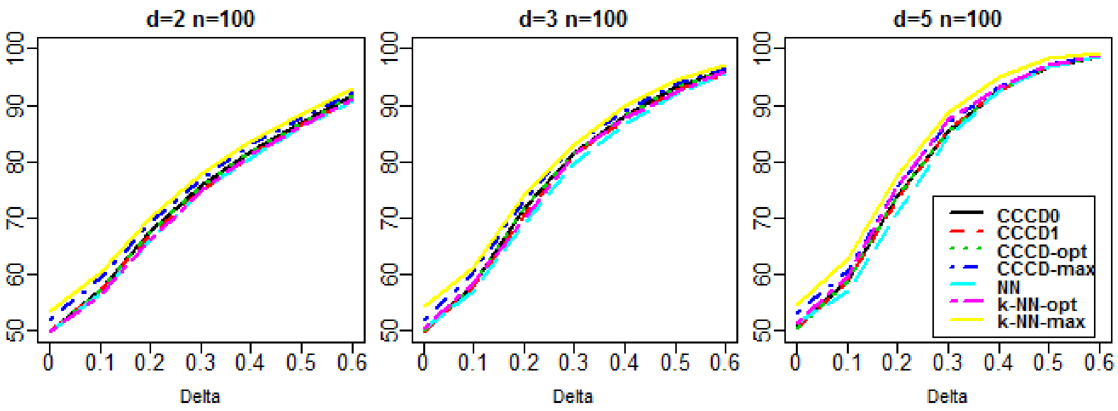

Figure 4: Correct classification rates (in percentage) of seven classifiers with datasets in $\mathbb{R}^d$ where $d \in \{2, 3, 5\}$ and the support seperating parameter $\delta \in \{0, 0.1, 0.2, \cdots, 1\}$.

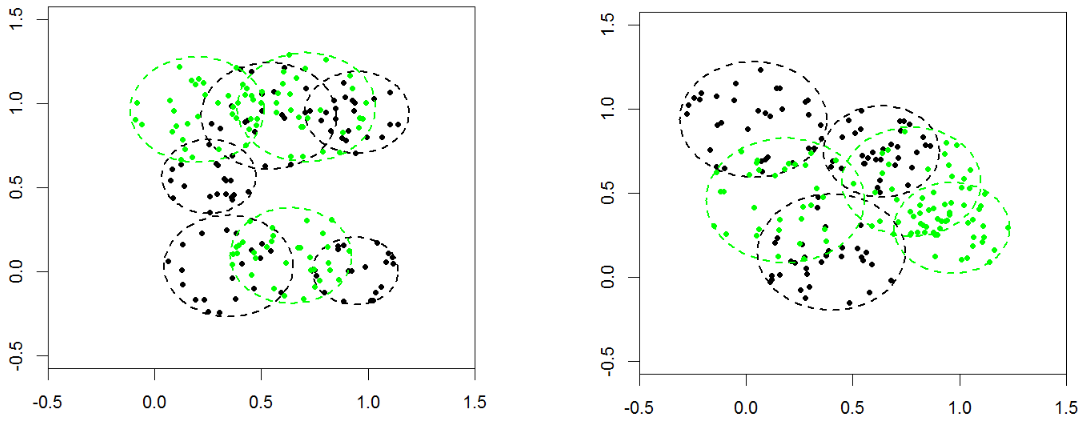

Figure 5: Simulation models of the two class (given in green and black) settings where the number of clusters for each class are $k_x = 5$ and $k_y = 3$ (left), and $k_x = 3$ and $k_y = 3$ (right). The support of clusters of each class are visualized as dashed lines.

n	CCCD0	CCCD1	CCCD-opt	CCCD-max	NN	k-NN-opt	k-NN-max
100	70.89	70.70	70.94	73.40	69.40	70.81	74.97
100	76.78	76.60	76.78	78.98	75.97	76.60	80.37

Table 1: Correct classification rates (in percentage) of the seven classifiers for $n = n_x = n_y = 100$.

model (i) suggests that CCCD works better than k-NN in cases when one support is embedded in the other, results with different class intensity levels show that CCCD-based classifiers actually perform better than k-NN classifiers in some cases. Hence, CCCD-based classifiers has shown to be more robust to differences in class intensity levels. Finally, we observe that optimum values of τ, CCCD-max, increases the correct classification rate up to %3 percent, allowing better performance than trivial cases of $\tau = 0$ and $\tau = 1$.

References

Bache, K. and Lichman, M. (2013). UCI machine learning repository.

Cannon, A. H. and Cowen, L. J. (2004). Approximation algorithms for the class cover problem. *Annals of Mathematics and Artificial Intelligence*, 40(3-4):215–223.

Cover, T. and Hart, P. (1967). Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on*, 13(1):21–27.

DeVinney, J., Priebe, C., Marchette, D., and Socolinsky, D. (2002). Random walks and catch digraphs in classification. *Computing Science and Statistics*, 34:107–117.

Fix, E. and Hodges, J. L. (1989). Discriminatory analysis. nonparametric discrimination: Consistency properties. *International Statistical Review*, 57(3):238–247.

Gates, G. W. (1972). The reduced nearest neighbor rule. *Information Theory, IEEE Transactions on*, 18:431–433.

Hart, P. (1968). The condensed nearest neighbor rule. *Information Theory, IEEE Transactions on*, 14:515–516.

Priebe, C. E., DeVinney, J. G., and Marchette, D. J. (2001). On the distribution of the domination number for random class cover catch digraphs. *Statistics & Probability Letters*, 55(3):239–246.

Priebe, C. E., Marchette, D. J., DeVinney, J. G., and Socolinsky, D. A. (2003). Classification using class cover catch digraphs. *Journal of Classification*, 20(1):003–023.