COPYRIGHT NOTICE

This is the author's version of the work. The definitive version was published in *IEEE Transactions on Pattern Analysis and Machine Intelligence* (TPAMI), 29(10):1746-1758, October 2007.

DOI: `http://dx.doi.org/10.1109/TPAMI.2007.1086`.

Condensed Nearest Neighbor Data Domain Description

Fabrizio Angiulli

Abstract—A simple yet effective unsupervised classification rule to discriminate between normal and abnormal data is based on accepting test objects whose nearest neighbors distances in a reference data set, assumed to model normal behavior, lie within a certain threshold. This work investigates the effect of using a subset of the original data set as the reference set of the classifier. With this aim, the concept of a reference consistent subset is introduced and it is shown that finding the minimum cardinality reference consistent subset is intractable. Then, the CNNDD algorithm is described, which computes a reference consistent subset with only two reference set passes. Experimental results revealed the advantages of condensing the data set and confirmed the effectiveness of the proposed approach. A thorough comparison with related methods was accomplished, pointing out the strengths and weaknesses of one-class nearest-neighbor-based training set consistent condensation.

Index Terms—Classification, data domain description, data condensation, nearest neighbor rule, novelty detection.

1 Introduction

Data domain description, also called one-class classification, is a classification technique whose goal is to distinguish between objects belonging to a certain class and all the other objects of the space. The task that needs solving in one-class classification is the following: given a data set of objects, called training or reference set, belonging to a certain object space, find a description of the data, i.e. a rule partitioning the object space into an accepting region, containing the objects belonging to the class represented by the training set, and a rejecting region, containing all the other objects. Data domain description is related to outlier or novelty detection, as the description of the data is then used to detect the objects deviating significantly from the training data.

Given a data set, also called reference set, of objects from an object space, and two parameters k and θ, we call Nearest Neighbor Domain Description rule (NNDD) the classifier that associates a feature vector $\delta(p) \in \mathbb{R}^k$ with each object p. The elements of $\delta(p)$ are the distances of p to its first k nearest neighbors in the reference set, and the classifier accepts p iff $\delta(p)$ belongs to the hypersphere (according to one of the L_r Minkowski metrics, $r \in \{1, 2, \ldots, \infty\}$) centered in the origin of $\mathbb{R}^k$ and having radius θ, i.e. iff $\|\delta(p)\|_r \leq \theta$.

The contribution of this work can be summarized as follows. The concept of *reference consistent subset* for the NNDD rule, which is a subset of the reference set that correctly classifies all the objects in the reference set, is defined and the relationship between the generalization of the NNDD classifier and size of the reference set is pointed out, concluding that replacing the original reference set with a reference consistent subset improves space requirements, response time, and generalization. It is shown that finding the minimum cardinality reference consistent subset is a computationally demanding task, and the algorithm CNNDD is provided that computes a reference consistent subset with only *two data set passes*. Experimental results show that the CNNDD algorithm achieves notable training set reduction and maintains or even improves the accuracy of the NNDD rule. Then the CNNDD algorithm is compared with related approaches, pointing out the strengths and weaknesses of one-class nearest neighbor-based training set consistent condensation. To conclude, robustness to noise and outliers is investigated.

The rest of the paper is organized as follows. In Section 2 relationship between the approach here proposed and relevant literature is presented. In Section 3 the NNDD rule and the concept of reference consistent subset are formally defined. In Section 4 the algorithm CNNDD is described. Section 5 reports experimental results and comparison with related methods. Finally, in Section 6 conclusions are drawn.

2 Related Work

The literature related to this work can be grouped into three main categories: nonparametric binary classification using the nearest neighbor rule, one-class classification, and outlier detection. Next, these approaches are briefly described.

In the nonparametric binary classification problem there is available a training set $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ of n pairs (x_i, y_i), $1 \leq i \leq n$, where x_i is an object from an object space and $y_i \in \{-1, 1\}$ is the corresponding class label. The *nearest neighbor rule* (1-NN-rule) [12] assigns the label y_j to a new object q, where x_j is the nearest neighbor of q in $\{x_1, \ldots, x_n\}$ according to a certain metric. This rule is based on the property that the nearest neighbor x_j of q contains at least half of the total

F. Angiulli is with DEIS, University of Calabria, Italy, email: f.angiulli@deis.unical.it

discrimination information contained in an infinite-size training set [6], [23], [8]. The generalization of the 1-NN-rule, the k-NN-rule, in which a new pattern q is classified into the class with the most members present among its k nearest neighbors in $\{x_1, \ldots, x_n\}$, has the property that its probability of error asymptotically approaches the Bayes error [9]. Since the k-NN-rule requires all the previously classified data to be stored, several techniques to reduce the size of the stored data have been proposed. Among these techniques, training set consistent ones aim at selecting a subset of the training set that classifies the remaining data correctly [15], [7], [4], [1]. These methods have the same goal as the CNNDD, but the subset they extract is consistent for the k-NN classification rule rather than for the NNDD rule. Thus, it is worth pointing out that, since the two decision rules are greatly different, the consistent subsets of the two rules are of a different nature. Even disregarding the fact that the k-NN-rule requires labelled data while the NNDD works on unlabelled data, there is no way of obtaining a subset consistent for the latter rule starting from a subset consistent for the former, or vice versa. Indeed, consider that objects belonging to one type of subset do not usually belong to the other type. This is because the condensed k-NN-rule searches for the objects mostly contributing to form the boundary between the classes in the data set, while the condensed NNDD rule searches for objects distributed along the data set shape and central with respect to the overall population.

The one-class classification task has been previously introduced. There are several approaches to one-class classification. In the nearest neighbor one-class classification method NN-d [25], a test object p is accepted if the distance to its nearest neighbor q in the training set is less or equal than the distance from q to its nearest neighbor in the training set. This measure is comparable with the Local Outlier Factor [3] used to detect outliers. The k-center method covers the data set with k balls with equal radii [28]. Ball centers are placed on training objects such that the maximum distance of all minimum distances between training objects and the centers is minimized. Placing balls is similar to solving the k-center problem introduced in the discrete location theory, where, given a set of cities, one has to pick k cities and build warehouses in them so as to minimize the maximum distance of any city from its closest warehouse (see for an example [16]). One-class classification techniques based on Support Vector (SV) Machines extend the SV algorithm to the case of unlabelled data [21], [24] (see Section 5.2 for a better description of this technique). In order to properly model outlier or noise possibly in the training set, model-based approaches, such as the one presented in [20], assume that the data set consists of a mixture of two populations, a regular population with distribution μ_{REG} and an *outlier* population with distribution μ_{OUT}. Assuming that these distributions are members of given parametric families, the goal is that of estimating their parameters. There are many other interesting approaches to one-class classification. The reader is referred to [26] for a comprehensive treatment.

Research on outlier detection in data mining focuses on providing techniques for identifying the most deviating objects in an input data set. Distance-based outlier detection was introduced in [17]: a point in a data set is a $DB(c, d)$-outlier with respect to parameters c and d, if at least fraction c of the points in the data set lies greater than distance d from it. This definition generalizes several discordancy tests to detect an outlier given in statistics and it is suitable when the data set does not fit any standard distribution. The definition of [19] is closely related to the previous one: given a k and n, a point p is an outlier if no more than $n-1$ other points in the data set have a higher value of D^k than p, where $D^k(p)$ denotes the distance of the kth nearest neighbor of a point p. In order to take into account the sparseness of the neighborhood of a point, [2] considers the measure $w_k(p)$ for each point p, denoting the sum of the distances to its k nearest neighbors. [11] provides further algorithms for distance-based anomaly detection. We point out that the measure $\|\delta(p)\|_r$ here used, generalizes all the distance-based measures, since $D^k(p) = \|\delta(p)\|_\infty$, and $w_k(p) = \|\delta(p)\|_1$.

3 The NNDD Rule

In the following a set of objects is denoted with U, with d a distance on U, D a set of objects from U, k a positive integer number, θ a positive real number, and r a Minkowski metric L_r, $r \in \{1, 2, \ldots, \infty\}$.

Given an object p of U, the *kth nearest neighbor* $nn_{D,\mathrm{d},k}(p)$ of p in D according to d is the object q of D such that there exists exactly $k-1$ objects s of D with $\mathrm{d}(p, s) \leq \mathrm{d}(p, q)$. If $p \in D$, then $nn_{D,\mathrm{d},1}(p) = p$. The k *nearest neighbors distances vector* $\delta_{D,\mathrm{d},k}(p)$ of p in D is

$$\delta_{D,\mathrm{d},k}(p) = (\mathrm{d}(p, nn_{D,\mathrm{d},1}(p)), \ldots, \mathrm{d}(p, nn_{D,\mathrm{d},k}(p))).^1$$

The *Nearest Neighbor Domain Description rule* (NNDD for short) $\mathrm{NNDD}_{D,\mathrm{d},k,\theta,r}$ according to $D, \mathrm{d}, k, \theta, r$, is the function from U to $\{-1, 1\}$ such that

$$\mathrm{NNDD}_{D,\mathrm{d},k,\theta,r}(p) = \mathrm{sign}(\theta - \|\delta_{D,\mathrm{d},k}(p)\|_r),$$

where $\mathrm{sign}(x) = -1$ if $x < 0$, and $\mathrm{sign}(x) = 1$ otherwise.

Intuitively, the NNDD rule returns 1 when the object belongs to the class represented by D, while it returns -1 when the object does not belong to that class. In the special case $k = 1$ and $\theta = 0$, the rule accepts an object p iff $p \in D$, while for $k = 1$ and $\theta > 0$, the rule accepts an object if it lies in the neighborhood of radius θ of some object in D.

Let f be $\mathrm{NNDD}_{D,\mathrm{d},k,\theta,r}$. The *accepting region* $\mathcal{R}(f)$ of f is the set $\{x \in U \mid f(x) = 1\}$. The *rejecting region* $\overline{\mathcal{R}}(f)$ of f is the set $U \setminus \mathcal{R}(f)$. An object $x \in \overline{\mathcal{R}}(f)$ is said to be an *outlier*. The *empirical risk*, or *training set error*, of the NNDD classifier f is the quantity

1. If there are less than k objects in D, then (the last) $k - |D|$ elements of $\delta_{D,\mathrm{d},k}(p)$ are equal to $+\infty$.

$$R^{emp}(f) = \frac{|D \cap \overline{\mathcal{R}}(f)|}{|D|}.$$

The empirical risk is directly proportional to the value of k and inversely proportional to the value of θ. Indeed, $\|\delta_{D,\mathrm{d},k-1}(p)\|_r \leq \|\delta_{D,\mathrm{d},k}(p)\|_r$, for $k > 1$. $R^{emp}(f)$ is certainly zero for $k = 1$ or for arbitrarily large values of θ.

When the reference set D is large, space requirements to store D and time requirements to find the nearest neighbors of an object in D increase. In the spirit of the reference set thinning problem for the k-NN-rule [15], [27], next the concept of NNDD reference consistent subset is defined, and then it is shown that finding a minimum NNDD reference consistent subset is NP-hard.

An NNDD *reference consistent subset* of D w.r.t. d, k, θ, r, is a subset S of D such that

$$(\forall p \in D)(\mathrm{NNDD}_{D,\mathrm{d},k,\theta,r}(p) = \mathrm{NNDD}_{S,\mathrm{d},k,\theta,r}(p)),$$

i.e. a subset of D that correctly classifies the objects in D.

The complexity of finding a minimum reference consistent subset is related to the complexity of the following decision problem: given an integer number m, $1 \leq m \leq |D|$, the NNDD *minimum reference consistent subset problem* $\langle D, \mathrm{d}, k, \theta, r, m\rangle$ is: is there an NNDD reference consistent subset S of D w.r.t. d, k, θ, r such that $|S| \leq m$?

Theorem 1. Let $r \in \mathbb{N}^+$ denote a finite Minkowski metrics L_r. Then the $\langle D, \mathrm{d}, k, \theta, r, m\rangle$ problem is NP-complete.

Proof. (Membership) Given a subset S of D, having size $|S| \leq m$, it can be checked in polynomial time that, for each $p \in D$, $\mathrm{NNDD}_{D,\mathrm{d},k,\theta,r}(p) = \mathrm{NNDD}_{S,\mathrm{d},k,\theta,r}(p)$.

(Hardness) The proof is by reduction to the *Dominating Set Problem* [14]. Let $G = (V, E)$ be an undirected graph, and let $m \leq |V|$ be a positive integer. The *Dominating Set Problem* is: is there a subset $U \subseteq V$, called *dominating set* of G, with $|U| \leq m$, such that for all $v \in (V - U)$ there exists $u \in U$ with $\{u, v\} \in E$?

Let $G = (V, E)$ be an undirected graph. Define the metric d_V on the set V of nodes of G as follows: $\mathrm{d}_V(u, v) = 1$, if $\{u, v\} \in E$, and $\mathrm{d}_V(u, v) = 2$, otherwise. Let $\theta_{k,r}$ be $(1+2^r(k-1))^{1/r}$. Now we prove that G has a dominating set of size m iff $\langle V, \mathrm{d}_V, k, \theta_{k,r}, r, m\rangle$ is a "yes" instance.

First, we note that, for each $v \in V$, $\|\delta_{V,\mathrm{d}_V,k}(v)\|_r \leq (0 + 2^r(k-1))^{1/r} \leq \theta_{k,r}$.

($\Rightarrow$) Suppose that G has a dominating set U such that $|U| \leq m$. Then U is a reference consistent subset of V w.r.t. $\mathrm{d}_V, k, \theta_{k,r}, r$. Indeed, let v a generic object of V. If $v \in U$, then $\|\delta_{U,\mathrm{d}_V,k}(v)\|_r \leq (0 + 2^r(k-1))^{1/r} < \theta_{k,r}$, otherwise $v \notin U$ and $\|\delta_{U,\mathrm{d}_V,k}(v)\|_r \leq (1+2^r(k-1))^{1/r} \leq \theta_{k,r}$.

($\Leftarrow$) Suppose that there is a reference consistent subset U of V such that $|U| \leq m$. By contradiction, assume that there is $v \in (V - U)$ such that, for each $u \in U$, $\{v, u\} \notin E$. Then, $\|\delta_{U,\mathrm{d}_V,k}(v)\|_r \geq 2k^{1/r} > \theta_{k,r}$, and U is not a reference consistent subset of V. It follows immediately that U is a dominating set for G. □

Theorem 1 also holds for the special case $k = 1$ and $r = \infty$. It follows immediately from Theorem 1 that the problem of computing the minimum size reference consistent subset is NP-hard.

Before concluding the section, the concept of sample compression scheme is recalled. A *sample compression scheme* is defined by a fixed rule $\sigma : D \mapsto \sigma(D)$ for constructing a classifier from a given set of data. Given a training set D, it is compressed by finding the smallest subset (the compression set) $S \subseteq D$ for which the classifier $\sigma(S)$ correctly classifies the whole set D. It is known that the size of a sample compression scheme can be used to bound generalization [18], [13].

It can be finally concluded from the discussion above, that replacing the reference set D with a reference consistent subset S of D has a twofold usefulness: both response time and generalization of the classifier are improved.

```
Algorithm CNNDD(DataSet,d,k,θ,r)
// — First data set pass —
InRefSet = ∅; OutRefSet = ∅;
for each (p_i in DataSet)
  δ = ∅;
  for each (p_j in OutRefSet)
    Update(δ, d(p_i, p_j), p_j);
    Update(δ_j^t, d(p_i, p_j), p_i);
  for each (p_j in InRefSet)
    if (‖δ‖_r > θ) Update(δ, d(p_i, p_j), p_j);
  if (‖δ‖_r > θ)
    for each (p_j in OutRefSet)
      Update(δ_j, d(p_i, p_j), p_i);
      if (‖δ_j‖_r ≤ θ)
        OutRefSet = OutRefSet − {p_j};
        InRefSet = InRefSet ∪ {p_j};
    Update(δ, 0, p_i);
    if (‖δ‖_r > θ) OutRefSet = OutRefSet ∪ {p_i}
    else InRefSet = InRefSet ∪ {p_i};
// — Second data set pass —
RefSet = InRefSet ∪ OutRefSet;
for each (p_i in (DataSet − RefSet))
  for each (p_j in OutRefSet) if (‖δ_j^t‖_r > θ)
    if (i < j) Update(δ_j^t, d(p_i, p_j), p_i);
// — Reference set augmentation —
IncrRefSet = ∅;
for each (p_j in OutRefSet) if (‖δ_j^t‖_r ≤ θ)
  for each (p_i in IncrRefSet) Update(δ_j, d(p_j, p_i), p_i);
  while (‖δ_j‖_r > θ)
    Let p_i be the object of (δ_j^t − δ_j) with the
      minimum value of d(p_i, p_j);
    Update(δ_j, d, p_i);
    IncrRefSet = IncrRefSet ∪ {p_i};
RefSet = RefSet ∪ IncrRefSet;
return(RefSet);
```

Fig. 1. The CNNDD rule.

4 The Condensed NNDD Rule

In this section the algorithm CNNDD is described. This algorithm computes a reference consistent subset for the NNDD rule with only *two* data set passes.

The algorithm, shown in Figure 1, receives in input the dataset $DataSet$ and parameters d, k, θ, and r, and outputs a reference consistent subset $RefSet$. Let f denote the classifier $\mathrm{NNDD}_{DataSet,\mathrm{d},k,\theta,r}$. We recall that $RefSet$ must be such that, for each p of $DataSet$, the property

$$f(p) = \mathrm{NNDD}_{RefSet,\mathrm{d},k,\theta,r}(p) \tag{1}$$

holds. $InRefSet$ and $OutRefSet$ are sets used to partition the objects of the reference consistent subset $RefSet$ as described in the following. Each object p_j in $OutRefSet$ has associated two heaps, δ_j and δ_j^t, storing, respectively, the k nearest neighbors of p_j in $RefSet$, and the k nearest neighbors of p_j in $DataSet$. Objects stored in $InRefSet$ have no heaps associated.

1st phase: first dataset pass. During this step, the set $OutRefSet$ stores the objects of $RefSet$ such that $\|\delta_j\|_r = \|\delta_{RefSet,\mathrm{d},k}(p_j)\|_r > \theta$, i.e. the objects of the reference set which are rejected by the NNDD rule, while $InRefSet$ contains the remaining objects of $RefSet$.

Furthermore, the heap δ, associated with the current data set object p_i, stores the k nearest neighbors of p_i in $RefSet$. Hence, $\|\delta\|_r = \|\delta_{RefSet,\mathrm{d},k}(p_i)\|_r$.

For each incoming data set object p_i, first of all, the distances among p_i and the objects p_j of $OutRefSet$ are computed and the heaps δ_j^t are updated. Next, until the value $\|\delta\|_r$ remains above the threshold θ, the distances among p_i and the objects p_j of $InRefSet$ are computed.

After having compared p_i with the objects in $OutRefSet$ and $InRefSet$, if $\|\delta\|_r$ remains above θ, then p_i is inserted in $RefSet$.[2] In this case, the heap δ is updated with the object p_i, and the heaps δ_i and δ_i^t associated with p_i are set equal to δ. Furthermore, the heaps δ_j associated with the objects already contained in $OutRefSet$ must be updated since now p_i belongs to $RefSet$: if, after updating δ_j, the value $\|\delta_j\|_r$ becomes less or equal to θ, then the object p_j is removed from $OutRefSet$ and inserted into $InRefSet$. In this case the heaps δ_j and δ_j^t are no longer useful and can be discarded.

$RefSet = OutRefSet \cup InRefSet$ being a subset of $DataSet$, then it is the case that $\|\delta_{DataSet,\mathrm{d},k}(p)\|_r \leq \|\delta_{RefSet,\mathrm{d},k}(p)\|_r$. Thus, the points p of $DataSet$ not stored in $RefSet$, are such that $\|\delta_{DataSet,\mathrm{d},k}(p)\|_r \leq \|\delta_{RefSet,\mathrm{d},k}(p)\|_r \leq \theta$, and Property (1) is guaranteed for these objects. Therefore, at the end of the first scan, the objects of the data set not belonging to $RefSet$ are correctly classified by it through the NNDD rule.

2. Note that, as long as there are less than k objects in the reference set $RefSet$, the condition $\|\delta\|_r > \theta$ is always satisfied, since at least one element of the vector δ is equal to $+\infty$, by definition of k nearest neighbors distances vector.

Furthermore, for each $p \in \overline{\mathcal{R}}(f)$, $\theta < \|\delta_{DataSet,\mathrm{d},k}(p)\|_r \leq \|\delta_{RefSet,\mathrm{d},k}(p)\|_r$, and, hence, $RefSet$ ($OutRefSet$ to be more precise) contains the set $\overline{\mathcal{R}}(f)$.

2nd phase: second dataset pass. The first dataset pass is not sufficient to assure consistency of the set $RefSet$, since this set could contain objects which are misclassified by $RefSet$ itself. It could be the case that an object belonging to $OutRefSet$, and hence rejected by the current $RefSet$, is not an outlier.

Indeed, let p_j stored in $OutRefSet$ at the end of the first scan. Unfortunately, $\|\delta_j^t\|_r > \theta$ does not imply that $p_j \in \overline{\mathcal{R}}(f)$, as p_j was not compared with all the data set objects during the first data set pass. Thus, in order to establish whether $\|\delta_{DataSet,\mathrm{d},k}(p_j)\|_r$ is greater than θ, a second data set scan is performed.

For each $p_j \in OutRefSet$, the heap δ_j^t is updated in order to compute the exact value of $\delta_{DataSet,\mathrm{d},k}(p_j)$, by comparing p_j with all the objects p_i in $(DataSet - RefSet)$ such that $i < j$, i.e. with the objects preceding p_j that are not stored in $RefSet$. In fact, a generic object p_j of $OutRefSet$ was compared, during the first scan, exactly with all the objects p_i of $DataSet$, $j < i$, and with all the objects $\{p_i \in RefSet \mid i < j\}$.

3rd phase: reference set augmentation: The third phase of the algorithm is introduced to guarantee Property (1) for the objects stored in $OutRefSet$ at the end of the first phase, but which are determined to be inliers at the end of the second phase.

To this aim, the set $RefSet$ is augmented with the set $IncrRefSet$ until consistency is achieved. In particular, for each $p_j \in OutRefSet$ such that $\|\delta_{DataSet,\mathrm{d},k}(p_j)\|_r \leq \|\delta_j^t\|_r \leq \theta$, i.e. such that it is not an outlier, $IncrRefSet$ is augmented with some nearest neighbors of p_j, until $\|\delta_{RefSet \cup IncrRefSet,\mathrm{d},k}(p_j)\|_r$ goes down to the threshold θ.

We note that at the end of the algorithm, the objects in $OutRefSet$ such that $\|\delta_j^t\|_r > \theta$ are the outliers of $DataSet$. This terminates the description of the CNNDD algorithm.

The CNNDD rule is suitable for disk-resident data sets, as it tries to minimize the number of I/O operations performing only two data set passes. As for the spatial cost of the CNNDD rule, it is $\mathcal{O}(|RefSet| \cdot k)$, owing to the space needed to store heaps associated with objects in $OutRefSet$. The temporal cost is $\mathcal{O}(|RefSet| \cdot |DataSet| \cdot (d + \log k))$, where d is the cost of computing the distance between two objects, and $\log k$ is the cost of updating a heap of k elements. The above-stated cost is a worst case, but usually each data set object is compared only with a fraction of the objects in the reference subset. Thus, the temporal cost of the CNNDD rule depends on the size of the computed reference subset and it is subquadratic in general, while it becomes quadratic when the reference subset consists of all the data set objects, i.e. if we set

$\theta = 0$. As shown in the following section, for values of θ of interest, the reference consistent subset is composed of a fraction of the data set objects.

5 Experimental Results

In this section experiments involving the CNNDD rule are reported. Before starting, next the data sets employed in the experiments are described. All the data sets are from the UCI Machine Learning Repository [10][3], except for the *Checkerboard* data set:

- **Checkerboard**: it is a synthetical data set composed of $2{,}000$ randomly generated points of the unit square partitioned into two classes representing the cells of a 4×4 checkerboard.
- **Image segmentation**: contains $2{,}310$ objects representing 3×3 pixel regions obtained from a database of outdoor images. Each object has 19 continuous attributes. The objects are partitioned into seven classes, each composed of 330 objects: *Brickface*, *sky*, *foliage*, *cement*, *window*, *path*, and *grass*.
- **Ionosphere**: consists of radar data collected by a system in Goose Bay, Labrador. The 351 objects of this data set have 34 continuous attributes and are partitioned into two classes: *Good* (225 objects, associated with instances showing evidence of some type of structure in the ionosphere), and *bad* (126 objects).
- **Iris**: this data set contains three classes of 50 instances each. Each class refers to a type of iris plant: *Setosa*, *Vericolour*, and *Virginica*. The objects have four attributes representing the length and width of both sepal and petal.
- **Letter recognition**: the instances of this data set have 16 numerical attributes representing statistical moments and edge counts associated with black and white images of the 26 capital letters in the English alphabet. The character images were based on 20 different fonts and each letter was randomly distorted to produce a total of $20{,}000$ images. The objects in the data set are partitioned into 26 classes associated with the capital letters of the alphabet. Each class contains approximately the same number of instances.
- **Satellite image**: consists of $6{,}435$ instances (obtained by merging both the training and the test set) generated from Landsat Multi-Spectral Scanner image. Each object has 36 attributes consisting of multi-spectral values of pixels in 3×3 neighborhoods in a satellite image. The class label is associated with the central pixel, and may be one of the following: *red soil* ($1{,}533$ objects), *cotton crop* (703 objects), *grey soil* ($1{,}358$ objects) *damp grey soil* (626 objects), *soil with vegetation stubble* (707 objects), and *very damp grey soil* ($1{,}508$ objects).
- **Shuttle**: this data set was used in the European Statlog project. It contains 8 attributes and $43{,}500$ instances. Approximately 80% of the data belongs to the class *Rad Flow*, that was assumed to represent the normal class, while the other instances belong to six different classes, that were merged to obtain a single exceptional class.
- **Vehicle**: the features of this data set were extracted from the silhouettes of four types of vehicles. There are 18 attributes for each object. The data set consists of 846 object partitioned into the following four classes of vehicles: *Opel* (218 objects), *Saab* (217 objects), *bus* (199 objects), and *van* (212 objects).
- **Wine**: these data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars (representing the three classes of the data set). The analysis determined the quantities of 13 constituents found in each of the three types of wines (the attributes of the objects). The objects are partitioned as follows: 59 objects into *class 1*, 71 objects into *class 2*, and 48 objects into *class 3*.

The experiments are organized as follows. First of all, the behavior of the CNNDD is studied. Then, the CNNDD rule is compared with some one-class classification methods, namely the NN-d method, the k-center method, and the one-class SVM algorithm. Finally, the effect of Minkowski's metric r used to compute the norm of the k nearest neighbors distances vector, and robustness to noise and outliers is investigated. In all the experiments, if not specified otherwise, the Euclidean distance was used, while the parameter r of the CNNDD method was set to 1.

5.1 The effect of condensing the data set

In this section the effect of condensing the reference set for the NNDD rule is investigated. Several experiments were performed. In each experiment, one of the data set classes above described was considered the normal one, while the other classes of the same data set formed the abnormal class.

During each experiment, the value of the parameter k was varied between 1 and 10, and for each distinct value of k, the parameter θ was varied in the range $[0, \theta_{\max}]$. The value $\theta_{\max}$ depends on the data set considered.

For any combination of the parameters k and θ, the empirical error, the false positive rate, and the detection rate of both the NNDD and CNNDD rules, and the size of the consistent reference subset, were computed.

The *false positive rate* (f.p., for short, in the following) is the fraction of normal objects rejected by the classifier. The *detection rate* (d.r., for short, in the following) is the fraction of abnormal objects rejected by the classifier.

It must be recalled that, since it is assumed that the data set is composed only of normal objects, the

3. Data sets were, somewhat randomly, selected among those whose objects are encoded as vectors of numeric attributes (since the CNNDD implementation we had available manipulates only these kinds of object).

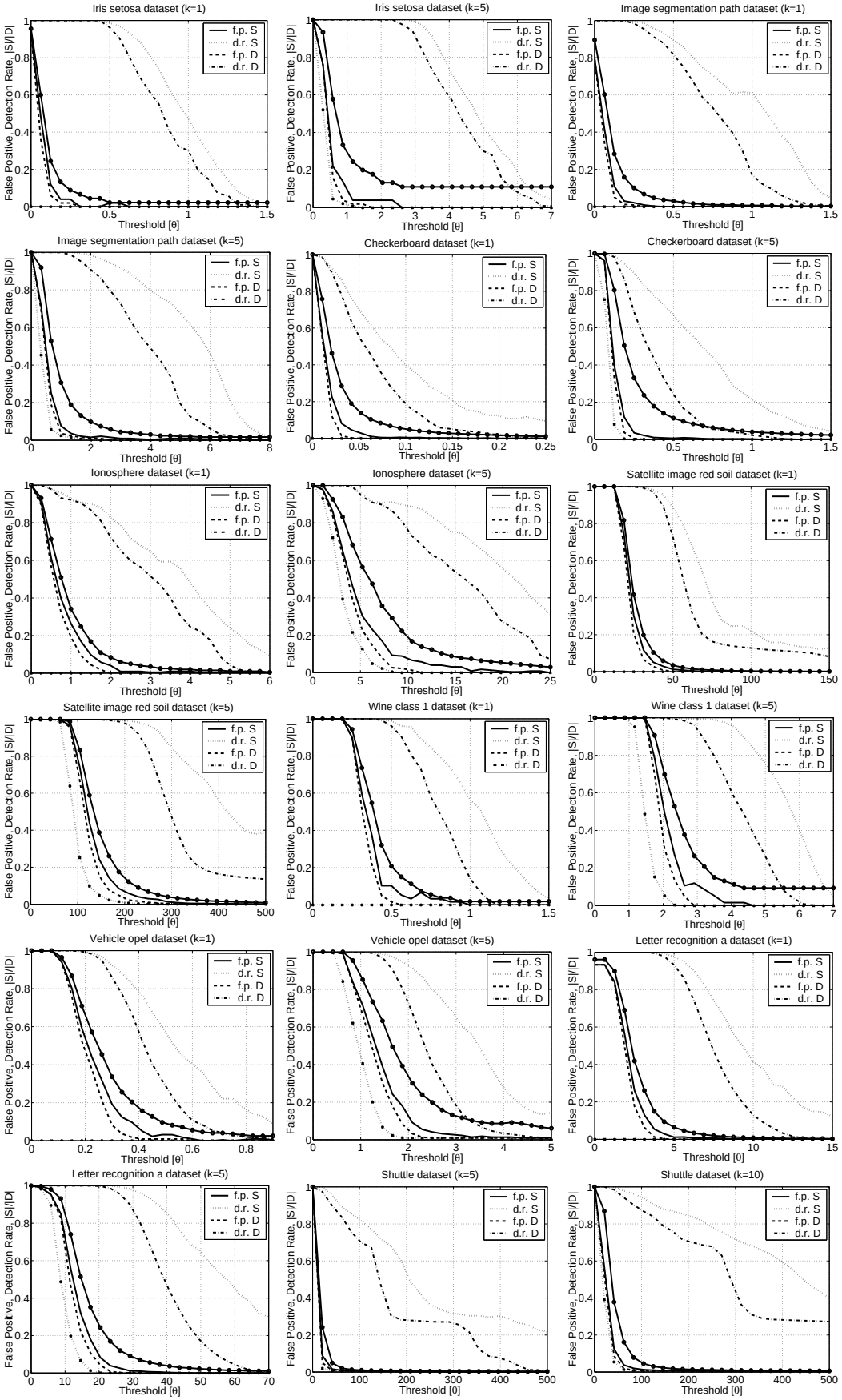

Fig. 2. Comparison between the CNNDD and the NNDD rules.

abnormal objects are unknown at learning time. Both false positive rate and detection rate were computed by 10-fold cross-validation.

Some of the experiments executed are reported in Figure 2 (only one class for data set, since the behavior of the method on the other classes was analogous, and those for $k = 1$ and $k = 5$, in order to show the effect of increasing the value of the parameter k). The x axis reports the threshold value θ, while the y axis varies between 0 and 1, and reports the false positive rate, the detection rate, the empirical error, and the normalized size of the reference consistent subsets.

Solid and dotted lines represent the false positive rate and the detection rate of the CNNDD rule respectively. Dashed and dash-dotted lines represent the false positive rate and the detection rate of the NNDD rule. The empirical error is represented by the dashed pointed lines. Finally, the solid pointed lines report the normalized size of the reference consistent subset computed by the CNNDD rule.

From these figures, it is clear that, for relatively large values of parameter θ, the CNNDD rule noticeably improves the detection rate over the NNDD rule with a little loss, or even with no loss, of false positive rate. Furthermore, in the same range of values of θ, the training set compression achieved by using the CNNDD rule is noticeable. Depending on the data set of interest, and on the desired trade-off between detection rate and false positive rate, the size of the reference consistent subset ranges from a few percent of the overall training-set up to 10%–20% of the training-set objects. It can be thus concluded that the consistent reference subset guarantees remarkable reference set size reduction. The best trade-off between classifier accuracy and compression ration is achieved in the curve elbow of the false positive rate.

Furthermore, as expected, when θ approaches zero, both the false positive rate and the detection rate approach one, while the consistent reference subset computed by the CNNDD rule tends to contain all the data set objects, as they are almost all outliers.

By observing Figure 2 it can be concluded that, when a certain value of false positive rate is fixed, by using a value for parameter k greater than one may improve the detection rate achievable at the expense of an increase in the size of the reference consistent subset.

Interestingly, it can be noticed that, for example, on the *Image segmentation path* data set, for $k = 5$ and $\theta = 2$ the CNNDD method achieves a detection rate of 0.984, while the detection rate of the NNDD method was about 0.907, with practically no loss of false positive rate (0.015 of CNNDD versus 0.012 of NNDD) and a reference set which is less than 10% of the whole data set. As a further example, consider the *Wine class 1* data set for $k = 5$ and $\theta = 4$, where the CNNDD improves the detection rate from 0.575 of the NNDD rule to about the 0.948 by employing a subset of only 11% of the data set. Also, consider the *Letter recognition* data set for $\theta \in [20, 40]$

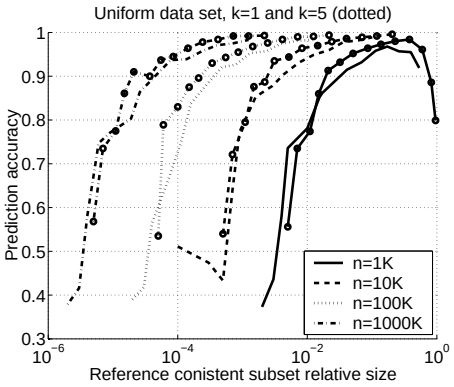

Fig. 3. How the compression ratio and the prediction accuracy vary with the training set size on a uniform data set.

and $k = 5$. The same behavior can be observed in other experiments shown and also in the experiments concerning the data set classes not reported here owing to space limitations.

If both normal and abnormal data are provided, a suitable combination of values for the parameters can be determined by executing the same kind of experiment described above. Nevertheless, the method being unsupervised, it can be employed when only normal data is available. In this case, as holds for any other unsupervised method, it can be difficult to determine the right value for the parameters.

However, in the case of the CNNDD rule, the following procedure can be profitably applied. As far as the value of parameter θ is concerned, it can be used the value θ^* such that the slope of the curve of the threshold θ versus the normalized size of the reference consistent subset $\rho = |S|/|D|$ is equal to a user-provided value α. As far as the value of parameter k is concerned, the value $k^* \geq 1$ can be used such that the size of the reference consistent subset achieved for $\theta = \theta^*$ does not exceed a user-provided ratio $\rho_{\max}$. As can be verified on the curves in Figure 2, using $\alpha \approx 45^\circ$ and $\rho_{\max} \leq 0.2$ provides a good classifier in almost all the experiments.

Before concluding this section, it is studied how the condensation ratio and the prediction accuracy achieved by the CNNDD method vary together with the training set size. With this aim, a family of synthetically generated training sets, called *Uniform* data set in the following, was considered. Each training set of the family is composed by n points ($n \in \{1K, 10K, 100K, 1000K\}$) uniformly distributed into the square $[0.25, 0.75]^2$, which it is assumed to represent the normal class. To measure the prediction accuracy, $1{,}000$ uniformly distributed random points of the unit square were employed. For each training set and $k \in \{1, 5\}$, the parameter θ was varied to obtain different consistent subsets and the prediction accuracy achieved was measured. Figure 3 shows the curves of the reference consistent subset relative size versus the prediction accuracy. It is worth noticing that the

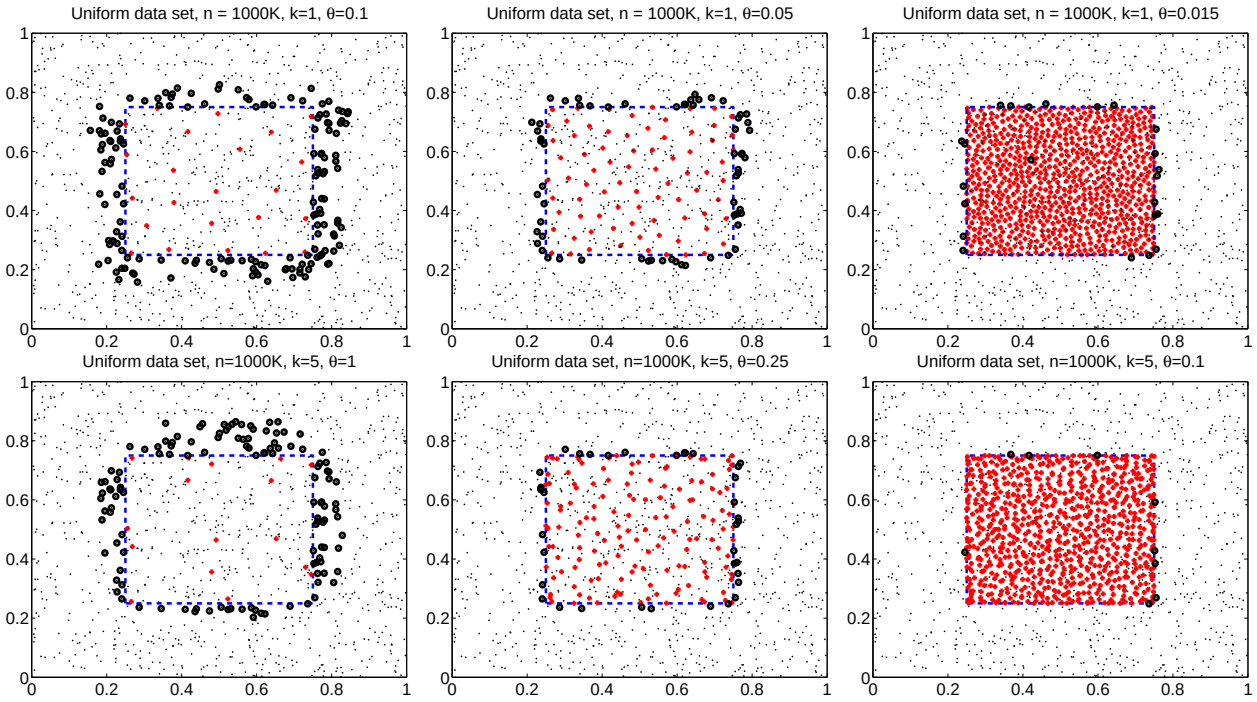

Fig. 4. Training set consistent subsets of the *Uniform* data set for various values of the parameters and points misclassified during prediction.

condensation ratio increases dramatically with increasing training set size, while the quality of the prediction remains unchanged, and, also, that by increasing the parameter k (the dotted lines are those for $k = 5$) the accuracy of the classifier is improved.

Figure 4 shows some examples of training set consistent subsets computed by the CNNDD algorithm on the *Uniform* data set composed of $n = 1{,}000K$ points, together with the misclassified points of the test set. For clarity, the training set points are not shown in the figure. The dashed curve represents the square $[0.25, 0.75]^2$ partitioning the points of the unit square into two classes. The small crosses represent the consistent subset points, while the dots are the test points. Circles are the test points misclassified by the consistent subset. From left to right and from top to bottom the sizes of the consistent subset are 23, 78, 792, 16, 144, and 795, while the prediction accuracies are 0.844, 0.937, 0.977, 0.876, 0.964, and 0.991. It is clear that by decreasing the value of θ the number of points composing the consistent subset increases, while the prediction accuracy is improved.

5.2 Comparison with other approaches

In this section, the CNNDD rule is compared with the NN-d [25], k-center [28], and one-class SVM [22] classifiers. These methods are described first.

The one-class classifier NN-d accepts a test object when its local density is larger or equal to the local density of its kth nearest neighbor. The local density of an object is estimated by computing the distance between the object and its kth nearest neighbor in the reference set without q. Thus, given a test object p, the NN-d method accepts p if

$$\frac{\mathrm{d}(p, nn_k(p, D))}{\mathrm{d}(nn_k(p, D), nn_{k+1}(nn_k(p, D), D))} \leq \theta$$

and rejects it otherwise ($k = 1$ and $\theta = 1$ are usually employed).

The k-center one-class method covers the dataset with k balls having equal radii. Ball centers μ_j are placed on training objects such that the maximum distance of all minimum distances between the training objects and the centers is minimized while all the objects are covered by some ball, i.e. the following measure is minimized:

$$\rho = \max_{p_i \in D} \; [\; \min_{j=1}^{k} \mathrm{d}(p_i, \mu_j) \;].$$

The radius ρ defines the boundary of the target class around the k selected centers.

The one-class SVM algorithm is a specialization, working in the presence of only positive data, of the standard two-class SVM algorithm, which, conversely, requires both positive and negative examples. Basically, the feature space is transformed via a kernel and then the origin of the transformed space is treated as the only member of the negative class. Thereafter, the standard two-class SVM algorithm is employed. The one-class SVM exploits the parameter $\nu \in (0, 1]$ in order to control the trade-off between the number of training-set examples accepted and the size of the support vector regularization term. Specifically parameter ν is both an upper bound on the fraction of outliers and a lower bound on the fraction of support vectors. The LibSVM [5] implementation of

Data set	CNNDD	NN-d	k-center	one-class SVM
Checkerboard	0.978 ($k = 7$)	0.937 ($k = 4$)	0.917	0.946 ($\gamma = 50.0$)
Image segmentation brickface	0.998 ($k = 1$)	0.983 ($k = 10$)	0.994	0.870 ($\gamma = 2.00$)
Image segmentation sky	1.000 ($k = 2$)	0.999 ($k = 3$)	0.998	0.995 ($\gamma = 0.20$)
Image segmentation foliage	0.962 ($k = 1$)	0.928 ($k = 3$)	0.845	0.891 ($\gamma = 5.00$)
Image segmentation cement	0.961 ($k = 3$)	0.876 ($k = 2$)	0.855	0.907 ($\gamma = 2.00$)
Image segmentation window	0.958 ($k = 2$)	0.926 ($k = 2$)	0.859	0.890 ($\gamma = 5.00$)
Image segmentation path	0.999 ($k = 2$)	0.995 ($k = 3$)	0.997	0.982 ($\gamma = 1.00$)
Image segmentation grass	0.997 ($k = 2$)	0.995 ($k = 6$)	0.995	0.956 ($\gamma = 0.001$)
Ionosphere good	0.959 ($k = 10$)	0.914 ($k = 9$)	0.937	0.918 ($\gamma = 0.10$)
Iris setosa	1.000 ($k = 1$)	0.989 ($k = 3$)	0.980	0.947 ($\gamma = 0.01$)
Iris versicolor	0.981 ($k = 5$)	0.958 ($k = 7$)	0.963	0.980 ($\gamma = 0.20$)
Iris virginica	0.960 ($k = 5$)	0.926 ($k = 4$)	0.921	0.937 ($\gamma = 0.10$)
Letter recognition A	0.998 ($k = 3$)	0.980 ($k = 4$)	0.992	—
Satellite image red soil	0.993 ($k = 1$)	0.938 ($k = 5$)	0.983	0.974 ($\gamma = 0.0001$)
Satellite image cotton crop	0.985 ($k = 5$)	0.849 ($k = 2$)	0.965	0.940 ($\gamma = 0.0001$)
Satellite image grey soil	0.974 ($k = 5$)	0.936 ($k = 3$)	0.953	0.963 ($\gamma = 0.0002$)
Satellite image damp grey soil	0.891 ($k = 5$)	0.819 ($k = 1$)	0.806	0.911 ($\gamma = 0.0001$)
Satellite image soil with vegetation stubble	0.944 ($k = 5$)	0.900 ($k = 1$)	0.848	0.851 ($\gamma = 0.0002$)
Satellite image very damp grey soil	0.950 ($k = 4$)	0.874 ($k = 2$)	0.889	0.942 ($\gamma = 0.0001$)
Shuttle	0.995 ($k = 2$)	0.995 ($k = 3$)	—	—
Vehicle opel	0.961 ($k = 5$)	0.937 ($k = 3$)	0.919	0.868 ($\gamma = 2.00$)
Vehicle saab	0.751 ($k = 5$)	0.698 ($k = 4$)	0.647	0.686 ($\gamma = 5.00$)
Vehicle bus	0.935 ($k = 2$)	0.912 ($k = 3$)	0.909	0.835 ($\gamma = 3.00$)
Vehicle van	0.740 ($k = 2$)	0.692 ($k = 1$)	0.719	0.649 ($\gamma = 5.00$)
Wine class 1	0.998 ($k = 5$)	0.993 ($k = 7$)	0.964	0.949 ($\gamma = 0.20$)
Wine class 2	0.916 ($k = 5$)	0.844 ($k = 4$)	0.857	0.847 ($\gamma = 0.10$)
Wine class 3	0.994 ($k = 3$)	0.988 ($k = 1$)	0.975	0.958 ($\gamma = 0.50$)

TABLE 1
Comparison of the CNNDD, NN-d, and one-class SVM methods through ROC areas.

the one-class SVM was used in the experiments below presented.

Before starting the comparison, it is of interest to point out major differences between k-center and CNNDD, since, among the methods above described, k-center is the most similar to the method here introduced. In particular, three aspects are to be taken into account: the form of the decision boundary, treatment of outliers, and algorithmic approaches.

Firstly, the form of the decision boundary of the k-center and CNNDD methods is different. Indeed, the accepting region of the k-center is the union of k balls centered on some data set objects. Conversely, the accepting region of the CNNDD is the union of at most $\binom{n}{k}$ regions, each associated with k distinct objects of the condensed set. For example, in a normed linear space, if the norm $r = 1$ is employed, it can be shown that these regions are convex sets "generated" by k elements of the condensed set. Thus, for $k = 2$ and $r = 1$, the accepting region of CNNDD is the union of hyper-ellipses. For $k > 2$, more complex regions, describing a boundary around the k points, are generated. As a consequence, the decision boundary of CNNDD may be more accurate than that of k-center, and this is especially evident whenever the objects of the target class are close to the objects of the other classes and/or the boundary is particularly complex, and also in the presence of noise or outliers (see below). For example, let an ellipse on the plane represent the accepting region of the target class. This region can be described exactly by the CNNDD with a subset composed of two objects (the focus of the ellipse), while the number of balls to be used by the k-center must approach infinity to reach the same level of detail.

Secondly, the k-center always accepts all the training set objects, while the CNNDD classifier rejects a fraction of the training set objects, directly proportional to the value of k and inversely proportional to the value of θ. This implies that the k-center is more sensitive to noise than the CNNDD. As an example, consider a training set composed of one thousand objects very close to each other, thus forming a cluster, plus a single outlying object far from the cluster. While the CNNDD method with $k = 2$ and a value of θ smaller than the distance separating the cluster from the outlier, covers the cluster with a hyper-elliptical region and leaves the outlier out of its accepting region, the k-center method with $k = 2$ covers the objects with two spheres, the first centered on the cluster and the second centered on the outlier. The neighborhood of the outlier being included in the accepting region of the k-center classifier, it is clear that it may degrade prediction accuracy, especially if it falls into the support of a different class. Furthermore, it follows from what is stated above that CNNDD can be used to

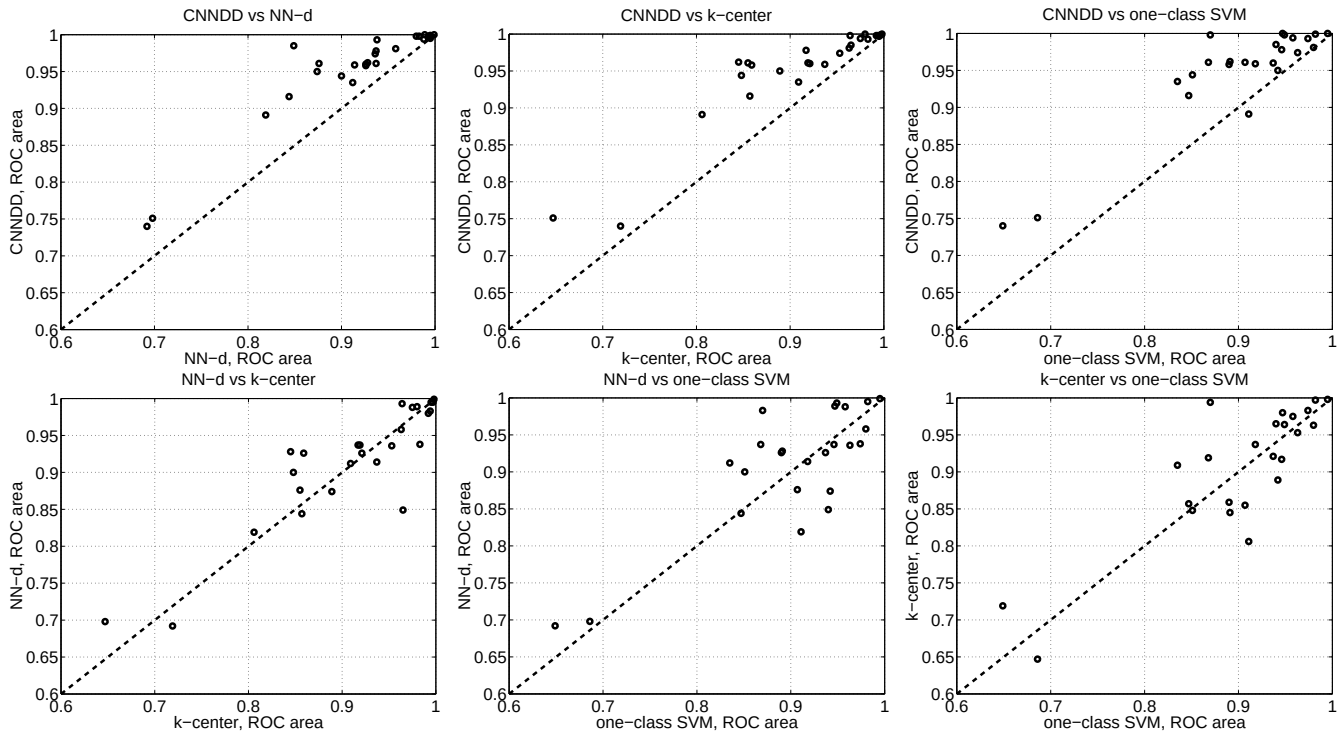

Fig. 5. Pairwise comparison of the ROC areas of the CNNDD, NN-d, k-center, and one-class SVM methods.

detect outliers in the input data set, while it is not the case of k-center.

Finally, even disregarding these basic differences between the two methods, it is clear that, since the k-center fixes the number k of centers in advance whereas the CNNDD fixes the threshold θ used to define the boundary around the subset objects, different algorithmic approaches are feasible to solve the two problems efficiently.

Comparison of the methods was made through ROC analysis. ROC curves are the plot of the false positive rate versus the detection rate, and the area under the curve gives an estimate of the ability of the method to separate inliers from outliers. ROC curves are computed by 10-fold cross-validation.

As far as the CNNDD and NN-d methods are concerned, parameter k was varied between 1 and 10, and, for each value of k, the parameter θ was varied between 0 and $\theta_{\max}$. As for the k-center method, the number k of centers was varied between 1 and the 90% of the training set objects. As regards the one-class SVM, parameter ν was varied between a value close to zero and 1.0, where n denotes the number of training set objects, and the RBF kernel was used, varying parameter γ between 10^{-4} and 10^2. For each method, the best area under the ROC curves computed as described above was then determined.

Table 1 shows the ROC areas of the three methods (together with the values for the parameters used to achieve that area). Interestingly, on all the data sets considered, the CNNDD rule performed better than all the other three methods.

As far as the one-class SVM method is concerned, in various experiments it was observed that, while for relatively small values of parameter γ, the ROC curve is not as good as the curve of the CNNDD method, instead for relatively large values of γ they are comparable. Nevertheless, often the curve of the one-class SVM method is not defined for small values of false positive rate (up to 5-10$\%$) and, hence, it is less accurate and eventually it scores a smaller ROC area.

As far as the NN-d method is concerned, its curve is always below the curve of the CNNDD method, and, as a result it scores a smaller ROC area. Furthermore, it must be pointed out that the NN-d rule uses all the data set objects as reference set.

A similar behavior was observed also for the k-center method. Basic differences between k-center and CNNDD were previously discussed.

Table 1 does not show the ROC areas of the one-class SVM on the *Letter recognition A* and *Shuttle* data sets, and of the k-center on the *Shuttle* data set, since computing them required too much computational effort.

The results presented in Table 1 are summarized in Figure 5, where a pairwise comparison of the four methods is accomplished. Each point (x, y) represents a distinct experiment, and it is such that x is the ROC area of the method on the abscissa, and y is the ROC area of the method on the ordinate. If the two methods are comparable, then the points will be symmetrically distributed along the dashed line connecting $(0, 0)$ to $(1, 1)$ and partitioning the unit square in two regions.

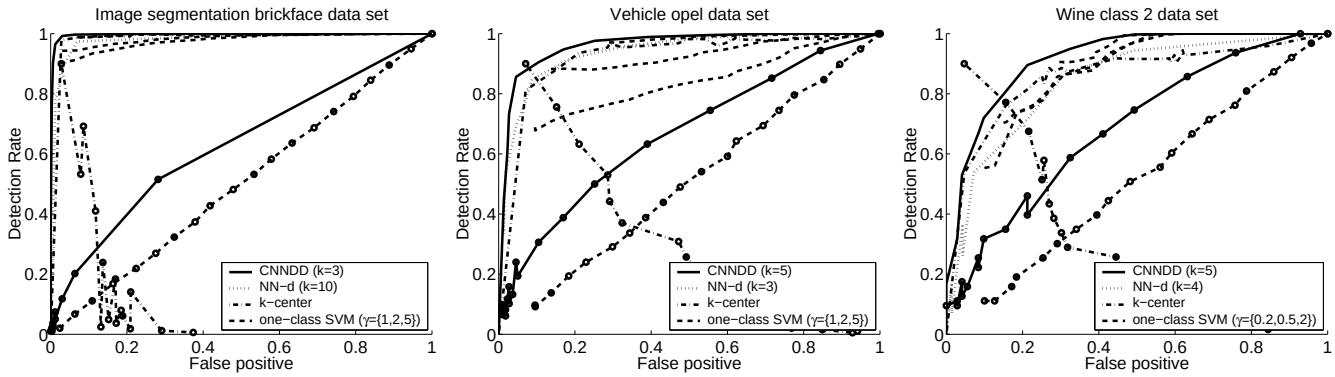

Fig. 6. Comparison of the CNNDD, NN-d, k-center, and one-class SVM methods: ROC curves and percentage of objects composing the model.

Otherwise, the points will lie mostly in one of the two above-defined regions. It is clear from these diagrams, that the CNNDD rule outperformed the other three methods in the experiments considered, while the NN-d, the k-center, and one-class SVM appear to be comparable on these experiments.

Figure 6 compares the ROC curves of the CNNDD (solid line), NN-d (dotted line), k-center (dash-dotted line), and one-class SVM (dashed lines) algorithms, and the compression ratios achieved by the CNNDD (solid pointed line), the k-center (dash-dotted pointed line), and one-class SVM (dashed pointed line).

These curves confirm the behavior described above. Indeed, the ROC curves of the NN-d and k-center methods were always below the ROC curve of the CNNDD method. As far as the one-class SVM, it is needed to increase the value of parameter γ of the RBF kernel in order to achieve a curve comparable to that of the CNNDD method. But, as parameter γ increased, we were not able to obtain a curve defined for all the values of false positive rate, in that the curve became undefined up to 5-10% of false positive rate. This is undesirable behavior, since usually this is just the range of values for the false positive rate one is interested in achieving after parameter tuning as a result of the training phase.

As far as the compression ratios are concerned, as expected the percentage of support vectors of the one-class SVM is approximatively equal to the false positive rate. It can be noticed that the number of objects composing the CNNDD reference subset was always greater than the number of support vectors. We were not able to realize whether this behavior is an intrinsic property of the minimum size NNDD reference consistent subset, or, otherwise, if it is due to the fact that the CNNDD rule is a greedy method computing an approximate solution, i.e. a reference consistent subset which is, in general, not minimum (recall that the problem of computing the minimum one is intractable). In any case, for low values of the false positive rate, the size of the reference consistent subset returned by the CNNDD method is slightly greater than the number of support vectors. Furthermore, it must be noticed that the lower compression ratio of the CNNDD is repaid by an ROC curve which is much more accurate. As for the compression ratio of the k-center method, it is clear that in order to obtain a good false positive/detection rate trade-off the number k of data objects to be employed as centers of the classifier is high, at least the 80% in the experiments reported in Figure 6.

Finally, Figure 7 shows the scaling behavior of the CNNDD algorithm and of the learning phase of the one-class SVM[4]. Execution times are those reported by the LibSVM implementation of the one-class SVM. We considered the *Satellite image* and *Shuttle* data sets since they are the largest data sets among those employed in the experiments.

As regards the one-class SVM, the curves reported are obtained by fixing the parameter ν to 0.1. As regards the CNNDD method, the value of θ is such that the CNNDD rule scores 0.1 false positive rate.

As for the *Satellite image* data set, the value of γ is that associated with the best detection rate achieved by the one-class SVM, while the value of k is that associated to the best detection rate achieved by the CNNDD rule.

As for the *Shuttle* data set, the curve for $k = 2$ of the CNNDD method (those associated with its best ROC area) is compared with the curve for $\gamma = 0.01$ of the one-class SVM, and also the curve for $k = 10$ of the former method with the curve for $\gamma = 0.1$ of the latter method.

It can be observed that, while the CNNDD algorithm seems to scale nearly linearly, training the one-class SVM is costly. Indeed, on the whole data set, the latter method can be one or two orders of magnitude slower than the former, while, owing to the different trend of the two curves, the ratio between the execution times of the two methods is even expected to increase if larger data sets are to be considered.

For example, on the *Shuttle* data set, composed of $34{,}108$ objects having 9 features each, the one-class SVM with parameters $\nu = 0.1$ and $\gamma = 0.01$ required about 170 seconds versus the nine seconds of the CNNDD

4. Experiments were performed on a Pentium Mobile 1700MHz based machine having 1GB of main memory and the Windows XP operating system.

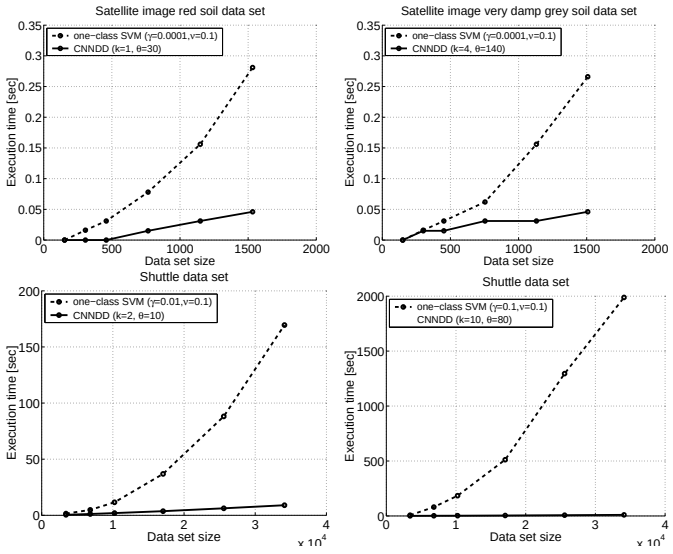

Fig. 7. Comparison of the execution times of the CNNDD and of the one-class SVM.

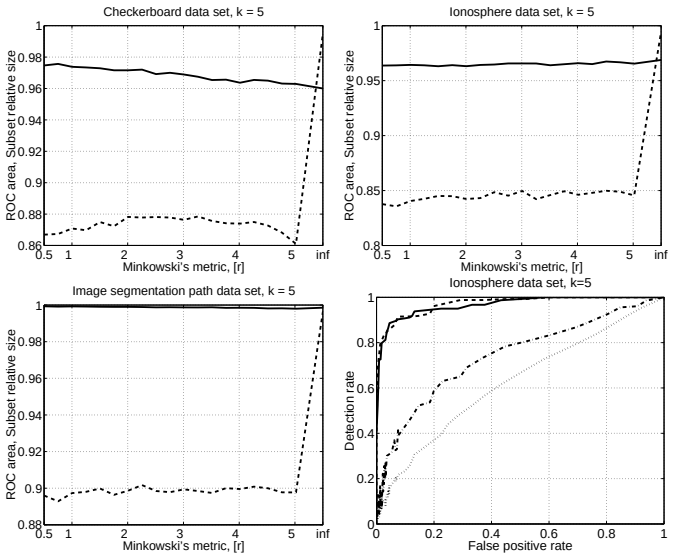

Fig. 8. Sensitivity to the Minkowski's metric r.

rule with parameters $k = 2$ and $\theta = 10$ (f.p.$= 0.1$). Furthermore, on the same data set the one-class SVM for $\nu = 0.1$ and $\gamma = 0.1$ required about $2{,}000$ seconds versus the ten seconds of the CNNDD rule for $k = 10$ and $\theta = 80$ (f.p.$= 0.1$).

5.3 Sensitivity to the norm and robustness to noise

In this section two important aspects of the CNNDD method are investigated, that is the effect of Minkowski's metric r used to compute the norm of the k nearest neighbors distances vector, and the robustness of the method to noise or outliers possibly belonging to the reference set.

First of all, the effect of norm r is considered. In order to measure the sensitivity of CNNDD to this parameter, r was varied in the range $[0.5, 5] \cup \{+\infty\}$, and both the area under the ROC curve, and the area under the curve of the false positive ratio versus the condensation ratio, say τ this area, were measured. Results concerning the *Checkerboard*, *Ionosphere*, and *Image segmentation path* data sets, are shown in Figure 8 ($k = 5$ was used in all the experiments). The solid curve represents the area under the ROC curve, while the dashed curve represents the ratio τ/τ_{max}, where τ_{max} denotes the greatest value of τ encountered.

As for the accuracy, while for the *Ionosphere* and for the *Image segmentation path* it remained practically unchanged, for the *Checkerboard* data set, by augmenting the value of r the accuracy gradually and slightly worsened. Thus, the accuracy of CNNDD appears to have low sensitivity to the choice of norm r. However, it must be noticed that for small values of r ($r \leq 1$) the accepting region of CNNDD follows the distribution of the reference set points more closely than for greater values of r. Since the two classes composing the *Checkerboard* data set are very close and the boundary is quite complex (it is the union of some axis parallel segments) this explains why for low values of r the method performed slightly well. The same behavior has not been observed in the other experiments since the classes are better separated.

As for the condensation ratio τ, for all the data sets, it remained almost constant for $r < +\infty$ (τ/τ_{max} is approximatively 0.85-0.9), while for $r = +\infty$ it reached its maximum value τ_{max}. Thus, it can be concluded that by using the infinity metrics the condensation ratio worsen. This behavior can be explained by noticing that, from the data reduction point of view, this norm is more demanding than the other norms. Indeed, under the $r = +\infty$ metrics, an object can be discarded from the reference set only if at least k objects lie within distance θ from it, while for any other metrics r, the same object can be discarded provided that there exists k objects such that the sum $\sum_i (d_i)^r$ of their distances d_i from it is less that $\theta^{1/r}$. It is evident that the former is a more severe condition than the latter. The value τ/τ_{max} is a short summary of the condensation achieved, but it does not represent a real ratio between sizes of reference consistent subsets (recall that it is an area). In order to visualize this ratio, Figure 8, on the right, reports, for $r = 1$ and $r = +\infty$, the ROC curves (solid $r = 1$, dashed $r = +\infty$), and the curves of τ (dotted $r = 1$, dash-dotted $r = +\infty$), obtained on the *Ionosphere* data set. While the ROC curves are very close, depending on the level of false positive rate, the ratio between the sizes of the consistent subsets may change noticeably. For example, for values of false positive rate around 0.1 the size of the subset doubles if infinity metrics is employed. A similar behavior was observed also on the other data sets. It can be concluded that the value $r = 1$, used in the rest of the paper, is in general a good choice for this parameter.

In order to study the robustness of the CNNDD method to noise, two experiments were executed.

First of all, the *Uniform* data set (10K two-dimensional

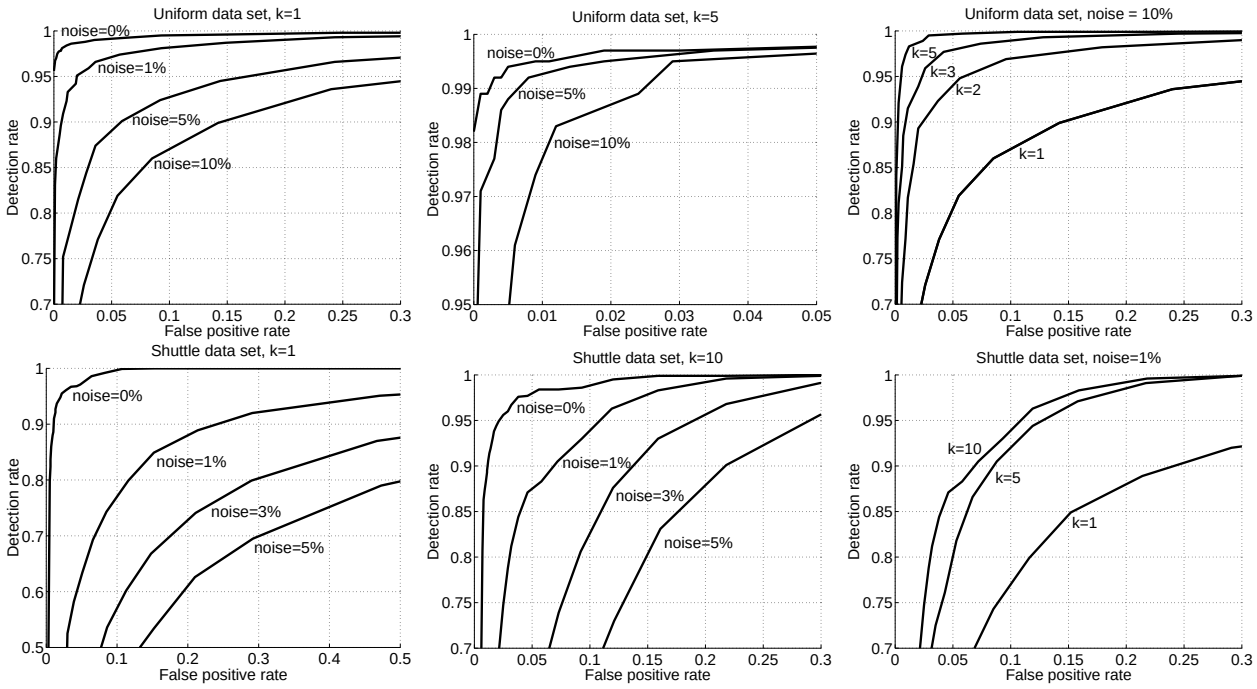

Fig. 9. Robustness to noise.

points) previously described was considered as training set. An inlier and an outlier test sets were generated to measure false positive rate and detection rate. The inlier data set is composed of 10K two-dimensional points into the square $A = [0.25, 0.75]^2$. The outlier data set is composed of 10K points into the region $B = [0, 1]^2 - A$. Then noisy versions of the *Uniform* training set were obtained by adding from the 1% (100 points) to the 10% (1,000 points) of white-like noise (randomly generated points belonging to B). For values of k ranging from 1 to 5, the ROC curves associated to the training sets with noise were computed. Figure 9 reports on the top these curves. On the top left there are curves for $k = 1$, while on the top center curves for $k = 5$. It is clear that by adding noise the accuracy of the CNNDD decreases, though even adding considerable amount of noise the rule produces a good classifier. Importantly, as the figure on the top right shows (relative to 10% of noise), by increasing the value of k the effect of the noise is mitigated and the resulting classifier is of remarkable quality.

An analogous experiment was executed on the *Shuttle* data set. The $34{,}108$ points of the *Rad Flow* class were equally partitioned in a training set and in an inlier test set (of $17{,}054$ points each), while the $9{,}392$ points of the other classes formed and outlier test set. Then noisy versions of the training set were obtained by adding from 1% (170 points) to 5% (850 points) of mislabelled points (randomly selected points belonging to the outlier test set). Also in this case, for values of k ranging from 1 to 10, the ROC curve associated with the training sets with outliers were computed (reported in Figure 9 on the bottom). On the bottom left there are curves for $k = 1$, on the bottom center curves for $k = 10$, while on the bottom right curves relative to 1% of noise. By observing these curves it is clear that the behavior of the method is unchanged even if "biased" noise is added. Furthermore, recall that in this second experiment the noisy points added are points belonging to the outlier test set, rather than points coming from the whole feature space.

It can be concluded that increasing the parameter k has the positive outcome of mitigating the impact of possibly noise and outliers, thus noticeable improving classifier accuracy. Indeed, as previously noted, by increasing k, the objects lying in the less densely populated regions of the feature space are rejected and no longer contribute to form the accepting region of the classifier.

6 Conclusions

In this paper the behavior of one-class classification based on a nearest neighbor training set consistent subset has been investigated. With this aim, the concept of reference consistent subset has been introduced, and the computational complexity of its computation has been investigated. A fast greedy algorithm, named CNNDD, was described that computes a reference consistent subset with only two reference set passes. Comprehensive experimental activity revealed strengths and weaknesses of the method.

As a future work, several extensions of the research done here are worthy of being examined, such as, for example, methods for computing subsets of a size close to the minimum one, the investigation of different definitions of subsets that may improve generalization, the application of the technique of condensation to other

nearest neighbor-based one-class classification methods, and exploring the power of feature selection in conjunction with training set condensation.

Acknowledgements. The author would like to thank the anonymous reviewers for their comments that greatly helped in improving the presentation of the work.

REFERENCES

[1] F. Angiulli. Fast condensed nearest neighbor rule. In *22nd International Conference on Machine Learning (ICML)*, Bonn, Germany, 7-11 August 2005.

[2] F. Angiulli and C. Pizzuti. Fast outlier detection in high-dimensional spaces. In *Proc. of the European Conf. on Principles and Practice of Knowledge Discovery in Databases*, pages 15–26, 2002.

[3] M. Breunig, H.P. Kriegel, R. Ng, and J. Sander. Lof: identifying density-based local outliers. In *Proc. of the ACM Int. Conf. on Management of Data*, 2000.

[4] V. Cerverón and F.J. Ferri. Another move toward the minimum consistent subset: A tabu search approach to the condensed nearest neighbor rule. *IEEE Transactions on Systems, Man, and Cybernetics–Part B: Cybernetics*, 31(3):408–304, 2001.

[5] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[6] T.M. Cover and P.E. Hart. Nearest neighbor pattern classification. *IEEE Trans. on Information Theory*, 13:21–27, 1967.

[7] B. Dasarathy. Minimal consistent subset (mcs) identification for optimal nearest neighbor decision systems design. *IEEE Trans. on Sys. Man. Cybernet.*, 24(3):511–517, 1994.

[8] L. Devroye. On the inequality of cover and hart. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 3:75–78, 1981.

[9] L. Devroye, L. Gyorfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer-Verlag, New York, 1996.

[10] C.L. Blake D.J. Newman, S. Hettich and C.J. Merz. UCI repository of machine learning databases, 1998.

[11] E. Eskin, A. Arnold, M. Prerau, L. Portnoy, and S. Stolfo. A geometric framework for unsupervised anomaly detection: detecting intrusions in unlabeled data. *Applications of Data Mining in Computer Security*, 2002.

[12] E. Fix and J. Hodges. Discriminatory analysis. non parametric discrimination: Consistency properties. In *Tech. Report 4*, USAF School of Aviation Medicine, Randolph Field, Texas, 1951.

[13] S. Floyd and M. Warmuth. Sample compression, learnability, and the vapnik-chervonenkis dimension. *Machine Learning*, 21(3):269–304, 1995.

[14] M.R. Garey and D.S. Johnson. *Computer and Intractability*. W. H. Freeman and Company, New York, 1979.

[15] P.E. Hart. The condensed nearest neighbor rule. *IEEE Trans. on Information Theory*, 14:515–516, 1968.

[16] D.S. Hochbaum and D.B. Shmoys. A best possible heuristic for the k-center problem. *Mathematics of operations research*, 10(2):180–184, 1985.

[17] E. Knorr and R. Ng. Algorithms for mining distance-based outliers in large datasets. In *Proc. of the Int. conf. on Very Large Databases*, pages 392–403, 1998.

[18] N. Littlestone and M. Warmuth. Relating data compression and learnability. Technical report, University of California, Santa Cruz, 1986.

[19] S. Ramaswamy, R. Rastogi, and K. Shim. Efficient algorithms for mining outliers from large data sets. In *Proc. ACM Int. Conf. on Managment of Data*, pages 427–438, 2000.

[20] G. Ritter and M.T. Gallegos. Outliers in statistical pattern recognition and an application to automatic chromosome classification. *Pattern Recognition Letters*, 18:525–539, April 1997.

[21] B. Schölkopf, C. Burges, and V. Vapnik. Extracting support data for a given task. In *Proc. of the Int. Conf. on Knowledge Discovery & Data Mining*, pages 251–256, Menlo Park, CA, 1995.

[22] B. Schölkopf, J. Platt, J. Shawe-Taylor, A.J. Smola, and R.C. Williamson. Estimating the support of a high-dimensional distribution. In *TR 87, Microsoft Research*, Redmond, WA, 1999.

[23] C. Stone. Consistent nonparametric regression. *Annals of Statistics*, 8:1348–1360, 1977.

[24] D. Tax and R. Duin. Data domain description using support vectors. In *Proc. of the European Symp. on Artificial Neural Networks*, pages 251–256, Bruges (Belgium), April 1999.

[25] D. Tax and R. Duin. Data descriptions in subspaces. In *Proc. of Int. Conf. on Pattern Recognition*, pages 672–675, 2000.

[26] D. M. J. Tax. *One-class classification*. PhD thesis, Delft University of Technology, June 2001.

[27] G. Toussaint. Proximity graphs for nearest neighbor decision rules: Recent progress. In *Tech. Report SOCS-02.5*, School of Computer Science, McGill University, Montréal, Québec, Canada, 2002.

[28] A. Ypma and R. Duin. Support objects for domain approximation. In *Proc of the ICANN*, 1998.