A Fast Hybrid Classification Algorithm based on the Minimum Distance and the k-NN Classifiers

Stefanos Ougiaroglou[*]
Dept. of Applied Informatics
University of Macedonia
156 Egnatia Street, 54006,
Thessaloniki, Greece
stoug@uom.gr

Georgios Evangelidis
Dept. of Applied Informatics
University of Macedonia
156 Egnatia Street, 54006,
Thessaloniki, Greece
gevan@uom.gr

Dimitris A. Dervos
Information Technology Dept.
Alexander T.E.I. of
Thessaloniki
57400, Sindos, Greece
dad@it.teithe.gr

ABSTRACT

Some of the most commonly used classifiers are based on the retrieval and examination of the k Nearest Neighbors of unclassified instances. However, since the size of datasets can be large, these classifiers are inapplicable when the time-costly sequential search over all instances is used to find the neighbors. The Minimum Distance Classifier is a very fast classification approach but it usually achieves much lower classification accuracy than the k-NN classifier. In this paper, a fast, hybrid and model-free classification algorithm is introduced that combines the Minimum Distance and the k-NN classifiers. The proposed algorithm aims at maximizing the reduction of computational cost, by keeping classification accuracy at a high level. The experimental results illustrate that the proposed approach can be applicable in dynamic, time-constrained environments.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Application – Data Mining

General Terms

Algorithms, Management, Experimentation

Keywords

classification, nearest neighbors, scalability, data reduction

1. INTRODUCTION

Classification is a data mining technique that attempts to map data to a set of classes [10]. A classification model or classifier is responsible for this mapping and can be evaluated by four criteria: (i) accuracy, (ii) scalability, (iii) robustness, and, (iv) interpretability. A major factor that the research on classification deals with is scalability, i.e., the ability of the classifier to work on large datasets. In this paper, the focus is on developing scalable classification algorithms without sacrificing accuracy.

There are two major categories of classifiers: eager and lazy [13]. Eager classifiers build a model based on the available data, and then, they classify all new instances using this model. In contrast, lazy classifiers do not build any model. They classify new instances by scanning the database at the time a new instance arrives. A typical example of a lazy classifier is the k-Nearest Neighbor (k-NN) classifier [6]. k-NN classification works by searching the available database (training set) for the k nearest items to the unclassified item. Then, the retrieved k nearest neighbors determine the class where the new instance belongs to. This class is the most common one among the classes of the retrieved k nearest neighbors. If ties occur, (two or more classes are most common) the new instance will be classified to the class determined either randomly or by the one nearest neighbor. The latter is the approach adopted in this work.

Despite all its advantages, the k-NN classifier involves one major drawback: it needs to compute all distances between a new instance and the training data. This property affects the scalability because as the size of the training set becomes larger, the cost increases linearly. Indexing methods can be used in order to reduce the cost of searching from linear to logarithmic time [3]. Unfortunately, index efficiency degrades with the increase of data dimensionality [18]. This phenomenon is known as the "dimensionality curse" and it is partially dealt with by applying a dimensionality reduction technique, such as Principal Component Analysis (PCA) [14]. However, there are cases where this approach cannot be applied successfully, or it leads to significant information loss, and, consequently, classification becomes less effective.

Other methods to reduce the k-NN classifier cost are the Data Reduction Techniques (DRT) [19, 17, 15]. There are two main algorithm categories of such techniques: (i) condensing algorithms that build a representative set that includes the close-border instances and has the advantage of low computational cost and small storage requirements, and (ii) editing algorithms that aim at improving the classification accuracy rather than reducing the computational cost. The main idea behind the condensing algorithms is that the "internal" instances can be removed without significant loss of accuracy, since they do not effect the decision boundaries. On the other hand, editing algorithms try to improve the

[*]Stefanos Ougiaroglou is supported by a scholarship from the State Scholarship Foundation of Greece (I.K.Y.)

SISAP '11, Jun 30 - Jul 01 2011, Lipari Island, Italy

accuracy by removing the noise and close-border instances leaving smother decision boundaries behind. Thus, the most typical ("internal") instances of a particular class are used to classify items near them.

Although DRTs are usually very effective, they introduce a complicated preprocessing step, since they build a model (condensing set) to speed-up the searching procedure. When classification tasks are applied to databases where frequent content changes occur, the repeated execution of this step, that ensures the effectiveness of the model, is prohibitive. In such dynamic environments, there is a need of a lazy, model-free classifier.

The Minimum Distance Classifier (MDC) [7] can be used as a very fast classification approach. MDC can be characterized as a very simple DRT since it computes a representative instance (centroid) for each one class. This is the vector obtained by averaging the attribute values of the items of each class. When a new instance t is to be classified, the distances between t and all centroids are calculated and t is classified to the class of the nearest centroid. MDC avoids the high computational cost of scanning the whole training data. On the other hand, MDC accuracy depends on how the items of the training set are distributed in the multidimensional space. The centroid based model introduced by MDC has been successfully applied for document categorization [9].

This paper introduces a new fast hybrid classification algorithm that does not need any speeding-up data structure (i.e. indexing) or any transformation of the training data (i.e. data reduction). The reduction of the computational cost is achieved by the combination of the k-NN and the Minimum Distance Classifiers. The proposed algorithm begins by computing a representative instance for each class. Then, it tries to classify new instances by examining the representative instances and by applying certain criteria. If the set criteria are not satisfied, it proceeds by applying k-NN search over the entire database.

The motivation is to address the problem of classifying large, high-dimensional datasets, where a sequential, exhaustive search of the whole dataset is required to locate the k nearest neighbors (conventional k-NN classifier). This is the case when indexing is not applicable and dimensionality reduction negatively affects the performance. Furthermore, complicated preprocessing procedures are avoided. The simple centroid based model of MDC involves only one pass over the training data to compute the mean vector of each class. The contribution of the work is summarized as follows:

- A novel, model-free classification algorithm is introduced which is independent of data dimensionality and avoids expensive preprocessing procedures on the training data, and, thus, it can be applied for repeated classification tasks in dynamic databases where frequent content changes occur, and,
- The main classification algorithm and two variations that extend its basic idea are considered in relation with the set goal for achieving high accuracy while reducing the computational cost as much as possible.

The rest of this paper is organized as follows. Section 2 presents related work, and Section 3 considers in detail the new proposed classification algorithm and its variations. In Section 4, experimental results based on real life datasets are presented, and the paper concludes in Section 5.

2. RELATED WORK

An important variation of the k-NN classifier is the distance-weighted k-NN [2]. It emphasizes on nearer neighbors since it weighs the contribution of the k-nearest neighbors according to their distance from the new sample. Consequently, it aims for improving the classification accuracy. In contrast, the present work focuses on the scalability factor.

Several data structures manage to efficiently index multidimensional data, and so, to speed-up the searching procedure. They are very effective when datasets with moderate dimensionality (e.g. 2-10) are used. In higher dimensions, the phenomenon of "dimensionality curse" renders those indexes irrelevant since their performance degrades rapidly and can become worse than that of the sequential scan of the whole database. The most significant multidimensional indexes are based on a tree data structure. Some of them are the R-Tree [8] and its variations, the KDB-Tree [16] and its variations, and the Vantage Point-Tree (VP-Tree) [20]. Moreover, the branch and bound algorithm proposed in [17] and enhanced in [5], and the incremental algorithm introduced in [12] are approaches that efficiently compute nearest neighbors using indexes of the R - Tree family. As already mentioned, all indexed-based approaches are dependent on data dimensionality. Many proposals for speeding up nearest neighbor searches rely on dimensionality reduction in order to effectively apply an index. A model on the effects dimensionality reduction has on the similarity search performance is presented in [1].

The first condensing data reduction algorithm was introduced by Hart [11]. Hart's algorithm, which is known as Condensing Nearest Neighbor (CNN) rule, attempts to reduce the computational cost of the k-NN classifier by selecting as representatives only the close border instances from the initial dataset. The number of selected instances is determined automatically and depends on the number of classes involved (the more the classes the more the close-border instances selected) and the level of noise in the data. These two parameters determine the computational cost of the CNN reduction procedure and the effectiveness of the condensing set produced. CNN rule tries to keep only the close-border items in the following way: It uses two bins, S and T. Initially, an item of the training data is placed in S and the remaining items are placed in T. Then, the algorithm tries to classify the content of T using the content of S. Whenever an item of T is misclassified, it is moved to S. The algorithms terminates when there is no move form T to S during a complete pass of T. The items that have been moved to S constitute the condensing set which will be used to classify new samples. Although, many CNN variations have been proposed, Hart's algorithm is the reference data reduction technique.

Many other researches focused and elaborated on condensing DRTs; Chen's algorithm [4] and the Learning Vector Quantization (LVQ) algorithms are the most well known cases. In contrast to Hart's algorithm, these methods generate new items that represent the initial dataset. The number of instances produced is determined by the user. Detailed reviews on DRTs can be found in [19], [17] and [15]. Contrary to the MDC, DRT procedures involve high computational cost and, thus, they are ineffective in dynamic environments that require frequent reconstruction of the condensing set.

3. THE PROPOSED METHOD

This section presents a fast, hybrid and model-free classification algorithm. In addition to the main algorithm, two variations that achieve extra computational cost savings are proposed. In the order presented, each one variation comprises an extension to its predecessor. As expected, the improvement comes at the cost of a decrease in accuracy.

The algorithm and its variations are based on the same idea: They initially search for nearest neighbors in a new, smaller dataset constructed by one pass over the training data. This dataset includes only a representative instance for each one class. Upon failure to meet the set acceptance criteria, classification proceeds by the conventional k-NN classifier. Each representative instance is computed by calculating the average value of each attribute in each one class. Thus, the computed vector can be considered to comprise the centroid of the cluster corresponding to the class. Therefore, if the initial dataset includes 1000 items in 15 dimensions and 10 classes, the new dataset will have only 10 items in 15 dimensions. Evidently, the more the dataset of centroids is used, the less the execution time involved. Subsections 3.1, 3.2, and 3.3 below, outline the main classification algorithm and its two variations.

3.1 Fast Hybrid Classification Algorithm

The Fast Hybrid Classification Algorithm (FHCA) is based on the difference of the distances between the new unclassified item and the representative centroids (see Figure 1). More specifically, for each new (incoming, unclassified) item x, the algorithm takes into consideration the two nearest centroids A, B (A is the nearest and B is the second nearest) and their distances from x: d(x, A), d(x, B). If the difference between d(x, A) and d(x, B) exceeds a predefined threshold (line 5), x is classified to belong to the class represented by centroid A (line 6), otherwise the k nearest neighbors are retrieved from the initial training set in order to determine the class of x (lines 7-10).

FHCA is more accurate than the two variations presented in subsections 3.2 and 3.3. In some cases, it reaches the accuracy of conventional k-NN classifier, at a significantly less computational cost. However, the performance depends on the value of the predefined threshold.

When the threshold parameter is set to zero, the centroid-based approach classifies all the new samples (since the “if” condition calculates to “true”). On the other hand, if the threshold is set to a relatively high value, it is possible that all new items are classified by the conventional k-NN classifier (the “else” clause in line 7 of Figure 1). These properties indicate that the threshold value adjustment should be made carefully, as it comprises a computational cost vs. accuracy trade-off decision for the application considered.

3.2 FHCA - Variation I

The first FHCA variation (FHCA-V1), illustrated in Figure 2, is an extension of the main FHCA algorithm, since it uses the distance difference criterion the same way FHCA does. In addition, FHCA-V1 attempts to classify even more new incoming items without falling back to the conventional k-NN classifier. In particular, if the distance difference criterion fails to classify the incoming item x (lines 6 and 7 in Figure 2), FHCA-V1 calculates the region of influence of each one class centroid involved. We define the class region of influence to be the average distance of the training set class items from the corresponding class centroid. In case x lies within the region of influence of only one class (class A in Figure 3), x is classified to belong to the class in question (lines 8 and 9 in Figure 2). Otherwise, if x lies within the region of influence of more than one class, the algorithm proceeds as in the conventional k-NN classifier (lines 10-13 in Figure 2). In practice, the only one difference between FHCA and FHCA-V1 is the “else if” part of pseudo-code in lines 8 and 9 of Figure 2.

```
FHCA (Threshold, k)
1.  Scan the training data to compute the class centroids
2.  For each unclassified item $x$ do
3.      Compute the distances between $x$ and the class centroids
4.      Find the nearest centroid $A$, and the second nearest centroid $B$,
        using the Euclidian distance metric
5.      If (distance($x$, $B$) - distance($x$, $A$)) $\geq$ Threshold then
6.          Classify $x$ to the class of centroid $A$
7.      else
8.          Retrieve the $k$ NNs from the initial training data
9.          Find the major class (the most common one among the $k$
            NNs. In case of a tie, it is the class of the Nearest Neighbor)
10.         Classify $x$ to the major class
11.     endif
12. endfor
```

Figure 1: Fast Hybrid Classification Algorithm

Contrary to the FHCA, FHCA-V1 requires two preprocessing passes, one for calculating the class representative instances (centroids, as in FHCA: line 1 in Figures 1 and 2), and one for calculating the class regions of influence in the training set data (line 2 in Figure 2). Even this extra pass over the training data is insignificant compared to the complicated preprocessing procedures involved by the data reduction techniques.

3.3 FHCA - Variation II

The second FHCA variation (FHCA-V2) extends FHCA-V1 to include one more classification criterion. The latter handles the case where the unclassified item x lies within more than one class regions of influence (Figure 5, lines 10 and 11 in Figure 4). In this case, x is classified to the class of the nearest centroid whose region of influence embraces it. The example in Figure 5 illustrates such a case: suppose that the distance difference criterion is not able to classify x, i.e. x lies closer to A than to B, but the difference between the two distances does not reach the predefined threshold. Also, suppose that x lies within the regions of influence of both A and B. In this case, FHCA-V2 classifies x to belong to the class of the nearest centroid (the class of centroid A in Figure 5), skipping the computational cost of the conventional k-NN classifier. It is only in cases where unclassified instances fail to meet both of the FHCA-V1 and FHCA-V2 set criteria that the algorithm proceeds to apply conventional k-NN classifier. In this respect, FHCA-V2 involves less computational overhead when compared to FHCA, FHCA-V1, and, of course, the conventional k-NN classifier.

3.4 Discussion

A key factor for the proposed classifier and its variations is the adjustment of the threshold (input) parameter. In the case of FHCA, the value of this parameter influences the

```
FHCA-V1 (Threshold, k)
1.  Scan the training data to compute the class centroids
2.  Re-scan the training data to compute the region of influence of
    each one class centroid
3.  For each unclassified item x do
4.      Compute the distances between x and the class centroids
5.      Find the nearest centroid A, and the second nearest centroid B,
        using the Euclidian distance metric
6.      If (distance(x, B) - distance(x, A)) ≥ Threshold then
7.          Classify x to the class of centroid A
8.      else if x belongs to the region of influence of only one class then
9.          Classify x to this class
10.     else
11.         Retrieve the k NNs from the initial training data
12.         Find the major class (the most common one among the k NNs.
            In case of a tie, it is the class of the Nearest Neighbour)
13.         Classify x to the major class
14.     endif
15. endfor
```

Figure 2: FHCA - Variation I

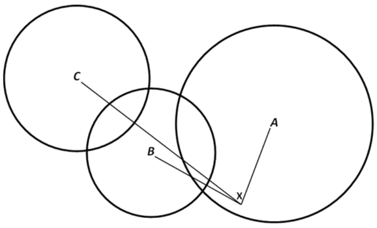

Figure 3: FHCA-V1 classification case

number of new incoming instances classified by the centroid-based model: the more the centroid-based approach is used, the less is the computational cost involved. In the cases of FHCA-V1 and FHCA-V2, the focus is on the incoming (unclassified) instances that cannot be classified by the distance difference criterion. Two additional centroid-based classification criteria are introduced, in an attempt to avoid the computational cost of the conventional k-NN classifier. The latter becomes the only one option available when both of the FHCA-V1 and FHCA-V2 set criteria fail to classify the unclassified instance. Obviously, FHCA-V2 utilizes the centroids dataset as much as possible (i.e. in all three set criteria) and represents the fastest variation. In contrast, FHCA is the slowest of the three approaches, since it applies centroid-based classification solely on the basis of the distance difference criterion.

A threshold auto-adjustment method is relatively easy to implement in the form of a routine that accepts a value for the desirable accuracy level, and it iteratively considers a number of different threshold values. Having reached the desirable accuracy level, the routine returns the corresponding threshold value.

It is noted that the worst-case scenario for the proposed classification approaches is when the centroid-based part does not classify any item of the testing set. In this case, the execution time involves the k-NN figure, the small overhead of the centroids creation (one pass of the training data) and the small overhead introduced by the cost of distance computations between testing data and the class centroids (e.g. if there are ten classes, ten distances must be computed for each item of the testing set).

```
FHCA-V2 (Threshold, k)
{pseudo-code lines 1-7 are as in Figure 2}
8.  else if x belongs to the region of influence of only one class then
9.      Classify x to this class
10. else if x belongs to the regions of influence of more than one class then
11.     Classify x to the class of nearest centroid whose region of influence
        embraces x
12. else
{pseudo-code lines 13-17 are as the lines 11-15 in Figure 2}
```

Figure 4: FHCA - Variation II

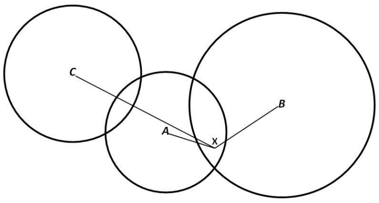

Figure 5: FHCA-V2 classification case

The proposed algorithm can be modified so that upon failure to use the centroid-model, the k-NN part takes into account only the items of the two classes with the nearest centroids to the query point. In this case, both the computational cost and the classification accuracy are highly depended on the number of classes. Experiments conducted on the datasets presented in Section 4, have shown that although this approach is very fast, the accuracy is significantly reduced.

4. PERFORMANCE EVALUATION

4.1 Datasets

The proposed algorithms were tested using five real life datasets distributed by the UCI Machine Learning Repository[1]. Using the filters provided by the UCI web interface, care was taken to retrieve the largest, appropriate for classification datasets. Table 1 summarizes the datasets used. The last table column lists the k value found to achieve the highest accuracy. Since, the k-NN classifier requires the computation of all distances between each item of the testing set and the training data, the computational cost of k-NN classifier can be easily estimated by multiplying the contents of second table column by the contents of third column. For example, 15,000 * 5,000 = 75,000,000 distances are computed for the letter recognition dataset.

In addition, classification tasks are usually applied to noisy training data. The removal of noise, introduces an extra preprocessing procedure. However, we are interested in developing classifiers that do not need any preprocessing task (either to remove the noise or to build a speed-up model such as condensing set or indexing structures).

To evaluate the performance of the proposed algorithm on noisy data, three more datasets were developed by adding random noise to three of the datasets of Table 1. Particularly, 40% of noise was added to the Letter recognition, Pendigits and Landsat satellite datasets. Namely, for each item of the training set of these datasets, the class attribute

[1] http://archive.ics.uci.edu/ml/

Table 1: Datasets description

Dataset	Training data	Testing data	Attributes	Classes	Best k
Letter recognition	15000	5000	16	26	4
Magic gamma telescope	14000	5020	10	2	12
Pendigits	7494	3498	16	10	4
Landsat sattelite	4435	2000	36	6	4
Shuttle	43500	14000	9	7	2

Table 2: T parameter values

Dataset	FHCA(T_1)	FHCA(T_2)	FHCA-V1(T_1)	FHCA-V1(T_2)	FHCA-V2
Letter recognition	1.6	0.8	1.7	0.7	1.4
Magic gamma telescope	17	10	16	6	14
Pendigits	44	18	35	17	35
Landsat sattelite	31	13	31	8	34
Shuttle	30	25	35	28	25
Letter recognition (noisy)	0.9	0.5	0.9	0.5	1.3
Pendigits (noisy)	26	13	33	12	14
Landsate satellite (noisy)	19	14	21	12	10

Table 3: Hart's Condensing NN Rule

Dataset	Condensing data	Reduction (%).	Computations	Best k
Letter recognition	2517	83.22	145,386,010	1
Magic gamma telescope	5689	59.36	217,900,759	22
Pendigits	312	95.84	7,940,953	1
Landsat sattelite	909	79.50	13,545,272	6
Shuttle	300	99.31	57,958,973	1
Letter recognition (noisy)	11806	21.29	205,175,615	16
Pendigits (noisy)	5822	22.31	51,891,038	20
Landsat satellite (noisy)	3469	21.78	17,918,173	30

was modified with a probability of 0.4. By executing the conventional k-NN Classifier on these noisy datasets, it was found that the highest accuracy is achieved for k=13, k=18 and k=21 respectively (the more the added noise, the higher the value of k needed to achieve the highest accuracy). The other two datasets were not transformed into a noisy mode, since the Magic telescope dataset has already a high level of noise and the Shuttle dataset is a skewed dataset with two very rare classes and approximately 80% of the data belonging to one of the seven classes.

4.2 Experimental setup

The proposed algorithm and its variations use the distance difference Threshold (T) as a parameter. This parameter should be adjusted, so that the desirable trade-off between accuracy and computational cost can be achieved. For this reason, several experiments with different T values were conducted. For FHCA and FHCA-V1, two T values are reported (T_1 and T_2). The first value builds an accurate classifier (comparable to the conventional k-NN classifier, when possible), and the second value builds a faster classifier which achieves up to 5% lower accuracy than the one build with the first value. For FHCA-V2, which is the fastest approach, only the T value that achieves the highest accuracy is reported. Table 2 presents the T values for each dataset.

FHCA and its variations involve the same centroid-based part of pseudocode (lines 5, 6 in Figure 1 and lines 6, 7 in Figures 2) which, effectively, comprises a Minimum Distance Classifier component. In this respect, the MDC component classifies the same number of items in each one testing set, despite the classification approach used (FHCA, FHCA-V1 or FHCA-V2). For comparison purposes, the experimental measurements of a 'Minimum Distance Classifier, only' approach are included in Table 4 of the following subsection.

Furthermore, for comparison purposes, the Hart's Condensing Nearest Neighbor (CNN) rule [11] was implemented. Table 3 presents the experimental results obtained from the execution of the CNN reduction procedure on the eight datasets (the five real life and the three noisy datasets). Column four lists the number of distance computations needed for the production of the condensing set. The last column lists the k value found to achieve the highest accuracy when the resulting condensing set is used to classify the items of the corresponding testing sets. Table 4 includes the performance of CNN: accuracy and computational cost obtained from the execution of k-NN classifier on the CNN condensing set (CNN k-NN). Of course, these values do not include the computational cost (fourth column in Table3) introduced by the condensing set construction (preprocessing step on the available training data).

4.3 Comparisons

In this subsection, the proposed classifier and its variations are compared to each other and to the conventional k-NN, Minimum Distance, and, CNN k-NN classification algorithms, by setting the T parameters values presented in Table 2. For each one dataset, two experimental measurements were taken for each one classification approach: (i) classification accuracy, and (ii) distance computations as a percentage of the distance computations needed for the conventional k-NN. To give a feeling of the actual computational cost, for the conventional k-NN this second measurement represents the actual number of distance computations.

Due to space restrictions, experimental results obtained by varying the value of T are not presented[2].

Letter recognition dataset: FHCA almost reached the accuracy level of k-NN, when the threshold value was set to T=1.6. In particular, FHCA was found to achieve an accuracy of 95.24% and a 15.6% reduction in the computational cost. To obtain a faster classifier, one should decrease the threshold value (T). For instance, when T was set to 0.8, FHCA achieved an accuracy of 90.78%, with 35% lower cost than that of k-NN. FHCA-V1 was affected by the extended overlapping of centroid regions of influence in the given dataset. As a result, the centroid-based part of FHCA-V1 managed to classify only a few more items (8%–10%) than that of FHCA. For T=1.6, FHCA-V1 was measured to classify the testing data with an accuracy of 91.96% and 25% less computational cost. Finally, for T=1.6, FHCA-V2 needed only 27% of the distance computations. However, the accuracy was only 71.6%.

Magic gamma telescope dataset: In this dataset, the proposed algorithm and its variations were measured to perform better than in the previous dataset. In addition, their measurements are comparable to these of CNN k-NN. Concerning FHCA, T=17 comprises a good choice for the threshold parameter, since FHCA achieves an accuracy value of over 80% while it reduces the cost by almost 56%. Although not shown in Table 4, for T=38, FHCA can achieve an accuracy of 81.36% that is very close to the accuracy of k-NN but with only a 10% improvement in computational cost. A fast classifier can be developed by setting T=10. In this case, FHCA makes predictions with an accuracy of 75.26% with only 23.48% of the k-NN cost. FHCA-V1 achieved its best accuracy performance (74.72%) for T=16, with 28.98% of the computational cost. For T=6, FHCA-V1 has only 9.64% of the k-NN cost (90.36% reduction) and classifies the testing data with an accuracy of 72%. FHCA-V2 was found to never exceed the accuracy value of 73% (for T=14, accuracy: 72.39%). However, FHCA-V2 executed very fast (for T=14, the cost was reduced by almost 90%).

Pendigits dataset: Concerning FHCA, two reference-worthy experimental measurements are obtained by setting T=44 and T=18. These adjustments achieved an accuracy of 97.08% and 92.02% respectively and had 62.74% and 30.89% of the k-NN cost. It is noted that for T=55, FHCA reached the k-NN accuracy, executing with almost 23% lower cost. Furthermore, for T=32, the cost is 51.4% and the accuracy 95%. FHCA-V1 achieved its best accuracy (88.54%) with 32.2% of the cost for T=35. For T=17, FHCA-V1 achieved an accuracy of 87.22% and required almost 25% of the k-NN cost. FHCA-V2 was measured to execute faster. For T=35, it needs 19.92% of the distance computations of k-NN and it achieves an accuracy of 86.54%.

Landsat satellite dataset: FHCA performed a little better than k-NN. More specifically, for T=38, it achieved the best possible accuracy value (90.85%), with a 32.8% decrease in computations. For all other threshold values, it fell a little behind in accuracy, but executed even faster. For instance, by setting T=31, FHCA achieved an accuracy of 90.05% (very close to the k-NN accuracy) and spent 57.03% of the k-NN computational cost. Also, for T=13, only 25% of the cost was required (accuracy: 85.1%). Finally, for T=26, FHCA achieved an accuracy of 89% with almost half the cost. The two variations performed almost the same in accuracy, with FHCA-V1 executing faster than FHCA-V2. For T=34, the two variations achieved their best accuracy levels (83.05% and 82.4%) and had 67% and 80% lower cost than k-NN, respectively.

Shuttle dataset: Shuttle is an imbalanced (skewed) dataset. Approximately 80% of the data belongs to one class. Therefore the default accuracy is 80%. When classification tasks execute against such datasets, the main goal is to obtain a very high accuracy (e.g. over 99%). As shown in Table 4, this goal is successfully fulfilled by k-NN and CNN k-NN. Additionally, to the very high accuracy that it achieved, CNN k-NN executed extremely fast, since it scanned a very small condensing set (only 300 items). This happened because the CNN-rule managed to reduce the training data at the minimum level.

For the dataset in question, FHCA achieved an accuracy of 99.82% with almost the half the cost (53.23%) of k-NN, for T=30. By setting T=25, the was reduced by over 60%, but the accuracy fell to 99.19%. It is worth mentioning that FHCA achieved its best accuracy (99.84%) by setting T=37 and had a cost reduction by 15%. The two variations did not achieve a reference-worthy accuracy. However, the proposed algorithm and its variations managed to classify with high accuracy the testing set items belonging to rare classes using their centroid-based part. There are many application domains where the correct prediction of rare classes is very critical (e.g. earthquake prediction, rare diseases, etc). In the shuttle dataset, there are 2 very rare classes both having only 17 items in the training set and 6 items in the testing set. For any T value, FHCA, FHCA-V1, and FHCA-V2 made 5 correct and 1 incorrect predictions.

Letter recognition dataset (noisy): For all noisy datasets, CNN k-NN and FHCA-V2 were affected by the addition of noise. The CNN-rule did not manage to drastically reduce the training (noisy) sets, and so, the computational cost gains were not significant. In contrast, the experimental results showed that FHCA and FHCA-V1 were not affected. In particular, for T=0.5, the FHCA accuracy was 86.06% and its cost was 35.31% lower than k-NN. For T=0.9, the accuracy was 91.06% and the cost was 16.95% lower than k-NN. FHCA achieved even better accuracy, but the cost savings were not significant. On the other hand, FHCA-V1 had an accuracy of 89.14% and 84.36%, for T=0.9 and T=0.5 respectively. The corresponding cost savings were 21.54% and 38.29%.

Pendigits dataset (noisy): On this dataset, FHCA reached the accuracy level of CNN k-NN at a 10% lower cost. Considering the additional high cost introduced by the construction of the condensing set of the CNN approach, the cost gains are actually much higher for FHCA. Moreover, FHCA reached the accuracy of k-NN with the same cost as CNN k-NN. Finally, for T=13, FHCA achieved an accuracy of 91.71% and had only 38.75% of the k-NN cost. Similarly, FHCA-V1 achieved an accuracy of 93.31% and 88.65% by setting T=33 and T=12 respectively. The corresponding savings in computational cost were 33.26% and 70.77% re-

[2] Detailed experimental results and diagrams available at: `https://sites.google.com/site/fhcalgo/files/FHCA.zip`

Table 4: Experimental results

Dataset		FHCA (T_1)	FHCA (T_2)	FHCA-V1(T_1)	FHCA-V1(T_2)	FHCA-V2	CNN k-NN	MDC	k-NN
Letter	Acc.:	95.24	90.78	92.06	87.00	71.46	91.9	58.08	95.68
recognition	Cost:	84.39	64.93	76.63	55.15	27.33	16.78	0.17	75,000,000
Magic gamma	Acc.:	80.02	75.26	74.72	72.00	72.39	80.64	68.92	81.39
telescope	Cost:	44.11	23.48	28.98	9.64	10.34	40.66	0.01	70,230,000
Pendigits	Acc.:	97.08	92.02	88.54	87.22	86.54	96.05	77.76	97.88
	Cost:	62.74	30.89	32.2	20.40	19.92	4.16	0.13	26,214,012
Landsat	Acc.:	90.05	85.1	83.00	80.70	82.40	89.75	77.50	90.75
satelite	Cost:	57.03	25.38	30.83	10.13	20.28	20.50	0.14	8,870,000
Shuttle	Acc.:	99.82	98.19	95.15	95.12	81.57	99.85	79.57	99.88
	Cost:	53.23	39.77	43.44	35.06	11.29	0.7	0.02	630,750,000
Letter	Acc.:	91.06	86.06	89.14	84.36	62.72	90.32	53.98	91.82
recogn. (noisy)	Cost:	83.05	64.69	78.47	61.71	21.47	78.71	0.17	75,000,000
Pendigits	Acc.:	96.17	91.71	93.31	88.65	78.7	96.20	75.90	97.00
(noisy)	Cost:	67.88	38.73	66.74	29.23	4.85	77.69	0.13	26,214,012
Landsat sat.	Acc.:	87.80	85.05	86.55	82.30	75.05	87.6	71.40	88.30
(noisy)	Cost:	63.33	47.58	63.13	36.08	8.28	78.22	0.14	8,870,000

spectively.

Landsat satellite dataset (noisy): The results are similar to the ones obtained on the Pendigits (noisy). FHCA had higher accuracy than CNN k-NN with 15% less cost. For T=21, FHCA-V1 had an accuracy of 86.55% with 63.17% of the cost. Finally, for T=12, FHCA-V1 achieved an accuracy of 82.3% and only 36.08% of the k-NN cost.

4.4 Discussion

For the datasets (i) Letter recognition, (ii) Pendigits, (iii) Landsat satellite, and, (iv) Shuttle, the proposed algorithms seem to be slower than CNN k-NN, however they are model-free, since they do not need any speed-up model produced by costly preprocessing procedures. The calculation of the class centroids is quite simple and inexpensive and can be executed before each classification task to take into account the latest database changes.

Furthermore, in the case of the Magic gamma telescope dataset, FHCA reached the accuracy of CNN k-NN with the same computational cost. This is attributed to the noise that exists in this dataset.

As expected, Hart's CNN rule was affected by the addition of noise. The preprocessing procedure of the CNN rule could not significantly reduce the items of the noisy datasets. Thus, for these datasets, in addition to the cost introduced by the condensing set construction, the sequential search of the condensing set involved a relatively high computational cost. Contrary to CNN, FHCA and FHCA-V1 were not significantly affected by the noise. The experimental results on the three noisy datasets showed that the later approaches manage to reach and exceed the CNN k-NN accuracy at a lower computational cost. FHCA-V2 is also affected by the addition of noise. This is because the third classification criterion, which handles the cases where the new item lies within more than one class regions of influence, cannot make predictions with high accuracy.

Last but not least, it should be noted that contrary to CNN, the adaptive schema offered by the proposed approach allows for the development of classifiers that reach the accuracy of the conventional k-NN classifier with significant savings in the computational cost.

5. CONCLUSIONS

In this paper, a fast, hybrid and model-free classifier is proposed. Speed-up is achieved by combining the Minimum Distance and the k-NN classifiers. Initially, the fast centroid-based model of MDC attempts to classify the new incoming instance. Upon failure, the new instance is classified via the k-NN approach. Although the proposed approach is quite simple, it manages to speed-up the classification process and can be useful in cases where data updates are frequent, thus, preprocessing of the training data for data reduction is prohibitive, or multidimensional index construction involving dimensionality reduction does not achieve acceptable classification accuracy.

Performance evaluation results show that significant computational cost reduction can be achieved, whereas, accuracy remains at high levels. In particular, the main classification algorithm (FHCA) met our expectations since it reached the accuracy level of the k-NN classifier and was not affected by noise. The two proposed variations of FHCA can be used in applications where there is a need for a less accurate but very fast classification approach. The decision on which of the three variations should be used and which threshold value is the most appropriate one depends on the application domain. Namely, these decisions should be made by taking into consideration the most critical parameter, i.e., the trade-off between accuracy and computational cost.

The effectiveness of the centroid-based model that the proposed classifier uses in order to speed-up the classification procedure is depended on the data distribution in the multidimensional space. In particular, it can be affected by the shape and the size of the clusters that the items of each class form. Next, we intend to address this issue by using more than one representative instances for each class. We intend to develop fast, hybrid and accurate classification algorithms that will not be depended on data distribution and dimensionality.

6. REFERENCES

[1] C. C. Aggarwal. On the effects of dimensionality reduction on high dimensional similarity search. In

Proc. of the ACM Principles of Database Systems (PODS 01), pages 256–266. ACM Press, 2001.

[2] C. G. Atkeson, A. W. Moore, and S. Schaal. Locally weighted learning. *Artificial Intelligence Review, 11:(1-5)*, pages 11–73, 1997.

[3] C. Boehm and F. Krebs. The k-nearest neighbor join: Turbo charging the kdd process. *Knowledge and Information Systems (KAIS)*, 6(6):728–749, 2004.

[4] C. H. Chen and A. JÃşzwik. A sample set condensation algorithm for the class sensitive artificial neural network. *Pattern Recognition Letters*, 17:819–823, 1996.

[5] K. L. Cheung and A. W. Fu. Distance browsing in spatial databases. *ACM Transactions on Database Systems*, 24(2):71–79, 1999.

[6] B. V. Dasarathy. Nearest neighbor (nn) norms: nn pattern classification techniques. *IEEE CS Press*, 1991.

[7] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. John Wiley and Sons, New York, 1973.

[8] A. Guttman. R-trees: A dynamic index for geometric data. In *Proc. ACM SIGMOD Int'l Conf. Management of Data*, pages 47–57, 1984.

[9] E. Han and G. Karypis. Centroid-based document classification: Analysis and experimental results. In *Proc. Fourth Principles and Practice of Knowledge Discovery in Databases (PKDD 2000)*, Lyon, France.

[10] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2000.

[11] P. E. Hart. The condensed nearest neighbor rule. *IEEE Trans on Information Theory*, 14(5):505–516, 1968.

[12] G. R. Hjaltason and H. Samet. Distance browsing in spatial databases. *ACM Transactions on Database Systems*, 24(2):71–79, 1999.

[13] M. James. *Classification Algorithms*. John Wiley and Sons Inc., 1985.

[14] I. T. Jolliffe. *Principal Component Analysis, 2nd ed.* Springer Series in Statistics, 2002.

[15] M. T. Lozano. *Data Reduction Techniques in Classification processes (Phd Thesis)*. Universitat Jaume I, 2007.

[16] J. T. Robinson. The k-d-b-tree: a search structure for large multidimensional dynamic indexes. In *Proceedings of the 1981 ACM SIGMOD international conference on Management of data*, SIGMOD '81, pages 10–18, New York, NY, USA, 1981. ACM.

[17] G. T. Toussaint. Proximity graphs for nearest neighbor decision rules: Recent progress. In *Proc. Interface-2002, 34th Symp. Computing and Statistics*, pages 83–106, 2002.

[18] R. Weber, H. Schek, , and S. Blott. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In *Proc. 24th Int'l Conf. Very Large Data Bases (VLDB 98)*, pages 194–205, 1998.

[19] D. R. Wilson and T. R. Martinez. Reduction techniques for instance-based learning algorithms. *Machine Learning*, 38:257–286, 2000.

[20] P. N. Yianilos. Data structures and algorithms for nearest neighbor search in general metric spaces. In *Proc. of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, page 311âĂŞ321, 1993.