Proc. 10th Int. DASFAA, Bejing, China, 2005.

Multi-represented *k*NN-Classification for Large Class Sets *

Hans-Peter Kriegel, Alexey Pryakhin, and Matthias Schubert

Institute for Computer Science
University of Munich
Oettingenstr. 67, 80538 Munich, Germany
{kriegel, pryakhin, schubert}@dbs.ifi.lmu.de

Abstract. The amount of stored information in modern database applications increased tremendously in recent years. Besides their sheer amount, the stored data objects are also more and more complex. Therefore, classification of these complex objects is an important data mining task that yields several new challenges. In many applications, the data objects provide multiple representations. E.g. proteins can be described by text, amino acid sequences or 3D structures. Additionally, many real-world applications need to distinguish thousands of classes. Last but not least, many complex objects are not directly expressible by feature vectors. To cope with all these requirements, we introduce a novel approach to classification of multi-represented objects that is capable to distinguish large numbers of classes. Our method is based on k nearest neighbor classification and employs density-based clustering as a new approach to reduce the training instances for instance-based classification. To predict the most likely class, our classifier employs a new method to use several object representations for making accurate class predictions. The introduced method is evaluated by classifying proteins according to the classes of Gene Ontology, one of the most established class systems for biomolecules that comprises several thousand classes.

Keywords: Multi-represented objects, classification, instance based learning, k nearest neighbor classifier.

1 Introduction

Modern information systems are collecting enormous amounts of data every day. In addition to the sheer amount of data, the complexity of data objects increases as well. Companies store more detailed information about their costumers, satellites take pictures with additional frequency spectra, and HTML-documents provide embedded multimedia content which makes them much more complicated

* Supported by the German Ministery for Education, Science, Research and Technology (BMBF) under grant no. 031U212 within the BFAM (Bioinformatics for the Functional Analysis of Mammalian Genomes) project which is part of the German Genome Analysis Network (NGFN).

than ordinary text documents. The analysis of large collections of complex objects yields several new challenges to data mining algorithms.

One of the most important tasks of data mining is classification. Classification learns a function $Cl : O \rightarrow C$ that maps each object $o \in O$ to the class $c \in C$ that it most likely belongs to. The class set C is a predefined set of categories. In order to make a class prediction, a classifier has to be trained. For the classification of complex objects, there are various important applications, e.g. the classification of proteins into functional catalogues or secure personal identification using several biometric characteristics. These applications yield interesting challenges to novel classification techniques.

First of all, the more complex a data object is, the more feature transformations exist that can be used to map the object to a representation suitable for data mining. Furthermore, many objects are describable by different aspects, e.g. proteins can be described by text annotations and amino acid sequences. This yields a problem for data mining in general because it is not clear which of these aspects is most suited to fulfill the given task. Therefore, it would be beneficial if a classification algorithm could employ all of the given representations of an object to make accurate class predictions. Another important aspect is that many classification algorithms rely on an object representation providing feature vectors. However, complex objects are often represented in a better way by treating them as sequences, trees or graphs. Last but not least, the number of classes in the given example applications can be exceptionally high. Gene Ontology [1], one of the most established class systems for proteins, currently has more then 14,000 classes and biometric databases will have to identify one special person among thousands of people. Though this problem is not directly connected to the complexity of the given data objects, it often co-occurs in the same application and should therefore be considered when selecting the classification method.

To cope with these challenges, we introduce a new classification technique based one k nearest neighbor (kNN) classification [2]. A kNN classifier decides the class of an object by analyzing its k nearest neighbors within the training objects. kNN classifiers are well-suited to solve the given problem because they do not have to spend additional effort for distinguishing additional classes. The new training objects are simply added to the training database and are only considered for classification if they are among the nearest neighbors of the object to be classified. Additionally, kNN classifiers can be applied to any type of object representation as long as a distance measure is available. Unfortunately, kNN classification has a major drawback as well. The efficiency of classification is rapidly decreasing with the number of training objects. Though the use of index structures such as the M-tree [3] or the IQ-Tree [4] might help to reduce query times in some cases, it does not provide a general solution. Another approach to limit the problem is the reduction of the training objects to some basic examples as proposed in [5]. However, these approaches are aimed at limited training data and are therefore very inefficient when applied to large training sets.

Thus, to apply kNN classification to the described classification scenario, we introduce a more efficient method to speed up kNN classification by employing

density-based clustering to reduce the necessary training instances. Afterwards, we introduce a new method for the classification of multi-represented (MR) objects. The idea of the method is to determine the k nearest neighbors in a database for each representation. Then, the class prediction is derived by considering the normalized distances within each result. To demonstrate the good performance, we apply our new method to four scenarios of protein classification. Each protein is represented by an amino acid sequence and a text annotation. Our results demonstrate that density-based clustering outperforms other methods of reducing the training set for kNN classification. Furthermore, the achieved results indicate that our new decision rule for multi-represented kNN classification yields better accuracy than other classification methods that are suitable for large class sets.

The rest of the paper is organized as follows. In section 2, we discuss related work on speeding up kNN classification and classification of multi-represented objects. Section 3 describes the use of density-based clustering to reduce the number of training instances without losing essential concepts. Additionally, our new approach to combine multi-represented classification is introduced. Section 4 provides an experimental evaluation based on protein data that consists of sequential and text representations. The last section sums up the introduced solutions and gives some directions for future work.

2 Related Work

k Nearest Neighbor Classifier. The k nearest neighbor (kNN) classification [2] mentioned above classifies a new data object o by finding its k nearest neighbors with respect to a suitable distance function. In its basic form, kNN classification predicts the class that provides the most training objects within the k-nearest neighbors. To the best of our knowledge, there exists no form of kNN classification that is directly applicable to multi-represented data objects. The common approach to apply kNN classification to this kind of data is to build a joint distance measure on the complete MR object. However, we argue that this method is not suitable to derive good results because it is not capable to weight the different representations on the basis of the given object.

Instance Reduction. In the last decades, the research community introduced several methods for instance reduction [5–9]. All approaches try to reduce the number of instances in the training set in a way that the classifier provides comparable or even better accuracy and demands less processing time. In [8] the authors discuss several reduction techniques and [10] illustrates an experimental evaluation of these algorithms on 31 data sets. This evaluation demonstrates that the RT3 algorithm [8] outperforms other techniques of instance reduction for many data sets. Another approach to instance reduction is called iterative case filtering (ICF)[5]. This novel and effective approach to data reduction employs two steps. The first step performs so-called "Wilson editing". It detects all instances that are classified incorrectly by the kNN classifier. These instances are afterwards removed. The second step calculates for each remain-

ing object the so-called *reachability* and *coverage* [5]. Every object o with $|reachable(o)| < |coverage(o)|$ is removed. The second step is iterated until no removable object exists. A broad experimental evaluation [11] on 30 databases compares ICF with the reduction technique RT3 [8]. Both algorithms achieve the highest degree of instance reduction while maintaining classification accuracy.

GDBSCAN. GDBSCAN [12] is a density-based clustering algorithm. Clusters are considered as dense areas that are separated by sparse areas. Based on two input parameters (ε and $MINPTS$), GDBSCAN defines dense regions by means of core objects. An object $o \in DB$ is called *core object*, if its ε-neighborhood contains at least $MINPTS$ objects. Usually clusters contain several core objects located inside a cluster and border objects located at the border of the cluster. In addition, the objects within a cluster must be "density-connected". GDBSCAN is able to detect clusters by one single pass over the data. The algorithm uses the fact, that a density-connected cluster can be detected by finding one of its core-objects o and computing all objects which are density-reachable from o. To determine the input parameters, a simple and effective method is described in [13]. This method can be generalized and used for GDBSCAN as well.

Classifier Fusion. The task of learning from objects, when more than a single classifier has beeen trained, has recently drawn some attention in the pattern recognition community [14–16]. In [15], the author describes the method of classifier fusion to combine the results from multiple classifiers for one and the same object. Furthermore, [15] surveys the four basic combination methods and introduces a combined learner to achieve combination rules offering better accuracy. In [17], a method for the hierarchical classification of MR objects was introduced. Though this method provides superior accuracy to the compared methods, it is not suitable for our described scenario because the efficiency of the method relies on the existence of a class hierarchy that can be exploited. Furthermore, the proposed classifier is based on Support Vector Machines that are not as generally applicable as kNN classification.

3 kNN Classification of Complex Objects

As mentioned in the introduction, classification of complex objects into large class sets yields the following challenges. First of all, the selected classification approach has to cope with the large number of classes without losing performance. Second, complex objects might be described by multiple representations. Furthermore, these representations might consist of varying object types, e.g. vectors, sequences and graphs. A good approach to handle these problems is kNN classification which does not need to spend additional efforts for distinguishing additional classes. Another benefit of kNN classifiers is that they do not rely on a special data type but can cope with any object type as long as there is a distance function. A drawback of kNN classification is that the classification time strongly depends on the number of training objects. Therefore, the number of training objects should be kept as low as possible to ensure efficient classification. In this paper, we discuss a method to reduce training instances

based on density-based clustering. Furthermore, we introduce a new method for the classification of multi-represented objects that is capable of achieving significantly better accuracy than the classification based on only one representation or related methods of classification.

In the following, we present a brief problem description. Afterwards, we introduce an approach to reduce the given training data with the help of density-based clustering. Finally, we use multiple object representations to derive accurate class predictions.

3.1 Problem Definition

In our given application scenario, we want to find a classifier $Cl : O \rightarrow C$ that maps each data object $o \in O$ to its correct class $c \in C$. The data space O is given by the cartesian product of m representations $R_1 \times \ldots \times R_m$. Each representation R_i consists of a feature space $F_i \cup \{-\}$. A feature space F_i may consist of varying data types. For comparing two objects $u, v \in F_i$, there exists a distance measure $dist_i : F_i \times F_i \rightarrow \mathbb{R}_0^+$. To apply our method, it is necessary that $dist_i$ is symmetric and reflexive. The symbol $\{-\}$ denotes that a particular object representation is missing. However, for a usable class prediction a tuple should provide at least one instance $r_i \in F_i$. To conclude, the task of multi-represented classification is to find a function $Cl_{mr} : (R_1 \times \ldots \times R_m) \rightarrow C$ that maps as many objects o to their correct class $c \in C$ as possible. For training, a set T of tuples (o, c) of objects $o = (r_1, \ldots, r_m)$ and their correct classes c are given to the classifier, the so-called training set. We denote in further sections the correct class of an object o by $c(o)$ and the class detected by multi-represented classification as $Cl_{mr}(o)$.

3.2 Density-based Instance Reduction

The performance of kNN classification depends on the number of objects in the training set. Though a lot of methods that reduce the training data for kNN classification have been proposed so far, most of these techniques perform poorly for large amounts of training data. In order to reduce the number of available training objects more efficiently, we suggest a novel approach – density-based instance reduction (DBIR).

The DBIR-algorithm works as follows. For each representation and each class, the training data is clustered by using the algorithm GDBSCAN. Let us note that the input parameters can be chosen as described in [13]. GDBSCAN provides a set of clusters $Clust = \{Clust_1, \ldots, Clust_j, \ldots, Clust_l\}$, where $j = 1, \ldots, l$ is the index of the cluster, and additionally a set of objects N that are noise, i.e. objects that cannot be associated with any clusters. An important characteristic of GDBSCAN for our problem is that the number of found clusters l is not predefined, but a result of the clustering algorithm. Thus, the number of important concepts is determined by the algorithm and not manually. Another important advantage of GDBSCAN is that it is capable to cluster any data type as long as there is a reflexive and symmetric distance measure to compare the objects. After clustering, DBIR iterates through the set $Clust$ and determines

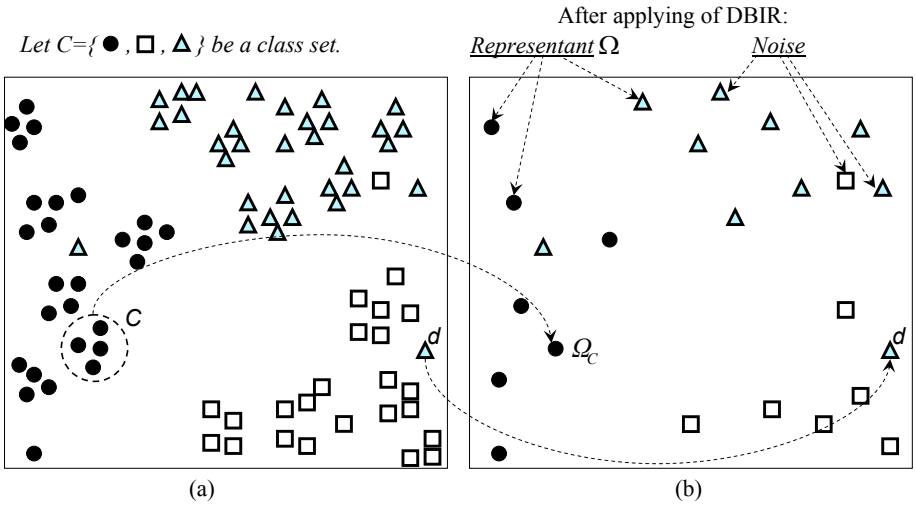

Fig. 1. (a) Objects before data reduction, (b) Objects after reduction by using of DBIR. The density-based cluster C can be reduced to a representant Ω_C. The noise object d is not removed. However, it can not change the decision of a kNN classifier with $k > 2$.

for each cluster $Clust_j$ a representant Ω_j. The representant Ω_j is the centroid of the cluster $Clust_j$ in the case of a representation given by a vector space and the medoid of the cluster $Clust_j$ otherwise. Afterwards, all objects belonging to the set $Clust_j \setminus \Omega_j$ are removed from the data set.

Like most other instance reduction methods, we assume that the training data for each class contains all important examples to specify a given class. To reduce the number of training objects without losing accuracy, we have to discard the training objects that are likely to represent a concept that is not typical for the given class. Furthermore, if a typical concept is described by several training objects, we reduce the representatives of this concept to a single one to save classification time. We argue that a density-based clustering of the training objects for a given class is sufficient to decide both cases. Objects that are not typical for a given class do not have any close neighbors and are usually separated from the rest of the training set. Thus, the noise objects in a density-based clustering are likely to correspond to these objects. Of course, it is possible that a noise object alone is an important concept. However, a single object is not likely to change the decision of a kNN classifier and the decision would most likely be wrong even without the deletion. Important concepts that are represented by several training objects are usually located very closely to each other in the feature space. Thus, these concepts are likely to correspond to a density-connected cluster in our density-based clustering. For each of these clusters it is sufficient that the training set contains a single object to represent it. Figure 1 displays both effects in a two dimensional example.

Our method has a runtime complexity of $O(\sum_{c_j \in C} |\{o \in O \mid c(o) = c_j\}|^2)$ for the case that it is not supported by index structures. ICF has a runtime complexity of $O(2 \times (\#Iteration) \times |DB|^2)$ where $\#Iteration$ is the number of iterations (in our experiments it was between 9 and 12) and $|DB|$ is the size of the database. Thus, our method is considerably faster than other state of the art feature reduction techniques.

As described above, we apply the DBIR-algorithm separately to the training objects in one representation and for one class. Afterwards we integrate all instances of a representation i into one training database DB_i. Let us note that it is possible to speed up k nearest neighbor queries in each of these training databases as long as there are suitable index structures for the given object type. For example, if the distance function is metric it might be beneficial to further increase the classification time by employing a metric tree like the M-Tree [3].

3.3 *k*NN-Classification of multi-represented objects

Based on the training databases for each representation, we apply the following method of kNN-based classification. To classify a new data object $o = (r_i, \ldots, r_m)$, the kNN sphere $sphere_i(o, k)$ in each representation with $r_i \neq$ " $-$ " is determined. Formally, the $sphere_i(o, k)$ can be described as follows:

$$sphere_i(o,k) = \{o_1, \ldots, o_k \mid o_1, \ldots, o_k \in DB_i \wedge \nexists o' \in DB_i \setminus \{o_1, \ldots, o_k\} \\ \wedge \nexists \xi, 1 \leqslant \xi \leqslant k : dist_i(o', r_i) \leqslant dist_i(o_\xi, r_i)\}$$

To combine these kNN spheres and achieve accurate classification, we first of all derive a confidence vector $cv_i(o)$ from each available $sphere_i(o, k)$. Let $c(o)$ denote the correct class of object o and let $d_i^{norm}(u, v)$ be a normalized distance function. Then the confidence vector for an object o with respect to its kNN sphere $sphere_i(o, k)$ for the representation i is defined as follows:

$$cv_i(o) = (cv_{i,1}(o), \ldots, cv_{i,|C|(o)}), \tag{1}$$

$$\forall j, 1 \leqslant j \leqslant |C| : cv_{i,j}(o) = \frac{\sum_{u \in sphere_i(o,k) \wedge c(u) = c_j} \frac{1}{d_i^{norm}(o,u)^2}}{\sum_{k=1}^{|C|} cv_{i,k}(o)} \tag{2}$$

To normalize our distance function for each representation, we apply the following modification:

$$d_i^{norm}(o, u) = \frac{dist_i(o, u)}{\max_{v \in sphere_i(o,k)} dist_i(o, v)} \tag{3}$$

where $dist_i$ is the distance function between two objects in the i-th representation. The normalization in formula 3 maps the distance values for each representation to the range $[0, 1]$ with respect to the radius of $sphere_i(o, k)$. Thus, the confidence vector of the i-th representation at the j-th position (cf. formula 2) is a normalized sum of the inverse quadratic distances.

	Set 1	**Set 2**	**Set 3**	**Set 4**
Name	Enzyme Activity	Metabolism	Transferase	Cell Growth
Number of Goal Classes	267	251	62	37
References to proteins	16815	19639	4086	4401

Table 1. Details of the test environments

After we have determined the confidence vectors $cv_i(o)$ for each representation i, we use a weighted linear combination for combining them. Let us note that the combination of confidence vectors to achieve multi-represented classification has been proposed in [15]. However, the used weights in the former approaches do not adjust to the individual classification object. We argue that in order to use each representation in a best possible way, a multi-represented decision rule must weight the influence of all available representations individually for each object.

To achieve this individual weighting, our classification rule is built as follows:

$$Cl_{mr}(o) = \max_{j=1,\ldots,|C|} \sum_{i=1}^{m} w_i \cdot cv_{i,j}(o) \tag{4}$$

where m is the number of representations and

$$w_i = \begin{cases} 0 & \text{, if } r_i = \text{" } - \text{"} \\ \frac{1+\sum_{j=1}^{|C|}(cv_{i,j}(o)\cdot\log_{|C|} cv_{i,j}(o))}{\sum_{k=1}^{m}(1+\sum_{j=1}^{|C|}(cv_{k,j}(o)\cdot\log_{|C|} cv_{k,j}(o)))} & \text{, otherwise} \end{cases} \tag{5}$$

The idea of our method is that a kNN sphere containing only a small number of classes and several objects of one special class is "purer" than a kNN sphere containing one or two objects for each of the classes. Thus, the "purer" a kNN-sphere for a representation is, the better is the quality of the class prediction that can be derived from this representation. To measure this effect, we employ the entropy with respect to all possible classes. The weight is now calculated by normalizing the entropy of its kNN sphere with respect to the entropy of the kNN spheres in all representations. As a result the weights of all representations add up to one. In conclusion, our decision rule for multi-represented objects measures the contribution of each available representation by the entropy in the local kNN spheres of all available representations.

4 Experimental Evaluation

4.1 Test Bed

In order to demonstrate the advantages of our approach, we carried out a versatile experimental evaluation. All algorithms are implemented in Java and were

Classification Accuracy (in %)								
	Set 1, Rep. 1	**Set 2, Rep. 1**	**Set 3, Rep. 1**	**Set 4, Rep. 1**	**Set 1, Rep. 2**	**Set 2, Rep. 2**	**Set 3, Rep. 2**	**Set 4, Rep. 2**
kNN	64.43	61.41	72.01	76.2	46.6	43.9	47.48	62.92
kNN DBIR	61.95	60.29	72.56	73.91	44.5	45.5	48.97	56.58
kNN ICF	46.44	35.56	47.92	40.72	37.85	33.21	31.37	34.58
Runtime of Instance Reduction (in sec.)								
DBIR	163.0	253.9	8.0	27.5	275.9	1069.6	36.6	119.9
ICF	12,809.1	15,616.7	590.0	632.0	93,416.8	112,248.2	4,258.0	3,772.0
Reduction Rate (in %)								
DBIR	26.1	27.4	33.1	32.0	28.1	22.9	33.8	35.0
ICF	57.0	64.3	71.8	77.7	37.8	46.5	64.0	65.5

Table 2. Experimental results Classification accuracy (in %) of kNN classifier on: unreduced data, data reduced by DBIR and ICF. Rune time (in sec.) and reduction rate (in %) reached by DBIR and ICF. (Using two representations Rep. 1 and Rep. 2.)

tested on a work station that is equipped with a 1.8 GHz Opteron processor and 8 GB main memory. We used the classification accuracy to measure the effectiveness of algorithms and 5-fold cross-validation to avoid overfitting.

The properties of each test bed are shown in table 1. The 4 test beds consist of 37 to 267 Gene Ontology[1] classes. The corresponding objects were taken from the SWISS-PROT [18] protein database and consist of a text annotation and an amino acid sequence of a protein. In order to obtain a flat class-system with sufficient training objects per class, the original environment was pruned.

We employed the approach described in [19] to extract features from the amino acid sequences. The basic idea is to use local (20 amino acids) and global (6 exchange groups) characteristics of a protein sequence. To construct a meaningful feature space, we formed all possible 2-grams for each kind of characteristic, which generated us the 436 dimensions of our sequence feature space. For text descriptions, we employed a TFIDF [20] vector for each description that was built of 100 extracted terms. We used the cosine distance function as distance measure for both representations.

4.2 Experimental Results

To demonstrate that DBIR is suitable for large data sets w.r.t. efficiency, we compared the run time needed for data reduction by using DBIR and ICF on single-represented data. As presented in table 2, the DBIR outperforms ICF in terms of efficiency, e.g. on the 1st representation of data set 1, DBIR needed only 163 sec. whereas ICF spends 12,809.1 sec. for the data reduction.

To show the effectiveness of DBIR, we compared the classification accuracy achieved by the kNN classifier on unreduced data, data reduced by DBIR and data reduced by ICF (cf. table 2). All these experiments were performed on

Classification accuracy (in %)				
	Set 1	**Set 2**	**Set 3**	**Set 4**
1st Representation, kNN DBIR	61.95	60.29	72.56	73.91
2nd Representation, kNN DBIR	44.5	45.5	48.97	56.58
1st and 2nd Representations, MR-kNN DBIR	**67.65**	**65.17**	**75.52**	**76.8**
1st Representation, NB	43.45	39.95	58.41	41.08
2nd Representation, NB	28.44	22.36	32.87	31.35
1st and 2nd Rep., NB with sum rule fusion	39.64	35.47	51.15	36.03
1st and 2nd Rep., kNN classifier fusion by sum rule	62.1	63.18	64.14	74.67
Average classification time per object (in msec.)				
1st Representation, kNN DBIR	196.1	198.87	38.22	39.86
2nd Representation, kNN DBIR	740.5	907.78	160.42	161.88
1st and 2nd Rep., MR-kNN DBIR	1,005.4	1,105.4	198.3	201.6
1st Representation, NB	45.06	43.54	15.4	9.04
2nd Representation, NB	155,91	150,75	48,34	29,62
1st and 2nd Rep., NB with sum rule fusion	206.37	198.3	61.54	36.73
1st and 2nd Rep., kNN classifier fusion by sum rule	1,251.3	1,456.2	295.6	316.8

Table 3. Classification accuracy (in %) and average classification time per object (in msec.) of our approach (MR-kNN DBIR) compared to: kNN on single representations reduced by DBIR; Naive Bayes (NB) on single representations and on multiple representations combined by sum rule [15]; kNN classifiers combined by sum rule.

single-represented data. The accuracy achieved by the kNN classifier on data reduced by using DBIR was for all of the data sets comparable to the unreduced data set. In contrast to these results, the classification accuracy achieved while using ICF was considerably lower. E.g. on the 1st representation of data set 1, the kNN classification on the data reduced by DBIR reaches 61.95% accuracy, whereas the kNN classification on the data reduced by ICF reaches only 46.44% accuracy. Though the reduction rate achieved by ICF is higher than that of DBIR, the clearly superior accuracy that is achieved by using DBIR indicates that ICF removed important information from the training data set.

In order to demonstrate the effectiveness of the proposed multi-represented kNN classifier (MR-kNN DBIR), we compared it to the kNN classifier on single representations, naive Bayes (NB) on unreduced single-represented data, NB classification combined by the sum rule and kNN classification combined by the sum rule. The sum rule described in [15] adds up the confidence vectors delivered by classifiers responsible for single representations. Table 3 illustrates the experimental results of this comparison. Our method showed the highest classification accuracy on all data sets and achieved a significant increase of accuracy in comparison to single-represented classification, e.g. on the first set the kNN classifier delivered 61.95% accuracy on the first and 44.5% accuracy on the second representation whereas our approach achieved a significantly higher accuracy of 67.65%. NB showed in our experiments low accuracy both on single representations and when combining single NB classifiers employing the sum

rule. Our method outperforms also the combination of kNN classifiers using the sum rule in all test environments (cf. table 3).

5 Conclusions

In this paper, we proposed a novel approach for classifying multi-represented data objects into flat class-systems with many classes. Our method aims at a common application scenario that can be described by the following characteristics: First, objects in modern applications often provide multiple representations which are derived from multiple views of the same data object. Second, complex objects might be described by representations that are not necessarily in feature vector form. Thus, a classifier should cope with a variety of data types. Last but not least, novel applications often provide large class sets that distinguish huge amounts of classes. Therefore, classifiers should be able to distinguish additional classes with a minimum of additional training and classification effort. To cope with these requirements, our new method for classification of multi-represented objects employs kNN classification because this approach is naturally able to handle the last two requirements. An important contribution of our method is a new way of instance reduction to limit the number of employed training objects and thus to speed up classification time without significantly loosing accuracy. To integrate the information of several representations, we present a new decision rule that employs a weighted combination of confidence values to derive a class prediction. The idea of the used weighting is to measure the entropy of each kNN sphere and thus representations are weighed in a different way for different data objects. In our experimental evaluation, we compared our new instance reduction technique called DBIR to one of the best performing instance reduction techniques so far, ICF. Our results indicate that DBIR is capable to reduce the training database faster and provides better accuracy than ICF. To demonstrate the effectiveness of our multi-represented kNN classifier, we compared the classification accuracy using related methods and employing classification based on single representations. The results demonstrate that our new method is capable of outperforming the compared approaches and significantly increases the accuracy by integrating all representations.

For future work, we plan to examine the use of various index structures to speed up classification. Furthermore, we plan to apply our method on the second application area mentioned in the introduction, biometric identification. This area yields several individual challenges like the combination of different classification methods. For example, facial features can be checked by kNN classification. However, in order to recognize a person by its speech pattern other ways like hidden Markov models are reported to provide better accuracy. Thus, a flexible model should support different classification algorithms. Another interesting direction is to further speed up classification by employing only some of the representations. For example, it might be unnecessary to query the sequence database if the text database provides sufficient confidence.

References

1. Consortium, T.G.O.: "Gene Ontology: Tool for the Unification of Biology". Nature Genetics **25** (2000) 25–29
2. Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE Transactions on information Theory **IT-13** (1967) 21–27
3. Ciaccia, P., Patella, M., Zezula, P.: "M-tree: An Efficient Access Method for Similarity Search in Metric Spaces". In: Proc. of the 23rd Int. Conf. on Very Large Data Bases, Morgan Kaufmann, San Francisco, CA, USA (1997) 426 – 435
4. Berchtold, S., Böhm, C., Jagadish, H., Kriegel, H.P., Sander, J.: "Independent Quantization: An Index Compression Technique for High-Dimensional Spaces". In: Int. Conf. on Data Engineering, ICDE 2000. (2000)
5. Brighton, H., Mellish, C.: On the consistency of information filters for lazy learning algorithms. In: PKDD. (1999) 283–288
6. Gates, G.: The reduced nearest neighbour rule. IEEE Transactions on Information Theory **18** (1972) 431–433
7. Ritter, G., Woodruff, H., Lowry, S.R. Isenhour, T.: An algorithm for the selective nearest neighbor decision rule. IEEE Transactions on Information Theory **21** (1975) 665–669
8. Wilson, H., Martinez, T.: Instance pruning techniques. In: Proc. 14th Int. Conf. on Machine Learning, Morgan Kaufmann Publishers (1997) 403–411
9. Aha, D.: Tolerating noisy, irrelevant and novel attributes in in instance-based learning algorithms. Int. Jurnal of Man-Machine Studies **36** (1992) 267–287
10. Wilson, H., Martinez, T.: Machine Learning, 38-3. Reduction Techniques for Instance-Based Learning Algorithms. Kluwer Academic Publishers, Boston. (2000)
11. Brighton, H., Mellish, C.: Data Mining and Knowledge Discavery, 6. Advances in Instance Selection for Instance-Based Learning Algorithms. Kluwer Academic Publishers. (2002)
12. Sander, J., Ester, M., Kriegel, H.P., Xu, X.: "Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and its Applications". In: Data Mining and Knowledge Discovery, Kluwer Academic Publishers (1998) 169–194
13. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise". In: Proc. KDD'96, Portland, OR, AAAI Press (1996) 291–316
14. Kittler, J., Hatef, M., Duin, R., Matas, J.: "On Combining Classifiers". IEEE Transactions on Pattern Analysis and Machine Intelligence **20** (1998) 226–239
15. Duin, R.: "The Combining Classifier: To Train Or Not To Train?". In: Proc. 16th Int. Conf. on Pattern Recognition, Quebec City, Canada). (2002) 765–770
16. Kuncheva, L., Bezdek, J., Duin, R.: "Decision Templates for Multiple Classifier Fusion: an Experimental Comparison". Pattern Recognition **34** (2001) 299–314
17. Kriegel, H.P., Kröger, P., Pryakhin, A., Schubert, M.: Using support vector machines for classifying large sets of multi-represented objects. In: Proc. SIAM Int. Conf. on Data Mining, Lake Buena Vista, Florida, USA. (2004) 102–114
18. Boeckmann, B., Bairoch, A., Apweiler, R., Blatter, M.C., Estreicher, A., Gasteiger, E., Martin, M., Michoud, K., O'Donovan, C., Phan, I., Pilbout, S., Schneider, M.: "The SWISS-PROT Protein Knowledgebase and its Supplement TrEMBL in 2003". Nucleic Acid Research **31** (2003) 365–370
19. Deshpande, M., Karypis, G.: "Evaluation of Techniques for Classifying Biological Sequences". In: Proc. of PAKDD'02. (2002) 417–431
20. Salton, G.: Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer. Addison-Wesley (1989)