BIOINFORMATICS ORIGINAL PAPER Vol. 22 no. 8 2006, pages 981–988
doi:10.1093/bioinformatics/btl027

Data and text mining

Enhancing instance-based classification with local density: a new algorithm for classifying unbalanced biomedical data

Claudia Plant[1], Christian Böhm[2], Bernhard Tilg[1] and Christian Baumgartner[1,*]

[1]Research Group for Clinical Bioinformatics, Institute for Biomedical Engineering, University for Health Sciences, Medical Informatics and Technology, Hall in Tyrol, Austria and [2]Institute for Computer Science, University of Munich, Germany

Received on September 29, 2005; revised on January 3, 2006; accepted on January 25, 2006
Advance Access publication January 27, 2006
Associate Editor: Alfonso Valencia

ABSTRACT

Motivation: Classification is an important data mining task in biomedicine. In particular, classification on biomedical data often claims the separation of pathological and healthy samples with highest discriminatory performance for diagnostic issues. Even more important than the overall accuracy is the balance of a classifier, particularly if datasets of unbalanced class size are examined.

Results: We present a novel instance-based classification technique which takes both information of different local density of data objects and local cluster structures into account. Our method, which adopts the basic ideas of density-based outlier detection, determines the local point density in the neighborhood of an object to be classified and of all clusters in the corresponding region. A data object is assigned to that class where it fits best into the local cluster structure. The experimental evaluation on biomedical data demonstrates that our approach outperforms most popular classification methods.

Availability: The algorithm LCF is available for testing under http://biomed.umit.at/upload/lcfx.zip

Contact: christian.baumgartner@umit.at

1 INTRODUCTION

Efficient and effective classification is a core problem in biomedical data mining. Some of the existing classification methods produce explicit rules, e.g. decision trees (DT), linear discriminant analysis, logistic regression analysis (LRA) and support vector machines (SVMs), etc. Other classification methods such as the *k*-nearest neighbor (*k*-NN) classifier are called instance-based because no explicit model is produced (Mitchell, 1997; Baumgartner *et al*., 2004). Many biological datasets consist of a complex cluster structure. Even class-pure subsets of the data objects may be composed of different clusters. In this case, the classes are not easily separable by planes, polynomial functions or combinations thereof and rule-based classifiers tend to break down in terms of accuracy. Often, the simple instance-based *k*-NN classifier performs better, but only if the point density is relatively uniform in all classes. Unbalanced datasets exhibiting a high variation in the number of data items per class tend to have regions of different density. Data objects situated in boundary regions between high and low density are always classified into the class of the region of higher density. For unsupervised data mining tasks, density-based clustering methods have become very successful owing to their robustness and efficiency (Ester *et al*., 1996; Ankerst *et al*., 1999). Recently, density-based methods for outlier detection have appeared, such as local outlier factor (LOF) or LOCI (Breuning *et al*., 2000; Papadimitriou *et al*., 2003). In contrast to distance-based methods local and global outliers can be discovered. In the density-based notion outliers are determined by taking the density of the surrounding region into account.

The general idea of our paper is to consider the cluster structure of the dataset and to use the information of different densities for classification. A data object is assigned to that class where it fits best into the local cluster structure. This idea can be formalized by defining a local classification factor (LCF) which is similar to the density-based outlier factors, but with an opposite intention. It assigns a data object to that class from which it is least considered as a local outlier. By adopting the concepts of density-based methods to classification, we obtain a high accuracy especially on unbalanced datasets.

2 SYSTEMS AND METHODS

2.1 Classification methods

Model-generating classification methods first learn a model from the training set which is then used to assign class labels to the unlabeled objects. LRA for example constructs a linear separating hyperplane between classes (Hosmer and Lemeshow, 2000). DT are usually rooted, binary trees with simple classifiers at each internal node recursively splitting the feature space (Quinlan, 1986, 1993). The Naive Bayes (NB) classifier is an approximation to an ideal Bayesian classifier which would classify an object based on the probability of each class given the object's feature variables. NB assumes Gaussian distributed data (Langley *et al*., 1992; Gelman *et al*., 2004). Inspired by the biological nervous system, artificial neural networks (ANN) can deal with arbitrary data distributions. Consisting of several layers of neurons, an input layer takes the input and distributes it to the hidden layers—which do all the necessary computations—and outputs the result to the output layer (Bishop, 1995; Mitchell, 1997). More efficient and less sensitive to the number of training examples than ANN, the SVM is one of the most successful learning algorithms. Using kernel functions data objects are transformed to a higher-dimensional space where a separating maximum margin hyperplane can efficiently be determined by solving a

*To whom correspondence should be addressed.

constrained dynamic optimization problem (Cortes and Vapnic, 1995; Vapnic, 1998; Platt *et al*., 2000; Cristianini and Shawe-Taylor, 2000).

Requiring no preprocessing, instance-based classifiers can very efficiently be applied to all types of data. The *k*-NN classifier simply assigns to an object the most frequent class label among its *k* nearest neighbors. On complex, high-dimensional and unbalanced datasets, the simple instance-based *k*-NN classifier sometimes outperforms other more sophisticated methods in terms of accuracy, as shown, for example, in Horton and Nakai (1997) for predicting protein cellular localization sites. Several extensions to *k*-NN have recently been proposed, such as using locally weighted Euclidian distance to determine neighborhoods that better reflect the local class distribution (Hastie and Tibshirani, 1996; Paredes and Vidal, 2000). Xie *et al*. (2002) proposed an instance-based Bayesian classifier using different distance neighborhoods for classification. In this paper, we show that ideas from density-based outlier detection can enhance instance-based classification.

2.2 Density-based outlier detection

Methods based on a density-based clustering notion have been successfully applied to outlier detection since they can cope with datasets exhibiting both sparse and dense regions. The LOF (Breuning *et al*., 2000) determines to which extent an object is an outlier with respect to (w.r.t.) its neighborhood. The neighborhood is here defined by the *k*-NN of an object. The density-based outlier factor LOCI (Papadimitriou *et al*., 2003) specifies the local neighborhood using range queries.

To the best of our knowledge, the classification problem has not been addressed before from the viewpoint of density-based clustering or outlier detection. We found our approach on the density-based clustering notion by defining a LCF assigning an object to the class of that cluster where the point fits best into according to the data density. The extensive experimental evaluation shows that the aspect of local density can significantly improve instance-based classification.

3 ALGORITHM

3.1 Using information of local density in data

For a data object q we compute a LCF w.r.t. each class $c_i \in C$ separately. We assign the object q to the class w.r.t. which it has the lowest LCF. In particular, the LCF consists of two parts:

- Direct Density (DD)
- Class Local Outlier Factor (CLOF).

The LCF is a weighted sum of these two aspects. Roughly speaking we assign an object q to class c_i if there is a high density of objects of class c_i in the region surrounding q. In addition, we claim that q is not an outlier w.r.t. the objects of class c_i in this region. In the following sections we explain these two parts in more detail. We introduce the concept of DD and define a simple and accurate outlier factor which is especially useful for classification. For illustration we use a two-dimensional synthetic dataset visualized in Figure 1a.

3.2 Direct density

Taking a global look at our demonstration dataset, the first impression probably is that class 2 is of much higher density than class 1. But since there may be regions of extremely different density among one class, we cannot globally specify the density of a class. However, we can locally examine the density of each class in the region of the object to be classified. For each class c_i the region surrounding the object q can be described by the set of the k-NN of q of class c_i.

DEFINITION 1. *Class k-nearest neighbors of an object q. For any positive integer k, the set of class k-NN of an object q w.r.t. class* $c_i \in C$, *denoted as* $NN_k^{c_i}$, *contains the objects of class* c_i *for which the following condition holds*:

If $|c_i| < k$: $NN_k^{c_i}(q) = \{p \in DB \mid p \in c_i\}$ *otherwise* $NN_k^{c_i}(q)$ *is a subset of k elements in database DB for which*

$$\forall (p) \in NN_k^{c_i}(q), \forall o \in c_i \backslash NN_k^{c_i}(q) : dist(p,q) < dist(o,q).$$

If a class contains less than k elements, the set $NN_k^{c_i}(q)$ *contains all objects of this class. If there are more objects,* $NN_k^{c_i}(q)$ *contains the class internal nearest neighbors. To capture the density of class* $c_i \in C$ *in the region surrounding the object q, we compute the mean value of the distances to the k-nearest neighbors of q belonging to class* c_i.

DEFINITION 2. *DD of class* c_i *w.r.t. q*

$$DD_q(c_i) = \frac{\sum_{p \in NN_k^{c_i}(q)} dist(p,q)}{|NN_k^{c_i}(q)|}.$$

We can use DD alone for classification by assigning an object q to that class where $DD_q(c_i)$ *is minimal. The concept of DD has several advantages to k-NN: Objects of rare classes get the chance to be correctly classified. We have no majority voting. Moreover, for the decision to which class an object should be assigned to we get a continuous value by computing the DD measure. So it is very unlikely to have a standoff situation. The result on our demonstration dataset using DD only is depicted in Figure 1b.* (*As described in section Experiments in more detail, we used* $k = 5$ *and 10-fold cross validation*). *Many objects of the sparser class 1 are wrongly classified. Intuitively they fit better in the cluster structure of their own class, so it should be possible to classify them correctly.*

3.3 Class local outlier factor

In addition to the DD, we now examine to which extent an object q is an outlier considering the local cluster structure of each class c_i separately. We define a density-based CLOF, similar to LOF (Breuning *et al*., 2000), but more suitable for classification. The idea that being an outlier is not a binary property is very useful for classification. Nevertheless, we cannot directly apply the LOF because it is based on the reachability distances of the data objects to reduce statistical fluctuations of the distances among objects significantly close to each other. Owing to this, the LOF of objects in the k-distance neighborhood of an object q is always similar to the LOF of q. This may be useful to discover meaningful outliers. However, for classification of an object q placed at the border between one or more classes we want to see even minor differences in the degree to which q is an outlier w.r.t. these classes. Instead of the reachability distance we use the distances to the k-NN, again computed class-wise separated. In addition to the DD as defined in Section 3.2, we need for the CLOF a measure for the indirect density of the class c_i, i.e. for the density of the region surrounding the object q excluding q itself.

DEFINITION 3. *Indirect density of class* c_i *w.r.t. q*

$$ID_q(c_i) = \frac{\sum_{p \in NN_k^{c_i}(q)} DD_p(c_i)}{|NN_k^{c_i}(q)|}$$

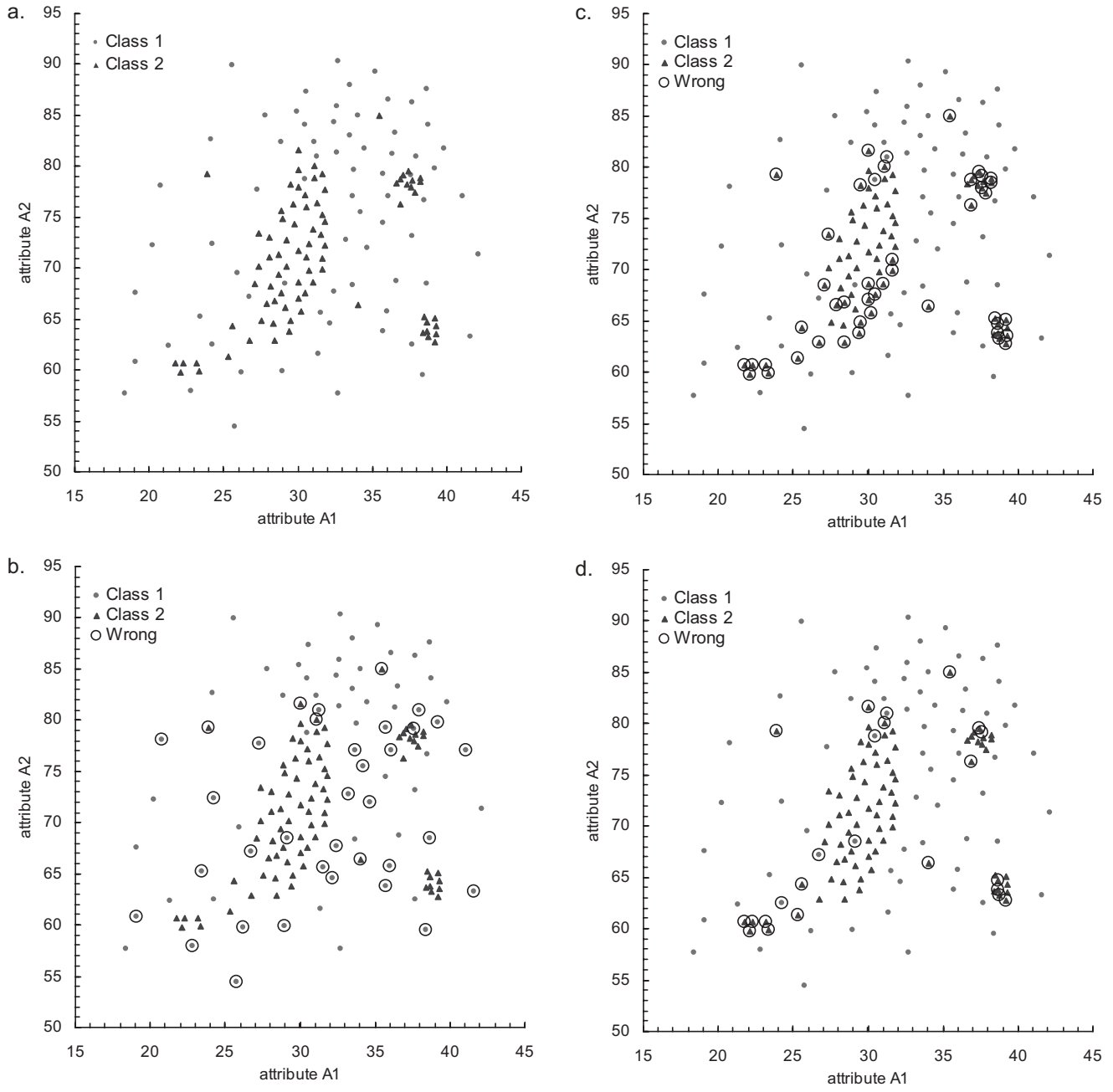

Fig. 1. (**a**) Two-dimensional demonstration dataset. (**b**) Result with direct density only. (**c**) Result with class local outlier only. (**d**) Result with LCF for $l = 6$ and $k = 5$.

Similar to the DD, the indirect density measure can be 0, *if there are at least k duplicates of class c_i in DB. For simplicity, we here assume that there are no duplicates. To deal with duplicates, we can base Definition* 1 *on the k distinct class nearest neighbors of the object in class c_i, with the additional assumption that there are at least k such objects. For the CLOF of an object q w.r.t. class c_i we consider the ratio of the direct and the indirect density of a class c_i w.r.t. q.*

DEFINITION 4. *Class local outlier factor of an object q*

$$CLOF_{c_i}(q) = \frac{DD_q(c_i)}{ID_q(c_i)}$$

The CLOF describes the degree to which an object q is an outlier to the local cluster structure w.r.t. class c_i. It is easy to see that for an object q located inside a cluster of objects of class c_i the CLOF is $\sim$1. *If q is an outlier w.r.t. class c_i it gets a significantly higher*

CLOF w.r.t. that class. The set of the indirect class nearest neighbors of q, denoted by ind $NN_k^{c_i}(q)$, contains all objects used to compute $ID_q(c_i)$.

DEFINITION 5. *Indirect class nearest neighbors of an object q*

$$ind\,NN_k^{c_i}(q) = \{o \in DB \mid \exists p \in NN_k^{c_i}(q) : o \in NN_k^{c_i}(p)\}$$

The set of objects used to compute the CLOF of an object q, i.e. $NN_k^{c_i}(q) \cup ind\ NN_k^{c_i}(q)$ can be regarded as the extended class neighborhood of q. The more homogenously the data distribution in this extended neighborhood is, the more CLOF(q) converges towards 1 and the tighter are the bounds for CLOF(q). To classify an object q using the CLOF, we compute $CLOFc_i(q)$ for each class $c_i \in C$ and assign q with the class c_i w.r.t. which its CLOF is minimal. The result on our demonstration dataset is depicted in Figure 1c.

Especially, at the margins of the clusters of the denser class 2 there are many wrongly classified objects. This attributes to the fact that the CLOF of the objects in these regions is similar w.r.t. both classes. Using DD, these objects are classified correctly.

3.4 Local classification factor

The main idea for the LCF is to combine the information of DD with the CLOF to overcome the drawbacks of both methods when used alone. It is not sufficient to require a high density of objects of class c_i in the region of the query point q to assign q to class c_i. The rule assigning q to the class w.r.t. which it has a smaller outlier factor leads to different mistakes. This owes to the fact that the CLOF completely ignores the local probability of the classes. Especially, if the CLOF of an object o is similar w.r.t. all classes we should assign the object to the most frequent class in its direct neighborhood.

DEFINITION 6. *LCF of an object q*

$$LCF_{c_i}(q) := DD_q(c_i) + l \cdot CLOF_{c_i}(q)$$

The LCF of an object q w.r.t. class c_i is the sum of its DD and its l-times weighted CLOF w.r.t. this class. We use a weighting factor l to determine to which extent the CLOF and the DD are relevant for classification. To classify an object q, we compute the LCF w.r.t. each class c_i for q and assign q to the class w.r.t. which its LCF is minimal.

In Figure 1d the final result on the demonstration dataset is depicted. Owing to the combination of both aspects, most classification errors disappear. In the following section we explain why we combine the two aspects in this way and give hints on a proper parameter choice.

3.5 Parameter choice for *k* and *l*

The parameter k determines the size of the region considered for computing the LCF. If k is chosen too small the local density cannot be appropriately characterized. k corresponds to the minimum cluster size, i.e. to the minimum number of objects of a class that should be regarded as a cluster. For our experiments, we used the training datasets to determine an appropriate value for k. In general, we defined k according to the recommendations for the k-NN classifier (range: $k = 3, \ldots, 15$). Larger values of k consider more neighbors, and therefore smooth over local characteristics, smaller values lead to limited neighborhoods.

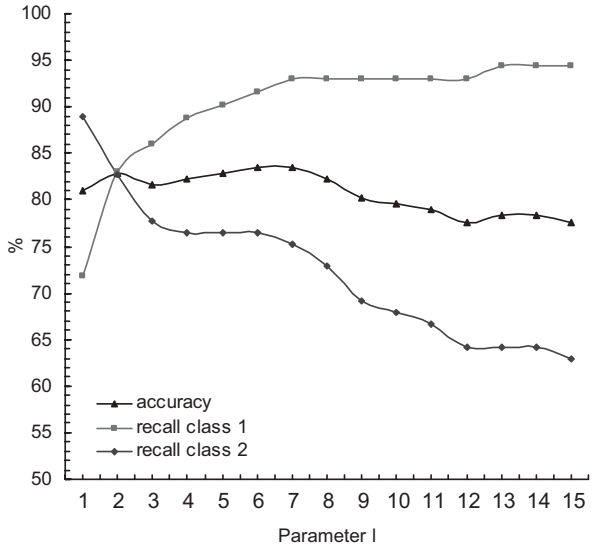

Fig. 2. Influence of parameter l on classification accuracy of synthetic data (data see Fig. 1).

The parameter l determines to which degree the outlier factor of an object q w.r.t. the classes $c_i \in C$ is relevant for its classification. A higher value for l leads to more correctly classified objects in the sparser classes, at the expense of incorrectly classified objects in the denser classes. Margin objects of the denser class often have a higher CLOF w.r.t. their own class than w.r.t. the sparser class. These objects are typically misclassified if the CLOF gets too much weight. Depending on the concrete application domain, l can be determined either to maximize the overall accuracy or to optimize recall and precision of a certain class. Particularly in biomedical data, high precision and recall on sparse classes are essential, since they often represent abnormal observations. Figure 2 shows accuracy and recall on the synthetic dataset for $k = 5$ and $l = 1, \ldots, 15$, similar characteristics can be observed considering precision. However, it is difficult to provide a general recommendation for parameter l because, as aforementioned, it depends on the given local data densities w.r.t. to the classes $c_i \in C$. On examined biomedical data, higher-dimensional datasets tend to larger l-values (metabolic data, $l = 35, \ldots, 55$), whereas lower-dimensional datasets show l-values close to 1 (e.g. synthetic $l = 2$, yeast $l = 0.1$, *Escherichia coli* $l = 0.1$) to be balanced in terms of recall or precision.

4 EXPERIMENTS

4.1 Biomedical data

LCF was tested and evaluated on one synthetic (cf. Figure 1a) and six real biomedical datasets as summarized in Table 1. Metabolic data were provided by a project partner (see Acknowledgement). Five datasets (yeast, *E.coli*, liver, iris and diabetes) come from the UCI Machine Learning Repository (Blake and Merz, 1998, http://www.ics.uci.edu/~mlearn/MLSummary.html). The table shows the dimensionality of data, the number of classes and objects and the number of objects per class. Detailed biological information and experimental results are described and discussed for each dataset separately throughout this section.

Table 1. Synthetic and biomedical datasets

Name	Classes	Dimensionality	Objects	Objects/class
Synthetic	2	2	152	71:81
Metabolic	2	45	57	38:19
Yeast*	10	8	1448	463:429:244:163: 51:44:35:30:20:5
*E.coli**	8	7	336	143:2:35:77:5:20:20:5
Liver*	2	5	345	145:200
Iris plant*	3	4	150	50:50:50
Diabetes*	2	8	768	500:268

Biological datasets marked by * come from the UCI Machine Learning Repository.

Table 2. Classification results on synthetic data

Classifier	Class	Corr.	Incorr.	Recall	Precision	Accuracy
LRA	1	37	34	52.1	56.9	59.2
	2	53	28	65.4	60.9	
SVM (poly)	1	32	39	45.1	65.3	63.2
	2	64	17	79.0	62.1	
SVM (radial)	1	36	35	50.7	60.0	61.2
	2	57	24	70.4	60.2	
5-NN	1	37	34	52.2	66.1	75.0
	2	77	4	95.1	80.2	
5-NN (weighted)	1	39	32	54.9	90.7	76.3
	2	77	4	95.1	70.6	
DT	1	32	39	45.1	76.2	67.8
	2	71	10	87.7	64.5	
NB	1	40	31	56.3	64.5	65.1
	2	59	22	72.8	65.6	
ANN	1	37	34	52.1	71.2	67.8
	2	66	15	81.5	66.0	
LCF ($l = 2$)	1	59	12	83.1	80.8	82.9
	2	67	14	82.7	84.8	
LCF ($l = 6$)	1	65	6	91.5	77.4	83.6
	2	62	19	76.5	91.2	83.6

Corr. = correctly classified, Incorr. = incorrectly classified instances.

4.2 Benchmark classifiers, validation and parameter settings

We compared LCF with six popular classification methods obtained from the publicly available WEKA data mining software (www.cs.waikato.ac.nz/ml/weka). For validation we used 10-fold cross validation. All classifiers were parameterized to optimize accuracy. For SVM we used both polynomial (of degree 2) and radial kernels, the cost factor c was appropriately chosen using the training dataset. We used the C4.5 decision tree algorithm with reduced error pruning. For ANN, we designed a single layer of hidden units with (number of attributes + number of classes)/2 hidden units, 500 epochs to train through and a learning rate of 0.3. For LRA and NB no advanced settings can be performed. We applied both weighted (1/distance) and unweighted k-NN with an Euclidian distance function and an appropriate value for k determined of the training datasets. For LCF we also used Euclidian distance and determined k and l of the training datasets.

4.3 Synthetic data

For demonstration issues, a two-dimensional synthetic dataset with classes of various local densities was taken into account (Fig. 1a). Here, class 2 is split up into three partitions that are separated from each other by objects belonging to the less dense class 1. This data structure was generated by using a data generator developed in-house. Table 2 summarizes classification accuracy, precision, recall in percent (%) and the number of correctly and incorrectly classified instances. For LCF the parameter k was set to 5, l was set to 2 and 6 respectively (cf. Fig. 2). LCF outperforms the other methods in terms of accuracy and balance of correctly classified instances between both classes for $l = 2$ (82.9%). LRA and SVM (polynomial, radial kernels) drop off in accuracy (59.2–63.2%) not being able to handle such complex data structures. DT, NB and ANN yielded higher accuracies (65.1–67.8%), but also lacked on the balance of correctly assigned objects within the two classes. k-NN, however, was able to further increase accuracy, but also classifies instances of the sparser class 1 predominantly to those of the denser class 2. Weighting only slightly attenuates this tendency.

4.4 Metabolic data

Classification in metabolomics has great potential for the development of automated diagnostics. After reviewing a certain population of healthy and diseased patients, abnormal metabolic profiles that are significantly different from a normal profile can be identified from data and thus can become diagnostic of a given disease (Baumgartner *et al*., 2004; Baumgartner and Baumgartner, 2006). The provided metabolic data, which was generated by modern tandem mass spectrometry (MS/MS) technology, contains concentration values of 45 metabolites [12 amino acids and 33 sugars (saccharides)] grouped into patients suffering from a multigenic metabolic disorder and healthy controls. Further information on data is strictly confidential. However, an anonymized test set is publicly available under http://biomed.umit.at/upload/lcfx.zip (2005).

Table 3 summarizes our experiments by setting parameter k again to 5 and parameter l to 35 for LCF. Owing to the small size of this dataset (57 instances) it is favorable to use a small k. It can be expected that metabolic data exhibits regions of various densities caused by a higher variation of metabolite concentration levels at the state of disease versus normal (Baumgartner and Baumgartner, 2006). The borders between healthy and pathological instances are blurred in this high-dimensional dataset containing overlapping clusters of both classes. Best accuracy was obtained for value of $l = 35$. Of all investigated classifiers LCF showed highest classification accuracy of 73.7% and a superior recall value of 68.4% for class 2, i.e. the abnormal metabolic profiles of diseased people. LCF results are the highest balanced in terms of recall and precision, and are comparable with LRA yielding correctly classified cases >50% in both classes. However, LRA lacks on accuracy of only 56.1%. SVM and ANN constitute similar accuracy values like LRA, but assign up to 80% of pathological cases to healthy subjects (false negative cases). The k-NN classifier demonstrates the best accuracy values within all benchmark classifiers, but breaks down in recall dramatically. The use of weighted k-NN does not help here. For diagnostic issues it is of highest importance to classify instances of smaller and sparser classes correctly, particularly if this class is represented by pathological cases. Thus, balance of correctly

Table 3. Classification results on metabolic data

Classifier	Class	Corr.	Incorr.	Recall	Precision	Accuracy
LRA	1	22	16	57.9	71.0	56.1
	2	10	9	52.6	38.5	
SVM (poly)	1	28	10	73.7	62.2	57.9
	2	5	14	26.3	31.3	
SVM (radial)	1	27	11	71.1	65.9	56.1
	2	5	14	26.3	31.3	
5-NN	1	37	1	97.4	68.5	68.4
	2	2	17	10.5	50	
5-NN (weighted)	1	36	2	94.7	67.9	66.6
	2	2	17	10.5	50	
DT	1	35	3	92.1	67.3	64.9
	2	2	17	10.5	40	
NB	1	31	7	81.6	77.5	68.4
	2	8	11	42.1	47.1	68.4
ANN	1	26	12	63.2	64.8	57.9
	2	7	12	31.6	36.8	57.9
LCF ($l = 35$)	1	29	9	76.3	82.9	73.7
	2	13	6	68.4	59.1	

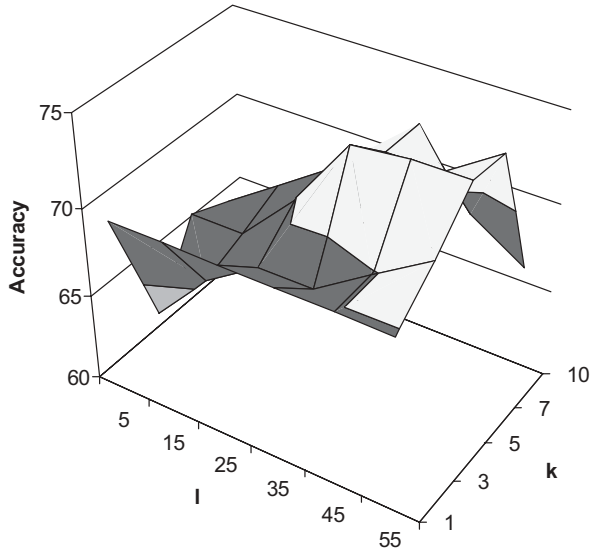

Fig. 3. Parameterization of LCF on unbalanced metabolic data. Classification accuracy depending on different k- and l-values is displayed.

classified objects between classes and high accuracy is essential for classifying diseased versus normal metabolite profiles so that LCF is an interesting tool to be used for diagnostics.

Figure 3 demonstrates classification accuracy of LCF as a 3D plot by setting parameter $k = 1, 3, 5, 7, 10$ and $l = 5, 15, 25, 35, 45, 55$. Best accuracy was achieved for $k = 5$ and l-values between 35 and 55.

4.5 Yeast data

The yeast dataset contains 1484 protein sequences labeled according to 10 classes (Horton and Nakai, 1996, 1997). Table 4 depicts classification results w.r.t. the three largest classes (1. cytoplasm, 2. nucleus and 3. mitochondria). The classes membrane protein (no N-terminal signal, uncleaved and cleaved signal, classes 4–6), extracellular, vacuole, peroxisome and endoplasmic reticulum (classes 7–10) consist of 5–163 instances and are not shown in detail. Parameter settings for LCF were $k = 12$ and $l = 0.1$. Comparing all classifiers, most of the errors are due to confusing cytoplasmic proteins with nuclear proteins and vice versa. This reflects a fundamental difficulty in identifying nuclear proteins. One reason is the fact that unlike other localization signals the nuclear localization signal does not appear to be limited to one portion of a protein's primary sequence. In some cases a protein without a nuclear localization signal may be transported to the nucleus as part of a protein complex if another subunit of the complex contains a nuclear localization signal (Zhao and Padmanabhan, 1988; Garcia-Bustos *et al*., 1991). In spite of this, LCF demonstrates the best balanced result for the first three classes w.r.t. recall (62.2, 59.7 and 60%) and precision (56.4, 57.4 and 63.8%), and an overall accuracy of 60.3%. LCF seems to be the best choice to identify nuclear proteins, however it is accompanied by a slight decrease of recall in class 1. In Table 5 the confusion matrix of LCF is shown in more detail. For the other classes not considered in Table 4 classification accuracy corresponds well to the results reported in Horton and Nakai (1997).

Table 4. Classification results on yeast data

Classifier	Class	Corr.	Incorr.	Recall	Precision	Accuracy
LRA	1	324	139	70.0	51.3	58.6
	2	198	231	46.2	61.7	
	3	139	105	57	62.1	
SVM (poly)	1	320	141	69.1	51.7	59.3
	2	217	212	50.6	60.4	
	3	128	116	52.5	66.0	
SVM (radial)	1	362	101	37.1	49.5	58.9
	2	162	267	37.8	64.8	
	3	139	105	57.0	67.1	
21-NN	1	327	136	70.6	52.7	59.2
	2	210	219	49.0	59.0	
	3	139	105	57.0	65.6	
21-NN (weighted)	1	331	132	71.5	55.8	61.9
	2	235	194	54.8	62.0	
	3	141	103	57.8	66.5	
DT	1	294	169	63.5	52.1	57.8
	2	223	206	52.0	57.8	
	3	116	128	47.5	64.1	
NB	1	324	139	70.0	51.5	57.6
	2	171	258	39.9	63.3	
	3	148	96	60.7	62.2	
ANN	1	301	162	65.0	54.1	59.4
	2	230	199	53.6	58.4	
	3	135	109	55.3	65.2	
LCF ($l = 0.1$)	1	288	175	62.2	56.4	60.3
	2	256	173	59.7	57.4	
	3	139	105	60.0	63.8	

With the exception of the ANN, DT and the weighted 21-NN classifier all other paradigms constitute a recall rate below 50% for nuclear proteins classification. For the k-NN classifier we used an optimized k value for this special dataset (Horton and Nakai, 1997).

Table 5. Confusion matrix for yeast data with LCF

Class	1	2	3	4	5	6	7	8	9	10
1	288	132	33	6	1	0	2	0	1	0
2	131	256	27	11	3	0	1	0	0	0
3	57	24	139	10	6	2	3	0	3	0
4	13	17	7	125	1	0	0	0	0	0
5	5	6	4	3	19	8	6	0	0	0
6	0	0	1	0	3	34	6	0	0	0
7	5	0	3	0	2	5	20	0	0	0
8	10	7	2	6	2	0	3	0	0	0
9	2	4	2	0	0	0	2	0	10	0
10	0	0	0	0	1	0	0	0	0	4

Here, weighting leads to an increase of overall accuracy (61.9%) and also of recall of class 2 (54.8%). However, the recall value of LCF is not reached. With l optimized for correctly identifying nuclear proteins ($l = 0.5$) we even obtain 66.0 % recall in class 2, but overall accuracy decreases to 56.8 % mainly because of incorrectly classified instances of the biggest class 1.

4.6 *E.coli* dataset

Similar to the yeast dataset, *E.coli* data describe seven protein location sites distributed to eight classes, i.e. cytoplasm (143), inner membrane without signal sequence (77), periplasm (52), inner membrane, uncleavable signal sequence (35), outer membrane (20), outer membrane lipoprotein (5), inner membrane lipoprotein (2) and inner membrane, cleavable signal sequence (2) (Horton and Nakai, 1996, 1997). Table 6 shows the confusion matrix for the *E.coli* dataset. Parameters for LCF were set to $k = 10$ and $l = 0.1$ Table 7 depicts precision and recall for the classes 2 and 4, the accuracy on these classes and the overall accuracy.

All examined classifiers show most classification errors due to mixing up of inner membrane proteins without a signal sequence (class 2) and inner membrane proteins with an uncleavable signal sequence (class 4). The accuracy on these classes (denoted by C) is $\sim$10% less than the overall accuracy (denoted by O). Classes 2 and 4 which are unbalanced (c.f. 77 versus 35 data) are very similar, both representing inner membrane proteins. Horton and Nakai (1997) explained the difficulty to separate both classes with the fact that the labelling of some of the training examples includes some uncertainty; that means some training instances are probably wrongly labeled. However, LCF performs best w.r.t. balancedness in these classes and is slightly better in terms of overall accuracy. Performance on the other classes corresponds well to the results described in Horton and Nakai (1997). This example shows that local density of data is useful for instance-based classification, especially if there are wrongly labeled instances. Here, the CLOF is not as sensitive as the ordinary or weighted k-NN classifier to capture wrongly labeled instances that are considered as outliers w.r.t. their own class. Test objects in their neighborhood also get a high CLOF so that they are not so likely to adopt the wrong class label.

Among the other classification methods, LRA shows best precision and recall on class 2, but performs not so well on the smaller sparser class 4. For k-NN we used $k = 7$ as described in Horton and Nakai (1997). Similar to the yeast dataset, weighting improves the result, but does not reach the results of LCF. NB tends to classify objects of classes 2–4, whereas highest recall in class 4 is achieved at the expense of recall in class 2.

Table 6. Confusion matrix for *E.coli* data with LCF

Class	1	2	3	4	5	6	7	8
1	140	0	3	0	0	0	0	0
2	3	63	2	9	0	0	0	0
3	4	1	47	0	0	0	0	0
4	1	9	0	25	0	0	0	0
5	0	0	3	0	16	1	0	0
6	0	0	1	0	3	5	0	0
7	0	1	0	0	0	1	0	0
8	0	1	1	0	0	0	0	0

Table 7. Classification results on *E.coli* data

Classifier	Class	Corr.	Incorr.	Recall	Precision	Accuracy
LRA	2	65	12	84.4	83.3	C: 77.7
	4	22	13	62..9	66.7	O: 87.2
SVM (poly)	2	64	13	83.1	84.2	C: 77.7
	4	23	12	65.7	69.7	O: 87.8
SVM (radial)	2	18	59	23.4	64.3	C: 16.7
	4	0	35	0	0	O: 47.9
7-NN	2	58	19	75.3	81.7	C: 72.3
	4	23	12	65.7	69.5	O: 86.0
7-NN (weighted)	2	63	14	81.8	84.0	C: 75.9
	4	22	13	62.9	71.0	O: 87.2
DT	2	60	17	77.9	75.0	C: 70.5
	4	19	16	54.3	55.9	O: 82.1
NB	2	56	21	72.7	87.5	C: 75.9
	4	29	6	82.9	61.7	O: 85.4
ANN	2	64	13	83.1	80.0	C: 76.8
	4	22	13	62.9	66.7	O: 86.1
LCF ($l = 0.1$)	2	63	14	81.8	82.9	C: 78.6
	4	25	9	71.4	73.5	O: 88.1

Table 8. Classification accuracy on all seven datasets in percent

Dataset	LRA	SVM	k-NN	DT	NB	ANN	LCF
Synthetic	59.2	63.2	76.3	67.8	65.1	67.8	**82.9**
Metabolic	56.1	57.9	68.4	64.9	68.4	57.9	**73.7**
Yeast	58.6	59.3	**61.9**	57.8	57.6	59.4	60.3
E.coli	87.2	87.8	87.2	82.1	85.4	86.1	**88.1**
Liver	68.1	**72.2**	59.1	68.1	55.4	71.8	70.4
Iris	94.0	**97.3**	96.0	94.0	96.0	**97.3**	**97.3**
Diabetes	**77.5**	77.3	73.2	73.8	76.3	75.3	75.1

Bold numbers indicate highest classification accuracy.

4.7 Iris, liver and diabetes dataset

Table 8 summarizes experimental results of all seven datasets including findings on three further UCI biomedical datasets (www.ics.uci.edu/~mlearn/MLSummary.html). There are only minor differences between most of the compared classifiers. The

liver data (provided by BUPA Medical Research Ltd., UK, www.bupa.co.uk) and iris dataset are rather balanced. The diabetes dataset (provided by the Washington University, St Louis, MO for the AAAI Spring Symposium on Artificial Intelligence in Medicine, 1994) has categorical and discrete valued attributes. Here, it is not likely to contain a complex data structure with areas of various densities. Nevertheless, the performance of LCF is among the best methods on these three datasets. However, model-based paradigms perform slightly better. As an efficient instance-based method, LCF performs in six of the seven datasets better than *k*-NN.

5 CONCLUSION

In this paper we focused on the problem of classification of objects using the density-based notion of clustering and outlier detection. We showed that these concepts can be successfully applied for classification in biomedicine. In particular, we proposed a local density-based classification factor combining the aspects of DD and a CLOF. A broad experimental evaluation demonstrates that our method is applicable on very different biological datasets. Our main focus here was on using multimodal unbalanced datasets. We demonstrated that our density-based classification method outperformed traditional classifiers especially on datasets representing a local cluster structure with varying density regions, which is of high practical relevance in various biomedical applications as demonstrated.

Nevertheless, there are several possible directions for future work. It would be interesting to investigate if a local adoption of the parameter *l* would yield to further improvement. Since many biological datasets are very high-dimensional, a dimensionality reduction before classification is required. It is also an interesting issue if and how the techniques of density-based clustering and subspace clustering can be used for selecting relevant attributes and especially combinations of attributes for classification, a field of our ongoing research.

ACKNOWLEDGEMENTS

The authors thank Biocrates Life Sciences GmbH, Innsbruck, Austria for providing anonymized metabolic data. Parts of this work are supported by the Austrian Research Promotion Agency Ltd. FFG (grand no. HITT-10 UMIT).

Conflict of Interest: none declared.

REFERENCES

Ankerst,M., Breunig,M.M., Kriegel,H.-P. and Sander,J. (1999) OPTICS: ordering points to identify the clustering structure. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD'99)*, Philadelphia, PA, pp. 49–60.

Baumgartner,C. and Baumgartner,D. (2006) Biomarker Discovery, Disease Classification and Similarity Query Processing on High-Throughput MS/MS Data of Inborn Errors of Metabolism. *J. Biomol. Screen.*, (in press).

Baumgartner,C. *et al.* (2004) Supervised machine learning techniques for the classification of metabolic disorders in newborns. *Bioinformatics*, **20**, 2985–2996.

Bishop,C.M. (1995) *Neural Networks for Pattern Recognition*. Oxford University Press, Oxford, UK.

Blake,C.L. and Merz,C.J. (1998) *UCI Repository of Machine Learning Databases*. Department of Information and Computer Sciences, University of California, Irvine.

Breuning,M.M., Kriegel,H.-P., Ng,R.T. and Sander,J. (2000) LOF: Identifying density-based local outliers. *Proceedings of the ACM International Conference on Management of Data (SIGMOD'00),* Dallas, TX, pp. 93–104.

Cristianini,N. and Shawe-Taylor,J. (2000) *An Introduction to Support Vector Machines and other Kernel-based Learning Methods*. Cambridge University Press, Cambridge, UK.

Cortes,C. and Vapnic,V. (1995) Support vector networks. *Mach. Learn.*, **20**, 273–297.

Ester,M., Kriegel,H.-P., Sander,J. and Xu,X. (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. In In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD'96)*, Portland, OR, pp. 226–231.

Gelman,A., Carlin,J.B., Stern,H.S. and Rubin,D.B. (2004) *Bayesian Data Analysis*. 2nd edn. Chapman & Hall/CRC Press, Boca Raton, FL.

Garcia-Bustos,J. *et al.* (1991) Nuclear protein localization. *Biochim. Biophys. Acta*, **1071**, 83–101.

Hastie,T. and Tibshirani,R. (1996) Discriminant adaptive nearest neighbor classification. *IEEE Trans. Pattern Anal. Mach. Intell.*, **18**, 607–616.

Horton,P. and Nakai,K. (1996) A probabilistic classification system for predicting the cellular localization sites of proteins. *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, **4**, 109–115.

Horton,P. and Nakai,K. (1997) Better prediction of protein cellular localization sites with the *k* nearest neighbors classifier. *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, **5**, 147–152.

Hosmer,D.W. and Lemeshow,S. (2000) *Applied Logistic Regression*. Wiley, New York.

Langley,P., Iba,W. and Thompson,K. (1992) An analysis of Bayesian classifiers. In *Proceedings of the 10th National Conference on Artificial Intelligence,* San Jose, CA, pp. 223–228.

Mitchell,T.M. (1997) *Machine Learning*. McGraw-Hill, Boston, MA.

Papadimitriou,S., Kitagawa,H., Gibbons,P.B. and Faloutsos,C. (2003) LOCI: Fast outlier detection using the local correlation integral. In In *Proceedings of the 19th International Conference on Data Engineering (ICDE'03),* Bangalore, India, pp. 315–327.

Paredes,R. and Vidal,E. (2000) A class-dependent weighted dissimilarity measure for nearest neighbor classification problems. *Pattern Recognit. Lett.*, **21**, 1027–1036.

Platt,J., Cristianini,N. and Shawe-Taylor,J. (2000) Large margin DAGs for multiclass classification. In *Proceedings of Neural Information Processing Systems (NIPS'99),* Denver, Colorado, pp. 547–553.

Quinlan,R.J. (1986) Induction of decision trees. *Mach. Learn.*, **1**, 81–106.

Quinlan,R.J. (1993) *C4.5: Program for Machine Learning*. Morgan Kaufmann, San Mateo, CA.

Vapnic,V. (1998) *Statistical Learn Theory*. Wiley, New York.

Xie,Z., Hsu,W., Liu,Z. and Lee,M.-L. (2002) SNNB: A selective neighborhood based naive bayes for lazy learning. In In *Proceedings of the 6th Pacific-Asia Conference on Knowledge Discovery (PAKDD'02),* Taipei, Taiwan, pp. 104–114.

Zhao,L.J. and Padmanabhan,R. (1988) Nuclear transport of adenovirus DNA polymerase is facilitated by interaction with preterminal protein. *Cell*, **55**, 1005–1015.