Improving 1NN strategy for classification of some prokaryotic organisms

M. Grbić[1], A. Kartelj[2], D. Matić[1], and V. Filipović[2]

[1] Faculty of Science and Mathematicas, University of Banja Luka, Mladena Stojanovića 2, 78000 Banja Luka, Republic of Srpska, Bosnia and Herzegovina
milanagrbic@yahoo.com, matic.dragan@gmail.com
[2] Faculty of Mathematics, University of Belgrade, Studentski trg 16 11000 Belgrade, Serbia
{kartelj, vladaf}@matf.bg.ac.rs

Abstract. Classification algorithms are intensively used in discovering new information in large sets of biological data. In cases when classification tasks involve nominal attributes, some of commonly used classification tools do not obtain results of satisfying quality, since mathematical operations and relations can not be directly applied to symbolic values. This problem often appears in the k-nearest neighborhood (KNN) classification because the standard Euclidean distance function can become burdened by the large number of irrelevant attributes, consequently producing inaccurate classification results.
In this paper we examine several metrics which can be applied to nominal attributes and for each metric we apply the appropriate KNN strategy. In order to justify the proposed approach, comprehensive experiments are performed on a dataset of prokaryiotic organisms. Experimental results indicate that the new classifications are more accurate than those obtained by the previously used methods, getting better results in seven of total of twelve cases.

Keywords: bioinformatics, classification, nearest neighbor, distance metrics, data mining

1. Introduction

There is a fast growth in the volume of data stored in biological databases. One of the particularly active area in bioinformatics is the development and application of the machine learning methods and classification algorithms in order to obtain more useful information from large sets of biological data.

During the classification process, the classifier uses a set of training records with known classes in order to learn how to predict the class of an record with an unknown class. During past decades, many power and robust classification tools have appeared on the market. Some of the most popular and frequently used such tools offers various techniques, allowing users to try and compare different machine learning methods on new and existing data sets.

Among many other commonly used software framework used for classification, we notice some of them: WEKA [1], KNIME [2], IBM SPSS [3] and IBM Intelligent Miner package [4].

Although these tools are proved to be reliable in many tasks, in classifications which involve many nominal attributes, these tools often do not obtain results of satisfying quality, since mathematical operations and relations need additional customization with respect to the nature of the considered data. As a consequence, such classifiers ignore nominal attributes and form the classification model based solely on numerical attributes. This approach usually leads to inaccurate and unreliable results. The problem of inability to handle nominal attributes especially appears in the classification algorithms based on the KNN strategy. KNN involves a distance function that measures the difference or similarity between two records. In KKN, there is an assumption that the class of a test record is equal to the most frequent class of the nearby records with respect to distance function, e.g. Euclidean distance function. When the classification algorithm has to deal with many nominal attributes, the calculation of the distance between two records can become burdened by the large number of irrelevant attributes. In such cases, we get inaccurate classification results or even no result at all, if all attributes are nominal, since the application of KNN strategy is impossible in such case.

To overcome these problems and enable the application of a KNN classifier to such datasets, new distance functions between attributes needs to be defined. In this paper we examine several metrics known in the literature, which can be applied to nominal attributes of a dataset of prokaryotic organisms.

The dataset analyzed in this paper consists of prokaryotic organisms and contains total of 30 attributes, from which 11 attributes are nominal. Earlier experiments presented in [5, 6] indicated that commonly used classification tools, mostly ignore nominal attributes and forms the classification based on only numerical ones. For each analyzed metric we apply the appropriate KNN strategy, enabling the classification process become more accurate.

This paper is organized as follows. In the next section we present a short description of the KNN method, as well as the overview of the metrics which are convenient for use in determining the distance between the considered data. In the section Experimental results we tested all of these metrics by applying the KNN strategy using them. We compare obtained results with the results of other classification methods presented in [6].

2. Nearest-neighbor classifier and distance metrics

In this section we give a short description of the nearest neighbor classifier and the distance metrics used in this paper.

2.1. Nearest neighbor classifier

Nearest neighbor classifier is a relatively simple and common used classification method which can be applied both for classification and regression. Since this method delays the process of modeling the training data until it is needed to classify test examples, this classifier is known as lazy learner. As a consequence, the efficiency of the KNN depends on the dimension of the training set (N_{tr})

and the number of attributes (N). For example, for each training vector the time complexity of 1-NN is $O(N_{tr}N)$.

The main principles of this method are based on finding all the training examples that are relatively similar to the attributes of the test example. Each example is represented as data point in n-dimensional space, where n is number of attributes. In the algorithm, distance (or similarity) between each test example $z = (x', y')$ and all the training examples $(x, y) \in D$ are calculated, in order to determine the nearest-neighbor list D_z. The k nearest neighbors of a given example z refer to the k training examples that are closest to z. These examples are further used to determine the class label of the test example. More precisely, once the nearest-neighbor list is obtained, the test example is classified based on majority class of its neighbors:

$$y' = \arg\max_{v} \sum_{(x_i, y_i) \in D_z} I(v = y_i),$$

where v is class label, y_i is class label for one of the nearest neighbors, and $I(\cdot)$ is an indicator function that returns value 1 if its argument is true and 0 otherwise.

The choice of the number k can significantly influence on the success of the classification. In some cases, if k is too small, the nearest-neighbor classifier may be susceptible to overfitting because of noise in the training data. On the other side, if k is too large, the nearest-neighbor classifier may misclassify the test record because its list of nearest neighbors may include data points that are located far away from its neighborhood [7].

If KNN is used for solving binary classification tasks, odd values of k are usually used to avoid ties, i.e., two classes labels achieving the same score. In the KNN presented in this paper, k takes each value from the set $\{1, 3, 5, 7, 9, 11, 13, 15\}$.

2.2. Distance metrics

As it is already mentioned, in cases when many nominal attributes are included in classification, standard metrics often can not be directly applied, since nominal attributes must be handled in a problem-specific way. In literature many distance functions for handling the nominal attributes are proposed. A detailed analysis of them is out of the scope of this paper and can be found for example in [8] and the references therein.

In this paper, for improving the classification process based on the KNN strategy, we decided to implement and test the following distance functions:

- Hamming-Euclidean overlap metric (HEOM)
- Frequency weighted overlap metric (FWOM)
- Heterogenous Valued difference metric (HVDM), actually three variants of this metric, slightly differing in the way of calculating the valued distance.

In addition, we implemented the numeric metric, which handles only the numeric attributes. We use this metric for calculating the distance between numerical attributes in HEOM, FWOM and HVDM metrics.

In the following subsections, we shortly describe the introduced metrics.

Numeric metric. Numeric metric ignores nominal attributes and bases classification model only on numerical ones. If $x = (x_1, x_2, ..., x_n)$ and $y = (y_1, y_2, ..., y_n)$ are two examples then

$$numeric(x,y) = \sqrt{\sum_{i=1}^{n} d_i^2(x_i, y_i)}.$$

In the considered dataset there are many NULL values assigned to numerical attributes. If in a pair of attributes only one NULL value appear, than we handle the problem in a simple, but effective way: the distance between the attribute x_i having a numeric value and the attribute y_i having NULL value is calculated as difference between x_i and the average value of the attribute i. In a case when both attributes are missing, than the distance is equal to 0. More precisely, the distance between two attributes is calculated by the following formula:

$$d_i(x_i, y_i) = \begin{cases} |x_i - y_i|, & x_i, y_i \neq NULL \\ |x_i - avg(i)|, & y_i = NULL \ \wedge \ x_i \neq NULL \\ |y_i - avg(i)|, & x_i = NULL \ \wedge \ y_i \neq NULL \\ 0, & \text{otherwise.} \end{cases}$$

$avg(i)$ is average value of i-th attribute.

Since the value $d_i(x_i, y_i)$ can be very large, it is divided by 4 standard deviations to scale value into a range that is usually of width 1. The numeric features are therefore normalized with $d(x_i, y_i) = \dfrac{|x_i - y_i|}{4\sigma}$, where σ is standard deviation.

HEOM metric. Hamming-Euclidean metric is a heterogeneous metric that use different attributes distance function on different kinds of attributes. This metric is introduced by Wilson and Martinez [8] and it is the combination of the Euclidean and Hamming metric. For nominal attributes, the Hamming distance is considered: the distance is equal to 0 if two attributes are equal and 1 if they are different or one of them is NULL. If attributes are numerical, the HEOM metric uses the Euclidean distance, which is similar as the distance calculated in the numeric metric.

Formally,

$$d_i(x_i, y_i) = \begin{cases} Hamming\ distance & \text{, if } i\text{-th attribute nominal;} \\ Euclidean\ distance & \text{, if } i\text{-th attribute numeric.} \end{cases}$$

Euclidean distance for attributes x_i and y_i is calculated as $|x_i - y_i|$. Similarly to the case of numeric metric, $d_i(x_i, y_i)$ can be very large, so numeric features are normalized by the formula $d(x_i, y_i) = \dfrac{|x_i - y_i|}{4\sigma}$, where σ is standard deviation.

Finally, if $x = (x_1, x_2, ..., x_n)$ and $y = (y_1, y_2, ..., y_n)$ are two examples then

$$heom(x,y) = \sqrt{\sum_{i=1}^{n} d_i^2(x_i, y_i)},$$

where $d_i(x_i, y_i)$ is Hamming-Euclidean distance.

FWOM metric. In HEOM metric all attributes have identical contributions to the overall distance. One way to control the influence of attributes is introducing different weights on different attributes. This approach is applied in the frequency weighted overlap metric (FWOM). The FWOM metrics is introduced in [9] and has a similar definition as HEOM metric, but the nominal attributes are assigned the appropriate weights, defined as

$$\omega_i = \frac{F(x_i) + F(y_i)}{F(x_i)F(y_i)}$$

where $F(x_i)$ and $F(y_i)$ are the frequencies of the attributes x_i and y_i in training data.

So, if $x = (x_1, x_2, ..., x_n)$ and $y = (y_1, y_2, ..., y_n)$ are two examples then

$$fwom(x,y) = \sum_{i=1}^{n} d(x_i, y_i),$$

where

$$d_i(x_i, y_i) = \begin{cases} \omega_i \cdot Hamming\ distance & \text{, if } i\text{-th attribute nominal;} \\ Euclidean\ distance & \text{, if } i\text{-th attribute numeric.} \end{cases}$$

HVDM metric. In HVDM metric, the valued difference metric instead Hamming metric is used for determine distance between nominal values. Valued difference metric is is defined as [8]:

$$HVDM(x,y) = \sqrt{\sum_{i=1}^{n} d_i^2(x_i, y_i)}$$

where n is number of attributes.

The function $d_i(x,y)$ returns a distnace between the two values x and y for attribute i and is defined as:

$$d_i(x,y) = \begin{cases} 1 & \text{, if } x \text{ or } y \text{ unknown;} \\ normalized\_vdm_i(x,y) & \text{, if } i \text{ is nominal;} \\ normalized\_diff_i(x,y) & \text{, if } i \text{ is numerical.} \end{cases}$$

The function $normalized\_diff_i$ is defined similarly to the previous cases when numerical attributes figure:

$$normalized\_diff_i(x,y) = \frac{|x-y|}{4\sigma_i}.$$

In order to deeper analyze the behaviour of the HVDM metric applied to the considered dataset, we considered three variants of calculating the distance between two records:

$$N1 : normalized\_vdm1_i(x,y) = \sum_{c=1}^{C} \left| \frac{N_{i,x,c}}{N_{i,x}} - \frac{N_{i,y,c}}{N_{i,y}} \right|,$$

$$N2 : normalized\_vdm2_i(x,y) = \sqrt{\sum_{c=1}^{C} \left| \frac{N_{i,x,c}}{N_{i,x}} - \frac{N_{i,y,c}}{N_{i,y}} \right|^2},$$

$$N3 : normalized\_vdm3_i(x,y) = \sqrt{C * \sum_{c=1}^{C} \left| \frac{N_{i,x,c}}{N_{i,x}} - \frac{N_{i,y,c}}{N_{i,y}} \right|^2},$$

where $N_{i,x}$ is the number of records in the training set that have value x for attribute i, $N_{i,x,c}$ is the number of records in the training set that have value x for attribute i and output class c. C is the number of output classes in the problem domain.

The difference between $N1$ and $N2$ is similar to a difference between Manhattan and Euclidean distance, while $N3$ is the function used in [10], where HVDM was first introduced. Using VDM, the average value for $N_{a,x,c}/N_{a,x}$ (as well as for $N_{a,y,c}/N_{a,y}$) is $1/C$. Since the difference is squared and then added C times, the sum is usually in the neighborhood of $C(1/C^2) = 1/C$. This sum is therefore multiplied by C to get it in the range $0,...,1$, making it roughly equal in influence to normalized numeric values.

3. Experimental results

This section contains experimental results obtained by application of the KNN classification algorithm to the chosen dataset of prokaryotic organisms.

All the tests are executed on the Intel i3-4000M CPU @2.4GHz with 12GB RAM under 64-bit Windows 10 Operating system. For each execution, only one thread/processor is used. The KNN algorithm is implemented in C programming language and compiled with Visual Studio 2012 compiler.

For each problem dataset, KNN is executed multiple times by selecting K from the set $\{1,3,5,7,9,11,13,15\}$ and by selecting different distance metric from the set {HEOM, FWOM, HVDM1, HVDM2, HVDM3, Numeric}.

Initially, each problem dataset is randomly separated to two parts: the first one is called the training subset and it consists of about 70% of records from the whole problem dataset, while the remaining 30% of records (test subset) is used to test the quality of KNN for a selected K and distance metric. Test accuracy is calculated as a percentage of accurately assigned classes to feature vectors from the test set. During this class assignment, only nearest neighbours from the training subset are considered.

3.1. Dataset collection

The data used in this work refer to the prokaryotic organisms. The data are extracted from the NCBI (National Center for Biotechnology Information) site (http://www.ncbi.nlm.nih.gov/genomes/lproks.cgi, as of February 9th, 2012). Later, some characteristics of organisms were added from the Patric (http://patricbrc.org) and Doe databases (http://img.jgi.doe.gov/). All data were stored in the table entitled Characteristics of Organisms. That table contains 1971 different records and for each record there are total of 30 attributes. In this research

TABLE 1: Considered classifications

Class.	Attributes	Target class	#nominal att.
Class 1	shape, organism size, arrangement	superkingdom	2
Class 2	shape, organism size, arrangement	phylum	2
Class 3	shape, motility, endospores	superkingdom	3
Class 4	shape, motility, endospores	phylum	3
Class 5	habitat, temp range, optimal temp	superkingdom	2
Class 6	habitat, temp range, optimal temp	phylum	2
Class 7	temp range, optimal temp	habitat	1
Class 8	pathogenic, oxygenreq, optimal temp	superkingdom	2
Class 9	pathogenic, oxygenreq, optimal temp	phylum	2
Class 10	oxygenreq, optimal temp	pathogenic	1
Class 11	habitat, motility	superkingdom	2
Class 12	habitat, motility	phylum	2

10 attributes are used: shape, organism size, motility, habitat, optimal temp, arrangement, endospores, pathogenic, oxygenreq and temperature range.

In the Table 1 we show an overview of 12 classifications analyzed in this work. The first column contains labels of classifications, the second column is the list of attributes which are used in particular classification, the third is the target class and in the forth column we show the number of nominal attributes used in the classification. For example, the first classification uses attributes: shape, organism size and arrangement. The target class is superkingdom (which can take values Bacteria or Archea). Attributes shape and arrangement are nominal, organism size is numerical, so there are two nominal and one numerical attribute in this classification. The third classification uses three nominal attributes (shape, motility, endospores) to form classification model for same target class as the first one.

3.2. Results of the classifications

In the Tables 2-13 the results obtained on these 12 classifications are shown. The first column of each table contains the number of considered neighbours, the second column contains the result obtained by HEOM metric and the third one by FWOM metric. The next three columns contain results obtained by the HVDM metrics (HVDM1, HVDM2, HVDM3 respectively), and the last column contains results obtained by using the numeric metric. The best obtained result is bolded. Since all attributes in classifications 3, 4, 11 and 12 are nominal, numeric metric can not be applied to them.

From the Table 2 (classification 1) one can see that the best score of correctly classified test data is obtained by the HEOM for 11 neighbors and with two variants of the HVDM metrics (HVDM1 and HVDM2) for 13 neighbors. The weakest results are obtained by the numeric metric. From the Table 3 it can be seen that the best result is obtained by the HEOM metric and 11 neighbors. Numeric metric again gives the weakest results, which indicates that the proposed distance metrics improves the success of the classification. In the third classification (Ta-

TABLE 2: Class. 1: Target class Superkingdom

	HEOM	FWOM	HVDM1	HVDM2	HVDM3	Numeric
1NN	90.88%	91.39%	90.71%	90.71%	90.71%	89.70%
3NN	94.76%	94.93%	94.93%	94.93%	94.76%	92.57%
5NN	95.44%	95.44%	95.78%	95.78%	95.78%	94.09%
7NN	95.27%	95.10%	95.27%	95.27%	95.27%	93.92%
9NN	95.78%	95.61%	95.61%	95.61%	95.61%	94.26%
11NN	**95.95%**	95.78%	95.78%	95.78%	95.61%	94.43%
13NN	95.44%	95.27%	**95.95%**	**95.95%**	95.27%	94.26%
15NN	95.44%	95.27%	95.44%	95.44%	95.44%	94.26%

TABLE 3: Class. 2: Target class Phylum

	HEOM	FWOM	HVDM1	HVDM2	HVDM3	Numeric
1NN	52.20%	53.21%	40.20%	40.20%	40.20%	41.39%
3NN	56.42%	56.42%	43.92%	43.92%	43.92%	43.41%
5NN	59.29%	59.12%	50.00%	50.00%	50.00%	48.48%
7NN	58.95%	58.61%	50.51%	50.51%	50.51%	49.16%
9NN	60.47%	60.14%	51.69%	51.69%	51.69%	50.84%
11NN	**60.64%**	60.14%	51.86%	51.86%	51.86%	51.01%
13NN	60.47%	60.30%	52.53%	52.53%	52.53%	50.34%
15NN	58.45%	58.45%	51.18%	51.18%	51.18%	51.35%

TABLE 4: Class. 3: Target class Superkingdom

	HEOM	FWOM	HVDM1	HVDM2	HVDM3	Numeric
1NN	93.41%	93.41%	93.41%	93.41%	93.41%	-
3NN	76.01%	76.01%	76.01%	76.01%	76.01%	-
5NN	**94.93%**	**94.93%**	**94.93%**	**94.93%**	**94.93%**	-
7NN	**94.93%**	**94.93%**	**94.93%**	**94.93%**	**94.93%**	-
9NN	**94.93%**	**94.93%**	**94.93%**	**94.93%**	**94.93%**	-
11NN	**94.93%**	**94.93%**	**94.93%**	**94.93%**	**94.93%**	-
13NN	**94.93%**	**94.93%**	**94.93%**	**94.93%**	**94.93%**	-
15NN	**94.93%**	**94.93%**	**94.93%**	**94.93%**	**94.93%**	-

TABLE 5: Class. 4: Target class Phylum

	HEOM	FWOM	HVDM1	HVDM2	HVDM3	Numeric
1NN	54.39%	54.39%	42.91%	42.91%	42.91%	-
3NN	52.03%	52.03%	42.91%	42.91%	42.91%	-
5NN	51.86%	51.86%	31.93%	31.93%	31.93%	-
7NN	55.91%	55.91%	31.93%	31.93%	31.93%	-
9NN	55.74%	55.74%	42.91%	42.91%	42.91%	-
11NN	56.93%	56.93%	42.91%	42.91%	42.91%	-
13NN	**58.78%**	**58.78%**	42.91%	42.91%	42.91%	-
15NN	**58.78%**	**58.78%**	42.91%	42.91%	42.91%	-

TABLE 6: Class. 5: Target class Superkingdom

	HEOM	FWOM	HVDM1	HVDM2	HVDM3	Numeric
1NN	95.61%	95.27%	95.27%	95.27%	95.61%	94.59%
3NN	96.11%	96.11%	96.11%	96.11%	96.11%	94.59%
5NN	96.11%	95.95%	95.95%	95.95%	95.95%	94.59%
7NN	96.79%	**97.13%**	96.62%	96.62%	96.62%	94.59%
9NN	96.79%	96.62%	96.62%	96.62%	96.62%	94.59%
11NN	96.79%	96.62%	96.62%	96.79%	96.62%	94.59%
13NN	96.79%	96.62%	96.62%	96.62%	96.62%	94.59%
15NN	96.45%	96.28%	96.62%	96.62%	96.62%	94.59%

TABLE 7: Class. 6: Target class Phylum

	HEOM	FWOM	HVDM1	HVDM2	HVDM3	Numeric
1NN	31.93%	31.93%	26.86%	26.86%	26.86%	39.19%
3NN	32.09%	32.09%	37.33%	37.33%	37.33%	44.59%
5NN	48.31%	**48.48%**	46.11%	48.14%	46.11%	44.59%
7NN	47.97%	47.97%	47.13%	47.13%	47.13%	44.59%
9NN	37.84%	38.01%	46.62%	46.62%	46.62%	44.59%
11NN	38.01%	38.68%	46.62%	46.62%	46.62%	44.59%
13NN	36.49%	37.33%	46.28%	46.28%	46.28%	44.59%
15NN	36.49%	36.99%	46.28%	46.28%	46.28%	44.59%

TABLE 8: Class. 7: Target class Habitat

	HEOM	FWOM	HVDM1	HVDM2	HVDM3	Numeric
1NN	29.73%	29.73%	29.73%	29.73%	29.73%	26.69%
3NN	30.57%	30.57%	30.57%	30.57%	30.57%	28.04%
5NN	**45.61%**	**45.61%**	45.44%	45.44%	**45.61%**	43.24%
7NN	44.76%	44.76%	44.76%	44.76%	44.76%	42.57%
9NN	44.76%	44.76%	44.76%	44.76%	44.76%	42.57%
11NN	44.93%	44.76%	44.93%	44.93%	44.93%	42.74%
13NN	31.93%	31.76%	31.93%	31.93%	31.93%	43.24%
15NN	30.91%	30.74%	30.91%	30.91%	30.91%	28.55%

TABLE 9: Class. 8: Target class Superkingdom

	HEOM	FWOM	HVDM1	HVDM2	HVDM3	Numeric
1NN	94.59%	94.59%	94.59%	94.59%	94.59%	94.59%
3NN	94.59%	94.59%	94.59%	94.59%	94.59%	94.59%
5NN	94.59%	94.59%	94.59%	94.59%	94.59%	95.78%
7NN	94.59%	94.59%	94.59%	94.59%	94.59%	94.59%
9NN	94.59%	**95.95%**	94.59%	94.59%	94.59%	94.59%
11NN	94.59%	94.59%	94.59%	94.59%	94.59%	94.59%
13NN	94.59%	94.59%	94.59%	94.59%	94.59%	95.78%
15NN	94.59%	95.10%	94.59%	94.59%	94.59%	94.59%

ble 4) all methods obtain similar results. In the fourth classification the best results are obtained by HEOM and FWOM metrics, for larger k, ($k = 13$ and $k = 15$). For the classification 5, all methods obtains similar results and the best one if reached by the FWOM metric and $k = 7$. In the classification 6, HVDM metrics are more successful in average, but the best result is again obtained by the FWOM metric and $k = 5$. In the classification 7 best results are obtained for $k = 5$ (HEOM, FWOM and HVDM3). In the classification 8 all methods achieves good and similar results and the best one is obtained by the FWOM metric and $k = 9$. In classification 9, FWOM and HEOM metrics obtains best results for larger values of k. In classification 10 and 11 all methods obtain similar results. In general, better results are obtained by larger values of k. In the last classification, best results are obtained by HEOM and FWOM metrics and $k = 15$.

3.3. The comparison with previous methods

Table 14 contains the results obtained by several classification algorithms: Sprinter from IBM Intelligent Miner package which is based on Decision Tree algorithm, CHAID from IBM SPSS Statistics 23 (SPSS) which is also based on the Decision tree algorithm, Nave Bayes algorithm from WEKA package and Jrip algorithm also from WEKA which is Rule-Based Classifier. All these results are extracted from [6]. The best result of each classification is bolded.

From the Table 14 it is evident that presented metrics can improve the NN strategy for classifications in $7/12$ cases. The presented strategy improves results for classifications 1-5, 8 and 11, achieving better results than those presented in [6]. The proposed methods can be applied to classification containing both numerical and nominal attributes. From a deeper analysis of the obtained data, one can conclude that HEOM and FWOM metrics behave similary. As expected, since HVDM 1-3 metrics are defined in a similar way, the obtained results are also similar.

4. Conclusions and future work

In this paper we presented several distance metrics that can be used for classifications which involve nominal attributes. In cases when many nominal attributes appear, standard classification tools usually ignore their appearance, causing inaccurate and unreliable results. In order to overcome this problem, we introduced several distance metrics that can be applied to the considered classifications.

Experimental results indicate a high reliability of the proposed methods. This strategy improves previously known results in seven of total of twelve cases. The obtained results indicate that this approach can be used for classification of such datasets.

This research can be extended in several ways. For example, the proposed algorithms can be applied to other biological datasets. In classifications where more attributes are considered, the proposed KNN approach can be combined

TABLE 10: Class. 9: Target class Phylum

	HEOM	FWOM	HVDM1	HVDM2	HVDM3	Numeric
1NN	27.36%	27.36%	28.21%	28.21%	28.21%	39.19%
3NN	27.20%	27.20%	29.39%	29.39%	29.39%	44.59%
5NN	33.45%	33.45%	36.82%	36.82%	36.82%	44.59%
7NN	44.09%	44.09%	44.59%	44.59%	44.59%	44.59%
9NN	44.09%	44.09%	44.59%	44.59%	44.59%	44.59%
11NN	43.92%	43.92%	37.84%	37.84%	37.84%	44.59%
13NN	**46.28%**	**46.28%**	44.59%	44.59%	44.59%	44.59%
15NN	**46.28%**	**46.28%**	44.59%	44.59%	44.59%	44.59%

TABLE 11: Class. 10: Target class Pathogenic

	HEOM	FWOM	HVDM1	HVDM2	HVDM3	Numeric
1NN	46.45%	46.45%	46.28%	46.28%	46.28%	47.97%
3NN	73.82%	73.82%	73.31%	73.31%	73.65%	67.06%
5NN	75.84%	75.84%	75.68%	75.68%	75.84%	66.22%
7NN	44.93%	44.93%	44.93%	44.93%	44.43%	48.65%
9NN	**77.20%**	**77.20%**	**77.20%**	**77.20%**	77.03%	66.55%
11NN	**77.20%**	**77.20%**	**77.20%**	**77.20%**	**77.20%**	66.55%
13NN	**77.20%**	**77.20%**	**77.20%**	**77.20%**	**77.20%**	66.55%
15NN	**77.20%**	**77.20%**	**77.20%**	**77.20%**	**77.20%**	66.55%

TABLE 12: Class. 11: Target class Superkingdom

	HEOM	FWOM	HVDM1	HVDM2	HVDM3	Numeric
1NN	84.29%	84.29%	84.29%	84.29%	84.29%	-
3NN	92.57%	92.57%	92.57%	92.57%	92.57%	-
5NN	92.57%	92.57%	92.57%	92.57%	92.57%	-
7NN	93.75%	93.75%	93.75%	93.75%	93.75%	-
9NN	**94.43%**	**94.43%**	**94.43%**	**94.43%**	**94.43%**	-
11NN	**94.43%**	**94.43%**	**94.43%**	**94.43%**	**94.43%**	-
13NN	**94.43%**	**94.43%**	**94.43%**	**94.43%**	**94.43%**	-
15NN	**94.43%**	**94.43%**	**94.43%**	**94.43%**	**94.43%**	-

TABLE 13: Class. 12: Target class Phylum

	HEOM	FWOM	HVDM1	HVDM2	HVDM3	Numeric
1NN	43.41%	43.41%	42.91%	42.91%	42.91%	-
3NN	44.09%	44.09%	10.64%	42.91%	42.91%	-
5NN	41.22%	41.22%	10.64%	10.64%	10.64%	-
7NN	41.72%	41.72%	42.91%	10.64%	10.64%	-
9NN	44.26%	44.26%	42.91%	42.91%	42.91%	-
11NN	45.95%	45.95%	42.91%	42.91%	42.91%	-
13NN	45.95%	45.95%	42.91%	42.91%	42.91%	-
15NN	**48.14%**	**48.14%**	42.91%	42.91%	42.91%	-

TABLE 14: Comparative results obtained by different classification algorithms

Class	Sprinter	CHAID	Naïve Bayes	Jrip	KNN	Improved NN
class1	93.75%	92.70%	92.56%	93.74%	90.37%	**95.95%**
class2	46.40%	51.10%	-	53.98%	49.16%	**60.64%**
class3	21.07%	93.30%	92.72%	94.25%	-	**94.93%**
class4	1.00%	56.50%	53.13%	54.48%	-	**58.78%**
class5	83.79%	96.60%	94.25%	95.77%	92.51%	**97.13%**
class6	4.00%	47.50%	-	43.82%	**52.02%**	48.48%
class7	8.00%	52.20%	49.07%	47.20%	**55.35%**	45.61%
class8	87.23%	94.00%	93.06%	94.59%	91.57%	**95.95%**
class9	9.00%	44.30%	0.00%	43.32%	**49.09%**	46.28%
class10	48.34%	**83.60%**	64.47%	82.24%	76.65%	77.20%
class11	55.83%	92.70%	92.22%	92.22%	-	**94.43%**
class12	0.00%	47.30%	**48.90%**	46.02%	-	48.14%

with a feature selection algorithm. It would be also interesting to analyse other distance metrics that can be adopt for classifications of prokaryotic organisms.

References

1. M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. H. Witten: The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1, (2009)
2. M.R. Berthold, N. Cebron, F.Dill, T. R. Gabriel, T. Kötter, T. Meinl, P. Ohl, C. Sieb, K. Thiel, B. Wiswedel: KNIME: The Konstanz Information Miner, Studies in Classification, Data Analysis, and Knowledge Organization (GfKL 2007), Springer, (2007)
3. IBM Corp. Released 2013. IBM SPSS Statistics for Windows, Version 22.0. Armonk, NY: IBM Corp.
4. IBM DB2 Intelligent Miner for Data, Using the Intelligent Miner for Data, First Edition, (2002)
5. Grbić M. "Analysis of classification algorithms applied to some prokaryotic organisms" (poster), The Ninth International Biocuration conference, Geneve, 2016.
6. Grbić M. "Grouping organisms by various classification methods depending on genotype and phenotype characteristics", Master thesis (in Serbian), Faculty of Mathematics, Belgrade, 2016.
7. Tan, Pang-Ning and Steinbach, Michael and Kumar, Vipin: Introduction to Data Mining. Pearson. (2006)
8. Wilson D., Martinez T.:: Improved Heterogeneous Distance Functions. Journal of Aritificial Intelligence Research. (1997)
9. Huang: A fast clustering algorithm to cluster very. In Reaserch Issues on Data Mining and Knowledge Discovery. (1997)
10. Wilson D., Martinez T.: Heterogeneous Radial Basis Function Networks Proceedings of the International Conference on Neural Networks (1996)