Multi-class nearest neighbor classifier for incomplete data handling

Bartosz A. Nowak[1], Robert K. Nowicki[2], Christian Napoli[3] and Marcin Wozniak[3]

[1] Institute of Computational Intelligence, Czestochowa University of Technology,
Al. Armii Krajowej 36, 42-200 Czestochowa, Poland
{bartosz.nowak,robert.nowicki}@iisi.pcz.pl
[2] Department of Mathematical Methods in Computer Science, University
of Warmia and Mazury, ul. Słoneczna 54, 10-710 Olsztyn, Poland
[3] Institute of Mathematics, Silesian University of Technology,
Kaszubska 23, 44-101 Gliwice, Poland
Marcin.Wozniak@polsl.pl

Abstract. In the paper this restriction has been dismissed. Through incorporation of certain elements of rough set and fuzzy set theories into k-nn classifier we obtain a sample based classifier with new features. In processing incomplete data, the proposed classifier gives answer in the form of rough set, i.e. indicated lower or upper approximation of one or more classes. The basic nearest neighbor algorithm has been designed to work with complete data vectors and assumed that each reference sample as well as classified sample belongs to one and the only one class. Indication of more than one class is a result of incomplete data processing as well as final reduction operation.

Keywords: nearest neighbor, missing values, rough sets.

1 Introduction

The nearest neighbors algorithm (k-nn) is one of the most widely known classification method [3]. Despite its simplicity, or thanks to it, k-nn is still one of the most resultant ones. The main idea is quite simple. The classifier which works in input space $\Omega = \Re^n$ has the set of reference samples with known membership to one of m classes. The s–th reference sample x_r can be described by the vector $\overline{\mathbf{v}}_r = [\overline{v}_{r,1}, \ldots, \overline{v}_{r,i}, \ldots, \overline{v}_{r,n}, \omega(x_r)]$. The set is knowledge of the classifier. The goal of the method is to indicate one of m classes for the sample x described by the vector $\overline{\mathbf{v}} = [\overline{v}_1, \ldots, \overline{v}_i, \ldots, \overline{v}_n]$. In the first step we calculate distances d_r between classified sample x and each reference sample x_r, i.e.

$$d_r = \|\overline{\mathbf{v}}_r - \overline{\mathbf{v}}\|, \tag{1}$$

where definition of the norm $\|\cdot\|$ could be various, but determined in current classification process. In the second step classifier should select k reference samples which are nearest to classified one. The order of reference samples is insignificant, however to make the selection at least partial sorting is necessary. We can apply in this process

some fast input data aggregation methods [18], [19], [20]. Finally, classifier counts the representatives of particular classes in selected samples. The most strongly represented class is assigned to the classified sample x. So, the definition of the norm $\|\cdot\|$ and cardinality of selected samples (parameter k) are the only parameters of the algorithm. Let us also note, that each reference sample belongs to exactly one class and the classifier indicates only one class for classified sample (excepting the draw in the final step). The usually applied norms require that all elements of vector $\overline{\mathbf{v}}_r$ and vector $\overline{\mathbf{v}}$ have known values. We would like to face this both restriction. Why? Because these are not fulfilled in real classification tasks.

The restriction of one class for one sample is in general unnatural. Let take a red fresh tomato from greengrocer. It undoubtedly can by classified as good one to make a sandwich and as good one to make a tomato soup and as good one to make a Greek salad. Many models of cars are good to drive to shopping or lift a child to school and go to the theater. So one class is not enough. We have proposed and presented in the paper fuzzy-like solution to extend k-nn classifier and make it capable to proceed multiple classification.

The problem of incomplete input data is inherent in many real applications of decision–making systems. In the industry, some information could be unavailable due to e.g. the measuring instrument failure, temporary exceeding the measure range in some port of monitoring process. In medical diagnosis procedures some tests are omitted because of the patient state, unacceptable cost, lack of reagents or rejected by the community because of beliefs. Moreover, it could be deemed unnecessary by a doctor. The decision support system cannot remain idle in such cases. Of course, this problem can be relatively easily solved by marginalization or imputation. In the marginalization the input space Ω is temporary reduced to the lower dimensionality of consideration space to the features of known values. Therefore, some elements of the system are just turned off. Thus sometimes the elimination of all incomplete samples includes also the marginalization. However, it is eventually accepted only in developing time. When we use the imputation, the unknown values are replaced by estimated ones. The palette of available methods is generally unlimited. The most primitive ones are confined to insertion of random, average or most common values. More sophisticated ones apply EM (Expectation Maximization) or k nearest neighbor algorithms, neural networks and fuzzy systems.

In the paper has been proposed alternative solution - rough sets [12]. In this theory an object can be classified to a positive region of the class (i.e. the object certainly belongs to the class), to a negative region of the class (i.e. the object certainly not belongs to the class) or to a boundary region of the class (i.e. it is not possible to determine if the object belongs to the class or not). Membership to these regions depends on the quality of object description. If this description is good enough, the object belongs either to the positive or negative regions. If the description is too weak, then the object belongs to the boundary region. In the rough set theory [12] as well as in the theory of evidence [15], we do not use the individual elements but some granules [13]. The granules contain elements which are indistinguishable basing on knowledge that we dispose. Thus, the size and the shape of granules depend on the used (known) knowledge about the elements. Obviously, the solutions combined nearest neighbor algorithm with rough set

theory and fuzzy logic are already present in the literature. Researchers have included to his/her systems also other methods inspired by nature. Some examples are [7], [17], [21], [14], [16], [6], [5], [4] and [1]. However our proposition is different. In it, when all input features have known and available values, the classifier works as elemental k–nn system. In the case, when some value is missing all possible outputs of the classifier are estimated. When only one decision is possible, the feature with missing value could be considered negligible and the classified sample is assigned to lower approximation (positive region) of classes indicated by the classifier. When more than one decision is possible, the classified sample is assigned to upper approximation of each class indicated by the classifier. Obviously, it is not possible to test all features with missing values. In presented solution we use the intervals and specific procedure shown below. Moreover, the final answer of the classifier is based on type reduction similar to used in neuro-fuzzy inference system.

2 Interval-type rough k-nn algorithm

The paper contains proposition of the algorithm which extends k-nn method by support of interval-type value of attributes and fuzzy-rough type answer. Although the algorithm considers whole spectrum of values inside all intervals, it analyses data only in limited set of important points, but with nearly the same results. This type of input values can be caused by imprecision or missing data. In the paper we use notation that value of i-th attribute of s-th sample is defined as $[\bar{v}_{s,i*}, \bar{v}^*_{s,i}]$. All intervals are normalized by the algorithm to $[0, 1]$ range:

$$\bar{v}^{norm}_{i*} = \frac{\bar{v}^{org}_{i*} - v^{min}_i}{v^{max}_i - v^{min}_i}, \bar{v}^{*norm}_i = \frac{\bar{v}^{*org}_i - v^{min}_i}{v^{max}_i - v^{min}_i}, \tag{2}$$

where $[v^{org}_{i*}, \bar{v}^{*org}_{s,i}]$ is original value of i-th attribute, $[v^{norm}_{i*}, \bar{v}^{*norm}_{s,i}]$ is normalized value of this attribute, v^{min}_i, v^{max}_i are minimal and maximal possible values of this interval. Missing values of attribute are replaced by the most possible wide interval ($[0, 1]$). Because of interval-type of input values, the distance is also interval. In the paper the distance between samples x_a and x_b is similar to typical taxicab metric:

$$d_*(x_a, x_b) = \sum_{i=1\ldots n} \begin{cases} 0 \text{ if } [v_{a,i*}, v^*_{a,i}] \cap [v_{b,i*}, v^*_{b,i}] \neq \emptyset \\ \min \left\{ \begin{array}{l} |v_{a,i*} - v^*_{b,i}|, \\ |v^*_{a,i} - v_{b,i*}| \end{array} \right\} \text{ else,} \end{cases} \tag{3}$$

$$d^*(x_a, x_b) = \sum_{i=1\ldots n} \max \left\{ \begin{array}{l} |v_{a,i*} - v^*_{b,i}|, |v_{a,i*} - v_{b,i*}|, \\ |v^*_{a,i} - v_{b,i*}|, |v^*_{a,i} - v^*_{b,i}| \end{array} \right\}. \tag{4}$$

For simplification of notation, the distance between current test sample and reference one (x_s) is defined as $[d_{s*}, d^*_s]$. The algorithm uses vector $\mathbf{d}^{srt}$ which stores sorted values of all beginnings end endings (d_{s*}, d^*_s) of distance intervals. Repeated values of this vector ($\mathbf{d}^{srt}$) are removed. Another information required by the method is stored in vectors ψ_*, ψ^*. They define how many samples have respectively ends or starts of the

distance interval ($[d_*, d^*]$) smaller or equal to defined distance represented by $\mathbf{d}^{srt}$:

$$\psi_{c*} = \left|\{x_s : d_s^* \leq d_c^{srt}\}\right|, \psi_c^* = \left|\{x_s : d_{s*} \leq d_c^{srt}\}\right|. \tag{5}$$

The algorithm also uses matrices Ψ_*, Ψ^*. The matrix Ψ_* stores numbers of reference samples that have ends of distance intervals not greater than specified distance and are members of defined class. Content of Ψ^* is the same, but beginnings of distance intervals are considered:

$$\Psi_{c,j*} = \left|\{x_s : d_s^* \leq d_c^{srt} \wedge x_s \in \omega_j\}\right|, \Psi_{c,j}^* = \left|\{x_s : d_{s*} \leq d_c^{srt} \wedge x_s \in \omega_j\}\right|, \tag{6}$$

where ω_j is j-th class, m is number of classes. An example values for $\Psi_{c,j*}$ and $\Psi_{c,j}^*$ are visible in Figure 1. In the paper we assume that single sample may belong to any number of classes or even to none, therefore $\sum_{j=1}^{m}(\Psi_{c,j*})$ is not always equal to ψ_{c*}, and $\sum_{j=1}^{m}(\Psi_{c,j}^*)$ is not necessarily the same as ψ_c^*. The algorithm defines $\neg\Psi_{c,j*}$, $\neg\Psi_{c,j}^*$:

$$\neg\Psi_{c,j*} = \left|\{x_s : d_s^* \leq d_c^{srt} \wedge x_s \notin \omega_j\}\right| = \psi_{c*} - \Psi_{c,j*}\,, \tag{7}$$

$$\neg\Psi_{c,j}^* = \left|\{x_s : d_{s*} \leq d_c^{srt} \wedge x_s \notin \omega_j\}\right| = \psi_c^* - \Psi_{c,j}^*\,, \tag{8}$$

which are opposite to $\Psi_{c,j*}$, $\Psi_{c,j}^*$ (Eq. 6). In the standard k-nn algorithm, $\psi_* = \psi^*$, $\Psi_* = \Psi^*$ and only reference samples which have distance not grater than $d_{c_{max}}^{str}$ would be chosen for voting. The index c_{max} is the same as index of the first element in ψ that has value not smaller than the main parameter of k-nn algorithm. Therefore, typical k-nn would choose the following samples for voting:

$$\left\{x_s : d_s \leq d_{c_{max}}^{str}\right\}, c_{max} = \arg\min_c(\psi_c \geq k). \tag{9}$$

The rate of samples that belong to specified class may be defined (for k-nn) as follows:

$$\tau_j = \frac{\Psi_{c_{max},j}}{\psi_{c_{max}}}. \tag{10}$$

In the paper $\mathbf{d}_*$ is not necessarily the same as $\mathbf{d}^*$ and authors observed, that there are possible solutions for k-nn algorithm for all distances $d_{c_{min}}^{str}$, $d_{c_{min}+1}^{str}, \ldots, d_{c_{max}}^{str}$, where:

$$c_{min} = \arg\min_c\left(\overline{\psi}_c \geq k\right), c_{max} = \arg\min_c\left(\underline{\psi}_c \geq k\right). \tag{11}$$

The distance $d_{c_{min}}$ is the first distance, where at least k distances have already started or ended, whereas $d_{c_{max}}$ is the first distance, where at least k distances have ended. Proposed algorithm calculates minimum (τ_{j*}) and maximum (τ_j^*) rate of reference samples that may belong to specified class (ω_j) and which are the k (or more in case of a tie) nearest neighbors of test sample. In order to calculate these values algorithm must iterate through distances $d_{c_{min}}^{str}$, $d_{c_{min}+1}^{str}, \ldots, d_{c_{max}}^{str}$. The numbers of samples which are members and not of single class depends on d_c and are in ranges $[\Psi_{c,j*}, \Psi_{c,j}^*]$, $[\neg\Psi_{c,j*}, \neg\Psi_{c,j}^*]$, respectively. Also the number of these samples, which are members or not of that class, must be not less than k. The case in which the previous conditions are satisfied and the number of samples equals k', $k' > k$ means situation, when

$(k+1)$-th, $(k+2)$-th ... (k')-th nearest neighbors are in the same distance as k-th nearest distance (a tie), therefore they must be considered during voting. During computation of τ_j^* the biggest possible ($\Psi_{j,c}^*$) number of samples that belong to ω_j are selected, and smallest possible (up to $\Psi_{j,c*}$) number of samples which are not members of the class are chosen. There are three cases:

1. In the first one $\Psi_{j,c}^* + \neg\Psi_{c,j*} < k$, therefore the number of chosen samples which are not member of current class equals $k - \Psi_{j,c}^*$.
2. When $\Psi_{j,c}^* + \neg\Psi_{c,j*} = k$, the number of chosen samples which are not members of current class are set to $k = \neg\Psi_{c,j*}$.
3. In the last case $\Psi_{j,c}^* + \neg\Psi_{c,j*} > k$ the number of chosen samples which are not members of current class equals $\neg\Psi_{c,j*}$.

$$\tau_j^* = \max_{c=c_{min}\dots c_{max}} (\tau_{c,j}^*), \tag{12}$$

$$\tau_{c,j}^* = \begin{cases} \dfrac{\Psi_{c,j}^*}{k} & \text{if } (\Psi_{c,j}^* + \neg\Psi_{c,j}^*) < k \\ \dfrac{\Psi_{c,j}^*}{k} & \text{if } (\Psi_{c,j}^* + \neg\Psi_{c,j*}) = k \\ \dfrac{\Psi_{c,j}^*}{\Psi_{c,j}^* + \neg\Psi_{j,c^*}} & \text{if } (\Psi_{c,j}^* + \neg\Psi_{c,j*}) > k. \end{cases} \tag{13}$$

After transformation:

$$\tau_j^* = \max_{c=c_{min}\dots c_{max}} \left(\frac{\Psi_{c,j}^*}{\max\left\{k, \Psi_{c,j}^* + \neg\Psi_{\sim c,j*}\right\}} \right). \tag{14}$$

Similarly to τ_j^*, during calculation of τ_{j*} the smallest possible number of samples that belong to the current class are selected, and number of samples which are not members of current classes equal $\neg\Psi_{c,j*}$. There are also three cases:

1. When $\Psi_{c,j*} + \neg\Psi_{c,j}^* < k$, the number of samples that belong to current class equal $k - \neg\Psi_{c,j*}$.
2. If $\Psi_{c,j*} + \neg\Psi_{c,j}^* < k$, then number of samples that belong to current class defined by $\Psi_{c,j*}$.
3. Otherwise number of samples that belong to current class is set to $\Psi_{c,j*}$.

$$\tau_{j*} = \min_{c=c_{min}\dots c_{max}} (\tau_{c,j*}), \tag{15}$$

$$\tau_{c,j*} = \begin{cases} \dfrac{k - \neg\Psi_{c,j}^*}{k} & \text{if } (\Psi_{c,j*} + \neg\Psi_{c,j}^*) < k \\ \dfrac{\Psi_{c,j*}}{k} & \text{if } (\Psi_{c,j*} + \neg\Psi_{c,j}^*) = k \\ \dfrac{\Psi_{c,j*}}{\Psi_{c,j*} + \neg\Psi_{c,j}^*} & \text{if } (\Psi_{c,j*} + \neg\Psi_{c,j}^*) > k. \end{cases} \tag{16}$$

After transformation,

$$\tau_{j*} = \min_{c=c_{min}\dots c_{max}} \left(\frac{\max\left\{k - \neg\Psi_{c,j}^*, \Psi_{c,j*}\right\}}{\max\left\{k, \Psi_{c,j*} + \neg\Psi_{c,j}^*\right\}} \right). \tag{17}$$

For the same example as in Figure 1 the values of $\tau_{c,j*}$ and $\tau_{c,j}^*$ are shown in Figure 2.

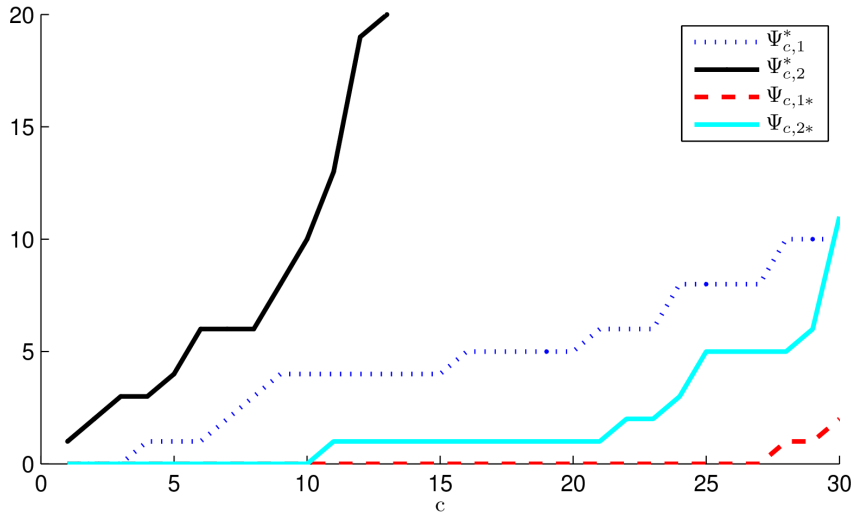

Fig. 1: Values of $\Psi_{c,j*}$ and $\Psi^*_{c,j}$ for 90% samples of BCW database and artificial test sample with values of all attributes equal 0.5 (after normalisation), and the first attribute with missing values. For this example $c_{min} = 8$, $c_{max} = 30$.

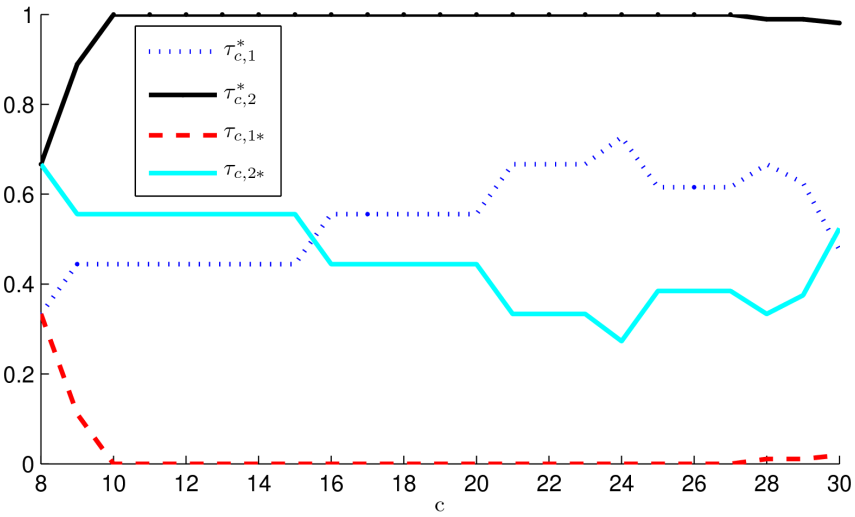

Fig. 2: Values of $\tau_{c,j*}$ and $\tau^*_{c,j}$ for the same situation as in Figure 1.

3 Simulations

Authors have examined influence of existence of missing values in test sample to results of the system. Proposed algorithm was tested using 10-fold cross validation [8] and databases from UCI Repository [2]. Therefore all results are averages for 10 simulations. For test purposes five databases were chosen, and parameter k was set empirically for all of them in range $[5, 13]$. The number of samples was equal from 214 to 699, the number of attributes from 4 to 34 and number of classes from 2 to 3. Properties of used databases, and chosen values of k are visible in Table 1. Proposed algorithm

Table 1: Properties of used databases from UCI repository and used value of k parameter

Data set name	k	number of samples	number of attributes	number of classes
Glass Identification	11	214	9	2
Ionosphere	9	351	34	2
Iris	5	150	4	3
Wine	11	178	13	3
Breast Cancer Wisconsin (BCW)	13	699	9	2

returns interval-type answer ($[\tau_{j*}, \tau_j^*]$) for each class for classification of test sample. This type of result is also lower (τ_{j*}) and upper (τ_j^*) approximation of membership of test sample to the class ω_j. All used datasets were published for benchmarking of classifiers. Although in used data sets single sample is member of only one of available classes, the algorithm operates separately of each class membership, and can be used for cases, where single sample is member of any number of classes. Proposed algorithm was tested using two criteria:

1. The first criterion compute rate of incorrect classification of sample to each class using threshold 0.5,

$$\varphi(X^{(t)}) = 1 - \frac{1}{m \cdot M^{(t)}} \sum_{s=1}^{M^{(t)}} \sum_{j=1}^{m} \begin{cases} 1 & \text{if } \begin{pmatrix} \tau_{j,s}^* > 0.5 \\ \wedge \tau_{j,s*} > 0.5 \\ \wedge x_s^{(t)} \in \omega_j \end{pmatrix} \vee \begin{pmatrix} \tau_{j,s}^* < 0.5 \\ \wedge \tau_{j,s*} < 0.5 \\ \wedge x_s^{(t)} \notin \omega_j \end{pmatrix}, \\ 0 & \text{otherwise,} \end{cases} \tag{18}$$

   where $X^{(t)}$ is set of test samples, $M^{(t)}$ is number of test samples, m is number of classes, $x_s^{(t)} \in \omega_j$ defines that test sample ought to be member of j-th class.
2. The second criterion compute rate of situation, where classifier refuse to answer if test sample is member of a class,

$$\psi(X^{(t)}) = \frac{1}{m \cdot M^{(t)}} \sum_{s=1}^{M^{(t)}} \sum_{j=1}^{m} \begin{cases} 1 & \text{if } \begin{pmatrix} \tau_{j,s}^* \geq 0.5 \\ \wedge \tau_{j,s*} \leq 0.5 \end{pmatrix}, \\ 0 & \text{otherwise.} \end{cases} \tag{19}$$

The results of evaluation of proposed algorithm are presented in Tables 2, 3, 4, 5, 6. With increase of number of removed values of attributes level of φ fall towards 0, in cost of increase of ψ.

Table 2: Results for Ionosphere Data Set

list of removed attributes	results [%] φ	ψ
$\emptyset$	13.1	0
v_{29}	7.7	13.7
v_{18}, v_{29}	4.5	25.1
v_{10}, v_{18}, v_{29}	0.8	52.1
$v_{10}, v_{18}, v_{29}, v_{33}$	0	77.2
$v_{8}, v_{10}, v_{18}, v_{29}, v_{33}$	0	86.3
$v_{8}, v_{10}, v_{15}, v_{18}, v_{29}, v_{33}$	0	91.2
$v_{7}, v_{8}, v_{10}, v_{15}, v_{18}, v_{29}, v_{33}$	0	98
$v_{7}, v_{8}, v_{10}, v_{15}, v_{18}, v_{27}, v_{29}, v_{33}$	0	100

4 Conclusions

Proposed algorithm is generalization of k-nn method. It returns interval type membership of test sample to each class, caused by interval-type value of attributes. In the research intervals were used only to represent missing values, but could be also used to present uncertainty of processed data. Proposed algorithm caused, that after increase of the set of removed values of attributes rate of wrong classification has not increased. It occurs because classifier does not give incorrect answers as the result of missing values, in other words if the classification is proper for complete input vector, it is also proper in case of incomplete data. In this properties result is similar like in rough-neuro-fuzzy classifiers presented in [9], [10], [11]. Moreover, processing with the rates τ allows to built system more suitable to work with samples could be belong to more than on class. The experimental studies confirm the expectations.

Acknowledgments The project was funded by the National Science Centre under decision number DEC-2012/05/B/ST6/03620.

References

1. Y. Bao, X. Du, and N. Ishii. Improving performance of the k-nearest neighbor classifier by tolerant rough sets. In *Cooperative Database Systems for Advanced Applications, 2001. CODAS 2001. The Proceedings of the Third International Symposium on*, pages 167–171, 2001.

Table 3: Results for Glass Identification Data Set

list of removed attributes	results [%] φ	ψ
v_9	0.5	33.7
v_8	0	79.4
v_7	0	41.2
v_6	0	69.2
v_5	0.5	23.4
v_4	0	37.3
v_3	0	95.3
v_2	0	28.6
v_1	0.5	34.7
$\emptyset$	8	0
v_8	0	79.4
v_2, v_8	0	100

Table 4: Results for Iris Data Set

list of removed attributes	results [%] φ	ψ
v_4	0	74.9
v_3	0	71.3
v_2	0	51.6
v_1	0	49.1
$\emptyset$	3.6	0
v_1	0	49.1
v_1, v_2	0	93.3
v_1, v_2, v_4	0	100

Table 5: Results Wine Data Set

list of removed attributes	results [%] φ	ψ
v_{13}	0	39.9
v_{12}	0	32.7
v_{11}	0	28
v_{10}	0	34.8
v_9	0	27.4
v_8	0	23.8
v_7	0	31
v_6	0	25.6
v_5	0	30.3
v_4	0	22.8
v_3	0	21.9
v_2	0	32.2
v_1	0	34.2
$\emptyset$	1.5	0
v_4	0	22.8
v_2, v_4	0	73.2
v_2, v_4, v_8	0	98.1
v_2, v_4, v_5, v_8	0	100

Table 6: Results for Breast Cancer Wisconsin Data Set

list of removed attributes	results [%] φ	ψ
v_9	0.1	24.3
v_8	0	27.6
v_7	0.4	18.8
v_6	0.3	30.9
v_5	0.1	19.3
v_4	0.3	24
v_3	0.1	21.3
v_2	0.1	23.5
v_1	0.6	20.9
$\emptyset$	2.9	1.6
v_8	0	27.6
v_8, v_9	0	94.7
v_2, v_8, v_9	0	100

2. collective work. Uci machine learning repository. http://archive.ics.uci.edu/ml/datasets.html.
3. T. Cover and P. Hart. Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on*, 13(1):21–27, January 1967.
4. M. He and Y. ping Du. Research on attribute reduction using rough neighborhood model. In *Business and Information Management, 2008. ISBIM '08. International Seminar on*, volume 1, pages 268–270, Dec 2008.
5. N. Ishii, I. Torii, Y. Bao, and H. Tanaka. Modified reduct: Nearest neighbor classification. In *Computer and Information Science (ICIS), 2012 IEEE/ACIS 11th International Conference on*, pages 310–315, May 2012.
6. N. Ishii, I. Torii, Y. Bao, and H. Tanaka. Mapping of nearest neighbor for classification. In *Computer and Information Science (ICIS), 2013 IEEE/ACIS 12th International Conference on*, pages 121–126, June 2013.
7. J. Keller, M. Gray, and J. Givens. A fuzzy k-nearest neighbor algorithm. *Systems, Man and Cybernetics, IEEE Transactions on*, SMC-15(4):580–585, July 1985.
8. R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. pages 1137–1143. Morgan Kaufmann, 1995.
9. R. Nowicki. On combining neuro-fuzzy architectures with the rough set theory to solve classification problems with incomplete data. *IEEE Trans. Knowl. Data Eng.*, 20(9):1239–1253, 2008.
10. R. Nowicki. Rough-neuro-fuzzy structures for classification with missing data. *IEEE Trans. Syst., Man, Cybern. B*, 39, 2009.
11. R. Nowicki. On classification with missing data using rough-neuro-fuzzy systems. *International Journal of Applied Mathematics and Computer Science*, 20(1):55–67, 2010.
12. Z. Pawlak. *Rough Sets: Theoretical Aspects of Reasoning About Data*. Kluwer, Dordrecht, 1991.
13. W. Pedrycz and A. Bargiela. Granular clustering: A granular signature of data. *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics*, 32(2):212–224, Apr. 2002.
14. M. Sarkar. Fuzzy-rough nearest neighbors algorithm. In *Systems, Man, and Cybernetics, 2000 IEEE International Conference on*, volume 5, pages 3556–3561 vol.5, 2000.
15. G. Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, Princeton, New Jork, 1976.
16. N. Verbiest, C. Cornelis, and R. Jensen. Fuzzy rough positive region based nearest neighbour classification. In *Fuzzy Systems (FUZZ-IEEE), 2012 IEEE International Conference on*, pages 1–7, June 2012.
17. T. Villmann, F. Schleif, and B. Hammer. Fuzzy labeled soft nearest neighbor classification with relevance learning. In *Machine Learning and Applications, 2005. Proceedings. Fourth International Conference on*, pages 11–15, Dec 2005.
18. M. Woźniak, Z. Marszałek, M. Gabryel, and R. K. Nowicki. Modified merge sort algorithm for large scale data sets. *Lecture Notes in Artificial Intelligence - ICAISC'2013*, 7895(PART II):612–622, 2013.
19. M. Woźniak, Z. Marszałek, M. Gabryel, and R. K. Nowicki. On quick sort algorithm performance for large data sets. In A. M. J. Skulimowski, editor, *Looking into the Future of Creativity and Decision Support Systems*, pages 647–656. Progress & Business Publishers, Cracow, Poland, 2013.
20. M. Woźniak, Z. Marszałek, M. Gabryel, and R. K. Nowicki. Triple heap sort algorithm for large data sets. In A. M. J. Skulimowski, editor, *Looking into the Future of Creativity and Decision Support Systems*, pages 657–665. Progress & Business Publishers, Cracow, Poland, 2013.
21. R. Yager. Using fuzzy methods to model nearest neighbor rules. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 32(4):512–525, Aug 2002.