MULTICLASS CLASSIFICATION STRATEGY BASED ON DIPOLES

Magdalena Topczewska

Faculty of Computer Science, Bialystok University of Technology

Abstract: The problem of multiclass classification is considered and resolved through the approach based on dipoles. The found hyperplane separates objects from different classes cutting between them and not through their middle. The crux is to define a suitable functional, which is small on lines with good separation power and little damage, easy to calculate and to minimize. The numerical tests were performed and the criterion modified in a way that preserves the intention of finding cuts between classes, which separate as many data points as possible. The approach was tested on some synthetic data sets using a recursive implementation.

Keywords: classification, multiclass problem, dipole

1. Introduction

The problem of classification is encountered in various areas, such as medicine to identify a disease of a patient, or industry to decide whether a defect has appeared or not. The aim of supervised classification methods is to construct a learning model from a labeled training data set to be able to classify new objects with unknown labels.

Assume that a training data set is given of the form $(\mathbf{x}_i, y_i)$, where $\mathbf{x}_i \in \mathbb{R}^n$ is a vector of attributes of the ith object and y_i is the ith class label of C_l where $l \in \{1,\ldots,k\}$. We aim at finding a learning model H such that $H(\mathbf{x}_i) = y_i$ for new unlabeled objects. The problem is simply formulated in the two class case, where the labels y_i are just +1 or -1 for the two classes involved. In such a case many different approaches have been proposed and developed over last few decades, for instance *LDA* (*Fisher's Linear Discriminant Analysis*) [11], *optimal Bayes rule*, *Support Vector Machines* (*SVM*) [22,5], classifiers using convex and piecewise linear (*CPL*) criterion functions [3,20], et al.

Zeszyty Naukowe Politechniki Białostockiej. Informatyka, vol. 8, pp. 77-90, 2011.

However, the case of multicategory classification is more complicated. Four groups of methods to solve such problems can be described. The first group includes methods, which can be naturally extended from binary problems. The second group is constituted by methods using decomposition into binary classification tasks. The third group consists of methods converted from binary to multiclass approaches by changing the criterion functions and the forth group is described by hierarchical classification methods.

1.1 Extensible methods

This group of methods gathers algorithms which may be applied regardless of the number of classes in a data set. This includes *Naive Bayes algorithm*, *k Nearest Neighbour* method, *Classification and Regression Trees* (*CART*) or *neural networks* (*NN*).

The *Naive Bayes* algorithm is a classification algorithm based on the Bayes rule. It assumes the attributes in a data set are all conditionally independent of one another. Regardless of whether the number of classes equals two or more, the method computes the posterior probability of that sample belonging to each class. The new object is classified according to the largest posterior probability using the maximum a posteriori decision rule [16].

The main idea of *k Nearest Neighbour* (*kNN*) method is to find the nearest k neighbours of a chosen or new object and then use a majority decision rule to classify the new sample [7,10]. All objects are treated as points in n-dimensional space and we imply that there is a distance or dissimilarity measure that can be computed between samples based on the independent variables, for instance Euclidean distance calculated typically. The voting majority rule of the *kNN* algorithm is not affected by the number of classes.

Next approach concerns building of decision trees. A tree tries to infer a split of the data based on the values of the attributes to produce a good generalization. The split at each node is based on the attribute that gives the maximum information gain and the leaf nodes correspond to class labels. A new unlabeled object is classified according to a path from the root node to the leaf node. Among considerable number of *Classification and Regression Trees* (*CART*) algorithms C4.5 and ID3 are widely known [17,18].

Neural networks can also be attached to the extensible methods group. *NN*s are feed-forward neural networks with signals propagated only forward through the layers units and consist of three types of layers: the input layer, which is feed with the data; the hidden layer(s) of units and the output layer of units. The output layer

has one unit for each diagnostic category, so-called 1-*of*-*k* encoding [16]. During the training phase the backpropagation learning algorithm adjusts weights of connections between units by propagation of the error among output layer and true classification labels. As the optimization tool the gradient descent method is applied to find the minimum of the error function.

The second type of neural networks in this paragraph are probabilistic neural networks (*PNN*). They belong to the Radial Basis Function (RBF) neural networks [16] and are comprised of three layers: input, hidden and output layers. The hidden layer consists of a pattern layer and a competitive layer of units. The pattern layer contains one unit for each object in the data set and applies a Gaussian density activation function, whereas the competitive layer consists of one unit for each class label which are activated only by pattern units associated only with the class of the trained object. In every unit of the competitive layer the probability of the object's membership to specific class is calculated. The output unit is activated according to the maximum probability value and the new object is classified to the activated unit's corresponding class.

1.2 Decomposition into binary classification problems

The most popular as well as basic and conceptually the simplest approach used in multiclass classification is to decompose the problem into multiple two-class classification problems and then solve them using efficient binary classifiers. There are a number of different approaches to decompose a *k*-class classification problem into two-class problems.

The first approach is called *one-versus-rest* (*OVR*) and the *i*th constructed binary classifier separates the *i*th class versus all other $k-1$ classes. The combined *OVR* decision function chooses the class for a new object that usually corresponds to the minimum value of the Hamming distance or the maximum value of a posteriori probability of the object's membership to each class. Even if described method is very simple, it might not be effective.

Another approach is *one-versus-one* (*OVO*) method. This method constructs one binary classifier for every pair of distinct classes and so, all together $\binom{k}{2} = k*(k-1)/2$ binary classifiers are constructed and using max-wins voting to decide to which class the new object should be placed.

In the *p-versus-q* (*PVQ*) approach *p* of the *k* classes are separated from the other *q* of the *k* classes. The process is repeated several times, each time a mix of *p* different classes against *q* different classes are chosen.

More complicated decomposition scheme is the Error-Correcting Output Codes (ECOC) method. The task is to convert k class classification problem into a large number l of binary problems and to use a unique codeword to a class instead of assigning each class label. An error correcting code is a l bit long, having unique codewords with a Hamming distance. Several methods for generating error correcting codewords and determining the l number were presented such as BCH codes [12,4], exhaustive codes [9], random codes [14] or scheme by Allwein et al. [1].

1.3 Reformulations of the objective function

This group of methods uses the conversion of criterion functions from binary to multiclass classification problems and includes among others multiclass Support Vector Machines (*MC-SVM*) and approaches by Weston and Watkins [24] and Crammer and Singer [8].

Method by Weston and Watkins is viewed as a natural extension of the binary SVM classification task. In the k-class case a single quadratic optimization problem of size $(k-1)*n$ is solved. This is identical to binary SVM when the number of classes is equal 2. The method reduces the number of support vectors needed to describe the decision functions [24].

Method proposed by Crammer and Singer [8] is similar to above-mentioned and the same size of $(k-1)*n$ quadratic problem is solved. The difference lies in using smaller number of slack variables in the constraints of the optimization problem. For both approaches the use of decompositions can provide a significant speed-up in the solution of the optimization problem [13].

1.4 Hierarchical classification

Another group of methods to solve the multiclass classification problem consists of approaches dividing the input space in hierarchical manner. Starting from the root node classes are divided into clusters and such a process is continued for each child node until the leaf nodes contain all objects from all classes of the data set. Finally, all k classes are arranged as a tree and classification of a new unlabeled object goes according to the path from the root to the leaf node. Detailed description of such methods can be found in [19]. Below only few methods are mentioned among many other .

Hierarchical Support Vector Machines (*HSVM*) solve a series of max-cut problems. Hierarchical and recursive partition of the set of classes into two-subsets, until the pure leaf nodes that have only one class label, are obtained [6]. The edge weights

of the undirected graph represent the Kulback-Leiber distance between the classes and that is used to find subclusters mostly distant from each other. Then at each internal node the SVM is applied to construct the discriminant function for a binary classifier.

Binary Hierarchical Classifier [15] is another approach. The algorithm builds a binary tree with k leaf nodes, each corresponding to one class, using $k-1$ binary classifiers. At root the algorithm finds the best feature projection that distinguishes the two groups and then the binary split into two clusters of classes is done. The process is repeated for subsequent nodes. This approach was performed comparable to ECOC, with the added advantage of using fewer classifiers [2]

Next described approach is called *Divide-By-2* [23] and as previous one uses only $k-1$ binary classifiers to form a binary tree. Instead of Fisher Discriminant, either k-means algorithm is applied for clustering the class means into two groups or the classes grand mean is used as a threshold. The method puts classes with means smaller to the grand mean in one cluster and those with larger mean to the other [2]. Binary classification is performed by binary SVM.

2. Approach based on dipols

The proposed approach for multiclass classification belongs to the hierarchical methods family [21]. In the first step the algorithm finds the *best* split of the data. We try to divide all the objects into two clusters to obtain the smallest number of wrongly classified objects. Objects belonging to different classes should not be situated on different sides of the cutting hyperplane. Next steps of the algorithm are repeated in two subspaces separately finding best splits of clusters of remained data. Finally, the hierarchy in a form of a tree is obtained. The path from the root to the leaf nodes determines the membership of the object to the appropriate class.

Linear classification for the two class case assumes that a suitable functional Φ: $\mathbb{R}^n \to \mathbb{R}$, $\Phi(\mathrm{x}) = \mathrm{w}^T\mathrm{x} - b$, can be constructed, such that for some indices l $\Phi(\mathrm{x}) < 0$ for all $\mathrm{x} \in C_l$, while for the remaining indices the corresponding value will be positive. In the case of k=2, one such functional suffices.

In the multiclass case the essential is to formulate the criterion function which is unaffected by scattered data and outliers that can occur in real data sets. The proposed criterion function is based on the dipoles that denote ordered pairs of data vectors. Concerning the two class classification case two types of dipoles are available - clean and mixed ones. The dipole is clear if both objects belong to the same class, otherwise it is mixed. Having k classes, there are k sorts of clean dipoles and $k*(k-1)$ mixed dipole types. A mixed dipole is good to be reflected by a functional featuring different

signs on the dipole's elements. On the other hand, mixed signs on a clean dipole are considered bad. One might simply count good and bad situations and make a balance. But of course, such an assessment is difficult to optimize, since it leads to an integer, hence discontinuous, objective.

Moreover, we prefer our assessment also to reflect how bad a bad cut is, and how safe a clean cut is. For example, a clean dipole (x,y) with $\Phi(x) = 0.001$ and $\Phi(y) = 1000$ is a potential risk, that a small perturbation of Φ may render it mixed, for x may go through the line. Analogously, a mixed dipole separated by Φ is not so safe if one of its parts has a small value $|\Phi(x)|$.

If the whole data set is represented by an $m \times (n+1)$ matrix D, where m denotes number of objects in a data set and n number of comprising values of attributes and knowing class indices, the proposition is to minimize a sum of the form

$$F(\Phi;\mathbf{D}) = \sum_i \sum_j \sum_{\mathbf{x}\in C_i} \sum_{\mathbf{y}\in C_j} F_{ij}(\Phi,\mathbf{x},\mathbf{y};\mathbf{D}) \tag{1}$$

where F is a function of the linear affine functional Φ. The space of all such functionals may be parametrized by a vector $\mathbf{p} = (\mathbf{w},b) \in \mathbb{R}^{m+1}$. The value of F depends obviously on the data, i.e. on the set of all feature vectors and their assignment to the k different classes.

The assessment function F is a double sum over class indices of contributions being double sums of terms of the form $F_{ij}(\mathbf{x},\mathbf{y})$. Each such term is a function of two real values that the functional Φ takes on the dipole with elements $\mathbf{x}$ and $\mathbf{y}$. For $i = j$ a dipole is clean. If $\Phi(\mathbf{x})$ and $\Phi(\mathbf{y})$ differ in sign, F_{ii} should be positive correspondingly. The bigger term in absolute value sets the sign of the dipole. The error is measured by the distance of the other one from the selected sign. The convex and piecewise linear functions studied in [3,21] may be applied to such a problem. We make F_{ii} the bigger the farther the smaller of the terms lays on the wrong side, starting from a given threshold on the good side.

The way of presenting the contributions F_{ij} opens another option. Basing on the whole set of data, we may decide that class C_i is positive (negative), and then calculate errors as in before. This would avoid making an individual decision for each dipole with both objects belonging to C_i. In fact, it is feasible to calculate the total error contribution for the clean dipoles from this class for both orientations and then take the smaller one.

In the case of $i \neq j$, different signs of $\Phi(\mathbf{x})$ and $\Phi(\mathbf{y})$ are in principle desired. To avoid problems with the mixed orientation, from i,j the index of the class with the highest value of Φ is chosen and class with this index is declared as the positive one. Then the piecewise linear error contributions for data from the positive class with too

small Φ value are added, and likewise for data from the negative class which are too large. It proved sensible to consider values below a positive ε already as too small for the positive class, and likewise above a negative ε as too big for the negative class. Obviously, for each pair (i, j) with $i \neq j$ there is no need to consider (j, i), hence the second sum may be restricted to $j \leqslant i$.

The main drawback of the method is evidently the high number of terms in the quadruple sum. Besides, in the classical setting, the sum of error terms is convex and piecewise linear, which makes it comparatively amiable for minimization. The present task is inherently nonconvex.

3. Results

With the aim of presenting the performance of proposed approach, a few examples are shown below.

3.1 Example 1

As a first example a synthetic data set containing four classes was generated. In every class there are 250 objects.

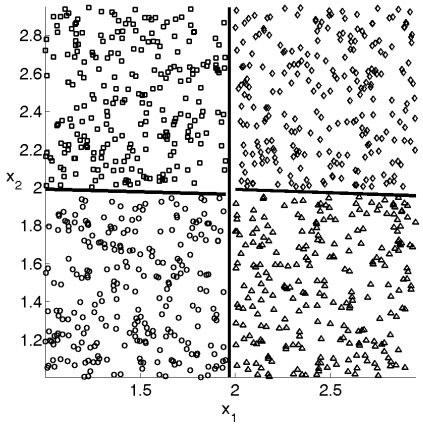
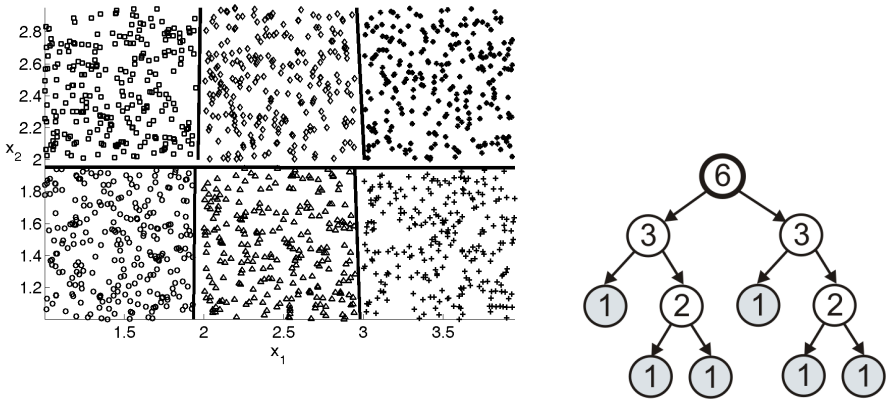


Fig. 1: Hierarchical classification of four classes

In the first step of the algorithm two classes are separated from the other two. Next, each halfplane is divided separately into two quarters with the final classification accuracy of 100%, see Fig. 1. Every quarter is associated with only one class.

3.2 Example 2

The second presented example describes classification problem of synthetic data sets and consisting of objects belonging to six and twelve different classes respectively. In each class there are 250 objects for the six classes data set and 100 objects for the remain data set.

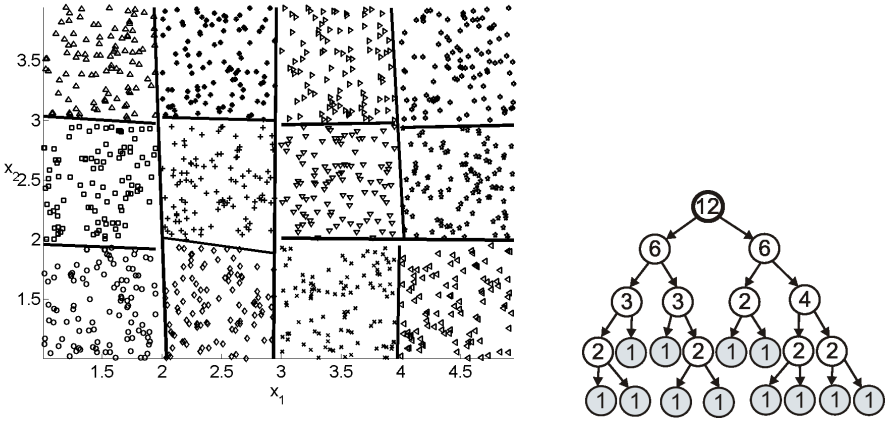


(a) data and successive splits (b) number of divided classes

Fig. 2: Hierarchical classification of six classes

In the first step of the algorithm six classes are divided into two clusters, each of three classes. In the next step, separately for two halfplanes the space is divided into two and one class and finally two remained classes are separated by a line. Analogically, three areas each associated with different class are obtained on the other side of a halfplane, see Fig. 2.

In the twelve class case the first split divided all classes into two groups containing 6 classes each. Next,recursively two clusters of 3 classes were achieved, whereas at the other side of the hyperplane the split gave a cluster of 2 and a cluster of 4 classes. Next eight splits divided space into membership areas of every class, see Fig. 3.



(a) data and successive splits (b) number of divided classes

Fig. 3: Hierarchical classification of twelve classes

3.3 Example 3

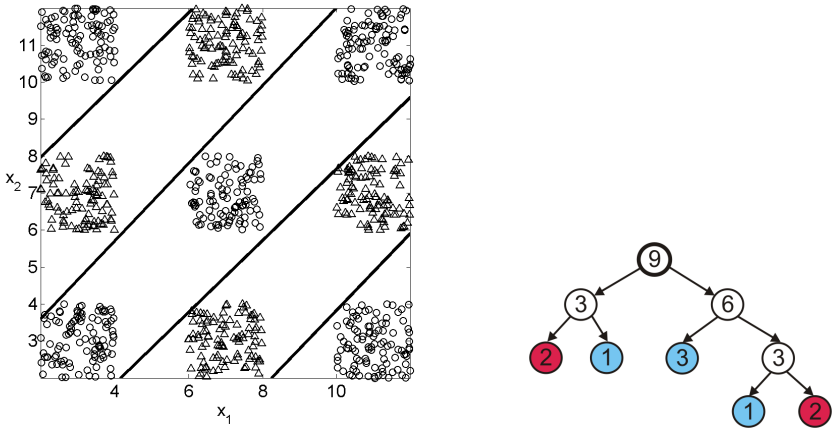
The third presented example describes classification problem of synthetic data sets and consisting of objects belonging to two classed set as a chessboard.

First division selected two clusters - 6 at the positive side of the hyperplane and 3 at the negative one. Next three splits gave areas of membersip of objects to two classes, see Fig. 4.

Calculations were performed in the Matlab system. Improvement of implementation by application of meshgrid shortened time about twenty times in relation to primal version of the method.

4. Conclusions

Finding a separating hyperplane for two classes by minimizing an error functional summing contributions for each poorly classified data point is by now common practice. Effective implementations in the framework of SVM or in terms of CPL functions are available and shown to work well for quite large sets of data. An approach based on dipoles is not needed. If, however, the number of classes increases, a dipole based criterion may be helpful to split large data first into smaller subsets, each containing not only less feature vectors, but above all a smaller number of classes. We start from a theoretical formulation of a criterion suitable for this task, modeled on



(a) data and successive splits (b) number of divided clusters

Fig. 4: Hierarchical classification of two classes

the approach presented in [3]. Next, we perform numerical tests and modify the criterion in a way that preserves the intention of finding cuts between classes and not through their middle, which separate as many data points as possible. However, we do not count in integers, but weigh by taking the distance of the object to the wrong side. This approach was tested on some synthetic data and performed well enough to be considered for implementation into a recursive production code.

The impact of quick and automatic classification of data in any field of computer aided activity human, like design, manufacturing, quality control or medicine, diagnostics, image analysis can hardly be overestimated. The presented problem is hard because of its inherently non-convex character. The solution applicable since it does not require human intervention. It easily and quickly breaks down a multiclass problem by divide and conquer into a sequence of smaller and smaller problems, which in a post-processing step may be handled by classical methods available for the two-class case.

The plan is to perform comparative analysis with the algorithms of the hierarchical classification family.

References

[1] E.L. Allwein, R.E. Schapire, Y. Singer, Reducing multiclass to binary: a unifying approach for margin classifiers, *Journal of Machine Learning Research*,

1:113-141, 2001.

[2] M. Aly, Survey on multiclass classification methods, Technical Report, Caltech, USA, 2005.

[3] L. Bobrowski, *Data exploration based on convex and piecewise criterion functions* (in Polish), Wydawnictwo Politechniki Białostockiej, Białystok, 2005.

[4] R.C. Bose, D.K. Ray-Chaudhuri, On A Class of Error Correcting Binary Group Codes, *Information and Control* 3:68–79, 1960.

[5] C.J. Burges, A tutorial on support vector machines for pattern recognition, *Data Mining and Knowledge Discovery*, 2:121-167, 1998.

[6] Y. Chen M.M. Crawford, J. Ghosh, Integrating Support Vector Machines in a hierarchical output space decomposition framework, In Proceedings of 2004 International Geoscience and Remote Sensing Symposium, Anchorage, Alaska, vol. 2, 949–953, 2004.

[7] T.M. Cover, P.E. Heart, Nearest neighbor pattern classification, *IEEE Trans. Inform. Theory*, IT 13:21-27, 1967.

[8] K. Crammer, Y. Singer, On the learnability and design of output codes for multiclass problems. *Proceedings of the Thirteen Annual Conference on Computational Learning Theory* (COLT 2000), Stanford University, Palo Alto, CA, June 28 - July 1, 2000.

[9] T.G. Dietterich, G. Bakiri, Solving multiclass learning problem via error correcting codes, *Journal of Artificial Intelligence Research*, 2:263-386, 1995.

[10] O.R. Duda, P.E. Heart, D.G. Stork, *Pattern Classification*, Second edition, John Wiley & Sons, 2001.

[11] R. A. Fisher, The use of multiple measurements in taxonomic problems, *Annals of Eugenics*, 7:179–188, 1936.

[12] A. Hocquenghem, Codes correcteurs d'erreurs (in French), Chiffres, Paris, 2, 147–156, 1959

[13] C.W. Hsu, C.J. Lin, A comparison of methods for multi-class support vector machines, *IEEE Trans. Neural Netw.*, 13:415-425, 2002.

[14] J.E. Gentle, *Random Number Generation and Monte Carlo Methods*, Springer Verlag, 1998.

[15] S. Kumar, J. Ghosh, M.M. Crawford, Hierarchical fusion of multiple classifiers for hyperspectral data analysis, *Pattern Analysis& Applications*, 5:210-220, 2002.

[16] T.M. Mitchell, *Machine Learning*, McGraw-Hill Science, New York, 1997.

[17] J.R. Quinlan, Induction of decision trees, *Machine Learning* 1(1):81-106, 1986.

[18] J.R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufman Publishers, Inc., 1993.

[19] C.N. Silla Jr., A.A. Freitas A survey of hierarchical classification across different application domains, Data Mining and Knowledge Discovery, 22:31-72, 2011.

[20] M. Topczewska, K. Frischmuth, Numerical aspects of weight calculation in classification methods, *In PTSK Conference*, Krynica Górska, Poland, Sept. 26-29, 2007.

[21] M. Topczewska, K. Frischmuth, Classification strategies based on dipols (in Polish), *Pomiary, Automatyka, Kontrola*, 56(6):632-635, 2010.

[22] V. N. Vapnik, *Statistical learning theory* Wiley J., 1998.

[23] V. Vural, J.G. Dy, A hierarchical method for multi-class support vector machines. In *Proceedings of the Twenty-First International Conference on Machine Learning*, 105-112, 2004.

[24] J. Weston, C. Watkins, Support vector machines for multi-class pattern recognition, In *Proceedings of the Seventh European Symposium On Artificial Neural Networks* (ESANN 99), Bruges, April 21-23, 1999.

STRATEGIA KLASYFIKACJI WIELOKLASOWEJ OPARTA NA DIPOLACH

Streszczenie W pracy rozpatrywane jest zagadnienie klasyfikacji w przypadku wieloklasowym oraz podejście oparte na dipolach. Poszukiwana hiperpłaszczyzna powinna rozdzielać obiekty należące do różnych klas, ale nie przecinając środka żadnej klasy. Zdefiniowano w tym celu odpowiedni funkcjonał, by przyjmował on małe wartości w przypadku prawidłowej klasyfikacji większości obiektów, był prosty do obliczenia i minimalizacji. Przeprowadzono testy numeryczne oraz dokonano modyfikacji kryterium, by znaleźć takie rozdzielenie klas, by odseparować możliwie dużo obiektów. Podejście było testowane na wybranych syntetycznych zbiorach danych przy wykorzystaniu implementacji w postaci wywołań rekurencyjnych.

Słowa kluczowe: klasyfikacja, problem wieloklasowy, dipol