Proceedings of the 2013 Federated Conference on
Computer Science and Information Systems pp. 99–105

RBF ensemble based on reduction of DAG structure

Marcin Luckner
*Faculty of Mathematics and Information Science
Warsaw University of Technology
pl. Politechniki 1, 00-661 Warszawa, Poland
Email: mluckner@mini.pw.edu.pl

Karol Szyszko
†Faculty of Mathematics and Information Science
Warsaw University of Technology
pl. Politechniki 1, 00-661 Warszawa, Poland
Email: szyszkok@student.mini.pw.edu.pl

***Abstract*—Binary classifiers are grouped into an ensemble to solve multi–class problems. One of proposed ensemble structure is a directed acyclic graph. In this structure, a classifier is created for each pair of classes. The number of classifiers can be reduced if groups of classes will be separated instead of individual classes. The proposed method is based on the similarity of classes defined as a distance between classes. For near classes the structure of DAG stays immutable. For the distant classes more than one is separated with a single classifier. In this paper, the proposed method is tested in variants based on various metrics. For the tests, several datasets from UCI repository was used and the results were compared with published works. The tests proved that grouping of radial basis functions into such ensemble reduces the classification cost and the recognition accuracy is not reduced significantly.**

***Index Terms*—Classification, Radial Basis Function, Directed Acyclic Graph, Support Vector Machines**

I. Introduction

RADIAL Basis Functions (RBF) can be used both as multi–class and binary classifiers. Binary RBF classifiers are useful, still improved linear models in a nonlinear subspace [1], [2] and can be used to create an ensemble of classifiers to solve multi–class problems. Such approach was confirmed by research on Support Vector Machines [3]. In fact, there is equivalence between a decision rule created by an SVM with radial kernels and radial networks [4]. Therefore, an ensemble of classifiers can be created for binary RBF classifiers as well as in the case of SVM equivalents.

Although one–step solutions were proposed [5], [6] to solve multi–class tasks by an SVM, they are not efficient [7]. An alternative solution is creation of an ensemble. Main approach to create an ensemble are discussed in [7], [8].

Proposed approaches are One–Against–All (one class is compared against the rest) [9], One–Against–One (a classifier is created for each pair of classes) [10], Error-Correcting Output Codes (class binarisation in order to enhance generalisation ability) [11], and several methods based on graph structures [12], [13].

Among several graph ensemble fusion methods can be used to solve multi–class problems using binary RBF classifiers [14] one of the most popular strategy is grouping SVM classifiers into a directed acyclic graphs (DAG) [13].

In the case of an n-classes problem a tree implementation requires $n-1$ classifiers and the average decision process uses $n-1/2$ classifiers. A DAG ensemble needs $n(n-1)/2$ classifiers to solve the same problem. However, only $n-1$ classifiers is used in the classification process and obtained results are usually better [15], [16].

In the work [17], the method for reduction of number of classifiers in a DAG structure was presented. The proposed method is based on a class similarity. Similar classes are grouped and separated from diametrical different classes as a whole group. In this case, a number of used classifiers is reduced. The method was projected for linear classifiers and verified on a single recognition task. The similarity calculated in the work was a derivative of the Euclidean metric. The algorithm was tested on a single recognition task.

This work is based on the algorithm proposed in [17]. However, several improvements with respect to the previously published works have been done. The algorithm has been tested on a wider set of classification problems. Researches on new problems resulted in modifications of the algorithm. The most important modification was done in limitation rules presented in Section II-C.

The algorithm was a basis for several models based on various definitions of similarity. The following metrics are used in modelling: the Euclidean, the Chebyshev, the Manhattan, the Minkowski, and the Pearson.

Models are verified on various datasets from UCI repository. Therefore, results can be compared with others works and the results of the modified algorithm were compared with results of RBF classifiers grouped into a DAG ensemble.

II. Reduced DAG ensemble

A. DAG ensemble

A directed acyclic graph G, which is described by the set of vertices $V(G)$ and the set of edges $E(G)$, can be declared as an ensemble of binary classifiers.

The vertices are grouped in layers. The first layer contains a single vertex – the root. Each subsequent layer has one more vertex. The last layer consists of n vertices where n is the number of recognised classes.

Each vertex from any layer except the last one has connections with two vertices from the next layer. Vertices on the last layer are leaves. Each leaf from $L(G)$ is connected with a final classification decision (one of the recognised classes). The rest of vertices $V(G)\backslash L(G)$ contains binary classifiers. The decision of the classifier defines which connected vertex

978-1-4673-4471-5/$25.00

will be activated next. If a leaf is activated a final decision is determined by a class connected with the leaf.

A vertex is also identified with a group of classes that can be achieved from the vertex. Each vertex can be declared as a root of sub–DAG. Such root represents a group of classes collected on the last layer of the sub–DAG. If $v \in V(G)$ is the root of the sub–DAG G_v then $L(G_v) \subseteq L(G)$ determines classes identified with the vertex.

Sub-DAGs can be also used to determine classifiers. A vertex $v \in V(G) \backslash L(G)$ has two successors v_i and v_j. The successors are identified with classes connected to $L(G_{v_i})$ and $L(G_{v_j})$ respectively. Therefore, the binary classifier in the vertex v divides dataspace between two groups of separable classes from sets $L(G_{v_i})$ and $L(G_{v_j})$.

The aim of the proposed method is to eliminate some of vertices from layers. Then an activated vertex can lie on a layer further than the next one. The example of reduced structure is presented in Figure 1. The elimination of vertices shortens the average classification time. However, not each vertex can be eliminated without a significant reduction of the accuracy ratio. Therefore, the selection of eliminated vertices will be based on similarity between classes.

B. Similarity

A similarity between classes is estimated on the base of a distance. The distance between classes $d(C_X, C_Y)$ depends on the distance between elements of those classes $d(x, y)$ and can be defined as the distance between nearest, furthest elements or as the average distance between all pair of elements. However, mentioned above distances are very time-consuming. Instead, the distance may be approximated as the distance between centroids (the centres of gravity for the classes)

$$d(C_X, C_Y) = d\left(\frac{1}{n_{C_X}} \sum_{x \in C_X} x, \frac{1}{n_{C_Y}} \sum_{y \in C_Y} y\right). \quad (1)$$

The equation (1) can be also used to calculate a distance between groups of classes. If a group is an union of classes $C_X = \bigcup_{i=1}^{k} C_i$ then all members of classes C_i, where $i = 1 \ldots k$, are treated as members of C_X. The distance between such groups can be calculated as (1).

The distance between an individual elements of the data space $d(x, y)$ depends on the selected metric. Usually it is the Euclidean metric

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

However, if the recognised elements are described by some specific features it is sometime better to select a different measure.

Two potential candidates are Manhattan and Chebyshev metrics. The Manhattan distance

$$d(x, y) = \sum_{i=1}^{n} |x_i - y_i| \quad (3)$$

should be calculated when individual features are independent, but their sum may be treated as a rational measure of the similarity. In the Chebyshev distance

$$d(x, y) = \max_{i \in \{1, \ldots, n\}} |x_i - y_i| \quad (4)$$

the similarity will depend on the maximal difference among features.

In the tests, two more metrics were used.

The Minkowski metric is a parameterised metric that becomes the Euclidean metric for $k = 2$.

$$d(x, y) = \left(\sum_{i=1}^{n} |x_i - y_i|^k\right)^{\frac{1}{k}}. \quad (5)$$

In this work $k = 3$ was used.

The Pearson metric bases on the Pearson product–moment correlation coefficient r_{xy} and presents inverse correlation between data vectors

$$d(x, y) = 1 - r_{xy}. \quad (6)$$

C. Creation of structure

The graph for the n–classes classification task has n layers where the last layer contains leaves labelled as recognised classes. The algorithm starts with the set $V(G)$ of n leaves. Therefore initially $V(G) = L(G)$.

In each step a pair of the vertices v_i and v_j is selected. The selected vertices are roots of DAGs with nearest groups of classes on the last layers $L(G_i)$ and $L(G_j)$.

$$(v_i, v_j) = \arg \min_{v_i, v_j \in V(G)} d(L(G_i), L(G_j)). \quad (7)$$

In this and the following equations, the notation $L(G_i)$ means all classes from the last layer (leaves) of the DAG with root in the vertex v_i. Therefore the distance $d(L(G_i), L(G_j))$ can be calculated as a distance between group of classes (1).

The new vertex v_k is added to the graph G. The vertices v_i and v_j are now its successors. The new graph G is given by the set of vertices

$$V(G) = \{v_k\} \cup V(G) \quad (8)$$

and the set of edges

$$E(G) = \{(v_k, v_i), (v_k, v_j))\} \cup E(G). \quad (9)$$

It is easy to observe that groups $L(G_i)$ and $L(G_j))$, determined by vertices v_i and v_j, may be still the nearest groups in the graph G. Moreover, the created group $L(G_k)$ is the union of groups $L(G_i)$ and $L(G_j)$ and consequently the union may be selected as the nearest with one of groups determined by the successors. Such situations should be avoided. Therefore, the set of vertices that are taken into consideration in the equation (7) should be limited by the following rules.

Under the second rule, that eliminates problem with the same pair selected again, the two vertices v_i and v_j can be joined if and only if the union of classes represented by them is not a subset represented by any existed vertex

$$\forall(v_k \in V(G))L(G_i) \cup L(G_j) \nsubseteq L(G_k). \tag{10}$$

The rule (10) is modification of the rule presented in [18]. The previous rule assumed that $L(G_i) \cup L(G_j) \neq L(G_k)$, however then the algorithm was not convergent for some datasets.

Under the first rule, that eliminates problem of joining successors with a predecessor, the two vertices v_i and v_j can be joined if and only if the set of classes represented by one of them is not a subset of the other

$$C(G_i) \nsubseteq C(G_j) \wedge C(G_j) \nsubseteq C(G_i). \tag{11}$$

Both conditions (11) and (10) can be used to create a limited set of allowed pairs of vertices

$$\begin{aligned} S_P &= \{(v_i, v_j) : v_i, v_j \in V(G) \\ &\wedge \quad L(G_i) \nsubseteq L(G_j) \wedge L(G_j) \nsubseteq L(G_i) \\ &\wedge \quad \forall(v_k \in V(G))L(G_k) \neq L(G_i) \cup L(G_j)\}. \end{aligned} \tag{12}$$

Moreover, the common part of classes is ignored when the distance is calculated and the final form of the formula (7) is

$$(v_i, v_j) = \arg \min_{(v_i,v_j)\in S_P} d(L(G_i)\backslash L(G_i \cap G_j), L_j \backslash L(G_i \cap G_j)), \tag{13}$$

where

$$L(G_i \cap G_j) = L(G_i) \cap L(G_j). \tag{14}$$

In each step of the algorithm, the two allowed vertices v_i, v_j are joined. The algorithm stops when no join can be made

$$\forall(v_i \in S_G)\exists(v_j \in S_G)L(G_i) \subseteq L(G_j) \vee L(G_j) \subseteq L(G_i). \tag{15}$$

D. Classification

The classification process starts in the root of DAG ensemble and finishes in a vertex from the last layer.

In the DAG ensemble, each classifier rejects one from the recognised classes. Therefore, if the vertex v_i is a root of a DAG with the last layer consists of vertices $L(G_i)$ then the vertex v_j, which is next in the classification path, is connected with classes defined by the following rule

$$L(G_j) = L(G_i) \setminus \{v_k\} \wedge v_k \in L(G_i). \tag{16}$$

In the reduced ensemble, the number of classification steps can be reduced, because a classifier can reject more than one class in one step and the rule (16) is replaced by

$$L(G_j) = L(G_i) \setminus V_k \wedge V_k \subset L(G_i). \tag{17}$$

When a vertex $v \in V(G)\backslash L(G)$ has two successors v_i and v_j, identified with classes connected to vertices $L(G_i)$ and $L(G_j)$ respectively, then the binary classifier in the vertex v divides dataspaces between two groups of separate classes $L(G_i) \setminus (L(G_i) \cap L(G_j)))$ and $L(G_j) \setminus (L(G_i) \cap L(G_j)$.

E. Time reduction

Three main elements that describe costs of the created classifiers are the construction time, the learning time, and the classification time.

The first element is a construction of the ensemble. In the case of a DAG, this element can be omitted. However, it is an important part of the presented method (described in Section II-C. If the cost of the ensemble creation is significant then potential reduction of the learning time can be balanced by a cost of the new element.

At the first glance, the learning time should be shorter in a reduced structure. However, a reduction of classifiers in the ensemble does not necessarily have to result in a reduction of the learning time. All classifiers in a DAG structure present the One–Against–One approach. Meanwhile, at least several classifiers in the reduced structure split dataspace between a class and a group of classes. It was proved that One–Against–One approach can results in lower costs than One–Against–All approach [8]. Therefore, the learning time can be longer for the reduced structure.

Considering pessimistic assumptions on the building time and the learning time, we should remember that the most important cost in the classifier evaluation is the classification time. The reduced structure has definitely lower classification costs that DAG.

In the n–classes recognition time, a DAG structure needs $n-1$ binary classifiers to assign the analysed case to one of recognised classes. If C is a set of analysed cases and C_i is a subset of cases assigned to the class labelled with i then the classification time for the set C is calculated as

$$\sum_{i=1}^{n} |C_i| * (n-1) * t = |C|(n-1) * t, \tag{18}$$

where t is the binary classification time, which should be equal for all binary classifiers from the ensemble.

In the case of the reduced structure, the same set C is defined as the union of cases $\hat{C}_i$ assigned to the class labelled with i by the new classifier. The number of binary classifiers used in the classification depends on the final classification decision and can be calculated as $d_i - 1$, where d_i describes number of vertices on the path from the root to the leaf assigned to the class labelled with i. Therefore, $d_i \leq n$.

The classification time for the set C is calculated as

$$\sum_{i=1}^{n} |\hat{C}_i| * (d_i - 1) * t. \tag{19}$$

From $d_i \leq n$ and $\sum_{i=1}^{n} \hat{C}_i = C = \sum_{i=1}^{n} C_i$ we know that

$$\sum_{i=1}^{n} |\hat{C}_i| * (d_i - 1) * t \leq \sum_{i=1}^{n} |\hat{C}_i| * (n-1) * t \leq |C|(n-1) * t. \tag{20}$$

Therefore, the classification time for the reduced structure is lower if at least one case from an analysed set belongs to the class with a reduced classification path.

III. Results and Discussion

In this work, four sets from UCI repository were used. Letter Image Recognition Data (Letter), Optical Recognition of Handwritten Digits (Optdigits), Glass Identification Database (Glass), and Wine recognition data (Wine).

TABLE I
DATASETS.

	Number of			
Dataset	Training data	Testing data	Classes	Attributes
Wine	125	53	3	16
Glass	150	64	6	16
Letter	15000	5000	26	16
Optdigits	3823	1797	10	64

In the case of Optdigits dataset and Letters relations between the training sets and the testing sets proposed in the reference works [15] and [16] had been used. However, in the case of datasets Wine and Glass solutions proposed in the reference works had been tested by the cross validation method. Therefore, the validation performance was measured by training 70 percent of the training set and testing the other 30 percent of the training set in these cases. Details on the sets are given in Table I.

The computational experiments for this section were done on an Intel Core i5–2500 with 8 GB of RAM.

All the problems were tested using RBF kernels. The accuracy rate was estimated using different kernel parameters γ and cost parameters C where $C = \{2^0, 2^1, \ldots, 2^{12}\}$ and $\gamma = \{2^{-12}, 2^{-11}, \ldots, 2^4\}$. The selection of parameters values is the most time–consuming part of the process. 221 tests must be done to check all pairs of parameters for one metric. On the testing computer, the test series lasted from few seconds to three hours depending on a dataset. Therefore, some proposals should be made to reduce the computation time.

Table II presents the accuracy ratio obtained by using different metrics. Results are very similar and any of metrics cannot be chosen as the best one. However, it can be observed that the Chebyshev and the Manhattan metrics gave worse results. The Minkowski metric, which had been gotten with $k = 3$, resulted in the accuracy similar as the Euclidean metric. In fact, the calculations can be limited to Euclidean and Pearson metrics to cover all of the best results.

The accuracy ratio obtained by the proposed method was compared with RBF classifiers ordered in the DAG structures, presented in works [15] and [16]. The detailed results including used parameters are presented in Table III.

The results of three compared approaches cannot be compared directly because works [15] and [16] covers different data set. Therefore, the proposed method should be compared with each work separately. The average accuracy ratio calculated among datasets discussed in the works is always minimal better for the proposed method.

The results of the proposed method were compared with two more state-of-the art algorithms: One–Against–All and One–Against–One methods. The average accuracy calculated for all data sets was 92.22 percent for the One–Against–One method, 91.92 percent for the One–Against–All method and 92.35 for the proposed method. Details on results and configuration are give in Table IV. The averages are very similar and any method cannot be pointed as the best one.

The most important aspect in the comparison is reduction of created vertices in the DAG structure. The number of vertices used to solve the Optdigits and Letter problems was reduced to 42 and 66 percent of vertices from the DAG structures. The proposed method reduced the number of RBF classifiers that has to be trained. Additionally, the average classification time will be shorter, because of reduction of decision paths. Two examples for Optdigits are shown in Figure 1.

Each graph has leaves marked with circles and vertices with classifiers marked with diamonds. Inside the diamonds two groups of classes are noticed. The classifier inside the vertex divides the data space between members of both groups.

The reduction of the average classification time is clearly visible in the case of the class zero. The class can be selected after three classification steps instead of nine as in the case of non–reduced DAG structure. In the proposed method, only two classes from the last layer are classified in nine steps.

The created graphs have the same number of vertices. However, the structures are different. First, numbers of vertices on layers are different. Moreover, different nearest classes were selected in the first step of the algorithm. For the Chebyshev metric, nearest classes are 1 and 8, whereas for the Euclidean metric the nearest classes are 3 and 9. The differences in the structure results in differences in the accuracy presented in Table II.

The ensembles created in the Letter problem is too complex to present a comparison between several graphs. However, the graph created using the Euclidean metric is presented in Figure 2. The letter 'L' can be recognised in 9 steps. 20 steps are needed to recognise such letters as 'K', 'G', 'Q', 'R', 'B', 'S', and 'Z'. Meanwhile the DAG structure needs 25 classifiers to recognise any letter. Therefore, both average and pessimistic costs are much lower in the proposed solution.

In the case of Glass problem, the reduction ratio is smaller. This is connected with the smaller number of recognised classes. In the Wine problem, a number of vertices cannot be reduced. However the algorithm determines the order of classifiers in the DAG. As it was shown in Table III, the determined order allows the ensemble to obtain the best result.

The method results in reduction of the classification time. In Table V, the learning time and the classification time obtained by DAG classifiers are compared with the time obtained by the proposed method. Additionally, the graph structure building time is presented in the case of the method from this work.

The proposed method reduces the classification time for complex tasks. Although, the classification time for various methods given in second has only an approximate character, the difference between methods bases on strong theoretical bases (Section II-E).

TABLE II
THE ACCURACY RATIO OBTAINED BY USING DIFFERENT METRICS. THE BEST RESULTS ARE EMPHASISED.

Dataset	Chebyshev	Euclidean	Manhattan	Pearson	Minkowski
Wine	98.86	**99.44**	**99.44**	**99.44**	**99.44**
Glass	73.79	**74.22**	73.29	73.81	**74.22**
Letter	97.42	97.63	97.63	**97.70**	97.35
Optdigits	97.5	**98.05**	97.94	**98.05**	**98.05**

TABLE III
THE ACCURACY RATIO OBTAINED BY THE PROPOSED METHOD IS COMPARED WITH RBF CLASSIFIERS ORDERED IN THE DAG STRUCTURES, PRESENTED IN WORKS [15] AND [16]. FOR EACH METHOD, USED PARAMETERS γ AND C ARE GIVEN. DAG VERTICES IS A NUMBER OF VERTICES CREATED BY DAG, AND VERTICES IS A NUMBER OF VERTICES CREATED BY THE PROPOSED METHOD. THE BEST RESULTS ARE EMPHASISED.

		Work [15]			Work [16]			Proposed method			
Dataset	DAG vertices	C	γ	accuracy	C	γ	accuracy	C	γ	accuracy	Vertices
Wine	3	2^2	2^3	**99.44**	2^8	2^{-9}	98.88	2^0	2^{-2}	**99.44**	3
Glass	21	2^{12}	2^1	73.49	2^{12}	2^{-3}	73.83	2^2	2^3	**74.22**	19
Letter	325	-	-	-	2^4	2^2	**97.98**	2^6	2^2	97.70	216
Optdigits	45	2^2	2^3	**98.44**	-	-	-	2^3	2^{-5}	98.05	19

TABLE IV
THE ACCURACY RATIO OBTAINED BY THE PROPOSED METHOD IS COMPARED WITH ONE AGAINST ONE AND ONE AGAINST ALL METHODS

	One–Against–One			One–Against–All			Proposed method		
Dataset	C	γ	accuracy	C	γ	accuracy	C	γ	accuracy
Wine	2^1	2^2	**99.44**	2^1	2^2	98.89	2^0	2^{-2}	**99.44**
Glass	2^{12}	2^1	73.01	2^{12}	2^{-3}	72.20	2^{12}	2^2	**74.22**
Letter	2^4	2^2	**97.98**	2^3	2^2	97.88	2^6	2^2	97.70
Optdigits	2^2	2^3	98.44	2^1	2^3	**98.72**	2^3	2^{-5}	98.05

TABLE V
THE LEARNING TIME AND THE CLASSIFICATION TIME FOR THE REFERENCE WORKS [15] AND [16]. THE BUILDING TIME, THE LEARNING TIME, AND THE CLASSIFICATION TIME FOR THE PROPOSED METHOD.

	Works [15], [16]		Proposed method		
Dataset	Learning [s]	Classification [s]	Building [s]	Learning [s]	Classification [s]
Wine	0.01	0.00	0.01	0.03	0.00
Glass	2.85	0.00	0.03	0.15	0.01
Letter	298.62	92.80	0.05	214.00	37.02
Optdigits	15.47	1.81	0.00	1.75	1.06

IV. CONCLUSION

In this work, the algorithm based on primary works [17], [18] was modified and used to create several models of RBF ensembles. The structures of created ensemble are reduced directed acyclic graphs. The method reduces a number of created classifiers. The classifiers, which discriminate distant classes, are replaced by the classifiers, which separate groups of classes. A theoretical estimation shows that the new structure should reduce the classification time in comparison to the DAG structure.

The algorithm was tested on four sets from UCI repository. The obtained results were compared with published results. The accuracy ratio obtained by proposed method is similar to presented in works [15] and [16]. Also results obtained by two state-of-the art algorithms One–Against–All and One–Against–One are nearly identical. The proposed algorithm should be also compared with other methods such as ECOC and Decision Template tested in [19], [20], but its main advantages is not the highest accuracy, but the significant reduction of the number of classifiers in the DAG structure.

The number of created classifiers was reduced to 42 and 66 percent of classifier from the DAG structure in some complex recognition tasks. The reduction of classifiers results in the reduction of the classification time.

REFERENCES

[1] F. Fernández-Navarro, C. Hervás-Martínez, P. Gutiérrez, M. Cruz-Ramírez, and M. Carbonero-Ruz, "Evolutionary q-gaussian radial basis functions for binary-classification," in *Hybrid Artificial Intelligence Systems*, ser. Lecture Notes in Computer Science, E. Corchado, M. GraÃąa Romay, and A. Manhaes Savio, Eds. Springer Berlin Heidelberg, 2010, vol. 6077, pp. 280–287. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-13803-4_35

[2] P. Chudzian, "Radial basis function kernel optimization for pattern classification," in *Computer Recognition Systems 4*, ser. Advances in Intelligent and Soft Computing, R. Burduk, M. KurzyÅĎski, M. WoÅşniak, and A. Å£oÅÆnierek, Eds. Springer Berlin Heidelberg, 2011, vol. 95, pp. 99–108. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-20320-6_11

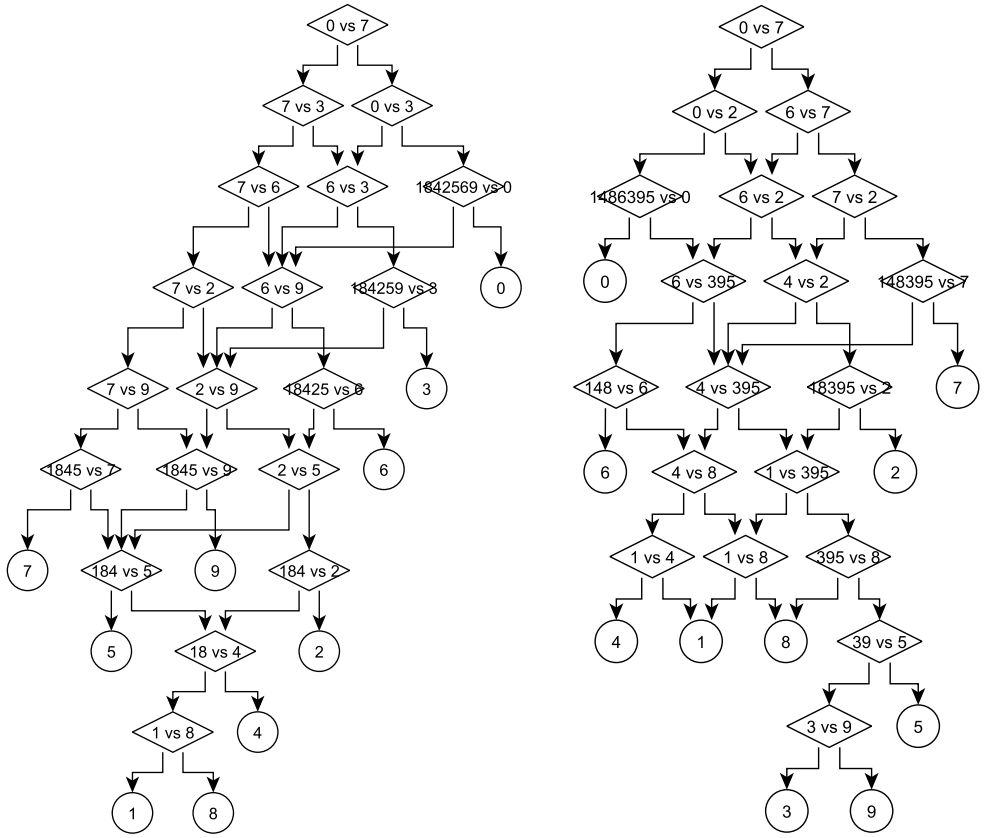

Fig. 1. Created reduced DAGs for the Optdigits recognition task. The ensembles were created using the Chebyshev 1(a) and the Euclidean 1(b) metrics.

[3] V. Vapnik, *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995.

[4] F. Girosi, "An equivalence between sparse approximation and support vector machines," *Neural Computation*, vol. 10, no. 6, pp. 1455–1480, 1998.

[5] J. Weston and C. Watkins, "Multi-class support vector machines," 1998.

[6] K. Crammer and Y. Singer, "On the learnability and design of output codes for multiclass problems," in *Proceedings of the Thirteenth Annual Conference on Computational Learning Theory*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc, 2000, pp. 35–46.

[7] H.-C. Kim, S. Pang, H.-M. Je, D. Kim, and S. Y. Bang, "Constructing support vector machine ensemble," *Pattern Recognition*, vol. 36, no. 12, pp. 2757–2767, December 2003.

[8] S. Abe, *Support Vector Machines for Pattern Classification (Advances in Pattern Recognition)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2005.

[9] K. P. Bennett, *Combining support vector and mathematical programming methods for classification*. Cambridge, MA, USA: MIT Press, 1999, pp. 307–326.

[10] U. H.-G. Kressel, *Pairwise classification and support vector machines*. Cambridge, MA, USA: MIT Press, 1999, pp. 255–268.

[11] M. Bagheri, Q. Gao, and S. Escalera, "Rough set subspace error-correcting output codes," in *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, 2012, pp. 822–827.

[12] M. A. Kumar and M. Gopal, "A comparison study on multiple binary-class svm methods for unilabel text categorization," *Pattern Recogn. Lett.*, vol. 31, pp. 1437–1444, August 2010.

[13] J. Platt, N. Cristianini, and J. ShaweTaylor, "Large margin dags for multiclass classification," in *Advances in Neural Information Processing Systems 12*, S. A. Solla, T. K. Leen, and K. R. Mueller, Eds., 2000, pp. 547–553.

[14] M. Abdel Hady and F. Schwenker, "Decision templates based rbf network for tree-structured multiple classifier fusion," in *Multiple Classifier Systems*, ser. Lecture Notes in Computer Science, J. Benediktsson, J. Kittler, and F. Roli, Eds. Springer Berlin Heidelberg, 2009, vol. 5519, pp. 92–101. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-02326-2_10

[15] Debnath, R., Takahide, N., Takahashi, and H., "A decision based one-against-one method for multi-class support vector machine," *Pattern Analysis and Applications*, vol. 7, no. 2, pp. 164–175, July 2004.

[16] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *Neural Networks, IEEE Transactions on*, vol. 13, no. 2, pp. 415–425, 2002.

[17] M. Luckner, "Reducing number of classifiers in dagsvm based on class similarity," in *Image Analysis and Processing âĂ¿ ICIAP 2011*, ser. Lecture Notes in Computer Science, G. Maino and G. Foresti, Eds., vol. 6978. Springer Berlin Heidelberg, 2011, pp. 514–523. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-24085-0_53

[18] ——, "Multiclass svm classification using graphs calibrated by similarity between classes," in *Knowledge-Based and Intelligent Information and Engineering Systems*, ser. Lecture Notes in Computer Science, A. KÃśnig, A. Dengel, K. Hinkelmann, K. Kise, R. Howlett, and L. Jain, Eds. Springer Berlin Heidelberg, 2011, vol. 6884, pp. 435–444. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-23866-6_46

[19] T. Wilk and M. Wozniak, "Soft computing methods applied to combination of one-class classifiers," *Neurocomputing*, vol. 75, no. 1, pp. 185–193, 2012.

[20] M. Galar, A. Fernández, E. B. Tartas, H. B. Sola, and F. Herrera, "An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes," *Pattern Recognition*, vol. 44, no. 8, pp. 1761–1776, 2011.

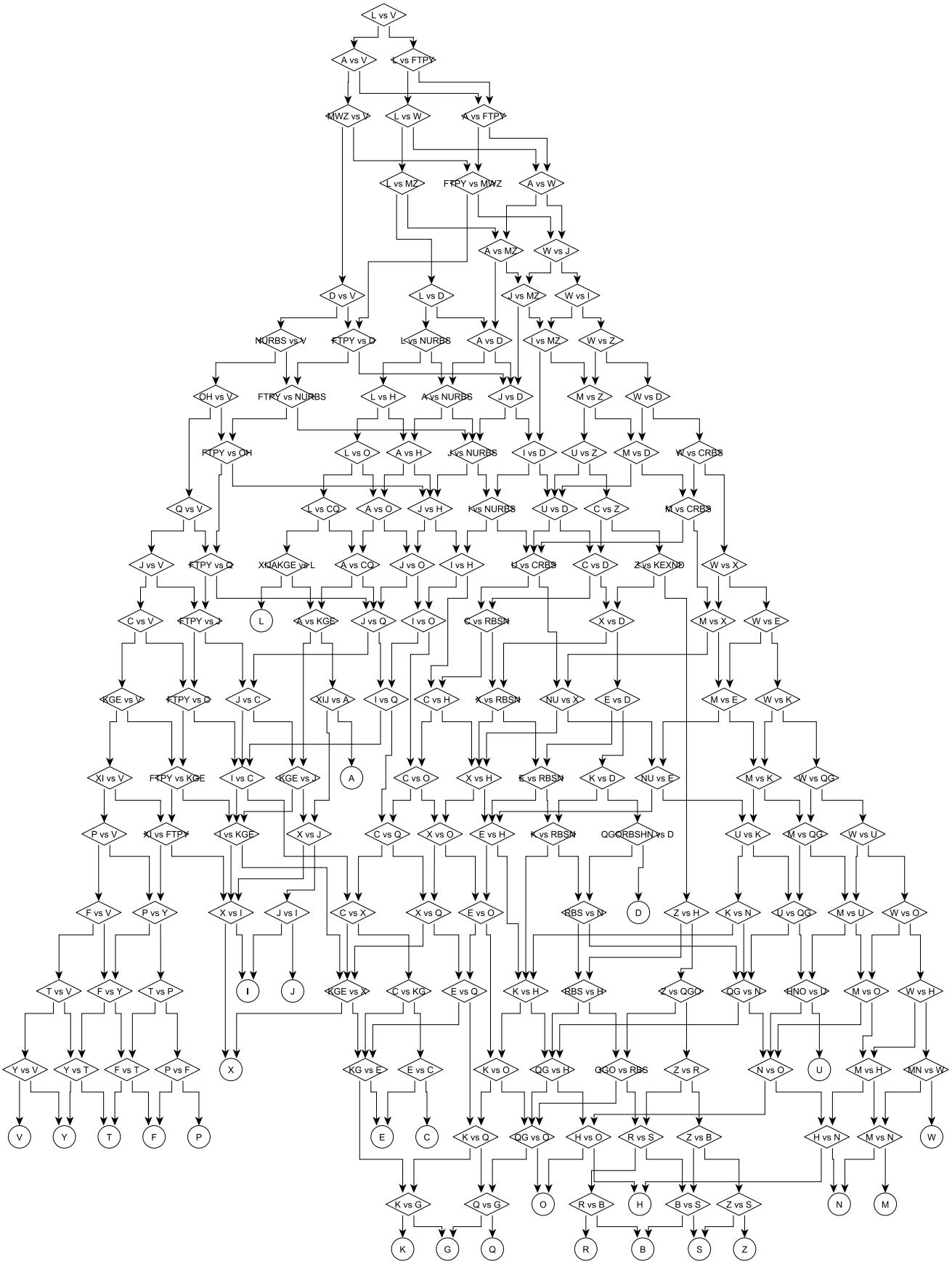

Fig. 2. Created reduced DAG for the Letters recognition task. The ensemble was created using the Euclidean metrics.