A Meta-classifier Approach for Medical Diagnosis

George L. Tsirogiannis, Dimitrios Frossyniotis,
Konstantina S. Nikita, and Andreas Stafylopatis

School of Electrical and Computer Engineering
National Technical University of Athens
Zographou 157 80, Athens, Greece

Abstract. Single classifiers, such as Neural Networks, Support Vector Machines, Decision Trees and other, can be used to perform classification of data for relatively simple problems. For more complex problems, combinations of simple classifiers can significantly improve performance. There are several combination methods, like Bagging and Boosting that combine simple classifiers. We propose, here, a new meta-classifier approach which combines several different combination methods, in analogy to the combination of simple classifiers. The meta-classifier approach is employed in the implementation of a medical diagnosis system and evaluated using three benchmark diagnosis problems as well as a problem concerning the classification of hepatic lesions from computed tomography (CT) images.

Keywords: machine learning, neural networks, diagnosis

1 Introduction

It is well known, that classifier combination approaches can provide solutions to tasks which either cannot be solved by a single classifier, or which can be more effectively solved by a multi-classifier combination scheme. The problem is that we do not know from the beginning which is the best classifier combination method for a particular classification task. In this work, we try to solve this problem by developing a new methodology that combines different combination methods in order to get better performance compared to each individual method. More specifically, in analogy to the combination methods considered, which combine simple classifiers, the proposed meta-classifier approach combines these methods at a higher level aiming at the best classification performance.

For the evaluation of our approach, we created a medical diagnosis system to classify medical data that have been collected and appropriately inserted into a knowledge base. The basic components used in the system are classifiers such as Neural Networks [18], Support Vector Machines [15, 21, 22] and C4.5 Decision Trees [5, 16, 17] along with different combination methods, such as Bagging [4] and Boosting [6, 7, 8, 9]. The key feature of the system, from a technical point of view, is that it involves an extra level above the combination of simple classifiers. Specifically, the lowest level consists of simple classifiers, whereas in the middle level there are combination methods that combine the classifiers of the level below. Such methods are Bagging, Boosting and a fuzzy multi-classifier algorithm (FuzzMCS) [10]. The upper

G.A. Vouros and T. Panayiotopoulos (Eds.): SETN 2004, LNAI 3025, pp. 154–163, 2004.

level represents the proposed meta-classifier approach that combines different methods of the middle level. The final decision of the system may be a class label with the corresponding reliability measure or a class probability. Four combination schemes of the combination methods are tested, namely, simple and weighted voting each using class labels or class probabilities. A different meta-classifier module is created for each diagnosis problem.

The meta-classifier is composed of the three levels mentioned above and operates in two phases. The first is the training phase during which the system is trained on known data for the problem. Additional parameter adaptation is embedded in the training phase which enables the system to select its parameters by its own and, thus, work autonomously without any intervention. Moreover, this feature allows the system to work properly for different medical diagnosis problems in a dynamic way. After training, the main working phase follows during which the system operates for the classification of new unlabeled data. The system has been empirically evaluated on known benchmark diagnosis problems as well as on the classification of hepatic lesions from computed tomography (CT) images [11].

As for relevant work done before, an example of an automatic diagnosis system is reported in [13]. This system tries to adapt the ECG processing and classification according to the patient. It uses a Mixture-of-Experts approach in which a Global Expert classifier is trained on a big ECG database and a Local Expert classifier is trained on a special recording of the patient's ECG. The adaptation in this case is based on the wide experience acquired by the database. Another system [23] is proposed as a general structure which allows the rule extraction of a decision. This is done by ensemble combining of Neural Networks (generalization ability) and C4.5 Decision Trees (rule readability). After experimentation, the performance of this system proved to be not reliable.

2 Meta-classifier

The meta-classifier approach extends the notion of multi-classifier combination schemes by combining methods instead of simple classifiers. While, in other words, combination methods such as Bagging and Boosting, take into consideration the decisions of different simple classifiers, such as Neural Networks, Support Vector Machines or Decision Trees, the meta-classifier combines the decisions of several different combination methods.

From an operational point of view, a specific method creates and trains a population of classifiers using training data for a particular problem. Then, for each new pattern presented to the system, each classifier assigns it to a class and, finally, the method reaches its decision by estimating which is the most probable class for the pattern, according to the decisions of the classifiers [2, 12, 14, 19]. If we consider this procedure as a compact module referred to as combination method, then the meta-classifier repeats the latter steps using combination methods instead of classifiers.

The motivation for the development of a meta-classifier approach is twofold. On the one hand, it is well known that, for complex problems, combination methods perform better than simple classifiers. Thus, it might be possible to further enhance performance by proceeding one step beyond that and combine combination methods. On the other hand, for a particular problem, there can be no prior knowledge of which

is the best method to use. It must be noted here that the selection of the best method for a particular classification problem is a time-consuming procedure and sometimes yields only indicative results. By using the combination of these methods in a meta-classifier approach, we might be able to eliminate this difficulty and obtain good performance without needing to select the best method.

3 Combination of Combination Methods

Let us consider a classification task with C classes. First, we apply M different methods to solve the problem, each of which points to one of the C classes, thus providing an output vector with elements y_i, $i=1,\ldots,C$, where $y_i \in [0,1]$ or $y_i \in \{0,1\}$ depending on the method. In the first case, each y_i can be considered as a probability measure for the corresponding class, whereas,. in the latter case, the y_i values correspond to class labels (as $y_i=1$ only when the pattern x belongs to class i).

In what concerns the combination of the methods, four different schemes are considered depending on the type of the output, as above, as well as on the voting/averaging technique, simple or weighted.

In weighted schemes, the weights correspond to a reliability measure assigned to each method, extracted from the error made on test data. We have selected this reliability to be calculated as 1 - *test error rate*. As the error rate falls into the [0,1] range, the reliability will be in the same range (the smaller the test error rate, the higher the reliability of the method).

Simple or crisp voting is a simple majority voting based on the decisions of the methods. The class with most votes will be selected as the class for the corresponding pattern.

In weighted voting, we assign a weight to each method, corresponding to its reliability as described above, and count the votes taking into account the weights

For simple averaging, the final decision will be computed by the relation $y_i = \sum_{m=1}^{M} y_i^m / M$. For weighted averaging, a normalized weighted sum is computed using the weight w^m, $m=1,\ldots,M$, for each method: $y_i = \sum_{m=1}^{M} y_i^m w^m / \sum_{m=1}^{M} w^m$.

So, we have four different combination schemes, crisp voting (class labels without weights), weighted voting (class labels with weights), average class probabilities (class probabilities without weights) and class probabilities weighted sum (class probabilities with weights).

4 Medical Diagnosis System

In this section, we describe the medical diagnosis system based on the meta-classifier approach. The system is designed to receive pre-processed arithmetic data. More specifically, the data are row vectors and each row corresponds to a pattern with the

values of the features and the label of the class. In the following, we present some special features of the system, pertaining to its hierarchical organization, automatic adaptation to the problem and parallel operation.

4.1 Hierarchical Organization

The system includes three types of modules that perform classification. The first type concerns simple classifier modules, like MultiLayered Perceptrons (MLPs), Support Vector Machines (SVMs) with RBF kernel, SVMs with polynomial kernel and C4.5 Decision Trees. The second type concerns combination methods that combine the above simple classifiers. Three algorithms are used in this system, namely Bagging, AdaBoost.M2 [9] and the FuzzMCS method that uses both supervised and unsupervised learning. Totally, ten methods are formed (each algorithm with each classifier type, excluding the use of SVMs of both types with AdaBoost.M2). The most complex module of the system is the meta-classifier that combines the ten methods. Generally, the modules of each level are controlled by those of the immediately upper level and control those of the lower level. This hierarchical organization allows simplicity of operation and easiness of expansion to use more methods or classifier types.

4.2 Automatic Adaptation to the Problem

A very attractive feature of the system is its ability to adapt itself to the problem for which it is created. This means that some parameters are chosen automatically, according to the performance on a validation set. For tuning the values of these parameters, the system uses half of the patterns of its training set as a validation set. Each classifier and each method are validated by selecting different values for their parameters. At the end, the set of parameters giving the best performance is selected. The range of parameter values that are going to be tested is properly predefined so as to cover most cases. After selection of parameter values, the system is supposed to have adapted itself to the problem and it is ready to be trained. Due to automatic adaptation, the system does not need an expert's opinion to tune it before putting it to work. So, a doctor can use the system without necessitating technical knowledge and is able to create anytime a new system for a new diagnosis problem.

Specifically, for the simple classifiers, the parameters concern their structure or their training algorithms. We chose to have only one parameter undefined for each type of classifier. For Multilayered Feed-forward Neural Networks it is their training epochs, for Support Vector Machines with polynomial kernel it is the degree of the polynomial, for Support Vector Machines with RBF kernel it is the dispersion of the exponential. The exception is the C4.5 Decision Trees that are completely defined irrelevantly of the problem. For the methods, there is only one parameter to tune and this is the number of sub-classifiers combined by the method. Originally, values that lead to small training times are selected. After the initialization, the values are gradually increased. The number of trials allowed is limited. Once this number is reached, no more trials are performed and the best values until then (with the lowest validation error) are kept.

Now, for the combination methods, we start with 2 classifiers in the ensemble (the lowest possible) and on each trial we increase the population by 1 until a maximum of

30 (according to experimental results reported in [3], 20 to 25 classifiers are usually enough) classifiers is reached.

4.3 Parallel Operation

The system has been constructed in such a way so as to have the ability of parallel operation from the upper to the lower level of hierarchy. In a parallel environment, all methods are trained simultaneously and independently, so the training time will be that of the slowest method. The same is the case when the system classifies new patterns, after training is completed. The third level of hierarchy (simple classifiers) also supports parallelism. Each classifier used by a method works independently of the others. The only exception is in the case of the AdaBoost.M2 method, where the training of the classifiers must be done in a serial way. With parallel organization we have significant reduction in time complexity, particularly in training, which is the most time-consuming phase. The trade-off for this gain is the increased computational resources needed. The implementation of this operation has been done using Java threads. However, the experiments that we present later were held on a single-processor system that does not take advantage of the parallelization abilities. On a multi-processor system the time needed would be severely reduced.

5 Experimental Results

We evaluated the system on three well-known benchmark medical problems from the UCI data repository [20], namely diabetes, breast-cancer and new-thyroid. Also, we tested the system on a problem concerning the classification of hepatic lesions from computed tomography (CT) images. The goal of this experimental study is to discover whether the meta-classifier approach exhibits better performance than the best single combination method or, at least, if we can use it in order to avoid searching for the best method for a particular problem. The comparisons are based on the test data available for each problem and the evaluation of the performance concerns the errors made in the classification of the test data.

More specifically, for each of the four problems, we create and train ten different meta-classifier systems. In the beginning of the training, the data sets are shuffled and divided into two parts. The first two thirds will compose the training set while the remaining one third will form the test set. Due to the shuffling originally done, these sets are different for each trial. At the end of training, the test error is extracted for each method used. As described in Section 3, four different combination schemes are considered and at each trial the test error is estimated for each scheme. By that, at the end, representative average error rates are formed, so as to compare not only the performance of the methods but the different combination schemes as well. Also, as described in Section 4, each meta-classifier system combines ten different combination methods.

In the next sub-sections, we briefly describe each problem and present experimental results. In the tables, we are using the numbering (1, 2, …,10) to denote the methods as follows: 1-Bagging with MLPs, 2-Bagging with SVMs having polynomial kernels, 3-Bagging with SVMs having polynomial kernels, 4-Bagging with C4.5

Decision Trees, 5-AdaBoost.M2 with MLPs, 6-AdaBoost.M2with C4.5 Decision Trees, 7-FuzzMCS with MLPs, 8-FuzzMCS with SVMs having RBF kernels, 9-FuzzMCS with SVMs having polynomial kernels, 10-FuzzMCS with C4.5 Decision Trees. Similarly, the combination schemes will be denoted by the letters (A, B, C, D) as follows: A-Crisp voting, B-Weighted voting, C-Average class probabilities, D-Class probabilities weighted sum.

5.1 The Diabetes Problem

The first benchmark problem concerns diabetes diagnosis in female members of the Pima Indian tribe of America. The data of the problem consists of 768 different patterns. Each of them has 8 arithmetic features (there are no missing values) and one class label, diabetic or not. Out of the 768 patterns, the 500 are for not diabetic behavior. The two thirds of the original data will compose the training set and the rest will be the testing data. The original data set is shuffled for each of the ten trials and the error rate is computed. Table 1 presents the results (mean, min and max values) for each of the ten methods combined and for each of the four combination schemes. The numbers presented are per cent rates.

Table 1. Results for the diabetes problem

	1	2	3	4	5	6	7	8	9	10	A	B	C	D
Min	21.9	19.9	21.1	20.7	20.7	20.3	26.6	22.3	22.3	23.8	18.8	20.3	19.5	19.5
Max	28.9	27.7	27.7	27.7	28.9	28.9	33.2	32	32	34	27.3	27.3	27.3	27.3
Mean	25.4	23.7	24	24.3	24.7	25.5	30.1	25.6	27	27.6	23.6	23.7	24.2	24

We first observe that Bagging with SVMs having RBF kernel has the best performance with an error rate of 23.7%. On the other hand, the best combination scheme for this problem is crisp voting with 23.6% error rate. We can observe that, in this case, the combination of the methods (meta-classifier) performs better than the best method by 0.1%.

5.2 The Breast-Cancer Problem

The breast-cancer problem is about the diagnosis of malignance of breast tumors. The data come from the University of Wisconsin. There are in total 699 patterns, each having 10 integer features (values between 1 and 10) and a class label (malignant or benign tumor). Out of 699 patterns, 458 are benign whereas the remaining 241 are malignant. The trials are performed in the same way as for the previous problem and the results are presented in Table 2.

Table 2. Results fort he breast-cancer problem

	1	2	3	4	5	6	7	8	9	10	A	B	C	D
Min	3	3	3	3	3	3	3.4	3	3.4	3	2.6	2.6	2.6	2.6
Max	5.6	5.2	6.4	6.9	5.2	7.7	6.9	5.2	6.9	6.4	5.6	5.2	4.7	4.7
Mean	4.1	4.1	4.7	4.3	4	4.6	4.8	4	5.2	4.6	3.6	3.6	3.5	3.5

We can observe that, in this case, the best methods are AdaBoost.M2 with MLPs and FuzzMCS with SVMs having RBF kernel, each yielding 4% average error rate. The best combination schemes are those that use class probabilities with 3.5% error rates. This means that the performance of the best combination scheme is better by 0.5% than the best method. This is a considerable improvement since the error rates for this problem are generally small and it is difficult to decrease them significantly.

5.3 The New-Thyroid Problem

The third benchmark problem used to evaluate the meta-classifier is the so called new-thyroid. This problem concerns the characterization of the functionality of the thyroid under three possible states: normal, hypothyroid and hyperthyroid. The data set includes 215 patterns each of which has 5 continuous arithmetic features. Out of them, 150 are normal, 35 are hyperthyroid and 30 are hypothyroid. The results of the ten trials are shown in Table 3.

Table 3. Results for the new-thyroid problem

	1	2	3	4	5	6	7	8	9	10	A	B	C	D
Min	1.4	5.6	6.9	1.4	1.4	2.8	0	0	0	0	0	0	0	0
Max	5.6	16.7	13.9	15.3	16.7	12.5	4.2	4.2	4.2	4.2	4.2	4.2	4.2	4.2
Mean	3.2	10.4	10.3	7.7	5.1	7.6	2.1	1.7	2.1	1.4	1.5	1.6	2.5	2.5

The FuzzMCS method with C4.5 Decision Trees is the best for this problem with an 1.4% error rate. The best combination technique is crisp voting with 1.5% average error rate. In this case, the meta-classifier approach is slightly worse than the best method but achieves performance very close to that.

5.4 Classification of Hepatic Lesions from Computed Tomography (CT) Images

Apart from the three benchmark problems on which we have tested the system so far, another problem concerning classification of hepatic lesions is used to evaluate the meta-classifier. The data for this problem come from Computed Tomography (CT) images, acquired at the Second Department of Radiology, Medical School, University of Athens [11] and they are not widely available with responsibility of the source. A total number of 147 images were acquired corresponding to 147 different patients. Out of them, 76 are healthy, 19 have cysts, 28 hemangiomas and 24 hepatocellular carcinomas. So, it is a problem with four classes and 147 different patterns. Each pattern has originally 89 features, but, by using genetic algorithms for dimensionality reduction (the procedure for this is described in [11]), 12 features are selected and used. The results of the experiment are presented in Table 4.

Table 4. Results for the problem of classification of hepatic lesions

	1	2	3	4	5	6	7	8	9	10	A	B	C	D
Min	22.4	22.4	12.2	22.4	22.4	18.3	16.3	22.4	22.4	22.4	18.3	16.3	18.3	16.3
Max	32.6	48.9	38.7	42.8	38.7	40.8	38.7	42.8	40.8	46.9	34.6	34.6	38.7	36.7
Mean	27.3	35.5	28.7	31.4	30.4	30.6	26.7	32.2	30.2	33.2	25.9	24.0	25.7	24.9

The best method for this problem is FuzzMCS with MLPs as classifiers yielding 26.7% error rate. As long as the combinations are concerned, the best one is weighted voting having 24% error rate. This performance is significantly better than that of the best method, pointing that the use of the meta-classifier is beneficial to the classification. Moreover, we observe that every combination scheme is better than the best method, which means that whatever is our combination choice, the performance of the meta-classifier will be high. We underline that, in general, the performance of the classification for this problem can be better if we use all 89 features or at least a more representative subset than the 12 finally used. However, having in mind that the comparison was our objective in this experiment, the use of 12 features was considered adequate.

6 Conclusions

In this work, a new methodology has been developed which combines several different combination methods, in analogy to the combination of simple classifiers by these methods, in an attempt to get better performance results than the best individual method. The aim of this meta-classifier approach is to combine combination methods in an efficient way improving performance and to avoid the selection of the best combination method - as we do not know in advance which the best one is. The latter involves time-consuming experimentation and depends on the complexity of the problem.

The proposed meta-classifier approach was implemented in a medical diagnosis system and evaluated on three benchmark diagnosis problems and a problem concerning the classification of hepatic lesions from computed tomography (CT) images. The first conclusion is that on average, the best combining method out of the four tested for the combination of the methods in the meta-classifier is the second in turn, the weighted voting. Despite the fact that it is outperformed by the first method (crisp voting) in the diabetes and the new-thyroid problem by 0.1%, it is considerably better in the hepatic lesions problem. These two methods are slightly worse than crisp and weighted averaging only in the breast cancer problem, a fact indicating that voting performs better than averaging. Generally, however, the best combination method depends each time on the particular classification problem. Comparing the performance of the weighted voting with that of the best method each time, in the diabetes problem the error rates are equal, in the breast cancer problem there is an enhancement of 0.4%, in the new-thyroid problem the combination is worse by 0.2% and in the hepatic lesions problem a significant improvement of 2.7% is observed. So, the main conclusion is that the combination of the combination methods enhances performance. In some data sets, the test error rate is on average lower than that of the best individual method used. When this is not the case, the combination exhibits performance analogous to that of the best method. Practically, this implies that -in the worst case- the combination of combination methods has almost the same performance as the best method. This allows us to avoid the search for the best method and directly use the meta-classifier method expecting to obtain the best performance. Ultimately, the system is a medical diagnosis aiding tool which provides to the doctor a suggestion-opinion along with a degree of reliability.

As for the future work that can be done, first of all we can test the system on a multi-processing environment, which is expected to severely reduce the time needed for training. Also, we can try to expand the range of types of simple classifiers used in the lower level of the system (for example we can use RBF Neural Networks). The same can be done for combining methods (for example we can use Mixture-of-Experts approaches). Another issue that might be possible to study would be the effect of the combination through a gating network properly trained instead of the voting or averaging combination methods used so far.

References

1. Alpaydin, E.: Multiple networks for function learning. Proceedings of the 1993 IEEE International Conference on Neural Networks, vol. I, pp. 27-32,1993.
2. Alpaydin, E.: Techniques for combining multiple learners. Proceedings of Engineering of Intelligent Systems, vol. 2, ICSC Press, pp. 6-12, 1998.
3. Bauer, E., Kohavi, R.: An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting and Variants. Machine Learning, Vol. 36, pp. 105-139, 1999.
4. Breiman, L.: Bagging Predictors. Technical report 421. Department of Statistics, University of California, Berkeley, 1994.
5. Breiman, L., Friedman, J., Olshen, R., Stone, C.: Classification and Regression Trees. Chapman and Hall, New York, 1994.
6. Drucker, H.: Boosting using Neural Networks. Springer-Verlag, 1998.
7. Drucker, H., Cortes, C., Jackel, L., LeCun, Y., Vapnik, V.: Boosting and other machine learning algorithms. Proceedings on the Eleventh International Conference on Machine Learning, pp. 53-61, New Brunswick, NJ, 1994.
8. Freund, Y.: Boosting a weak learning algorithm by majority. Information and Computation 121, vol. 2, pp. 256-285, 1996.
9. Freund, Y., Schapire, R.E.: Experiments with a new boosting algorithm. Proceedings on the Thirteenth International Conference on Machine Learning, pages 148-156, 1996
10. Frosyniotis, D., Stafylopatis, A., Likas, A. : A divide-and-conquer method for multi-net classifiers. Pattern Analysis and Applications, Vol. 6, pp. 32-40. Springer-Verlag. 2003
11. Gletsos, M., Mougiakakou, S.G., Matsopoulos, G., Nikita, K.S., Nikita, A.S.: A Computer-Aided Diagnostic System to Characterize CT Focal Liver Lesions: Design and Optimization of a Neural Network Classifier. IEEE Transactions on Information Technology in Biomedicine, Vol. 7, No. 3, September 2003.
12. Hansen, L., Salamon, P.: Neural Network Enembles. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, pp. 993-1001, 1990.
13. Hu, Y.H., Palreddy, S., Tompkins, W.J.: A Patient-Adaptable ECG Beat Classifier Using a Mixture-of-Experts Approach. IEEE Transactions on Biomedical Engineering, Vol. 44, No. 9, September 1997.
14. Maclin, R., Opitz, D.: Popular Ensemble Methods: An empirical study. Journal of Artificial Intelligence Research 11, 169-198, 1999.
15. Platt, J.: Fast Training of Support Vector Machines using Sequential Minimal Optimization. Advances in Kernel Methods - Support Vector Learning, MIT Press, 1998.
16. Quinlan, J.R. : Bagging, Boosting and C4.5. Proceedings on the Thirteenth National Conference on Artificial Intelligence. AAAI Press and the MIT Press, 725-730. 1996
17. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo, California, 1993.
18. Rumelhart, D.E., Hinton,G.E., Williams, R.J.: Learning representations of backpropagation errors. Nature (London), vol. 323, pp. 533-536, 1986.

19. Sharkey, A.J.C.: Combining Artificial Neural Nets: Ensemble and Modular Multi-Net Systems. Springer Press, 1999.
20. UCI Machine Learning Databases Repository, University of California-Irvine, Department of Information and Computer Science. [ftp://ftp.ics.edu/pub/machine-learning-databases]
21. Vapnik, V.N.: Principles of risk minimization for learning theory. Advances in Neural Information Processing Systems, vol. 4, pp. 831-838, San Mateo, CA, Morgan Kaufmann, 1992.
22. Vapnik,V.N.: The Nature of Statistical Learning Theory. Wiley, New York, 1998.
23. Zhou, Z.H., Jiang, Y.: Medical Diagnosis With C4.5 Rule Preceded by Artificial Neural Network Ensemble. IEEE Transactions on Biomedical Engineering, Vol. 7, No. 1, March 2003.