A HYBRID DECISION SUPPORT TOOL
Using ensemble of classifiers

S. B. Kotsiantis, P. E. Pintelas
Educational Software Development Laboratory, Department of Mathematics, University of Patras, Greece
Email: sotos@math.upatras.gr, pintelas@math.upatras.gr

Keywords: decision support systems, artificial intelligence

Abstract: In decision support systems a classification problem can be solved by employing one of several methods such as different types of artificial neural networks, decision trees, bayesian classifiers, etc. However, it may happen that certain parts of instances' space are better predicting by one method than the others. Thus, the decision of which particular method to choose is a complicated problem. A good alternative to choosing only one method is to create a hybrid forecasting system incorporating a number of possible solution methods as components (an ensemble of classifiers). For this purpose, we have implemented a hybrid decision support system that combines a neural net, a decision tree and a bayesian algorithm using a stacking variant methodology. The presented system can be trained with any data, but in the current implementation is mainly used by tutors of Hellenic Open University to identify drop-out prone students. However, a comparison with other ensembles using the same classifiers as base learner on several standard benchmark datasets showed that this tool gives better accuracy in most cases.

1 INTRODUCTION

Recently in the area of decision support systems the concept of combining classifiers is proposed as a new direction for the improvement of the performance (Turban and Aronson, 1998). An ensemble of classifiers is a set of classifiers whose individual decisions are combined in some way to classify new instances. Numerous methods have been suggested for the creation of ensemble of classifiers (Dietterich, 2001). Mechanisms that are used to build ensemble of classifiers include: i) Using different subsets of training data with a single learning method, ii) Using different training parameters with a single training method, iii) Using different learning methods.

An accessible and informal reasoning, from statistical, computational and representational viewpoints, of why ensembles can improve results is presented in (Dietterich, 2001). The main reason may be that the training data can not provide sufficient information for choosing a single best classifier from the set of hypotheses, because the amount of training data available is too small compared to the size of the hypothesis space.

Although, many methods of ensemble creation have been proposed, there is as yet no clear picture of which method is best. This is in part because only a limited number of comparisons have been attempted and several of those have concentrated on comparing boosting to bagging (Breiman, 1996), (Schapire et al., 1998), (Bauer and Kohavi, 1999).

In this work, we have implemented a hybrid decision support system that combines three different learning methods: the Naive Bayes, the C4.5 and the BP algorithm using a stacking variant methodology. We performed a comparison with other ensembles on several standard benchmark data sets and we took better accuracy in most cases.

Section 2 presents the most well-known methods for building ensembles of classifiers, while section 3 discusses the proposed ensemble method. Experiment results and comparisons of the proposed combining method with other ensembles in several data sets are presented in section 4. We briefly present the implemented hybrid decision support tool in Section 5. Finally, we conclude in Section 6 with summary and further research topics.

2 ENSEMBLES OF CLASSIFIERS

As we have already mentioned the concept of combining classifiers is proposed as a new direction for the improvement of the performance. The goal of

classification result integration algorithms is to generate more certain, precise and accurate system results. This section provides a brief survey of methods for constructing ensembles.

Bagging is a method for building ensembles that uses different subsets of training data with a single learning method (Breiman, 1996). Given a training set of size *t*, it draws *t* random instances from the data set with replacement (i.e. using a uniform distribution), these *t* instances are learned, and this process is repeated several times. Since the draw is with replacement, usually the instances drawn will contain some duplicates and some omissions as compared to the original training set. Each cycle through the process results in one classifier. After the construction of several classifiers, taking a vote of the predictions of each classifier gives the final prediction.

Another method that uses different subsets of training data with a single learning method is the boosting approach (Freund and Schapire, 1996). Boosting is similar in overall structure to bagging, except that keeps track of the performance of the learning algorithm and concentrates on instances that have not been correctly learned. Instead of choosing the *t* training instances randomly using a uniform distribution, it chooses the training instances in such a manner as to favour the instances that have not been accurately learned. After several cycles, the prediction is performed by taking a weighted vote of the predictions of each classifier, with the weights being proportional to each classifier's accuracy on its training set.

It has been observed that for both bagging and boosting, an increase in committee size usually leads to a decrease in prediction error, but the relative impact of each successive addition to a committee is ever diminishing. For both bagging and boosting, much of the reduction in error appears to have occurred after ten to fifteen classifiers. But boosting continues to measurably improve their test-set error until around 25 classifiers for decision trees (Optiz and Maclin, 1999).

Another approach for building ensembles is to use a variety of learning algorithms on all of the training data and combine their predictions according to a voting scheme. The intuition is that the models generated using different learning biases are more likely to make errors in different ways. Among the combination methods, majority vote is the simplest to implement, since it requires no prior training (Ji and Ma, 1997).

Another method for combining classifiers called grading and learns a meta-level classifier for each base-level classifier (Seewald and Furnkranz, 2001). The meta-level classifier predicts whether the base-level classifier is to be trusted (i.e., whether its prediction will be correct). The base-level attributes are used also as meta-level attributes, while the meta-level class values are + (correct) and − (incorrect). Only the base-level classifiers that are predicted to be correct are taken and their predictions combined by summing up the probability distributions predicted.

Stacked generalization (Wolpert, 1992), or stacking, is another approach that uses a variety of learning algorithms. Stacking combines multiple classifiers to induce a higher-level classifier with improved performance. A learning algorithm is used to determine how the outputs of the base classifiers should be combined. The original data set constitutes the level zero data and all the base classifiers run at this level. The level one data are the outputs of the base classifiers and another learning process occurs using as input the level one data and as output the final prediction. A straightforward extension proposed by (Ting and Witten, 1999) is to use class probability distributions instead of predictions. This allows each base classifier to express uncertainty by returning estimated probabilities for all classes instead of just the one predicted class. Multi-response linear regression (MLR) was used for meta-level learning (Ting and Witten, 1999). Other researchers used model tree induction instead of MLR keeping everything else the same for better results (Dzeroski and Zenko, 2002). Recently, other authors modified the method so as to use only the class probabilities associated with the true class (Seewald, 2002) and the accuracy seems to be improved.

3 PROPOSED METHODOLOGY

As we have already mentioned we combine the Naive Bayes, the C4.5 algorithms and BP algorithm using a stacking variant methodology. Its basic idea may be derived as a generalization of voting as follows. Let us consider the voting step as a separate classification problem, whose input is the vector of the responses of the base classifiers. Simple voting uses a predetermined algorithm for this, namely to count the number of predictions for each class in the input and to predict the most frequently predicted class. Stacking replaces this with a trainable classifier. This is possible, since for the training set, we have both the predictions of the base learners and the true class. The matrix containing the predictions of the base learners as predictors and the true class for each training case will be called the meta-data set. The classifier trained on this matrix will be called the meta-classifier or the classifier at the meta-level. While stacking (Ting and Witten, 1999)

uses all class probabilities for all models, our method uses only the class probabilities associated with the true class. The dimensionality of the meta-data set is reduced by a factor equal to the number of classes, which leads to faster learning. However, this modification cannot change the accuracy for the two class problems since the probability of one class is one minus the probability of the other class. Concerning the choice of the algorithm for learning at the meta-level, we have explored the use of model trees instead of MLR (Seewald, 2002) since model trees naturally extend MLR to construct piecewise linear approximations. Model trees have the same structure as decision trees, with one difference: they employ a linear regression function at each leaf node to make a prediction. The most well known model tree inducer - M5′ (Wang and Witten, 1997) – is used by our system.

In the following, we briefly refer to the learning algorithms that are used as base learners. The most commonly used C4.5 algorithm (Quinlan, 1993) is the representative of the decision trees in our system. Naive Bayes (NB) classifier, which is used in our system, is the simplest form of Bayesian networks (Jensen, 1996). Finally, the most well-known neural network learning algorithm - Back Propagation (BP) (Mitchell, 1997) - is used in our system.

4 COMPARISONS AND RESULTS

For the purpose of our study, we used 22 well-known data set by many domains from the UCI repository (Blake and Merz, 1998). These data sets were hand selected so as to come from real-world problems and to vary in characteristics. In order to calculate the classifiers' accuracy, the whole training set was divided into ten mutually exclusive and equal-sized subsets and for each subset the classifier was trained on the union of all other subsets. Then, cross validation was run 10 times for each algorithm and the median value of the 10-cross validations was calculated.

It must be mentioned that we used the free available source code for these algorithms by (Witten and Frank, 2000) for our experiments. We have tried to minimize the effect of any expert bias by not attempting to tune any of the algorithms to the specific data set. Wherever possible, default values of learning parameters were used. This naïve approach results in lower estimates of the true error rate, but it is a bias that affects all the learning algorithms equally.

To start with, we empirically compare the proposed stacking ensemble with the plain classifier NB, C4.5, BP as well as their bagging and boosting versions with 25 sub-classifiers. In the following Tables, win (v) indicates that the specific method (in the column) performed statistically better than the proposed stacking method. It must be mentioned that the resulting differences between algorithms were assumed statistically significant when $p<0.01$ because p-value less than 0.05 is not strict enough, if many classifiers are compared in numerous data sets (Salzberg, 1997). Loss(*) indicates that the stacking process performed statistically better than the specific method (in the column). In all the other cases, there is no significant statistical difference between the results. In the following Tables, we also present the average accuracy of all tested data set for each classifier and ensemble.

In the last raw of the Table 1 one can see the aggregated results. The proposed ensemble (Stacking′) is significantly more accurate than Naïve Bayes (NB), C4.5 and BP in eight, four and seven out of the 22 data sets respectively, while it is significantly less accurate in none data set. The proposed ensemble is significantly more accurate than bagging C4.5, bagging NB and bagging BP in two, eight and seven out of the 22 data sets respectively, while it has significantly higher error rates than theses ensembles in none data set.

The proposed ensemble (Stacking′) is significantly more accurate than boosting C4.5 and boosting NB in three and five out of the 22 data sets respectively, whilst boosting C4.5 is significantly more accurate in one data set and boosting NB in none data set. Moreover, the proposed ensemble is significantly more accurate than boosting BP in seven out of the 22 data sets respectively, whilst it is significantly less accurate in one data set.

To sum up, the performance of the proposed ensemble is more accurate than the other well-known ensembles that use only one of the C4.5 or NB or BP algorithms. The average relative accuracy improvement starts with 2% in relation to boosting C4.5 and exceeds 7% in relation to boosting BP.

Subsequently, we compare the proposed stacking methodology (Stacking′) with:

- The methodology of selecting the best classifier of the NB, C4.5 and BP according to 10-cross validation (BestCV) (Schaffer, 1993).
- Grading methodology using the instance based classifier IBk with ten nearest neighbors as the meta level classifier (Seewald and Furnkranz, 2001) and NB, C4.5 and BP as base classifiers.
- Voting methodology using NB, C4.5 and BP as base classifiers (Ji and Ma, 1997)
- Stacking methodology that constructs the meta-data set by adding the entire pre-dicted class probability distribution instead of only the most likely class. We used NB, C4.5 and BP as base classifiers and both MLR (Ting and Witten,

1999) as well as model tree induction (Dietterich, 2000) as meta level classifier.

- Stacking using only the class probabilities associated with the true class as meta-data set and MLR as meta level classifier (Seewald, 2002).

Table 1: Comparing Stacking´ with the plain classifier NB, C4.5, BP as well as their bagging versions

	Stacking´	C4.5	NB	BP	Bagging C4.5	Bagging NB	Bagging BP	Boost C4.5	Boost NB	Boost BP
anneal	98.79	98.57	86.59*	92.78*	98.83	86.94*	93.52*	99.60	95.20*	92.81*
audiology	74.92	77.26	72.64	43.82*	81.29	72.10	46.50*	84.62v	78.20	43.74*
autos	80.95	81.77	57.41*	54.38*	83.85	57.15*	56.39*	86.05	57.12*	56.36*
balance	90.37	77.82*	90.53	87.09*	82.33*	90.29	86.18*	76.91*	92.11	87.19*
breast-w	96.12	95.01	96.07	95.97	96.31	96.07	96.44	96.51	95.55	95.90
colic	84.72	85.16	78.70*	82.58	85.23	78.94*	83.75	82.01	77.46*	79.44*
credit-g	74.77	71.25*	75.16	72.75	74.17	75.13	74.97	72.79	75.09	73.96
diabetes	76.41	74.49	75.75	76.56	75.67	75.57	76.84	72.81*	75.88	76.5
haberman	73.20	71.05	75.06	74.64	72.06	74.86	75.69	71.12	73.94	74.51
heart-c	83.08	76.94*	83.34	81.39	79.54	83.24	82.81	79.60	83.14	80.99
heart-h	83.75	80.22	83.95	81.37	79.91	84.16	83.27	78.28	84.67	81.30
heart-statlg	83.48	78.15	83.59	82.11	81.11	83.41	83.22	80.15	82.30	81.07
hepatitis	83.09	79.22	83.81	81.30	81.63	84.39	84.25	82.74	84.23	82.64
ionosphere	91.31	89.74	82.17*	85.84*	92.23	81.94*	87.09	93.62	91.12	89.09
iris	95.67	94.73	95.53	96.27	94.80	95.53	96.67	94.47	95.07	95.47
lymp/rapy	81.15	75.84	83.13	80.24	79.14	83.50	82.75	83.09	80.67	80.70
monk2	80.02	57.75 *	56.83 *	75.7	61.15 *	56.49 *	69.03 *	61.86 *	56.83 *	99.88v
mushroom	100.0	100.0	95.76*	100.0	100.0	95.56*	100.0	100.0	100.0	100.0
vehicle	74.09	72.28	44.68*	50.07*	74.91	45.73*	55.03*	77.16	44.68*	50.32*
sonar	76.90	73.61	67.71	73.30	79.03	67.96	76.98	83.03	81.21	79.77
vote	96.64	96.57	90.02*	94.60	96.53	90.09*	95.17	95.24	95.19	95.45
zoo	97.13	92.61	94.97	61.21*	93.29	95.07	72.97*	95.38	97.23	61.21*
Av. Accac.	85.30	81.82	79.70	78.36	83.77	79.73	79.98	83.96	81.68	79.92
W/D/L		0/18/4	0/14/8	0/15/7	0/20/2	0/14/8	0/15/7	1/18/3	0/17/5	1/14/7

In the last raw of the Table 2 one can see the aggregated results. The proposed ensemble (Stacking´) is significantly more accurate than stacking with MLR procedure, grading, stacking with model tree, stacking using only the class probabilities associated with the true class as meta-data set with MLR procedure and BestCV in one out of the 22 data sets, while it is significantly less accurate in none data set. Similarly, the proposed ensemble is significantly more accurate than voting in two out of the 22 data sets, while it is significantly less accurate in none data set. The average relative accuracy improvement of the proposed stacking methodology is about 1%-1.5% in relation to the remaining methods.

As a conclusion, our approach performs slightly better than existing ensembles. It is not a surprise that stacking with multi-response model trees performs better than stacking with multi response linear regression. The results of (Frank et al., 1998) who investigated classification via regression, it was showed that classification via model trees per-forms extremely well, i.e., better than multi response linear regression and better than C5.0 (a successor of C4.5 (Quinlan, 1993), especially in domains with continuous attributes. This indicates that multi response model trees are a very suitable choice for learning at the meta-level, as confirmed by our experimental results.

5 IMPLEMENTED SYSTEM

The primary scope of the presented decision support tool was to automatically identify dropout-prone students in university level distance learning using students' key demographic characteristics and their marks in a few written assignments as training set (Kotsiantis et al., 2003). While the tutors still have an essential role in monitoring and evaluating student progress, the tool can compile the data required for reasonable and efficient monitoring.

Table 2. Comparing Ensembles

	Stacking´	BestCV	Voting	Grading	Stacking with MLR	StackingC with MLR	Stacking with model trees
anneal	98.79	98.57	98.92	98.90	98.53	98.56	98.85
audiology	74.92	77.26	74.93	74.31	72.83	75.55	72.41
autos	80.95	81.77	78.11	78.04	82.40	81.67	80.37
balance	90.37	90.53	89.87	90.08	87.11*	87.07*	93.29
breast-w	96.12	95.84	96.45	96.55	96.20	96.18	96.10
colic	84.72	84.99	84.18	84.54	84.56	84.56	84.69
credit-g	74.77	75.11	74.75	74.52	74.76	74.80	74.74
diabetes	76.41	75.74	76.77	75.94	76.63	76.67	76.41
haberman	73.20	73.99	74.70	73.70	72.55	73.00	73.20
heart-c	83.08	82.85	82.57	82.75	83.21	83.34	83.08
heart-h	83.75	83.95	83.24	83.31	83.95	83.85	83.82
heart-statlog	83.48	83.44	83.07	82.93	84.22	84.33	83.52
hepatitis	83.09	82.63	82.97	82.64	83.16	83.24	83.09
ionosphere	91.31	89.77	92.25	92.05	91.31	91.31	90.92
iris	95.67	94.93	96.00	95.93	95.13	95.07	95.73
lymphotherapy	81.15	81.56	82.32	82.40	80.63	80.81	80.15
monk2	80.02	75.7	64.13 *	72.88	79.03	80.02	79.33
mushroom	100.0	100.0	100.0	100.0	100.0	100.0	100.0
vehicle	74.09	72.28	72.72	72.25*	74.23	74.20	71.05*
sonar	76.90	73.38	78.71	78.71	76.71	76.71	76.71
vote	96.64	96.57	96.18	95.91	96.57	96.57	96.52
zoo	97.13	93.59*	94.78*	96.15	96.05	96.25	96.83
Aver. Accuracy	85.30	84.75	84.44	84.75	84.99	85.17	85.04
W/D/L		0/21/1	0/20/2	0/21/1	0/21/1	0/21/1	0/21/1

However, the application of the tool is not restricted to predict drop-out prone student, it can also enable users to explore any data and build the proposed model for forecasting and classification.

The tool expects the training set as a spreadsheet in CSV (Comma-Separated Value) file format. The tool assumes that the first row of the CSV file is used for the names of the attributes. There is not any restriction in attributes' order. However, the class attribute must be in the last column.

Once the database is in a single relation, each attribute is automatically examined to determine its data type (for example, whether it contains numeric or symbolic information). A feature must have the value ? to indicate that no measurement was recorded. A problem can be created by a field in the database that is of type integer but whose contents are not used arithmetically. Changing the field to one where the numbers are treated as nominal values will eliminate the possibility of the system creating inappropriate rules.

After opening the data set that characterizes the problem for which the user wants to take the prediction, the tool automatically uses the corresponding attributes for training the proposed ensemble algorithm. The tool is available in the web page: http://www.math.upatras.gr/~esdlab/Desicion-Support-Tool/

After the training of the model (this takes some time to complete, from few seconds to few minutes), the user is able to see the produced ensemble. The tool can also predict the class of either a single instance or an entire set of instances (batch of instances). It must be mentioned that for batch of instances the user must import an Excel cvs file with all the instances he/she wants to have predictions.

Moreover, the implemented tool can present useful information about the imported data set such as the presence or not of missing attribute values, the frequency of each attribute value etc. Finally, the tool provides on-line help for novice users.

6 CONCLUSION

Hybrid decision support systems have shown to be effective in many applicative domains and can be considered as one of the main current directions in decision support research. If we are concerned for the best possible classification accuracy, it might be difficult or impossible to find a single classifier that performs as well as a good ensemble of classifiers. When designing an ensemble with stacking methodology, one may choose from a set of available classifiers those whose combination will derive the best over-all stacked classifier. In this

study, we present a stacking variant methodology that uses three different learning methods: the Naive Bayes, the C4.5 and the BP algorithms as base classifiers and M5′ as meta level classifier. A number of comparisons with other ensembles that use the C4.5 or NB or BP as base classifiers showed that this method gives better accuracy in many cases.

In spite of these results, no general method will work always. Therefore, we can only state that a particular method for creating an ensemble can be better than the best single model and continue to work on identifying the generation and combination methods that can best solve different classification problems.

The stacked generalization architecture for classifier combination has still many open questions. For example, there are currently no strict rules saying which base classifiers should be used and what features of the training set should be used to train the combining classifier.

In a future work, we will use a feature selection pre-process before the usage of the stacking. Feature subset selection is the process of identifying and removing as much irrelevant and redundant features as possible. This will reduce the dimensionality of the data enabling the proposed ensemble to operate faster and maybe more effectively.

REFERENCES

Bauer, E., Kohavi, R, 1999. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning* 36, 105–139.

Blake, C.L., Merz, C.J, 1998. UCI Repository of machine learning databases. Irvine, CA: University of California, Department of Information and Computer Science. (www.ics.uci.edu/~mlearn/MLRepository.html)

Breiman, L, 1996. Bagging Predictors. *Machine Learning* 24, 123-140.

Dietterich, T.G, 2001. Ensemble methods in machine learning. In Kittler, J., Roli, F., eds. Multiple Classifier Systems. *LNCS Vol. 1857*, Springer, 1–15.

Dzeroski, S., Zenko, B., 2002. Is Combining Classifiers Better than Selecting the Best One. *ICML 2002*: 123-130.

Frank, E., Wang., Y., Inglis, S., Holmes, G., & Witten, I. H., 1998. Using model trees for classification. *Machine Learning* 32, 63-76.

Freund, Y., Schapire, R., 1996. Experiments with a New Boosting Algorithm, Proceedings: *ICML'96*, p. 148-156.

Jensen, F., 1996. An Introduction to Bayesian Networks. Springer.

Ji, C., Ma, S., 1997. Combinations of weak classifiers. *IEEE Transaction on Neural Networks* 8, 32–42.

Kotsiantis, S., Pierrakeas, C., Pintelas, P., 2003. Preventing student dropout in distance learning systems using machine learning techniques, Proceedings of Seventh International Conference on Knowledge-Based Intelligent Information & Engineering Systems, Lecture Notes in Artificial Intelligence, Vol. 2774, Springer-Verlag, 267-274.

Mitchell, T., 1997. *Machine Learning*. McGraw Hill.

Opitz, D., Maclin, R., 1999. Popular Ensemble Methods: An Empirical Study, *Artificial Intelligence Research* 11, 169-198, Morgan Kaufmann.

Quinlan, J.R., 1993. *C4.5: Programs for machine learning*. Morgan Kaufmann, San Francisco.

Salzberg, S., 1997. On Comparing Classifiers: Pitfalls to Avoid and a Recommended Approach, *Data Mining and Knowledge Discovery* 1, 317–328.

Schaffer, C., 1993. Selecting a classification method by cross-validation. *Machine Learning* 13, 135-143.

Schapire, R. E., Freund, Y., Bartlett, P., & Lee, W. S., 1998, Boosting the margin: A new explana-tion for the effectiveness of voting methods. *The Annals of Statistics* 26, 1651–1686.

Seewald, A. K., Furnkranz, J., 2001. An evaluation of grading classifiers. In Advances in Intelligent Data Analysis: *Proceedings of the Fourth International Symposium (IDA-01)*, pages 221–232, Berlin, Springer.

Seewald, A.K, 2002. How to Make Stacking Better and Faster While Also Taking Care of an Unknown Weakness, in Sammut C., Hoffmann A. (eds.), *Proceedings of the Nineteenth International Conference on Machine Learning (ICML 2002)*, Morgan Kaufmann Publishers, pp.554-561.

Ting, K., & Witten, I., 1999. Issues in Stacked Generalization, *Artificial Intelligence Research* 10, 271-289, Morgan Kaufmann.

Turban, E., Aronson, J., 1998. *Decision Support Systems and Intelligent Systems*, Prentice Hall.

Wang, Y., Witten, I., 1997, Induction of model trees for predicting continuous classes, In *Proc. of the Poster Papers of the European Conference on ML*, Prague, 128–137.

Witten, I., Frank, E. (2000), Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, Morgan Kaufmann, San Mateo, 2000.

Wolpert, D., 1992, Stacked Generalization. *Neural Networks* 5, 241–260.