Analysing the Trade-off between Comprehensibility and Accuracy in Mimetic Models[1]

Ricardo Blanco-Vega, José Hernández-Orallo, María José Ramírez-Quintana

Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia, C. de Vera s/n, 46022 Valencia, Spain.
{rblanco, jorallo, mramirez}@dsic.upv.es

Abstract. One of the main drawbacks of many machine learning techniques, such as neural networks or ensemble methods, is the incomprehensibility of the model produced. One possible solution to this problem is to consider the learned model as an oracle and generate a new model that "mimics" the semantics of the oracle by expressing it in the form of rules. In this paper we analyse experimentally the influence of pruning, the size of the invented dataset and the confidence of the examples in order to obtain shorter sets of rules without reducing too much the accuracy of the model. The experiments show that the factors analysed affect the mimetic model in different ways. We also show that by combining these factors in a proper way the quality of the mimetic model improves significantly wrt. other previous reports on the mimetic method.

1. Introduction

In this paper we analyse and improve a general method for converting the output of any incomprehensible model into one simple and comprehensible representation: set of rules. The goal of converting any data mining model into a set of rules may seem a chimera, but there are two feasible ways of achieving it. First, using many specific "translators" to convert each kind of model into rules. Secondly, using a general "translator" to convert any model into sets of rules.

There have been many techniques developed for the first approach, especially to convert neural networks into rules (rule extraction techniques), and also for other representations, such as support-vector machines. The second approach, even though it would be more generally applicable, it has not been analysed in the same extent, probably because it was unclear in which way a set of rules could be extracted from any kind of model, independently of its representation.

The solution to this problem cannot be easier, but it was recently been presented by Domingos [2][3]: we can treat the learned model as an oracle and generate a new labelled dataset with it (invented dataset). Next, the labelled dataset is used for learning a decision tree, such as C4.5, which ultimately generates a model in the form of rules. The process is shown in Figure 1. The first learning stage (top) uses any data mining modelling technique to obtain an accurate model, called the *oracle*. With this model we label a random dataset *R*, and jointly with the training set *T*, we train a second model, using a

[1] This work has been partially supported by CICYT under grant TIN 2004-7943-C04-02, Acción Integrada Hispano-Austríaca HU2003-003 and Generalitat Valenciana.

comprehensible data mining technique. The second model is called the *mimetic* model.

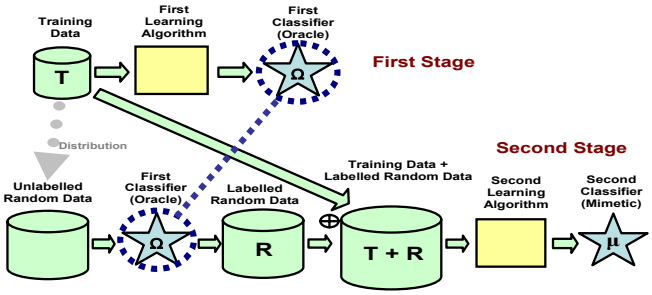

Fig.1. Mimetic Technique

The inventor of the technique, Domingos called it CMM (Combined Multiple Models) and used bagging [2] as oracle and C4.5rules as the final comprehensible model. He used a fixed number of randomly generated examples (1,000 for all the datasets). These were also joined to the original training set for learning the decision tree. In [4] we further analysed the method (which we called "mimetism") experimentally, for boosting, a different ensemble method. The results are quite consistent with Domingos. Additionally, in [5] we analysed the technique theoretically, proving that 100% fidelity is achievable with unpruned decision trees, as long as a sufficient large random sample is generated.

From all these results, we have learned some things about how "mimetic classifiers" work: A great number of random examples is necessary to achieve high fidelity, but the number of rules is also high. The use of the original training set (jointly with the random examples) is beneficial for the accuracy.

However, there are other issues where the behaviour of the method is not so clear: The method has not been applied to other oracles, especially oracles which are not ensembles, such as neural networks. The relationship between the comprehensibility of the resulting model and some factors such as: degree of pruning, number of random examples generated, etc., is also unknown.

In this work we analyse the method regarding these issues, concentrating especially on how short the sets of rules can be obtained without sacrificing too much the fidelity of the mimetic model with respect to the oracle.

In order to settle a precise reference metric, we define the following "quality metric", which represents a trade-off between comprehensibility (roughly represented here by the number of rules of the mimetic model) and accuracy:

$$Q = \frac{(\,Acc(Mim) - Acc(Ref)\,) \,/\, Acc(Ref)}{(Rules(Mim) - Rules(Ref)) \,/\, Rules(Ref)} \qquad (1)$$

The "reference model" (*Ref*) represents a comprehensible model learned directly with the original training set, such as C4.5 while Mim represents the mimetic model (possibly C4.5 as well). Obviously, if the results with the mimetic procedure are not better than with the reference model there would be no point in using the mimetic technique. As we will see, the factors that affect this quality metric Q are manifold and complex.

The paper is organised as follows. Section 2 introduces the experimental setting which has been used to perform the analysis of the mimetic technique.

Section 3 studies the influence of pruning on the quality and fidelity of mimetic classifiers using neural networks and boosting as oracles. The relation between the invented dataset size and the quality of the mimetic model is analysed in Section 4. A function to estimate the optimal size of the random dataset for each problem is also included. Section 5 modifies the random dataset by taking the confidence of the oracle into account. Section 6 includes a joint analysis about the combination of factors. Section 7 presents the conclusions and future work.

2. Experimental Setting

In this section we present the experimental setting used for the analysis of the mimetic method described in this paper. For the experiments, we have employed 20 datasets (to see Table 1) from the UCI repository [1]. For the generation of the invented dataset we use the technique proposed in [4].

Table 1. Information about datasets used in the experiments

No.	Dataset	Attr.	Num.Attr.	Nom.Attr.	Classes	Size	Missing
1	anneal	38	6	32	6	898	No
2	audiology	69	0	69	24	226	Yes
3	balance-scale	4	4	0	3	625	No
4	breast-cancer	9	0	9	2	286	Yes
5	cmc	9	2	7	3	1,473	No
6	colic	22	7	15	2	368	Yes
7	diabetes	8	8	0	2	768	No
8	hayes-roth	4	0	4	3	132	No
9	hepatitis	19	6	13	2	155	Yes
10	iris	4	4	0	3	150	No
11	letter	16	16	0	26	20,000	No
12	monks1	6	0	6	2	556	No
13	monks2	6	0	6	2	601	No
14	monks3	6	0	6	2	554	No
15	mushroom	22	0	22	2	8,124	Yes
16	sick	29	7	22	2	3,772	Yes
17	vote	16	0	16	2	435	Yes
18	vowel	13	10	3	11	990	No
19	waveform-5000	40	40	0	3	5,000	No
20	zoo	17	1	16	7	101	No

We have considered two kinds of oracles: Neural Networks and Boosting, using their implementations in the Weka data mining package (MultilayerPerceptron and AdaBoostM1, respectively). Also, the reference and the mimetic classifiers are constructed with the J48 algorithm included in Weka. The number of boost iterations is 10 in the AdaBoostM1 algorithm. In what follows, we denote the neural network oracle as NN, the Boosting oracle as Boost, the reference classifier as J48 and the mimetic classifier as Mim. Finally, when we show average results of many datasets, we will use the arithmetic mean of all datasets. For all the experiments, we use 10-fold cross-validation.

3. Analysis of Pruning

In this section we analyse how the quality metric and the mimetic classifier fidelity are affected by the use of different pruning degrees in the Mim algorithm. To do this, several experiments have been performed modifying the confidence threshold for pruning in the J48 algorithm; we have considered the following

values: 0.0001, 0.001, 0.01, 0.02, 0.05, 0.10, 0.15, 0.20, 0.25, and 0.3. In all cases, we have fixed the size of the invented random dataset to 10,000.

Figure 2 shows the average results when the oracle is a neural network (Mim NN) and when the oracle is Boosting (Mim Boost). We have also included as reference the accuracy obtained by the oracles and by the J48 algorithm learned with the original training set.

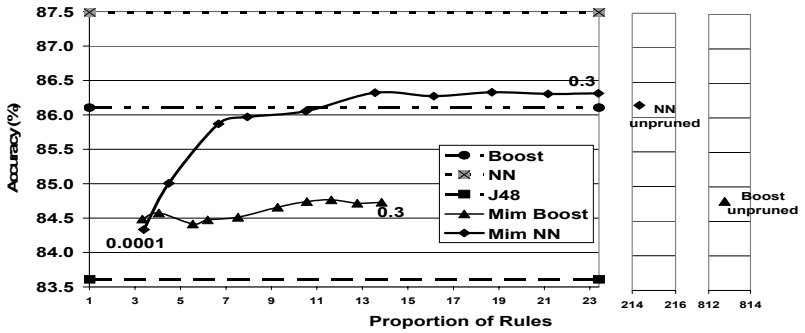

Fig. 2. Accuracy vs Proportion of rules of Mim for several degrees of pruning

As we can see, Mim has a better behaviour when the oracle is the neural network. The reason is that the neural network is in average a better oracle than Boosting (in terms of accuracy). Also, both of them are better than J48. In Figure 2 we can see that the increase in accuracy practically reaches its maximum with a pruning degree of 0.01 at a proportion of rules around 7 times more than the reference J48 classifier. This "optimal" point is corroborated by the quality metric which is also maximum for this point.

From all the previous results, it seems that pruning, or at least the pruning method included in J48, gives poor manoeuvrability to get good accuracy results with fewer rules. For these datasets, the best qualities are obtained with an increase of almost 2 points in accuracy but with decision trees which are 7 times larger than the original ones. Hence, in the following section, we study the influence of other more interesting factors such as the invented dataset size.

4. Analysis of the Invented Dataset Size

Previous works on mimetic classifiers [2][3][4] have considered a fixed size for the invented dataset (usually between 1,000 or 10,000). However, it is clear that this value is relatively small for datasets such as "letter" and relatively large for datasets such as "hayes-roth". In this section we want to better analyse the relationship between the invented dataset size and the quality of the mimetic model (in terms of the quality metric defined in Section 1).

In order to study this factor, we have performed experiments with several sizes for the invented dataset, from $0.3n$ to $6n$, where n is the size of the training set. Each increment is 0.3, making a total of 20 different sizes per dataset. We use the neural network as oracle and J48 as mimetic classifier.

Figure 3 shows how the quality metric evolves for increasing size of the invented dataset (the horizontal axis shows the proportion of rules). As we can

see, there is a maximum point at (3.06, 0.0071), with an invented dataset of 1.5 times larger than the training dataset, which means that, as expected, very short datasets have very low accuracy but, on the other side, it is not beneficial to generate datasets that are too big.

Since accuracy does not grow linearly, it is clear that we have a saturation point for the quality metric as the one shown in the picture. However, this saturation point is not reached at the same point for each dataset. Hence, considering 1.5 as a good value for all datasets would not be a good choice.

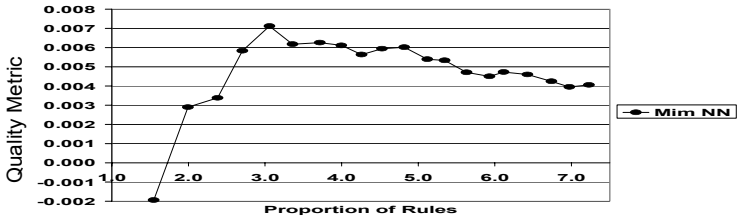

Fig. 3. Quality Metric vs Proportion of rules of Mim for several sizes

Having this in mind, we try to estimate the optimal size for each dataset. In order to do this, first, with the previous experiments, we will determine the size of the invented dataset for which the quality metric gives the best value. With this, we will have a different best factor for each of the 20 datasets. What we will do next is to use this data for estimating a function that returns the size of the random dataset which would be optimal for a new dataset.

For this, we use the following variables for each of the 20 datasets: the number of nominal attributes (NomAttr), the number of numerical attributes (NumAttr), the number of classes (Classes) and the size of the training dataset (Size). The output of the function is the factor φ (size random/size train) with respect to the original training set. More formally, we want to estimate the following function:

$$\varphi = f(\text{NomAttr}, \text{NumAttr}, \text{Classes}, \text{Size})$$

Due to the small number of examples for this estimation (20 datasets) we have used a simple modelling technique: multiple linear regression (with and without independent coefficient). In this way, we can estimate the optimal invented dataset as follows:

$$n = \varphi \times Siz.\ \ \varphi = 0.05097 \times NumAttr - 0.01436 \times NomAttr + 0.17077 \times Classes + 0.00003 \times Size$$

Table 2 shows the values estimates by the previous equation and the actual values for 5 fresh datasets.

Table 2. Estimated and real quality results for 5 fresh datasets. n max = Size when Q is maximum (actual). n calc = Size calculated with the formula (estimated). Q max = Actual quality metric. Q n calc = Estimated Q. Q 150% size = Q for fix invented dataset size of 150%

No	Dataset	Attr	NumAttr	NomAttr	Classes	Size	n max	n calc	Q max	Q n calc	Q 150% size
1	Autos	25	15	10	7	205	1051	374	-0.01716	-0.03600	-0,02439
2	CarsW	6	0	6	4	1728	466	1124	0.14232	0.09200	0,05983
3	Credit-a	15	6	9	2	690	2235	372	-0.00042	-0.00335	-0,00084
4	Heart-c	13	6	7	5	303	81	324	0.07948	0.04000	0,01813
5	Hypothyroid	29	7	22	4	3772	1018	3172	-0.00132	-0.00230	-0,00219
	Average								**0.04058**	**0.01807**	**0.010108**

As we can see, the estimation is not perfect, and the estimated values (Q n calc) are usually below the real values (Q max). However, if we look at the average values, the average quality obtained by using this estimation is significantly better

(0.018) than that obtained by a fix invented dataset size of 150% which is 0.01. This corroborates the idea of considering an appropriate size for each dataset, depending on, at least, the previous factors (number of nominal and numeric attributes, number of classes, and, size of the training set).

5. Use of Confidence in the Mimetic Method

When an invented dataset is used to learn a mimetic model, it is usually generated without taking into account the confidence of the oracle over this set of examples. We use the confidence of an example as the estimated probability of the predicted class. It seems reasonable to think that the quality of the mimetic model would improve if we use only those examples of the invented dataset for which the oracle gives a high confidence. In this section we analyse how the confidence of the invented dataset can be used to improve the mimetic technique.

In order to do this, once the invented dataset R has been constructed and added to the training set T, we process this set in the following way. First, we remove from $R+T$ all examples whose confidence value is below a confidence threshold t_c. Note that the examples of T are never removed because they have a confidence value of 1. Next, we remove the repeated examples. Finally, we duplicate the remaining examples a certain number of times depending on its confidence value. The resulting set, which we denote as D_{RT}, is used for training the mimetic model. The number of times that an example must occur in D_{RT} is defined as follows: let C_e be the confidence value of an example e and F a given repetition factor, then the number of occurrences of e in D_{RT} is $occ(e)= round(C_e \times F)$. Since the objective of the occ function is to determine whether the example must be duplicated or not, each example e for which $occ(e)=0$ remains in D_{RT}. Note that, as the repetition factor increases, examples with high confidence become more significant, and they may occur more than once in D_{RT}.

For the experiments the confidence thresholds t_c used were 0, 0.3, 0.9, 0.95, 0.98, 0.99 and 1.0, and for each one we used a repetition factor ranging from 1 to 4. The size of the invented dataset R was 10,000. Figure 4 shows these results.

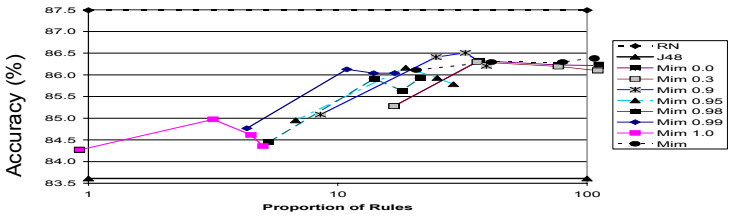

Fig. 4. Accuracy vs Proportion of Rules of Mim depending on a confidence and a repetition factor

We have observed that in the case of confidence threshold t_c=1.0 and repetition factor F=1, the process in general does not add invented examples to T (only adds a small number of invented examples for two datasets for which there were invented examples with confidence=1). In some problems, we even got smaller datasets D_{RT} than the original training sets. This is caused by the fact that some original datasets had repeated examples that were eliminated.

For the case of t_c=0.99 and F=1, we get a size of invented examples in D_{RT} around 3,000. If we contrast this value to a size of invented examples in D_{RT}

around 7,000 when we use t_c=0.0 (3,000 invented examples approx. are removed because they are repeated), we see that an important percentage of examples are given a confidence ≥ 0.99 by the NN. Consequently, with t_c=0.99 we have an intermediate situation which is more on the left of Figure 4 than the original Mim. Additionally, the accuracy is almost totally reached with this case (86.1).

Regarding the repetition factor, the behaviour is quite similar for all cases, but has different interpretations. For instance, for t_c=0.99 and for F=1 all the remaining examples are included once and for F=2 all the remaining examples are included twice. The important accuracy increase between these two cases can be justified by the fact that J48 has a limitation on the minimum number of examples per node, and this duplication allows J48 to be more detailed.

To confirm these observations we show in Figure 5 the quality metric for these experiments. As we can see, the best quality metric is obtained using a threshold confidence of 0.99 (Mim 0.99) with a repetition factor of 2.

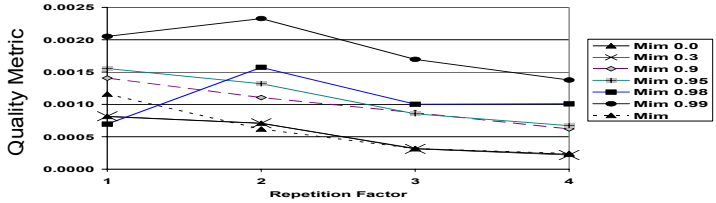

Fig. 5. Quality Metric vs Repetition Factor for different confidence threshold

6. Combination of Factors

Finally, we made an experiment combining some of the results obtained in the previous experiments. We used pruning at 0.01 and 0.1, the size of the invented dataset was set to the value predicted by the estimated model in Section 4, and the repetition factor was set to 2 and the level of confidence to 0.99. Table 3 shows the results of these both scenarios.

Table 3. Experimental results obtained by the combination of factors.

No.	NN	J48		Mim 0.01			Mim 0.1		
Dataset	Acc	Acc	Rules	Acc	Rules	Ratio	Acc	Rules	Ratio
1	98.89	98.56	39.50	98.18	48.60	1.23	98.39	53.05	1.34
2	83.21	77.33	30.20	85.05	53.30	1.77	85.76	53.10	1.76
3	90.84	78.40	39.60	77.88	32.83	0.83	79.60	52.45	1.32
4	67.96	74.08	7.50	70.39	3.17	0.42	72.17	19.15	2.55
5	50.86	51.57	155.70	54.51	47.67	0.31	51.69	228.55	1.47
6	81.94	85.13	5.50	84.40	5.20	0.95	85.44	6.25	1.14
7	74.42	74.19	19.20	74.53	27.20	1.42	72.53	63.50	3.31
8	81.20	68.58	19.00	74.74	22.63	1.19	77.25	24.65	1.30
9	80.06	79.43	9.40	79.35	2.27	0.24	79.63	5.55	0.59
10	96.81	94.96	4.70	95.33	4.77	1.01	94.67	4.65	0.99
11	82.08	87.98	1,158.10	87.65	1,037.20	0.90	85.69	16,655.10	14.38
12	100.00	97.12	30.10	100.00	28.00	0.93	100.00	28.00	0.93
13	100.00	63.29	24.50	65.72	1.00	0.04	65.72	1.00	0.04
14	98.49	98.92	14.00	96.95	9.97	0.71	98.92	13.70	0.98
15	100.00	100.00	25.00	99.90	90.10	3.60	99.98	161.70	6.47
16	96.84	98.68	28.60	98.32	10.65	0.37	98.32	10.65	0.37
17	94.49	96.55	5.80	95.49	2.33	0.40	96.22	5.70	0.98
18	93.15	79.75	128.00	79.50	155.75	1.22	82.93	494.9	3.87
19	95.02	92.39	8.30	92.82	12.03	1.45	92.68	14.15	1.70
20	83.54	75.36	290.70	76.54	177.10	0.61	76.02	1450.2	4.99
Avg.	87.49	83.61		84.36		0.98	84.68		2.52

The results with pruning level at 0.01 show that the three main factors considered (pruning, invented dataset size and relevance of the examples), if used together,

can dramatically reduce the number of rules. In fact, the average results show that the number of rules is even below J48 with its default parameters. In this scenario, however, the increase in accuracy is mild (from 83.61 to 84.36). The picture changes when we see the results with pruning level at 0.1. In this case, accuracy increases to 84.68 with a size of the models which only rises to 2.52 times more rules than the original J48 model. The quality is 0.0084.

7. Discussion and Conclusions

Summing up, from previous works and after the analysis on some of the separated factors (especially pruning), it seemed that it was almost impossible to improve the quality metric. Reducing the number of rules systematically entailed a reduction of accuracy and vice versa. However, the study of factors such as the size of the invented dataset and the modification of the distribution of examples are better tools to maintain significant improvements in accuracy while significantly reducing the number of rules. These final combined results suggest that there is still margin to pursue in this line, and that good compromises can be found, turning the mimetic technique originally introduced by Domingos, into a real useful and general technique for knowledge discovery.

Additionally, this work provides a further insight on how mimetic classifiers work. The use of the confidence of the oracle in order to modify the distribution of examples is one of the main new contributions of this work and suggests that the increase in number of rules can be partially due to overfitting to low-confidence examples generated by the oracle.

Finally, as future work, we would like to investigate several issues. For instance, instance selection methods could be useful for reducing the size of the invented dataset. The evaluation of mimetic models with other metrics, such as AUC (Area Under the ROC Curve), would also be interesting, since decision trees have better AUCs when the tree is not pruned [6]. Another issue to study would be to analyse the use of confidence without the training set, thus making the mimetic technique more generally applicable.

References

1. Black C. L.; Merz C. J.UCI repository of machine learning databases, 1998.
2. Domingos, P. Knowledge Discovery Via Multiple Models. *Intelligent Data Analysis*, 2(1-4): 187-202, 1998.
3. Domingos, P. Learning Multiple Models without Sacrificing Comprehen-sibility, *Proc. of the 14th National Conf. on AI*, pp:829, 1997.
4. Estruch, V.; Ferri, C.; Hernandez-Orallo, J.; Ramirez-Quintana, M.J. Simple Mimetic Classifiers, *Proc. of the Third Int. Conf. on Machine Learning and Data Mining in Pattern Recognition*, LNCS 2734, pp:156-171, 2003.
5. Estruch, V.; Hernández-Orallo, J.: Theoretical Issues of Mimetic Classifiers, TR DSIC, http://www.dsic.upv.es/~flip/papers/mim.ps.gz, 2003.
6. Provost, F.; Domingos, P. Tree induction for probability-based rankings, *Machine Learning*, 52 (3): 199-215, 2003.