Knowledge Acquisition through Machine Learning: Minimising Expert's Effort

Ricardo Blanco-Vega, José Hernández-Orallo, María José Ramírez-Quintana
Dep. de Sist. Informáticos y Computación, Universidad Politécnica de Valencia,
C. de Vera s/n, 46022 Valencia, Spain
{rblanco, jorallo, mramirez}@dsic.upv.es

Abstract

Machine learning can be applied to solve the knowledge acquisition bottleneck in many areas where an expert makes predictions to single cases, such as diagnosis, estimation, etc. The idea is to query the expert with as many cases as possible and get their answers. With this data we train a machine learning model which mimics the expert's behaviour. This is just a simple application of a modelling technique known as "mimetism", which has many other applications. This "soft" approach to knowledge acquisition has many advantages: any machine learning technique can be used, the expert must only answer simple questions (cases) and we can combine the decisions of several experts easily. However, one problem of this approach is that we do not know in advance how many cases we will need to ask in order to get a good model which is accurate wrt. the expert's knowledge. Obviously, as more data is labelled by the expert better results are obtained. However, asking thousands of cases to the expert is usually impractical. In this paper, we analyse the behaviour of knowledge acquisition through mimetic learning according to two factors: accuracy and comprehensibility of the resulting model and we devise a method to compute the minimum number of cases that we need to ask the expert to attain a certain quality level.

1. Introduction

In many areas, we need the help of one or more experts to support decision making. Since the persistent need of these experts is costly, several approaches have been presented to substitute or automate these decisions, most notably through the use of expert systems. One of the main problems in expert systems is the knowledge acquisition bottleneck [7], since many experts are not able to write down their knowledge in clear and unambiguous rules. They usually behave by explicit rules, rules of thumb and unconscious rules. Even in case the expert is able to write down all their knowledge this requires a high effort, can be very time-consuming, is difficult to maintain and sometimes the result is a transcribed model that cannot be applied in a fully automated way since there are still some ambiguity.

In many applications, such as diagnosis, estimation, detection, selection, etc., cases are described by a fixed series of attributes (either nominal or numerical) and a dependent value (either nominal or numerical). The expert's model predicts the dependent value according to the rest of attributes. This model structure is similar to predictive models in machine learning, where we have cases with input variables and an output variable.

In this case, the knowledge acquisition problem can be addressed by machine learning techniques. The idea is that we can query the expert to construct our model. Several methods exist for this, and this kind of learning is called "query learning" [1], where the expert acts as an oracle.

This "soft" approach has many advantages, we control the representation of the model (rules, equations, ...), once it is finished it is unambiguous and hence fully automatisable and, finally, we can query several experts (either combined or specialised to parts of the problem).

However, the "query learning" approach, in general, has some disadvantages for this application: not many machine learning techniques are designed for learning by making queries, in some cases the queries can be complex, and, most importantly, most query learning paradigms assume the expert must be there during training, and give the answers immediately. Otherwise the learning algorithms could delay for hours or even days, which would make the expert's availability problem even worse. Additionally, in existing analysis of query learning, the analysis of the complexity of the model has not been taken into account.

In this paper, we investigate the use of the mimetic method ([3],[6]) for knowledge acquisition. The mimetic method just uses simple cases, i.e. unlabelled examples, for which the expert must only provide the output value. The result is a dataset, which can be then used to train a model by using any off-the-shelf machine learning method. As we can see in the following picture, the

method can be applied easily by using any machine learning tool or data mining package:

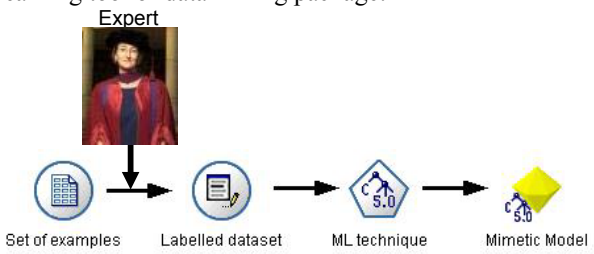

Figure 1. Mimetic process with expert oracle

The process starts then by asking the expert for the labels for a set of examples. With this we have a labelled dataset which we use to train a model by using our favourite machine learning technique.

As expected, the model will be more accurate the greatest the number of examples that we can ask the expert is. In fact, this can be as accurate as we want provided that we can ask the expert as many cases as we want ([3],[4],[6]) .

However, there is no knowledge acquisition technique whatsoever where we can have an infinite expert's availability. Consequently, we have to analyse how many cases we require in order to get a model which captures the expert's knowledge with high accuracy and high comprehensibility, with a minimum number of queries.

In this analysis we have to realise first that accuracy is benefited by an increasing number of cases, but comprehensibility is generally not. On the contrary, the greater the number of cases we have, the more complex the model will be, generally. One idea to solve this dilemma might seem to use pruning (for those machine learning techniques which have such a technique), but this would require to generate much more cases than really needed, saturating our expert uselessly. As we will see, we can control the comprehensibility of our models (in terms of number of rules) by gauging the number of cases that we ask the expert to label.

It is also important to decide in an initial state how many cases will be required, without the need of coming and going to the expert for more cases repeatedly. In order to avoid this, we devise a method based on learning curves which is able to predict the number of cases that will be required given a trade-off between accuracy and comprehensibility of the models.

The paper is organised as follows. Section 2 introduces an MML approach to address the optimality of a model generated from a dataset labelled by the expert, taking into account several factors, accuracy, query cost and comprehensibility. From it, Section 3 presents an approach to determine the optimal size of the dataset applying a modification of the MML principle. Section 4 shows the setting used in Section 5 for the experimental evaluation of our approach. Section 6 presents how to use this approach in practice for a given knowledge acquisition problem. Finally, we summarise and conclude the paper with the results and the future work.

2. Trade-off Analysis

The mimetic method ([3],[4],[5],[6]) is a technique for converting an incomprehensible model into one simple and comprehensible representation. Basically, it considers the incomprehensible model as an oracle, which is used for labelling an invented dataset. Then, a technique which can generate comprehensible models (for instance, a decision tree) is trained with the invented dataset. The mimetic technique has usually been used for obtaining comprehensible models from ensemble methods (Domingo's original idea, [4],[5],[6]) or from other non-comprehensible sources, such as neural networks [3]. In this paper we propose to use this technique in a different way, by assuming that the oracle is not an incomprehensible model but a human expert. Then, this expert labels the invented data which is the only dataset used for training the mimetic model.

It has been shown ([3], [4]) that the following three factors of the mimetic model are related: the size of the invented dataset used for training the mimetic model, its number of rules and its accuracy. So, the accuracy increases as the size of the invented data increases, whereas fewer rules (and thus, greater comprehensibility of the model) are obtained using smaller invented datasets. From these results a trade-off between accuracy and number of rules (comprehensibility) seems to be needed. The factor which is more suitable for carrying out this study is the size of the invented dataset, since it is related to both accuracy and the number of rules.

Even though in this case the source is a human expert, the main idea is to consider the construction of the mimetic model as a learning problem from a dataset. A very common way of analysing the relation between size of the model and the level of error is the minimum message length (MML) principle [9]. We use MML in order to determine the optimal size of the invented dataset that maximises accuracy and comprehensibility of the mimetic model.

For a hypothesis H and data D, we have from Bayes theorem:

$$p(H \cap D) = p(H) \cdot p(D|H) = p(D) \cdot p(H|D)$$

where $p(H)$ is the *prior* probability of hypothesis H, $p(H|D)$ is the *posterior* probability of hypothesis H and $p(D|H)$ is the *likelihood* of the hypothesis, actually a function of the data given H.

From Shannon's Communication Theory we know that with an optimal code, the *message length* of an event E, MsgLen(E), where E has probability $p(E)$, is given by $\text{MsgLen}(E) = -\log_2(p(E))$. Therefore:

$$\text{MsgLen}(H \cap D) = \text{MsgLen}(H) + \text{MsgLen}(D|H) \quad (1)$$

As we can see in (1), the message is split in two parts: the first one corresponds to the model (its message length), and the second one corresponds to the data given the model (the message length for encoding the data of D that are errors w.r.t. H).

Finally, the MML principle establishes that models with shorter encoded messages are preferable.

Now, we use the above result for determining the cost of the mimetic model. Given a model M learned by applying the mimetic technique by using an invented dataset D labelled by an expert as training set, we define the cost of M as:

$$\text{Cost}(M) = \text{MsgLen}(M) + \text{MsgLen}(D|M) + \text{Query}(D) \quad (2)$$

Note that, unlike (1), we have included in (2) the cost of querying the expert for labelling the invented dataset D (Query(D) factor), due to the limited availability of the expert, as we have said in Section 1. Then, the problem of determining an optimal size for D can be seen as an optimisation process whose objective is to maximise the accuracy and the comprehensibility (in terms of number of rules) of the model given some constraints (the number of queries), which are also included in the equation. As we will see next, these constraints are connected to the rest by the learning curves for the mimetic model, which represent the variability of the number of rules and errors of the model depending on the size of the invented dataset.

3. Optimisation of the Size of D using a Modified MML

In this section we present our approach based on a modification of the MML principle (which takes into consideration the cost of labelling the data) to determine an optimal size for the invented dataset which is used for learning the mimetic model. First, we obtain the learning curves using several data sets from the UCI repository [2] and then we use them in the previous equation (2).

3.1. Learning Curves for the Mimetic Models

As we have mentioned in Section 2, we will use the learning curves of the mimetic model to estimate the optimal points for equation (2). These curves represent the relationship between the factors we are interested in: number of rules w.r.t. size of the dataset (D) and number of errors w.r.t. size of D. To obtain them, we have induced a hundred of mimetic models (for each dataset) varying the size of D from an initial size that corresponds to the size of the training dataset in the UCI, and increasing the size 5% each time. Figures 2 and 3 show, as an example, the curves obtained for the balance-scale dataset.

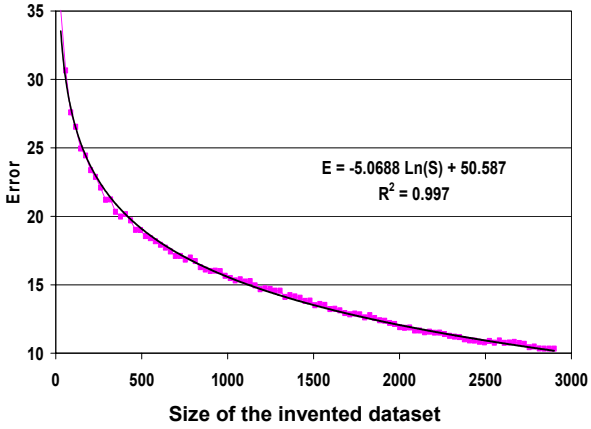

Figure 2. Error vs. size for the balance-scale dataset

For the first one, the learning curve *Error vs. Size* can be described as

$$E = \alpha * \text{Ln}(S) + \beta \quad (3)$$

where E is the error, S the size and α and β are constants (determined by a linear regression on the logarithmic equation). Analogously, the learning curve *Number of Rules vs. Size* [8] can be described as

$$R = \delta * S + \lambda \quad (4)$$

where R is the number of rules, S is the size of D and δ and λ are constants (also determined by linear regression). Both functions will be used in the following subsection in order to obtain the optimal size S.

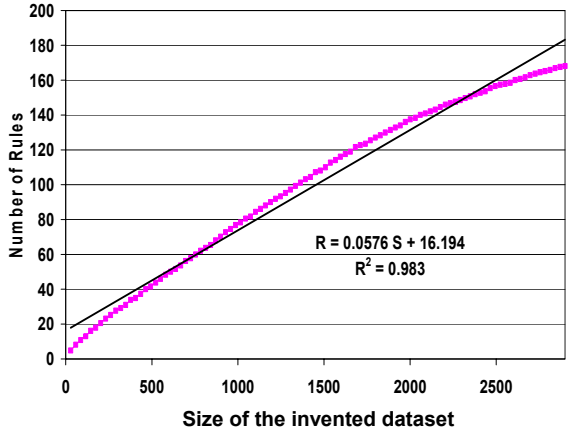

Figure 3. Number of rules vs. size for the balance-scale dataset

Regarding Figure 3, it can be noted that there exists a clear linear relation between the number of rules of the model and the size of the invented dataset. This relation will be used to simplify equation (2).

3.2. Calculating the optimal size by a modified MML

Considering that a model M consists of R rules and that the cost of encoding each rule is cr, the cost of encoding M can be approximated by:

$$\text{MsgLen}(M) \approx R*cr \quad (5)$$

Likewise, if there exist E errors and the cost of encoding each of them is ce, then the message length of dataset D given M can also be approximated by:

$$\text{MsgLen}(D/M) \approx E*ce \quad (6)$$

Finally, the cost of querying the expert is

$$\text{Query}(D) \approx |D|*cq \quad (7)$$

where $|D|$ is the size of dataset D and cq is the cost of labelling an example.

Replacing (5), (6) and (7) in (2) we have that:

$$\text{Cost(Model)} \approx R*cr + E*ce + |D|*cq \quad (8)$$

Considering the linear relation between the number of rules R and the size of the invented dataset $|D|$, we can approximate Equation (8) by

$$\text{Cost(Model)} \approx R*cr' + E*ce \quad (9)$$

where R and cr' represents the number of rules and the size of D, and the costs of encoding a rule and labelling an example, respectively. The last equation is the objective function for the optimisation process. Then, we replace in (9) R by formula (4) and E by formula (3) and finally we derive the resulting equation w.r.t. S, which we use for obtaining its critical points (maximum and minimum). The first derivative (equation (9)) shows that the optimal size of D is proportional to the quotient α/δ:

$$S_{opt} = -K*\alpha/\delta \quad (10)$$

where K is a constant of proportionality between the unitary costs ce and cr' (fixed by ce/cr').

To know whether S_{opt} is a maximum or a minimum, we calculate the second derivative, which is

$$-K*\alpha/S^2 \quad (11)$$

Since K and S are both positive, the sign of (11) only depends on the sign of α. If α is positive, it means that the number of rules of the mimetic model decreases as the size of the training set increases which is an anomalous behaviour (and we do not use it since it means that it is not possible to compute a minimum size for D). Then the parameter α is negative, and hence the S_{opt} value corresponds to a minimum.

Now, we are ready to use this calculus for practical applications (Section 5). As we will see, for a given problem, we generate mimetic models using invented datasets of increasing size each time. Then, we approximate the learning curves and compute their parameters. Finally, we apply formula (8) to obtain the optimal final size of the invented dataset.

4. Experimental Setting

In this section, we present the setting used for the experimental evaluation of our approach included in the following section. Instead of using real experts, we use datasets, from which we learn a neural network which acts as our "expert". We have chosen this scenario because the neural network is able to capture complex patterns and is quite different from the technique we use for the mimetic model, a decision tree.

In order to cover several situations and kinds of problems, we have used 18 datasets (showed in Table 1) from the UCI repository [2].

For the generation of the invented dataset we use the uniform distribution which corresponds to the worst case: we have not used the original dataset and we do not know the real data distribution. If we were able to use this later information, however, the results of the mimetic model would be improved, since the mimetic technique usually works better when the invented dataset is generated according to the real distribution.

Table 1. Datasets used for the experiments

No.	Data	Num. Atr.	Nom. Atr.	Classes	Size
1	anneal	6	32	6	898
2	audiology	0	69	24	226
3	balance-scale	4	0	3	625
4	breast-cancer	0	9	2	286
5	Cmc	2	7	3	1,473
6	Colic	7	15	2	368
7	diabetes	8	0	2	768
8	hayes-roth	0	4	3	132
9	hepatitis	6	13	2	155
10	iris	4	0	3	150
11	monks1	0	6	2	556
12	monks2	0	6	2	601
13	monks3	0	6	2	554
14	sick	7	22	2	3,772
15	vote	0	16	2	435
16	vowel	10	3	11	990
17	waveform-5000	40	0	3	5,000
18	zoo	1	16	7	101

The neural network, which "acts" as the expert, is the MultilayerPerceptron method in the Weka data mining package [10], with the default parameters. The mimetic classifiers are constructed with the J48 algorithm included in Weka also using its default configuration. Finally, when we show average results of many datasets, we will use the arithmetic mean of all datasets. For the accuracies and number of rules shown in all the experiments, we used 10-fold cross-validation.

5. Experimental Evaluation

Given the previous experimental setting, in this section we show how we can reliably estimate the learning curves with just three models. This will make it possible to ask the expert the set of cases in a few rows, not more and not less than needed, and without being online with the algorithm.

In particular, we want to estimate the previous parameters α and δ isolated in equation 10 before. We analyse if we can estimate these values accurately by using only three models with different sizes (we have used sizes of 5%, 150% and 450% of size of the original dataset).

Table 3 shows the parameters α and δ for the 18 datasets, estimated using these three models for each. We also show the determination coefficients. The average value for this coefficient is 0.97 for the error and 0.94 for the number of rules, which means that 97% of the error variability is explained by the Error vs. Size learning curve and 94% of the variability in the number of rules is explained by the Rules vs. Size learning curve.

Table 2. Parameters and determination coefficients for the learning curves with three points (*n*=3)

Dataset	n=3			
	Error vs Size		Rules vs Size	
	α	R^2	δ	R^2
1	-4.5711	0.97	0.0776	0.98
2	-7.9902	0.99	0.1808	0.99
3	-5.4376	1.00	0.0591	0.99
4	-0.4968	0.74	0.0949	1.00
5	-1.5936	0.93	0.0956	1.00
6	-2.66	0.95	0.0097	0.96
7	-1.2611	0.98	0.0445	1.00
8	-7.197	0.96	0.0699	0.93
9	-3.628	0.98	0.0456	0.99
10	-8.9288	0.97	0.0498	0.98
11	-7.6452	0.95	0.0081	0.49
12	-7.5818	0.95	0.1045	0.97
13	-4.1454	0.94	0.0043	0.73
14	-0.2817	1.00	0.0068	0.99
15	-1.5628	0.99	0.0267	0.99
16	-4.9703	1.00	0.2144	1.00
17	-3.1991	0.99	0.1099	1.00
18	-10.865	0.99	0.1083	0.99
Avg		0.97		0.94

In order to confirm the previous results and clarify that we can estimate this optimal point with a curve obtained from only three different sizes, we compare the estimated cost with the value obtained from equation (9) and the best possible cost (obtained by analysing much more sizes, 100). This is shown in Table 3.

We show that there is no significant difference between the estimation and the real point, as it is whon by a test t on the results on the 18 datasets. For these values we used K=10, but similar results are obtained with other values of K.

Table 3 Hypothesis test for K=10

Dataset	Minimum Cost	
	Estimated from 3 sizes	Best from 100 sizes
1	213.15	202.922
2	525.72	481.648
3	224.54	227.275
4	326.45	307.864
5	568.83	543.152
6	172.27	179.361
7	298.85	289.335
8	191.32	218.889
9	226.46	225.327
10	23.83	90.752
12	250.30	297.48
14	67.79	61.349
15	85.04	74.915
16	810.62	782.523
17	443.81	427.34
18	142.27	158.163
Avg	285.70	285.52
Average difference =		-0.19
Standard deviation =		28.42929
	tc=	-0.02606
	t(15,0.01)	2.947

6. Application Procedure

Once shown that three sizes are enought to obtain an almost identical optimal point than examining all the possible sizes, we apply this general result to our problem at hand: we want to minimise the number of interviews to the expert without exceeding the number of required cases to be labelled.

In order to do this, we propose the following procedure:

```
size_set= {10, 20};  // initial number of examples
margin= 0.1;  // percentage of error wrt. the optimum size.

i= 20;
while(true) {
   Ask_Expert_Until(i);
   opt= Estimate_Opt_Value(size_set);
    if ((opt < i) || (i/opt > 1 - margin))
      break;
    else {
      i= opt;      // quick approach
      size_set= size_set ∪ { i };
    }
}
```

Figure 4. Procedure to estimate the optimal size

The algorithm starts asking 20 examples to the expert (from which we take one model with 10 examples and another one with 20 examples). With this we compute a first curve, from which we can estimate the optimum value (this is done in the funcion Estimate_Opt_Value()). If the estimated size is smaller than the number of examples asked to the expert (strange situation) or the deviation between the estimated value and the number of

examples is small then we stop. Otherwise, we ask the expert the remainder cases until the estimated size.

An example of the trace of the previous algorithm for the "zoo" dataset is as follows:

Iteration	*i*	*opt*	*i/opt*
1	20	207	0.1
2	207	340	0.6
3	340	353	0.97 (STOP)

As we can see, we only execute three iterations (three interviews to the expert) and without exceeding the optimal value (in this case it was finally around 350), we can find the optimal point for the knowledge acquisition problem. Similar results are obtained for the other datasets.

7. Conclusions

In this paper we have analysed a scenario where knowledge acquisition is made through simple queries (several case outputs) to one or more experts. This scenario is not applicable for any knowledge acquisition problem, but it can be a practical, easy-to-implement and general approach in many situations, especially in diagnosis and problems where the cases are well-structured but the expert's model is not easily explainable in a dozen of rules.

In these situations, the mimetic technique is directly applicable, since we do not need anything more than the expert, some unlabelled data (which can be generated by a simple uniform distribution) and any machine learning technique. As we know, the greater the dataset which the expert is able to label, the better the results will be. This sets up a dilemma between the quality of the model which captures the experts' knowledge and the size of the model on one hand (and hence, its comprehensibility) and the cost/availability of the expert on the other. Using learning curves for several datasets, we have seen that the cost of the expert (size of the dataset) and the size of the model are linearly correlated (at least for decision trees), and hence we can simplify our MML analysis to just two factors: the accuracy aimed and the size of model/data. These two factors are reflected by a single ratio K, which can be set for different contexts.

Given these general results we have devised a methodology to estimate the number of cases needed to obtain the "optimal" model, in terms of the trade-off between accuracy, number of queries and size of the model. We have seen that we can approach this value quite reasonably by usually three iterations, which means that we will ask a few cases to the expert on a first interview, and on the second and third interview we will know almost exactly the number of cases which will be required.

The results and the proposed methodology are then a step forward in making knowledge acquisition through machine learning much more practical and easy, which can help to solve the knowledge acquisition bottleneck.

As future work, we would like to investigate several issues. The possibility of grouping similar cases by clustering techniques and then ask the expert to label the clusters, in order to minimise the expert effort. Also, we would like to study how our approach can be applied for other machine learning methods and other machine learning tasks (e.g. regression).

Acknowledgements

This work has been partially supported by Ministerio de Ciencia y Tecnología of España project SELF under grant TIN 2004-7943-C04-02, Generalitat Valenciana project META-MIDAS under grant GV04A-389, SEIT-ANUIES scholarship and license commission of the Dirección General de Institutos Tecnológicos of México. We also thank the anonymous reviewers for their suggestions and, most especially, for pointing some of the future work mentioned above.

References

[1] Angluin. D. (1987) *Queries and concept learning*. Machine Learning, 2:319-342.

[2] Black C. L.; Merz C. J. (1998) UCI repository of machine learning databases

[3] Blanco-Vega R., Hernández-Orallo J., Ramírez-Quintana Ma. J. Analysing the Trade-off between Comprehensibility and Accuracy in Mimetic Models. The 7th International Conference on Discovery Science, 2004.

[4] Domingos, P. (1998). Knowledge Discovery Via Multiple Models. Intelligent Data Analysis, 2(1-4): 187-202.

[5] Domingos, P. Learning Multiple Models without Sacrificing Comprehensibility, Proc. of the 14th National Conf. on AI, pp:829, 1997.

[6] 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.

[7] Feigenbaum, Edward A. (2003). Some challenges and grand challenges for computational intelligence. Journal of the ACM (JACM), 50:32-40.

[8] Oates Tim; Jensen David (1997). The Effects of Training Set Size on Decision Tree Complexity. Proc of the Fourteenth Intl Conf. on Machine Learning: 254 – 262. Morgan Kaufmann.

[9] Wallace, C.S., Boulton, D.M. (1968). An information measure for classification. Computer Journal 11 185-194

[10] Witten, Ian H. and Frank, Eibe. (2000). Data Mining: Practical ML tools with Java implementations. Morgan Kaufmann.