Using a Domain Expert in Semi-supervised Learning

Angela Finlayson and Paul Compton

School of Computer Science and Engineering,
The University of New South Wales Sydney 2052, Australia
{angf,compton}@cse.unsw.edu.au

Abstract. Semi-supervised learning requires some data to be labeled but then uses this in conjunction with a large amount of unlabeled data to learn a model for a domain. Since the labeled data should be representative of the range of unlabeled data available, the aim of this research is to identify which data should be labeled. An approach has been developed where a domain expert starts to label unlabeled data and also writes rules to classify such data. The labeled data are also used as machine learning training data. If the expert rules and the rules developed by machine learning agree on a label for an unseen datum, the label is accepted and the case automatically added to the training data for learning, otherwise the case is checked by the expert and if the label from the rules is wrong, the expert provides the correct label and a rule to correctly classify the case. Further data is then processed in the same way. Results from a number of datasets using a simulated expert as the domain expert suggest that this method produces more accurate knowledge bases than other semi-supervised methods using similar amounts of labeled data and the resultant knowledge bases are as accurate as having all the data labeled.

Keywords: Semi-supervised learning, active learning, knowledge engineering, ripple-down rules.

1 Introduction

Semi-supervised learning is concerned with trying to learn from a dataset which contains unlabeled data and a relatively small amount of labeled data [1,2]. The main focus of semi-supervised learning research has been to find synergies between the labeled and unlabeled data for learning. For example, one might use the varying distribution of a large volume of unlabeled data to find possible clusters and then use labeled data within the clusters to label the unlabeled data. A wide range of interesting techniques have been developed for combining labeled and unlabeled data [1] (this review updated in 2008).

One approach to semi-supervised learning is to exploit disagreements between learners [3]. Disagreement-based learning arose out of research in co-training [4], where two learners can learn from two different views or aspects of the labeled data, (e.g. web page headings versus web page text). If one learner is more confident of correctly labeling some unlabeled data the label it assigns can then be used by the

Y.S. Kim et al (Eds.): PKAW 2014, LNCS 8863, pp. 99–111, 2014.

weaker learner. Similarly one can exploit the differences between different learning algorithms even when using data from a single view [5]. Active learning has also been used together with semi-supervised learning to identify which data in the unlabeled data would it be most useful to have labeled by a domain expert [6,7]. Parsazad et al use an evolutionary approach motivated by the idea of generating antibodies to antigens to identify which data should be labeled [8].

In a method such as that of [8] all the data is available and the goal is to see which of this data should be labeled by a domain expert. A different problem is where there is a stream of incoming data and one needs to decide which data needs to be referred to a domain expert for labeling.

2 Aim

The aim of the research here was to develop techniques to identify which data in a data stream should be labeled by a domain expert to facilitate semi-supervised without considering data that is as yet unseen. We [9] and others e.g. [10] have previously considered a number of techniques for identifying when a data stream contains a case different from those seen previously and one of these is the basis of the approach here. Two different knowledge bases are used to process incoming cases, one built manually and the other built by machine learning. If they both agree on the label for some unlabeled data, that classification is accepted otherwise the data is checked by a domain expert and if the case was not labeled correctly by the manual knowledge base, the expert provides the correct label and also a rule to correct the knowledge base so it will assign this label to the case.

As will be described below the domain expert used in the experiments here is a simulated expert. This is not a part of the proposed semi-supervised method, as in industrial applications a human domain expert would be used; however, using a simulated expert makes it possible to carry out multiple repeat studies and investigate a number of domains.

3 Methods

3.1 Summary

The data we stream in these experiments as unlabeled data comes from already labeled data sets. The simulated expert is a knowledge base built from the initially labeled data using machine learning. This includes all the data used for training or testing, as the simulated expert is meant to be an oracle who can correctly label any case in the domain. To ensure this, any cases misclassified by the machine-learning knowledge base built from all the data are removed from the training data used for later semi-supervised learning. For example pruning may result in training cases being misclassified. We call this knowledge base a “simulated expert” since it can correctly classify any of the cases that will be later used in the training part of the experiments and can also provide a classification rule for a given case using a selection of conditions from its rule trace for that case.

With the simulated expert built, the data is then split into 50% test data and 50% training data with cases the simulated expert can't correctly classify removed from the training data, but not the test data. We include all data in the test data to allow comparison with other methods. Two knowledge bases are then developed, one built from rules provided by the domain expert (the simulated expert) and one built by machine learning using all the training data which has been labeled to date. The process can start with both knowledge bases empty, or with some cases initially labeled and for which rules have been acquired from the expert. An unlabeled case is passed to both knowledge bases. If both the manually built knowledge base and the machine learning knowledge base give the same classification for the unlabeled case, then that label is attached to that case regardless of whether it is the correct label or not. If the two knowledge bases disagree, or neither is able to assign a label to the data, then the (simulated) domain expert is consulted to provide the correct label for that case and to add a rule to correctly assign the correct label. Rules are added to the knowledge base, using the Ripple-Down Rule knowledge acquisition method discussed below. A number of different experiments have been carried out using different initial labeling strategies, or having no cases labeled initially.

When all the stream of training data has been processed the manually built knowledge base and the machine-learning-built knowledge base are tested on the 50% of the data kept as test data. Note that the test data used included data the simulated expert could not correctly classify to allow for comparison with other methods. For comparison we also tested the performance of the machine learning using number of different selections of cases to be labelled. All studies were repeated 10 times with the selection of data randomized between test and training data. The results shown are average of the 10 experiments ± one standard deviation.

3.2 Datasets Used

We report here on the use of the technique on 13 different datasets; with the results of one dataset presented in detail and the others in summary. The detailed study uses data that comes from the Garvan Institute of Medical Research. The dataset used is a larger version of the thyroid dataset available from the UCIrvine Machine Learning Repository. It contains 43,472 records dating from 1979 to 1990. As with the UCIrvine data, this data had been run through the medical expert system GARVAN-ES1 [11] to ensure consistent classifications were provided for the data. However, in the studies here the full range of the 56 classes provided by Garvan-ES1 (and Garvan endocrinologists) were used, whereas for the UCIrvine data, the data was classified into fewer more coarse-grained classes. It is likely that differences between some of the 56 classes is small, but the endocrinologists expected the expert system to emulate the way in which they distinguished such classes, so we used the full 56 classes to provide a difficult real-world domain. The distribution of classes is very skewed with 16 classes covering 95% of the cases (with the top class covering 69% of cases) and the bottom 13 classes covering less than 10 records each. Another difficulty is that if one selects different subsets from the 11 years of patient records for machine learning, quite different knowledge bases are produced [12]. Although using GARVAN-ES1 to classify the records, ensured consistent classification, the population distribution probably varied over time because the records available depend on the patterns of referral and investigation by clinicians, particularly specialist endocrinologists who requested the bulk of the thyroid tests.

The dataset contains 7 numerical fields providing laboratory results, and fields for patient age and sex, the name of referring clinician or clinic and the referring clinician's (brief) clinical notes. The fields for age, sex, referral and clinical notes were preprocessed by the GARVAN-ES1 expert system, so that data used in this experiments consist of 8 numerical fields (including age) and 21 Boolean attributes extracted from the preprocessed data. There is a large amount of missing data in the dataset. The missing data for the laboratory test results range from 15% missing for one laboratory test to 95% missing for another. On average a patient record includes 3.6 laboratory results out of a possible 7. This is not because the data has been lost, but because clinicians will only request those tests that they think are relevant to their particular diagnostic hypothesis or patient management query. As well about half the records do not contain clinical notes. In summary this is a fairly difficult real-world dataset for machine learning. Of particular significance is that the UCIrvine Garvan data is seen as too skewed for semi-supervised learning [13]. The Garvan data used here is even more skewed because of the inclusion of classes with very few representatives.

The other datasets used are standard datasets from the UCIrvine dataset repository.

3.3 The Domain Expert

In the experiments carried out the domain expert built hundreds of different knowledge bases to explore different options. It is clearly impossible to carry out such large numbers of knowledge acquisition experiments with a human expert, and in the knowledge acquisition literature there are virtually no repeat experiments using a human expert. To make the experiments possible a simulated expert was used. The task for the simulated expert was to correctly classify data and also to provide a rule to give that classification. Any classification expert system for a domain that also provides a rule trace could be used as this sort of simulated expert. A selection of the conditions from the rule trace can be used as if they were the conditions an expert might provide for a rule to give the desired classification. This sort of simulated expert has long been used for evaluating various Ripple-Down Rule algorithms [14]. In previous work various selections of conditions from the rule trace have been used, as a crude way of simulating levels of expertise; here 75% of the conditions in the rule trace were selected randomly, (with standard numerical rounding used where required to specify the number of conditions in the rule). We do not wish to suggest that this sort of simulated expert creates the same sort of rules as a human expert; however for the purpose of the experiments here it does provide a useable rule. In particular human experts will probably identify the most important and discriminatory conditions for a rule, even if these are not complete, whereas as a random selection of conditions from a rule trace may select less relevant conditions.

The simulated expert was built with J48 and using all the data to be used later as training or test data in the semi-supervised learning experiments. Any data that was misclassified by the resulting knowledge base, because of pruning etc was discarded from the data to be used as training data in the semi-supervised learning experiments. This meant that although the simulated expert could correctly label all the cases in the training data for semi-supervised learning. However, whether its rules were good rules or not is a different question. This corresponds with normal practice with a domain expert, where one expects the domain expert be able to correctly classify items in the domain, but may or may not provide optimal rules to give these classification. Although not necessary the simulated expert was relearned for each run of the semi-supervised learning experiments and more importantly a different randomization of test and training data was created for each run. The use of a simulated expert is not intrinsic to the proposed semi-supervised learning method; it is purely to make repeat experiments feasible, which would not be possible with a human expert.

3.4 Ripple-Down Rules

Other types of knowledge-based technology could be used, but Ripple-Down Rules (RDR) are ideal for this application as rules are added case by case, whenever a case is given the incorrect conclusion (perhaps no conclusion) by the knowledge base. There is a fixed structure into which new rules are automatically located, so the domain expert is only concerned with the rule being added, not the overall structure of the knowledge base. The new rule is either a refinement correcting a previous rule, or a rule covering a new type of case. The new rule is tested against previous stored cases and if it gives the wrong conclusion for these cases, further conditions are add to refine the rule. In practice an expert only has to see two or three such cases before the rule is sufficiently precise to exclude all previous cases. Using the simulated expert, if the initial rule provided is not precise enough, further conditions are randomly selected from the rule trace and added to the rule until previous cases are excluded.

RDR are fairly widely used in industry, particularly for interpreting medical laboratory data and log data shows that across 10s of 1000s of rules and many knowledge bases, the median time pathology experts take to add a rule and exclude past cases is 78secs [15]. A number of companies use RDR for various applications including, IBM where RDR are used for data cleansing [16].

Richards provides a detailed review of a number of different RDR techniques, but all with the key features outlined above [17]. In the experiments below simple Single Classification RDR (SCRDR) is used. In SCRDR the knowledge base is constructed as a binary tree, with a rule at each node and the classification provided by the last satisfied rule. A new rule added to give the correct classification for a case is added at the end of the rule evaluation path for that case. Since it is then the last satisfied rule, the new rule will thus provide the conclusion for the case, rather than the previous rule that gave the wrong classification. If no rule had previously fired, the new rule will be added at the end of the all-false branch and is only evaluated if no other previous rule is satisfied by the data. Generally most rules are added to the end of the all-false branch giving a very unbalanced tree. There are a number of other RDR structures, but all with the same essential features as SCRDR [17]. The most common structure used industrially is so-called Multiple-Classification RDR where an n-ary tree is produced; however SCRDR is very simple and appropriate for the studies here using standard datasets where cases have only a single label.

We should emphasize that the use of RDR or any particular version of RDR is not essential to the idea of using a domain expert in disagreement-based semi-supervised learning; however, RDR is a very convenient way of building a knowledge base case by case as required in our proposed method. Secondly, data from many knowledge bases in industrial use show that an expert can add a rule to an RDR system in a couple minutes (median 78 secs), making the approach we suggest quite feasible for industrial use [15]. Thirdly, RDR also leads to more consistent labeling, as the expert is required to distinguish the current case from previous cases given a different label [18].

3.5 Machine Learning

The machine-learning algorithm used was J48 from the WEKA machine tool bench [19] which is an open-source implementation of the C4.5 algorithm. The reason for using J48 or C4.5 is that a decision tree learner proceeds by selecting the feature which best partitions the data into subsets and then recursively splits the subsets by further feature selection until a subset contains a single class, or there is no further information gain from any feature. This contrasts significantly with RDR which for each case writes a rule to correctly classify that case. That is, the rule is an attempt to specify the particular pattern of which that case is an instance. Since we are seeking to exploit disagreement between the two knowledge bases, we expect that the two approaches of dealing with a class at a time versus selecting features to best separate classes as whole, may be useful in identifying disagreement. There are a number of machine-learning algorithms which learn RDR, including RIDOR from WEKA based on Induct[12] but it seemed more appropriate to use a learning method with perhaps more chance of disagreement. We used the default pruning settings for J48.

4 Results

The essential feature of the proposed method is that a sequence of cases is processed and the (simulated) expert consulted to check any case where the manually developed rules assign a different label to the case than the knowledge base developed using J48, or if neither has assigned a label. If both knowledge bases agree on the label, that label is used for the case regardless of whether it is right or wrong.

In the following experiments we also evaluated various scenarios in which the simulated expert labels and provides rules for some initial cases independent of whether there is a difference between the expert rules and the J48 knowledge base. This provides some priming for the learning. One strategy is to require the expert to check a number of initial cases for each rule, before using those cases for learning.

If the rule misclassified a case, another rule would be added as above. The idea is that each rule represents a certain data pattern and you need a few correct examples of each data pattern before J48 can be expected to learn that pattern. Fig. 1 shows the errors on the test data for no data initially labeled and requiring the expert to see either 5 or 10 cases which have been covered by a each rule before using cases covered by that rule for learning. Fig. 1 and the following figures also show other comparative data. The red line shows the error from the RDR rules on the test data for the different initial priming strategies. The orange line shows the J48 error on the test data when the only data used for learning are the cases actually referred to the simulated expert for checking. These will be correctly labeled, but there are far fewer than when the automatically labeled cases are also included. The grey line shows the J48 error on test data when the training data has the same number of cases as referred to the expert, but selected randomly from correctly labeled training data. The final point on this grey line is when all training data with the correct labels are used for learning. The green line also uses all the training data, but with the labels assigned during the semi-supervised learning process. I.e. some of the labels will have been correctly assigned because of disagreement between the knowledge bases, or because neither can classify the case, but the majority will have be automatically assigned because both knowledge bases agreed on the classification, regardless of whether their agreed classification was right or wrong. Finally the blue line is the number of cases referred to the expert for label assignment and therefore also the number of cases used for the machine leaning experiments shown by the orange and grey lines.

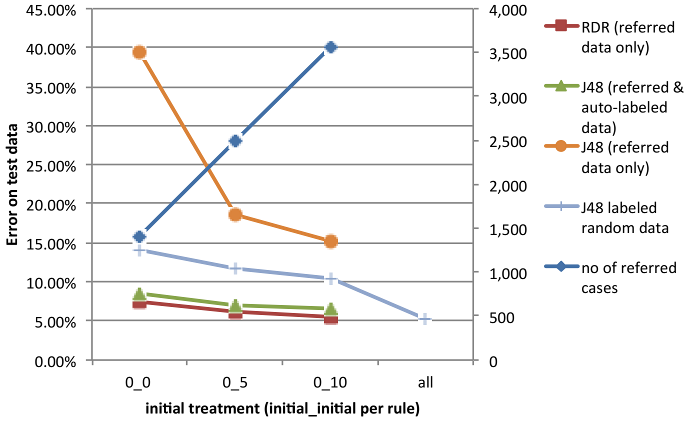

Fig. 1. Inductive learning results for different initial labeling of cases

In Fig. 1 the RDR errors are similar to but slightly less than the J48 errors using all the data labeled by the experimental protocol, no doubt because J48 is likely to prune small classes. Because the RDR errors are smaller we tend to focus on these as the best measure of the proposed method in the following discussion. Using an initial 10 cases per rule labeled by the expert and then cases referred by the method (0_10 on the figure) Fig. 1 shows that the RDR achieve an accuracy (5.44% ± 1.48%,) on the test data. This is not significantly different (P<.0.05) from the accuracy (5.13% ± 0.15%) using all the (correctly labeled) training data (21,178 cases). The method may seem to require a lot of training data with 3,558 cases labeled by the expert, but for this difficult dataset when 3,558 correctly labeled cases are randomly selected from the training data, there is a much higher error of 10.40 ± 0.52%. Even if no initial cases are referred to the expert (0_0), and the flagging of cases to be manually labeled by expert depends entirely on the semi-supervised method the RDR error is only 7.49% ± 1.99%, with 1,410 cases seen. Applying J48 to a random selection of 1,410 correctly labeled cases results in much greater errors, 14.09% ± 0.91%. Clearly using the proposed method to select the cases to be labeled by a domain expert gives much better results than the expert labeling a random selection of data. It is interesting to note that using only the actual referred cases as training data gives much worse errors than randomly selecting the same number of cases. This is no doubt because the method tends to select more anomalous cases rather than cases representing the population distribution. Even requiring the first 5 or 10 cases seen by each rule to be correctly labeled by the expert is unlikely to produce the same distribution of cases as in this very unbalanced dataset as a whole.

We applied a further priming strategy of requiring an overall initial number of cases to be labeled by the expert as well as the initial cases per rule strategy. Fig. 2. shows the results using this approach. The most obvious difference from Fig. 1 is that using the actual referred cases produces a much better result for larger numbers of initial referrals as the initial selection is more likely to represent the population distribution. For more than 1000 initial cases and 10 initial cases per rule, and then using the method for further cases, J48 performance using only referred cases exceeds selecting the same number of correctly labeled cases randomly. RDR accuracy exceeds the accuracy of using all the training data labeled, from group 1000_5 upwards. Again we assume the reason is that RDR is likely to be better for patterns with small number of representatives. A rule can be written for a single case, but J48 learning requires sufficient examples for each pattern.

The overall pattern of results for the Garvan data is shown in Table 1. In this table the numbers of labeled training cases are shown in increasing order in the first column. The second column shows the J48 errors (± 1SD) on the 21,736 test cases when the correctly labeled training cases were randomly selected. The bottom entry in this column is the error when J48 was used on all the training data. The third column shows the protocol used in semi-supervised learning. E.g. 5_500 means that the first 500 cases were all correctly labeled by the expert and if necessary rules added, then the first 5 cases seen by each rule were correctly labeled and if necessary rules were added, and finally any cases flagged by the semi-supervised method were correctly labeled and if necessary rules were added. The total number of cases in column 1, used in each of the experiments where chosen from the number of cases labeled by the expert both initially and by the semi-supervised process specified in column 2. The fourth column shows the J48 error on test data using all the training cases from the semi-supervised learning, i.e. cases actually referred to the expert (numbers in column 1) and also the cases automatically labeled during the semi-supervised process because the J48 and

RDR knowledge bases agreed. The final column shows the error of the RDR knowledge base on the test data. The RDR were built by the expert adding a rule when a case referred for labeling was not given the correct label by the RDR knowledge base.

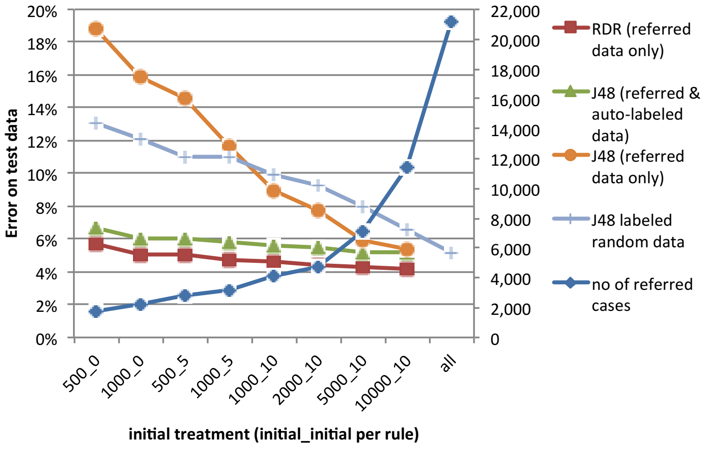

Fig. 2. Inductive learning results for different initial labeling of cases

Table 1. Summary of error against number of expert-labeled training data

No of training cases	J48 error (training cases randomly selected)	Initial cases (semi-supervised)	J48 error (semi-supervised training cases)	RDR error (semi-supervised training cases)
1,410	14.09% ± 0.91%	0_0	8.44% ± 2.22%	7.49% ± 1.99%
1,765	11.78% ± 0.57%	0_500	7.03% ± 1.78%	5.71% ± 0.42%
2,209	10.97% ± 0.46%	0_1000	6.51% ± 1.34%	5.09% ± 0.21%
2,500	13.07% ± 0.67%	5_0	6.65% ± 0.44%	6.14% ± 1.86%
2,831	11.02% ± 0.48%	5_500	6.02% ± 0.24%	5.05% ± 0.28%
3,169	10.40% ± 0.39%	5_1000	6.00% ± 0.27%	4.78% ± 0.21%
3,558	12.13% ± 0.524%	10_0	5.77% ± 0.15%	5.44% ± 1.48%
4,153	9.97% ± 0.61%	10_1000	5.65% ± 0.13%	4.63% ± 0.17%
4,768	9.24% ± 0.35%	10_2000	5.49% ± 0.13%	4.44% ± 0.12%
7,159	7.95% ± 0.27%	10_5000	5.21% ± 0.12%	4.25% ± 0.12%
11,365	6.62% ± 0.20%	10_10000	5.19% ± 0.14%	4.15% ± 0.11%
21,178 (all)	5.13% ± 0.15			

Table 1 shows that regardless of the number of correctly labeled training cases, the semi-supervised protocol always produces smaller (J48 or RDR) errors than randomly selecting the same number of correctly labeled training cases. Also the RDR knowledge base always has a smaller error than the J48 knowledge base, and the results for the last four protocols were smaller than J48 learning from all the training data. Even for 500 initial cases and 5 per rule (5_500), with a total of 2,831 expert-labeled training cases the error is smaller than J48 learning from all 21,178 cases correctly labeled by the expert. This is no doubt because an RDR rule can be written for a single instance of a data pattern where J48 needs a number of examples – a problem with this very unbalanced dataset, even with very large numbers of training examples. Using the method to label only 10.4% of the cases (the 0_1000 protocol) gives the same accuracy as the expert labelling all the data, while randomly selecting the same number of cases of cases doubles the error.

Table 2. Summary of error against number of expert-labeled training data for various datasets

Dataset	Total Training data nos.	J48 error using total training data	Partial training data nos.	J48 errors using random partial data	RDR errors using selected partial data
mushroom	4062	0.0%	23 (0.6%)	12.9%	0.8%
vote	211	5.0%	14 (6.6%)	9.6%	5.0%
breast-w	342	4.9%	24 (7%)	10.0%	3.7%
breast cancer	109	25.2%	8 (7.3%)	31.5%	25.9%
iris	73	8.0%	9 (12.3%)	30.7%	17.3%
pima	322	23.2%	41(12.7%)	27.9%	19.8%
diabetes	321	22.1%	42 (13.1%)	26.3%	19.0%
balance scale	282	19.8%	42 (14.9%)	30.0%	18.8%
ionosphere	174	11.4%	28 (16.1%)	23.9%	5.1%
soya bean	328	11.7%	81 (24.7%)	32.5%	10.2%
liver (BUPA)	146	32.4%	38 (26.0%)	41.0%	26.6%
sonar	101	29.8%	37 (36.6%)	37.5%	12.5%

We also investigated a number of other datasets as shown in Table 2. The same semi-supervised strategy has been used for each dataset and for simplicity does not include any initial referral of cases to the expert as this may vary between datasets. The datasets shown in the first column all come from the UCIrvine data repository. 50% of the data was used for training and 50% for testing and again for each of the 10 repeat studies the data was randomised. The total numbers of training data are shown in the second column, which excludes the cases pruned when building the simulated expert, as discussed above. The third column is the error on test data using a J48 knowledge base built on all the training data. The fourth column shows the numbers of training cases referred to the expert for labeling by the method, and also as a percentage of the total training data. The fifth column shows errors on test data using a J48 knowledge base built using the same number of training cases as in column four, selected randomly from the training data correctly labeled. The sixth column shows errors on test data for the RDR knowledge base built by the semi-supervised process.

The errors using the RDR knowledge bases are closely similar to or smaller than the error using J48 with all the labeled training data for every dataset except Iris. The semi-supervised method also requires far fewer cases to be labeled to achieve the same results. Across the 12 datasets an average of 14.8% of the data needed to be labeled by an expert using the semi-supervised process. For the Iris dataset, the error was worse for the RDR system than for J48 with all the training data labeled (12.3% vs 8.0%). When we used an initial priming for the Iris dataset of requiring the first 2 cases seen by each rule to be referred to the expert, the RDR error dropped to 4.0 %, with the number of cases the expert needed to see rising from 9 to 11. We used a number of initial priming strategies for all the datasets, but only the Iris dataset required this to achieve the same error as using all the training data.

5 Discussion

We have described a technique for semi-supervised learning which identifies data to be labeled by a domain expert: A domain expert and machine learning are used to build two knowledge bases to classify unlabeled cases one by one. If both agree on the classification for that data, that classification is accepted as the label and the case is added to the labeled training data and the machine learning knowledge base rebuilt. Otherwise the domain expert is asked to provide the correct label and if necessary to add a rule to correctly classify that case. The now correctly labeled case is added to the training data, the machine learning knowledge base rebuilt and more unlabeled cases processed.

As shown with a number of datasets, this approach greatly reduces the amount of data need to be labeled by a domain expert to produce a knowledge base of the same accuracy as the expert labeling all the data; for some datasets in Table 2 only a few percent of the data needed to be labeled.

Although we divided datasets into training and test data, in practice the process could easily be ongoing, as industrial evidence shows it only takes a couple of minutes to add rule – even across thousands of rules [15]. It should also be noted that the method here introduces an extra error by training the system only on cases the simulated expert can correctly classify, but testing it on data that included cases the simulated could not correctly classify. This was done only to enable comparison with other methods' results and the error it introduces would not occur in an industrial setting.

For the Garvan and Iris datasets, as well as the data identified by the semi-supervised method, some initial data had to be seen by the domain expert in order to produce the same accuracy as labeling all the data. It is only with hindsight that we know that this was not necessary for other domains. For new unknown domains it would probably be important to have some sort of initial labeling strategy. Other strategies warrant investigation, for example such as having less confidence in nodes of the J48 decision tree where cases have been pruned. This would enable a probabilistic decision to be made about the number of cases to be initially checked manually.

Comparing our results with an empirical study of various semi-supervised methods [13], our method produced more much accurate knowledge bases across the nine datasets in common (P<0.005, paired t-test). The average accuracy of our method was 89.5% ± 8.83% vs 80.5% ± 10.99% for the best results from [13] for the same datasets. 75% of the data was used for training in [13] while we used 50%, so that the fixed 10% of the training data selected for training in [13] would be equivalent to 15% of our 50% training data. The average amount labeled training data we used was 14.8% and for 8 of the 12 datasets in Table 2 less than 15% of the data needed to be labeled. Our method also produces equivalent accuracies to having all the data labeled. Very skewed datasets such as the Garvan data were excluded in [13], whereas with our method using 10.4% of the Garvan data achieved the same accuracy as having all data labeled. In general our proposed method seems to produce better accuracies for the same amount of training data than the methods included in [13]. The possibly superior performance of our method is perhaps because the method uses a significant source of extra information in the rules provided by the expert. Using a different approach [8] also identified which cases to label. With two datasets in common, our method produced a higher accuracy for fewer cases for Iris (0_2 method), while for Soyabean the method in [8] produced higher accuracy with fewer cases. Semi-supervised methods like [8] and those in [13] use all the training data in a batch mode, for example to identify clusters of unlabeled data associated with labeled instances. Our method deals with individual cases as they occur in a data stream.

The most distinctive feature of the method we propose is that when an expert labels a case they may also be required to provide a rule to classify the case. Ripple-Down Rules are a convenient way of adding rules case by case, but other methods could be used and also other machine learning methods than J48.

Acknowledgment. This research was supported by an Australian Research Council Discovery Grant.

References

1. Zhu, X.: Semi-supervised learning literature survey. TR1530. Computer Science, University of Wisconsin-Madison (2005)
2. Chapelle, O., Schölkopf, B., Zien, A.: Semi-supervised learning. MIT Press, Cambridge (2006)
3. Zhou, Z.-H., Li, M.: Semi-supervised learning by disagreement. Knowledge and Information Systems 24(3), 415–439 (2010)
4. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: Proceedings of the Eleventh Annual Conference on Computational Learning Theory, pp. 92–100. ACM (1998)
5. Goldman, S., Zhou, Y.: Enhancing supervised learning with unlabeled data. In: ICML 2000 Proceedings of the Seventeenth International Conference on Machine Learning, pp. 327–334 (2000)

6. Tur, G., Hakkani-Tür, D., Schapire, R.E.: Combining active and semi-supervised learning for spoken language understanding. Speech Communication 45(2), 171–186 (2005)
7. Zhu, X., Lafferty, J., Ghahramani, Z.: Combining active learning and semi-supervised learning using gaussian fields and harmonic functions. In: ICML 2003 Workshop on the Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining, pp. 58–65 (2003)
8. Parsazad, S., Saboori, E., Allahyar, A.: Data Selection for Semi-Supervised Learning. arXiv preprint arXiv:1208.1315 (2012)
9. Finlayson, A., Compton, P.: Run-time validation of knowledge-based systems. In: Proceedings of the seventh International Conference on Knowledge Capture, pp. 25–32. ACM (2013)
10. Dazeley, R., Park, S.S., Kang, B.H.: Online knowledge validation with prudence analysis in a document management application. Expert Systems With Applications 38(9), 10959–10965 (2011)
11. Horn, K., Compton, P.J., Lazarus, L., Quinlan, J.R.: An expert system for the interpretation of thyroid assays in a clinical laboratory. Aust. Comput. J. 17(1), 7–11 (1985)
12. Gaines, B., Compton, P.: Induction of Ripple-Down Rules Applied to Modeling Large Databases. Journal of Intelligent Information Systems 5(3), 211–228 (1995)
13. Guo, Y., Niu, X., Zhang, H.: An extensive empirical study on semi-supervised learning. In: 2010 IEEE 10th International Conference on Data Mining (ICDM), pp. 186–195. IEEE (2010)
14. Compton, P., Preston, P., Kang, B.: The Use of Simulated Experts in Evaluating Knowledge Acquisition. In: Gaines, B., Musen, M. (eds.) Proceedings of the 9th AAAI-Sponsored Banff Knowledge Acquisition for Knowledge-Based Systems Workshop, Banff, Canada, pp. 12.11–12.18. University of Calgary (1995)
15. Compton, P., Peters, L., Lavers, T., Kim, Y.-S.: Experience with long-term knowledge acquisition. Paper Presented at the Proceedings of the Sixth International Conference on Knowledge Capture, KCAP 2011, Banff, Alberta, Canada, pp. 49–56. ACM (2011)
16. Dani, M.N., Faruquie, T.A., Garg, R., Kothari, G., Mohania, M.K., Prasad, K.H., Subramaniam, L.V., Swamy, V.N.: Knowledge Acquisition Method for Improving Data Quality in Services Engagements. In: IEEE International Conference on Services Computer (SCC), Miami, pp. 346–353. IEEE (2010)
17. Richards, D.: Two decades of Ripple Down Rules research. The Knowledge Engineering Review 24(2), 159–184 (2009)
18. Wang, J.C., Boland, M., Graco, W., He, H.: Use of ripple-down rules for classifying medical general practitioner practice profiles repetition. In: Compton, P., Mizoguchi, R., Motoda, H., Menzies, T. (eds.) Proceedings of Pacific Knowledge Acquisition Workshop PKAW 1996, Coogee, Australia, pp. 333–345 (1996)
19. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: an update. ACM SIGKDD Explorations Newsletter 11(1), 10–18 (2009)