EXPERIMENTS WITH SATURATION FILTERING FOR NOISE ELIMINATION FROM LABELED DATA

Borut Sluban (1), Nada Lavrač (1), Dragan Gamberger (2), Andrej Bauer (3)
(1) Jožef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia
(2) Rudjer Bošković Institute, Bijenička 54, 10000 Zagreb, Croatia
(3) Faculty of Mathematics and Physics, Jadranska 21, 1000 Ljubljana, Slovenia
e-mail: borut.sluban@gmail.com

ABSTRACT

For classification tasks, data filtering is aimed at improving prediction accuracy. This paper presents some upgrades of the existing saturation filter, which is used for the elimination of noisy examples from labeled data. The main contribution of this paper is the exhaustive experimental evaluation of saturation filtering. The filter was applied on 12 different datasets and its performance was tested with four different learning algorithms. The evaluation showed the improvement of prediction accuracy on most of the datasets and statistical testing proved that the performance of a learning algorithm was significantly better with the use of the saturation filter than without it.

1 INTRODUCTION

In predictive data mining learning methods are used to induce models or theories from class-labeled data. The induced models are used for classification or prediction. In a classification task, data is usually formed from examples which are labeled by the class to which they belong. The task is to find a model that will enable a newly encountered instance to be classified.

A common problem affecting the prediction accuracy of an induced model is noise in the training data. Noise usually means outliers and random errors in training examples (erroneous attribute values and/or erroneous classification). Therefore appropriate noise handling procedures are essential to ensure the prediction accuracy and applicability of induced models.

One of the approaches to noise handling is filtering of noisy examples from the training dataset. In this way the model induced from the filtered data will be less complex and more accurate when classifying unseen examples. The upgrade of a filtering algorithm, as well as the experimental work presented in this paper are based on the *saturation filter* introduced in [3].

The paper is structured as follows. In Section 2 the theoretical background and the idea of the saturation filter is briefly described. Section 3 describes the reimplementation of the saturation filter. An overview of the testing results is given in Section 4. The paper concludes with some directions for further work in Section 5.

2 SATURATION FILTER

The idea of the saturation filter, introduced in [3], follows the Occam's razor principle, which suggests that among all the hypotheses (models) that are correct for all the training examples one should select the simplest hypothesis. This principle is implicitly used for noise elimination, since by choosing a simpler hypothesis we try to avoid overfitting a noisy training dataset.

A *target concept* of a given problem domain is defined as the source of all possible correct examples of the concept. The task of inductive learning is to find a good representation of the target concept in a selected hypothesis language. This representation is called a *target theory*. By following the Occam's razor principle and selecting the simplest hypothesis which correctly represents the target concept we get the so-called *target hypothesis*. Elimination of noisy examples from the training set helps in the induction of the target hypothesis, since a hypothesis induced from noiseless data will be less complex and more accurate when classifying unseen examples.

The name for the filter was derived from the saturation property of training data, meaning that a training dataset is saturated, if it can be used to induce a target theory. The approach to obtain a saturated training set was based on the observation that the elimination of noisy examples, in contrast to the elimination of examples for which the target theory is correct, reduces the *CLCH* value of the training set, where *CLCH* stands for the *Complexity of the Least Complex correct Hypothesis*.

Suppose that a complexity measure c for a hypothesis language is defined and that for any hypothesis H its complexity $c(H)$ can be determined. Than for a training set E one can determine the complexity of the least complex hypothesis which is correct for all the examples in E: this complexity, denoted by $g(E)$ is called the *CLCH* value.

In [3] it was shown that if E is noiseless and saturated (containing enough training examples to be able to induce a correct target hypothesis from it), then – if a noisy example e_n is added to E – it follows that $g(E) < g(E_n)$, where $E_n = E \cup \{e_n\}$ and e_n is a noisy example for which the target hypothesis is not correct. The property $g(E) < g(E_n)$ means that a noisy example in E_n can be detected as the one that enables *CLCH* value reduction. The approach in an iterative form is applicable also when E_n includes more than one noisy example.

In this way noisy examples can be detected and eliminated from the training set. But it must be noted that although the saturation property of a training set is the main theoretical condition for the mentioned filter, in practice it is usually not possible to satisfy the saturation condition. Despite that, the filtering algorithm is still applicable since there still may be enough training examples that form a saturated subset for some subconcept of the target concept.

3 IMPLEMENTATION

The reimplementation of the saturation filter and the testing of its performance were made with the help of the open source software for data mining called *Orange* [2]. It is software for data mining through visual programming or Python scripting, it includes many modules and tools for data preprocessing, modeling and knowledge discovery in databases.

For the elimination of noisy training examples with the saturation filter we had to choose a complexity measure that would distinguish between models induced from training data. The idea was to build an *unpruned* decision tree with machine learning tools available in Orange. Since an unpruned decision tree is a classification model which is correct for all the training examples, we chose the number of nodes of the unpruned decision tree as the complexity measure of the model.

To construct the saturation filter we needed two methods. The first one was a *saturation test*. At first it computes the complexity of the classification model for the given training set, then in each step it excludes only one training example and computes the complexity of a classification model induced from the rest of the training examples. If the complexity of the new model is smaller than the one computed in the beginning, then the excluded example is marked as potentially noisy. However at this point it is not yet finally excluded, it is returned to the training set and the same procedure is repeated for each training example. The examples which had the most effect in reducing the complexity of the classification model with their exclusion are labeled as the *most noisy* and are passed on to the second method. If there is no such example, then the training set is considered to be saturated.

The second method, the *filter*, randomly chooses one among the most noisy examples and finally excludes it from the training dataset, while the others are returned back. This is repeated as long as the saturation test finds noisy examples, meaning that a saturated subset has not yet been obtained.

While testing the filter on different datasets, in some cases the number of examples suggested as being noisy by the saturation test was quite high (relative to the size of the training set). In this case the filter excluded them all one by one. To avoid excluding a subset of examples that represent a small subconcept of the target concept, an addition to the saturation filter was introduced as the tolerance level parameter *t*. If the parameter is specified the filtering process stops when the size of the set of potentially noisy instances in an iteration of the saturation test exceeds the given percentage *t* of the size of the training set.

4 EVALUATION OF RESULTS

The experimental work done with the previously described reimplementation of the saturation filter and the results obtained from the testing are presented in this section.

4.1 Datasets used for testing

The datasets used for testing the performance of the saturation filter were obtained from the Orange website and are mostly from the *UCI Machine Learning Repository* [1]. For simplicity only 2-class datasets were chosen. The first eight datasets are real-life datasets containing noise, whereas the last four are artificially generated (precise representations of a concept or a set of all possible configurations of a concept) and do not contain any noise (Table 1).

Dataset	No. of examples	No. of attributes	Class ratio	Description
Breast Cancer LJ	286	9	70:30	Breast cancer of patients in Ljubljana, 1988.
Breast Cancer WI	683	9	65:35	Breast cancer of patients in Wisconsin, USA, 1991.
BUPA	365	6	58:42	Liver disorder, male patients.
Credit Approval	690	15	56:44	Approval of credit applications.
Heart Disease	303	13	54:46	Presence of heart disease in the patient.
MONK-3	554	6	52:48	Target concept: ($A_5 = 3$ and $A_4 = 1$) or ($A_5 \neq 4$ and $A_2 \neq 3$).
SA - Heart	462	9	65:35	Coronary heart disease, male patients, SAR
Voting	435	16	61:39	Congressional voting records. USA, 1984.
MONK-1	556	6	50:50	Target concept: $A_1 = A_2$ or $A_5 = 1$.
MONK-2	601	6	66:34	Target concept: exactly two of the attributes have the value 1.
Promoters	106	57	50:50	Promoter DNA sequences of the bacteria E. Coli.
Tic Tac Toe	985	9	65:35	All possible endgame configurations of the game Tic Tac Toe.

Table 1: *Datasets used for testing the performance of the saturation filter.*

4.2 Algorithms used in the experimental evaluation

For testing the performance of the saturation filter, four learning algorithms from the Orange library were used for the induction of classification models. These are:

- Decision tree learner (unpruned)
- Decision tree learner (pruned)
- Naïve Bayes classifier
- Rule learning algorithm (CN2)

4.3 Classification accuracies of the basic saturation filter

Classification accuracies were computed using 10-fold cross-validation for all the four classifiers induced from non-filtered and filtered datasets.

The testing results are presented in Table 2. Among the first eight datasets filtering with the saturation filter showed to achieve better classification results on:

- four datasets with all four learning algorithms
- one dataset with three learning algorithms
- two dataset with two learning algorithms
- one dataset with one learning algorithm

However on two domains (Breast cancer LJ and SA-heart) no learning algorithm except the Naïve Bayes classifier managed to outperform the default classifier.

The classification accuracies on the last four datasets which were artificially generated, were expected to be worse if filtering were applied (since removing non-noisy examples would result in the induction of a less accurate classification model). These expectations were fully confirmed on two datasets, but on the other two dataset there were (a bit surprisingly) some improvements with one or two learning algorithms. However, the Naïve Bayes classifier did not reach the classification accuracy of the default classifier on the MONK-2 dataset.

It is also interesting to compare the third and the fourth column in Table 2, where we can see that the combination of the saturation filter and the unpruned decision tree learner performs comparable to (in some cases even better than) the decision tree learner which has its own built-in noise handling procedure, called pruning. However, according to statistical *t-test*, the differences are due to large standard deviations in most cases insignificant.

4.4 Improvements with tolerance level parameter *t*

In addition to the reimplementation of the original saturation filter described in [3], we have proposed its improvement using tolerance level parameter *t*, whose goal is to prevent the exclusion of certain subsets which could possibly represent a small subconcept of the target concept. The application of the parameter makes sense only in the case of datasets where the saturation test seems to find "larger" sets of potentially noisy examples.

Parameter values used for testing were 0.015, 0.02, 0.025 and 0.03, which means that the filtering process stops if the size of the set of potentially noisy examples in an iteration of the saturation test exceeds the given percentage *t* of the size of the training set. In Table 3 the improved classification accuracies are shown in bold. The values listed are the highest classification accuracies achieved, mostly achieved by setting *t* value to 0.015 or 0.02.

Dataset	**Unpruned decision tree**		**Pruned decision tree**		**Naïve Bayes Classifier**		**Rule learning alg. (CN2)**	
	non-filtered	filtered	non-filtered	filtered	non-filtered	filtered	non-filtered	filtered
BreastCancerLJ	61.9 (±11.4)	67.1 (±6.74)	67.2 (±9.36)	67.1 (±6.74)	72.7 (±3.06)	73.1 (±4.81)	67.9 (±6.05)	67.2 (±5.11)
BreastCancerWI	94.3 (±2.99)	93.8 (±2.78)	94.3 (±2.98)	93.7 (±2.64)	96.8 (±2.16)	96.5 (±1.89)	94.7 (±2.10)	95.9 (±2.16)
BUPA	60.5 (±6.54)	63.4 (±6.80)	60.5 (±6.54)	63.4 (±6.80)	67.0 (±5.76)	67.3 (±6.70)	60.5 (±9.45)	66.1 (±6.32)
Credit Approval	83.8 (±3.60)	85.1 (±2.83)	85.4 (±3.07)	85.5 (±3.11)	85.7 (±2.78)	85.8 (±3.09)	75.2 (±3.46)	73.8 (±7.05)
Heart Disease	65.4 (±7.13)	70.0 (±10.8)	65.7 (±7.56)	69.3 (±10.9)	82.8 (±8.30)	83.2 (±7.48)	68.6 (±11.5)	69.3 (±8.02)
MONK-3	96.6 (±2.84)	97.8 (±2.26)	98.9 (±2.18)	98.9 (±2.18)	96.4 (±3.04)	96.4 (±3.04)	88.6 (±8.89)	91.0 (±9.56)
SA - Heart	58.6 (±8.53)	60.4 (±6.66)	58.6 (±8.53)	60.4 (±6.66)	69.2 (±7.58)	69.5 (±7.19)	56.7 (±9.76)	60.0 (±7.91)
Voting	83.9 (±3.67)	96.1 (±3.90)	96.1 (±4.30)	96.3 (±3.78)	90.1 (±4.78)	93.3 (±4.87)	93.6 (±3.94)	95.4 (±3.72)
MONK-1	98.4 (±2.03)	100.0 (±0.00)	98.4 (±2.03)	100.0 (±0.00)	74.6 (±6.07)	74.6 (±6.07)	90.5 (±7.65)	86.3 (±8.68)
MONK-2	76.0 (±4.12)	66.2 (±5.83)	73.2 (±3.67)	66.0 (±5.79)	62.4 (±3.06)	61.9 (±3.23)	92.0 (±2.76)	86.3 (±5.11)
Promoters	81.3 (±6.84)	83.7 (±8.99)	83.1 (±6.74)	82.7 (±10.8)	85.9 (±10.5)	85.8 (±9.79)	71. 7 (±13.7)	72.0 (±13.1)
Tic Tac Toe	86.5 (±5.14)	85 7 (±3.54)	86.4 (±5.24)	85.7 (±3.54)	70.3 (±5.75)	71 1 (±5 47)	86.5 (±3.13)	84.5 (±4.90)

Table 2: *Classification accuracies of four different learning algorithms on 12 different datasets*

Dataset	**Unpruned decision tree**		**Pruned decision tree**		**Naïve Bayes Classifier**		**Rule learning alg. (CN2)**	
	non-filtered	filtered	non-filtered	filtered	non-filtered	filtered	non-filtered	filtered
BreastCancerLJ	61.9 (±11.4)	67.1 (±6.74)	67.2 (±9.36)	**68.9** (±8.69)	72.7 (±3.06)	73.1 (±4.81)	67.9 (±6.05)	**68.9** (±3.41)
BreastCancerWI	94.3 (±2.99)	**95.2** (±3.43)	94.3 (±2.98)	**95.2** (±3.36)	96.8 (±2.16)	96.5 (±1.89)	94.7 (±2.10)	**96.1** (±2.16)
SA - Heart	58.6 (±8.53)	60.4 (±6.66)	58.6 (±8.53)	60.4 (±6.66)	69.2 (±7.58)	**69.7** (±6.80)	56.7 (±9.76)	**60.8** (±9.65)
Promoters	81.3 (±6.84)	**88.5** (±5.91)	83.1 (±6.74)	**86.6** (±6.51)	85.9 (±10.5)	85.8 (±9.79)	71.7 (±13.7)	**72.8** (±7.29)

Table 3: *Improved classification accuracies (bold) by the use of tolerance lever parameter t.*

4.5 Statistical evaluation of the results

To compare the performance of two learning algorithms on several datasets the *Wilcoxon signed-rank test* is used. In this case we wanted to see if the classification accuracy results obtained by the combinations of saturation filter and learning algorithm were statistically significantly better than the classification accuracy results obtained from only the learning algorithm.

First, the Wilcoxon signed-rank test was performed for the two algorithms on all the 12 datasets presented in Table 1, in order to get an idea how the filter performs on representative set of datasets. But since we did not expect improvements of the classification accuracy when applying the filter on the last four datasets which do not contain noise, we made the Wilcoxon test also separately for the first eight datasets which contain noise. At the end the test was made once more on these same eights datasets, but with the classification accuracies obtained by using the saturation filter with tolerance level parameter *t*.

The results from these three Wilcoxon signed-rank tests made on two different sets of datasets are presented in Table 4. The table has to be understood in the following way: the algorithm combining the saturation filter and the learning algorithm is better than the algorithm without the filter, with probability *p* stated in the table.

Datasets	Unpruned decision tree learner	Pruned decision tree learner	Naïve Bayes classifier	Rule learning algorithm (CN2)
All (8+4)	**0.0499**	0.6101	0.1579	0.8753
Only noisy (8)	**0.0173**	0.1415	**0.0499**	0.0929
Noisy (8) (with param. *t)*	**0.0117**	**0.0117**	**0.0423**	**0.0423**

Table 4: *p-values obtained from the Wilcoxon signed-rank test (in the case where the combination of the saturation filter and the learning algorithm is better than only the learning algorithm).*

By choosing statistical significance level $\alpha = 0.05$, we see from Table 4, that the application of the saturation filter on all 12 datasets yielded statistically significantly better results only while using the unpruned decision tree learner (bold in Table 4). If the Wilcoxon test is made only on the eight noisy datasets, then the filtering showed to be statistically significantly better with the use of the unpruned decision tree learner and the naïve Bayes classifier.

The most interesting Wilcoxon signed-rank test results however, for statistical significance level $\alpha = 0.05$, were obtained in the case where the saturation filter with tolerance level parameter *t* was used. The combination of the saturation filter with tolerance level parameter *t* and all four learning algorithm showed to be statistically significantly better than only the learning algorithms with no filtering applied.

5 CONCLUSION

Considering the experimental test results we can conclude that our implementation of the saturation filter combined with all four different learning algorithms results in improving the classification accuracy by previous filtering of training data. The use of the filter shows improvements on most datasets, and with the additional use of tolerance level parameter *t* the improvement on even few more datasets can be observed. Finally, the Wilcoxon signed-rang test shows statistical significance of the improvement obtained by the use of the saturation filter.

At the negative side, however, the implemented algorithm is rather slow, due to its iterative process of searching for noisy training examples. Therefore, the current implementation is practically suitable only for datasets of the size up to 1000 instances (or a few 1000s). Another shortcoming of the current implementation is the complexity measure used in the filtering process: it might be better to use a more sensitive complexity measure that could better distinguish between noisy and non-noisy training examples.

Improvements of these two shortcomings, along with testing on more datasets and testing with more learning algorithms, is the subject of further work, which could make the saturation filter more widely applicable and make it perform even better.

Acknowledgements

The authors are grateful to Petra Kralj Novak for helpful advice on the implementation in Orange, and Dragi Kocev for help with the statistical evaluation of the results.

References

[1] A. Asuncion, D.J. Newman, UCI Machine Learning Repository, http://www.ics.uci.edu/~mlearn/MLRepository.html, 2007.

[2] J. Demšar, B. Zupan, G. Leban, Orange: From Experimental Machine Learning to Interactive Data Mining, http://www.ailab.si/orange, 2004, Faculty of computer and information science, Univerza v Ljubljani.

[3] D. Gamberger, N. Lavrač, Conditions for Occam's Razor Applicability and Noise Elimination, Lecture Notes in Artificial Intelligence: Machine Learning: ECML-97 (M. Van Someren, G. Widmer, ur.), vol. 1224, Springer-Verlag, 1997, str. 108-123.

[4] B. Sluban, Saturation filter for noise elimination from labeled data, Diploma Thesis, Faculty of Mathematics and Physics, University of Ljubljana, 2009.