# Experiments in Meta-level Learning with ILP

Ljupčo Todorovski[1,2] and Sašo Džeroski[2]

[1] Faculty of Medicine, Institute for biomedical informatics
Vrazov trg 2, 1000 Ljubljana, Slovenia
Ljupco.Todorovski@mf.uni-lj.si
[2] Jožef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia
Saso.Dzeroski@ijs.si

**Abstract.** When considering new datasets for analysis with machine learning algorithms, we encounter the problem of choosing the algorithm which is best suited for the task at hand. The aim of meta-level learning is to relate the performance of different machine learning algorithms to the characteristics of the dataset. The relation is induced on the basis of empirical data about the performance of machine learning algorithms on the different datasets.
In the paper, an Inductive Logic Programming (ILP) framework for meta-level learning is presented. The performance of three machine learning algorithms (the tree learning system C4.5, the rule learning system CN2 and the k-NN nearest neighbour classifier) were measured on twenty datasets from the UCI repository in order to obtain the dataset for meta-learning. The results of applying ILP on this meta-learning problem are presented and discussed.

## 1 Introduction

In the area of machine learning a large number of different algorithms have been developed. When considering new datasets for analysis using these algorithms, the problem of choosing the most suitable one(s) occurs. The choose of the appropriate machine learning algorithm, can be especially time consuming in the process of knowledge discovery in very large datasets. This problem can be solved using meta-level machine learning, i.e. by learning to predict how well each of the machine learning algorithms can perform on the dataset on the basis of the dataset itself. Using this predictor, users can discard algorithms that are not suitable for the dataset at hand and save a lot of effort in trying out all the algorithms.

The concept relating the performances of different machine learning algorithms to the characteristics of the datasets can be induced from empirical data using an arbitrary machine learning algorithm. The empirical data contain information about the performance of different machine learning algorithms on some set of datasets. In state-of-the-art meta-learning studies the concept is induced using attribute oriented machine learning algorithms for rule induction [1,3]. In order to use such algorithms a fixed set of attributes describing the datasets have

J.M. Żytkow and J. Rauch (Eds.): PKDD'99, LNAI 1704, pp. 98–106, 1999.

to be chosen. This set usually includes statistical and information-theory measures [3]. The description of the datasets using a fixed set of attributes can be problematic in several ways. First, the characteristics of the dataset are always measured for the whole dataset only, which is much less informative than using measures for individual attributes. This lack of information can be partly compensated using some advanced measures for the distribution of the data in the dataset. However, calculating these measures can be more complex then actually applying some of the machine learning algorithms to the dataset at hand.

Using more powerful formalisms for dataset description can be a way to surpass these problems. In the paper, we introduce an Inductive Logic Programming (ILP) framework for meta-learning. This framework includes some of the measures used in previous state-of-the-art meta-level learning studies. But it also extends the possibilities of describing datasets with the possibility of including statistical and information-theory measures for parts of the dataset (for each attribute and example) and not only the dataset as a whole. Using ILP learning systems, the concept relating this extended dataset description to the performance of different machine learning algorithms can be induced. In preliminary experiments with the presented ILP framework, we used the ILP system FOIL. The performance of three classification algorithms on twenty datasets was measured and related to the dataset features.

The paper is organized as follows. The meta-learning ILP framework is introduced in Section 2. In Section 3 the preliminary results of the experiments with thirteen datasets are presented. Section 4 concludes with a discussion on related work and some directions for further work.

## 2 Meta-level Learning: An ILP Framework

In state-of-the-art meta-learning studies, such as [3] and [1], a fixed set of properties for the whole dataset are used for the dataset description. Considering the whole dataset at once in calculating the properties can be problematic because of the mixture of different types of attributes in the dataset. Some standard statistical measures, such as mean and standard deviation, are used for continuous attributes only, and other, such as median and entropy, are preferred for discrete attributes in the dataset. All measures used in the propositional formalism for dataset description should be well defined for both continuous and discrete attributes in order to calculate their averages among all the attributes in the dataset. In the recent study [6] the problem of averaging the measures among different types of attributes has been addressed. However, the propositional framework used in the study prevents the use of measures for individual attributes in the dataset.

In the ILP framework for the dataset description, summarized in Table 1, the propositional properties are used along with some properties which are calculated for each attribute in the dataset. Measures used in the framework resemble the measures used in other meta-learning studies. These include standard simple measures like number of attributes and examples, some statistical measures for

**Table 1.** Relations used for description of datasets.

| Relation | Description |
|---|---|
| dataset(D) | dataset's identification |
| attr(D,A) | attribute's identification |
| num_of_attrs(D,V) | number of attributes |
| attr_cont(D,A) | continuous attribute |
| num_of_cont_attrs(D,V) | number of continuous attributes |
| attr_disc(D,A) | discrete attribute |
| num_of_disc_attrs(D,V) | number of discrete attributes |
| num_of_bin_attrs(D,V) | number of binary attributes |
| num_of_classes(D,V) | number of classes |
| class_entropy(D,V) | entropy of class |
| num_of_examples(D,V) | number of examples |
| values(D,V) | mean number of values of discrete attributes |
| entropy(D,V) | mean entropy of discrete attributes |
| skewness(D,V) | mean skewness of continuous attributes |
| kurtosis(D,V) | mean kurtosis of continuous attributes |
| mutual_inf(D,V) | mean mutual information of class and attributes |
| perc_of_na_values(D,V) | percentage of unknown values |
| attr_values(D,A,V) | number of values of the discrete attribute |
| attr_entropy(D,A,V) | the entropy of the discrete attribute |
| attr_stddev(D,A,V) | standard deviation of the continuous attribute |
| attr_skewness(D,A,V) | skewness of the continuous attribute |
| attr_kurtosis(D,A,V) | kurtosis of the continuous attribute |
| attr_class_mutual_inf(D,A,V) | mutual information of class and attribute |
| perc_of_attr_na_values(D,A,V) | percentage of unknown values of the attribute |

continuous attributes (mean, standard deviation, skewness and kurtosis) and entropy of discrete attributes. The mutual information between attributes and class is calculated using Siverman's method [10]. Beside averages among all the attributes in the dataset, the calculations for each attribute are also used in the dataset description (the lowest part of Table 1).

## 3 Experiments

Three different propositional classification algorithms were used in the experiments: tree-learning algorithm C4.5 [8], rule-learning algorithm CN2 with $m$-estimate [4,5] and $k$-nearest neighbour ($k$-NN) algorithm [10]. These algorithms were used both for base-level and meta-level learning. For base-level learning, they were applied to twenty datasets from the UCI Repository of Machine Learning Databases and Domain Theories [7]. For meta-level learning, the three propositional algorithms as well as two ILP systems FOIL [9] and TILDE [2] were applied to the results of base-level learning, as described below.

### 3.1 Experimental Setting

The measure of performance used in the experiments is the error rate of the classifier on the unseen examples. For each learning algorithm, the error rate for each of the twenty datasets was measured using stratified 10-fold cross validation. The dataset was first partitioned into ten folds with equal sizes and similar class distributions. The average error rates on unseen examples (over the ten folds) for twenty datasets are given in Table 2.

**Table 2.** Average error rates (in percents) of three classification algorithms on twenty datasets.

| **Dataset** | C4.5 | CN2 | $k$-NN | **Dataset** | C4.5 | CN2 | $k$-NN |
|---|---|---|---|---|---|---|---|
| australian | 15.34 | 16.50 | 14.06 | bridges-td | 17.64 | 13.73 | 14.82 |
| bridges-type | 44.57 | 44.28 | 42.36 | chess | 0.33 | 0.45 | 3.47 |
| diabetes | 27.51 | 26.06 | 26.04 | echocardiogram | 33.54 | 36.63 | 28.19 |
| german | 28.90 | 25.90 | 27.00 | glass | 31.27 | 33.60 | 29.31 |
| heart | 23.31 | 22.94 | 17.41 | hepatitis | 17.90 | 17.43 | 15.38 |
| hypothyroid | 0.83 | 1.13 | 2.12 | image | 3.29 | 6.54 | 3.13 |
| iris | 5.33 | 6.68 | 6.00 | labor | 21.34 | 11.01 | 14.33 |
| lenses | 16.67 | 26.66 | 30.00 | machine | 28.19 | 31.99 | 30.59 |
| soya | 8.05 | 8.51 | 16.83 | tic-tac-toe | 14.83 | 1.77 | 4.60 |
| vote | 3.72 | 3.69 | 10.59 | zoo | 5.00 | 5.91 | 3.91 |

Additionally, the parameters for C4.5 and CN2 were optimized to minimize the error rate using 10-fold cross validation on the training data of each fold from the previous stage. Nine parts of the training data were used to build the classifier. Its error rate was then measured on the remaining part. The parameters that minimize the average error rate over the 10 folds of the training data were chosen to perform the experiment for measuring the performance of the classification algorithms on the testing data. The values of two C4.5 parameters were optimized: minimal number of examples in the leaf node (possible values from 1 to 5), and tree pruning parameter (from 0% to 100% with step 5%). In the experiments with CN2 the values of parameter $m$ (0, 0.01, 0.1, 0.2, 0.5, 1, 2, 4, 8, 16, 32, 64, 128) and rule significance level (0%, 95% and 99%) were optimized. The optimal value of the parameter $k$ (possible values from 1 to 100) in the experiments with $k$-NN classifier was chosen using leave-one-out method as described in [10].

To prepare the data for meta-level learning task, we classified the algorithms for each of the twenty datasets in two classes: applicable and inapplicable. The algorithms with low error rates were considered applicable and others were considered inapplicable. The error rate limit for classification was used as in [3]: the algorithms with error rates within the interval

$$\left[\ err_{min}, err_{min} + k \cdot \sqrt{err_{min}(1 - err_{min})/n_{test}}\ \right)$$

are considered applicable. $err_{min}$ denotes the lowest of the three error rates for the dataset, $n_{test}$ is the number of test examples and $k$ is an error margin

parameter. The classification of the algorithms for $k = 0.25$ is summarized in Table 3.

**Table 3.** Applicability of the machine learning algorithms for twenty datasets.

| **Dataset** | C4.5 | CN2 | $k$-NN | **Dataset** | C4.5 | CN2 | $k$-NN |
|---|---|---|---|---|---|---|---|
| australian | | | √ | bridges-td | | √ | √ |
| bridges-type | √ | √ | √ | chess | √ | | |
| diabetes | | √ | √ | echocardiogram | | | √ |
| german | | √ | | glass | √ | | √ |
| heart | | | √ | hepatitis | | √ | √ |
| hypothyroid | √ | | | image | √ | | √ |
| iris | √ | √ | √ | labor | | √ | |
| lenses | √ | | | machine | √ | | √ |
| soya | √ | √ | | tic-tac-toe | | √ | |
| vote | √ | √ | | zoo | √ | | √ |

We used the ILP system FOIL for the meta-level experiments, along the base-level learning algorithms. Three different datasets for meta-level learning were constructed, one for each classification algorithm. The target relations were `appl_c45(D)`, `appl_cn2(D)` and `appl_knn(D)` defined as in Table 3. All the relations from Table 1 were used as a background knowledge. The propositional learning algorithms use the attributes based on the subset of relations from Table 3 of the form `relation(D,V)`. In order to evaluate the obtained models, we used the leave-one-out method. Following this method, we used all but one examples to build a model, while the remaining example was used for testing.

### 3.2 Results of the Experiments

We used FOIL in two series of experiments. In the first one (labeled FOIL in the tables) the default values for the parameters were used. To examine the importance of newly introduced relations, which can not be included in the experiments with propositional machine learning system, we also performed another series of experiments (labeled FOIL-ND in the tables). In this second series, the values of the parameters are set, so that no determinate literals are included in the model. The determinate literals in the case of meta-learning are exactly the literals of the form `relation(D,V)` used in the propositional experiments.

When using FOIL with default parameters setting, the induced concepts use the determinate literals only. Thus, the induced concept do not include any of the newly introduced relations, measuring the properties of individual attributes. In part, this is due to the heuristics used in FOIL. To surpass this a different parameters setting was used, so that only indeterminate literals are included in the concept, if they are available. Still, some of the indeterminate literals used in our framework (e.g. `attr_class_mutual_inf(D,A,V)`) are defined for all the

**Table 4.** Concepts induced with ILP system FOIL.

| appl_c45 | |
|---|---|
| appl_c45(A) :-<br>class_entropy(A,B),<br>B>0.991231.<br><br>appl_c45(A) :-<br>num_of_bin_attrs(A,B),<br>B>13. | appl_c45(A) :-<br>not(kurtosis(A,_1)),<br>attr_entropy(A,B,C), C<=1.<br><br>appl_c45(A) :-<br>not(entropy(A,_1)),<br>attr_kurtosis(A,B,C),<br>C>10.1512.<br><br>appl_c45(A) :-<br>entropy(A,B), B>2.27248. |
| **appl_cn2** | |
| appl_cn2(A) :-<br>class_entropy(A,B),<br>perc_of_na_values(A,C),<br>C>4.70738, B>0.276716.<br><br>appl_cn2(A) :-<br>mutual_inf(A,B),<br>B>4.32729. | appl_cn2(A) :-<br>perc_of_attr_na_values(A,B,C),<br>attr_disc(A,B), C>2.30794. |
| **appl_knn** | |
| appl_knn(A) :-<br>num_of_attrs(A,B),<br>num_of_disc_attrs(A,C),<br>C<=6, B<>C.<br><br>appl_knn(A) :-<br>values(A,B),<br>B>4.15385. | appl_knn(A) :-<br>not(entropy(A,_1)).<br><br>appl_knn(A) :-<br>skewness(A,B), B<=1.45483. |

datasets and attributes and do not make any discrimination between positive and negative examples. With heuristics used in FOIL such literals are not induced in the induced concepts.

The concepts induced with FOIL are presented in Table 4. The only concept based on the property of a single attribute is the one for the applicability of CN2. It states that CN2 is applicable to the datasets which contain discrete attribute with more then 2.3% unknown values. It should be noted that this concept was induced with FOIL-ND, which gained maximal accuracy in leave-one-out experiments for CN2 (see Table 6). Another indeterminate literals that occurs in the concepts are `not(entropy(A,_1))` (stating that all attributes in the dataset `A` are continuous) and `not(kurtosis(A,_1))` (all attributes in `A` are discrete).

**Table 5.** Concepts induced with ILP system TILDE.

appl_c45

```
class_entropy(A,C) , C > 0.991231 ?
+--yes: yes [9 / 9]
+--no: num_of_bin_attrs(A,D) , D > 13 ?
       +--yes: yes [2 / 2]
       +--no: no [9 / 9]
```

appl_cn2

```
attr_kurtosis(A,C,D) , D > 22.7079 ?
+--yes: no [5 / 5]
+--no: attr_class_mutual_inf(A,E,F) , F > 0.576883 ?
       +--yes: kurtosis(A,G) , G > 3.87752 ?
       |        +--yes: yes [7 / 7]
       |        +--no: num_of_examples(A,H) , H > 270 ?
       |               +--yes: yes [3 / 3]
       |               +--no: no [3 / 3]
       +--no: no [2 / 2]
```

appl_knn

```
num_of_attrs(A,C) , C > 19 ?
+--yes: no [4 / 4]
+--no: num_of_examples(A,D) , D > 57 ?
       +--yes: num_of_bin_attrs(A,E) , E > 15 ?
       |        +--yes: no [1 / 1]
       |        +--no: yes [12 / 13]
       +--no: no [2 / 2]
```

The concepts induced with the ILP system TILDE are presented in Table 5. The only concept based on the property of a single attribute (kurtosis of a single attribute and mutual information between the class and the attribute) is the one for the applicability of CN2.

**Table 6.** Accuracy of the meta-level models measured using leave-one-out method.

| **Dataset** | C4.5 | CN2 | $k$-NN | FOIL | FOIL-ND | TILDE | default |
|---|---|---|---|---|---|---|---|
| appl_c45 | 16/20 | 16/20 | 14/20 | 16/20 | 7/20 | 18/20 | 11/20 |
| appl_cn2 | 9/20 | 5/20 | 11/20 | 9/20 | 13/20 | 9/20 | 0/20 |
| appl_knn | 9/20 | 11/20 | 9/20 | 10/20 | 11/20 | 14/20 | 12/20 |
| Sum | 34/60 | 32/60 | 34/60 | 35/60 | 31/60 | 41/60 | 23/60 |

Finally, the results of the leave-one-out experiments are summarized in Table 6. Please note here, that the model induced in each leave-one-out experiment can differ from the others (and the ones presented in Tables 4 and 5), but the accuracy of the classifiers was our primary interest in these experiments. It can

be seen from the table that FOIL has a slightly better and FOIL-ND a comparable accuracy with respect to the propositional machine learning systems. TILDE outperforms other machine learning systems on two out of three meta-learning tasks.

## 4 Discussion

The work presented in the paper extends the work already done in the area of meta-learning in several ways. First, an ILP framework for meta-level learning is introduced. It extends the methodology for dataset description used in [3] with non-propositional constructs which are not allowed when using propositional classification systems for meta-level learning. ILP framework incorporates measures for individual attributes in the dataset description. The ILP framework is also opened for incorporating prior expert knowledge about the applicability of classification algorithms. Also all the datasets used in the experiments are public domain and the experiments can be repeated. This was not the case with the StatLog dataset repository where more then half of the datasets used are not publicly available. Another improvement is the use of a unified methodology for measuring the error rate of different classification algorithms and the optimization of their parameters.

The ILP framework used in this paper was build to include the measures used in the state-of-the-art meta-learning studies. It can be extended in several different ways. Beside including other more complex statistical and information theory based measures, it can be also extended with the properties measured for any subset of attributes or examples in the dataset. Individual or set of examples from the dataset can also be included in the description.

From the preliminary results based on the experiments with only twenty datasets it is hard to make strong conclusions about the usability of the ILP framework for meta-level learning. The obtained models can capture some chance regularities beside the relevant ones. However, the results of the leave-one-out evaluation method show slight improvement of the classification accuracy when using an ILP description of the datasets. This improvement should be further investigated and tested for statistical significance performing experiments for other datasets from the UCI repository. To obtain a larger dataset for meta-level learning, experiments with artificial datasets should also be performed in the future.

## Acknowledgments

This work was supported in part by the Slovenian Ministry of Science and Technology and in part by the European Union through the ESPRIT IV Project 20237 Inductive Logic Programming 2. We greatly appreciate the comments of two anonymous reviewers of the proposed version of the paper.

## References


1. Aha, D. (1992) Generalising case studies: a case study. In *Proceedings of the 9th International Conference on Machine Learning*, pages 1–10. Morgan Kaufmann.
2. Blockeel, H. and De Raedt, L. (1998) Top-down induction of first order logical decision trees. *Artificial Intelligence*, 101(1–2): 285–297.
3. Brazdil, P. B. and Henery, R. J. (1994) Analysis of Results. In Michie, D., Spiegelhalter, D. J., and Taylor, C. C., editors: *Machine learning, neural and statistical classification*. Ellis Horwood.
4. Clark, P. and Boswell, R. (1991) Rule induction with CN2: Some recent improvements. In *Proceedings of the Fifth European Working Session on Learning*, pages 151–163. Springer.
5. Džeroski, S., Cestnik, B. and Petrovski, I. (1993) Using the m-estimate in rule induction. *Journal of Computing and Information Technology*, 1:37–46.
6. Kalousis, A. and Theoharis, T. (1999) NEOMON: An intelligent assistant for classifier selection. In *Proceedings of the ICML-99 Workshop on Recent Advances in Meta-Level Learning and Future Work*, pages 28–37.
7. Murphy, P. M. and Aha, D. W. (1994) *UCI repository of machine learning databases* [http://www.ics.uci.edu/˜mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science.
8. Quinlan, J. R. (1993) *C4.5: Programs for Machine Learning*. Morgan Kaufmann.
9. Quinlan, J. R. and Cameron-Jones, R. M. (1993) FOIL: A midterm report. In Brazdil, P., editor: *Proceedings of the 6th European Conference on Machine Learning, volume 667 of Lecture Notes in Artificial Intelligence*, pages 3–20. Springer-Verlag.
10. Wettschereck, D. (1994) *A study of distance-based machine learning algorithms*. PhD Thesis, Department of Computer Science, Oregon State University, Corvallis, OR.