Machine Learning Project 2010-2011

Meta-learning from an experiment database

Summary. The Machine Learning Experiment Database, `http://expdb.cs.kuleuven.be/`, is a database containing over half a million descriptions of experimental runs with machine learning systems. These descriptions include: which system was run on which dataset, with which parameter settings, under which experimental conditions, and measurements about the outcome of the experiment (e.g., how long the run took, what the accuracy of the learned model was, etc.)

Analyzing such a database can tell us a lot about the behavior of machine learning systems. The use of machine learning to learn about machine learning systems is also called meta-learning. Meta-learning has been a popular research subject during the nineties, but afterwards interest in the topic dwindled, due to given the limited availability of meta-learning data at that time, and the strong dependence of results on parameter settings of learning systems. The Machine Learning Experiment Database now solves those problems by making available an unprecedented amount of data.

The goal of this project is to analyze this machine learning experiment database, using standard SQL queries and using data mining tools, and to prepare a report in which you summarize your findings, and make suggestions about ways in which this experiment database could be improved. You can work on this project in a group consisting of two students. The total time allocated for completing the project is **60 hours** per person. A **first report** describing how you will address the task must be handed in by **Friday, November 12, 2010**. The deadline for handing in the **final report** is **Thursday, December 23, 2010**.

The educational goal of this project is threefold. First, you will get some practice with applying the machine learning techniques you have seen on a real problem. Second, since this is meta-learning, the models you obtain can by themselves teach you something about the properties of machine learning algorithms. Third, you will train some research skills: the literature on meta-learning contains interesting background information and possibly good suggestions on how to proceed; reading this literature will help you to obtain more meaningful results.

1 Meta-learning

Meta-learning refers to "learning about learning". Over the last decades, several researchers have tried to address questions such as: "Under what circumstances is it better to learn a decision tree than to train a neural network?", "Could I predict in advance, just by looking at the dataset, that method A will not work well", etc. Examples of such research are the European projects Statlog and MetaL, in which many groups collaborated to find answers to these questions. In the more recent MetaL project, a database was constructed containing information about a number of datasets (about 30) and how well different machine learning systems performed on it. The datasets were described using attributes such as size of the datasets, number of attributes, number of classes (they were all classification problems), number of numerical / symbolic attributes, the amount of noise in the data, the amount of missing values in the data, the accuracy of a decision tree learnt from the data, the accuracy of a nearest neighbor algorithm, ... This database (which consisted of a single table) was then mined in the hope of finding interpretable rules, such as "On small datasets, decision tree learners tend to do less well than linear regression".

While some useful results came out of these projects, this approach turned out not to yield the insights the researchers had hoped for. Part of the reason was that the meta-learning dataset consisted of very few examples (one per dataset) and many descriptive attributes, which makes learning difficult. There was also the criticism that there was typically one run of a machine learner per dataset, which was considered to be representative; but in fact, many learning systems have multiple parameters that can have a large influence on their behavior.

The results of the MetaL project, together with pointers to related literature, are available at `http://www.ofai.at/research/impml/metal/metal-bib.html`.
A good overview on meta-learning is also given in a chapter of Joaquin Vanschoren's doctoral dissertation, which is made available on Toledo. These are good starting points for finding your way in the literature on meta learning.

2 The Experiment Database

The basic idea behind Experiment Databases is the following. Imagine that you have a database in which each possible machine learning system has been run on each dataset ever created, and that every imaginable measurement about the run was recorded and stored. If you wanted to find out how well learner A performs, in terms of accuracy, on dataset B, you could simply query that database instead of setting up the experiment yourself. If you would want to know what the average accuracy is of method A, as compared to method B, on datasets that have less than 1000 examples and more than 100 attributes, this again could be answered using a simple SQL query to the database. And so on. Clearly, meta-learning would become a lot easier if you had such a database.

Of course it is not possible to have all machine learning systems (especially the ones that have not yet been developed) in the database, nor to have all existing datasets. But you can have a large sample of them. The Machine Learning Experiment Database that is online at `http://expdb.cs.kuleuven.be` contains results of more than half a million experiments with over 50 machine learning systems on over 80 datasets. That is not a bad starting point for meta-learning.

The `expdb.cs.kuleuven.be` website contains documentation on the structure of the experiment database, contains example queries, and pointers to the meta-learning literature as well as to papers about experiment databases. Perhaps the most informative articles (available online) are

- H. Blockeel, and J. Vanschoren (2007). Experiment databases: Towards an improved experimental methodology in machine learning. In: Knowledge Discovery in Databases: PKDD 2007, 11th European Conference on Principles and Practice of Knowledge Discovery in Databases, Proceedings. Lecture Notes in Computer Science 4702, pp. 6-17.
- Joaquin Vanschoren, Bernhard Pfahringer, Geoffrey Holmes (2008). Learning from the Past with Experiment Databases. In: PRICAI 2008: Trends in Artificial Intelligence, 10th Pacific Rim International Conference on Artificial Intelligence, Hanoi, Vietnam, December 15-19, 2008, Proceedings. Lecture Notes in Computer Science 5351, pp. 485-496.

3 Learning from a relational database

The experiment database at `expdb.cs.kuleuven.be` is a relational database. It contains multiple tables, some describing datasets, others describing learners, experiments, etc., and all of these are

linked. With a language like SQL, you can query such a relational database in a relatively simple way. (Example queries are available on the website.) It is not so easy to *mine* such a database, however. Most machine learning or data mining tools expect the data they will analyze to be available in one single table: one row per instance, one column per attribute. This limitation can be addressed in two different ways.

Approach 1: Summarize the data in one table. Study the database and think about what properties might be useful and relevant for a particular learning task. Then define an SQL query that creates one table in which all the relevant information is stored. Next, you can use a standard data mining tool (such as Weka) to analyze the data in the table. If you want to move on to a next learning task, you can create a new table (different tables may be needed for different learning tasks) and repeat the process.

Approach 2: Use a relational learner. A relational learner is a system that can handle data in multiple tables in a relational database. An example of such a system is the ACE tool (http://dtai.cs.kuleuven.be/ACE/, login `mlstudent`, password `aceplease`). This is an inductive logic programming tool; it wants data in a logical format. You will first need to rewrite the data in the database into this logical format. Next, you can learn first-order logical models from the data. To do this, you will need to set up some configuration files that define the kind of patterns that ACE should look for. To answer a different question, you need not create a new table, you can just change the configuration.

The ACE tool has a steep learning curve: it will take you a fair amount of time to familiarize yourself with it. Using it within the scope of this project will be a challenge. We therefore recommend the first approach.

4 Tasks

This is an overview of different tasks that you will need to address. Indicative timings are given for each task. These are estimates: some tasks may take more or less time, depending on your own interests and skills.

1. *Literature Study*: Familiarize yourself with the concept of an experiment database, and with meta-learning. Get a feeling for what has been done before in this context, and what are the open questions. The Experiment Database website contains multiple pointers to background literature on these subjects, and some pointers are given above. Estimate: 10 to 15 hours per person.

2. *Querying the database*: Familiarize yourself with the experiment database. Think of a few meta-learning related questions, and formulate these questions in SQL. Interpret the results. Based on the outcome, you may want to try new questions. Estimate: together with point 4, 20-30 hours per person. (You can shift focus to points 2 or 4 depending on your interests.)

3. *Data Representation*: Before you can use machine learning or data mining tools on the experiment database, you need to rewrite the data from the Experiment Database into a suitable format. There are two options here. The first option is to extract from the database a single table with data that you will consecutively mine using a standard data mining tool, for instance, Weka. (You can create multiple such tables for multiple experiments.) The second option is to rewrite the data into a logical format and apply an inductive logic programming system to it. ACE is an example of a tool that can be used to mine relational data. The second option is in

principle more powerful, but will likely be more challenging; a tool like ACE has a relatively steep learning curve. Estimate: some 10 hours per person.

4. *Data Mining*: Once the database is ready to be mined, try to find patterns in the data. Example patterns that you can try to look for, are: (estimated, together with point 2, at 20-30 hours per person)
   - *Learning curves*: how does the performance of machine learning systems change with the size of the data set? You can look at both time and accuracy of the resulting models.
   - *Clustering*: Use clustering methods (hierarchical, flat, conceptual, ...) on the data and analyze the results. Are there machine learning systems that behave "similarly"? You can try to cluster machine learning systems (according to accuracy, runtimes, other measures, ...); you can also try to cluster datasets according to how well machine learning systems perform on them.
   - *Correlations*: Between what machine learning systems do you find the highest correlation in terms of their accuracy on the different datasets? Are there systems for which you find a negative correlation? Formulate some questions of your own as well.
   - *Recommendation*: Try to build a model that, based on the description of a dataset, predicts which system is best used on that dataset, and (somewhat more challenging) with what parameters. Since this is a challenging model to learn, you could try to build a model that predicts which parameters to use for a single algorithm first, or, similarly, a model that chooses between two learning algorithms only.
5. Optional: Build a recommender system that, given a new dataset, quickly analyzes the dataset and, based on this, proposes a learning system that should be used. (This was the ultimate goal of the MetaL project.) This system will likely use the recommendation model mentioned in the previous point. If that model uses features that are difficult to compute from a fresh dataset, consider dropping those features from the model. It may be a good idea to program this system in Java and have it interface with the Weka system, since that system already contains code for learning, representing and applying models. (0-20 hours per person.)
6. Write a report with your findings. (5 hours each)

5 Planning, Requirements, Evaluation

You can work on this project in groups of two persons. Form the groups as quickly as possible, and e-mail the names of the members of your group to kurt.driessens@cs.kuleuven.be.

What to hand in: In a first phase of the project, we ask you to write down in a brief report (1 or 2 pages) explaining how you will address these tasks. Describe what literature you have read. Give examples of questions that you want to find answers for, and give an idea of how you intend to find the answers (Querying or mining the database? What data representation will you use, what data mining systems will you use?). This report must be handed in by **November 12, 2010**. Please send your report in Adobe PDF format to the aforementioned e-mail address. The quality of this report can influence your final score for the project, so do your best to come up with a good plan.

After the reports are handed in, you will get feedback on your report, and, where necessary, we may give more concrete guidelines on how to proceed.

By **December 23**, you will need to hand in a final report, describing details of your approach and the results obtained.

- Clearly state the questions you address with your research
- Describe how you try to answer them (what experiments were performed, how do you measure success, etc.),
- Write out the conclusions you draw from your experiments together with a scientifically supported motivation for these conclusions. Such a motivation should include descriptions of used techniques and performed experiments and a report and discussion of the results of these experiments.
- Report the total time you spent on the project, and how it was divided over the different tasks mentioned (were the time estimates in this document realistic?)
- Comment on the experiment database: was it easy to use, what could be improved, ...

The total length of your report should be at most 10 pages. In addition, you should hand in any software you have written to complete the task.

Evaluation:

- You will be evaluated on the quality of your two reports and on your approach to solving the tasks. The most important criterion is that you understand well the machine learning techniques that you use and that you have applied them in a sound manner, and that you interpret the models resulting from the data mining work in a meaningful way.
- There will be an oral discussion of your project report in January.

Time: The time allocated for completing this project is **60 hours**. This *does not* include studying the Machine Learning course material, but it *does* include the literature study specific for this project's topic. If you find some of the above estimates for the different parts unrealistic, mention this in your report. Remember that you can divide the work among two people!

Questions: Please direct any questions that you may have about the project to the Toledo forum.

Good luck!