Implementation of the Active Learning Challenge in the Causality Workbench Virtual Lab

Isabelle Guyon ISABELLE@CLOPINET.COM ***Clopinet, California, USA***

Abstract

We implemented the Active Learning challenge (discussed at workshops held in conjunction with AISTATS 2010 and WCCI 2010) using the Virtual Lab of the Causality Workbench. The Virtual Lab is a web portal with a Matlab® backend. This report gives a brief description of the software architecture and gives examples of uses of the Matlab GLOP package, which powers the Virtual Lab.

1. Introduction

The Virtual Lab is a web-based platform implemented in PHP/MySQL with a Matlab® backend, see `http://www.causality.inf.ethz.ch/workbench.php`. It allows users to carry out virtual experiments on data generating systems, which emulate real systems. The users are given a budget of virtual cash to run their experiments. They can place queries to gain access to data, which are answered by the system in exchange for virtual cash.

For the Active Learning challenge, 13 datasets were wrapped into Matlab objects in the GLOP package (Generative Lab Object Package): six development datasets, six datasets used for final testing, and one toy dataset used for illustration purpose. The GLOP package managed the user's cash account, computed performances, and answered queries. The GLOP package is available for download from `http://www.causality.inf.ethz.ch/repository.php?id=23`.

The goal of active learning is to learn a task as fast as possible while using as few labeled examples as possible. In active learning experiments, one starts with an unlabeled dataset. A single labeled experiment is initially provided (seed). Other labels must be purchased for virtual cash. The performance of the participants is monitored with their learning curve, which plots the classification performance on test (or validation) data, as a function of the amount of virtual cash spent (which is proportional to the number of examples). The website of the Active Learning challenge is available at `http://www.causality.inf.ethz.ch/activelearning.php`.

2. Architecture of the Virtual Lab

The Virtual Lab is an environment accessible via the Internet. Users must be registered to place queries and get answers. Each user is known to the outside world via his WorkbenchID and can remain anonymous. Registered users have a private area called "My Lab" in which

they can monitor the progress of their experiments. A summary of the results is displayed on a public leaderboard.

2.1. Software architecture

Users interact with the Virtual Lab via a web interface, supported by scripts written in the PHP scripting language embedded into HTML. The interface has a MySQL database backend, which holds tables of users, experiments and results. Submissions are performed via an upload page, which also records the use name, experiment name, model (or dataset name), and date. The submitted zip archive is then renamed to include this information. For example, a query submitted by user John on dataset SYLVA with experiment name "verif", on 03/25/2010 at 15h 34m 03s would be named:

`John_sylva_verif_2010-03-25-153403.zip`

When a query is submitted to the Virtual Lab (in the form of a number of files following a specific format and bundled in the zip archive), it is entered in a queue. Jobs are processed in the order received. For every job Matlab gets launched. A Matlab script then parses the file name to recuperate information and instantiates a **query object** (see Section 3) that loads the query information. A **model object** for the correct model/dataset (SYLVA in our example) is also instantiated. If the user has already started an experiment with the same name, a saved version of the corresponding Matlab object gets reloaded in the model object.

Matlab then processes the query. The model object is updated to record changes in the virtual cash account and prediction performances. The query object is modified to hold the query answer. Both objects are saved and the Matlab session is terminated. A script periodically checks whether query answers have become available and updates the database.

The overall process is summarized in Figure 1.

2.2. Custom version of the Virtual Lab

We have implemented several challenges using the Virtual Lab. Each time, we created a slightly customized version of the interface to run the challenge. Examples are found at: `http://www.causality.inf.ethz.ch/challenges.php`. For the Active Learning challenge, registered users were allowed to:

- Download data from the Data page `http://www.causality.inf.ethz.ch/activelearning.php?page=datasets`.
- Submit results and queries bundled as a zip archive from the Submit page `http://www.causality.inf.ethz.ch/activelearning.php?page=submit`.
- Retrieve their results from the personal Mylab page `http://www.causality.inf.ethz.ch/activelearning.php?page=mylab` or from the leaderboard `http://www.causality.inf.ethz.ch/activelearning.php?page=leaderboard`.

The queries and answers follow a specific format described on the Instruction page `http://www.causality.inf.ethz.ch/activelearning.php?page=instructions#cont`. This is

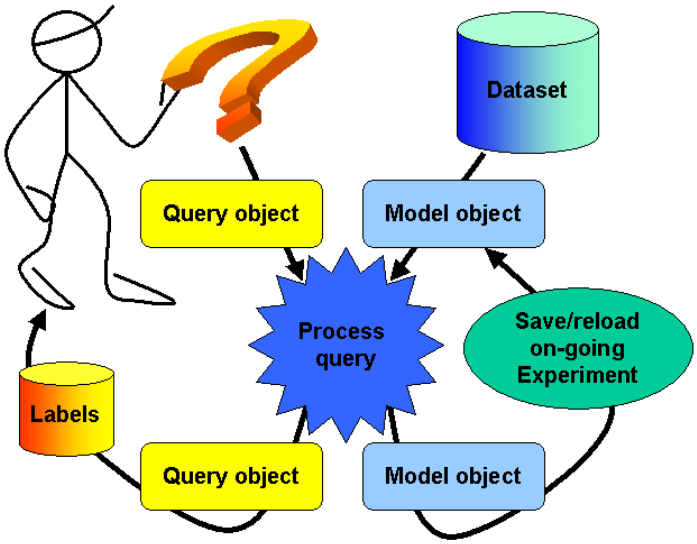

Figure 1: **Conducting experiments in the Virtual Lab.**

a simplified version of the more general query-answer format of the virtual lab, described at `http://www.causality.inf.ethz.ch/workbench.php?page=info`.

Submitted queries must be bundled in a .zip archive and include the following text files:

- **dataname.sample -** Row numbers of the samples for which you want labels (one number per line). You can ask only for the labels of training examples (the first T lines in the data matrix for T training examples).
- **dataname.predict -** Prediction scores of the target variable (label) for all the examples in the dataset (one number per line; as many lines as there are lines in the data table). The scores can be any number, larger numerical values indicating higher confidence in positive class membership. See the Evaluation page for more details.

We asked the participants to always send predictions when they placed a query. Predictions could be sent without requesting labels (i.e. an archive may contain only dataname.predict and not dataname.sample) but no request for labels was answered unless a valid prediction file dataname.predict was provided.

The answers to the query were made available through the My Lab page, as a zip archive containing the following files:

- **dataname.README -** An message indicating whether the query was processed successfully.
- **dataname.sample -** The identity of the samples for which the labels are provided (line number in the data matrix).
- **dataname.label** - The corresponding labels (target values).

- **dataname.score** - The AUC score for the predictions made (see Evaluation).
- **dataname.ebar** - An estimate of the error bar on the AUC.
- **dataname.global_score** - The area under the learning curve (see Evaluation).
- **dataname.totalcost** - The cost of the query. Usually equal to the number of labels provided, but may be less if some labels have been queried before and already paid for.

In addition, the graph of the learning curve was made available.

2.3. Usage

For the Active Learning challenge, step-by-step instructions were provided to the participants:

1. Get the data from the Data page.
2. Prepare your experiment: Choose one of the datasets, e.g., ALEX. Eventually preprocess the data. Give a name to your experiment, e.g., "myexp1". An experiment is a set of query/answers, starting from an initial budget allowing you to purchase all the labels, and ending when you run out of virtual cash. A new experiment is created when you submit your first query.
3. Iterate:
   - **Predict -** Using the examples with known labels (at the first iteration use the seed example, which has a positive label), train a predictor and provide prediction scores for ALL the examples of the dataset (including those used for training). Any sort of numeric prediction score is allowed, larger numerical values indicating higher confidence in positive class membership.
   - **Sample -** Choose among the remaining unlabeled examples those for which you want the purchase labels. You may only query examples in the first half of the dataset (training examples). If you query test examples, you will not receive those labels (and not be charged either).
   - **Submit a query -** Format your predictions and chosen samples using the Query Format. Submit it via the Submit page. Select the name of the dataset (e.g., ALEX). Enter the name of a new experiment to create it (e.g., "myexp1"), or select the name of an on-going experiment. Give a description of this entry to remember what you did (the description will be for your eyes only, it will not show on the leaderboard).
   - **Retrieve the labels -** After submitting your query, you are redirected to the My Lab page. Refresh periodically the page until the results of your query are ready for download.

During the development period, the challenge participants used the six development datasets. They could experiment on the same dataset as many times as they wanted.

Every time they obtained a new budget allowing them to purchase all the training labels. After the first experiment, they obtained all the training labels and could use them to develop algorithms without going through the website. This may have facilitated their work. But we urged them to try the submission system to make sure it worked for them because during final testing, they could only perform one experiment on each final dataset.

2.4. Sample Matlab code

We provided sample queries, and sample Matlab code to help the participants prepare their queries. The sample code processed the queries and generated learning curves for the example dataset ALEX. The sample Matlab code also works in conjunction with GLOP (see Section 3) and can emulate the functioning of the Virtual Lab. It may be downloaded from: `http://www.causality.inf.ethz.ch/al_data/Sample_code.zip`.

3. The Generative Lab Object Package (GLOP)

We developed an object-oriented interface to easily incorporate new data generating models, implemented in Matlab. Data generating models may include artificial models, realistic simulators of real systems, or wrappers of datasets of real data. For the Active Learning challenge, we used only wrappers of real datasets.

GLOP is based on two simple abstractions:

- query object, and
- model object.

The query object holds the query provided by the user or the query answer (including data delivered by the data generating model). It has a fixed structure. The model object is a template from which new data generating models can be derived.

GLOP model objects are equipped with a number of generic methods. At the Matlab prompt, the following commands allow the user to find information on the models:

```
> whoisglop             % List the models
> default(model)        % Get default values of a model
> methods(model)        % List the methods
> properties(model)     % List the properties (data members)
```

The following commands allow the user to create a query of a given type. The query types presently supported include TRAIN, TEST, OBS, SURVEY, EXP, PREDICT, AL, and PREPRO. The type AL was created for the Active Learning challenge. Here are a few examples of instantiations of query objects:

```
> q=query('OBS 20');    % Create an "observation query" to get observational data
> q=query('TRAIN');     % Create a "train" query to get the whole training set
> q=query('TEST');      % Create a "test" query to get the whole test set
```

For the challenge, a Matlab script gets called every type a query is placed by a participant in the form of a "dataname.sample" and a "dataname.predict" files bundled in a zip archive. The script instantiate an AL query object using the following command:

```
> q=query('file');      % Create a query by loading the data in file.zip
```

There are presently 28 models in GLOP, 13 of which have been programmed for the Active Learning challenge. Those include the development dataset models:

`ibn_sina, hiva, nova, orange, sylva, and zebra,`

the final evaluation datasets:

`a, b, c, d, e, f,`

and the toy data model "alex" (which stands for "active learning example").

Here are a few examples of model instantiations:

```
> a=alarm({'cost_per_sample=1', 'cost_per_var_observation=1', ...
  'cost_per_var_manipulation=2'});
> a=sprinkler;
```

For the Active Learning challenge, all models are derived from the class *al_model*, which itself inherits from the generic object *model*. A given model is usually associated with a task to be completed. Important methods of the *model* object include:

```
> initial_budget(a)             % Shows the virtual cash budget allocated for the task
> task_n_pricing(a)             % Describes in words the task to be performed and costs ass
> [q, a]=process_query(a, q);   % Processes the query q
```

Note that in Matlab syntax, a method is called with the object name as first argument. Other arguments follow. Hence, *process_query* is a method of the objects of class *model* and has a query as argument. The method *process_query* is overloaded for objects of type *al_model*. In particular, it updates the calculation of the learning curve using the latest results provided with the query.

4. Conclusion

The Virtual Lab of the Causality Workbench was implemented with the idea of benchmarking causal discovery algorithm. It is also a very effective tool to organize sophisticated machine learning challenges unrelated to causality but requiring query/answer interactions between the participant and a central server. Additionally, the Matlab backend allows us to perform easily a variety of result analyzes and provide on-line performance feed-back in the form of graphs, such as learning curves. This greatly expands the possibilities in terms of challenge design. The website of the Active Learning challenge remains open for post-challenge submissions at `http://www.causality.inf.ethz.ch/activelearning.php`. A more detailed technical report on the Virtual Lab is available in (Guyon et al., 2009).

References

I. Guyon, C. Aliferis, G. Cooper, A. Elisseeff, O. Guyon, J.-P. Pellet, P. Spirtes, and A. Statnikov. The causality workbench virtual lab, 2009.

Acknowledgments

This development of the Virtual Lab of the Causality Workbench was funded by the U.S. National Science Foundation under Grant N0. ECCS-0725746. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. We are very grateful to all the members of the causality workbench team for their contributions and in particular to our co-founders Constantin Aliferis, Greg Cooper, André Elisseeff, Jean-Philippe Pellet Peter Spirtes, and Alexander Statnikov. The website implementation was done by MisterP.net who provided exceptional support.