Abstract

This paper describes a system for the automatic evaluation and deployment of language models and sequence recognizers over the web. First we discuss the need for web-based evaluation, and some of the problems that we have overcome in order to make this successful. We then go on to describe the client and server algorithm deployment models. We illustrate the approach with results on some language modelling and sequence recognition datasets.

1. Introduction

In some fields the evaluation can be done largely theoretically - a classic example being sorting algorithms. In areas such as pattern recognition, however, where the emphasis is on dealing with noisy real-world data, the behaviour of an algorithm on large quantities of such data is perhaps the most important test.

In an ideal world, having designed an implemented a new pattern recognition algorithm, we would be able to just tell an evaluation server about the existence of the implementation, and leave the server to organise the job of evaluating the algorithm on lots of datasets, using widely accepted evaluation methods.

In previous work we described an Internet-based system for evaluating pattern recognition algorithms.

The rest of this Paper is structured as follows. The next Section discusses related research. Section ?? describes our initial prototype, Section ?? runs through a simple example problem and Section 5 concludes.

2. Related research

Related work on performance evaluation infrastructure can be classified under three general headings: data archives, competitions, and algorithm test-harnesses, though research has typically not addressed one area exclusively.

Archives of data for evaluation are widely available: for example the University of California at Irvine Machine Learning Repository [1] has a wide range of PAMI databases (123 at present) available for download, and the Pilot European Image Processing Archive at University of Essex [2] provides a similar wide range of image databases.

In the field of handwriting recognition, both on-line [3] and off-line databases are widely used. Although all these databases enable direct comparison of different algorithm results on the same data, there is no standardisation of APIs, resulting in unnecessary software overheads to access the databases.

Finally, algorithm test-harnesses such as Delve [6] and HATE [7] go one step further than competitions in providing not just a dataset and problem specification, but also an evaluation framework which tries to ensure objectivity. Algoval takes some inspiration from these systems but aims to be more automated, easier to use and address a wider range of problems than Delve. More recently the Open Mind Initiative [8] has some of the same motivations as Algoval, though Algoval is more focussed on providing an infrastructure for automatic algorithm evaluation and seems to be further developed implementation-wise.

3. Users and requirements

First we outline the main classes of potential user of such a system. Currently, the problem definer is assumed to be at the server-side and have full system privileges. This is something we would like to change in future versions of the system. All other classes of user interact with the system using a standard web-browser and may use it from anywhere on the Internet.

Problem definer: the problem definer lays down a precise specification for a problem. This must involve speci-
fying the nature and format of the data and the interfaces between the evaluation framework and a candidate solution algorithm.

**Algorithm developer:** with Algoval the developer writes an algorithm that conforms to the specified interface for a problem, then simply submits the code (or a URL for the code) to Algoval for automatic evaluation. It must be emphasised that the extra effort of conforming to a particular interface will in most cases be minimal, while the time saved by automatic evaluation will be significant.

**Dataset provider:** Algoval can automatically test a wide range of algorithms on any submitted dataset, providing that the dataset is in an allowed format or is supplied wrapped in a class that implements the relevant interface. Hence, the dataset provider gets rapid feedback on how difficult his data is to deal with.

**Results consumer:** This is any person interested in the results of applying the algorithms on the system to the datasets on the system. Such people might include companies wishing to build pattern recognition systems from high-performance components, government funding bodies trying to objectively rate the quality of some research or researchers interested in which algorithms perform best on which data.

### 3.1. Algorithm Provider

The algorithm provider must: register as a user; read the problem description on the competition web-site; download the interface for the problem; design an algorithm that implements the interface; inform the algoval server of the URL of the root class algorithm or otherwise upload a zip file that contains all the classes used by the algorithm.

We wrote the algorithm shown in Table ?? and uploaded it to the system. We also uploaded several other versions, for L0 (corresponding pixels have a difference of zero if within a certain threshold of each other - set to 10 - and one otherwise) and L2 (Euclidean) norms, and fast versions of these that only look at every other pixel on each row, and every other row (hence, the fast versions sub-sample the images and only compare a quarter of the pixels compared by the full versions.

### 4. System Maintenance and Scalability

An important question is how much effort the system is to maintain when up and running.

For example, if we are just running a dataset server, then the cost is low.

What about changes to the interfaces?

**Problem lifecycle.**

### 5. Conclusions

Empirical evaluation is a fundamental part of developing any pattern recognition algorithm or system. In this paper we have described a prototype implementation of an evaluation system that operates over the Internet. This system automates the process of testing algorithms on datasets, and of publishing the results of those tests on the Internet. Wide adoption of such a system would avoid much of the wasteful duplication of effort that currently occurs in the evaluation process.

**Acknowledgements** Support for this work from the UK Post Office is gratefully acknowledged.

**References**


---

1Currently only the URL option is implemented