# Thesis Experiments

Andrew Healy

July 8, 2016

For the following experiments, the provers under consideration are Z3, Alt-Ergo, CVC3, CVC4, Yices, veriT. The sets *training* and *test* are disjoint subsets of the 919 individual proof obligations generated by Why3 from the 117 example WhyML programs included in the standard Why3 distribution. The independent/predictor variables are various metrics statically derived from the proof obligation goals: number of operators, number of variables, number of constants etc. The dependent/repsponse variables are the *time* taken by each prover (measured in seconds) and the *result* returned by Why3. These were dynamically measured with a statistical confidence interval of 90%. The ratio of the size of the training set to the test set is approximately 3:1.

*Null hypothesis:*
**Static metrics derived from proof obligation goals do not indicate the performance of any SMT prover.**

## 1 Predicting Prover Result

Given the set of possible prover results

$$Res = \{Valid, Invalid, Unknown, Timeout, Error\}$$

Can the correct prover result $r \in Res$ be predicted for an arbitrary proof obligation $p \in training$ for each prover?

### 1.1 Research Questions addressed

- Can the single most effective solver for a Why3 proof obligation be predicted by learning from static metrics?
- Can a useful ranking of solvers be predicted for a Why3 proof obligation?

### 1.2 Methodology

**Data preprocessing:** Perform standard scaling of the independent/predictor variables in *training*. Call this matrix $X$

Whether the procedure below is implemented separately for each prover depends on whether the learning algorithm supports multivariate output (e.g. Decision Trees) or not (e.g Support Vector classifiers).

1. Separate *result* columns from *time* of the dependent/response variables in *training*. Call this matrix $y$
2. Fit the classification model on $X$ and $y$
3. For each proof $p$ obligation in *testing*:
    a. Scale $p$'s independent/predictor variables with the scaler used in step 1.
    b. Predict the result $r$ on the model

### 1.3 Evaluation

We will compare our classifier against $|Res|$dummy classifiers: that is, $\mathcal{CLF}_r$ will always choose result $r$ for each proof obligation in *testing*. The evaluation of multiclass classification where the class weights are not balanced (i.e. we have far more *Valid* responses in *training* than *Error*, thankfully) can be given with the **Weighted Empirical Error**: Let $W_c$ be the weight of the class $c$. Set the weights such that $\frac{1}{W_c} \sim \frac{1}{n} \sum_{i \leq n} 1_{c_i = c}$ and define the weighted empirical error

$$err_W(g) = \frac{1}{n} \sum_{i \leq n} W_{c_i} 1_{g(x_i) \neq c_i}$$

Another evaluation metric for multiclass classification is the **Receiver Operating Characteristic** (ROC). This is a plot of the rate of false positives against the rate of false negatives for each class. Commonly used for evaluating binary classifications, it can be extended to the multiclass case by computing the mean of the scores for each class (fig: 1). Our classes would be *Res*. steeper curves (approaching the top left) and a higher **Area Under Curve** (AUC) scores are better: they show a low false positive rate and high true positive rate.

## 2 Predicting Prover Time

Can the time a prover takes to return a result be accurately predicted for an arbitrary proof obligation $p \in training$ for each prover?

### 2.1 Research Questions addressed

(same as 1.1)

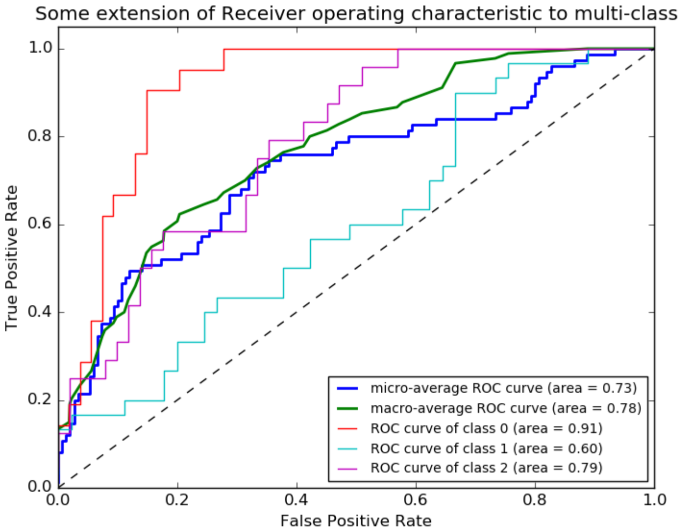

Figure 1: Example ROC plot for multiclass classification: taken from $http : //scikit - learn.org/stable/auto\_examples/model\_selection/plot\_roc.html$

## 2.2 Methodology

*time* is a continuous-valued variable measured in seconds generally in the range $[0.001, 12.3]$ (depending on the prover's interaction with Why3). There are a number of possibities for predicting this variable - either attempting to predict it directly as a regression task or by treating it as a classification task by the approximating the actual values into discrete bins. Each have their distinct methodolgies but share the same initial data preprocessing step (as in section 1.2)

### 2.2.1 Regression

1. Separate *time* columns from *result* of the dependent/response variables in *training*. Call this matrix $y$

2-3. Same as steps 2-3 in section 1.2; replacing classification with regression to predict *time*

### 2.2.2 Binary Classification

The rationale behind treating this as a binary classification problem is that by inspecting the timings for each prover, it is clear that the vast majority of results are returned in a very short amount of time (*Valid*, *Invalid* or instantly

*Unknown*, probably), or close to the timeout value (*Timeout* and a large proportion of *Unknown* and *Error* results).

**Data preprocessing: (in addition to section 1.2)**
For the vector $T$ of *time* reponses for each prover:
a. Define $lo$ as the 0.25 quartile of $T$ and $hi$ as the 0.75 quartile.
b. $t' = \begin{cases} hi, & \text{if } t > mean(T) \\ lo, & \text{otherwise} \end{cases}$
c. $y$ is the vector of binarized $t'$ times

Otherwise the methodolgy is the same as section 1.2 with the result being the predicted *time* (which will be one of each prover's $hi$ or $lo$ values).

### 2.2.3 Multiclass Classification

Increasing the number of bins from 2 gives a better chance of returning an accurate result through classification.

**Data preprocessing: (in addition to section 1.2)**
a. Bin the values in *time* so that each bin has an approx. equal number of values and the number of bins does not exceed 5 (to increase the liklihood of accurate predictions)
b. Assign each value to it's corresponding bin
c. $t' = mean(bin_t)$
c. $y$ is the vector of discretised $t'$ times

(same as section 1.2 - with the result being the mean of the predicted bin)

## 2.3 Evaluation

### 2.3.1 Regression

A number of metrics to evaluate regression models can handle the multi-output case. An average of the error for all outputs is returned. In our case, the output is a vector of reals corresponding to out set of provers.

- The **Mean Squared Error** determines the squared difference between the predicted value and the actual value.

- The $R^2$ **Score** (also called the coefficient of determination) is a measure of how well the model is expected to predict future values. It takes into account the variance of the dependent/response variables.

We will compare our regressor to a dummy regressor which always returns the mean of *time*.

Table 1: Evaluation of using 3 methods to predict time (with dummy values apart from **Actual** row. Note that the Classification metrics cannot be used for the reggressors)

| | **AUC** | **Weighted Emp. Err.** | **Mean Sq. Err.** | $R^2$ **Score** |
|---:|---:|---:|---:|---:|
| **Actual** | 1.0 | 0.0 | 0.0 | 1.0 |
| **Regressor** | - | - | 0.5 | 0.7 |
| mean | - | - | 0.7 | 0.3 |
| **Binary** | 0.5 | 0.4 | 0.4 | 0.6 |
| *lo* | 0.2 | 0.6 | 0.6 | 0.5 |
| *hi* | 0.2 | 0.6 | 0.6 | 0.5 |
| **3-bins** | 0.7 | 0.3 | 0.3 | 0.8 |
| Bin1 | 0.3 | 0.7 | 0.7 | 0.4 |
| Bin2 | 0.3 | 0.7 | 0.7 | 0.4 |
| Bin3 | 0.3 | 0.7 | 0.7 | 0.4 |

### 2.3.2 Binary and Multiclass Classification

We can use all the metrics from sections 1.3 and 2.3.1 to end up with a table similar to Table 1. Here, *hi*, *lo*, Bins1-3 are dummy classifiers whose predicted constants can be used for regression metrics and as labels for classification metrics.