Closed Loop Machine Learning: Reproduction and Evaluation of Phase A Results

Alireza Tamaddoni-Nezhad
Stephen Muggleton

Department of Computing, Imperial College
180 Queen's Gate, London SW7 2BZ, UK

email: {atn,shm}@doc.ic.ac.uk

December 2001

Abstract

This report aims to present an effort for reproducing phase A results of the Closed Loop Machine Learning project. In this report the experimental method which has been used for testing ASE-Progol in phase A of the project is explained and the results based on this experimental method are reproduced. The performance of ASE-Progol is tested using the same testing strategy which has been used in the previous reports of the project. In addition, we have used a test strategy which uses a different test-set. The results of both test strategies are presented and discussed.

1 Introduction

The purpose of the Closed Loop Machine Learning project has been to develop a framework for "Automatic Experimentation" which involves Machine Learning for generating hypotheses and Robotics for devising trials to discriminate between hypotheses. In this framework there is a closed loop between the process of forming hypotheses and the collection of data. The long term goal is to use this framework in Functional Genomics to discover the function of genes.

According to the project proposal [1], the main objectives of the project are: "to test whether closed loop Machine Learning Systems can (i) efficiently converge to accurate hypotheses and (ii) be physically realised using

robotics and successfully applied to a discovery task in functional genomics". For this purpose a system called ASE-Progol (Active Selection of Experiments with Progol) has been developed. ASE-Progol is an Active Learning system which uses the Inductive Logic Programming (ILP) system Progol5.0 [6] for generating hypotheses together with a CART-like algorithm [2] to select trials which minimize the expected cost of experimentation. More details about the design and implementation of ASE-Progol can be found in [3, 4].

To date, ASE-Progol has been tested on: a) a small and simplified model of functional genomics and b) a metabolic pathway from the aromatic amino acid pathway of yeast. The results of these studies which correspond to phase A and phase B of the project are reported in [3] and [4] respectively.

In this report we have reproduced the results of phase A experiments using the same experimental method as used in [3]. We have also used a test strategy for measuring the predictive accuracy of ASE-Progol which is different from the test strategy used in the previous reports. The experimental settings, test strategies and the results of the experimentation are discussed in the next section.

2 Reproducing Phase A results

As mentioned earlier, in this report we aim to reproduce the results of phase A of the Closed Loop Machine Learning project. These results are based on an experiment which was initially reported in [3]. This experiment aimed to study the performance of ASE-Progol on an abstract and simplified model of functional genomics. According to [3] the purpose of this experiment is: "to investigate whether the cost of converging upon an accurate hypothesis is significantly reduced if ASE-Progol samples trials at random, rather than selecting them so as to minimise the cost of experimentation."

In the following sections, we first explain the experimental materials and method and then represent and discuss the results.

2.1 Materials

In this experiment one approach to functional genomics, namely the effect of single-gene-deletion growth trials, has been modeled by a logic program. This logic program and the relevant metabolic pathway are shown in Table 1 and Figure 1 respectively. During the experimentation some of `code/2` facts (which correspond to the function of unknown genes) are removed from the model and the ability of ASE-Progol to 'cost-efficiently'

Table 1: A logic program which represents the functional genomics model which has been used in phase A experiments.

```
phenotypic_effect(Gene, Growth_medium):-
     nutrient_in(Nutrient, Growth_medium),
     metabolic_path(Nutrient, Mi),
     enzyme(E, Mi, Mj),
     codes(Gene, E),
     metabolic_path(Mj, Mn),
     essential_molecule(Mn),
     not(path_without_E(Growth_medium, Mn, E)).

nutrient_in(Nutrient, Growth_medium):- element(Nutrient, Growth_medium).

metabolic_path(A, A).
metabolic_path(A, B):- enzyme(_, A, B).
metabolic_path(A, B):- enzyme(_, A, X), metabolic_path(X, B).

path_without_E(Growth_medium, Mn, E):-
     nutrient_in(Nutrient, Growth_medium),
     path_without_E(Nutrient, Mn, E).
path_without_E(A,A,_).
path_without_E(A,B,E):- enzyme(E2,A,B),not(E=E2).
path_without_E(A,B,E):- enzyme(E2,A,X),not(E=E2),path_without_E(X,B,E).

essential_molecule(ess_mol_1).       essential_molecule(ess_mol_2).
essential_molecule(ess_mol_3).       essential_molecule(ess_mol_4).

enzyme(enzyme_a, nut_1, metabolite_1).
enzyme(enzyme_b, nut_2, metabolite_2).
enzyme(enzyme_c, nut_3, metabolite_3).
enzyme(enzyme_d, metabolite_1, metabolite_4).
enzyme(enzyme_e, metabolite_1, metabolite_5).
enzyme(enzyme_f, metabolite_1, metabolite_6).
enzyme(enzyme_g, metabolite_2, metabolite_6).
enzyme(enzyme_l, metabolite_6, metabolite_7).
enzyme(enzyme_h, metabolite_2, ess_mol_4).
enzyme(enzyme_i, metabolite_3, ess_mol_4).
enzyme(enzyme_j, metabolite_4, ess_mol_1).
enzyme(enzyme_k, metabolite_5, ess_mol_2).
enzyme(enzyme_m, metabolite_7, ess_mol_3).

codes(gene_a, enzyme_a).   codes(gene_b, enzyme_b).   codes(gene_c, enzyme_c).
codes(gene_d, enzyme_d).   codes(gene_e, enzyme_e).   codes(gene_f, enzyme_f).
codes(gene_g, enzyme_g).   codes(gene_h, enzyme_h).   codes(gene_i, enzyme_i).
codes(gene_j, enzyme_j).   codes(gene_k, enzyme_k).   codes(gene_l, enzyme_l).
codes(gene_m, enzyme_m).
```

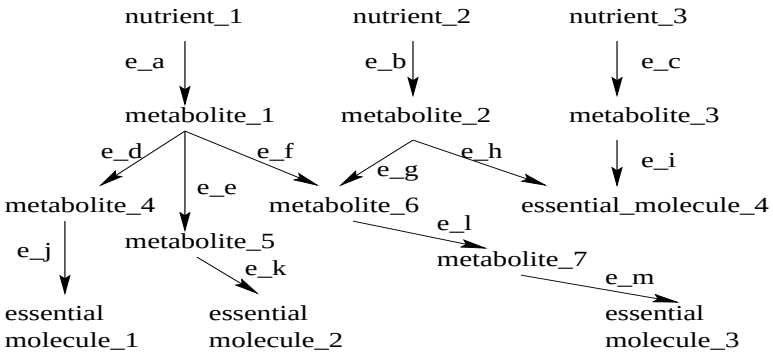


Figure 1: A graph which represents the metabolic pathway of the genomics model in Table 1.

Table 2: Cost A and Cost B of Growth Media which has been used in the experiments.

Growth medium	Cost of Trial	
	Costs A	Costs B
nutrient_1	10	10
nutrient_2	20	100
nutrient_3	30	1000
nutrient_1, nutrient_2	30	110
nutrient_1, nutrient_3	40	1010
nutrient_2, nutrient_3	50	1100
nutrient_1, nutrient_2, nutrient_3	60	1110
Sum of costs	240	4440

recover the performance of the model (or learn the function of the genes) is measured. The cost of the experimentation is assumed to be equal to the total cost of the growth media (nutrients) which have been used in the growth trials during the experimentation. In this experiment it is assumed that the costs of the growth media are as shown in Table 2. It is also assumed that the result of each trial can be represented by positive and negative instances of the predicate `phenotypic_effect/2`. For example `phenotypic_effect(gene_a, [nutrient_1, nutrient_2])` represents a growth trial in which a mutant strain of an organism is created by removing `gene_a` and the growth media contains `nutrient_1` and `nutrient_2`. The result of a trial can be positive (growth) or negative (no growth). For example, $\neg$ `phenotypic_effect(gene_a, [nutrient_2])` shows that no growth is observed for `gene_a` and `nutrient_2`. There are only 3 nutrient in the model, so there are 7 possible growth media for each of the 13 gene in the model. Hence, we have 91 examples for the predicate `phenotypic_effect/2`. According to the metabolic pathway in Figure 1, 45 of theses examples are positive and the other 46 are negative examples.

2.2 Method

As mentioned earlier, the purpose of this experiment is to study the performance of ASE-Progol and its ability to recover an incomplete genomics model. For this purpose, ASE-Progol uses 'Theory Completion' which is implemented in Progol5.0. Theory Completion, Progol5.0 and the results of testing Progol5.0 on the functional genomics model are presented in [6]. Unlike the experiments with Progol5.0, in the experiments with ASE-Progol it is assumed that ASE-Progol is given one example in each iteration (the result of a growth trial). In the existing implementation, the result of a trial is determined by the oracle – a file which contains the results of all possible trials – rather than by the laboratory.

During the experimentation, the genomics model shown in Table 1 is made incomplete by randomly removing a number of `code/2` facts. This number varies between 5, 9 and 13 in different experiments. ASE-Progol is executed on the incomplete model such that trials are selected which: a) minimise the expected cost of experimentation b) sampled at random. The performance is then measured for the recovered model for both ASE-Progol and random sampling of trials.

In the present experiment, we use two different strategies for testing the performance of ASE-Progol:

1. Test Strategy 1. This is the original strategy which has been used

Table 3: The experimental method as in [3]. This has been used in both TEST STRATEGY 1 and TEST STRATEGY 2.

```
1 for k in (0, 4, 8) do
2    select k codes/2 facts at random;
3    remove the other 13 − k codes/2 facts from the model;
4    for each one of these 13 − k codes/2 facts do
5       Add the codes/2 fact to the Stochastic Logic Program;
6       for i in 1 to 10 do
7          execute ASE-Progol twice such that trials are:
8             1. selected which minimise the expected cost of experimentation.
9             2. sampled at random.
10         for each of (ase random) do
11            determine the accuracy and cumulative cost (CC) of experimentation
12            at each iteration of the CLML cycle.
13         end
14      end
15      for each of (ase random) do
16         for j in (40 80 120 160 200 240) [a] do
17            for i in 1 to 10 do
18            estimate accuracy when CC = j
20            end
21         end
22      end
23   end
24end
25for each of (ase random) do
26   for j in (40 80 120 160 200 240) do
27      calculate the mean and standard error of the accuracy when CC = j
28   end
29   plot j×100/Max versus accuracy with horizontal error bars, where Max is
30   the limit on the cost of the experimental resources which may be consumed.
31   for l in (78 83 88 93 98)[b] do
32      estimate CC when accuracy = l
33   end
34end
35plot CC_Random / CC_ASE versus l and (CC_Random − CC_ASE) / CC_Random × 100 versus l.
```

[a] for costs B j was assigned the values $(1040, 1720, 2400, 3080, 3760, 4440)$ rather than $(40, 80, 120, 160, 200, 240)$ as shown in Table 2.

[b] These values were chosen because they fall within the range of accuracy values which were achieved by both the ASE and Random approaches.

to measure the predictive accuracy of ASE-Progol. According to [3]: "The test-set for any given gene was the seven examples of the observable predicate `phenotypic_effect/2` for the gene in question. The performance measure used was predictive accuracy on the observable predicate. The performance of the hypothesis with the highest compression was measured". In this test strategy, the predictive accuracy of a part of the model, which involves only one gene, is measured on a test-set which contains only examples of the gene in question (i.e. 7 examples of the observable predicate `phenotypic_effect/2`).

2. TEST STRATEGY 2. In this test strategy, we measure the predictive accuracy of ASE-Progol on a test-set which contains all 91 examples of the observable predicate `phenotypic_effect/2`. This strategy has been also used to test Progol5.0 on a natural language data-set as well as the functional genomics model [6]. The purpose of this test strategy is to measure the predictive accuracy of the whole model on the complete set of examples.

The experimental method which has been used for both TEST STRATEGY 1 and TEST STRATEGY 2 is the same as used in [3] and is summarized in Table 3. In the first part of the method, the predictive accuracy and Cumulative Cost (CC) of the experimentation at each iteration of the Closed Loop Machine Learning are computed and recorded (line 11). In the second part, the predictive accuracy for an specific Cumulative Cost (CC) is estimated (line 18) by finding the points which correspond to the two CCs which are closest to j, fining the line $y = mx + c$ which goes through these points and then calculating y when $x = j$. When this leads to extrapolation above or below the valid range $(0 - 100)$ the estimate is taken to be 100 or 0 respectively. Similarly, the Cumulative Cost for a specific accuracy is estimated (line 32) by finding the costs which correspond to the two previously interpolated/extrapolated accuracies which are closest to l, fining the line $y = mx + c$ which goes through these points and then calculating y when $x = l$. When this leads to extrapolation above or below the valid range $(0 - Max)$ the estimate is taken to be Max or 0 respectively.

2.3 Results

The results of the experiment which are plotted for both TEST STRATEGY 1 and TEST STRATEGY 2 include:

1. Resources Consumed versus Predictive Accuracy (Tables 4 and 7).

2. The difference in cumulative cost (expressed as a percentage of the random cumulative cost) versus predictive accuracy (Tables 5 and 8).

3. The ratio of cumulative costs versus predictive accuracy (Tables 6 and 9).

These results are plotted separately based on the number of `code/2` facts which are removed from the model (5, 9 and 13). The results are also plotted as an average for all executions of ASE-Progol [1].

2.4 Discussion

The results of the experimentation based on TEST STRATEGY 1 are consistent with the results which have been reported in [3]: "the cost of converging upon a hypothesis with an accuracy in the range $80-95\%$ is reduced if trials are selected by Closed Loop Machine Learning (CLML) rather than if they are sampled at random".

The results of TEST STRATEGY 1 also show that when all resources are consumed ASE-Progol always converges to a hypothesis with an accuracy in the range $95-100\%$ (Table 4). However, the levels of the predictive accuracies, which have been achived by ASE-Progol, are different when TEST STRATEGY 2 is used. For example, according to Table 7, the maximumm average accuracy is around 88.5% when 5 `code/2` facts are removed from the model, 71% when 9 `code/2` facts are removed and 55% when all 13 `code/2` facts are removed from the model. According to the table, the maximum average accuracy on all executions of ASE-Progol is around 66%. The results of TEST STRATEGY 2 are consistent with the fact that less predictive accuracy is expected when more `code/2` facts are missing.

As mentioned earlier, the difference between TEST STRATEGY 1 and TEST STRATEGY 2 is in the number of examples which have been used as a test set for measuring the predictive accuracy. In TEST STRATEGY 1, the predictive accuracy of the whole model (with 5, 9 and 13 removed `code/2` facts) is measured on a test-set which contains only examples of the gene in question (i.e. 7 examples of the observable predicate `phenotypic_effect/2`). Whilst, in TEST STRATEGY 2 the test-set contains all possible examples (i.e. 91 examples of the observable predicate `phenotypic_effect/2`).

One possible drawback of these test strategies is that they cannot be reliable in the case of *overfitting* the training data [5]. For example, in TEST

[1] Note that the separate plots are not appeared in [3] and only the average results for all executions were reported.

Table 4: TEST STRATEGY 1: Resources Consumed vs Predictive Accuracy. Resources consumed $= \frac{CC}{Max} \times 100$ where CC is the cumulative cost of the trials performed so far during training and Max is the limit on the cost of the experimental resources which may be consumed during training. Predictive Accuracy is measured on the seven examples of the observable predicate `phenotypic_effect/2` for the gene in question.

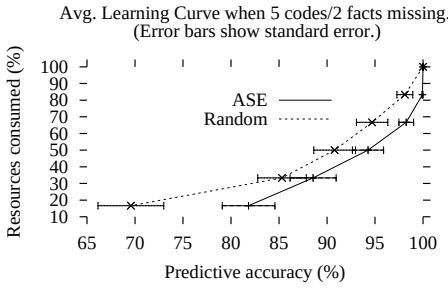
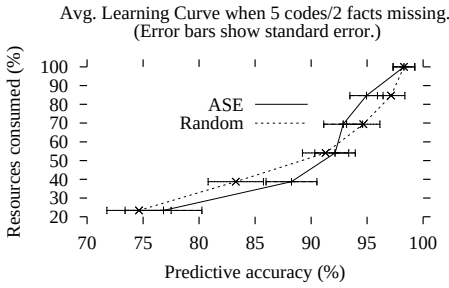
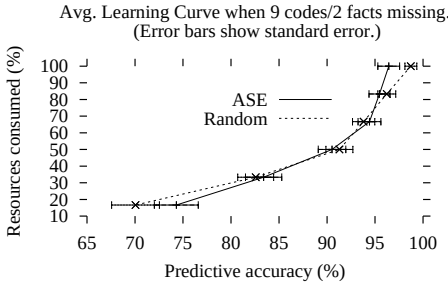
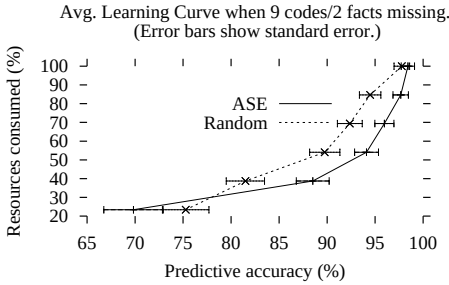
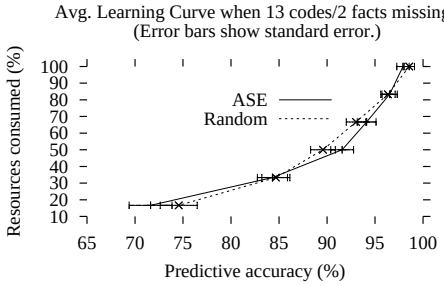
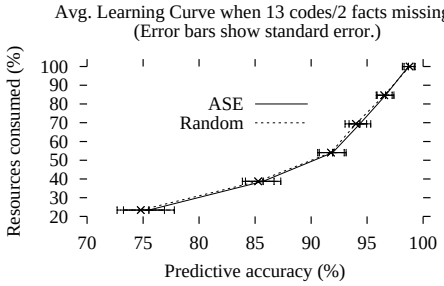
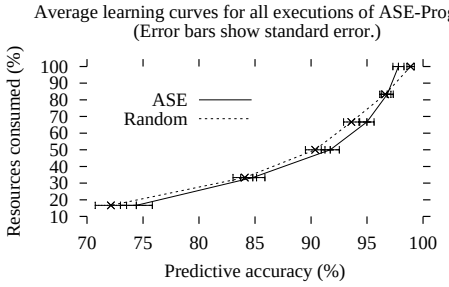
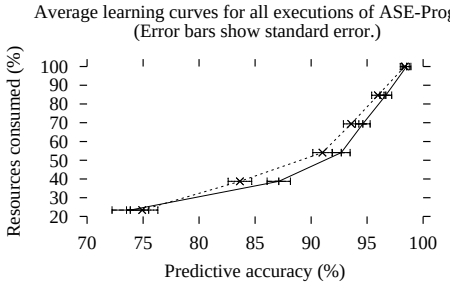
Costs A	Costs B
	
	
	
	

Table 5: TEST STRATEGY 1: Saving in Cumulative cost vs Predictive Accuracy. Savings in cumulative cost = $\frac{(CC_{Random} - CC_{ASE})}{CC_{Random}} \times 100$ where CC_{Random} is the cumulative cost when trials are sampled at random and CC_{ASE} is the cumulative cost when trials are selected to minimise the expected cost of experimentation. Predictive Accuracy is measured on the seven examples of the observable predicate `phenotypic_effect/2` for the gene in question.

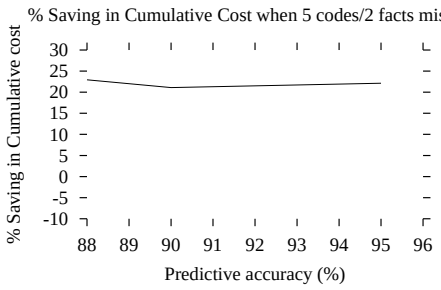
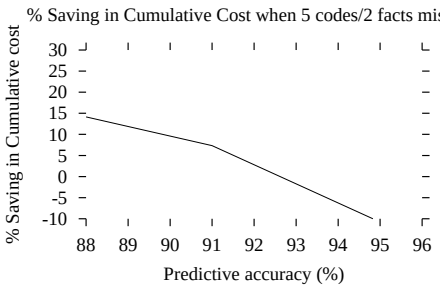
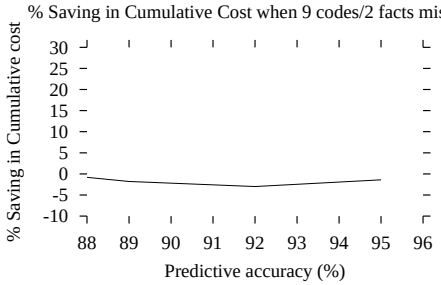
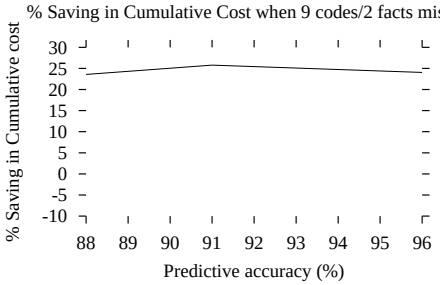
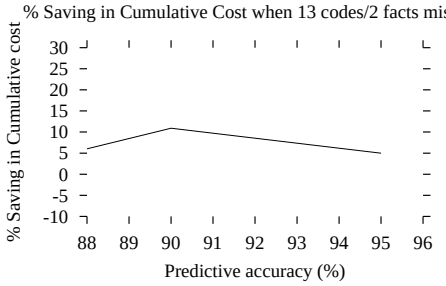
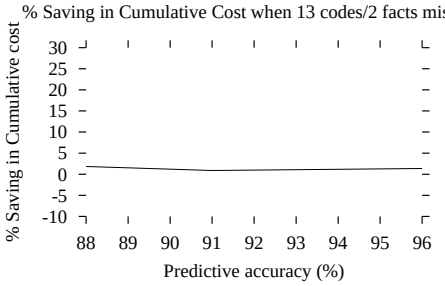
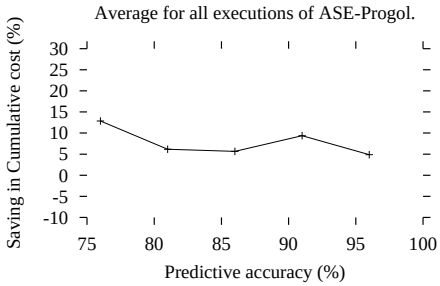
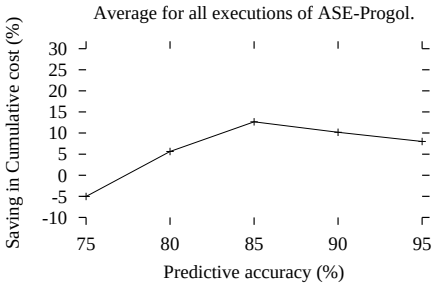
Costs A	Costs B
	
	
	
	

Table 6: TEST STRATEGY 1: Ratio in Cumulative cost vs Predictive Accuracy. Ratio of cumulative costs = $\frac{CC_{Random}}{CC_{ASE}}$ where CC_{Random} is the cumulative cost when trials are sampled at random and CC_{ASE} is the cumulative cost when trials are selected to minimise the expected cost of experimentation. Predictive Accuracy is measured on the seven examples of the observable predicate `phenotypic_effect/2` for the gene in question.

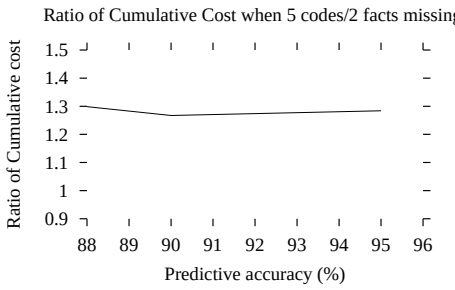
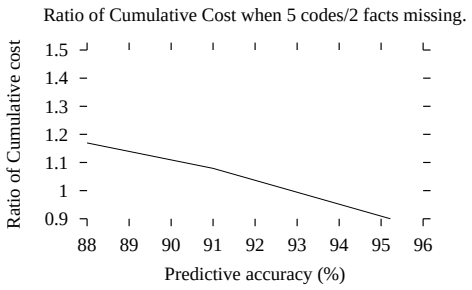
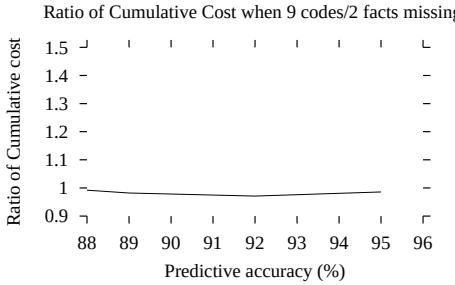
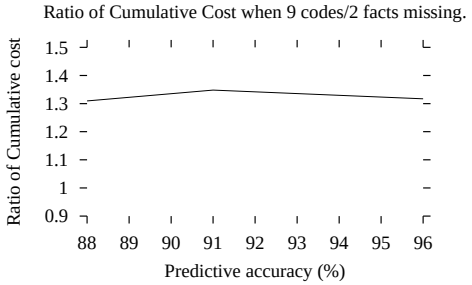
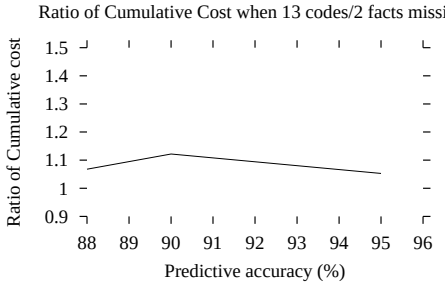
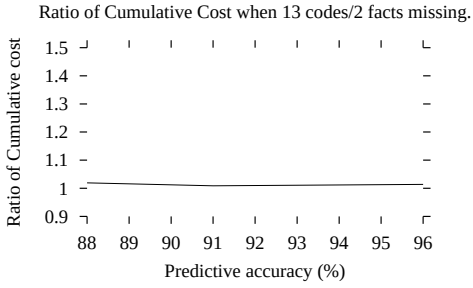
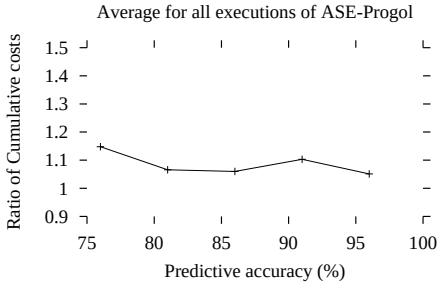
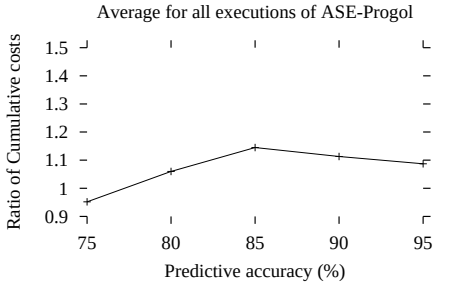
Costs A	Costs B
	
	
	
	

Table 7: TEST STRATEGY 2: Resources Consumed vs Predictive Accuracy. Resources consumed $= \frac{CC}{Max} \times 100$ where CC is the cumulative cost of the trials performed so far during training and Max is the limit on the cost of the experimental resources which may be consumed during training. Predictive Accuracy is measured on all 91 examples of the observable predicate `phenotypic_effect/2`.

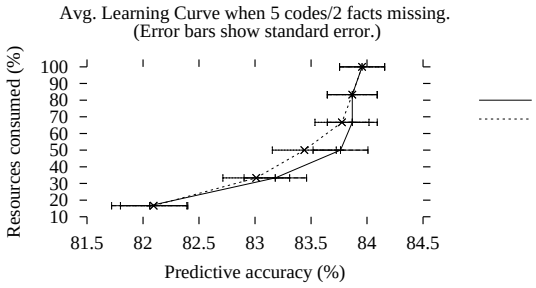
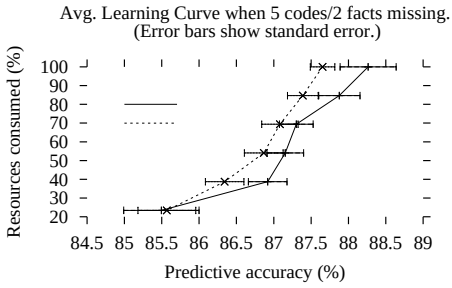
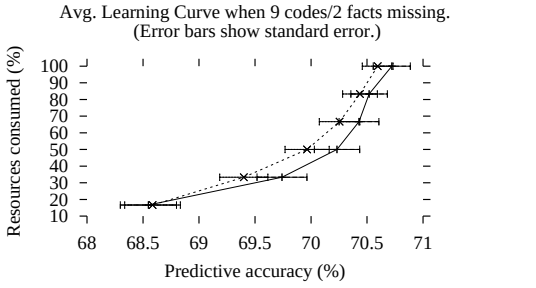
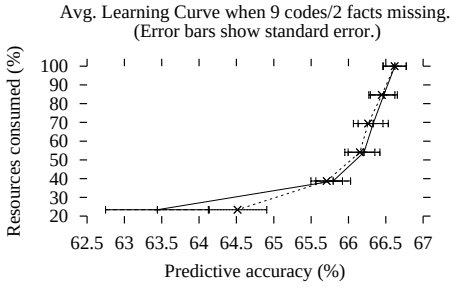
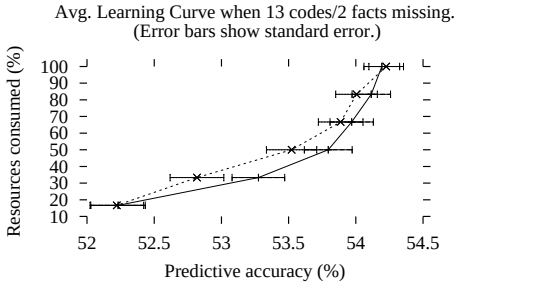
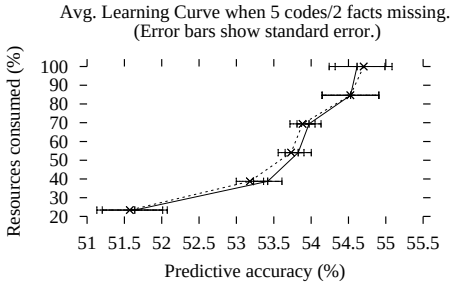
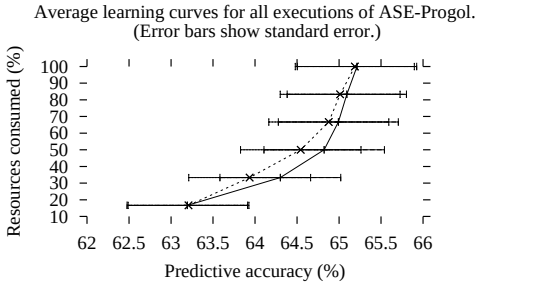
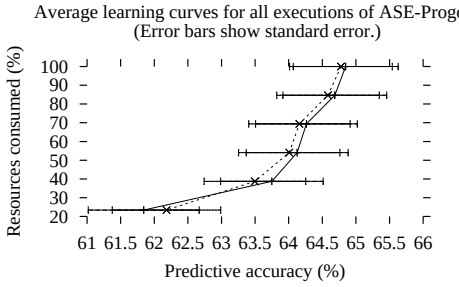
Costs A	Costs B
	
	
	
	

Table 8: TEST STRATEGY 2: Saving in Cumulative cost vs Predictive Accuracy. Savings in cumulative cost = $\frac{(CC_{Random} - CC_{ASE})}{CC_{Random}} \times 100$ where CC_{Random} is the cumulative cost when trials are sampled at random and CC_{ASE} is the cumulative cost when trials are selected to minimise the expected cost of experimentation. Predictive Accuracy is measured on all 91 examples of the observable predicate `phenotypic_effect/2`.

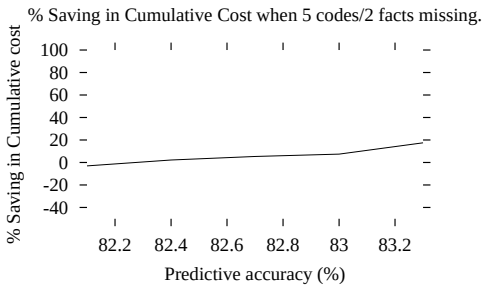
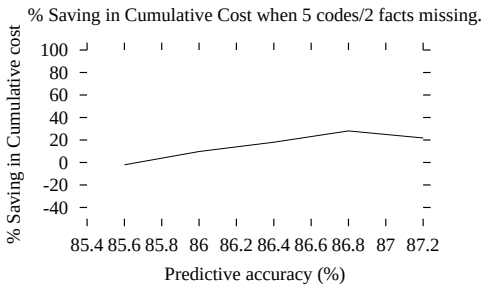
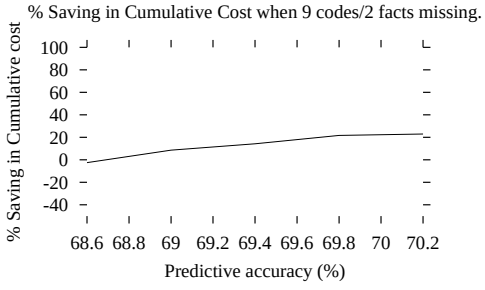
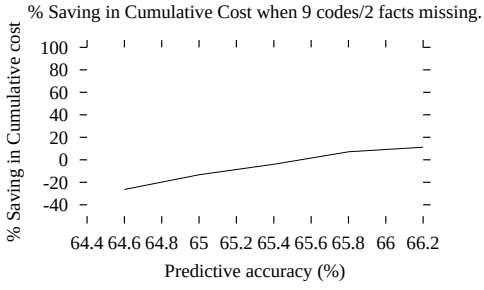
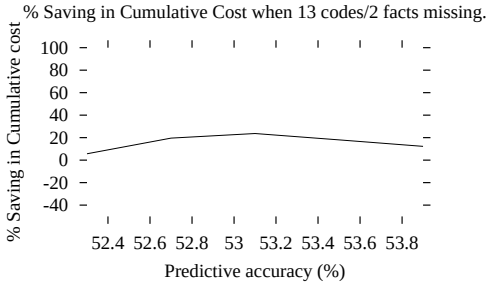
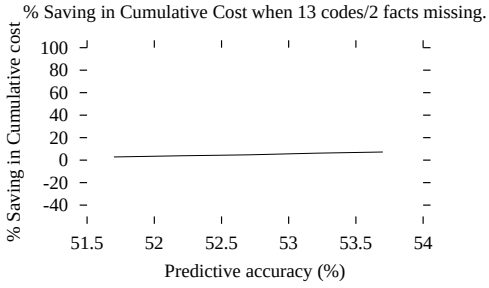
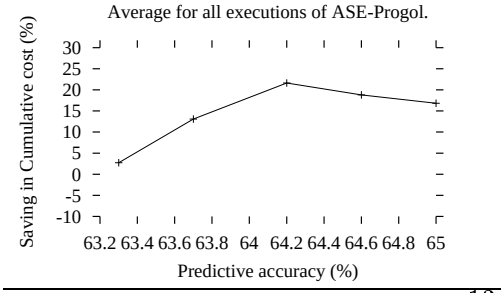
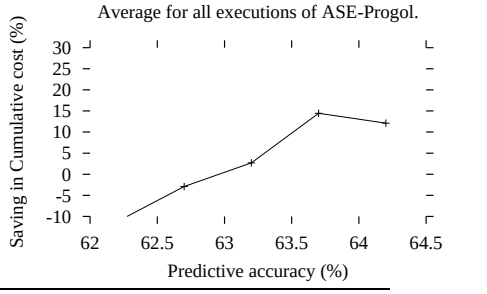
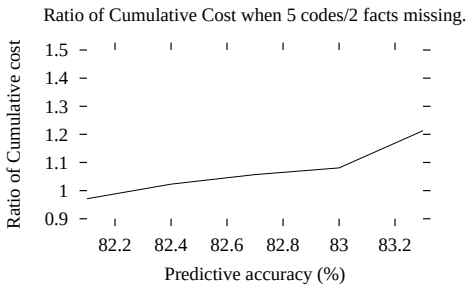
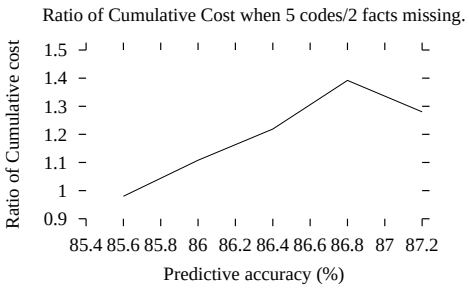
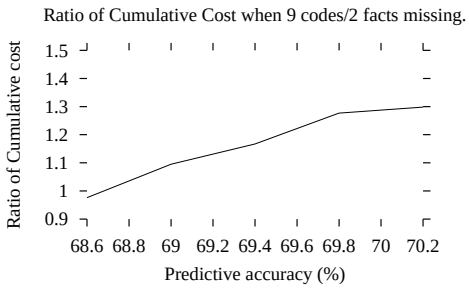
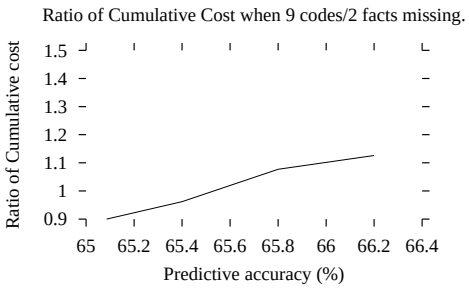
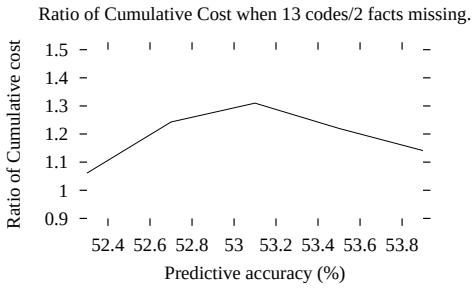
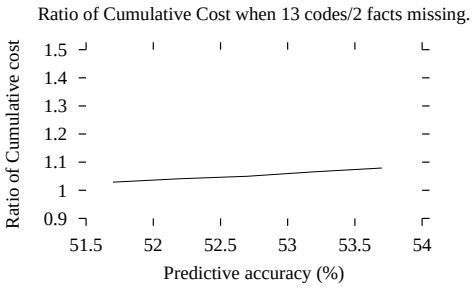
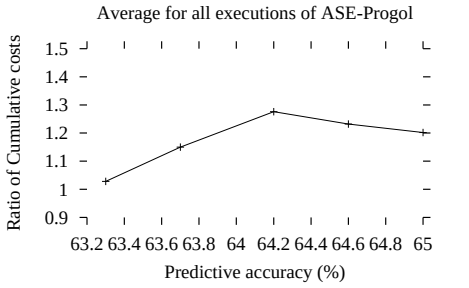
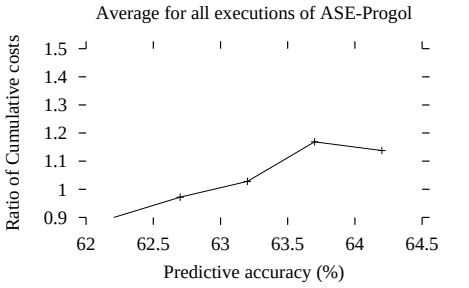
Costs A	Costs B
	
	
	
	

Table 9: TEST STRATEGY 2: Ratio in Cumulative cost vs Predictive Accuracy. Ratio of cumulative costs = $\frac{CC_{Random}}{CC_{ASE}}$ where CC_{Random} is the cumulative cost when trials are sampled at random and CC_{ASE} is the cumulative cost when trials are selected to minimise the expected cost of experimentation. Predictive Accuracy is measured on all 91 examples of the observable predicate `phenotypic_effect/2`.

Costs A	Costs B
	
	
	
	

STRATEGY 1 assume that in the ith run of ASE-Progol the gene in question is `gene_m`. In each cycle of the closed loop learning, ASE-Progol is given one example of the observable predicate `phenotypic_effect/2` (i.e. the result of a growth trial). This training example can be one of the 7 possible examples for `gene_m`:

```
phenotypic_effect(gene_m, [nut_1]).
phenotypic_effect(gene_m, [nut_2]).
:- phenotypic_effect(gene_m, [nut_3]).
phenotypic_effect(gene_m, [nut_1,nut_2]).
phenotypic_effect(gene_m, [nut_1,nut_3]).
phenotypic_effect(gene_m, [nut_2,nut_3]).
phenotypic_effect(gene_m, [nut_1,nut_2,nut_3]).
```

The closed loop learning continues until all trials have been tried (i.e. there is no more `phenotypic_effect/2` facts for `gene_m`). So that after 7 cycles the training-set contains all of the above examples and is exactly the same as the test set for `gene_m` which has been used to measure the predictive accuracy. In other words, we have a small training-set (which is added one example in each cycle) and a small test-set which are equal after 7 cycles. Hence, the test-set cannot be expected to provide a safety check against overfitting the training set. This problem is especially important for ASE-Progol, because the assumption "only one trial per cycle" (or only one training example per cycle) increases the risk of overfitting the training data. This could be even worse if we have noise in the training data. To have a more reliable validation test, one approach is to use a separate test-set which is distinct from the training-set.

Another drawback of the existing experimental method is that in each experiment more than one `code/2` facts (5, 9 and 13) are removed from the model while ASE-Progol is given only examples of a single gene. In other words, in this method we try to measure the predictive accuracy of ASE-Progol on a model with 5, 9 and 13 missing `code/2` facts, while we know that ASE-Progol can learn at most one `code/2` fact each time [2]. One alternative to this method could be removing only one `code/2` fact in each experiment, trying to recover the model by ASE-Progol and then measuring the predictive accuracy of the recovered model on the complete set of examples. In this case, the predictive accuracy only reflects the degree of recovery by ASE-Progol and is not affected by the number of missing `code/2` facts.

[2]Note that in the experiments in [6], Progol5.0 is given examples of different genes and therefore removing more than one `code/2` fact makes sense.

3 Conclusion

In this report we have reproduced the results of phase A of the Closed Loop Machine Learning project. According to the previous reports of the project [1, 3], the main purpose of the experiments in phase A have been to test whether ASE-Progol can efficiently converge to accurate hypotheses.

In our reproduction of phase A results, we have used the same experimental method as used in [3] (TEST STRATEGY 1). In addition, we have used a different test strategy to measure the predictive accuracy of ASE-Progol (TEST STRATEGY 2).

The results of both TEST STRATEGY 1 and TEST STRATEGY 2 show that the cost of experimentations is reduced if trials are selected by Closed Loop Machine Learning (CLML) rather than if they are sampled at random. The results of experiments using TEST STRATEGY 1 also show that in all cases ASE-Progol converges to an accurate hypothesis (with an accuracy between 95-100%).

In this report we also evaluated the existing experimental method and provided some suggestions for improving this method.

References

[1] Case for support: Closed loop machine learning. ESPRC Research Proposal GR/M56067, 1998.

[2] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. *Classification and Regression Trees.* Wadsworth, Belmont, 1984.

[3] C. H. Bryant and S. H. Muggleton. Closed loop machine learning. Technical Report YCS 330, University of York, Department of Computer Science, Heslington, York, YO10 5DD, UK., 2000.

[4] C.H. Bryant, S.H. Muggleton, S.G. Oliver, D.B. Kell, P. Reiser, and R.D. King. Combining inductive logic programming, active learning, and robotics to discover the function of genes. In *Machine Intelligence 18.* Electronic Transactions in Artificial Intelligence, 2001. (in press).

[5] T. Mitchell. *Machine Learning.* McGraw-Hill, New York, 1997.

[6] S.H. Muggleton and C.H. Bryant. Theory completion using inverse entailment. In *Proc. of the 10th International Workshop on Inductive Logic Programming (ILP-00)*, Berlin, 2000. Springer-Verlag.