An Evaluation of Ensemble Learning for Software Effort Estimation

Leandro Minku

CERCIA, School of Computer Science, The University of Birmingham

Introduction

Software cost estimation:

- Set of techniques and procedures that an organisation uses to arrive at an estimate.
- Major contributing factor is effort (in person-hours, person-month, etc).
- Overestimation vs. underestimation.

Several software cost/effort estimation models have been proposed.

ML models have been receiving increased attention:

- They make no or minimal assumptions about the data and the function being modelled.

Research Questions

Question 1

Do readily available ensemble methods generally improve effort estimations given by single learners? Which of them would be more useful?

Question 2

If a particular method is singled out, what are the reasons for its better behaviour? Would that provide us with some insight on how to improve software effort estimation?

Question 3

How can someone determine what model to be used considering a particular data set?

Research Questions

Question 1

Do readily available ensemble methods generally improve effort estimations given by single learners? Which of them would be more useful?

Question 2

If a particular method is singled out, what are the reasons for its better behaviour? Would that provide us with some insight on how to improve software effort estimation?

Question 3

How can someone determine what model to be used considering a particular data set?

Research Questions

Question 1

Do readily available ensemble methods generally improve effort estimations given by single learners? Which of them would be more useful?

Question 2

If a particular method is singled out, what are the reasons for its better behaviour? Would that provide us with some insight on how to improve software effort estimation?

Question 3

How can someone determine what model to be used considering a particular data set?

Experimental Design

Learning machines: MLPs, RBFs, RTs, Bagging+MLPs, +RBFs, +RTs, Random+MLPs, NCL+MLPs.

- Databases:
  - Data sets: cocomo81, nasa93, nasa, cocomo2, desharnais, 7 ISBSG organization type subsets.
  - Outliers elimination (K-means) + risk analysis.
- Performance measures:
  - MMRE, PRED and correlation.
  - T-student statistical tests + Wilcoxon tests.
- Parameters:
  - Parameters chosen based on 5 preliminary executions using all combinations of 3 or 5 parameter values.
  - Best MMRE parameters chosen for 30 final runs.

Experimental Design

Learning machines: MLPs, RBFs, RTs, Bagging+MLPs, +RBFs, +RTs, Random+MLPs, NCL+MLPs.

- Databases:
  - Data sets: cocomo81, nasa93, nasa, cocomo2, desharnais, 7 ISBSG organization type subsets.
  - Outliers elimination (K-means) + risk analysis.
- Performance measures:
  - MMRE, PRED and correlation.
  - T-student statistical tests + Wilcoxon tests.
- Parameters:
  - Parameters chosen based on 5 preliminary executions using all combinations of 3 or 5 parameter values.
  - Best MMRE parameters chosen for 30 final runs.

Experimental Design

Learning machines: MLPs, RBFs, RTs, Bagging+MLPs, +RBFs, +RTs, Random+MLPs, NCL+MLPs.

- Databases:
  - Data sets: cocomo81, nasa93, nasa, cocomo2, desharnais, 7 ISBSG organization type subsets.
  - Outliers elimination (K-means) + risk analysis.
- Performance measures:
  - MMRE, PRED and correlation.
  - T-student statistical tests + Wilcoxon tests.
- Parameters:
  - Parameters chosen based on 5 preliminary executions using all combinations of 3 or 5 parameter values.
  - Best MMRE parameters chosen for 30 final runs.

Experimental Design

Learning machines: MLPs, RBFs, RTs, Bagging+MLPs, +RBFs, +RTs, Random+MLPs, NCL+MLPs.

- Databases:
  - Data sets: cocomo81, nasa93, nasa, cocomo2, desharnais, 7 ISBSG organization type subsets.
  - Outliers elimination (K-means) + risk analysis.
- Performance measures:
  - MMRE, PRED and correlation.
  - T-student statistical tests + Wilcoxon tests.
- Parameters:
  - Parameters chosen based on 5 preliminary executions using all combinations of 3 or 5 parameter values.
  - Best MMRE parameters chosen for 30 final runs.

Comparison of Learning Machines

Menzies et al TSE'06 proposes survival selection rules:

- If MMREs are significantly different according to a paired t-test with 95% of confidence, the best model is the one with the lowest average MMRE.
- If not, the best method is the one with the best:
  1. Correlation
  2. Standard deviation
  3. PRED(N)
  4. Number of attributes

Results:

Table: Number of Data Sets in which Each Method Survived. Methods that never survived are omitted.

PROMISE Data	ISBSG Data	All Data
RT: 2	MLP: 2	RT: 3
Bag + MLP: 1	Bag + RTs: 2	Bag + MLP: 2
NCL + MLP: 1	Bag + MLP: 1	NCL + MLP: 2
Rand + MLP: 1	RT: 1	Bag + RTs: 2
	Bag + RBF: 1	MLP: 2
	NCL + MLP: 1	Rand + MLP: 1
		Bag + RBF: 1

Comparison of Learning Machines

What methods are usually among the best?

Table: Number of Data Sets in which Each Method Was Ranked First or Second According to MMRE and PRED(25). Methods never among the first and second are omitted.

(a) Accoding to MMRE

PROMISE Data	ISBSG Data	All Data
RT: 4	RT: 5	RT: 9
Bag + MLP: 3	Bag + MLP 5	Bag + MLP: 8
Bag + RT: 2	Bag + RBF: 3	Bag + RBF: 3
MLP: 1	MLP: 1	MLP: 2
	Rand + MLP: 1	Bag + RT: 2
	NCL + MLP: 1	Rand + MLP: 1
		NCL + MLP: 1

(b) Acording to PRED(25)

PROMISE Data	ISBSG Data	All Data
Bag + MLP: 3	RT: 5	RT: 6
Rand + MLP: 3	Rand + MLP: 3	Rand + MLP: 6
Bag + RT: 2	Bag + MLP: 2	Bag + MLP: 5
RT: 1	MLP: 2	Bag + RT: 3
MLP: 1	RBF: 2	MLP: 3
	Bag + RBF: 1	RBF: 2
	Bag + RT: 1	Bag + RBF: 1

- RTs and bag+MLPs are more frequently among the best considering MMRE than considering PRED(25).
- The first ranked method's MMRE is statistically different from the others in 35.16% of the cases.
- The second ranked method's MMRE is statistically different from the lower ranked methods in 16.67% of the cases.
- RTs and bag+MLPs are usually statistically equal in terms of MMRE and PRED(25).

Risk Analysis – Outliers

How good/bad is the behaviour of these best methods to outliers?

- MMRE usually similar or better than for non-outliers.
- PRED(25) usually similar or worse.

Even though outliers are projects to which the approaches have more difficulties in predicting within 25%, they are not the projects to which the approaches give the worst estimates.

Research Questions – Revisited

Question 1

Do readily available ensemble methods generally improve effort estimations given by single learners? Which of them would be more useful?

- Even though bag+MLPs is frequently among the best methods, it is statistically similar to RTs.
- RTs are more comprehensive and have faster training.
- Bag+MLPs seem to have more potential for improvements.

Why Were RTs Singled Out?

- Hypothesis: As RTs have splits based on information gain, they may work in such a way to give more importance for more relevant attributes.
- A further study using correlation-based feature selection revealed that RTs usually put higher features higher ranked by the feature selection method in higher level splits of the tree.
- Feature selection by itself was not able to always improve accuracy.

It may be important to give weights to features when using ML approaches.

Research Questions – Revisited

Question 2

If a particular method is singled out, what are the reasons for its better behaviour? Would that provide us with some insight on how to improve software effort estimation?

- RTs give more importance to more important features. Weighting attributes may be helpful when using ML for software effort estimation.
- Ensembles seem to have more room for improvement for software effort estimation.

Research Questions – Revisited

Question 3

How can someone determine what model to be used considering a particular data set?

- Effort estimation data sets affect dramatically the behaviour and performance of different learning machines.
- So, it would be necessary to run experiments using existing data from a particular company to determine what method is likely to be the best.
- If the software manager does not have enough knowledge of the models, RTs are a good choice.

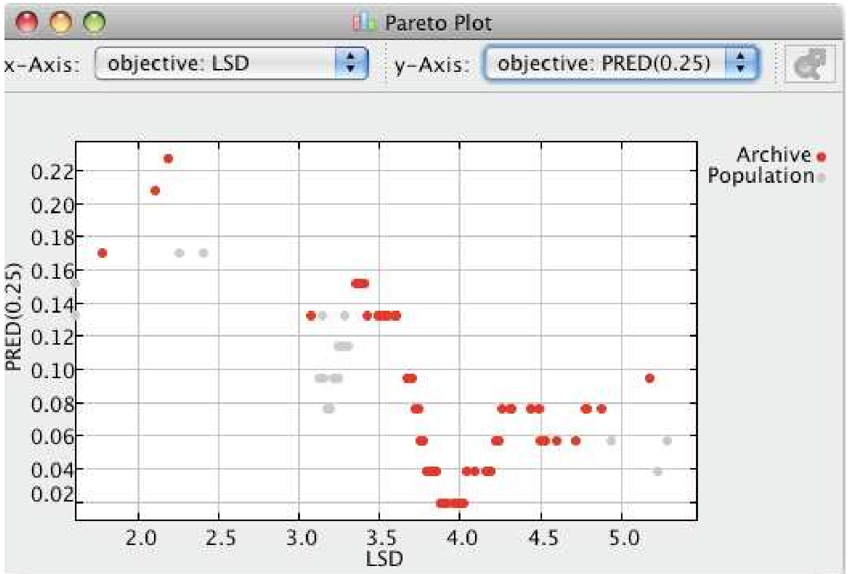
But... What about the Different Performance Measures?

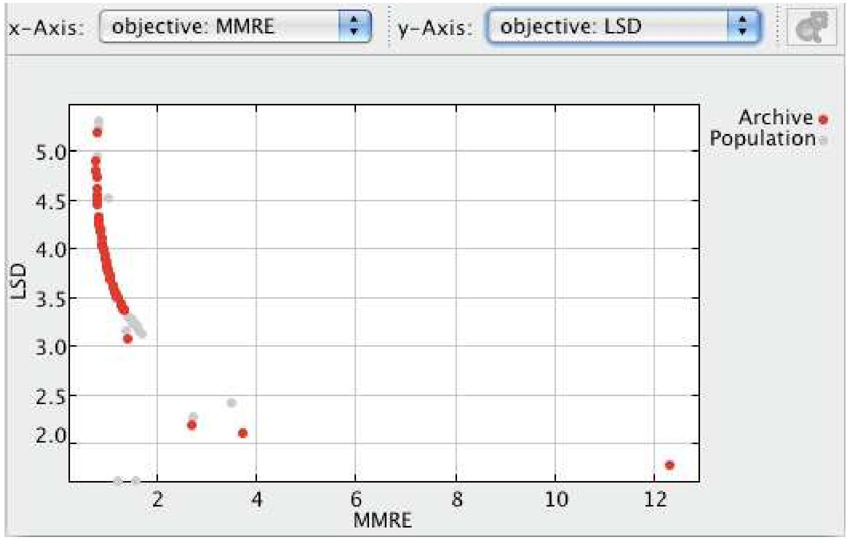
- Better MMRE does not always mean better PRED(25) – outliers show an example.
- Other examples: for Nasa, RTs are ranked 1^{st} in terms of MMRE, but 5^{th} in terms of PRED(25)...
- In general, RTs and bagging+MLPs were usually among the best both in terms of MMRE and PRED(25).
- But, if we have a particular company (set of projects) in hands, is there a most important measure to be considered first?
- Is it possible to get a good trade-off among measures?

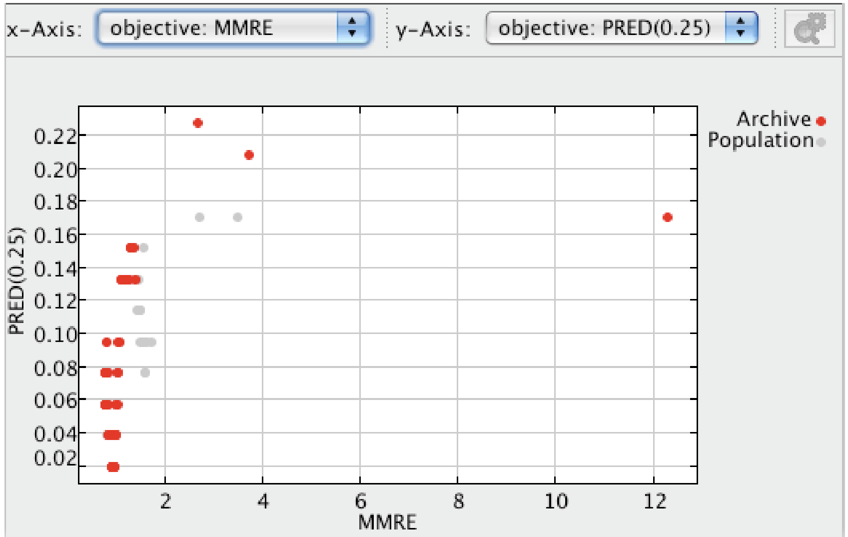
MOEA Approach

- Use MOEA to learn models. E.g., HaDMOEA to learn MLP weights.
- Each objective is a different performance measure (e.g., MMRE, PRED(25), LSD).
- Pareto front may help us to choose a model or a trade-off.
- Pareto front may help us to understand the relationship among performance measures.

Preliminary Results – Cocomo81







- Better LSD, better PRED.
- Improve PRED, similar MMRE. Best PREDs, worst MMREs.
- Better LSD, worse MMRE.

Preliminary Results – Cocomo81

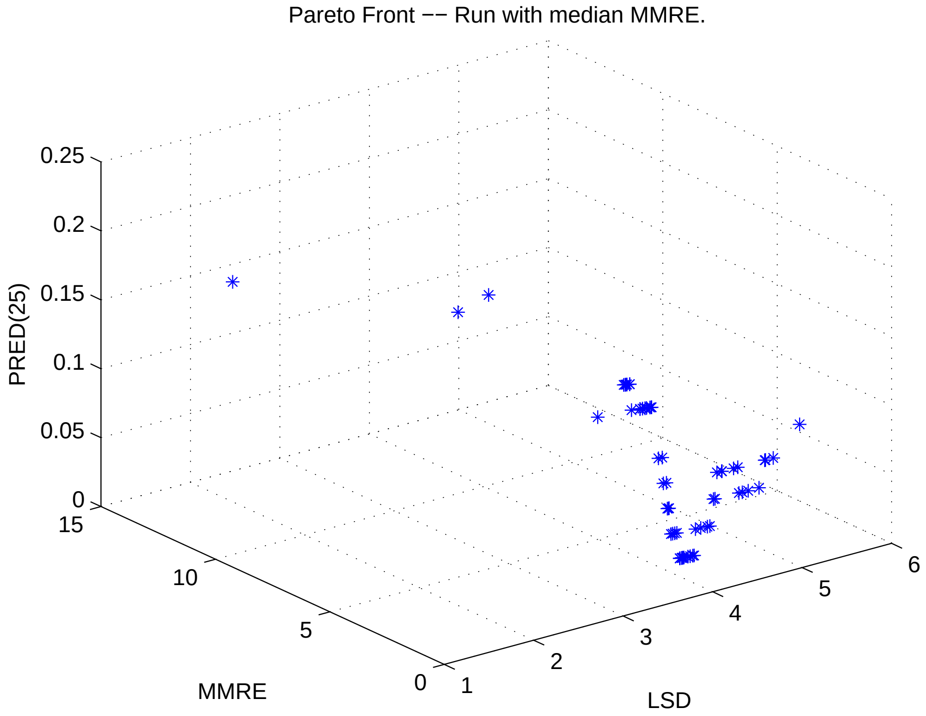


Table: "Ideal" Trade-off vs MLP Results Considering 30 Runs. "Ideal" trade-off: ensemble of the MLPs with the best objective value, for each objective.

	LSD	MMRE	PRED(25)
Ens3	1.91 +- 0.61	2.25 +- 1.77	0.17 +- 0.11
MLP	NaN	2.79 +- 1.67	0.13 +- 0.12

Conclusions and Future Work

Conclusions:

- Evaluation of readily available ensemble methods.
- Insight on how to improve software effort estimation.
- Insight on how to choose a model.

Future work:

- MOEA analysis with more datasets.
- Use insight gained from evaluation to improve software effort estimation.