Ensemble learning for software effort estimation: from static to dynamic solutions

Leandro Minku (`www.cs.bham.ac.uk/~minkull`)

CERCIA, School of Computer Science, The University of Birmingham

May 23, 2013

Introduction – SEE

Software effort estimation (SEE):

- Estimation of the effort required to develop a software project.
- Effort is measured in person-hours, or person-months, etc.
- Based on features such as required reliability, programming language, development type, team expertise, etc.
- Main factor influencing project cost.
- Overestimation vs. underestimation.

Introduction – ML and ensembles

ML for SEE:

- Use completed projects as training data to create SEE models.
- Decision support tools.
- Recent work has shown that ensembles and locality obtain competitive performance.

  Y. Kultur, B. Turhan, and A. Bener. ENNA for estimating software development costs. KBS, 2009.

  E. Kocaguneli, T. Menzies, and J. Keung. On the value of ensemble effort estimation. IEEE TSE, 2012

  T. Menzies et al. Local vs. global lessons for defect prediction and effort estimation. IEEE TSE, 2012

  L. Minku and X. Yao. Ensembles and Locality: Insight on Improving Software Effort Estimation. IST, 2012.
- Most work so far uses offline learning, i.e., static models.

In this talk...

Ensembles of learning machines for SEE:

- A static solution: offline ensemble based on a multi-objective formulation of the problem.

  L. Minku and X. Yao. Software Effort Estimation as a Multi-objective Learning Problem. TOSEM, 2012.

  http://www.cs.bham.ac.uk/~minkull/publications/MinkuYaoTOSEM12.pdf

- A dynamic solution: online ensemble for dealing with online changing environments.

  L. Minku and X. Yao. Can Cross-company Data Improve Performance in Software Effort Estimation?

  PROMISE, 2012.

Multi-objective Ensembles for SEE: a static solution

One of the keys to ensembles' performance:

- Base learners should be <u>diverse</u>.

  G. Brown, J. Wyatt, R. Harris, X. Yao. Diversity creation methods: A survey and categorisation. Information Fusion, 2005.

- They should make different errors on the same examples.

Different performance measures for evaluating SEE models can behave very differently.

- MMRE, PRED, LSD, MAE, etc.

Aim:

- Not to decide on which measure is the best for evaluating SEE models.
- To use the fact that they behave very differently in SEE to create diverse SEE ensembles.

SEE as a Multi-Objective Learning Problem

How to do that?

- SBSE is very related to software prediction systems.

  M. Harman. The relationship between search based software engineering and predictive modeling. PROMISE, 2010.

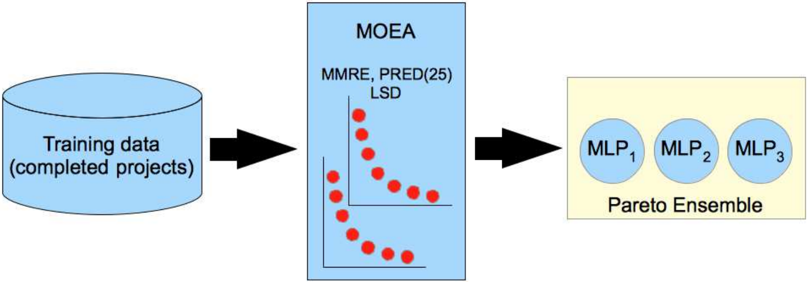
  M. Harman and J. Clark. Metrics are fitness functions too. METRICS 2004.

  F. Ferrucci, C. Gravino, R. Oliveto, and F. Sarro. Genetic programming for effort estimation: an analysis of the impact of different fitness functions. SSBSE, 2010.

- Each performance measure is an objective to be optimised.

- Multi-Objective Evolutionary Algorithm (MOEA) can be used to generate SEE ensembles.

Creating Multi-objective Ensembles for SEE



- MultiLayer Perceptrons (MLPs) are the evolved SEE models.
- Weights and bias represented as real-valued vectors, which are optimised by the MOEA.
- Training performances in terms of MMRE, PRED(25) and LSD guide the optimisation.
- Why these measures? Because they behave very differently.

Experiments

- Data sets: cocomo81, nasa93, nasa, cocomo2, desharnais, 7 ISBSG organization type subsets.
- Performance measures for evaluation on test set: MMRE, PRED(25), LSD, MdMRE, MAE, MdAE.
- Comparing approaches:
  - MLP, RBF;
  - REPTree, Bagging+MLP, Bagging+REPTree, log + EBA;
  - Bagging+RBF, Rand+MLP, NCL+MLP.

Friedman test detected statistically significantly different performance across data sets.

Comparison Against Other Approaches

Number of times ranked best:

Approach	LSD	MMRE	PRED(25)	MdMRE	MAE	MdAE
Pareto Ens	1	6	5	5	5	7
RT	4	3	2	3	2	2
Bag+RT	5	0	1	0	2	1
Bag+MLP	0	2	2	2	1	1
Log + EBA	0	2	2	2	1	1
Bag+RBF	3	0	1	0	2	0
Rand+MLP	0	0	0	1	0	0
RBF	0	0	0	0	0	1
Total	13	13	13	13	13	13

Pareto ensemble is more often ranked first than other approaches for all performance measures, except LSD.

Pareto ensemble is never ranked worst more than twice.

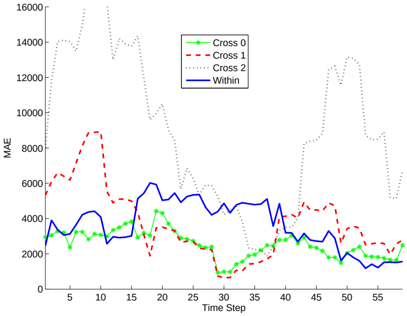
SEE Changing Environments

- Most SEE work does not consider the chronology of the projects.
- However, SEE operates in online changing environments.

  C. Lokan and E. Mendes. Applying moving windows to software effort estimation. ESEM, 2009.

  L. Minku and X. Yao. Can Cross-company Data Improve Performance in Software Effort Estimation? PROMISE, 2012.
- Changes can affect the performance of the SEE models.

SEE Changing Environments



ISBSG2001

Concept drifts can make:

- Models that were useful become obsolete.
- Models that were not useful become helpful.

How to go from static to dynamic solutions?

Online Ensembles for SEE: a dynamic solution

One of the keys to ensembles' performance:

- Base learners should be diverse.
- Diversity is even more important in changing environments.

  L. Minku, A. White and X. Yao. The Impact of Diversity on On-line Ensemble Learning in the Presence of Concept Drift. IEEE TKDE, 2010.

- Diverse ensembles can be composed of models that do well in different <u>concepts</u>.

Online Ensembles for SEE: a dynamic solution

One of the keys to ensembles' performance:

- Base learners should be diverse.
- Diversity is even more important in changing environments.

  L. Minku, A. White and X. Yao. The Impact of Diversity on On-line Ensemble Learning in the Presence of Concept Drift. IEEE TKDE, 2010.

- Diverse ensembles can be composed of models that do well in different concepts.

Different companies behave differently.

- Offline SEE studies show that cross-company learning leads to similar or worse performance than single-company learning.

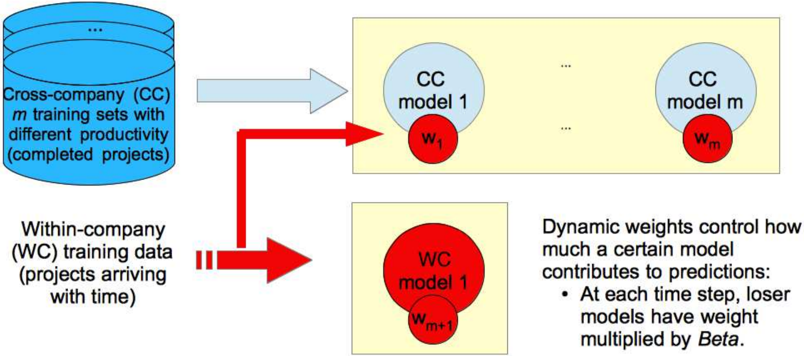
  B. Kitchenham, E. Mendes, and G. Travassos. Cross versus within-company cost estimation studies: A systematic review. IEEE TSE, 2007.

- However, changes in a company may cause it to behave more similarly to other companies (previous slide).

Creating Online Ensembles for SEE

Aim: To use cross-company models to create diverse ensembles able to dynamically adapt to changes.

Dynamic Cross Company Learning (DCL):



Experiments

Approaches:

- WC Regression Trees (RTs).
- DCL.
- WC Dynamic Weighted Majority (WC-DWM) – online ensemble approach.
- CC-DWM – first trained on existing CC data and then on WC data stream.

Data sets:

- 3 ISBSG and 2 Promise.

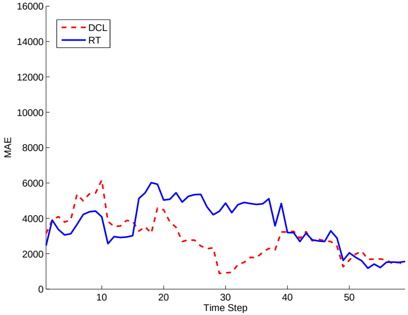
Performance measures:

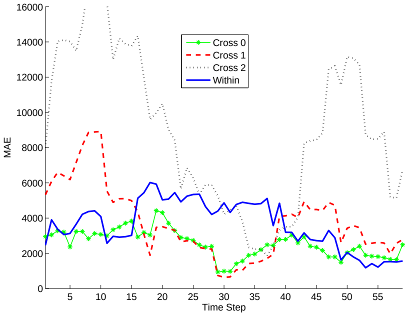
- Mean Absolute Error (MAE) over next ten projects, and overall MAE across time steps.

Friedman test detected statistically significantly different MAE across data sets. Only DCL could improve upon RT.

Sample Result

Performance at each time step DCL vs RT.





ISBSG'2001 (≈ 1.5 year)

- Considerable improvement over periods of several months.
- Improvements are mostly in line with the best performance among the base models.
- There is still some room for improvement.

Conclusions

- Ensembles can be useful not only for offline, but also for online SEE.
- We have seen how to create:
  - Static SEE ensembles based on multi-objective learning (Pareto ensemble).
  - Dynamic SEE ensembles based on multi-company data (DCL).
- Both ensembles have shown to perform comparatively well for SEE.
- Cross-company learning can improve performance.
- There is still room for improvement in the Pareto ensemble in terms of model choice.
- There is still room for improvement in DCL in terms of weight update.

References

L. Minku and X. Yao. Software Effort Estimation as a Multi-objective Learning Problem, **ACM Transactions on Software Engineering and Methodology**, 2012 (accepted). Final author's version available at: www.cs.bham.ac.uk/~minkull/publications/MinkuYaoTOSEM12.pdf

L. Minku and X. Yao. Can Cross-company Data Improve Performance in Software Effort Estimation?, **Proceedings of the 8th International Conference on Predictive Models in Software Engineering (PROMISE'2012)**, p. 69-78, Lund, Sweden, September 2012.