On the Terms Within- and Cross-Company in Software Effort Estimation

Leandro L. Minku
Department of Computer Science, University of Leicester, LE1 7RH, Leicester, UK
leandro.minku@leicester.ac.uk

ABSTRACT

Background: the terms Within-Company (WC) and Cross-Company (CC) in Software Effort Estimation (SEE) have the connotation that CC projects are considerably different from WC projects, and that WC projects are more similar to the projects being estimated. However, as WC projects can themselves be heterogeneous, this is not always the case. Therefore, the use of the terms WC and CC has been questioned as potentially misleading and possibly unhelpful. **Aims:** to raise awareness of the SEE community in terms of the problems presented by the terms WC and CC, and to encourage discussions on the appropriateness of these terms. **Method:** existing literature on CC and WC SEE is discussed to raise evidence in favour and against the use of these terms. **Results:** existing evidence suggests that the terms WC and CC are helpful, because distinguishing between WC and CC projects can help the predictive performance of SEE models. However, due to their connotation, they can be misleading and potentially lead to wrong conclusions in studies comparing WC and CC SEE models. **Conclusions:** the issue being tackled when investigating WC and CC SEE is heterogeneity, and not the different origins of the software projects per se. Given that the terms WC and CC can be misleading, researchers are encouraged to discuss and consider the problems presented by these terms in SEE papers. Labelling projects as "potentially homogeneous" and "potentially heterogeneous" may be safer than directly labelling them as WC and CC projects.

CCS Concepts

•**Computing methodologies** → *Supervised learning by regression;* •**Software and its engineering** → *Software creation and management;*

Keywords

Software effort estimation, cross-company effort estimation, transfer learning, machine learning

PROMISE 2016, September 09 2016, Ciudad Real, Spain

ISBN 978-1-4503-4772-3/16/09…$15.00

DOI: http://dx.doi.org/10.1145/2972958.2972968

1. INTRODUCTION

Software Effort Estimation (SEE) based on Machine Learning (ML) has been investigated by the research community for many years. ML approaches can be used to build SEE models based on a set of completed projects, referred to as the training set. Each completed project is referred to as a training project, and is represented by a set of input features describing the project (e.g., size, programming language, staff expertise, etc) and a target (i.e., the effort required to develop the project). Once an SEE model is built, it can provide estimations for future projects.

One of the challenges faced by ML for SEE is the fact that the predictive performance of ML models depends on the size of the training set. A training set that is not sufficiently large [2] results in poor ML models. Given that the training set proceeding from Within a given Company (WC) is typically small and expensive to collect, much research has been devoted to investigating the use of Cross-Company (CC) training projects. CC projects are projects proceeding from different companies.

Nevertheless, the use of CC projects is itself challenging, because the relationship between the available input features and effort in one company may be different from the relationship in another company. There are various possible reasons for that. For example, if different companies give different interpretations to data collection guidelines, the same project could be described by different input features in different companies. If different companies adopt largely different practices not captured by the available input features, then projects conducted considerably differently could be described by the same input features. The direct use of a training set containing examples that represent different relationships between input features and targets can hinder the predictive performance of ML models [2, 3].

As different companies may or may not present different relationships between input features and effort, one may think that it is not surprising that some studies found CC SEE models to perform worse than WC SEE models, and others found them to perform similarly [4, 7]. If the relationship presented by the WC and CC projects is different, SEE models built by directly using all available CC projects would perform worse than WC SEE models. Otherwise, such CC SEE models would perform similarly. However, this is not the only possible reason for the usually similar or worse predictive performance obtained by CC SEE models. Another possible reason is that WC projects may themselves be heterogeneous, despite the connotation of this term implying the opposite.

The term 'heterogeneity' is typically used to refer to differences in the input features describing the projects, but it could also refer to differences in the relationship between input features and effort. Heterogeneity among WC projects may occur as a result, e.g., of a company having different departments adopting widely different practices or employing staff with considerably different backgrounds. Therefore, when a study comparing CC against WC SEE models concludes that these models perform similarly, there is a possibility that the performance was similar not because the relationship between input features and effort was the same in the CC and WC projects, or because the CC SEE models were successful in addressing the differences between WC and CC projects. The performance may have been similar because the WC projects were so heterogeneous, that they behaved like if they were CC projects. This should be carefully considered when comparing WC and CC SEE models.

The possibility of WC projects being themselves heterogeneous raises questions in terms of the appropriateness of using the terms WC and CC to distinguish between projects in SEE. If WC projects can be heterogeneous enough to be considered CC projects, these terms can be misleading. Researchers may inadvertently think that CC SEE models were useful for having obtained similar predictive performance to that of WC SEE models, when in fact they may have been comparing models that could be both considered as CC SEE models, due to the heterogeneity of their training projects. Therefore, this is a significant issue to be discussed. Moreover, if separating WC and CC projects does not help improving predictive performance, the use of these terms could even be unhelpful. In a recent CREST Open Workshop attended by researchers from the area of predictive modelling in software engineering (http://crest.cs.ucl.ac.uk/cow/44/), a few senior researchers also expressed concerns that these terms may be inappropriate, despite having been used for many years in the field.

With that in mind, this position paper aims at raising awareness of the wider community in terms of the problems presented by the terms WC and CC in SEE, and encouraging discussions on the appropriateness of these terms. Sections 2 and 3 discuss existing literature in order to raise evidence in favour and against the use of these terms. Evidence found in favour was mainly related to the helpfulness of separating WC and CC projects. Evidence against was mainly in terms of misleading conclusions that these terms may imply. Tables 1 and 2 summarise the arguments used and ML approaches considered in this study. Section 4 concludes the paper with final remarks regarding the past and future use of these terms. This paper is focused on conventional software projects. A discussion based on the web effort estimation literature is left as future work.

2. STANDARD ML APPROACHES

This section discusses the terms WC and CC in view of existing work that uses CC training projects for building SEE models based on standard ML approaches. 'Standard' here refers to approaches from the ML literature that have not been modified for the specific purpose of SEE. In particular, an approach that uses stepwise linear regression would be considered standard even after applying log and removing influential projects associated to high Cook's distance, as this is a standard procedure adopted by many applications of linear regression. Table 2 provides a list of standard approaches that, according to previous systematic literature reviews [4, 7], were used in studies with conclusive results comparing WC and CC SEE models on conventional projects.

Due to space constraints, this section does not discuss existing work from this category in detail, but concentrates on analysing the terms WC and CC based on such work. For a comprehensive review of existing work on this type of approach, the reader is refered to [4, 7]. As shown by these review papers, previous work in this category found CC SEE models to achieve similar or worse performance than WC SEE models. Some of these studies used CC training projects to augment their existing WC training sets in an attempt to improve predictive performance, e.g., [6]. Others used only CC training projects to build the CC SEE models, e.g., [1]. The benefit of doing so is the following. If the CC SEE model achieves at least similar predictive performance to that of WC SEE models *and such WC SEE models perform well enough*, then single companies can save the effort of collecting WC training projects.

2.1 Arguments Against the Terms WC and CC

As explained in section 1, even though the terms WC and CC have the connotation of more similar and more different projects, projects coming from a single company may be heterogeneous. The fact that the terms WC and CC by themselves do not make that explicit can be misleading. For instance, a CC SEE model performing similarly to a WC SEE model is typically considered as a positive result. Four out of the six conclusive studies on conventional projects identified by Kitchenham et al. [4] showed that CC SEE models performed similarly to WC SEE models. However, as WC projects can themselves be heterogeneous, the reasons for the similar performance are usually unclear. Potential reasons include the following:

1. the CC projects were similar to the WC projects, making them useful for estimating WC projects, or
2. the CC projects were considerably different from the WC projects, but the underlying ML approach was successful in tackling such differences, or
3. the heterogeneity of the WC projects was high, causing them to behave like if they were CC projects.

Reasons (1) and (2) mean that CC projects were helpful for estimating projects of a given company, which is a positive result. However, reason (3) means that CC projects were not helpful. Therefore, any results indicating a similar performance between WC and CC SEE models should be analysed and interpreted with caution.

Given that these issues are mainly related to the heterogeneity of projects, the main issue that we are trying to tackle is not necessarily the use of projects from different companies, but the use of heterogeneous projects.

2.2 Arguments For the Terms WC and CC

Even though the main issue to be tackled is the use of heterogeneous projects, separating projects according to their company of origin could be helpful. Two out of the six conclusive studies on conventional projects identified by Kitchenham et al. [4] found WC SEE models to be better than CC SEE models. This suggests that, in some cases, the distinction between WC and CC will successfully separate projects that are heterogeneous with respect to the projects being

Table 1: Summary of arguments

Arguments against the terms WC and CC	Arguments for the terms WC and CC
- The real issue we are trying to tackle is heterogeneity, but CC projects are not necessarily more heterogeneous. - Similar performance between WC and CC models may not mean that CC models were successful. - It is unclear what should be considered as WC and CC projects. - It may be possible to propose an adapted approach that successfully handles heterogeneity independent of the company of origin.	- Standard CC models sometimes perform worse than WC models, i.e., it may be safer to separate CC and WC projects. - Separating projects according to company of origin is sometimes successful in capturing some heterogeneity. - No adapted approach so far has been always successful in tackling heterogeneity without distinguishing projects based on companies of origin. - Existing adapted approaches were able to sometimes improve predictive performance in comparison to WC models by separating WC and CC projects.

Table 2: Summary of ML approaches

Standard	Adapted
- Ordinary least squares regression [4, 7] - Stepwise regression [4] - Stepwise ANOVA [4, 7] - Robust regression [4, 7] - CART with stepwise regression in the leaves [4, 7] - CART [4, 7] - k-nearest neighbour [4, 7]	- Relevancy filtering [5] - DCL [8] - Dycom [9]

estimated. Therefore, separating WC and CC projects can avoid hindering the performance of the resulting SEE model, despite not helping to reduce the amount of WC projects that need to be collected for building SEE models.

It is also worth mentioning that a worse predictive performance obtained by a CC SEE model results from the CC projects being considerably different from the WC projects and the underlying ML approach being unable to treat that. This means that the CC projects were more different from the WC projects than the WC projects are different among themselves, and not that there is no heterogeneity among the WC projects. Separating WC and CC projects can be helpful in this case, and dealing with the heterogeneity among WC projects could further help improving SEE.

3. ADAPTED ML APPROACHES

This section discusses the terms WC and CC in light of the literature on approaches that have been modified within the software engineering community in order to achieve better predictive performance in SEE. These approaches have been proposed to cope with the heterogeneity of SEE projects, and are listed in table 2.

Kocaguneli et al. [5] used a relevancy filtering mechanism to tackle heterogeneity. This mechanism creates binary trees to represent training projects and provide SEEs. The variance of the efforts of projects associated to subtrees of the binary trees are analysed. Projects corresponding to subtrees of high variance are filtered out, as they are likely to incur poor SEEs. Experiments showed that CC SEE models obtained similar predictive performance to WC SEE models in 19 cases and worse in 2. However, CC conventional projects were considered as projects of different types (e.g., embedded, organic) or projects proceeding from different centres of the same company. This means that the WC and CC conventional projects actually came all from the same company, i.e., the terms WC and CC were used loosely. It is unknown whether this approach would still achieve similar results if the CC conventional projects came from entirely different companies. When applied to web effort estimation, CC models obtained similar performance to WC models in 6 cases and worse in 2, in terms of mean absolute error [5].

Dynamic Cross-company Learning (DCL) is an adapted ML approach that uses an ensemble of CC and WC SEE models [8]. The WC SEE model is trained only on WC projects, whereas different CC SEE models are built to represent CC projects with different levels of productivity. DCL then attempts to automatically identify, over time, how well the CC and WC SEE models represent the relationship between input features and effort in the company being estimated. The models belived to best represent this relationship have their estimations emphasised in order to improve SEE. This approach was tested with regression trees on five datasets [8]. The experiments showed that using CC projects in addition to WC projects can lead to improvements over the predictive performance of WC SEE models.

Another approach called Dycom was proposed with the aim of reducing the number of WC projects needed for training [9]. This approach also separates WC and CC models, and creates different CC models for different levels of productivity. However, it learns functions to map the relationship between input features and effort represented by the CC SEE models into the relationship currently presented by the single company being estimated. In a study using regression trees and five datasets, Dycom achieved similar or slightly better predictive performance than WC SEE models, while using much less WC training projects [9].

3.1 Arguments Against the Terms WC and CC

Existing literature does not have much discussion in terms of what should be considered as a WC or a CC project. If a company has branches in the USA and India, should projects from these branches be considered WC projects? Or CC projects? If the WC projects are themselves heterogeneous, should some of them be considered as CC projects? Determining what projects should actually be considered WC or CC is not necessarily straightforward and may depend on experts' knowledge about their company. Therefore, some authors have started to use the term loosely, meaning that CC projects could actually be projects from the same company as the WC projects [5].

While being a possible way to deal with the unclear meaning of these terms, this could be misleading if it results in CC projects being less heterogeneous than projects coming strictly from different companies. For instance, the fact that CC SEE models using relevancy filtering achieved similar predictive performance to WC SEE models [5] may imply that relevancy filtering was successful in tackling heterogeneity. However, as the CC projects used in those experiments came from the same company as the WC projects, this may not be the case. Further analysis is needed to check whether the similar predictive performance was not a result of the CC projects coming from the same company as the WC projects. Moreover, further analysis is also needed to check whether the similar performance was not a result from the WC projects themselves being very heterogeneous, as discussed in section 2.1.

Filtering approaches are the approaches that most support dropping the terms CC and WC. As WC projects can themselves be heterogeneous, filtering attempts to treat them in the same way, without distinguishing them from each other. If there is no need for distinguishing them, then there is no need for using the terms WC and CC either. This would mean that these terms are not only misleading, but also would loose their helpfulness explained in section 2.2. However, additional analyses are still needed to further support this argument by checking if relevancy filtering is really successful in tackling heterogeneity, as explained above.

The approach Dycom could potentially be used without distinguishing between WC and CC projects. Heterogeneity could be dealt with by separating all training projects according to productivity, no matter if they are WC or CC projects. If this modified version of Dycom is successful at tackling heterogeneity without distinguishing between WC and CC projects, the terms WC and CC would become unnecessary. Such investigation is left as future work.

3.2 Arguments For the Terms WC and CC

From the studies above, there is some evidence to suggest that filtering can help tackling potential differences between the projects being predicted and the training projects. However, filtered CC SEE models sometimes performed worse than WC SEE models [5]. This supports the use of the terms WC and CC. The fact that DCL and Dycom achieved similar or better performance than WC SEE models by explicitly distinguishing between WC and CC projects [8, 9] also favours the use of these terms. It shows that separating WC and CC data can help improving predictive performance. It is worth noting that this does not mean that the heterogeneity within the WC dataset should not be tackled.

4. CONCLUSIONS

Many studies indicate that predictive performance can be improved by separating WC from CC data [4, 7, 9, 8]. This means that the CC projects were considerably different from WC projects in those cases. As distinguishing between WC and CC projects can be helpful for predictive performance in SEE, this suggests that the terms WC and CC are helpful. However, at times, WC projects may be as heterogeneous as CC projects. And, at times, CC projects may be similar to WC projects [8]. Therefore, the terms WC and CC have to be interpreted and used with caution. They can be misleading if people interpret CC as different and WC as homogeneous with respect to the projects being estimated. Instead, CC projects are projects that we believe have good potential to be different from the WC projects being estimated, but they may not be. WC training projects are projects believed to be more homogeneous with respect to the WC projects being estimated, but they may not be.

Overall, when investigating CC and WC projects in SEE, what we are really trying to tackle is heterogeneity. Separating WC and CC projects is only helpful because CC projects are projects that we believe are likely to be heterogeneous, even though they may not be. Therefore, the topic of research being addressed when investigating WC and CC projects is not the topic of WC and CC itself, but the topic of heterogeneity. A good measure of heterogeneity that would allow us to completely drop the terms WC and CC is yet to be proposed in the literature. However, given that the terms WC and CC can be misleading, researchers are hereby urged to discuss and consider the fact that WC projects can themselves be heterogeneous in SEE research papers. In fact, labelling projects as "potentially homogeneous" and "potentially heterogeneous" may be safer than directly labelling them as WC and CC projects, as it would make this issue explicit.

The work presented in this paper can be extended in order to provide a systematic literature review; to discuss the field of web effort estimation; and to investigate whether a modified version of Dycom and another filtering approach used in the context of web effort estimation [10] could further support arguments against the terms WC and CC.

5. ACKNOWLEDGEMENTS

I would like to thank the participants of the 44th CREST Open Workshop and the anonymous reviewers of a paper that is currently under review for their insightful comments. I would also like to thank the anonymous reviewers of this paper for their constructive comments.

6. REFERENCES

[1] L. Briand, T. Langley, and I. Wieczorek. A replicated assessment of common software cost estimation techniques. In *ICSE*, pages 377–386, Limerick, 2000.

[2] V. S. Cherkassky and F. Mulier. *Learning from Data: Concepts, Theory, and Methods*. John Wiley & Sons, Inc., New York, 1998.

[3] G. Ditzler, M. Roveri, C. Alippi, and R. Polikar. Learning in nonstationary environments: A survey. *IEEE CIM*, 10(4):12–25, 2015.

[4] B. Kitchenham, E. Mendes, and G. Travassos. Cross versus within-company cost estimation studies: A systematic review. *IEEE TSE*, 33(5):316–329, 2007.

[5] E. Kocaguneli, T. Menzies, and E. Mendes. Transfer learning in effort estimation. *ESE*, 20(3):813–843, 2015.

[6] M. Lefley and M. Shepperd. Using genetic programming to improve software effort estimation based on general data sets. In *GECCO*, pages 2477–2487, Chicago, 2003.

[7] S. G. McDonell and M. Shepperd. Comparing local and global software effort estimation models – reflections on a systematic review. In *ESEM*, pages 401–409, Madrid, 2007.

[8] T. Menzies, E. Kocaguneli, L. Minku, F. Peters, and B. Turhan. *Sharing Data and Models in Software Engineering*, chapter How to Adapt Models in a Dynamic World. Morgan Kaufmann, USA, 2014.

[9] L. L. Minku and X. Yao. How to make best use of cross-company data in software effort estimation? In *ICSE*, pages 446–456, 2014.

[10] B. Turhan and E. Mendes. A comparison of cross- versus single-company effort prediction models for web projects. In *SEAA*, pages 285–292, 2014.