A study on selecting similar projects in software effort estimation

Kosuke Ito †
Department of Management System Engineering
Tokyo Metropolitan University, Tokyo, Japan
Tel: +81-42-585-8606, Email: Ito-kosuke1@ed.tmu.ac.jp

Xiao Xiao †
Department of Management System Engineering
Tokyo Metropolitan University, Tokyo, Japan
Tel: +81-42-585-8462, Email: xiaoxiao@tmu.ac.jp

Hisashi Yamamoto
Department of Management System Engineering
Tokyo Metropolitan University, Tokyo, Japan
Tel: +81-42-585-8674, Email: yamamoto@tmu.ac.jp

Abstract. In recent years, software development process has become diverse and complex. Analogy based software effort estimation (ABE) selects projects similar to a target project from software development historical data, and calculates the effort of the target project using the effort of selected similar projects. ABE is a widely used estimation method because this method can reflect the individuality of a project. However, it is not decided clearly that which similarity measure should be used in the selection of similar projects. Furthermore, since different similarity measures select different projects, the selection of similarity measures directly affects the estimation accuracy. This paper proposes a method that regards projects selected by multiple similarity measures as similar projects. The performance of multiple similarity measures-based method is compared with that of single similarity measure-based method using actual software development historical data.

Keywords: software effort estimation, analogy based estimation, Euclidean distance, weighted Euclidean distance, cosine similarity

1. INTRODUCTION

Estimating software development effort at the early stage of the software development process is necessary to let software development project succeed. COCOMO (Boehm, 1 981) is one of the classical software effort estimate methods. COCOMO estimates effort by using the scale of software. However, high-quality estimation is very difficult by using this kind of classical method because that recently software has become diverse and complex. Those methods that can reflect individuality of the project are required from such a background.

It is known that analogy based software effort estimation (ABE) (Ohsugi et al., 2004; Mendes et al., 2003) is one of the methods that enables reflection of the individuality. ABE goes through two procedures to predict effort. First, look for similar projects resemble target project from historical data. Second, calculate the effort of the target project by using the actual effort of similar projects. It is clear that ABE is such a method that can enable high-quality estimation because it uses only similar projects. Ohsugi et al. (2004) points that collaboration filtering, which is one kind of ABE, is superior to conventional stepwise regression when the loss rate of data is relatively high. Mendes et al. (2003) shows that case based reasoning, which is also one kind of ABE, is superior to conventional methods such as regression tree (Rokach and Maimon, 2008) and stepwise regression.

As described above, there are a number of studies i ndicating that ABE is a high-quality estimation method, but only a few studies focus on the issue of how and what to choose the similarity measure for ABE. Euclidea n distance (ED) is the most popular similarity measure which is applied to ABE. It measures the physical distance between target and similar projects. However, it is difficult to find the difference of diverse projects by using only such basic similarity measures. Example of similarity measure besides distance includes cosine similarity (COS). COS regards projects as a vector, and measures similarity using the cosine of the angle between two vectors (two projects). The selected similar projects depend on whether to use "distance" or "angle" of the vectors (the projects). However, previous studies always apply single similarity measure to ABE. Using both the distance and the angle of the vectors (the projects) at the same time may help to find such projects that truly resemble the target project.

This paper proposes a method that regards projects selected by multiple similarity measures as similar projects, and compare it with conventional ABE that uses single similarity measure. The remaining part of this paper is organized as follows. Section 2 introduces the related work on ABE methods. Section 3 proposes multiple similarity measures-based effort estimation method. Section 4 describes the experiments and reports the results. Section 5 suggests some future works and concludes this paper.

2. ANALOGY BASED EFFORT ESTIMATION

2.1 Case Based Reasoning

Case based reasoning (CBR) is a method for the purpose of the solutions to the problem, and it is studied in the field of artificial intelligence. Mukhopadhyay et al. (1992) proposes *Estor* as a model of CBR, and shows that it has higher accuracy than COCOMO and function point method. However, *Estor* has a weak point that it depends on the expert who select the similar projects of target project.

Shepperd and Schofield (1997) proposes a model of CBR which does not depend on the expert. Their method is to select three similar projects and consider the algebraic average of the efforts of similar projects to be the predicted effort. In addition, ED (Euclidean distance) is applied as similarity measure. The ED from target project a to historical project p is expressed in equation (1).

$$ED_{ap} = \sqrt{\sum_{j=1}^{n}(x'_{aj} - x'_{pj})^2}. \tag{1}$$

Here, x'_{ij} is the normalized value of feature j $(j = 1,2,\dots,n)$ of project i $(i = 1,2,\dots,m)$.

Mendes et al. (2003) compares the performance of CBR methods where three similarity measures are applied. They conclude that weighted Euclidean distance (WED) is the best similarity measure among ED, WED and maximum measure. The WED from target project a to historical project p is expressed in equation (2).

$$WED_{ap} = \sqrt{\sum_{j=1}^{n} w_j (x'_{aj} - x'_{pj})^2}. \tag{2}$$

Here, w_j is the weight of feature j. It is decided by its correlation coefficient of effort. It becomes $w_j = 2$ if the correlation coefficient between feature j and effort is high. Otherwise, it becomes $w_j = 1$.

2.2 Collaboration Filtering

Collaboration filtering (CF) is a general technique of the famous recommendation system which recommends item to users in EC (electronic commerce) sites. The basic thinking of CF is that two users with a similar evaluation to an item will do similar evaluation for other items. For example, if both user A and user B like item C, and user A also like item D, then CF recommends item D to user B.

Ohsugi et al. (2004) apply CF to software effort estimation and show that CF is superior to conventional stepwise regression in the data that has high loss rate. Specifically, they employ COS (cosine similarity) as similarity measure instead of ED. COS is the similarity measure calculating the angle of the vectors (the projects) but not physical distance. The COS between target project a and historical project p is expressed in equation (3).

$$COS_{ap} = \frac{\sum_{j=1}^{n}(x'_{aj} \times x'_{pj})}{\sqrt{\sum_{j=1}^{n}(x'_{aj})^2}\sqrt{\sum_{j=1}^{n}(x'_{pj})^2}}. \tag{3}$$

3. MULTIPLE SIMILARITY MEASURES-BASED EFFORT ESTIMATION

This section proposes four multiple similarity measures-based effort estimation methods that consider both the distance and the angle of the vectors (the projects). In this paper, we employ ED and WED as distance measure, while COS as angle measure.

In addition, because the scale of values of historical project features are different, we normalize the data. The normalized value of actual value x_{ij} is expressed in equation (4).

$$x'_{ij} = \frac{x_{ij} - \min(x_j)}{\max(x_j) - \min(x_j)}, \tag{4}$$

where $\max(x_j)$ and $\min(x_j)$ means the maximum and minimum value of feature j among m projects.

3.1 Commonly Selection Method

Commonly selection method (CSM) is the method that considers relative importance of similar projects. Important project is selected as a similar project in both senses of distance measure (ED or WED) and angle measure (COS). In other words, important project is the project that its distance to the target project is short, while at the same time, its angle to the target project is small. CSM calculates the effort of target project by using the weighted mean of similar projects. Let feature b of a project be its effort, then the effort of target project a, say $\hat{x}_{ab}$, is calculated as shown in equation (5).

$$\hat{x}_{ab} = \frac{\sum_{p \in similar\ projects}(x_{pb} \times w_p \times amp(a,p))}{\sum_{p \in similar\ projects} w_p}. \quad (5)$$

Here, x_{pb} is feature b of similar project p, *i.e.*, the effort of similar project p. w_p is the weight of similar p roject p. If similar project p is selected by both distance and angle measures, then $w_p = 2$. If similar project p is selected by distance measure, then $w_p = 1$. If similar pro ject p is selected by COS, then $w_p = 0$. $amp(a,p)$ divi des the scale of target projects a by the scale of similar project p. $amp(a,p)$ between target project a and historic al project p is expressed in equation (6).

$$amp(a,p) = \frac{x_{as}}{x_{ps}}. \quad (6)$$

Here, s is a feature index that stands for scale of s oftware. We use recorded function point or number of li nes of code as the scale of software. We can revise the differences between the scale of target project a and sim ilar project p by using $amp(a,p)$. We show the procedur e of CSM below.

1. Select $k1$ number of projects which are the most si milar ones to target project a based on distance me asure.
2. Select $k2$ number of projects which are the most si milar ones to target project a based on angle meas ure.
3. Set $w_p = 2$ if certain project p is selected by both distance and angle measures. Set $w_p = 1$ if certain project p is selected by only distance measure. Set $w_p = 0$ if certain project p is selected by only angl e similarity.
4. Calculate $\hat{x}_{ab}$ using equation (5).

3.2 Synthetic Measure Method

Synthetic measure method (SMM) is the method tha t considers distance measure and angle measure to be on e similarity measure. Synthetic similarity measure betwee n target project a and historical project p is expressed in equation (7).

$$sim(a,p) = \frac{distance\ measure}{COS_{ap}}. \quad (7)$$

Here, *distance measure* means ED_{ap} expressed in eq uation (1), and WED_{ap} expressed in equation (2). COS_{ap} is expressed in equation (3).

CSM calculates the effort by including important simi lar projects as well as not so important similar projects. On the other hand, SMM selects k projects which are th e most similar ones to target project using synthetic simi larity measure. We show the procedure of SMM below.

1. Select k number of projects which are the most si milar ones to target project a based on synthetic si milarity measure.
2. Calculate $\hat{x}_{ab}$ using equation (8).

$$\hat{x}_{ab} = \frac{\sum_{p=1}^{k}(x_{pb} \times amp(a,p))}{k}. \quad (8)$$

3. EVALUATION

This section explains the evaluation method and evaluation results. For better understanding, we show the candidates of estimation accuracy comparison in Figure 1.

4.1 Data Set

We use four data sets for evaluations: (i) Albrecht (Albrecht and Gaffney, 1983), (ii) Kemerer (Kemerer, 1987), (iii) Desharnais (PROMISE Software Engineering Repository) and (iv) Kitchenham (Kitchenham, 2004). Table 1 lists the properties of these data sets. Each data set is different in sample size, and features of each project are recorded. We exclude the project data that has missing values, and such features that are considered to be inappropriate in software effort estimation are also excluded.

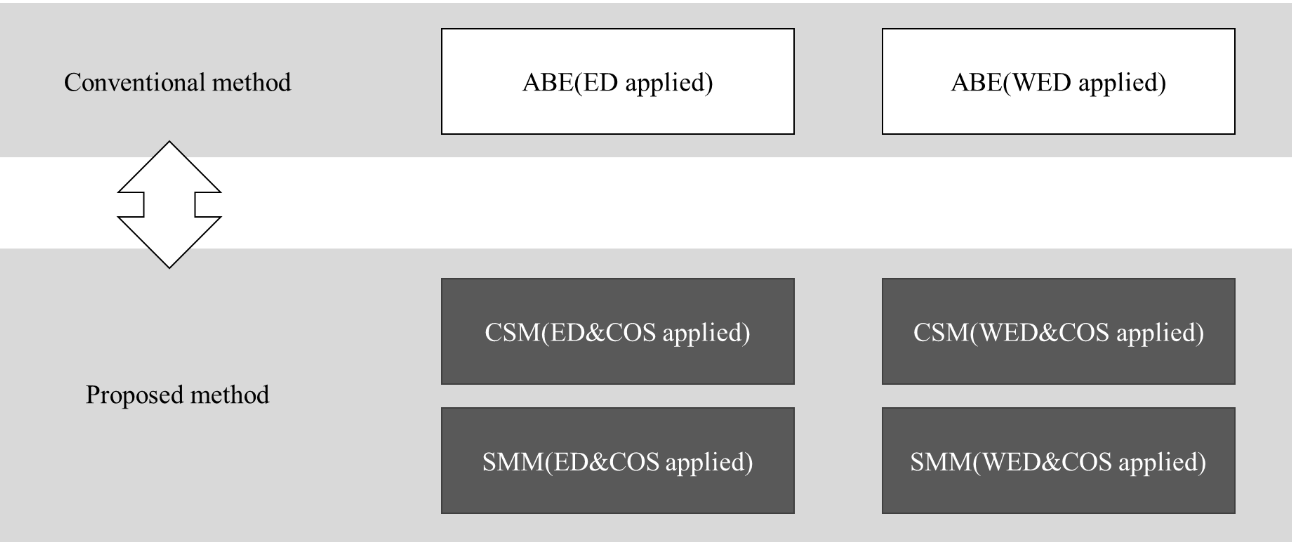

Figure 1: Candidates of comparison.

Table 1: Datasets

Data set	Number of projects	Number of features	Effort mean	Effort median	Effort min	Effort max
Albrecht	19	7	2515.8	1290.0	0.5	105.2
Kemerer	15	3	219.2	130.3	23.2	1107.3
Desharnais	77	6	4833.9	3542.0	546.0	23940.0
Kitchenham	135	3	3169.1	1557.0	219.0	113930.0

4.2 Evaluation Procedure

We use leave-one-out cross-validation method for evaluation. The procedure is shown below. For each data set:

1. Select a target project a from m number of project data, and consider other projects as the historical data.
2. Select similar projects which resemble target project a from historical data by CSM, SMM or ABE. In this paper, we set $k1 = k2 = k = 3$.
3. Calculate the predictive effort of target project a.
4. Carry out procedure 1~3 for all a $(a = 1,2,\dots,m)$.

4.3 Evaluation Criteria

We use Pred25, MBRE (mean of balanced relative error) and MdBRE (median of balanced relative error) as evaluation criteria of the predictive accuracy. MBRE is the mean of BRE (balanced relative error), and MdBRE is the median of BRE. The BRE of project *i* is expressed in equation (9).

$$BRE_i = \frac{|y_i - \widehat{y_i}|}{\min(y_i, \widehat{y_i})}. \tag{9}$$

Here, y_i is actual effort of project *i*. $\hat{y}_i$ is predictive effort of project *i*.

Generally, Pred25 expresses the ratio of the number of projects that MRE (magnitude of relative error) is less than 0.25 to the number of the overall projects m. However, MRE does an unbalanced evaluation for an excessive prediction and an under prediction (Molokkenostvold and Jorgensen, 2005). Thus, we use BRE which can balance evaluation as evaluation criteria in this paper. Pred25 is expressed in equation (10).

$$\text{Pred25} = \frac{\sum_{i=1}^{m} isAccurate(BRE_i)}{m} \times 100, \tag{10}$$

Table 2: Correlation between features and effort.

(a) Albrecht data set.

Features	Function points	OUT	SLOC	INQ	FILE	IN	Language dummy variable1	Language dummy variable2
Correlation coefficients	0.943	0.898	0.859	0.851	0.769	0.662	-0.191	0.050

(b) Kemerer data set.

Features	KSLOC	Software dummy variable1	Months	Software dummy variable2	Software dummy variable3
Correlation coefficients	0.722	0.323	0.219	-0.206	-0.157

(c) Desharnais data set.

Features	Points non adjust	Length	Envergure	Team experience	Manager experience	Language dummy variable1	Language dummy variable2
Correlation coefficients	0.725	0.653	0.417	0.259	0.160	0.161	0.041

(d) Kitchenham data set.

Features	Adjusted function points	Actual duration	Project type dummy variable1	Project type dummy variable2	Project type dummy variable3	Project type dummy variable4	Project type dummy variable5
Correlation coefficients	0.982	0.593	0.142	-0.119	-0.024	-0.023	-0.017

where *isAccurate* (BRE_i) is expressed in equation (11).

$$isAccurate(BRE_i) = \begin{cases} 1 & BRE_i \leq 0.25 \\ 0 & BRE_i > 0.25 \end{cases}. \qquad (11)$$

4.4 Preliminary Experiment

Preliminary experiment is conducted to decide w_j which is the weight of feature j when using WED as similarity measure. Table 2 is a list of correlation coefficients between each feature and effort for all the four data sets. As a result, the feature which has strongest correlation to effort is the feature to express the scale of the project in all data set. Therefore, we set the weight of "Function points (in Albrecht data)", "KS LOC (in Kemerer data)", "Points non adjust (in Desharn ais data)" and "Adjusted function points (in Kitchenham data)" to be $w_j = 2$, and that of the other features to be $w_j = 1$.

4.5 Results

Table 3 and Table 4 show the comparison results of estimation accuracy of conventional methods and propos ed methods. In Table 3 and Table 4, ABE(ED) and AB E(WED) represents the conventional ABE methods where

Table 3: Results in Albrecht and Kemerer data.

Albrecht Data							
Method	Pred25 (%)	MBRE	MdBRE	Method	Pred25 (%)	MBRE	MdBRE
ABE(ED)	42.1	0.954	0.405	ABE(WED)	31.6	**0.938**	0.403
CSM(ED&COS)	42.1	0.971	0.405	CSM(WED&COS)	42.1	1.062	0.391
SMM(ED&COS)	42.1	0.954	0.405	SMM(WED&COS)	**47.4**	0.956	**0.311**
Kemerer Data							
Method	Pred25 (%)	MBRE	MdBRE	Method	Pred25 (%)	MBRE	MdBRE
ABE(ED)	6.7	0.869	**0.730**	ABE(WED)	6.7	0.885	**0.730**
CSM(ED&COS)	13.3	0.864	**0.730**	CSM(WED&COS)	13.3	0.897	**0.730**
SMM(ED&COS)	**20.0**	**0.758**	**0.730**	SMM(WED&COS)	**20.0**	0.761	**0.730**

Table 4: Results in Desharnais and Kitchenham data.

Desharnais Data							
Method	Pred25 (%)	MBRE	MdBRE	Method	Pred25 (%)	MBRE	MdBRE
ABE(ED)	37.7	0.495	0.361	ABE(WED)	39.0	**0.486**	**0.359**
CSM(ED&COS)	37.7	0.504	0.377	CSM(WED&COS)	**40.3**	0.496	0.377
SMM(ED&COS)	36.4	0.500	0.375	SMM(WED&COS)	37.7	0.498	0.411
Kitchenham Data							
Method	Pred25 (%)	MBRE	MdBRE	Method	Pred25 (%)	MBRE	MdBRE
ABE(ED)	28.9	1.137	0.565	ABE(WED)	**30.4**	**1.107**	0.565
CSM(ED&COS)	28.9	1.159	0.568	CSM(WED&COS)	29.6	1.133	**0.549**
SMM(ED&COS)	28.9	1.160	0.565	SMM(WED&COS)	**30.4**	1.131	0.565

ED or WED is employed as the similarity measure. While CSM(ED&COS), CSM(WED&COS), SMM(ED&COS) and SMM(WED&COS) represents the proposed methods where corresponding similarity measures are employed. The bold-faced letters show that the best method in each evaluation criteria in each data set.. For example, SMM (ED&COS)'s MBRE is best value in Kemerer data.

In Table 3, proposed method is superior to conventional method in many cases. Particularly, Pred25 is greatly improved by using SMM. On the other hand, in Table 4, proposed method is inferior to conventional method a little in some cases. These results show that estimation accuracy turns worse a little by using proposed method in Desharnais and Kitchenham data.

Table 3 and Table 4 show that estimation accuracy of the proposed method greatly varies according to the difference of data set. Table 5 is a list of variance of efforts and features to express the scale of the project in each data set. Table 5 shows that the data which showed high-quality estimation of SMM is the high-variance data.

Table 5: List of variance of efforts and features.

Data set	**Variance of effort**	**Variance of feature to express scale**
Albrecht	0.084	0.093
Kemerer	0.057	0.111
Desharnais	0.032	0.031
Kitchenham	0.008	0.008

Table 6: Difference in selected projects between ABE(WED) and SMM(WED&COS) (Kemerer data).

Project No.	KSLOC	Effort	Effort/KSLOC	Note
15	60.2	69.9	1.16	Target project
10	39.0	72.0	1.85	selected by both methods
2	40.5	82.5	2.04	selected by both methods
6	50.0	84.0	1.68	selected by only ABE(WED)
13	161.4	157.0	0.97	selected by only SMM(WED&COS)

Table 7: Difference in selected projects between ABE(WED) and SMM(WED&COS) (Desharnais data).

Project No.	Points non adjust	Effort	Effort/(Points non adjust)	Note
17	108	3192	29.56	Target project
50	131	3136	23.94	selected by both methods
20	86	840	9.77	selected by both methods
48	192	5817	30.30	selected by only ABE(WED)
74	297	2800	9.43	selected by only SMM(WED&COS)

We consider for a hypothesis that the proposed method leaves a good result when using high-variance data. Table 6 shows the difference of the similar projects between conventional and proposed methods in Kemerer data. Looking at proportion Effort/KSLOC, we can see that project No. 13 which is selected by only SMM(WED&COS) shows nearer value to the target project No. 15 than project No. 6 which is selected by only ABE(WED). This difference is considered to be the reason that, BRE obtained from SMM(WED&COS) improves 0.377 compared with the case of ABE(WED) in effort estimation of project No.15. Table 7 shows the difference of the similar projects between conventional and proposed methods and in Desharnais data. However, the result is reverse to Table 6. Project No.74 that shows far values of proportion Effort/(Points non adjust) is selected by only SMM(WED&COS). In this data set, BRE obtained from SMM gets 0.544 worse compared with the case of using ABE(WED) in effort estimation of project No.17. From these results, it can be concluded that the SMM tends to select truly similar projects especially in high-variance data.

5. CONCLUSION

This paper proposed a multiple similarity measures-based software effort estimation method. We compared it with conventional analogy-based estimation method through real data analysis. As a result, we concluded that SMM could execute high-quality estimation in high-variance data. However, as a result of having obtained it in this paper, it was the result that obtained from only four data sets. We will apply the proposal to more data sets to ensure its effectiveness and the lessons learned here.

REFERENCES

Albrecht. A and Gaffney. J (1983) " Software function, source lines of code and development effort prediction: a software science validation, " *IEEE Transactions on Software Engineering*, **9**, 639-648.

Boehm. W. B (1981), "Software engineering economics," *Prentice Hall*.

Kitchenham. B, Pfleeger. S, McColl. B, and Eagan. S (2004), " An empirical study of maintenance and development estimation accuracy" *Journal of Systems and Software*, **64**, 57-77

Kemerer. C (1983), "An empirical validation of software cost estimation models," *Commun.ACM*, **30**, no.5, 416-429.

Mendes. E, Watson. I, Triggs. C, Mosley. N and Counsell. S, (2003), "A comparative study of cost estimation Models for web hypermedia applications," *Empirical Software Engineering*, **8,** 163-196.

Molokkenostvold. K and Jorgensen. M (2005), "A comparison of software project overruns-flexible versus Sequential Development Models", *IEEE Transactions on Software Engineering*, **31**, no. 9, 754-766.

Mukhopadhyay. T, Vicinanza. S and Prietula. M (1992), "Examining the feasibility of a case-based reasoning model for software effort estimation," *MIS Quarterly*, **16**, no.2, 155-171.

Ohsugi. N, Tsunoda. M, Monden. A, and Matsumoto. K (2004), ``Effort estimation based on collaborative filtering,'' *In Proc. of International Conference on Product Focused Software Process Improvement*, **5**, 274-286.

PROMISE software engineering repository, *http://promise.site.uottawa.ca/SERepository/*

Rokach, L., and Maimon, O. (2008), "Data mining with decision trees: theory and applications", World Scientific Pub Co Inc.

Shepperd. M and Schofield. C (1997), "Estimating software project effort using analogies," *IEEE Transactions on Software Engineering*, **23**, no. 12, 736-743.