Utilizing Cluster Quality in Hierarchical Clustering for Analogy-Based Software Effort Estimation

Jack H.C. Wu
College of Professional and Continuing Education
The Hong Kong Polytechnic University
Hung Hom, Hong Kong
hcjwu@speed-polyu.edu.hk

Jacky W. Keung
Department of Computer Science
City University of Hong Kong
Kowloon Tong, Hong Kong
Jacky.Keung@cityu.edu.hk

Abstract— **Analogy-based software effort estimation is one of the most popular estimation methods. It is built upon the principle of case-based reasoning (CBR) based on the *k*-th similar projects completed in the past. Therefore the determination of the *k* value is crucial to the prediction performance. Various research have been carried out to use a single and fixed *k* value for experiments, and it is known that dynamically allocated *k* values in an experiment will produce the optimized performance. This paper proposes an interesting technique based on hierarchical clustering to produce a range for *k* through various cluster quality criteria. We find that complete linkage clustering is more suitable for large datasets while single linkage clustering is suitable for small datasets. The method searches for optimized *k* values based on the proposed heuristic optimization technique, which have the advantages of easy computation and optimized for the dataset being investigated. Datasets from the PROMISE repository have been used to evaluate the proposed technique. The results of the experiments show that the proposed method is able to determine an optimized set of *k* values for analogy-based prediction, and to give estimates that outperformed traditional models based on a fixed *k* value. The implication is significant in that the analogy-based model will be optimized according the dataset being used, without the need to ask an expert to determining a single, fixed *k* value.**

Keywords-Software Metrics and Measurements; Software Effort Estimation; Analogy; K-NN; Clustering

I. INTRODUCTION

In software engineering, software effort estimation (SEE) by analogy has been known to provide reasonable prediction accuracy, and it has been subject to intensive research for many years. One of the influential factors determining the prediction performance of analogy-based effort estimation is the value of *k,* in which it will consider *k*-many similar projects completed in the past to base estimate, traditionally this has been arbitrarily determined by experts or using different dataset training methods. The objective of this study is to determining the most appropriate value of *k* for each individual case in a training dataset, so the analogy-based model is further optimized.

In this paper, we introduce a technique that involves hierarchical clustering for determining the value of *k* nearest neighbor for analogy-based software effort estimation. For conventional analogy-based effort estimation, the value of *k* is fixed for all testing cases. This has been shown to be far from optimal in previous studies [1]. However, determining the best value of *k* for each testing case is time consuming. We propose to use hierarchical clustering with specific cluster quality criteria to determine the value of *k* efficiently and automatically.

For a suitable clustering algorithm, we studied and compared complete-linkage, single-linkage, and average-linkage hierarchical clustering algorithms for determining an appropriate value of *k* based on the size of the cluster in which the testing case that belongs to. For the stopping condition of the clustering algorithms, we adopted five cluster quality criteria. Each has its own merits and preference in producing clusters with certain characteristics (e.g., small specific clusters versus large general clusters).

The rest of the paper is organized as follows. Section 2 outlines the research work, which is related to finding the best value of *k* for analogy-based cost estimation. Section 3 describes our proposed approach involves choosing a clustering algorithm, determining the number of clusters using different cluster quality criteria as the stopping condition for the clustering algorithm and determining the range of *k.* In Section 4, experiment results on various datasets from the PROMISE repository are reported. Finally, Section 5 concludes the paper.

II. BACKGROUND

A. *Analogy-based Effort Estimation*

Analogy-based effort estimation for software engineering has been studied for a long time and shown success [2]. The premise of analogy-based estimation (ABE) is based on the principle: "Projects that are similar with respect to a set of project features will also be similar with respect to effort" [3].

Those projects completed, and that are similar to the new target project under investigation will be utilized as candidate estimator to produce effort estimates via different solution adaptation methods. However given n project cases in a dataset, one of the main challenges in analogy-based effort estimation is to correctly determine the number of required reference cases or projects that exhibits close similarity to the target project.

Four major steps are usually performed in analogy-based effort estimation [2]: (1) *k* similar training cases are identified

978-1-5386-0497-7/17/$31.00 ©2017 IEEE

using some distance metric (e.g., Euclidean distance); (2) use the information from the *k* similar cases identified in the previous step to estimate effort for the current test case; (3) adaptation is performed to increase the accuracy of the estimation; and (4) retain the estimation setting for future application.

The value *k* can be either static or dynamic. For the ABE model trained by a static *k*, it is set to a default value which is usually between the 1 to 5 according to studies in software effort estimation literature [4-7], the same value of *k* holds for all the testing cases in a dataset assuming all the training case *i*, and its nearest *k* cases will be used. For dynamic *k*, for each case *i*, its nearest k_i will be different, and so is the prediction performance generated by that ABE model [8]. In the dynamic approach, it is suggested that *k* should be adjusted according to the task under investigation [9].

The Theoretical Maximum Prediction Accuracy (TMPA) [1] for the best value of *k* has been proposed as an optimal performance measure where models can be built to reach this threshold. It is understood that TMPA is obtained in the ideal situation in which the best value of *k* is used for each individual project. While our aim is to consider TMPA as the goal and produce estimations as close to TMPA as possible, we propose a more practical way to estimate *k* using hierarchical clustering algorithms and incorporating a randomized process such that each project case can have its own *k*.

B. Clustering technique in ABE

According to a study by Azzeh et al. [10], Bisecting *k*-medoids clustering algorithm is used to obtain a better prediction for *k*. It uses a centroid-based clustering algorithm which groups similar individual cases within a dataset into clusters. The Bisecting *k*-medoids algorithm recursively applies the basic *k*-medoids algorithm until the stopping condition is met. The result will be a hierarchical binary tree structure. However, in [10], only the compactness (1) is used for measuring the quality of the clusters, and it is the only measure used as the stopping condition for the recursive algorithm. Note that (1) is only an example for measuring compactness:

$$Compactness = \frac{1}{T}\sum_{i=1}^{m}\sum_{j=1, x_j \in C_i}^{T}\left\|x_j - C_{ic}\right\|^2 \quad (1)$$

where *m* is the number of clusters, x_j is the feature vector for the *j*th case in the cluster C_i, C_{ic} is the centroid vector for the cluster C_i, and ||●|| is the Euclidean distance. In fact, cluster quality can be measured by both *compactness* and *isolation* [11]. Based on the idea of compactness and isolation above, Raskutti and Leckie [12] proposed four different criteria for accessing the quality of clusters for a clustering algorithm which is applied to document retrieval. They are: (a) Minimum Total Distance; (b) Separated Clusters; (c) Object Positioning; and (d) Number of Objects Correctly Positioned.

III. RANDOMIZED-K HIERARCHICAL CLUSTERING

This section introduces our proposed approach in determining the most appropriate value of *k* for each individual case in a training dataset. In our approach, we utilize hierarchical clustering algorithm to help us to determine *k*. Various cluster quality criteria are incorporated during clustering in order to produce the best result. Moreover, a randomized technique is incorporated into the procedure in order to increase its flexibility and efficacy.

A. Clustering Algorithms

Clustering is related to many disciplines and plays an important role in a broad range of applications [13]. Here we use clustering algorithms to determine the number of cases that is similar to the testing case (i.e. the number of analogies should be used to estimate the effort). We will use hierarchical clustering instead of the *k*-means clustering, because in k-means clustering one has to determine the value of *k* first, which could be another challenge adds to the complexity. Therefore we use hierarchical clustering to obtain the structure of the dataset in terms of a hierarchical binary tree.

B. Cluster Quality Criteria in Hierarchical Clustering for ABE

When performing hierarchical clustering, in the beginning, each project is in a cluster of its own, the clusters are then sequentially combined into larger clusters. A hierarchical binary tree structure is formed after the clustering. In the highest level, the entire dataset is considered as one single cluster. While in the lowest level, every single project is considered as one cluster. Therefore we need a mechanism to determine the suitable number of clusters in the dataset. This involves the determination of the cluster quality in each of the iterations. We adopted the four cluster quality criteria in [12] to access the stopping condition of the clustering algorithm.

In particular, we tested a total of five different methods to determine the stopping condition, after which the number of clusters in the training set is obtained. They are: (a) Fixed Threshold; (b) Minimum Total Distance; (c) Separated Clusters; (d) Object Positioning; and (e) Number of Objects Correctly Positioned.

C. Determine the Feasible Range of k

After determining the number of clusters using one of the cluster quality criteria, we are able to determining the size of the cluster in which the testing project i is located. The size of the cluster provides us an initial guess for the value of k. However, it may not be the optimal value (i.e., TMPA). We allow a range to be formed according to the initial value by increasing and decreasing the initial value by r. The value of r is tested in our experiments using various datasets.

D. Application Procedure for Analogy

Overall, our proposed approach can be summarized into the following steps: (1) Calculate the distances between every pair of the projects; (2) For a testing project *i*, perform a hierarchical clustering on the nearest n projects, n depends on the size of the dataset; (3) Define an interval for the value of *k* based on the size of the cluster which project *i* belongs; (4) Use a random number generated within the interval range in Step (3) as the value of *k* in the analogy based estimation; (5) Perform post-processing of the k-nearest project to estimate the

effort of project *i*. The mean effort of the k-nearest project becomes the estimated effort of the testing project.

As our focus in this paper is to study different hierarchical clustering with different cluster quality criteria, we do not perform feature subset selection which would further complex the problem. While it is well-known that performing feature subset selection would benefit the effectiveness [14], it would probably increase the effectiveness of both the fixed k and also our technique. Moreover, the feature subset selection technique suitable for fixed k may be different from the feature selection subset technique when clustering is involved. Therefore for increasing the fairness and clearness of the comparison, feature subset selection is not performed in this study.

IV. EMPIRICAL EXPERIMENTS

This section reports the experiment results applying our technique to determine the value of *k* in analogy-based effort estimation. There are four main components that we would like to test in our proposed technique. First component is the clustering scheme, i.e., among the three hierarchical clustering algorithm (complete-, single-, and average-linkage) which of them would be the best for the estimation of *k*. Second component is the five different cluster quality criteria (described in Section 3.2). Third component is whether Euclidean distance or squared Euclidean distance is better when clustering is involved. Note that there are two steps in our technique which involves the distance metric. When we reach the nearest *n* projects for clustering and during the clustering. The distance metric used in these two steps may not be the same. Because clustering algorithms depend heavily on the distance metric in order to obtain a good result. The last component is whether a randomized *k* is better than that without randomization. Our objective is to obtain a fully automated scheme, which can determine the value of *k* efficiently and effectively without the need to search for the optimal value of *k* iteratively.

A. Datasets

We use various datasets from the PROMISE repository. Specifically, six well-known software effort estimation datasets have been used, and their statistics are described in Table I. Among the datasets, the China dataset contains the highest number of projects with large variance, while Nasa93 and Kemerer contain only a small number of projects.

TABLE I. STATISTICS OF THE DATASETS USED.

Dataset	Cases #	Effort min	Effort max	Effort mean
Desharnais	77	546	23940	4833.9
Cocomo81	63	5.9	11400	683.5
Maxwell	62	583	63694	8223.2
China	499	26	54620	3921
Nasa93	18	8.4	824	624.4
Kemerer	15	23.2	1107.3	219.2

B. Evaluation Criteria

We use a leave-one-out cross validation to test our algorithms. For each run, one project *i* is selected to be the test case and all the other projects are training cases. We report and compare our results using the commonly used mean magnitude relative error (MMRE) measure.

Foss et al. [15] in their extensive study on various prediction accuracy metrics concluded that the MMRE performance measure by itself is unreliable in some circumstances and may provide misleading indication, as such, and for the sack of easy interpretation and comparison with existing studies, in addition to MMRE we will use Wilcoxon signed-rank test to statistically confirm their performance indication.

C. Results of Different Clustering Schemes

Using three different hierarchical clustering algorithms with five different cluster quality criteria, each dataset will result in fifteen results for all the combinations. We will use the commonly used Euclidean distance in this set of comparisons. We compare the MMRE results when using different hierarchical clustering algorithms in Table II.

TABLE II. MMRE results of the three hierarchical clustering.

Dataset	Complete	Single	Average
Desharnais	**0.503**	0.507	0.530
Cocomo81	3.092	3.289	**1.616**$^{@\ \#}$
Maxwell	0.797	0.807	**0.779**$^{@\#}$
China	**0.386**$^{\#}$	0.408	0.399
Nasa93	1.460	**1.333**$^{@\$}$	1.509
Kemerer	0.632	**0.630**$^{\$}$	0.667

@ indicates the MMRE result is significantly better than that from complete-linkage clustering with 90% C.I.
\# indicates the MMRE result is significantly better than that from single-linkage clustering with 90% C.I.
$ indicates the MMRE result is significantly better than that from average-linkage clustering with 90% C.I.

From the results in Table II, we observe that the performance of hierarchical clustering is highly related to the size of the datasets. For larger datasets such as China, complete-linkage clustering would produce the best result which is significantly better than the result from single-linkage clustering. For medium-sized datasets such as Cocomo81 and Maxwell, average-linkage clustering produces the best results which are significantly better than both complete-linkage clustering and single-linkage clustering. Finally, for small-sized datasets (i.e., Nasa93 and Kemerer with less than 20 project cases), single-linkage clustering performs the best which is significantly better than average-linkage clustering in both cases.

D. Effect of Randomized-k

In this section we examine the effect of adding Step (4) to the overall procedure. Adding the randomization step will increase the flexibility of the value of k such that it may alleviate the problem when the clustering algorithm does not perform well in a particular dataset.

Table III shows the MMRE results of with and without randomized-k. The value of the range r from 1 to 3 is tested. For the results with randomized-k (i.e., r = 1, 2, 3), the MMRE is the average of 10 repeated runs.

TABLE III. MMRE RESULTS WITH AND WITHOUT RANDOMIZED-K.

Dataset	Without Randomized-*k*	r=1	r=2	r=3
Desharnais	0.503	**0.492**	0.504	0.508
Cocomo81	1.616	**1.614**	1.784	1.955
Maxwell	**0.779**	0.835	0.854	0.878
China	**0.386**	0.388	0.401	0.415
Nasa93	**1.333**	1.359	1.410	1.527
Kemerer	**0.630**	0.633	0.712	0.735

From the results, as r increases, the range of the value of k increases and has a great impact on the performance. However, when r is 1, there is no significant difference between with and without randomized k. If one has to use randomized-k, the value of r should be very small. This coincides with studies within the literature of software effort estimation in that the value of k is confined to a very limited range of values [4-6].

E. Comparison with Fixed-k

Table IV shows that results from our approach and our implementation of the one proposed by [9]. Both methods involve some randomization in the algorithms, our approach induce a randomization on the value of k, which initially comes from the result of the hierarchical clustering. The Baker approach involves randomization in taking the sample of N training projects out of the total T projects. In order to obtain a more stable result, both sets of experiments are repeated 10 times. We also compare the results with fixed k approach with the value of k equals to 1, 3, and 5.

TABLE IV. MMRE RESULTS WHEN USING OUR HIERARCHICAL CLUSTERING TECHNIQUE WITH VARIABLE K, BAKER'S APPROACH, AND FIXED K (=1, 3, 5).

Dataset	Our Method	Baker	Fixed *k* = 1	Fixed *k* = 3	Fixed *k* = 5
Desharnais	**0.492**$^{@\#}$	0.584	0.601	0.500	0.500
Cocomo81	1.614	**1.601**	1.619	3.207	3.200
Maxwell	0.835	0.875	0.914	0.908	**0.796**
China	**0.388**	0.395	0.413	0.400	0.415
Nasa93	**1.359**$^{\#}$	1.386	1.509	1.596	1.610
Kemerer	**0.633**$^{\#}$	0.682	0.736	0.780	0.755

@ indicates the MMRE result is significantly better than that from Baker with 90% C.I.

\# indicates the MMRE result is significantly better than that from all the Fixed *k* with 90% C.I.

Our results are significantly better than that from the fixed k in three out of six datasets. Except for the Maxwell dataset, all of our MMRE results are better than that from fixed k approach. When comparing to the Baker approach, except for the cocomo81 dataset, all of our MMRE results are better. One of the main advantages of our technique over the Baker approach is that we do not need to search for the value of k iteratively. The clustering algorithm together with the suitable cluster quality criteria can output a fully automated value of k, which the results are comparable to the Baker approach.

V. CONCLUSIONS

To conclude, this study proposed a novel approach by applying hierarchical clustering to determine the best values of *k* for analogy-based software effort estimation (ABE) before its model building, largely removes the need to arbitrarily define a fixed *k* in a traditional analogy-based setting. Result shows that the proposed technique utilizing hierarchical clustering effectively determine the most suitable value of *k* for the testing case *i*, which would result in a more optimized ABE model for software effort estimation. The proposed method is well defined and fully automated that is able to determine the value of k for ABE automatically, and thus a major improvement to analogy-based software effort estimation.

REFERENCES

[1] J. W. Keung, "Theoretical Maximum Prediction Accuracy for Analogy-Based Software Cost Estimation," in *Software Engineering Conference, 2008. APSEC '08. 15th Asia-Pacific*, 2008, pp. 495-502.

[2] M. Shepperd and C. Schofield, "Estimating software project effort using analogies," *Software Engineering, IEEE Transactions on*, vol. 23, pp. 736-743, 1997.

[3] J. W. Keung, B. A. Kitchenham, and D. R. Jeffery, "Analogy-X: Providing Statistical Inference to Analogy-Based Software Cost Estimation," *Software Engineering, IEEE Transactions on*, vol. 34, pp. 471-484, 2008.

[4] U. Lipowezky, "Selection of the optimal prototype subset for 1-NN classification," *Pattern Recognition Letters*, vol. 19, pp. 907-918, 8// 1998.

[5] C. Kirsopp and M. Shepperd, "Making inferences with small numbers of training sets," *Software, IEE Proceedings* -, vol. 149, pp. 123-130, 2002.

[6] E. Mendes, I. Watson, C. Triggs, N. Mosley, and S. Counsell, "A Comparative Study of Cost Estimation Models for Web Hypermedia Applications," *Empirical Software Engineering*, vol. 8, pp. 163-196, 2003/06/01 2003.

[7] Y. F. Li, M. Xie, and T. N. Goh, "A study of project selection and feature weighting for analogy based software cost estimation," *Journal of Systems and Software*, vol. 82, pp. 241-252, 2// 2009.

[8] E. Kocaguneli, T. Menzies, A. Bener, and J. W. Keung, "Exploiting the Essential Assumptions of Analogy-Based Effort Estimation," *Software Engineering, IEEE Transactions on*, vol. 38, pp. 425-438, 2012.

[9] D. R. Baker, *A Hybrid Approach to Expert and Model Based Effort Estimation*: ProQuest, 2007.

[10] M. Azzeh and Y. Elsheikg, "Learning Best K analogies from Data Distribution for Case-Based Software Effort Estimation," in ICSEA 2012, *The Seventh International Conference on Software Engineering Advances*, 2012, pp. 341-347.

[11] R. Dubes and A. K. Jain, "Validity studies in clustering methodologies," *Pattern Recognition*, vol. 11, pp. 235-254, // 1979.

[12] B. Raskutti and C. Leckie, "An evaluation of criteria for measuring the quality of clusters," presented at *the Proceedings of the 16th international joint conference on Artificial intelligence - Volume 2*, Stockholm, Sweden, 1999.

[13] P. Berkhin, "A Survey of Clustering Data Mining Techniques," in *Grouping Multidimensional Data*, J. Kogan, C. Nicholas, and M. Teboulle, Eds., ed: Springer Berlin Heidelberg, 2006, pp. 25-71.

[14] Z. Chen, T. Menzies, D. Port, and B. Boehm, "Feature subset selection can improve software cost estimation accuracy," *SIGSOFT Softw. Eng. Notes*, vol. 30, pp. 1-6, 2005.

[15] T. Foss, E. Stensrud, B. Kitchenham, and I. Myrtveit, "A simulation study of the model evaluation criterion MMRE," *Software Engineering, IEEE Transactions on*, vol. 29, pp. 985-995, 2003.