Improving Analogy Software Effort Estimation using Fuzzy Feature Subset Selection Algorithm

Mohammad Azzeh
Department of Computing
University of Bradford
Bradford BD7 1DP, U.K.
M.Y.A.Azzeh@brad.ac.uk

Daniel Neagu
Department of Computing
University of Bradford
Bradford BD7 1DP, U.K.
D.Neagu@brad.ac.uk

Peter Cowling
Department of Computing
University of Bradford
Bradford BD7 1DP, U.K.
P.I.Cowling@brad.ac.uk

ABSTRACT

One of the major problems with software project management is the difficulty to predict accurately the required effort for developing software applications. Analogy Software effort estimation appears well suited to model problems of this nature. The analogy approach may be viewed as a systematic development of the expert opinion through experience learning and exposure to analogue case studies. The accuracy of such model depends on characteristics of datasets. This paper examines the impact of feature subset selection algorithms on improving the accuracy of analogy software effort estimation model. We proposed a feature subset selection algorithm based on fuzzy logic for analogy software effort estimation models. Validation using two established datasets (ISBSG, Desharnais) shows that using fuzzy features subset selection algorithm in analogy software effort estimation contribute to significant results as other algorithms: Hill climbing, Forward subset selection, and backward subset selection do.

Categories and Subject Descriptors

D.2.9 [Software Engineering]: Management—cost estimation.

General Terms

Algorithms, Management, Measurement.

Keywords

Analogy software effort estimation, Feature selection, Fuzzy logic

1. INTRODUCTION

Software project effort estimation is a dynamic field due to the continuous changes in software project development: new types (web-based projects, e-tools, games) and team interaction (off-shoring, networking) challenges increase the number of defining features and also require new modeling techniques. Analogy Software Effort Estimation (ASEE) appears well suited to model problems of this nature. The analogy approach may be viewed as a systematic development of the expert opinion through experience learning and exposure to analogue case studies. It requires identification of a list of main project features the effort estimation will be based upon. Then analogous but completed projects are found, for which the effort is known. However, the accuracy of such models depends on two prime parameters are: selected features and similarity measurement method. This paper focuses on the issue of using feature subset selection to identify optimal feature subset that leads to better estimation accuracy. Chen et al [4] show that accuracy of COCOMO software effort estimation model can be improved by using WRAPPER searching algorithms. To that end, in ASEE, each software project is described by many features which might be discrete, continues and categorical. Using all available features in the estimation process may result in reduction in the overall level of effort estimation accuracy [13] and therefore increasing the severity of project risk because some of these features may be irrelevant, redundant or because of model complexity and additional noise. To overcome this issue, Feature Subset Selection (FSS) algorithms have been reasonably investigated in ASEE. Feature Subset Selection is a process of finding a feature subset that provides a model with similar or better performance than using all available features. There are some potential benefits of using FSS in analogy estimation such as reducing time of training and utilization, improving data understanding and visualization, and reducing data dimensionality [6].

Searching for optimal feature subset is usually characterized into two major types: Wrapper and Filter [1, 13]. Wrapper uses some machine algorithms to assess the fitness of selected feature subset, while Filters uses some statistical approaches to determine the fitness of feature subset. Wrapper starts with empty space, then some feature combination is added to the solution space. Meanwhile the algorithm checks if there is improvement comes from selected attributes. The algorithm stops when there are no more feature combinations or when there has been no better improvement in the model [4, 22]. It has been criticized that Wrapper approach is computationally far more intensive than Filter approach, but can find better feature subset [7, 13]. Hill climbing, simulated annealing, forward and backward selection algorithms are all form of Wrapper approach.

The vast majority of FSS algorithms differ primarily in feature selection strategy and evaluation criteria. Yet, there are no definitive rules or procedures that govern selection of the optimal feature subset [26]. Evaluation criteria are considered the main challenge in deriving highly predictive features. For example, in the case of ASEE, the optimal features subset is selected based on prediction accuracy and the relation between input and output. The one that gives better prediction is considered as highly predictive feature. However, many evaluation criteria have been

PROMISE'08, May 12–13, 2008, Leipzig, Germany.

published in previous studies [15, 16], where new machine learning based induction algorithms were also provided. These algorithms are computationally expensive but maintain a high level of accuracy.

In this paper we proposed searching algorithm for ASEE based on fuzzy c-means (FCM) clustering and fuzzy logic. The FFSS is exhaustive search in nature where all possible solutions are searched, we attempt to evaluate fitness of feature subset based on minimum similarity between fuzzy clusters which take into account information about data structure. Whilst exhaustive search uses MMRE with Jack Knifing Hold-out strategy to assess the fitness of feature subset. Kadoda et al [12] reported that using MMRE and Jack-knifing it is untrustworthy especially when the feature subset is evaluated later by n-fold cross validation. We validate our approach against several searching algorithms implemented in ArchAngel Tool [13] such as: Hill climbing, forward subset selection, backward subset selection and exhaustive search, where two established datasets: ISBSG (release 10) and Desharnais were used in the empirical validation. The rest of the paper is organized as follows: section 2 presents analogy software effort estimation related works. Section 3 introduces fuzzy logic. Section 4 our proposed algorithm is introduced. In Section 5 we present the datasets used in empirical evaluation. In section 6 we discuss the results of using FSS in analogy estimation. Section 7 describes threats to case study validity. Section 8 concludes with an analysis of using FSS algorithms and their impact on ASEE.

2. ANALOGY SOFTWARE EFFORT ESTIMATION

The development of efficient analogy software effort estimation models has been a research target for quite a long time. Several attempts have been carried out to improve the accuracy of analogy effort estimation by using different Artificial Intelligence (AI) techniques such as Neural Networks [8], Evolutionary algorithms [5], and Heuristic algorithms [14]. Auer et al. [2] address the issue of replacing conventional features subset selection by more flexible features dimension weighting. The weight is calculated by scaling weight dimensions from 0 to 1. They used ArchAngel Tool to predict the effort for new project. The results obtained showed that using extensive number of features with weight dimension present better accuracy than using optimal feature list. Conversely, Mendes et al. [21, 20] showed that using few optimal features with adaptation rules gave better accuracy than using all features. Mendes et al [20] showed also how the accuracy of analogy estimation can be improved by using Feature Subset Selection (FSS) in which two datasets with different characteristics have been used in the case study. The first dataset presents a continuous effort function translated as strong linear relationship between size and effort. The second dataset presents no linear relationship between size and effort. Results showed that using feature subset selection with adaptation rules does not always increase accuracy in the dataset with continuous effort function.

There have been a number of studies attempting to investigate the impact of feature selection on ASEE [4, 18, 10, 21, 14, 17, 12, 13]. Kadoda et al.[12] reported that the proper selection of optimal feature subset has significant impact on accuracy of ASEE. Kirsopp et al [13] compared between various FSS algorithms in ASEE, they showed that exhaustive search is computationally far intensive but produces the most predictive features. Hill climbing and random search are also significant when dataset size is too large and prediction accuracy is important. Idre et. al. [10] proposed alternative approach for analogy software effort estimation based on fuzzy logic and linguistic quantifiers. They employed feature weighting based on productivity ratio to identify the relevance of each feature. In some sense, this approach does not have learning ability [9] and the results are not promising. Another study [17] employed weighted Euclidean distance to measure distance between two software projects in order to see the more relevant features to the effort.

3. FUZZY LOGIC

Fuzzy logic [28] provides a representation scheme and mathematical operations for dealing with uncertain, imprecise and vague concepts. It introduces the idea of association of the linguistic terms with degrees such as: Team experience, application size, productivity and so forth - all come on a sliding scale. For instance, when we say *Team experience* is *low*, *Team experience* is a linguistic variable (universe of discourse) while *low* is a fuzzy set. Any element x of universe of discourse *Team experience* belongs to the fuzzy set *low* by a membership function value $\mu_{low}(x)$. For crisp set representations, the element *x* belongs to the set *low* if and only if *x* is one of the elements of the set *low* and is given by membership value one, otherwise zero. Moreover, Fuzzy logic appears well suited to deal with uncertain and imprecise measurement of software metrics [9, 19].

Property 1. Fuzzy set *A* is called normal fuzzy set if it has at least one element x in the universe of discourse whose membership value is unity or height of $A=1$. A fuzzy subset that is not normal is called subnormal [23].

4. FUZZY FEATURE SUBSET SELECTION ALGORITHM

4.1. Approaching Degree

The approaching degree [24] is a method used to assess the similarity between two given fuzzy sets in a particular universe of discourse X [15]. Let assume *S(X)* be a power set of normal fuzzy sets with $A_k^j \neq 0$ and $A_k^j \neq X$. Let *A, B* be two normal fuzzy sets where $A, B \in S(X)$. The similarity degree between two fuzzy sets *A* and *B* is assessed in equation 1:

$$SM(A,B) = \min\Big((A \bullet B), \overline{(A \oplus B)}\Big) \quad \textbf{(1)}$$

where $(A \bullet B)$ the inner product is defined by equation 2 and $(A \oplus B)$ is the outer product defined by equation 3:

$$(A \bullet B) = \max(\min[\mu_A(x), \mu_B(x)]) \ \ , x \in X \quad \textbf{(2)}$$

$$(A \oplus B) = \min(\max[\mu_A(x), \mu_B(x)]) \ \ , x \in X \quad \textbf{(3)}$$

Particularly, when the value of *SM(A,B)* approaches a value 1, this represents that the two fuzzy sets A and B are "more closely similar". When *SM(A,B)* approaches a value 0, the two fuzzy sets are "more dissimilar". Since each fuzzy set represent a cluster we represent similarity between two clusters same to similarity between two fuzzy sets.

The evaluation mechanism used to evaluate the best feature subset is given in the following definitions:

Definition 1. Similarity between two clusters for given *m*-dimensional features:

let C_i be a fuzzy cluster given by $C_i=\{C_{i1},C_{i2},..,C_{im}\}$ where M is the number of features in a particular feature subset, C_{i1} is the membership function for the ith cluster in the feature 1, C_{i2} is the membership function for cluster i in the feature 2 and so forth. Then the similarity between two clusters C_x and C_y is given by:

$$SM(C_x, C_y) = \max_{l=1}^{M}(W_l * SM_{F_l}(C_{xl}, C_{yl})) \qquad , x \neq y \; .$$

$\sum_{l=1}^{M} W_l = 1$, W_l is normalized weighting factor representing the importance of some features among others.

Definition 2. Overall similarity between all clusters in a feature subset S_i is given as*:*

$\mathrm{E}_{s_i} = \min_{j=1}^{N}(SM(C_x, C_y))$ Where N represents number of possible combination between pair of clusters.

Definition 3. Predictive feature subset D_{best} is the one with minimum E_{si}.

4.2 The FFSS Algorithm

The goal of our algorithm is to assess the similarity degree between two given fuzzy clusters and consequently the similarity between all clusters for all selected features. We presume that as long as the similarity degrees between all pairs of clusters for selected features is as small as it can be, then the clustered data are well distributed. Hence, they present predictive features. To explain our approach by example, let us have N data samples described by 3 dimensional features as shown in Figure 1 and distributed using FCM algorithm into 3 clusters as shown in Figure 2. These clusters are fuzzified to fuzzy sets on their universes of discourse as shown in Figures 3, 4 and 5. According to Def. 1, 2 and 3, the similarity between all cluster pairs for a feature subset is given by:

$SM(C1,C2)=max[W_{FA}*SM_{FA}(C1,C2), \; W_{FB}*SM_{FB}(C1,C2), W_{FC}*SM_{FC}(C1,C2)]$

$SM(C1,C3)= max[W_{FA}*SM_{FA}(C1,C3), W_{FB}*SM_{FB}(C1,C3), W_{FC}*SM_{FC}(C1,C3)]$

$SM(C2,C3)= max[W_{FA}*SM_{FA}(C2,C3), W_{FB}*SM_{FB}(C2,C3), W_{FC}*SM_{FC}(C2,C3)]$

Then overall similarity:
E_{si}=min$[SM(C1,C2),SM(C1,C3),SM(C2,C3)]$

After calculating E_{si} for all possible feature subsets, the best feature subset is one with minimum E_{si}

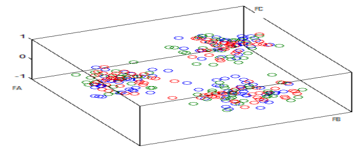

Figure 1: N data sample distributed in 3 dimensions.

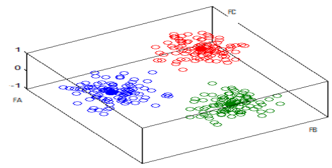

Figure 2: Clusters of N data sample.

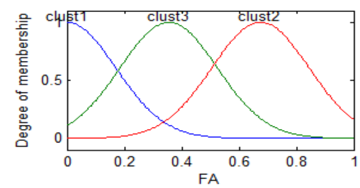

Figure 3: Membership functions for feature A.

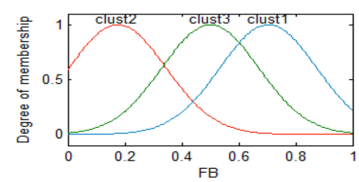

Figure 4: Membership functions for feature B.

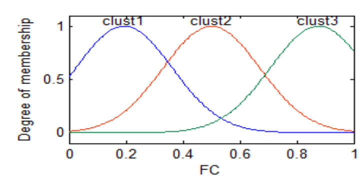

Figure 5: Membership functions for feature C.

Our proposed algorithm is shown in Figure 6 and is described by five main steps:

Step1: select a particular feature subset from available feature space(e.g. S1={F1,F3,F4}) and weight each feature in this subset by finding out value of R-squared that indicates how much the selected feature can explain the output which in our case is the total development effort. R-squared is the fraction of the variation in the value of the observed value that is explained by the regression [19]. The weighing parameter is crucial task in the similarity measures because it takes into account the relevancy of particular features for effort estimation. ***Step 2***: Fuzzify each feature subset using fuzzy identification model based on FCM clustering. To fuzzify clustered dataset, there are two main ways: first is the expert knowledge [27] which is formed in if-then-rules where parameters and memberships are tuned using input and output data. The second is no prior knowledge about the system [27], so the model is built based on particular algorithms. However, the fuzzy model was constructed based on the second approach where membership functions obtained by FCM and projection. ***Step 3***: Assess similarity degree between all pairs of clusters in each feature subset using definition 1. ***Step 4***: Assess overall similarity degree in each feature subset using definition 2.

Step 5. Evaluation: after calculating similarity degrees for all features subsets, we extract the subset that has minimum similarity degree E_{si} according to definition 3. The output of the algorithm is the best feature subset.

Input:
$D(F_1,F_2,...F_M)$//input dataset (NxM)
Out //Output Dataset(Nx1)
Output:
D_{best} //feature subset of high predictive features.
begin
***Do**:*
***Step1**:Select feature subset to be searched S_i.*
***Step2**:Fuzzify the feature subset (S_i).*
***Step3**:For feature subset S_i, assess similarity degree between all pairs of clusters (i.e. fuzzy sets) in all features in S_i.*
***Until all feature subsets are searched**.*
***Step4**:Evaluate each feature subset S_i using E_{si}*
***Step5**:* **Evaluation: best feature subset is one with minimum E_{si}.**
$\min_{i=1}^{k}(E_{s_i})$
End;

Figure 6: The FFSS algorithm

5. DATASETS

The first dataset used is the ISBSG Repository (release 10, January 2007), which currently contains more than 4000 software projects gathered from different software development companies [11]. All Projects involved in the ISBSG repository are described by several numerical and categorical attributes. The second dataset used is Desharnais [3] dataset which is described by 10 features. In order to assess the efficiency of the proposed algorithm we omitted features and projects with missing values.

Table 1: Data set Characteristic

Characteristic	ISBSG	Desharnais
No. of features	14	10
No. of Projects	400	77

5.1 Data Normalization

Data Normalization represents the transformation of attribute values in the range between 0 and 1 so all attributes have the same degree of influence [21] to a further processing. In our research we normalized all features selected according to equation 4.

$$Norm(x_i) = \frac{x_i - min(X)}{max(X) - min(X)} \quad (4)$$

5.2 Performance Evaluation Criteria

There are many evaluation criteria for software effort estimation introduced in the literature, among them we applied the most frequent evaluation criteria such as: Magnitude of Relative Error (MRE), Mean Magnitude of Relative Error (MMRE), prediction at level e Pred(e) and Median of MRE values (MdMRE) [21]. The summary of evaluation criteria is outlined in Table 2.

Table 2. Evaluation criteria

Evaluation criterion	***Definition***
MRE	Computes the degree of estimating error in an individual estimate and should be less than 25% to be acceptable. $MRE_i = \frac{\mid actual_i - estimated_i \mid}{actual_i}$
MMRE	Mean of MRE values. It should be less than 25% to be acceptable. $MMRE = \frac{1}{n}\sum_{i=1}^{n} MRE_i$
MdMRE	Median of MRE values. It should be less than 25% to be acceptable.
Pred(0.25)	Proportion of prediction within 25% of actual value for all predictions. It should be larger than 75% to be acceptable. $Pred(e) = \frac{\#\ projects\ with\ MRE \le e}{\#\ projects}$

5.3. Research Methodology

In order to evaluate the performance of our FFSS on analogy software effort estimation model, we compared it against various FSS algorithms used in ASEE such as: Hill climbing, Exhaustive search, Forward subset selection and backward subset selection. Since the evaluation criterion for these algorithms is MMRE with Jack knifing hold-out strategy, we intend to use n-fold cross validation to compare between analogy models in order to make the estimation unbiased. We should note here that the following shared considerations have been taken into account when we developed ASEE models:

1. Similarity measurement that was used to assess similarity degree between software projects was Euclidean distance.
2. Adaptation strategy used was mean of K nearest neighbors where K is less or equal to 3.

The methodology used in empirical evaluation is described as follows:

1. For each dataset we find the best feature subset using FFSS, Hill climbing, Exhaustive search, Forward subset selection and backward subset selection algorithms.
2. For each best feature subset ASEE model is developed based on above considerations.
3. For each model, using n-fold cross-validation we evaluate the impact of feature subset on estimation accuracy using (MMRE, MdMRE, Pred(25%)) evaluation criteria.

6. RESULTS

As in any software effort estimation models, the effective of such model depends on the characteristics of dataset used. In other words, Analogy model appears well suited when relation between effort and independent features are discontinues and nonlinear [20]. The main objective of the paper is to select optimal feature subset based on Fuzzy logic with taking into account information about structure of data. Our FFSS algorithm is exhaustive search in nature because it evaluates all possible feature subsets to find one with minimum similarity between its clusters. Exhaustive search that is implemented in ArchAngel Tool employs MMRE with Jack knifing hold-out strategy to optimize accuracy of the feature subset, which means to build analogy model for each possible feature subset and evaluate MMRE perdition accuracy of it, so there will be 2^n analogy models before building the robust one. However, Kirsopp and shepperd [13] indicated that exhaustive search produces the optimal feature subset among other searching techniques because it enumerates all possible solutions, but it needs intensive computations. Therefore they recommended it when size of dataset is quite small. On the other hand, when dataset is too large for exhaustive search and prediction accuracy is important, Hill climbing and Random search is the recommended solution. Otherwise Forward and backward subset selection are preferable solution. The results obtained from both datasets corroborate these facts.

Kirsopp and shepperd [13] stated that it is difficult to compare between FSS algorithms that deal with classification with algorithms implemented ArchAngel Tool which use prediction accuracy as evaluation criteria. In fact, the same prediction accuracy measures are used to evaluate the model which extremely leads to biased estimation. However, since FSS algorithms employed Jack knifing strategy we use n-fold cross-validation to evaluate all analogy models.

The best feature subsets obtained by these algorithms and our FFSS algorithm are given in tables 3 and 7. From ISBSG dataset we can see that *Development type, Resource level* and *Average team size* attributes repeated by most of the algorithms which in general forms the optimal solution obtained by exhaustive and random search. Furthermore, it is not surprisingly that (random, exhaustive and hill climbing) produce same optimal feature subset which is considered the most predictive features. The steepest ascent (SA) hill climbing [13] starts from randomly selected point and evaluates each possible move towards best available neighborhood. This process is repeated until no better solution than current. On the other hand, the random mutation (RM) hill climbing does not move to neighborhood until better solution is found which means it does not need to explore the entire neighborhood before making a move. In our research we employed RM hill climbing.

Tables 4, 5 and 6 present comparison between various FSS algorithms in ASEE in terms of MMRE, MdMRE and Pred(25%) respectively. The results obtained from table 5 shows that using all features leads to worse estimation accuracy. This corroborates that using all features is not suitable choice for AESS and especially when number of features is high. Using other algorithms show better estimation accuracy than using all features.

In terms of accuracy, Our FFSS has a significant impact on ASEE for both datasets with comparable MMRE and Pred results. But it is still a bit less significant than Exhaustive and hill climbing search.

It is obvious that less number of features is not usually an indicator that a feature subset performs better in ASEE than using all features or feature subsets with higher number of features. Despite the efficiency of computation of forward and backward subset selection, our FFSS performs better in ASEE for ISBSG dataset with slight difference in MMRE value. But these two algorithms still perform better than using other all features.

The same procedure is followed for Desharnais dataset. The best feature subsets are given in Table 7. It is apparent that there are some common features between all algorithms such as (*development environment and Adjusted FP)* which forms the subset obtained by exhaustive and random search algorithm. The combination of these features performs better than using all features. It is interesting to note that Forward selection and hill climbing produced the most predictive feature subset. However, Both Hill climbing and forward subset selections are steepest ascent because the entire near features is evaluated and move with best result to next position [14]. The main disadvantage of these algorithms that they don't examine all possible feature subsets and therefore cannot guarantee they produce best feature subset [4, 13].

The results for Desharnais dataset shown in Tables 8, 9 and 10 revealed that our FFSS algorithm did not produce better accuracy than Exhaustive and forward selection search irrespective of analogy adaptation strategy used. But it is still significant for ASEE. The overall evaluations show there is slight difference between other FSS algorithm in analogy estimation.

Using adaptation strategies such as (mean of two analogies, mean of three analogies) does not always contribute to better estimation. For the sake of our research we did not use other adaptation strategy such as: inverse ranked, distance based adaptation and rule based.

Lastly, we can observe that our FFSS algorithm has significant impact on ASEE with slight difference than exhaustive and Hill climbing search. Computation time of our FFSS is less than exhaustive and hill climbing, but still larger than forward and backward selection.

Table 3: Best feature subsets for ISBSG obtained by various FSS algorithms

Algorithm	Best feature subset
All features	*All*
Exhaustive search, Hill climbing, Random search	*Development type, Resource level, Average team size*
Forward subset selection	*Development type, Resource level, Maximum team size*
Backward subset selection	*Adjusted function points, Project elapsed time, Development time, Average team size, Enquiry count, Output count, Interface count, added count.*
FFSS	*Maximum Team size, Development type, Resource level, Average team size*

Table 4: MMRE comparison for ISBSG dataset

Algorithm used in analogy model	**One analogy**	**mean of 2 analogies**	**mean of 3 analogies**
All features	37.74%	41.0%	29.4%
Exhaustive search, Hill climbing, Random search	28.25%	30.3%	30.2%
Forward subset selection	33.3%	30.4%	31.2%
Backward subset selection	34.7%	38%	34.4%
FFSS	28.7%	30.6%	32.2%

Table 5: MdMRE comparison for ISBSG dataset

Algorithm used in analogy model	**One analogy**	**mean of 2 analogies**	**mean of 3 analogies**
All features	31.6%	33%	20.62%
Exhaustive search. Hill climbing, Random search	21.9%	24%	20.8%
Forward subset selection	21.0%	21.4%	20.7%
Backward subset selection	22.6%	28.7%	25.2%
FFSS	21.8%	22.3%	22.7%

Table 6: Pred(25%) comparison for ISBSG dataset

Algorithm used in analogy model	**One analogy**	**mean of 2 analogies**	**mean of 3 analogies**
All features	38.7%	38.7%	54.7%
Exhaustive search, Hill climbing, Random search	55.4%	50.7%	55.3%
Forward subset selection	58.7%	54.7%	55.3%
Backward subset selection	54.7%	45.3%	54.0%
FFSS	54.7%	52.0%	49.3%

Table 7: Best feature subsets for Desharnais obtained by various FSS algorithms

Algorithm	**Best feature subset**
All features	*All*
Exhaustive search, Forward selection, Hill climbing, Random search	*Adjusted FP, Development environment*
Backward subset selection	*Duration, Team experience, Non adjusted FP, Transaction, development environment.*
FFSS	*Team experience, Manager experience, Transaction, Envergure, PointsNonAjust.*

Table 8: MMRE comparison for Desharnais dataset

Algorithm used in analogy model	One analogy	mean of 2 analogies	mean of 3 analogies
All features	60.1%	51.5%	50.0%
Exhaustive search, Forward subset selection, Hill climbing, Random search	38.2%	39.4%	36.4%
Backward subset selection	42.4%	43.9%	46.6%
FFSS	40.2%	40.3%	38.5%

Table 9: MdMRE comparison for Desharnais dataset

Algorithm used in analogy model	One analogy	mean of 2 analogies	mean of 3 analogies
All features	41.7%	41.0%	36.1%
Exhaustive search, Forward subset selection Hill climbing, Random search	30.8%	38.0%	30.9%
Backward subset selection	38.4%	37.4%	34.6%
FFSS	32.4%	33.3%	31.7%

Table 10: Pred(25%) comparison for Desharnais dataset

Algorithm used in analogy model	One analogy	mean of 2 analogies	mean of 3 analogies
All features	31.9%	31.2%	37.7%
Exhaustive search, Forward subset selection Hill climbing, Random search	42.9%	31.2%	44.2%
Backward subset selection	36.4%	38.9%	35.0%
FFSS	39.8%	39.8%	42.4%

7. THREATS TO THE STUDY VALIDITY

The major threat to validity of our study is the population model. It is very hard to select representative data, for example, ISBSG dataset is described by more than 30 attributes which contains many missing and outlier values. We performed pre-processing stage to identify the most representative data by ignoring projects and attributes with missing values. It is argued that removing those projects could loss some valuable information. Pre-processing stage resulted in 400 projects for ISBSG dataset and 77 projects for Desharnais dataset.

We used MMRE as evaluation criteria for all FSS algorithms implemented in ArchAngel Tool and then it was used to assess prediction accuracy. This is not reasonable because MMRE has been criticised that is unbalanced in jack knifing validation and leads to overestimation more than underestimation [25, 26]. Therefore, to avoid this problem we used n-fold cross validation.

We also compared our FFSS algorithm with other FSS implemented in ArchAngel Tool. In software engineering, a new proposed model should be compared with baseline model. The common FSS algorithm used in ASEE is exhaustive search.

We used two datasets in empirical evaluation. Both datasets exhibits typical characteristics of software effort estimation dataset. Thus, we believe that both datasets are sufficient to validate our models.

8. CONCLUSIONS

This paper focuses on examining the impact of Feature Subset Selection algorithms on analogy software effort estimation. We proposed Fuzzy Feature Subset Selection for analogy software effort estimation and then compared with other popular algorithms used in this kind of estimation. From the empirical evaluation we concluded that:

1. MMRE is not often the best evaluation criteria for FSS algorithm used in ASEE. Using classification and structure of data could also help in optimizing the accuracy of ASEE. Characteristics of dataset play crucial role in Analogy software effort estimation. Thus, information about structure of data should be also taken into account.
2. The fuzzy feature subset selection (FFSS) has a significant impact on accuracy of ASEE.
3. Comparison between two datasets shows that there is no proof that a particular algorithm would perform constantly better on all datasets than others.
4. Our FFSS algorithm produces significant results comparing with exhaustive search and forward selection. This shows that data classification can also contribute to best feature subset selection.
5. Exhaustive search is recommended when dataset size is small and prediction accuracy is important. While Hill climbing is recommended when dataset size is too large and prediction accuracy is also required. FFSS, Forward selection and backward selection are recommended when computation time is important.

Further work is necessary to increase confidence that using fuzzy logic can contribute to optimal feature selection in ASEE. Thus, it can lead to better improvement in accuracy prediction.

9. ACKNOWLEDGMENTS

We would like to thank ISBSG as well as Promise Repository for granting us permission to utilize their datasets in this research.

10. REFERENCES

[1] Aha, D.W., Bankert, R.L. 1994. Feature selection for case-based classification of cloud types: an empirical comparison. In: Working Notes of the AAAI-94 Workshop on Case-Based Reasoning, 106–112.

[2] Auer S. B. M. 2004. Increasing the Accuracy and Reliability of Analogy-Based Effort Estimation with Extensive Project Feature Dimension Weighting, Proceedings of the International Symposium on Empirical Software Engineering (ISESE'04), 147-155.

[3] Boetticher G., Menzies T. Ostrand T. 2007. PROMISE Repository of empirical software engineering data http://promisedata.org/ repository, West Virginia University, Department of Computer Science.

[4] Chen, Z., Menzies, T., Port, D.,Boehm, B. 2005, Feature Subset Selection Can Improve Software Effort Estimation Accuracy, Workshop Predictor Models in Software Eng. PROMISE '05,ACM, St. Louis, Missouri USA, 1-6 .

[5] Chiu N.-H., Huang S.-J. 2007. The adjusted analogy-based software effort estimation based on similarity distances. Journal of Systems and Software 80, 628-640.

[6] Guyon I. , Elisseeff A. 2003. An introduction to variable and feature selection, Journal of Machine Learning Research, 3, 1157 – 1182.

[7] Huan, L., Lei,L. 2004. Toward Integrating Feature Selection Algorithms for Classification and Clustering, IEEE Transactions on Knowledge and Data Engineering, 17, 491-502.

[8] Huang X., Ho D., Ren J., Capretz L. F. 2005, Improving the COCOMO model using a neuro-fuzzy approach, Journal of Applied Soft Computing, 7, 29-40.

[9] Huang X., Ho D., Ren J., Capretz L. 2005. A Soft Computing Framework for Software Effort Estimation. Soft Computing Journal, Springer, 10, 170-177. available at www.springeronline.com.

[10] Idri, A., Abran, A., Khoshgoftaar, T. 2001. Fuzzy Analogy: a New Approach for Software Effort Estimation, 11th International Workshop in Software Measurements. 93-101.

[11] ISBSG. 2007. International Software Benchmarking standards Group, Data repository release 10, Site: http://www.isbsg.org.

[12] Kadoda, G., Michelle C., Chen, L., Shepperd, M. 2000. Experience using Case Based reasoning to predict Software project effort, Proceeding of Fourth international conference on empirical Assessment and evaluation in software engineering. 1-17.

[13] Kirsopp, C., Shepperd, M. 2002. Case and Feature Subset Selection in Case-Based Software Project Effort Prediction, Proc. 22nd SGAI Int'l Conf. Knowledge-Based Systems and Applied Artificial Intelligence.

[14] Kirsopp C. , Shepperd M. J. , Hart J. 2002. Search Heuristics, Case-based Reasoning and Software Project Effort Prediction, Proceedings of the Genetic and Evolutionary Computation Conference, 1367-1374.

[15] Kohavi, R., Sommerfield, D. 1995. Feature subset selection based using the wrapper method: over fitting and dynamic search pace topology. In proceedings of first international conference on knowledge discovery and data mining. 192-197.

[16] Langley, P. 1994. Selection of relevant features in machine learning. In proceedings of AAAI Fall symposium on Relevance, 1-5.

[17] Li, J., Ruhe, G. 2005. Attribute Selection and Weighting Using Rough Sets for Effort Estimation by Analogy—Initial Results, Technical Report SEDS-TR-05102, Software Engineering Decision Support Laboratory at the University of Calgary, Canada.

[18] Li, J., Ruhe, G. 2006. A Compartive Study of attribute weighting Heuristics for Effort Estimation by Analogy. In Proceedings of ISESE, Brazil, 66-74.

[19] Martin, C. L., Pasquier, J. L., Yanez, C M., Gutierrez, A. T. 2005. Software Development Effort Estimation Using Fuzzy Logic: A Case Study. In proceeding of Sixth Mexican International Conference on Computer Science (ENC'05), 113-120.

[20] Mendes E., Mosley N. 2002. Further investigation into the use of CBR and stepwise regression to predict development effort for Web hypermedia applications, 79-90.

[21] Mendes, E., Mosley, N., Counsell, S. 2003. Do adaptation rules improve web effort estimation?, In Proceedings of the fourteenth ACM conference on Hypertext and hypermedia (Nottingham, UK). 173-183.

[22] Menzies, T. , Chen Z., Hihn, J. Lum, K. 2006. Selecting Best Practices for Effort Estimation. IEEE Transaction on Software Engineering. 32, 883-895.

[23] Musflek, P., Pedrycz, W., Succi, G., Reformat, M. 2000. Software Effort Estimation with Fuzzy Models. ACM SIGAPP Applied Computing Review 8, 24-29.

[24] Ross, T.J. 2004. Fuzzy Logic with engineering applications, John Wiley& Sons.

[25] Shepperd, M. J., Schofield, C. 1997. Estimating Software Project Effort Using Analogies, IEEE Trans. Software Eng. 23, 736-743.

[26] Tron, F., Stensrud, E., Kitchenham, B., Myrtveit, I. 2003. A Simulation Study of the Model Evaluation Criterion MMRE, IEEE Transactions on Software Engineering, 29, 985-995.

[27] Xu, Z., Khoshgoftaar, T., 2004. Identification of fuzzy models of software effort estimation. Fuzzy Sets and Systems 145, 141-163.

[28] Zadeh, L. 1997. Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. Journal of fuzzy sets and systems 90, 111-127.