Software Effort Estimation Based on Weighted Fuzzy Grey Relational Analysis

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
Delivering accurate software effort estimation has been a research challenge for a long time, where none of the existing estimation methods has proven to consistently deliver an accurate estimate. Previous studies have demonstrated that estimation by analogy (EBA) is a viable alternative to other conventional estimation methods in terms of predictive accuracy. EBA offers a way to use a formal method with data from a past project to derive a new estimate. Two important research areas in EBA are addressed in this paper: software projects similarity measurement and attribute weighting. However, the inherent uncertainty of attribute measurement makes similarity measurement between two software projects subject to considerable uncertainty. To tolerate such inherent uncertainty we propose a new similarity measurement method by combining the advantages of Fuzzy Set Theory and Grey Relational Analysis. In addition, since each attribute has different influence on the project retrieval we propose a new approach to deal with this issue based upon the idea of Kendall's coefficient of concordance between the similarity matrix of project attributes and the similarity matrix of known effort values of the dataset. Our results show improved prediction accuracy when multiple project attributes are used with determined weights.

Categories and Subject Descriptors

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

General Terms
Management, Measurement

Keywords
Software effort estimation, Similarity measurement, Attribute weighting, Fuzzy modeling.

1. INTRODUCTION
Estimating the most likely project effort with high precision is still a largely unsolved problem [13, 21]. This problem lies in the fact that software effort estimation is a complex process due to the number of factors involved, including the human factor, the complexity of sizing the software product and the variety of development environments [2]. However, for any software effort estimation model to be successful in industry it should be first trusted by software practitioners and produces sound credible predictions [1]. Software effort estimation by analogy (EBA) has achieved considerable attention from researchers for over two decades [15, 19, 20, 22, 23, 24]. EBA is a form of Case-based reasoning which aims to identify solution for a new problem based on previous solutions from the set of similar cases. Unlike black-box techniques such as neural networks, the EBA is by nature transparent in which its process can be easily understood and explained to practitioners and other users. The attractiveness of EBA in software engineering stems from being able to model the complex relationship between effort and other software attributes that are described by quantitative and qualitative scale type [13, 23, 24]. Furthermore, the developers and users may be willing to accept such kind of estimation because it mimics human problem solving.

The key accuracy to EBA is choosing the appropriate similarity measure that attempts to retrieve the most similar historical projects to the project under estimation. This issue has been investigated and evaluated in previous studies of software cost estimation. The common used techniques are based on nearest neighborhood techniques such as Euclidean distance, Weighted Euclidean distance and Maximum measures. In these algorithms the closest analogue to a project p_i is the project with minimum distance. Mendes et al. [22] compared between different types of distance metrics in EBA and revealed that using different metrics produce dissimilar results which in turn show the importance of selecting appropriate similarity measure on effort estimation.

However, despite the widespread use of nearest neighborhood based techniques in EBA, there are certain limitations that affect project retrieval [13, 24]. **First**, they are sensitive to the irrelevant attributes [6] and the degree of attribute influence on the effort estimates. **Second**, categorical attributes are problematic in which it is difficult to handle categorical variables other than binary valued variable: attributes match or fail to match with no middle ground [24]. **Third**, most of software attribute values are measured based on human knowledge which is often vague and imprecise. This uncertainty is associated with a lack of precise knowledge which is a matter of guessing rather than exact measurement. Therefore, we may find that two projects are similar with respects to attributes but their efforts are completely different. Variability of data type also increases the uncertainty in assessing the similarity degree between two software projects. A **fourth** criticism that was stated by Shepperd et al. [24]: "*They fail*

to take into account information which can be derived from the structure of the data, thus they are weak for higher order attribute relationships".

The contribution of this paper is twofold: the first and most important objective is to model and tolerate the uncertainty in similarity measurement between two software projects when they are described by numerical and categorical data type. Therefore we use two popular techniques: Fuzzy Set Theory (FST) [27] and Grey Relational Analysis (GRA) [7, 8]. We also define two levels of similarity measure: Local and Global similarity measure. In this regard, three types of local similarity measures are defined in accordance with data type scale (i.e. numerical, ordinal, and nominal scale). For numeric data type we used FST [27] to tolerate uncertainty in local similarity measures between target project p_o and historical project p_i at the j^{th} numeric feature. For the ordinal scale we used the Grow's formula [9] that assesses similarity between two ordinal values based on their ranking. After measuring all local similarity degrees the weighted GRA method is then used to assess the global similarity between two projects and tolerate uncertainty associated with using different data types in local similarity. The second contribution is to find an appropriate attribute weighting to enhance similarity measure. The procedure of attribute weighting as explained in section 4.3 depends on the measure of Kendall's coefficient of concordance [17] between the similarity matrix of project attribute values and the similarity matrix of known effort values of the dataset. The paper also evaluates our approach on two real-world project datasets, and compares its performance with the conventional analogy approach.

The rest of the paper is organized as follows: section 2 presents related work for EBA with GRA and attribute weighting. Section 3 introduces Fuzzy modeling. In section 4 the weighted FGRA is introduced. In Section 5 we present the datasets used in empirical evaluation. In section 6 we discuss the results of applying our model on employed datasets. Section 7 presents the conclusions of this paper.

2. RELATED WORK

2.1 Related Works of GRA and Fuzzy Logic in Estimation by Analogy

The accuracy of EBA has been intensively investigated in previous studies [12, 15, 19, 20, 22, 23, 24]. In this paper we are specifically interested with those studies that used GRA or Fuzzy logic in estimation by analogy. In terms of applying Fuzzy logic in EBA, Idri et al. [13] proposed an alternative approach to EBA called Fuzzy analogy model. The model basically attempted to express linguistic quantifiers (ordinal scale) as Fuzzy sets instead of using their ordered rank. They developed a new Fuzzy similarity between two software projects that are described only by linguistic quantifiers such as low, fair and high. Although the model is a promising technique for handling categorical data it may not perform well over other datasets like ISBSG [14] that are structurally dissimilar to the COCOMO dataset. In addition to that, the model needs sufficient information about how to replace the ordinal data with Fuzzy sets.

Grey Relational Analysis (GRA) is an important method of Grey System theory [7, 8] which is used to determine the relationship (similarity) between two data series [11]. It was developed by Deng [7, 8] to study uncertainty in system models and process incomplete small datasets. The attractiveness of GRA to software effort estimation stems from its flexibility to model complex nonlinear relationship between effort and other cost drivers [10, 11]. Furthermore, the GRA has the ability to learn from a small number of cases which is effective in the context of data-starvation [25]. In software engineering, little research has been carried out to exploit GRA in the estimation process. Huang et al. [11] used a genetic algorithm with GRA in order to adjust the weight factor associated with the weighted GRA. Experiments on various well established datasets revealed that the weighted GRA with genetic algorithms has a significant impact on the accuracy of software effort estimation. Song et al. [25] proposed a software prediction model based on GRA called GRACE. They used GRA to select an optimal feature set based on the similarity degree between effort attribute and other attributes. The attributes that exhibit large similarity are selected to form the optimal feature set. The attributes in this model are preferably continuous rather than categorical which is the main shortcoming of this model. The GRA is later used to derive new estimate by finding the closest case that approximately agrees with current case on all effort drivers. Their model has outperformed other prediction models such as neural networks, decision tree and stepwise regression, etc. Hsu et al. [10] proposed various weighted GRA models for software effort estimation. The investigated models are distance-based weight, linear weight, non-linear weight, maximal weight and correlative weight. They reported that weighted GRA performs better than non-weighted GRA in software effort estimation. The linearly weighted GRA outperforms other weighted GRA.

2.2 Related works of attributes weighting

Various attribute weighting approaches have been investigated in software estimation:

- Human Judgment: the attribute weights are given by experts based on past experiences.
- Identical weights: all attributes have equal weights.
- Statistical approaches such as Mantel's correlation, inverse variance or range values. Keung et al. [16] proposed a new technique to select and weighting attributes based on Mantel's correlation between distance matrix of effort and distance matrix of each attribute [16]. The method attempts to find the appropriateness of dataset to EBA and then find appropriate weights for each selected attributes based on Mantel's randomness test.
- Set each project attribute weight to either 0 or 1 based on optimizing estimation quality metric such as *MMRE* or *Pred(25%)*, using some searching techniques [6]. Once these attributes are selected, they are all given the same weight. A typical example is that a study undertaken by Kirsopp et al. [18] who compared between various Wrappers selection algorithms in EBA using Angel tool. The results showed that using Wrapper algorithms lead to better prediction accuracy than using all attributes. This demonstrated that exhaustive search produced better accuracy results than other searching techniques but it is still computationally far intensive when dataset size is too large. Others such as Hill climbing and Random search are also significant when dataset size is too large and prediction accuracy is important.
- Extensive attribute weighting technique: Auer et al. [1] addressed the issue of replacing the attribute selection by

extensive attribute weighting technique based on brute-force algorithm. They claimed searching for an optimal attribute subset fails to account for the influence of each attribute on project similarity and for the volatility of the resulting attribute weights over the lifetime of a growing project database, which makes EBA less acceptable. The basic principle of their technique is much alike the brute-force selection algorithm where each attribute's weight is calculated by scaling weight dimension from 0 to 1 and optimized based on the estimation quality metric such as *MMRE*. The results obtained showed that using extensive number of attributes with weight dimension produced better accuracy than using optimal attribute subset and improving volatility leading to greater acceptance by practitioners. Nevertheless, this approach is computationally intensive in spite of they claimed that model calibration only takes place when the historical dataset is updated and once completed, estimates are obtained in real-time.

Above all, most of the existing attribute weighting techniques are limited to estimation quality metrics and computationally far intensive especially heuristic algorithms. In this paper we used the underlying assumption of EBA: The projects that are similar in respect of their attribute values are also similar in terms of their effort values, as a baseline to reflect the influence of each attribute on similarity measurement. We used Kendall's coefficient of concordance between the similarity matrix of project attribute values and the similarity matrix of known effort values of the dataset as explained in section 4.3.

3. FUZZY MODELING

Fuzzy logic and sets as introduced by Zadeh [27] provide a representation scheme and mathematical operations for dealing with uncertain, imprecise and vague concepts. Fuzzy logic is a combination of a set of logical expressions with Fuzzy sets. Zadeh [27] defined the meaning of the membership for Fuzzy sets to be a continuous number between zero and one. Each Fuzzy set is described by membership function such as Triangle, Trapezoidal, Gaussian, etc., which assigns a membership value between 0 and 1 for each real point on universe of discourse.

Fuzzy model can be constructed by one of two ways either by expert knowledge or using algorithms. The former, uses the experience that is formed in if-then-rules expressions where parameters and memberships are tuned using input and output data. The latter uses algorithms such as Fuzzy C-means (FCM) [4] to create membership functions. However, the Fuzzy model in this paper was constructed based on the second approach where membership functions obtained by using *genfis3* function that is already implemented in MATLAB®.

4. WEIGHTED FGRA MODEL

4.1 Problem Definition

The basic assumption of EBA is that given a historical dataset H, and given a new project to be estimated along with its attributes, EBA attempts to retrieve the projects with similar attributes in *H*. Those project's efforts are then used to derive an effort estimate for the new project. Each historical dataset H is basically defined as $H =< P, A, V, T >$ where P is the set of completed projects $P = \{p_1, p_2, .., p_i, .., p_n\}$ (all projects in P are described by common attributes $A = \{a_1, a_2, .., a_j, .., a_m, effort\}$, where a_j is a specific attribute used to define software projects and *effort* is a specific attribute used to define the actual effort needed to accomplish a software project). V is the domain of attribute values of all projects in P, i.e. $V = \bigcup_{\forall i} V_i$ where:

$$V_i = \{a_1(p_i), a_2(p_i), ..a_j(p_i), .., a_m(p_i), effort(p_i) \mid p_i \in P \; a_j, effort \in A\}$$

and $effort(p_i) \in R^+$ where R^+ is positive real number space. T is a set of the attributes types $T = \{t(a_j) \mid a_j \in A, j = 1,2..., m\}$, $t(a_j)$ represents the corresponding data type of attribute a_i which in this paper can hold one of these possible values {*'Numeric', 'Ordinal', 'Nominal'*}. For a special case $t(effort)$ should always be numeric while others depend on the scale of measurement.

Let u be the project to be predicted. The project u should first meet the following conditions: (1) *it shares the same set of attributes A except effort attribute which is unknown and will be predicted later*. (2) *The historical dataset H should remain unchanged during prediction process*. (3) *No missing values are allowed for any attribute values* $a_j(u) \neq \phi$, $a_j \in A$. The statement problem of EBA is expressed as follows: For a project u whose $effort(u)$ is unknown and to be estimated, only $V_u = \{a_1(u), a_2(u), .., a_j(u), ..a_m(u) \mid a_j \in A\}$ is given. Therefore the $effort(u)$ is estimated from the set $S(u) = \{p_i \in P \mid Sim(u, p_i) \geq \beta\}$ of N projects that are similar to project u_k, i.e. those with a maximal similarity degree. Where β is the threshold of similarity degree. The final estimate of project u is an adaptation of all top N retrieved efforts in $E(u) = \{effort(p_i) \in V \mid p_i \in S(u)\}$, i.e. $effort(u) = adp(E(u))$.

4.2 Project Retrieval

In order to find the most similar projects in the set P to the project under estimation u, the project u are assessed against all projects in P over a set of attributes in terms of local similarity and global similarity measures. Local similarity measure ($\Delta(a_j(p_i), a_j(u)) : a_j \times a_j \rightarrow [0,1], a_j \in A$) is defined to assess the similarity degree between project u and each historical project $p_i \in P$ in respect of a related attribute a_j. Global similarity degree is used to aggregate all local similarity degrees and thus is defined as function of these local similarity measures. The rationale behind this mechanism is to reduce the uncertainty in the project similarity measure that is caused by human imprecision and variability of data types, which results in finding software projects that are similar with respects of their attributes but they their efforts' values are extremely different.

4.2.1 Local Similarity Measures

Based on the categorization of data type we define a local similarity measure ($0 \leq \Delta(a_j(p_i), a_j(u)) \leq 1$) for each data type as shown in Eq. (1) where their details are explained in the following subsections. One of the primary purposes of classifying variables according to their level or scale of measurement is to

facilitate the choice of appropriate similarity measure used to assess the degree of matching between two software projects.

$$\Delta(a_j(p_i),a_j(u)) = \begin{cases} \Delta_{NOM} & if\ t(a_j) = 'NOMINAL' \\ \Delta_{ORD} & if\ t(a_j) = 'ORDINAL' \\ \Delta_{NUM} & if\ t(a_j) = 'NUMERICAL' \end{cases} \quad (1)$$

4.2.1.1 *Numerical Scale*

Since software developers often have measurements that are inaccurate, inexact, or of low confidence [27], we present the Fuzzy similarity between two values at the j^{th} continuous attribute to measure local similarity degree.

The use of Fuzzy modeling to assess the degree of similarity between two values in continuous attribute a_j requires determination of number of Fuzzy sets, and their membership functions. In this paper we used MATLAB function *genefis3* to construct such Fuzzy model with triangular membership function. This function uses FCM algorithm to find membership values for each observation in all clusters and then generates Fuzzy membership function for each Fuzzy cluster as shown in Figure 1.

For continuous data type, each attribute a_j is fuzzified and replaced by corresponding Fuzzy sets as shown in Figure 1. After constructing Fuzzy sets for each continuous attribute a_j, each value $a_j(p_i)$ is replaced by its corresponding membership values $\overline{a_j(p_i)}$ as depicted in Eq. (2).

$$\overline{a_j(p_i)} = \{\mu_{c_j^1}(p_i),\mu_{c_j^2}(p_i),...,\mu_{c_j^d}(p_i)\}, \quad i = 1,2,...n,\ j = 1,2,..m. \quad (2)$$

The local similarity between two continuous values at attribute a_j is defined in Eq. (3).

$$\Delta_{NUM}(a_j(p_i),a_j(u)) = \frac{\sum_{d=1}^{C} \min(\mu_{c_j^d}(p_i),\mu_{c_j^d}(u))}{\sum_{d=1}^{C} \max(\mu_{c_j^d}(p_i),\mu_{c_j^d}(u))} \quad (3)$$

where $\mu_{c_j^d}(p_i)$ is the membership value of i^{th} project in d^{th} Fuzzy set. If $\Delta_{NUM}(a_j(p_i),a_j(u))$ is one then the two projects are identical and if $\Delta_{NUM}(a_j(p_i),a_j(u))$ is zero then the two projects are totally dissimilar.

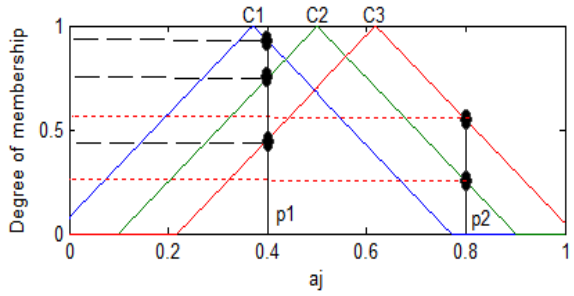

Figure 1 Fuzzy sets for a_j

For example, suppose $a_j(p_1) = 0.4$ and $a_j(p_2) = 0.8$ as shown in Figure 1, the corresponding membership values set for each numeric value is: $\overline{a_j(p_1)}$ ={0.94, 0.78, 0.45} and $\overline{a_j(p_2)}$ ={0.0, 0.28, 0.55}, the similarity degree is calculated as follows:

$$\Delta_{NUM}(a_j(p_1),a_j(p_2)) = \frac{\min(0.94,0) + \min(0.78,0.28) + \min(0.45,0.55)}{\max(0.94,0) + \max(0.78,0.28) + \max(0.45,0.55)}$$

$$\Delta_{NUM}(a_j(p_1),a_j(p_2)) = \frac{0 + 0.28 + 0.45}{0.94 + 0.78 + 0.55} = 0.322$$

4.2.1.2 *Nominal scale*

In many cases we cannot measure variable in quantitative way, but it is possible to measure it in term of category. A nominal value [13] is used when number is only symbols to represent different classes of entities in which no natural order among the categories. Nominal scale type categories software entities into different possible classifications, for example, '*developmentType*' attribute can be classified into three main categories: *new development, enhancement and re-development.* For this data type there is no way to calculate $\Delta_{NOM}(a_j(p_i),a_j(u))$ other than binary valued. In other words, the similarity degree here is a kind of comparison either 1 when they are similar and 0 when they are different because we are only interest to know whether are the same or not. The nominal value of $a_j(u)$ is not necessary to be with the domain values of a_j at the time of prediction.

$$\Delta_{NOM}(a_j(p_i),a_j(u)) = \begin{cases} 1 & a_j(p_i) = a_j(u) \\ 0 & a_j(p_i) \neq a_j(u) \end{cases} \quad (4)$$

4.2.1.3 *Ordinal scale*

The ordinal scale type [13] is the second type of categorical data that adds extra information about an ordering of categories to classification of entities, for example, *Team_Skill* variable can be measured as: *low, medium, high*. Hence, we can notice that the categories for an ordinal scale of data have a natural order which means that the categories in ordinal attribute are assigned by non-arbitrary numbers in an orderly manner. The similarity between two values of ordinal scale is defined based on the Grow's formula [9] as shown in Eq. (5). The Equation suggests to mapping the ordinal values to their ranking numbers and then finding similarity between their ranking positions represented by their ranking numbers. The closer two values in their ranking positions, the more similar they are.

$$\Delta_{ORD}(a_j(p_i),a_j(u)) = 1 - \frac{|a_j(p_i) - a_j(u)|}{|\max(a_j) - \min(a_j)|} \quad (5)$$

For example, suppose a_j = *"Team_skill"* is ordinal attribute consists of the following categories {1-Very Low, 2- Low, 3-Moderate, 4- High, 5- Very High}, where the ordinal scale with its symbolic values being mapped to integers in equal similarity. The local similarity between *Team_skill*(P_1)='Very Low' and *Team_skill*(u)='High' is given as follows:

$$\Delta_{ORD}(a_j(p_1), a_j(u)) = 1 - \frac{|1-4|}{|5-1|} = 0.25$$

4.2.2 Global Similarity Measure Based on GRA

Since many different data types are involved in measuring local similarity degrees, this could increase uncertainty in project retrieval. Therefore we used Grey Relational Coefficient ($\gamma(a_j(p_i), a_j(u))$ [10] to tolerate uncertainty in local similarity degrees between target project and each historical project as shown in Eq. (6). Then we used Grey Relational Grade ($\Gamma(u, p_i)$) [10] to compute the global similarity degree between target project u and all historical projects $p_i \in P$.

$$\gamma(a_j(p_i), a_j(u)) = \frac{\min\limits_{i,j} \Delta(a_j(p_i), a_j(u)) + \xi \max\limits_{i,j} \Delta(a_j(p_i), a_j(u))}{\Delta(a_j(p_i), a_j(u)) + \xi \max\limits_{i,j} \Delta(a_j(p_i), a_j(u))} \quad (6)$$

Where $\Delta(a_j(p_i), a_j(u)) = 1 - \Delta(a_j(p_i), a_j(u))$ since GRC can accept local similarity as distance measure (i.e. zero when two projects are totally similar and one when they are totally dissimilar). $\xi \in [0,1]$ is the distinguishing coefficient used to minimize the difference between Δ and $\max\limits_{i,j} \Delta$.

The global similarity measure $\Gamma(u, p_i)$ is a function of local similarity degrees $\Gamma(u, p_i) = f(\gamma(a_1(p_i), a_1(u)), ..., \gamma(a_m(p_i), a_m(u)))$, aims at finding the overall similarity degree between target project u and comparative project $p_i \in P$. The $\Gamma(u, p_i)$ takes values between 0 and 1. When the value of $\Gamma(u, p_i)$ approaches the value 1, the two projects are regarded "more closely similar". When $\Gamma(u, p_i)$ approaches a value 0, the two projects are regarded "more dissimilar".

$$\Gamma(u, p_i) = \sum_{j=1}^{M} w_j * \gamma(a_j(p_i), a_j(u)) \quad (7)$$

$$\text{where } \sum_{j=1}^{M} w_j = 1 \quad (8)$$

The attribute weight $w_j \in W$ is calculated based on Kendall's coefficient of concordance as explained in the next section. To ensure that our similarity measures conform to the general concept of similarity measure, the global similarity measure respects the following properties:

1. $0 \le \Delta(a_j(p_i), a_j(u)) \le 1$
2. $0 \le \Gamma(u, p_i) \le 1$
3. $\Gamma(u, p_i) = 1$, if and only if $\forall a_j \in A : \Delta(a_j(p_i), a_j(u)) = 1$.
4. $\Gamma(u, p_i) = 0$, if and only if $\forall a_j \in A : \Delta(a_j(p_i), a_j(u)) = 0$.
5. $\Gamma(u, p_i) = \Gamma(p_i, u)$.
6. $\Gamma(u, p_i) > \Gamma(u, p_q)$, $i \ne q$, if and only if $\exists a_j \in A : \Delta(a_j(p_i), a_j(u)) > \Delta(a_j(p_q), a_j(u))$.

After finding similarity between the reference project and each comparative project, it is necessary to retrieve the project that exhibits the largest similarity with the reference project. Therefore the historical projects are ranked in accordance to their $\Gamma(u, p_i)$. This procedure called Grey Relation Rank (GRR) [25] which attempts to rank all historical projects according to their global similarity degree with target project. If $\Gamma(u, p_x) > \Gamma(u, p_y) > ... > \Gamma(u, p_z)$ then $p_x > p_y > ... p_z$ is the similarity order for the project u_k .

4.3 Attribute Weighting

Accurate determination of attribute weights is difficult to obtain in practice because assessed weights are always subject to response error in addition to the required model. The main goal of this step is to find appropriate attribute weights that reflects the significant relationship between each attribute with the effort. In this paper we propose a new method based upon the use of Kendall's coefficient of concordance (also known as *Kendall's W*) [17] between rankings of two corresponding pair of rows of similarity matrix based project attribute values (δ_{a_j}) and similarity matrix based project effort values (δ_{effort}), excluding diagonal elements. After calculating *Kendall's W,* we use simple average to compute overall agreement between two matrices.

Kendall's *W* is a non-parametric statistic [17] that can be used for assessing agreement among different rankings for the same set of objects. Kendall's *W* ranges from 0 (no agreement) to 1 (complete agreement). Therefore it can be considered as weight. The *Kendall's W* between two corresponding pair of rows is calculated based upon the following equation:

$$WK_i = \frac{12\sum \overline{R_i}^2 - 3n(n+1)^2}{n(n^2-1)} \quad (9)$$

Where *n* is the number of observations, $\overline{R_i}$ is the average rank for the i^{th} row.

In the presence of ties the *Kendall's W* is computed as follows:

$$WK_i = \frac{12\sum \overline{R^2} - 3n(n+1)^2}{n(n^2-1) - \left(\sum T_j / k\right)} \quad (10)$$

$$T_j = \sum_{i=1}^{g_j} t_i^3 - t_i, \; j = 1, \cdots k \quad (11)$$

Where *k* is the number of column, g_j is the number of groups of ties in column *j* and t_i is the number of tied elements in g_j .

If the *Kendall's W* is 1, then all the ranking have been totally agreed. If *Kendall's W* is 0, then there is no overall trend of agreement among the different rankings, and their rankings may be regarded as essentially random. Intermediate values

of *W* indicate a greater or lesser degree of agreement among the various rankings.

The proposed method assumes that the projects that are similar in terms of attribute values $a_j \in A$ should be also similar with effort values and thus their similarity ranking should be also agreed. Our assumption is that the attribute that presents perfect agreement with effort will be given a larger weight.

The procedure of attribute weighting is given as follows:

1. Construct similarity matrix for the effort, and for attribute $a_j \in A$.

$$\delta_{a_j} = \begin{bmatrix} 1 & \Delta_{aj}12 & \Delta_{aj}13, \ldots & \Delta_{aj}1n \\ \Delta_{aj}21 & 1 & \Delta_{aj}23, \ldots & \Delta_{aj}2n \\ \ldots & \ldots & 1 & \ldots \\ \Delta_{aj}n1 & \Delta_{aj}n2 & \Delta_{aj}n3, \ldots & 1 \end{bmatrix}$$

$$\delta_{effort} = \begin{bmatrix} 1 & \Delta_{e}12 & \Delta_{e}13, \ldots & \Delta_{e}1n \\ \Delta_{e}21 & 1 & \Delta_{e}23, \ldots & \Delta_{e}2n \\ \ldots & \ldots & 1 & \ldots \\ \Delta_{e}n1 & \Delta_{e}n2 & \Delta_{e}n3, \ldots & 1 \end{bmatrix}$$

where $\Delta_{aj}12 = \Delta(a_j(p_1), a_j(p_2))$ and $\Delta_e 12 = \Delta(effort(p_1), effort(p_2))$ and so forth

2. Calculate *Kendall's W* (WK_i) for each corresponding pair of rows, and then calculate overall $WK(a_j)$.

$$WK(a_j) = \frac{1}{n}\sum_{i=1}^{n} WK_i \qquad (12)$$

3. Repeat steps 1 and 2 for all attributes $a_j \in A$

4. Compute normalized weight $w_j \in W$ of the a_j as shown in Eq. (13):

$$w_j = \frac{WK(a_j)}{\sum_{j=1}^{m} WK(a_j)} \qquad (13)$$

For example, Suppose

$$\delta_{a_1} = \begin{bmatrix} 1 & 0.8 & 0.53 & 0.67 & 0.47 \\ 0.8 & 1 & 0.33 & 0.47 & 0.67 \\ 0.53 & 0.33 & 1 & 0.87 & 0 \\ 0.67 & 0.47 & 0.87 & 1 & 0.13 \\ 0.47 & 0.67 & 0 & 0.13 & 1 \end{bmatrix}$$

and,

$$\delta_{effort} = \begin{bmatrix} 1 & 0.93 & 0.31 & 0.6 & 0.69 \\ 0.93 & 1 & 0.24 & 0.53 & 0.76 \\ 0.31 & 0.24 & 1 & 0.71 & 0 \\ 0.6 & 0.53 & 0.71 & 1 & 0.29 \\ 0.69 & 0.76 & 0 & 0.29 & 1 \end{bmatrix}$$

$$WK_i = \begin{bmatrix} 0.7 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Then the $WK(a_j)$=0.94 which shows perfect agreement between this attribute and the effort.

5. DATASETS

The paper evaluates our approach on two real-world project datasets, and compares its performance with the conventional analogy approach. The first dataset is Desharnais dataset which originally consists of 81 software projects collected from Canadian software houses [5]. This dataset is described by 11 attributes, one dependent attribute which is the effort measured in '1000 person-hours', and 10 independent attributes: '*TeamExp*', '*ManagerExp*', '*YearEnd*', '*Length*', '*Transactions*', '*Entities*', '*PointsAdjust*', '*DevEnv*', '*PointsNonAjust*', and '*Language*'. Due to the existance of missing values in 4 projects, these projects are excluded from the dataset because they are misleading the estimation process. The second dataset is COCOMO dataset [5] which originally includes 63 software projects are described by 16 cost derivers (effort multipliers). One numeric attribute measured by Kilo Delivered Source Instructions (KDSI). 15 out of 16 are measured on a scale composed of six categories: (*very low, low, nominal high, very high, and extra high)* where each category is associated with numeric value. Despite the fact that both datasets are now over 20 years old, they are still commonly used to assess the comparative accuracy of new techniques.

6. EXPERMENTAL METHODOLOGY AND EVALUATION

6.1 Evaluation Criteria

To evaluate the accuracy of the proposed estimation method, we have used common evaluation criteria in the field of software cost estimation. Magnitude Relative Error (*MRE*) computes the absolute percentage of error between actual and predicted effort for each reference project.

$$MRE_{p_i} = \frac{|effort(p_i) - \overline{effort(p_i)}|}{effort(p_i)} \qquad (11)$$

Where $effort(p_i)$ and $\overline{effort(p_i)}$ are the actual value and predicted values of project p_i.

Mean Magnitude Relative Error (*MMRE*) calculates the average of *MRE* over all reference projects. Despite of the wide usage of *MMRE* in estimation accuracy, there has been a substantive discussion about efficacy of *MMRE* in estimation process. *MMRE* has been criticized that is unbalanced in many validation circumstances and leads often to overestimation [24]. Moreover, *MMRE* is not always reliable to compare between prediction methods because it is related to the measure of *MRE* spread. Therefore we used one-sample Wilcoxon signed rank test and Wilcoxon rank-sum test to investigate the statistical significance of all the results, setting the confidence limit at 0.05. The Wilcoxon signed rank test is a nonparametric test that compares

the median of a sample of numbers against a hypothetical median. The reason behind using these tests is because all absolute residuals for all methods used in this study were not normally distributed. Since the *MMRE* is sensitive to an individual outlying prediction, when we have a large number of observations, we adopt median of *MREs* for the *n* projects (*MdMRE*) which is less sensitive to the extreme values of *MRE*. We also used Mean Magnitude Relative Error to estimated value (*MMER*).

$$MMRE = |P|^{-1} \sum_{p_i \in P} | MRE_{p_i} | \qquad (12)$$

$$MdMRE = \underset{p_i \in P}{median}(MRE_{p_i}) \qquad (13)$$

$$MMER = |P|^{-1} \sum_{p_i \in P} \frac{| effort(p_i) - \overline{effort(p_i)} |}{\overline{effort(p_i)}} \qquad (14)$$

Pred (ℓ) is used as a complementary criterion to count the percentage of estimates that fall within less than ℓ of the actual values. The common used value for ℓ is 25%.

$$pred(\ell) = \frac{\lambda}{N} \times 100 \qquad (15)$$

Where λ is the number of projects that have $MRE_i \le \ell\%$, and N is the number of all observations. A software estimation method with lower *MMRE*, *MdMRE*, and higher *Pred(25)* shows its derived estimates are more accurate than other methods. We also used Boxplot of absolute residuals as alternatives to simple summary measures because they can give a good indication of the distribution of residuals and can help explain summary statistics such as *MMRE* and *Pred(25)*.

6.2 Discussion of the Results

To investigate the performance of our proposed method we used jack-knifing procedure (also called leave one-out cross validation) [22] which validates the error of the prediction method. Jack-Knifing procedure involves dividing the dataset into multiple training and validation sets and aggregating the accuracy across all validation sets. In each iteration one observation is held out once as test data and the method is trained on the remaining observations. The training set is used to find appropriate weights for project retrieval then MRE of test observation is evaluated. Thus, the evaluation procedure is executed *n* times according to the number of observations.

The use of FCM to construct Fuzzy model for numeric attributes requires determination of the correct number of clusters (*C*). yet there is no prior way to determining the actual number of clusters unless we use some clustering validating index like Xie and Beni formula [26]. This procedure is similar to trial and error procedure; therefore in this section we demonstrate the impact of number of Fuzzy clusters on the prediction accuracy by varying *C* from 2 to 10 with increment by 1. The number of clusters varies for each dataset in every jack-knife iteration according to the number of projects, number of numeric attributes and distribution of data. For the distinguishing coefficient ξ we use the default value that was suggested by Deng [7, 8] which is 0.5.

Tables 1 and 2 show the accuracy of respective methods using *MMRE*, *MdMRE*, *MMER* and *Pred(25)* for Desharnais and COCOMO respectively. The results obtained in Table 1 shows that the weighted FGRA produced credible estimates in general and especially when number of clusters increases. Not surprisingly, the result are in general good if we consider the lower *MMRE*, *MdMRE* and higher *Pred(25)*. Similarly, the results obtained in Table 2 show the prediction accuracy improve as the number of Fuzzy clusters increases. This corroborates our assumption that presumes the more compact Fuzzy clusters (i.e. coherence clusters) is the more efficient to deliver a good prediction. By analysing the obtained number of clusters we generally observe that the datasets require sufficient number of clusters and preferably more than 8 clusters to produce comparable accuracy. Yet even in the highly not clustered circumstances the weighted-FGRA method was able to produce a predictive accuracy with *MMRE*=32.8% for Desharnais and *MMRE*=28.5% for COCOMO. This is indicative of the possibility of being able to handle uncertainty in similarity measurement and thus to take into account the structure of dataset when project retrieval is performed. Furthermore, we should not ignore the importance of attribute weighting on the project retrieval and therefore on the accuracy. Unlike attribute selection methods that are performed once before validating the prediction method, the proposed method attempts to generate different weights for each individual test observation which takes into account the structures of training dataset. In addition, the attribute weights identified are based on a robust Kendall's coefficient of concordance approach rather than on trial-and error approach. Therefore, we may find each project is retrieved based on different weights.

Table1. Prediction accuracy for Desharnais

C	MMRE%	MdMRE%	MMER%	Pred(25)%
2	32.8	24.3	38.7	54.5
3	26.3	15.7	29.0	64.9
4	27.6	18.0	39.3	64.9
5	20.9	11.4	32.6	74.0
6	14.5	8.3	26.2	83.1
7	13.1	7.7	24.5	88.3
8	12.8	7.7	24.1	88.3
9	11.4	7.4	22.7	90.9
10	***11.3***	***7.2***	***21.8***	***91.1***
C: Number of Clusters				

Table2. Prediction accuracy for COCOMO

C	MMRE%	MdMRE%	MMER%	Pred(25)%
2	28.5	20.5	37.8	56.7
3	29.3	18.6	34.3	55.0
4	29.1	16.9	31.2	56.7
5	25.8	15.2	28.1	63.3
6	25.7	16.9	28.8	61.7
7	23.1	13.9	25.5	68.3
8	23.1	14.8	25.6	66.7
9	***19.9***	***13.9***	***22.9***	***70.0***
10	21.4	16.0	25.7	66.7
C: Number of Clusters				

To ensure that the results obtained are not by chance we investigated the statistical significance of weighted-FGRA on each dataset using one-sample Wilcoxon signed rank test for residuals as shown in Table 3, setting test value to zero. In this test if the resulting *p*-value is small ($p<0.05$), then the sample data are not symmetrical about the test value and therefore a statistically significant difference can be accepted between the sample median and the test value. The residuals obtained using the weighted-FGRA method were not significantly different from the test value zero. This suggests that the data does not give any reason to conclude that the residuals median differs from the hypothetical median (i.e. zero). So we can safely conclude that the medians of residuals generated by weighted-FGRA are not different from zero but it is not exactly same. Thus, there is advantage to these datasets obtaining their effort estimations using our proposed weighted-FGRA method.

Table3. Statistical significance of weighted FGRA			
Dataset	**Signed rank**	**Zval**	***p*-value**
Desharnais	1336	-0.84	0.4007
COCOMO	527.5	-1.06	0.2882

Figure 2 depicts the Boxplot of absolute residuals for Desharnais and COCOMO. In general, the Boxplots revealed that the boxes length is quite small which indicates reduced variability of absolute residuals. The median values for both datasets are quite similar and close to zero which revealed that at least half of predictions are accurate if we consider lower MMRE, MdMRE and higher Pred(25). The lower tails for COCOMO is much smaller than upper tails which mean the absolute residuals are skewed towards the minimum value. The figure shows many outliers for Desharnais more than COCOMO which may be related to the spread of effort values in Desharnais dataset.

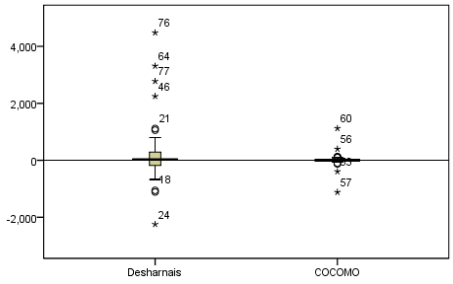

Figure 2. The Boxplot of residuals for weighted-FGRA

The performances of weighted FGRA method are investigated against conventional EBA method. For the comparison purpose we used weighted Euclidean distance and Exhaustive feature selection that are implemented in Angel tool [32]. EBA is trained by the similar procedure described above and the best variants on training set of weighted-FGRA are selected as the candidate for comparisons.

The results obtained in Table 4 shows the FGRA produced better estimation accuracy than conventional EBA. Our results demonstrate that applying attributes weights is an effective strategy to account for the influence of each attribute on the prediction accuracy. Prediction accuracy of weighted-FGRA outperformed the conventional EBA that uses equal weights. In other words, the conventional EBA method uses identified relevant attributes and treats them with equal weights (i.e. $w=1$ for each selected attribute and $w=0$ otherwise). The attributes selected by EBA were on the basis of optimizing *MMRE* results so that it is not surprisingly that it performs best in terms of this indicator.

Table 4. Comparison between weighted-FGRA and EBA				
	Desharnais		**COCOMO**	
	Weighted-FGRA	EBA	Weighted-FGRA	EBA
MMRE%	11.3	38.2	19.9	29.0
MdMRE%	7.2	30.2	13.9	25.0
MMER%	21.8	45.7	22.9	44.2
Pred(25)%	91.1	42.86	70.0	51.7

To ensure the weighted-FGRA outperformed conventional EBA we used Wilcoxon sum rank test. From Table 5 we found statistical significance between FGRA and EBA method. Suggesting that, there is difference if the predications generated using FGRA or other methods and based on the accuracy comparison in Table 4 we can safely conclude that our proposed method outperformed the conventional EBA.

Table5. Statistical significance of weighted FGRA Vs. EBA			
Weighted FGRA Vs. EBA	**Rank sum**	**Zval**	***p*-value**
Desharnais	3893	-7.49	<0.01**
COCOMO	3140	-2.6	0.0099**
****: statistical significant at 99%**			

Figures 3 and 4 show the Boxplot of absolute residuals for Desharnais and COCOMO dataset respectively. Both figures follow the same visual representation. The figures revealed that the box length of weighted-FGRA is much smaller than EBA which demonstrates reduced variability in absolute residuals. The box of weighted-FGRA overlays the lower tail (i.e. the absolute residuals are skewed towards the minimum value) which also presents accurate estimation than EBA for both datasets. The median and the range of absolute residuals of weighted-FGRA is smaller than median of EBA which shows that at least half of the predictions of weighted-FGRA are more accurate than EBA.

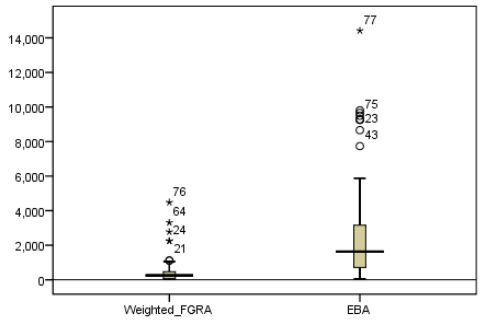

Figure 3. Comparison based on Boxplot of absolute residuals for Desharnais

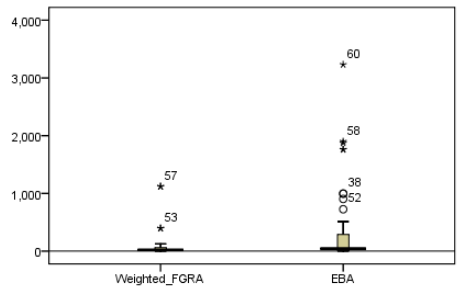

Figure 4. Comparison based on Boxplot of absolute residuals for COCOMO

7. CONCLUSIONS

The inherent uncertainty in software measurement increases the challenge in providing good estimates. Software effort estimation has to deal with considerable uncertainty and this paper has shown advantages in explicitly modeling this uncertainty in similarity measurement. Fuzzy logic and GRA have particularly shown their values to deal with uncertain and complex problem, especially when dealing with different data scale type in similarity measurement. The proposed similarity measure overcomes some limitations in previous similarity measures as explained in section one. Since each attribute has a different influence on project retrieval, we proposed a new approach based on Kendall's coefficient of concordance to reflect the influence of each attribute on the project retrieval. The proposed method shows it effectiveness in terms of using all available attributes without the need for attribute selection.

We also investigated the impact of number of Fuzzy clusters on the prediction accuracy. Typically, we found that when the number of clusters increases the corresponding accuracy is also improved, and generally 9 clusters best worked for the employed datasets. Our results also indicate improved prediction performance using this proposed method, and outperformed conventional EBA in terms of accuracy and statistical significance. The impact of attribute weighting approach to different datasets is still unknown therefore further investigation on other datasets may be required in the future.

8. ACKNOWLEDGMENTS

Authors would like to thank PROMISE for Desharnais and COCOMO datasets

9. REFERENCES

[1] Auer S., Trendowicz, A., Huanschmid, E., Biffl, S. 2006, Optimal Project feature Weights in Analogy-Bases Cost Estimation: Improvements and Limitations, Journal of IEEE Transaction on Software Engineering 32, 83-92.

[2] Azzeh, M., Neagu, D., Cowling, P. 2008, Fuzzy Feature subset Selection for Software Effort Estimation, *International workshop on software predictors PROMISE'08* (part of ICSE'08), Leipzig, Germany, pp.71-78.

[3] Azzeh, M., Neagu, D., Cowling, P. 2008, Software Project Similarity Measurement based on Fuzzy c-Means, *International Conference on software process*, Leipzig, Germany, pp. 123-134.

[4] Bezdek, J.C., 1981, Pattern Recognition with Fuzzy Objective Function Algorithms, Kluwer Academic Publishers, Norwell, MA, New York.

[5] 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.

[6] 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.

[7] Deng, J. 1989, Introduction to grey system theory, *Journal of Grey System* 1:1–24.

[8] Deng, J. 1989, Grey information space, *Journal of Grey System* 1: 103–117.

[9] Gower, J. C., 1971, A general coefficient of similarity and some of its properties, Journal of Boimetrics 27: 857-872.

[10] Hsu, C.J., Huang, C.Y. 2007, Improving Effort Estimation Accuracy by Weighted Grey relational Analysis During Software development, *14th Asia-Pacific Software Engineering Conference*, pp. 534-541.

[11] Huang, S-J., Chiu N-H., Chen L-W. 2007, Integration of the grey relational analysis with genetic algorithm for software effort estimation. *European Journal of operational and research* 188: 898-909.

[12] Huang, S.J., Chiu, N.H. 2006, optimization of analogy weights by genetic algorithm for software effort estimation. *Journal of Information & software technology*: 48 : 1034-1045

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

[14] ISBSG. 2007, International Software Benchmarking standards Group, Data repository release 10, web site: http://www.isbsg.org (visited 20 August 2008).

[15] Jorgensen, M., Indahl, U., Sjoberg, D. 2003, Software effort estimation by analogy and "regression toward the mean", *Journal of Systems and Software* 68: 253-262.

[16] Keung, J., Kitchenham, B. 2008. Experiments with Analogy-X for software cost estimation, *19th Australian Conference on software engineering*, pp. 229-238.

[17] Kendall M., Gibbons J. D., 1990, Rank Correlation methods. Fifth edition, Edward Arnold.

[18] Kirsopp, C., Shepperd, M. 2002, Case and Feature Subset Selection in Case-Based Software Project Effort Prediction, *Proceedings of 22nd International Conference on Knowledge-Based Systems and Applied Artificial Intelligence (SGAI'02)*.

[19] Li, J., Ruhe, G. 2008, Multi-criteria Decision Analysis for Customization of Estimation by Analogy Method AQUA+, *International workshop on software predictors PROMISE'08*, Leipzig, Germany, pp. 55-62.

[20] Li, J., Ruhe, G. 2008, Analysis of attribute weighting heuristics for analogy-based software effort estimation method AQUA+, Journal of Empirical Software Engineering 13: 63-96.

[21] Martin, C. L., Pasquier, J. L., Yanez, C. M., Gutierrez, A. T. 2005, Software Development Effort Estimation Using

Fuzzy Logic: A Case Study, *proceeding of Sixth Mexican International Conference on Computer Science (ENC'05)*, pp. 113-120.

[22] Mendes, E., Watson, I., Triggs, C., Mosley, N., Counsell, S. 2003, A comparative study of Cost Estimation models for web hypermedia applications, *Journal of Empirical Software Engineering* 8:163-193.

[23] Mittas, N., Athanasiades, M., Angelis, L. 2007, improving analogy-based software cost estimation by a resampling Method, *Journal of Information & software technology*.

[24] Shepperd, M. J., Schofield, C. 1997 Estimating Software Project Effort Using Analogies, *IEEE Transaction on Software Engineering* 23:736-743.

[25] Song, Q., Shepperd, M., Mair, C. 2005 Using Grey Relational Analysis to Predict Software Effort with Small Data Sets, *Proceedings of the 11th International Symposium on Software Metrics (METRICS'05)*, pp. 35-45.

[26] Xie, X. L., Beni, G. 1991, A validity measure for Fuzzy clustering, *IEEE Transactions on Pattern Analysis Machine Intelligence* 13: 841-847.

[27] Zadeh, L. 1997, Toward a theory of Fuzzy information granulation and its centrality in human reasoning and Fuzzy logic. *Journal Fuzzy sets and Systems* 90: 111-127.