Applied Soft Computing xxx (2016) xxx–xxx

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Improved estimation of software development effort using Classical and Fuzzy Analogy ensembles

Ali Idri [a,*], Mohamed Hosni [a], Alain Abran [b]

[a] Software Projects Management Research Team, ENSIAS, Mohammed V University, Rabat, Morocco
[b] Department of Software Engineering, École de Technologie Supérieure, Montréal, Canada

ARTICLE INFO

Article history:
Received 31 December 2015
Received in revised form 29 July 2016
Accepted 4 August 2016
Available online xxx

Keywords:
Software development effort estimation
Ensemble effort estimation
Analogy
Fuzzy logic

ABSTRACT

Delivering an accurate estimate of software development effort plays a decisive role in successful management of a software project. Therefore, several effort estimation techniques have been proposed including analogy based techniques. However, despite the large number of proposed techniques, none has outperformed the others in all circumstances and previous studies have recommended generating estimation from ensembles of various single techniques rather than using only one solo technique. Hence, this paper proposes two types of homogeneous ensembles based on single Classical Analogy or single Fuzzy Analogy for the first time. To evaluate this proposal, we conducted an empirical study with 100/60 variants of Classical/Fuzzy Analogy techniques respectively. These variants were assessed using standardized accuracy and effect size criteria over seven datasets. Thereafter, these variants were clustered using the Scott-Knott statistical test and ranked using four unbiased errors measures. Moreover, three linear combiners were used to combine the single estimates. The results show that there is no best single Classical/Fuzzy Analogy technique across all datasets, and the constructed ensembles (Classical/Fuzzy Analogy ensembles) are often ranked first and their performances are, in general, higher than the single techniques. Furthermore, Fuzzy Analogy ensembles achieve better performance than Classical Analogy ensembles and there is no best Classical/Fuzzy ensemble across all datasets and no evidence concerning the best combiner.

1. Introduction

Estimating the effort required to develop a software project is a key activity in software project management and Wen et al. [1] define the Software Development Effort Estimation (SDEE) as 'the process of predicting the effort required to develop a software system'. Both under and overestimation of effort may lead to loosing contracts for software companies and-or to failure in software project management [2]. The effort is usually measured in person-month/hours, that is the number of person-month/hours spent in developing a given software project [3]. The success of any software development project depends significantly on how accurate the estimation of its required effort was; hence, estimating effort accurately remains a challenge for the SDEE community.

Several effort estimation techniques have been proposed and can be grouped into three main categories [4]: expert judgment which consists on consulting one or more experts to determine the effort estimation [5]; parametric techniques which are derived from the statistical and/or numerical analysis of historical project data [6–8]; and machine learning (ML) techniques which are based on a set of artificial intelligence techniques, such as artificial neural networks (ANN), genetic algorithms (GA), analogy-based or case based reasoning (CBR), decision trees, and genetic programming [1,9].

Jørgensen and Shepperd [10] conducted a systematic literature review (SLR) of software effort estimation studies performed between 2000 and 2004. They identified 11 estimating techniques that had been used in the 304 selected studies and found that the regression technique was the dominant one (i.e. adopted by 49% of their selected studies). In the last two decades, the ML techniques have received increasing attention from the software researchers as illustrated by the Wen et al. [1] SLR based on ML techniques used in SDEE area from 1990 until 2010. Their study identified in their 84 selected studies eight ML techniques: CBR or analogy [11], ANN [12], decision trees [13], bayesian networks [14], support vector regression [15], genetic algorithms [16], genetic programming [17], and association rules [18]. They found that CBR and ANN were the most used techniques, with 37% and 26% respectively.

Analogy-based reasoning approaches have proved to be promising techniques in software effort prediction field and their use is increasing amongst software researchers. In fact, the SLR conducted

* Corresponding author:
E-mail addresses: ali.idri@um5.ac.ma (A. Idri), hosni.mohamed1@gmail.com (M. Hosni), alain.abran@etsmtl.ca (A. Abran).

http://dx.doi.org/10.1016/j.asoc.2016.08.012
1568-4946/

by Jørgensen and Shepperd have shown that the use of analogy-based software effort estimation models is increasing over time (from 9% in the period 1990–1999 to 15% in the period 2000–2004). Estimation by analogy has two main advantages: (1) they can model complex relationships between the dependent variable (such as effort or cost) and the independent variables (cost drivers)[19–23]; and (2) they may be easily understood by users as they mimic the human problem solving approach (as opposed to black-box approaches like artificial neural network) [19,23,24].

Despite their advantages, Classical analogy-based effort estimation techniques are limited by their inability to adequately handle linguistic values and to manage imprecision and uncertainty. Therefore, a new technique, called Fuzzy Analogy was proposed and evaluated by Idri et al. [9,25–31]. This technique extends the Classical Analogy method by integrating fuzzy logic to adequately handle categorical features (refer to Section 2.2.2 for more details on Fuzzy Analogy). Unlike Classical Analogy which represents categorical data by means of classical intervals, Fuzzy Analogy deals with categorical/numerical values by transforming them into linguistic ones represented by fuzzy sets [26]. The use of linguistic values instead of numbers or classical intervals serves many purposes [32]:

- They are easier to understand than numerical values;
- they make allowance for imprecision;
- they generalize numbers (only when precise information is available); and
- they accept the finite ability of the human mind to resolve detail and store precise information.

In fact, the systematic map and review of analogy-based software effort estimation techniques (ASEE) performed by Idri et al. [33] showed that in 65 of their selected papers ASEE techniques tend to yield acceptable estimates: the means of MMRE, MdMRE and Pred(25) were 49.8%, 29.4% and 51.2% respectively. Also, it was found that ASEE techniques outperform the other estimation techniques in most of their 65 selected papers. In addition, the estimation accuracy is improved when analogy is used in combination with another technique such as fuzzy logic and genetic algorithms to generate estimates.

However, despite the large number of techniques proposed and investigated in SDEE, none of them has proved to be the best under all circumstances, since their performances vary from one dataset to another, which makes them unstable [34,35]. Thus, Shepperd and Kadoda [35], suggested that it would be more fruitful to determine the best technique in a particular context rather than determining the best absolute single technique, and they concluded that the combination of multiple estimators performs better than any single estimator. This conclusion was confirmed by other studies [36–38]. The prior studies in SDEE found that the combination of many single estimation techniques into an ensemble may lead to more accurate estimates than those obtained when using a single one. Furthermore, Seni and Elder [39] confirmed the same conclusion when they used ensemble techniques for prediction in data mining area. Note that in the literature on SDEE various authors use different terms for the same concepts such as single-solo-base, and models-techniques-methods. For sake of clarity and consistency, we have adopted the following terms: solo technique (one variant of one SDEE technique) and ensemble technique (a combination of more than two variants of one or many SDEE techniques) in the subsequent sections of this paper.

In SDEE, a combination of more than one solo estimation technique into an ensemble under a specific combination rule is called Ensemble Effort Estimation (EEE) techniques. The estimation of a given ensemble is the aggregation, by means of a combination rule, of effort estimation values (solo estimates) provided by solo techniques that compose the ensemble – see Fig. 1.

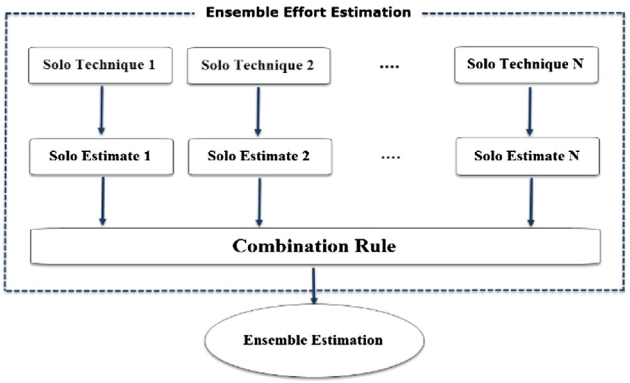


Fig. 1. Ensemble Effort Estimation process.

The literature on SDEE defines two types of EEE techniques:

A Homogeneous EEE to refer to: (1) an ensemble that combines at least two different solo variants of the same SDEE technique or (2) a combination of one ensemble learning (such as. Bagging [40], Negative Correlation [41] or Random [42]) and one solo technique [43–46].
B Heterogeneous EEE to refer to an ensemble that combines at least two different solo techniques of two different SDEE techniques [34,36,37,47].

Since each solo estimation technique has its strengths and weaknesses [1,10,33], the objective of EEE is to mitigate these weaknesses and consolidate advantages by combining solo techniques through an ensemble, which may lead to more accurate estimates than can be obtained from solo techniques [34]. A number of empirical studies have been performed to assess whether EEE techniques may lead to better estimation accuracy. The majority of studies performed have combined solo ML techniques into homogeneous or heterogeneous ensembles [34,36,43,44]. Among these solo ML techniques, the crisp ASEE technique was used by Kocaguneli et al. [36] to generate only heterogeneous ensembles, while it was used by Azzeh et al. [44] and Wu et al. [45] to generate only homogeneous ensembles.

Even though a number of EEE techniques have been proposed in the literature [36,44,45,47] and to the best of our knowledge, no study has yet focused on the EEE techniques construction based on Fuzzy Analogy. Thus, since Fuzzy Analogy has been proposed to overcome a major limitation of Classical Analogy when dealing with categorical data, this study evaluates whether Fuzzy Analogy ensembles outperform solo Fuzzy Analogy techniques as well as Classical Analogy ensembles. Furthermore, although Classical Analogy techniques were the subject of some studies in EEE (e.g., Azzeh et al. [44] varied the number of closest analogues and adaptation strategy, Wu et al. [45] varied similarity measures and adaptation strategy and Kocaguneli et al. [36] varied the number of closest analogues), this study is the first varying all the parameters of Classical Analogy as it is explained in Section 3.1 (i.e. similarity measures, number of closest analogues and adaptation strategy). From a methodology viewpoint, this study uses the same experimental protocol of Azzeh et al. [44] except that this study uses two analogy solo techniques (Classical/Fuzzy Analogy instead of only Classical Analogy) and three combination rules (arithmetic average, median and inverse rank weighted average instead of only arithmetic average). In addition, the general idea of selecting the best solo techniques based on their error of this study is the same

Please cite this article in press as: A. Idri, et al., Improved estimation of software development effort using Classical and Fuzzy Analogy ensembles, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.08.012

as Kocaguneli et al. [36] and Azhar et al. [47] even if this study uses a different experimental protocol.

Specifically, this paper aims at investigating the potential of two types of ASEE techniques (Classical and Fuzzy Analogy) into homogeneous ensembles using three linear rules over seven datasets. To this end, four research questions (RQs) are investigated:

(RQ1): Is there evidence that the ensembles based on Classical Analogy outperform solo Classical Analogy techniques?

(RQ2): Is there evidence that the ensembles based on Fuzzy Analogy outperform other solo Fuzzy Analogy techniques?

(RQ3): Among the three linear rules (combiners) used in this study, which of them provides a better accuracy for the Classical/Fuzzy Analogy ensembles?

(RQ4): Do Fuzzy Analogy ensembles outperform Classical Analogy ensembles?

The main contributions of this empirical study are the following:

- Evaluating a large number of solo Classical/Fuzzy Analogy;
- Evaluating Classical Analogy ensembles by varying all the Classical Analogy parameters;
- Evaluating whether Fuzzy Analogy ensembles outperform solo Fuzzy Analogy techniques; and
- Evaluating whether Fuzzy Analogy ensembles outperform Classical Analogy ensembles.

The structure of the paper is as follows: Section 2 presents an overview of EEE techniques and the two ASEE techniques used in this paper. Section 3 describes the variants of Classical and Fuzzy Analogy used in this study. Section 4 presents the empirical methodology pursued throughout this research. Section 5 presents and discusses the empirical results. Section 6 presents the threats to validity of this study. Section 7 presents the conclusions and future works.

2. Background and related work

This section presents an overview of EEE techniques and related work investigating EEE techniques and their findings. Finally, the two ASEE techniques investigated in this paper are presented and explained.

2.1. Ensemble effort estimation techniques

EEE techniques have been the subject of many studies in SDEE and have been proposed in order to improve the prediction accuracy of solo techniques. While solo estimation techniques may have their own advantages and limitations, EEE techniques aim to overcome their limitations and consolidate their advantages. An EEE technique differs from any solo estimation technique as it establishes an ensemble composed by different solo techniques. EEE technique attempts to reduce the estimation errors by combining different estimates provided by its solo techniques.

According to Fig. 1, building ensembles generally involves three stages: (1) Select a set of solo techniques; (2) Predict the effort with the selected solo techniques separately; and (3) Aggregate the solo estimates by means of a specific combination rule. However, in order to achieve a high prediction accuracy by an ensemble (e.g. more accurate estimation than its solo techniques), the solo techniques that compose the ensemble should satisfy two conditions: high accuracy and diversity [48]. In other words, the estimation of an EEE technique is influenced by the estimation of each solo technique. Therefore, poor solo estimates may lead to poor estimation of ensemble. Hence, the solo techniques should be as accurate as possible. Also, the solo techniques should be diverse (e.g. make different errors in the same data point). Consequently, each solo technique can cancel the estimation errors done by other solo techniques. Otherwise, an ensemble that integrates non-diverse solo techniques may produce a lower estimation accuracy than its solo techniques [2].

To examine the research on the use of EEE techniques in SDEE, Idri et al. [49] have performed a systematic literature review in which 24 EEE studies published between January 2000 and January 2016 were selected. The review was carried out to analyze EEE techniques from six viewpoints: solo models used to construct ensembles, ensemble estimation accuracy, rules used to combine solo estimates, accuracy comparison of EEE techniques with solo models, accuracy comparison between EEE techniques and methodologies used to construct ensemble methods. The principal findings of their SLR were as follows:

- Sixteen solo models were used to construct both types of ensembles. The machine learning solo models are the most frequently used.
- Both types of EEE ensembles were investigated: Heterogeneous and Homogeneous ensembles. The homogeneous ones were the most commonly investigated since they were used in 17 out of 24 selected papers.
- Twenty combiners were used by EEE techniques. They can be grouped into two families of rules: linear and non-linear. The linear ones were the most frequently used and in particular the arithmetic mean combiner.
- EEE techniques are more accurate than solo techniques and in particular when they used linear combination rules.
- There is no confirmation concerning the best EEE techniques.
- Four methodologies were used to construct EEE techniques: three of them rely mainly on statistical tests (such as Wilcoxon test and Scott-knott) based on the error to select the best candidate solo techniques.

Since this study is concerned with the use of analogy ensembles, an overview of related work is presented. Table 1 summarizes the findings of eight studies selected in Idri et al. [49] that used analogy/CBR to construct both types of ensembles (i.e. homogeneous and heterogeneous) as well as information about the accuracy criteria used to assess techniques, combination rules and methodology used to select ensemble members in each study.

It can be noticed from Table 1 that:

(1) To the exception of the work conducted by Azzeh et al. [44], all EEE studies of Table 1 used the MMRE accuracy criterion which has been evaluated as biased toward underestimates [53];
(2) all studies of Table 1 used linear combination rules to generate the estimation of their ensembles;
(3) Analogy/CBR solo techniques were used in five and three EEE studies to construct heterogeneous and homogeneous ensembles respectively; and
(4) to the exception of the work conducted by Kocaguneli et al. [52], EEE techniques are more accurate than solo models. In particular, the study conducted by Kocaguneli et al. [52], that is a projection of Khoshgoftaar et al. [54] work in SDEE area, shows that there is no improvement in accuracy when a combination of learners is used to predict the effort. Note that it is the same conclusion as Khoshgoftaar's et al. work in the software quality area. However, Kocaguneli et al. admit in their subsequent work on EEE techniques [36] that they made a mistake when they presumed that all techniques are candidates for combination into ensembles.

Please cite this article in press as: A. Idri, et al., Improved estimation of software development effort using Classical and Fuzzy Analogy ensembles, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.08.012

Table 1
Summary of the literature review of Analogy Ensemble Effort Estimation in [49].

Author (s)	Candidate models	Accuracy criteria	Combination rule used	Findings/Methodology
Azzeh et al. [44]	40 variants of CBR	SA, Effect Size, MAE, AE, MAE, LSD, MBRE, MIBRE.	Mean.	The selection of the appropriate solo models for each dataset was done in 4 steps: (1) Examination of the prediction accuracy of each model based on SA and Effect Size: any method that did not achieve an effect size greater than 0.5 and an MAE greater than that of random guessing was eliminated. (2) The best cluster of methods was identified by means of the SK algorithm. (3) Methods that belonged to the best cluster were scored by applying the Borda Counting method based on 4 error measures: MAE, LSD, MBRE, and MIBRE. (4) Ensembles were constructed by combining the top 2, 3, 4, etc. models under a mean combiner. This process was done for 8 datasets separately. They concluded that ensembles of adjusted analogy were stable and may be superior to solo adjusted analogy.
Wu et al. [45]	18 variants of CBR	MMRE, MdMRE, Pred (25).	Mean, Median, Weighted mean combination, outperformance combination.	They constructed homogeneous EEE based on different variants of CBR. These variants of CBR were constructed by using different similarity measures and adaptation strategies. The weights of features were optimized using particle swarm optimization. The ensemble contains 6 variants of CBR using three combination rules. They concluded that their proposed ensemble was more accurate than solo CBR techniques, and the weighted mean combination can get the best result. The evaluation was done using two datasets through a threefold cross validation approach.
Mittas et al. [50]	Iterated bagging + CBR	MMRE, MdMRE, MMER, MdMER, MSE.	Mean.	The experiments conducted applying the proposed ensemble to artificial and three real datasets using LOOCV evaluation technique. The results showed significant improvement under various accuracy measures over single CBR technique.
Kocaguneli et al. [36]	Analogy with 1 nearest neighbor Analogy with 5 nearest neighbors Stepwise regression CART with pruning CART without pruning Neural net with two hidden layers Simple linear regression Principal components regression Partial least squares regression	MAE, MMRE, MdMRE, MMER, Pred(25), MBRE, MIBRE.	Mean, Median, IRWM.	They performed a study in which they evaluate the potential of heterogonous and homogenous EEE techniques. They used 9 learners and 10 preprocessing options. A learner with a preprocessing stage is called a solo technique. Each solo technique was evaluated by using 7 performance measures over 20 datasets. Each of the 90 solo techniques was compared to the other 89 over all 20 datasets (i.e. a round-robin process) using the leave one out cross validation (LOOCV). The comparison was made by applying an algorithm that uses the Wilcoxon nonparametric statistical test. The outputs of the algorithm are the number of times that the given solo technique wins, ties and loses against the remaining techniques. Thereafter, the solo techniques were stored according to their number of losses (a smaller number of losses means best). They selected 13 solo techniques: these are considered to be superior and used to construct ensembles. The construction of ensemble is done with top two, four, eight and thirteen superior solo techniques, under 3 combiners (mean, median, inverse ranked weighted mean) which gives 12 ensembles in total. This study concluded that the ensemble techniques are considerably superior to the solo techniques.
Azhar et al. [47]	Analogy with 1 nearest neighbor Analogy with 5 nearest neighbors Stepwise regression CART with pruning CART without pruning Neural net with two hidden layers Simple linear regression Principal components regression Partial least squares regression	MAE, MMRE, MdMRE, MMER, Pred(25), MBRE, MIBRE,	Mean, Median, IRWM.	They conducted a study in which they used two methodologies to construct ensembles. The first is the replication of Kocaguneli's work with some differences: using one dataset (Tukutuku), choice of 16 superior solo techniques and the construction of ensembles from the top two, four, eight, twelve and sixteen superior techniques (15 ensembles). The second is by using a Scott-Knott algorithm based on absolute errors of 90 solo techniques calculated through LOOCV validation. The best subgroup contains 19 solo techniques. Two ensembles were constructed by combining these 19 solo techniques using the median and mean rules. The conclusion of this work were the same as the results reported by Kocaguneli et al. [36].

Please cite this article in press as: A. Idri, et al., Improved estimation of software development effort using Classical and Fuzzy Analogy ensembles, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.08.012

Table 1 (*Continued*)

Author (s)	Candidate models	Accuracy criteria	Combination rule used	Findings/Methodology
Elish [37]	*K*-nn (CBR) + SVR + radial basis function network (RBF) + MLP +DT	MMRE, Pred (25), Evaluation Function.	Median.	The results obtained suggest that solo techniques are not reliable since their performance is inconsistent and unstable across different datasets. Further, the ensembles provide more reliable performance than solo techniques. In three out the five datasets, the ensemble model outperformed the individual models. In the other two datasets, the ensemble model achieved the second best performance. The validation method used is LOOCV.
Hsu et al. [51]	COCOMO + Linear regression + Analogy + Grey relational analysis + ANN.	MMRE, MdMRE, Pred (25).	Equally weighted, Median weighted, Weighted adjustment based on a criterion (WCR).	The results obtained prove that the proposed combinations of solo methods can be a useful method for improving software effort estimations. Also, the WCR combination improves significantly from the best single method. The experiments were performed using 7 datasets.
Kocaguneli et al. [52]	Gaussian Process + MLP + RBF + SVR + K-nn (CBR) + Locally Weighted Learning + Bagging (fast DT) + Additive Regression with Decision Stump + Random subspace + Decision Stump + M5P + Conjunctive Rule + Decision Table	MMRE, Pred (30).	Average.	Their findings suggest that the utilization of complex machine learning algorithms (ensemble) does not necessarily result in higher prediction performances since the proposed ensemble did not improve the estimation accuracy in the three datasets used.

MMRE: Mean Magnitude of Relative Error; MdMRE: Median Magnitude of Relative Error; Pred(25): predictions falling within 25 percent of the actual values; MBRE: Mean Balanced Relative Error; MIBRE: Mean Inverted Balanced Relative Error; MAE: Mean Absolute Error; LSD: Logarithmic Standard Deviation; MSE: Mean Squared Error; MMER: Magnitude of Relative Error to the Estimate; IRWM: Inverse Ranked Weighted Mean; CART: Classification and Regression trees.

2.2. Analogy-based software effort estimation – ASEE: an overview

Analogy-based reasoning (also known as case-based reasoning – CBR) is one of the ML effort estimation techniques. The CBR techniques use the assumption that similar software projects are most likely to have similar efforts: conceptually, the effort required to develop a new software project is based on the known efforts of its closest analogues.

The CBR technique is a special form of analogy that follows the process described as a 4-stage cycle consisting of [55]:

(1) Retrieve the most similar case or cases to the target problem.
(2) Reuse the past information and solution to solve the new problem.
(3) Revise the proposed solution to better adapt the target problem.
(4) Retain the parts of current experience in the case-base for future problem solving.

This study used two solo analogy-based estimation techniques: Classical Analogy and Fuzzy Analogy.

2.2.1. Classical Analogy

Classical Analogy has been proposed as a viable alternative to other software cost/effort estimation techniques [11]. The use of Classical Analogy to estimate software effort estimation involves 3 steps: (1) Identification of cases; (2) Retrieval of similar projects; and (3) Case adaptation. These steps are described below.

a.) Identification of cases

The aim of this step consists on selecting the optimal subset of features that describe the software project. In fact, each project is described by a set of attributes that are believed to give the most accurate estimation. Those attributes can be continuous or categorical, and they are usually normalized (scaling) to serve as a basis for finding the historical projects that are similar to the target project. Thus, the estimated effort is directly influenced by the selected features.

Determining the optimal subset of features remains a complex task in the estimation analogy process and in SDEE in general, since it has a direct impact on the similarity evaluation. Therefore, the features selection in SDEE area has been the subject of many studies and different techniques have been proposed which are based on statistical methods, fuzzy logic and genetic algorithms [23,27,56,57].

b.) Retrieval of similar case

The purpose of this step is to calculate the level of similarity between the target project and the historical projects in order to identify its similar projects and select the closest ones. Several similarity measures have been proposed in literature. The ones that will be used in this study are described in Section 3.1. Note that the similarity between projects was assessed using the formula of Eq. (1):

$$d\left(P_i, P_j\right) = \frac{\sum_{l=1}^{d} D\left(\mathrm{P_{i,l}}, \mathrm{P_{j,l}}\right)}{d} \quad (1)$$

Where:

- d is the number of attributes that describe the projects;
- $\mathrm{P_{i,l}}$ and $\mathrm{P_{j,l}}$ are the values of the *l*th attribute of projects $\mathrm{P_i}$ and $\mathrm{P_j}$ respectively; and
- D is the distance between projects $\mathrm{P_i}$ and $\mathrm{P_j}$ for *l*th attribute. (The distance measures used in this study are listed in Section 3.1).

c.) Case adaptation

This step consists on generating the estimation of effort for the target project. This estimation is computed by aggregating the actual efforts of the similar historical projects using an adaptation technique. Several adaption techniques have been used such as average [11], median [58] and inverse ranked weighted mean [59].

Please cite this article in press as: A. Idri, et al., Improved estimation of software development effort using Classical and Fuzzy Analogy ensembles, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.08.012

2.2.2. Fuzzy Analogy

Fuzzy Analogy has been proposed for the first time in software effort estimation in 2002 by Idri et al. [9] in order to overcome the limitations presented by Classical Analogy when dealing with categorical features that are derived from numerical data. As in Classical Analogy, Fuzzy Analogy involves three steps: (1) Identification of cases; (2) Retrieval of similar projects; and (3): Case adaptation. Each step is a fuzzification of its equivalent in the Classical Analogy procedure. These steps are described below.

a.) Identification of cases

The goal of this step is to describe each project by a set of relevant and independent attributes. These attributes can be measured by numerical as well as linguistic values. In Fuzzy Analogy, numerical values are transformed into linguistic ones rather than into classical interval as in the Classical Analogy technique. Let us suppose that a project P is described by M numerical and linguistic variables (V_j). Then, for each variable V_j, a measure with linguistic values is defined (A_k^j). Each linguistic value A_k^j is represented by a fuzzy set with a membership function $\boldsymbol{\mu}_{A_k^j}$. These fuzzy sets and their membership functions can be built in two ways: (1) empirically using expert knowledge [60], or (2) automatically using clustering techniques [61].

In the case when the descriptions of software attributes were insufficient to empirically build their fuzzy representations, Fuzzy Analogy used an automated process to build fuzzy sets and their membership functions [61]. The proposed fuzzy set generation process is based on the Fuzzy C-Means clustering technique (FCM) and a real coded genetic algorithm (RCGA). This process consists of two main steps, as shown in Fig. 2. First, the well-known FCM algorithm is used to generate the desired clusters (fuzzy sets)[62].

FCM was first proposed by Dunn [63] and generalized by Bezdek [62]. It is one of the most popular clustering methods. It is an iterative algorithm designed to find cluster centers that minimize the objective function of Eq. (2):

$$MinJ_p(U, C) = \sum_{i=1}^{i=n}\sum_{j=1}^{j=c}\left(u_{ij}\right)^m \|x_i - c_j\|^2 \tag{2}$$

$$\sum_{i=u_{ij}}^{c} = 1, \forall j = 1, \ldots, n$$

Where:

- c is the desired number of clusters;
- n is the number of data instances;
- U = (u_{ij}) is the partition matrix, containing the membership values of all data points in all clusters;
- m is a control parameter of fuzziness;
- X = $\{x_1, \ldots, x_n\}$ is a dataset of points; and
- C = (c_i) is the set of cluster centers.

Second, an RCGA is used to build membership functions for these fuzzy sets [64,65]. These membership functions can be trapezoidal, triangular, or Gaussian. The RCGA builds a set of membership functions (μ_j), $1 \le j \le c$, which interpolates and minimizes the mean square error, defined by Eq. (3):

$$MSE(\mu_1, \ldots, \mu_c) = \frac{1}{n}\sum_{j=1}^{j=n}\|(\mu_1(x_j), \ldots, \mu_c(x_j), -(u_{1j}, \ldots, u_{cj})\|^2 \tag{3}$$

Subject to $\sum_{i=1}^{c}\mu_j(x_i) = 1$, for all x_i and $\mu_j(x_i) = u_{ij}$, $1 \le i \le n$; $1 \le j \le c$.

b.) Retrieval of similar cases

The purpose of this step is to measure the similarity between the target project and the historical projects knowing that the projects are described by linguistic values such as 'high', 'very low' and 'low'. To achieve that, Fuzzy Analogy proposes a set of new measures based on fuzzy logic [66]. The process of measuring the similarity using these measures involves two steps:

- Individual similarities: assessing the similarity between two projects P_1 and P_2, according to each individual attribute V_j describing P_1 and P_2 by means of one of the formulas (4.1) or (4.2) of Eq. (4)

$$S_{V_j}(P_1, P_2) = \left\{\begin{array}{c} max - \min\ aggregation \\ max_k min\left(\mu_{A_K^l}(P_1), \mu_{A_K^l}(P_2)\right) \end{array}\right\}(4.1)$$
$$\left\{\begin{array}{c} sum - product\quad aggregation \\ \sum_{k}^{\mu_{A_K^l}}(P_1) * \mu_{A_K^l}(P_2) \end{array}\right\}(4.2) \tag{4}$$

- Global similarities: evaluate the overall similarity $S(P_1, P_2)$ by aggregating the individual similarities $S_{V_j}(P_1, P_2)$ using Regular Increasing Monotone (RIM) linguistic quantifiers, such as 'all', 'most', 'many', or 'there exists'. The choice of the appropriate RIM linguistic quantifier, Q, depends on the characteristics and needs of each environment. Q (Eq. (5)) indicates the proportion of individual distances that we feel necessary for a good evaluation of the overall distance. For example, if we choose the linguistic quantifier 'all', that means all the individual similarities were considered in the evaluation of $S(P_1, P_2)$. The choice of the linguistic quantifier used in this study is explained in Section 3.2. The overall similarity of P_1 and P_2, $S(P_1, P_2)$ is given by one of the formulas of Eq. (6).

$$Q = r^{\alpha}; \alpha > 0 \tag{5}$$

$$S(P_1, P_2) = \left\{\begin{array}{c} \text{All of } \left(S_{V_j}(P_1, P_2)\right) \\ \text{Most of } \left(S_{V_j}(P_1, P_2)\right) \\ \text{Many of } \left(S_{V_j}(P_1, P_2)\right) \\ \ldots \\ \text{There exists of } \left(S_{V_j}(P_1, P_2)\right) \end{array}\right. \tag{6}$$

c.) Case adaptation

The goal of this step is to calculate an estimate for the new project (P) by using the actual effort values of similar projects. This step involves two stages:

1. Decide how many similar projects will be used in the adaptation phase to estimate the effort of the new project. Fuzzy Analogy has proposed a strategy based on the similarity measures $S(P, P_i)$ and the definition adopted in the studied environment for the proposition 'P_i is a project that is highly similar to P'. Intuitively, P_i is highly similar to P if $S(P, P_i)$ is in the vicinity of 1. To represent

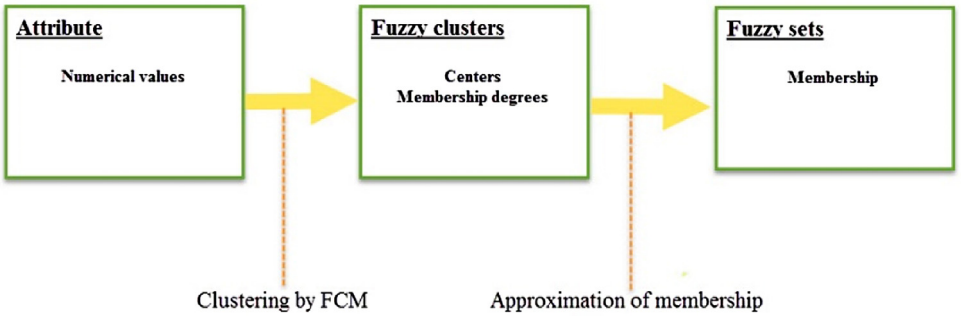


Fig. 2. Fuzzy set generation process.

the value 'vicinity of 1', we use a fuzzy set defined in the interval [0,1].
2. Adapt the chosen analogies in order to generate an estimate for the new project. Fuzzy Analogy uses the weighted mean of all known effort projects in the dataset, the weights being the similarity distance.

The formula is given by Eq. (7).

$$Effort(P) = \frac{\sum_{i=1}^{N} \mu_{vincity\ of\ 1}(S(P,P_i)) * Effort(P_i)}{\sum_{i=1}^{N} \mu_{vincity\ of\ 1}(S(P,P_i))} \quad (7)$$

3. Variants of Classical and Fuzzy Analogy

In this section, we define the variants of Classical Analogy as well as Fuzzy Analogy that will be considered as candidate solo techniques to construct the ensembles.

3.1. 100 Variants of Classical Analogy techniques

When using Classical Analogy there is a set of parameters to decide upon [67]. The prior works claim that combination of different parameters gave different prediction accuracy [19,45,68]. These parameters are: Feature subset selection, similarity measure, scaling, number of analogies and analogy adaptation. In this study three parameters were varied:

- Similarity Measure: it measures the level of similarity between projects. Several similarity measures based on projects distance have been proposed in the literature [69]. This study uses five measures of distance: un-weighted Euclidean distance, Manhattan distance also known as City block, Minkowski distance, Chebyshev distance and the Squared-chord distance which are presented in Eqs. (8)–(12) respectively and used in combination with Eq. (1). These distance measures were previously investigated in many ASEE studies and have led to diverse accuracy predictions [19,31,58,68]. Hence, this research aims to evaluate their impacts on the accuracy of Classical Analogy ensembles.

$$\text{Un-weighted Euclidean distance } d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad (8)$$

$$\text{Manhattan distance } d(x,y) = \sum_{i=1}^{n}|x_i - y_i| \quad (9)$$

$$\text{Minkowski distance } d(x,y) = \sqrt[p]{\sum_{i=1}^{n}(x_i - y_i)^p} \quad (10)$$

$$\text{Chebyshev distance } d(x,y) = |x-y|\max_{1\le i\le n} \quad (11)$$

$$\text{Squared-chord distance } d(x,y) = \sum_{i=1}^{n}(\sqrt{x_i} - \sqrt{y_i})^2 \quad (12)$$

Where p is an integer, $i = 1, 2, \ldots, n$.

In this study, the parameter p for Minkowski distance (Eq. (10)) was fixed to 3 as in [68].

- Number of analogies: The number of analogies (k) refers to the number of most similar cases used to generate the estimation. Several studies in SDEE investigated the impact of the number of analogues on the estimation accuracy [19,24,70]. The literature recommended that it is sufficient to use 1–5 nearest analogies in order to construct accurate Classical Analogy technique. Therefore, we adopt this recommendation and then the values of k were varied from 1 to 5.
- Analogy adaptation: consists on selecting the adaptation strategy to derive the estimation effort. In this study, four adaptation strategies were used: arithmetic mean [11], arithmetic median [58], inverse distance weighted mean and inverse ranked weighted mean [59].

Note that before evaluating the similarity between two projects, the features must be normalized in order to keep their degrees of influence the same. In this study we use max-min normalization rule.

Summing up, 100 variants of Classical Analogy technique are evaluated through several accuracy measures and datasets. (5 distances * 5 analogues * 4 adaptation strategies). To the best of our knowledge this is the first EEE study that uses this number of solo Classical Analogy techniques to build EEE techniques.

3.2. 60 Variants of Fuzzy Analogy techniques

As for Classical Analogy, Fuzzy Analogy has many parameters to decide upon. These parameters are: α-RIM linguistic quantifier, parameter of fuzziness (m), shape of membership functions, number of fuzzy clusters, and case adaptation. In this study, three parameters were varied: the parameter of fuzziness (m), the shape of membership functions and the number of fuzzy clusters. Earlier works on Fuzzy Analogy reported that the variation of these parameters led to diverse accuracy results [26,28,29,71].

- Parameter of fuzziness (m): Prior works on FCM reported that the result of clustering is significantly affected by the variation of parameter m [72,73]. However, there is no recommended value of m. Recently, Zhou et al. [74] reported that the optimal interval of the parameter m is from 2 to 3.5. In our study, we ranged the parameter m from 1.5 to 3.5 with increments of 0.5.
- Shape of membership functions: Many shapes of membership functions can be used such as Triangular, Trapezoidal and Gaussian. However, the works by Idri et al. [26,28,29,71] conclude

Please cite this article in press as: A. Idri, et al., Improved estimation of software development effort using Classical and Fuzzy Analogy ensembles, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.08.012

that the estimation of Fuzzy Analogy achieves a better accuracy when Triangular or Trapezoidal rather than Gaussian shapes were used. Therefore, this study adopts the Triangular and Trapezoidal shapes to construct EEE techniques.
- Number of Fuzzy Clusters: In this study, the desired number of clusters was varied within the interval [2,7] for all features in each dataset. Indeed, the aim of clustering is to reduce a set of an infinite number of values to a set of finite ones, usually between 2 and 7, in order to deal adequately with some datamining challenges such as outliers, missing data, and categorical vs numerical data [75,76].

Prior works of Idri et al. found that the α-RIM linguistic quantifier has a direct impact on estimation results [9,26,28,71]. When α tends towards zero, a lower accuracy is obtained, because the overall similarity takes into account fewer attributes among all those describing software projects. Otherwise, when α tends towards ∞, the accuracy increases, since additional attributes are considered in the evaluation of the overall similarity. Hence this study fixed α to 500 in all evaluations. As for individual similarities, the max-min aggregation (Eq. (4.1)) was used. Concerning the case adaptation, we used the weighted mean of all known effort projects in the dataset, the weights being the similarity degrees.

In summary, 60 (5 parameter m * 2 shapes * 6 clusters) variants of Fuzzy Analogy technique were investigated in this study. These 60 solo techniques are evaluated through several accuracy measures and datasets. To the best of our knowledge this is the first study that investigates EEE techniques based on the solo Fuzzy Analogy technique.

4. Empirical design

This section presents the empirical design, starting with the performance measures used to evaluate the solo techniques and ensemble techniques of this study. The Scott-Knott test is explained next. Further in this section, the methodology pursued to build ensembles is detailed and the abbreviations of solo and ensemble techniques used are presented. The descriptions of the datasets used in this study are given at the end of the section.

4.1. Performance measures

The SDEE literature claims that the most frequent accuracy measures used to assess the performance of effort estimation techniques are the mean magnitude of relative error (MMRE) and the Pred (0.25) [1,33]. MMRE and Pred(0.25) are defined in Eqs. (15) and (16) respectively. These measures are derived from the magnitude of the relative error (MRE) as shown in Eq. (14). This MRE criterion has been criticized by some researchers for being biased toward under-estimates, which makes it not significant for being an accuracy measure [77–79]. In order to avoid the problem of MRE-based criteria, Miyazaki et al. [79] proposed two accuracy measures: Mean Balanced Relative Error (MBRE) and Mean Inverted Balanced Relative Error (MIBRE) which are considered less venerable to bias and asymmetry. Eqs. (18) and (19) present the formulae of MBRE and MIBRE respectively. In addition, the logarithmic standard deviation (LSD) is used in the SDEE literature as an accuracy criterion (Eq. (20)) [2,78,80].

Another accuracy measure, mean of absolute error (MAE) (Eq. (17)), does not present any of the problems mentioned above. It is computed by averaging the total of absolute errors (AE) (Eq. (13)). However, it is not easy to interpret it since the residuals are not normally distributed. Shepperd and MacDonell [53] suggest a new accuracy measure Standardized Accuracy (SA) based on MAE. SA evaluates whether a given estimation technique outperforms the baseline of a random guessing P_0 –see Eq. (21). The interpretation of SA is that the ratio represents how much better a prediction technique P_i is than random guessing (P_0). So, a high value means that P_i is much better than random guessing, a value near to zero is discouraging and a negative value would be worrisome [53].

Note that Shepperd and MacDonell [53] recommend using the 5% quantile of the random guessing to estimate the likelihood of non-random estimation. The interpretation of the 5% quantile for random guessing is similar to the use of α for conventional statistical inference, which means that any accuracy value that is better than this threshold has a less than one in twenty chance of having been randomly occurred.

To verify if the predictions of a model are generated by chance and if there is an improvement over random guessing the Effect Size criterion defined by Eq. (22) was used. The absolute values of Δ can be interpreted in terms of the categories proposed by Cohen [81]: small ($\simeq$0.2), medium ($\simeq$0.5) and large ($\simeq$0.8). A medium or large value of Δ indicates an acceptable degree of confidence on the model predictions over random guessing.

Hence, we adopt the accuracy measures (MAE, MIBRE, MBRE, LSD and SA) which are not biased toward under-estimates instead of MRE-based measures such as MMRE.

$$AE_i = |e_i - \hat{e}_i| \tag{13}$$

$$MRE = \frac{AE_i}{e_i} \tag{14}$$

$$MMRE = \frac{1}{n}\sum_{i=0}^{n} MRE_i \tag{15}$$

$$Pred\,(0.25) = \frac{100}{N}\sum_{i=1}^{N} \begin{cases} 1 if MRE_i \le 0.25 \\ 0\ otherwise \end{cases} \tag{16}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n} AE_i \tag{17}$$

$$MBRE = \frac{1}{N}\sum_{i=1}^{N} \frac{AR_i}{\min\left(e_i, \hat{e}_i\right)} \tag{18}$$

$$MIBRE = \frac{1}{N}\sum_{i=1}^{N} \frac{AR_i}{\max\left(e_i, \hat{e}_i\right)} \tag{19}$$

$$LSD = \sqrt{\frac{\sum_{i=1}^{n}\left(\lambda_i + \frac{s^2}{2}\right)^2}{n-1}} \tag{20}$$

$$SA = 1 - \frac{MAE_{p_i}}{\overline{MAE}_{p_0}} \tag{21}$$

$$\Delta = \frac{MAE_{p_i} - \overline{MAE}_{p_0}}{S_{p_0}} \tag{22}$$

Where:

- e_i and $\hat{e}_i$ are the actual and predicted effort for the ith project.
- $\overline{MAE}_{p0}$ is the mean value of a large number runs of random guessing. This is defined as, predict a $\hat{e}_i$ for the target project i by randomly sampling (with equal probability) over all the remaining n-1 cases and take $\hat{e}_i = e_r$ where r is drawn randomly from 1... $n \wedge r \neq i$. This randomization procedure is robust since it makes no assumption and requires no knowledge concerning a population.
- MAE_{p_i} mean of absolute errors for a prediction technique i.
- S_{p_0} is the sample standard deviation of the random guessing strategy.

Please cite this article in press as: A. Idri, et al., Improved estimation of software development effort using Classical and Fuzzy Analogy ensembles, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.08.012

- $\lambda_i = \ln(e_i) - \ln(\hat{e}_i)$
- s^2 is an estimator of the variance of the residual λ_i.

As for the evaluation, we adopt the jackknife validation. The jackknife, or "leave one out", is a cross-validation technique [82] in which the target project is excluded from the dataset and estimated by the remaining projects in the historical dataset

4.2. Statistical test: Scott-Knott – SK

The Scott-Knott – SK algorithm is an exploratory clustering algorithm used in the analysis of variance (ANOVA) context. It was proposed in 1974 by Scott and Knott [83] to find distinct non-overlapping (e.g. homogeneous) groups based on the multiple comparisons of treatment means. Since the proposition of the SK algorithm, many other hierarchical cluster analysis approaches have been proposed, such as: Jollife (1975) [84], Cox and Spjotvoll (1982) [85], and Calinski and Corsten (1985) [86]. However, the SK algorithm is still the most frequently used due to its simplicity and robustness [87–89].

The SK test requires that its inputs be normally distributed. However, Borges and Ferreira [90] evaluated the power and the type I error rates of the SK, against Tukey and Student-Newman-Keuls (SNK) tests, in a wide variety of experimental situations, with normality and non-normality error distributions. The results showed that: (1) SK outperformed Tukey and SNK tests, (2) SK is robust against violation of normality assumptions, and (3) In some cases, SK was eight times higher than Tukey test. In addition, the SK test (unlike Tukey and SNK tests) avoids ambiguous results in which two treatments are statistically different and are equal to another treatment. Furthermore, in the context of ensemble methods applied in large datasets, the SK, Tukey and Hsu tests were used to select the best cluster among different classifications models and the evaluation results have shown the superiority of SK [91]. Moreover, SK test was already used in SDEE to select solo techniques of EEE techniques [44,47]. It was also used by [92,93] to comparing, clustering and ranking multiple SDEE techniques.

Therefore, the SK test is used since: (1) it shows high performance compared to other statistical tests, and (2) its ability to select the best non-overlapping groups of solo techniques having similar predictive capabilities (e. g. select the solo techniques that have smaller MAE values with no significant difference between them).

4.3. Constructing ensembles

This section presents the methodology used to build ensembles based either on solo Classical Analogy or on solo of Fuzzy Analogy techniques. Note that the methodology used to evaluate solo techniques and construct ensembles is the same used by Azzeh et al. [44] due to the fact that: (1) According to Wen et al. [1], it is suitable to adopt a uniform methodology for evaluating the performance of different SDEE techniques in order to easily compare the results of different studies in SDEE; and (2) The experimental design of Azzeh et al. [44] used a set of contemporary and unbiased accuracy measures.

The methodology adopted to construct ensembles in this study involves 5-steps as shown in Fig. 3:

(1) Assess the absolute accuracy of each variant of Classical and Fuzzy Analogy in terms of SA and effect size in each dataset (Section 3 presents all these variants);
(2) select the solo techniques that have high accuracy values in terms of SA and Δ (e.g. SA values greater than the SA of the 5% quantile of random guessing and present at least a medium improvement in terms of effect size (e.g. $\Delta > 0.5$);

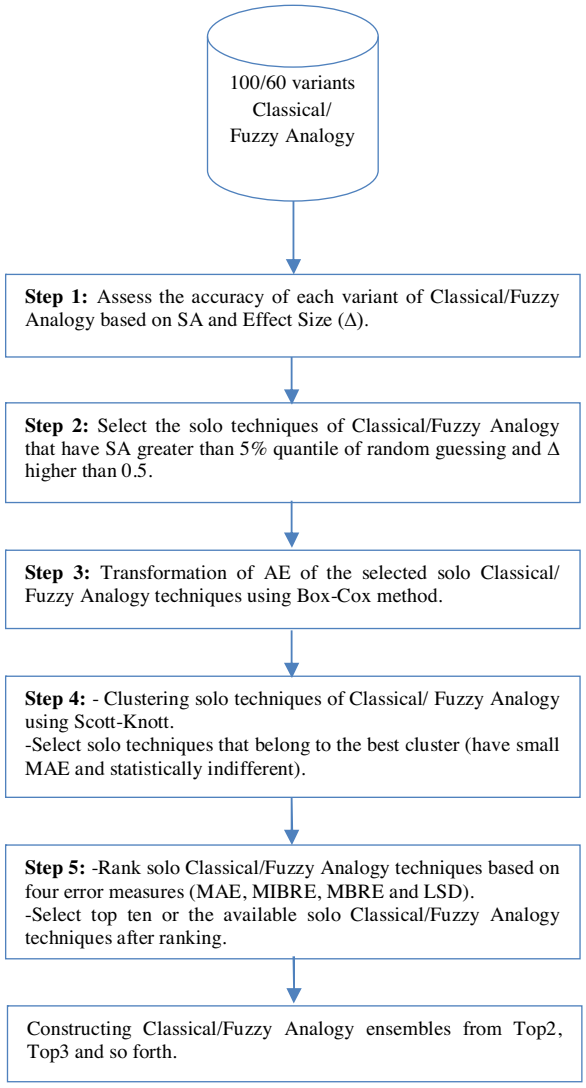


Fig. 3. Methodology steps to select solo techniques for Classical/Fuzzy ensembles construction.

(3) since the SK algorithm required its inputs to be approximately normally distributed, evaluate whether the distribution of residuals of selected solo techniques of Steps 1–2 are normally distributed or not. For that purpose the Kolmogorov-Smirnov statistical test was used [94]. Since the residuals are usually not normally distributed, the well-known transformation method Box-Cox is used [95];
(4) the SK test was performed on the transformed AE to cluster the selected solo techniques of Steps 1–2 into non-overlapping groups. The solo techniques that belong to the same group have similar predictive capability. Further, the solo techniques which will be selected are those of the best group (e.g. with the smallest MAE); and
(5) to ensure the accuracy of the selected solo techniques of Step 4, rank them according to four performance measures: MAE, LSD, MBRE and MIBRE, rather than relying on the rank obtained by SK which only rely on MAE. These measures, as explained in Section 4.1 are believed to be less vulnerable to bias and asymmetry. The final ranking obtained is computed using Borda count procedure which was used for the first time in SDEE by Azzeh et al. [44] for the same objective. Indeed, Myrtveit et al. [77] and Mittas et al. [92] deduced that the conclusion on which technique is best depends on the chosen performance indica-

Please cite this article in press as: A. Idri, et al., Improved estimation of software development effort using Classical and Fuzzy Analogy ensembles, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.08.012

Table 2
Abbreviations of distances and adaptation strategies of Classical Analogy.

Distance	Abbreviation	Adaptation strategy	Abbreviation
Euclidean	EU	Mean	ME
Manhattan	MA	Median	MD
Minkowski	MI	Inverse ranked weighted mean	IR
Chebyshev	CH	Inverse distance weighted mean	WD
Squared-chord	SC		

tors which may lead to contradictory results: each technique may have different ranking according to different performance measures. Hence, this study uses the Borda count procedure that allows us to draw a collective conclusion by taking into account four performance measures with equal weights.

After conducting the 5-steps, we construct the ensembles from Top2, Top3... Top *M* of selected solo techniques of Step 5, where *M* is the position rank of the *M* techniques. Note that this study only selects the top ten solo techniques of Step 5 (in case where the selected solo techniques of Step 5 are less than ten, we used the available ones), since the prior works conducted presume that the solo techniques of ensemble must be accurate [36,48]. As for the combination rules, we use three linear combiners: arithmetic mean, arithmetic median and inverse ranked weighted mean.

The Classical/Fuzzy ensembles are evaluated in terms of SA and effect size in order to check whether they produce meaningful predictions. Thereafter, the ensembles and the top ten solo techniques are clustered using the SK test in order to identify the best subgroup. The techniques (ensembles and solo techniques) that belong to the best cluster are ranked using the Borda count on four performance measures (LSD, MAE, MIBRE and MBRE) in order to identify which ones are the most accurate. Finally, both Classical Analogy ensembles and Fuzzy Analogy ensembles are evaluated in order to identify the best ones and SK tests were performed to check whether there is a statistical significance between them. All of these empirical evaluations were performed for each dataset.

4.4. Abbreviations of the proposed solo and ensembles methods

In order to shorten the names of proposed techniques (solo and ensembles techniques) and to assist the reader, we adopt the following naming rules in the rest of this paper.

- We abbreviate the name of each variant of Classical Analogy as follows:

Distance-number of analogues-adaptation strategy.

Table 2 lists the abbreviation of each distance as well as the adaptation strategy. The number of analogues (*k*) used here varies from 1 to 5. For example, EU4WD refers to the solo analogy technique that uses Euclidean distance (EU) as similarity measure, four closest analogues (4) and inverse distance weighted mean (WD) as adaptation strategy.

- For the Fuzzy Analogy variants, we abbreviate them as follows:

Shape-number of cluster-degree of similarity of closest projects-parameter m.

In this study, we have used two shapes: Triangular and Trapezoidal, abbreviated into TN and TP respectively. The number of clusters used varies from 2 to 7. The degree of similarity of closest projects used is all projects that are in the vicinity of 1. Concerning the parameter *m*, it was varied from 1.5 to 3.5 with increments of 0.5: then we refer to 1.5 by 1, 2.0 by 2 and so forth. For example, the solo technique TN713 means that the Fuzzy Analogy technique uses triangular (TN) as shape to build the membership functions and uses 7 clusters with parameter *m* fixed at 2.5 and uses all projects that are in vicinity to 1.

- For the ensembles, we abbreviate as follows:

Type of solo techniques-combination rule-number of solo techniques.

The type of solo techniques can be *C* to refer to the Classical Analogy technique and *F* to refer to the Fuzzy Analogy technique. For the combination rules: AVR, MED and IRW are used to indicate average, median and inverse ranked weighted mean respectively. The number of solo techniques of ensembles varies from 2 to 10. For example, FAVR8 ensemble means a fuzzy ensemble (F) which is constructed by the combination of eight (8) top techniques and using average (AVR) as combiner.

4.5. Datasets

Since the characteristics of datasets influence the performance of effort estimation techniques, seven datasets were selected in order to evaluate the performances of solo and ensembles techniques. These datasets contain 1063 historical projects in total. They came from two repositories:

- The PRedictOr Models In Software Engineering (PROMISE) data repository which is a publicly available online data repository [96]. The six datasets selected from this repository are: Albrecht [8], COCOMO81 [6], China dataset [96], Desharnais [97], Kemerer [98], and Miyazaki [99].
- The International Software Benchmarking Standards Group data repository (ISBSG) is a non-profit organization which maintains the ISBSG data repository of projects contributed by organizations across the world [100] that have been measured using a recognized functional size measurement method. ISBSG repository release 8 used in this study contains more than 2000 software projects that are described by more than 50 numerical and categorical attributes.

For the PROMISE datasets, all features were used in the experiments. Tables A1–A6 of Appendix A list the selected cost drivers of the six PROMISE datasets respectively. However, for ISBSG R8 dataset, we conduct a data preprocessing phase to select data (projects and attributes) in order to only retain data with high quality [27,30,31,71]. This phase consists of two steps:

Step 1: Project selection:

The purpose of this step is to select historical projects with high quality data. Therefore, we used the following four criteria as in [27,30,31,71]:

1 Data Quality Rating: This field contains an ISBSG rating code of A, B, C, or D applied to the project data by the ISBSG quality reviewers. This code denotes the soundness and integrity of the data of each project: "A = The data submitted was assessed as being sound with nothing being identified that might affect its integrity. B = The submission appears fundamentally sound but there are some factors which could affect the integrity of the submitted data. C = Due to significant data not being provided, it was not possible to assess the integrity of the submitted data. D = Due to one factor or a combination of factors, little credibility should be given to the submitted data."

The projects selected for this study are those with a rating code A or B.

Please cite this article in press as: A. Idri, et al., Improved estimation of software development effort using Classical and Fuzzy Analogy ensembles, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.08.012

Table 3
Data Quality criteria for project selection.

Criteria	Selected values	Discarded Values
Data Quality Rating	A or B	C and D
Resource Levels	1 or 2	3 and 4
Counting Approach	IFPUG	NESMA, COSMIC-FFP, etc.
Development Type	New Development	Enhancement and Redevelopment

2 Resource levels: These levels indicate the type of data collected about the people whose time is included in the work effort data reported. The data collection instrument identifies four levels of resources: "1 = development team effort (e.g. project team, project management, project administration). 2 = development team support (e.g. database administration, data administration, quality assurance, data security, standards support, audit & control, technical support). 3 = computer operations involvement (e.g. software support, hardware support, information center support, computer operators, network administration). 4 = end users or clients e.g. user liaisons, user training time, application users and/or clients."

The projects selected are those with resource levels 1 or 2. Indeed, in software effort estimation literature, the development effort includes only the effort expended on the activities of the development team and its support.

3 Counting approach: This field indicates the technique used to count the function points, such as IFPUG, NESMA, or COSMIC-FFP. The projects selected are those that have used the IFPUG approach, since some of the effort drivers in Table A7 of Appendix A, such as Input Count, Output Count, and File Count, are only relevant for the IFPUG technique.

4 Development type: This field indicates whether the included project is a new development, an enhancement, or a redevelopment project. The projects included here are those with new development type since the paper deals with software development effort.

Table 3 summarizes the quality criteria that have to be met for project selection. A total of 148 projects that satisfied all the criteria were selected for this study.

Step 2: Attribute selection

This step aims to identify the attributes to be used as the inputs (effort drivers) to either Classical or Fuzzy Analogy for software effort estimation. Selecting the optimal subset of features remains a complex task in the analogy procedure and has a large impact on the similarity measure. Therefore, several studies have been conducted on attribute selection in software effort estimation using different techniques such as fuzzy logic [57], genetic algorithms [101,102], and statistical methods [103]. In the case of the ISBSG dataset, in which the projects are described with more than 50 numerical and linguistic attributes, the solution adopted was to allow the estimators to use the attributes that meet the following criteria: the estimators believe that they are relevant for effort estimation, and they are the most appropriate in their environment. Since the aim of this study is to deal only with numerical attributes, ten numerical attributes were selected from the ISBSG dataset, as they are usually considered in the literature as relevant effort drivers [31,71]. Table A7 of Appendix A lists the ten selected attributes of the ISBSG dataset.

Table 4 gives the description of the seven selected datasets, including the number of attributes, the number of historical projects, the unit of effort, and the minimum, maximum, mean, median, skewness and kurtosis of efforts. Note that the effort is measured by Man-hours for the China, Desharnais and ISBSG datasets while it is measured by Man-months for the remaining four datasets. These datasets are relevant for evaluating the solo and ensembles techniques, since they are diverse in terms of their fields, their sources and their sizes.

From the statistics of Table 4, we can conclude that the effort values of all datasets are not normally distributed, since their skewness coefficient are different to zero and none of their kurtosis coefficient are equal to three: this presents a challenge for the software community to develop accurate predictive techniques [44,104]. Note that the original COCOMO81 contains 63 projects and that the one used in this study is composed of 252 projects. In fact, in the original COCOMO81 dataset, each cost driver is measured using a rating scale of six linguistic values (very low, low, nominal, high, very high and extra-high). For each couple of project and linguistic value, four numerical values have been randomly generated according to the classical interval used to represent the linguistic value (63*4 = 252, see [105] for details on how we have obtained 252 from 63 projects).

5. Results and discussions

This section presents and discusses the empirical results of the evaluations conducted in this study. Firstly, all solo techniques (Classical and Fuzzy Analogy variants) were evaluated and sorted for each dataset. Secondly, ensembles were constructed and evaluated. Finally, Fuzzy Analogy ensembles were compared with Classical Analogy ensembles.

Different tools were used in order to perform the empirical evaluations of this study. Concerning the estimation by Classical Analogy, we developed a software prototype using C#.NET under a Microsoft environment. As for the Fuzzy Analogy, we developed a software prototype with Matlab 7.0 under a Microsoft environment. The Box-cox transformation and the statistical tests (Kolmogorov-Smirnov and SK tests) were performed through R Software [106].

5.1. Evaluation of solo techniques

This step consists of evaluating all solo techniques investigated in this study over the seven datasets. At this stage, all techniques that fail to show a predictive capability better than random guessing (e.g. high SA) and a medium improvement in terms of effect size (e.g. $\Delta > 0.5$) are discarded. In other words, we examine if the obtained MAE of each solo technique on each dataset is less than the 5% quantile of random guessing: hence, we have a confidence that the selected techniques are actually predicting and not guessing. To confirm that the selected solo techniques are predicting, we evaluate also the effect size: the Δ value should be higher than 0.5, which means a medium effect improvement over guessing. The evaluation results of solo Classical and Fuzzy Analogy techniques are presented and discussed separately in the following subsections.

5.1.1. Solo Classical Analogy techniques

Table B1 of Appendix B summarizes the evaluation of the accuracy of all solo Classical Analogy variants in terms of SA and effect size over the seven datasets where the baseline method is random guessing. Moreover, $SA_{5\%}$ presented in the second row presents the SA of 5% quantile of random guessing.

We notice that:

- All solo Classical Analogy techniques evaluated on ISBSG dataset are better than random guessing.
- For the Albrecht dataset, except for MI distance, solo Classical Analogy techniques provide meaningful predictions (i.e. better than 5% quantile of random guessing).
- For the China dataset, when MI distance is used with one or two analogues, solo Classical Analogy techniques produce worse pre-

Please cite this article in press as: A. Idri, et al., Improved estimation of software development effort using Classical and Fuzzy Analogy ensembles, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.08.012

Table 4
Descriptive statistics of the 7 selected datasets.

Dataset	Size	Unit	#Features	Effort					
				Min	Max	Mean	Median	Skewness	Kurtosis
Albrecht	24	Man/Months	7	0.5	105	21.87	11	2.30	4.7
COCOMO81	252	Man/Months	13	6	11400	683.44	98	4.39	20.5
China	499	Man/Hours	18	26	54620	3921.04	1829	3.92	19.3
Desharnais	77	Man/Hours	12	546	23940	4833.90	3542	2.03	5.3
ISBSG	148	Man/Hours	10	24	60270	6242.60	2461	3.05	11.3
Kemerer	15	Man/Months	7	23	1107	219.24	130	3.07	10.6
Miyazaki	48	Man/Months	8	5.6	1586	87.47	38	6.26	41.3

Table 5
Best solo Classical Analogy of each dataset in terms of SA and Effect Size **Δ**.

Datasets	Classical Analogy variants	SA (%)	Δ
Albrecht	MA1IR, MA1MD, MA1ME, MA1WD	74.14	−3.78
China	SC4ME	72.73	−14.016
COCOMO81	MA1IR, MA1MD, MA1ME, MA1WD	96.16	−9.789
Desharnais	EU5MD	45.60	−4.854
ISBSG	SC3ME	60.75	−7.173
Kemerer	EU5WD	50.23	−1.969
Miyazaki	MA4WD	54.33	−1.959

Table 6
Best solo Fuzzy Analogy of each dataset in terms of SA and Effect Size **Δ**.

Dataset	Fuzzy Analogy variants	SA (%)	Δ
Albrecht	TP312	84.10	−4.287
China	TP612	70.93	−13.67
COCOMO81	TP713	98.36	−10.01
Desharnais	TP214, TN214	46.04	−4.901
ISBSG	TP312	62.42	−7.371
Kemerer	TN515	50.75	−1.99
Miyazaki	TP515	58.37	−2.104

dictions than random guessing. The same results were obtained by MI3IR, MI4IR and MI5IR techniques.

- For the COCOMO81 dataset, solo Classical Analogy techniques using MI distance generate worse predictions than random guessing, except for MI5MD.
- For the Desharnais dataset, solo Classical Analogy techniques that use CH, EU, MA, and SC distances with k ranged from 3 to 5 analogues under IR adaptation strategy gave worse results than random guessing. The same results were obtained with MI2IR and MI4IR techniques.
- For the Kemerer dataset, solo Classical Analogy techniques using IR adaptation strategy give worse predictions than random guessing for all distances (except for EU and MA when $k = 1$ or 2 analogues). Solo Classical Analogy techniques with CH distance give worse predictions than random guessing when using MD adaptation with one analogue, ME adaptation with 1 or 4 analogues and WD adaptation with 1, 4 or 5 analogues. As for solo Classical Analogy techniques using SC distance with one analogue also give worse results than random guessing. The same results are obtained for Classical Analogy with SC distance using two analogues except for SC2WD. In addition, all techniques using MI distance give worse results.
- For Miyazaki dataset, solo Classical Analogy techniques that use MI distance with 2 analogues generate worse predictions than random guessing. Further, the same results are given by IR adaptation strategy (except when using one analogue). However, the remaining techniques fall beyond the 5% quantile of random guessing for all datasets.

In terms of effect improvement, the effect size test shows large effect size over all datasets, which implies large effect improvement over guessing.

Hence, 20, 11, 19, 14, 48 and 7 solo Classical Analogy techniques were excluded for Albrecht, China, COCOMO81, Desharnais, Kemerer and Miyazaki datasets respectively since they fail to show an improvement with respect to random guessing.

Table 5 lists the best prediction variants of Classical Analogy for each dataset in terms of SA. As it can be observed from Table 5, the prediction accuracy in terms of SA of each solo Classical Analogy varies from one dataset to another. Thus, each dataset favors a specific variant of Classical Analogy technique.

Only two datasets (Albrecht and COCOMO81) favor the same Classical Analogy variants, which are based on MA similarity measure, one analogue and four adaptation strategies. The Miyazaki dataset advantages the Classical Analogy variant based on MA similarity measure, 4 analogues and WD as adaptation strategy. In addition, the two China and ISBSG datasets prefer solo Classical Analogy variant that uses SC distance as similarity measure and median as adaptation strategy. However, they differ in number of analogues ($k = 4$ for China, $k = 3$ for ISBSG). Finally, Desharnais and Kemerer datasets favor the same similarity measure (EU distance) as well as the same k ($k = 5$), and differ in adaptation strategy (MD for Desharnais, WD for Kemerer).

5.1.2. Solo Fuzzy Analogy techniques

Table B2 of Appendix B summarizes the evaluation of Fuzzy Analogy variants in terms of SA and effect size over the seven datasets where the baseline method is random guessing. Moreover, $SA_{5\%}$ in the second row presents the SA of 5% quantile of random guessing.

Table B2 of Appendix B shows that for ISBSG, COCOMO81, Albrecht, China, Miyazaki and Desharnais, Fuzzy Analogy generate better estimates than random guessing whatever the shape, number of clusters and parameter of fuzziness m. However, for Kemerer dataset, TN411, TN511, TN513, TN615, TN713, TN715, TP312, TP514, TP612 and TP715 solo techniques give worse results than random guessing (their SA ranged from 16.87% to 32.9%). As for the effect size, the values of Δ belong to the large category, which implies a large effect improvement over guessing. Hence, 10 solo Fuzzy Analogy techniques for Kemerer dataset were excluded, since they failed to show an improvement with respect to random guessing. For the rest of datasets, all solo Fuzzy Analogy variants were selected.

Table 6 presents the best solo Fuzzy Analogy techniques for each dataset. As shown, all the datasets favor only one solo technique except for the Desharnais dataset which prefers two solo Fuzzy Analogy technique (TP214 and TN214). Moreover, we observe that all datasets favor only the variants which used a trapezoidal shape to represent the membership functions, except for the Desharnais dataset which prefers both shapes (Trapezoidal and Triangular) and Kemerer dataset which prefers the Triangular shape. This confirms the results of our previous studies [61,71]. Furthermore, Albrecht and ISBSG favor the same solo technique (TP312), whereas the rest of the datasets prefer different variants of Fuzzy Analogy (e.g. different number of clusters or different values of parameter m).

Please cite this article in press as: A. Idri, et al., Improved estimation of software development effort using Classical and Fuzzy Analogy ensembles, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.08.012

5.2. Ranking of solo Classical and Fuzzy Analogy techniques

The next step in our methodology consists of testing whether the absolute errors of the selected solo techniques are normally distributed or not. As mentioned in Section 4.3, this step is performed since the SK clustering analysis requires that its inputs should be approximately normally distributed. Unfortunately, the errors of the selected solo techniques do not verify the assumption of normality. The test of normality is performed by the Kolmogorov-Smirnov test. Therefore, the absolute errors of all selected solo techniques of each dataset were transformed to be approximately normally distributed. This transformation is conducted using the Box-Cox transformation. Thereafter, we performed the SK test in order to cluster the solo selected techniques into non-overlapping groups and identified the best group based on MAE. The solo techniques that belong to the same group have similar predictive capability. The best group contains the solo techniques that have the smallest variation of MAE.

However, the ranking obtained by SK is not sufficient since the solo techniques of the best group are statistically similar, and this ranking is only based on MAE. That is why we used in addition the Borda count procedure to rank the variants across four error measures (MAE, LSD, MBRE and MIBRE) with equal weight for each dataset. Note that all solo techniques are ranked in ascending order over all error measures, where lower accuracy means better. The top 10 techniques according to the Borda count are presented in details in the following subsections.

5.2.1. Ranking of solo Classical Analogy techniques

As shown in Fig. 4, 10 clusters were identified in COCOMO81 dataset, 9 in China dataset, 4 in ISBSG dataset, and 2 in Desharnais, Albrecht and Miyazaki datasets: this implies that these datasets favor certain Classical Analogy variants. This is not the case of Kemerer dataset since only one cluster was identified. Further, the number and the configuration of solo techniques that belong to the best clusters vary from one dataset to another. This means that the performances of solo Classical Analogy techniques are highly affected by the dataset characteristics (e.g. size of projects and features), and shows that no solo technique has the same rank across all datasets.

For China dataset, the best subgroup contained variants of Classical Analogy with only SC and MA distances and did not contain variants with IR adaptation strategy or one analogue. As for COCOMO81 dataset, the best subgroup included only variant with $k = 1$ or $k = 3$. Moreover, it did not contain variants of Classical Analogy with MI or CH distances. The best subgroups of ISBSG and Miyazaki did not contain any variant of Classical Analogy using MI distance. We notice that the best subgroups of Desharnais and Albrecht contained a large number of solo Classical Analogy techniques while the best subgroup of Kemerer included all the variants of Classical Analogy selected in the first step.

From the analysis above, we can conclude that:

(1) There is no consistency in the number of solo Classical Analogy techniques that belong to the best group of each dataset.
(2) The performances of Classical Analogy techniques are influenced by datasets characteristics which lead to different clusters when applying SK algorithm.
(3) There is no evidence on the best number of analogues (k) for each dataset.
(4) Variants of Classical Analogy based on MI distance did not give good results since it did not appear in the best subgroup of the datasets, except for Desharnais.
(5) No variant of Classical Analogy outperformed the others distinctly across all datasets, which prove that solo Classical Analogy techniques behave differently according to each dataset.

Table 7 lists the top ten variants of Classical Analogy techniques based on the rate provided by Borda count using four performance measures. Table 7 shows that no variant of Classical Analogy was ranked first in all datasets. Also, variants based on MI and CH distances do not appear in the top ten of any dataset. Besides, Classical Analogy variants using SC distances are the most frequent in the top ten of the datasets: they appeared in 6 out of 7 datasets and were the only Classical Analogy variants present in the top ten of ISBSG and China datasets. Classical Analogy variants using IRW as adaptation strategy appear in the top ten of only three datasets (Albrecht, COCOMO81 and ISBSG). Regarding the number of analogues, we notice that only the top ten of ISBSG dataset contained all five values of k. Moreover, only $k = 3$ was present in 6 out of 7 datasets.

The principal findings are:

(1) No solo Classical Analogy technique proved to be the best across all datasets.
(2) There is strong evidence about the performance of SC distance; since it was the most frequent distance in the top ten solo Classical Analogy techniques of all datasets.
(3) There is no evidence about the best number of analogues k.
(4) The size of the dataset does not affect the number of analogues.

5.2.2. Ranking of solo Fuzzy Analogy techniques

As shown in Fig. 5, 10 clusters were identified in COCOMO81 dataset, 4 in China dataset, and 3 in ISBSG dataset, which means that these datasets prefer some variants of Fuzzy Analogy. However, the remaining datasets (Desharnais, Kemerer, Albrecht and Miyazaki datasets) did not favor any set of solo Fuzzy Analogy techniques since all variants show statistically similar predictive capabilities (only one cluster was identified by SK). Furthermore, as in the case of Classical Analogy, the number and the configuration of variant techniques that belong to the best clusters change from one dataset to another. This means that the performances of solo Classical Analogy techniques are highly affected by the dataset characteristics (e.g. size of projects and features), and shows that no technique has the same rank across all datasets.

The best subgroup in ISBSG dataset contained variants of Fuzzy Analogy with 4, 5, 6 and 7 clusters while the best subgroups in COCOMO81 and China datasets included variants of Fuzzy Analogy with 5, 6 and 7 clusters. We notice that these datasets have a large number of historical projects and high number of features (See Table 4). Note that both shapes (TN and TP) belong to the best subgroups of all datasets.

From the observations above we can conclude that:

(1) There is no consistency in the number of techniques that belong to the best group for each dataset.
(2) The performances of solo Fuzzy Analogy techniques are influenced by datasets characteristics which lead to different clusters when applying the SK algorithm.
(3) Regarding the number of clusters, datasets with large number of projects and features favor a large number of clusters.
(4) Solo Fuzzy Analogy techniques behave differently according to each dataset, since none of them outperformed the others across all datasets.

Table 8 lists the top ten variants of Fuzzy Analogy techniques based on the rates provided by Borda count using four performance measures. In addition, as can be seen from Table 8, Fuzzy Analogy variants based on TP shape were ranked first in all datasets

Please cite this article in press as: A. Idri, et al., Improved estimation of software development effort using Classical and Fuzzy Analogy ensembles, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.08.012

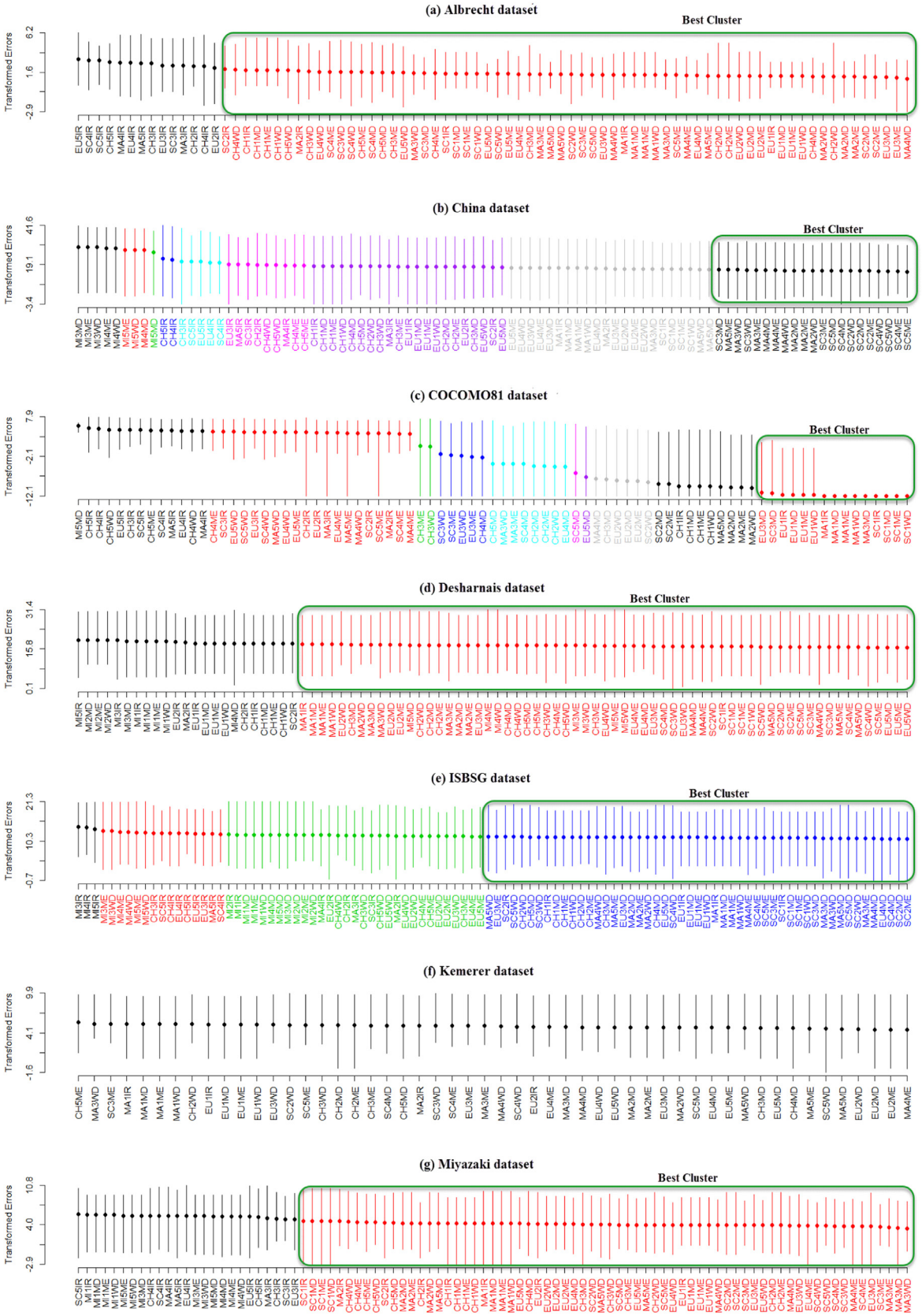


Fig. 4. SK test of solo Classical Analogy techniques on each dataset. The x-axis represents the selected techniques stored where the better positions start from the right side. The y-axis represents the transformed AEs, each vertical line shows the variation of transformed AEs for each technique, and the small circle represents the mean of transformed AEs. Green box means best cluster. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.).

Please cite this article in press as: A. Idri, et al., Improved estimation of software development effort using Classical and Fuzzy Analogy ensembles, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.08.012

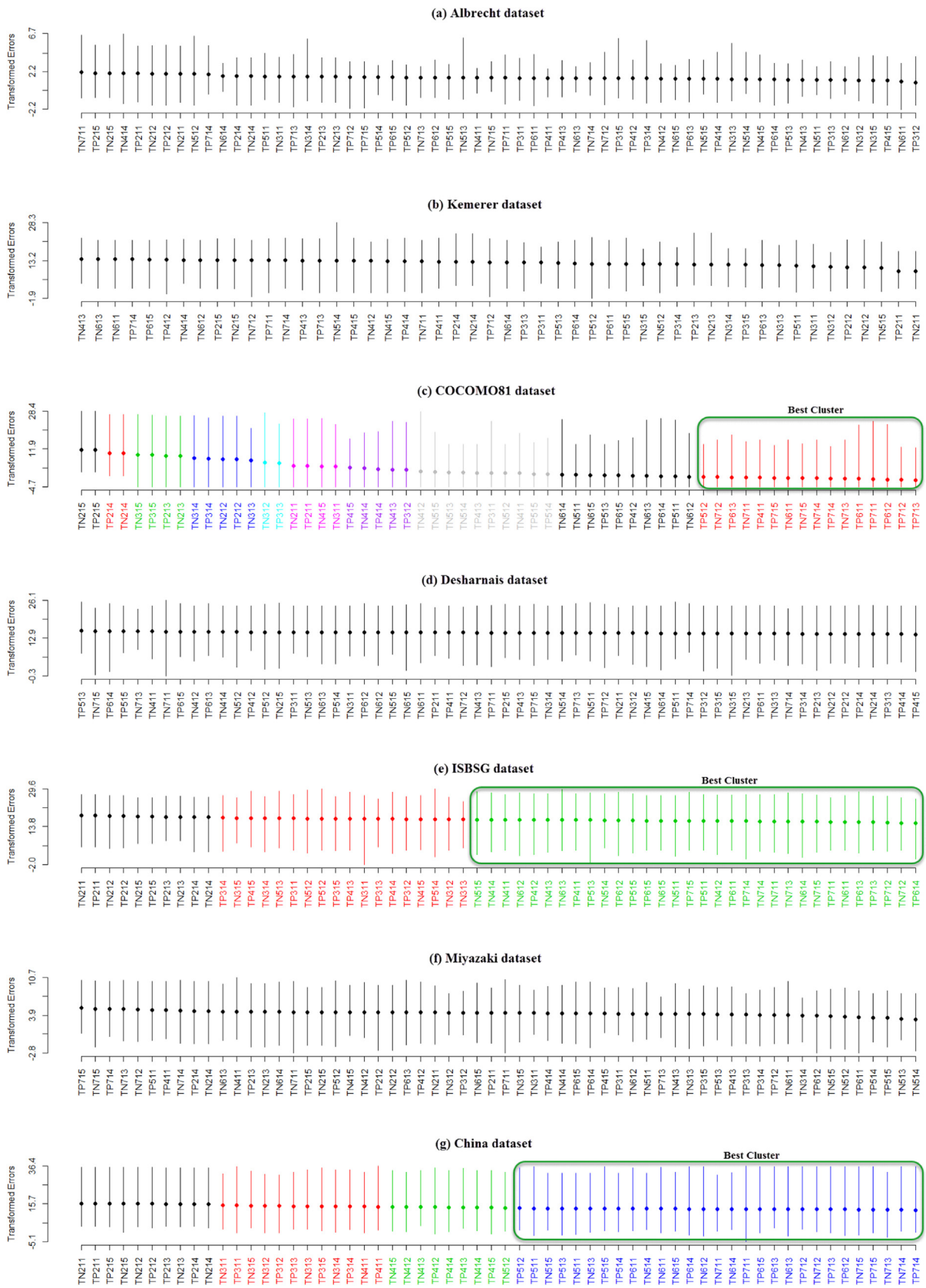


Fig. 5. Plot of SK test of solo Fuzzy Analogy techniques in each dataset. The x-axis represents the selected techniques stored where the better positions start from the right hand side. The y-axis represents the transformed AEs, each vertical line shows the variation of transformed AEs for each technique, and the small circle represents the mean of transformed AEs. Green box means best cluster. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Please cite this article in press as: A. Idri, et al., Improved estimation of software development effort using Classical and Fuzzy Analogy ensembles, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.08.012

Table 7
Top ten solo Classical Analogy techniques based on Borda count.

Rank	Dataset						
	Albrecht	China	COCOMO81	Desharnais	ISBSG	Kemerer	Miyazaki
1	SC2ME	SC5ME	MA1WD	EU5WD	SC2MD	MA5ME	MA4WD
2	SC2MD	SC4ME	MA1ME	SC3MD	SC2ME	EU5ME	MA4ME
3	SC2IR	SC3ME	MA1MD	EU5ME	SC3ME	EU4MD	SC4ME
4	EU3MD	SC5WD	MA1IR	MA5WD	SC4ME	EU4ME	MA3WD
5	EU2ME	SC4WD	MA3MD	SC3ME	SC4MD	EU4WD	SC4WD
6	EU2MD	SC2ME	SC3MD	SC4ME	SC5ME	EU5WD	SC5WD
7	EU3ME	SC2MD	SC1WD	MA5ME	SC2WD	MA5WD	SC5ME
8	MA2ME	SC4MD	SC1ME	SC5ME	SC2IR	MA4ME	MA3ME
9	MA2MD	SC3WD	SC1MD	EU5MD	SC3WD	MA5MD	MA5ME
10	SC4ME	SC2WD	SC1IR	MA4WD	SC1IR	EU2WD	EU4ME

except for Kemerer dataset. Also, for COCOMO81, China and ISBSG datasets, we notice that the top ten Fuzzy Analogy variants are based on a large number of clusters (6 and 7). For the m parameter, we observe that $m = 1.5$ did not appear in the top ten Fuzzy Analogy variants of three datasets (ISBSG, China and Miyazaki).

The main findings are:

(1) No solo Fuzzy Analogy variant proved to be the best across all datasets.
(2) There is no evidence about the outperformance of solo Fuzzy Analogy techniques based on TP shape over TN shape, since the top ten techniques are composed by the both type of shapes.
(3) There is no consensus about the best value of m, since every dataset favor different values of m.This step consists in building and evaluating Classical Analogy ensembles and Fuzzy Analogy ensembles for each dataset. In fact, we construct ensembles from Top2, Top3... Top10 based on the top ten solo techniques of Tables 7 and 8. Thus, these solo techniques are more accurate than the others; moreover, the vertical lines of the best clusters identified by SK (see Figs. 4 and 5) show that the best solo techniques have different variations in terms of AE which means that they are diverse. For example, in Fig. 4-a, the MA4MD solo technique (first vertical line from right) have a transformed MAE around 0.89 and the transformed AE ranged from −2.92 to 2.99, while EU3MD (third vertical line from right) technique have a transformed MAE around 0.89 and the transformed AE ranged from −0.33 to 2.78. Hence, these solo techniques are satisfying the two criteria (e.g. high accuracy and diversity) explained in Section 2.2. The conventional names of ensembles are presented in Section 4.5.

5.3. Evaluation of ensemble techniques

At first, we constructed ensembles and evaluated them with respect to (1) random guessing baseline, in order to assess whether they are predicting or guessing, and (2) the effect size to confirm if there is an improvement or not. Second, we examined the top ten solo techniques and the 27 ensembles for each dataset by clustering them using SK and storing the ones that belong to the best cluster by means of Borda count based on 4 performance measures. The purpose of this second stage is to assess if ensembles techniques are more accurate than solo techniques.

The next two subsections present the results obtained for Classical and Fuzzy Analogy ensembles.

5.3.1. Classical Analogy ensembles

Table 9 reports the SA values and effect size of all constructed Classical Analogy ensembles over the seven datasets where the baseline method is random guessing. Table 9 shows that all Classical Analogy ensembles fall beyond the 5% quantile of random guessing over all datasets, which is not the case of solo Classical Analogy techniques. In summary, these ensembles perform better than random guessing since they are constructed using only the solo techniques that performed better than random guessing too. Further, all ensembles show a large improvement over guessing, since the values of effect size are larger than 0.8 (e.g. $\Delta > 0.8$).

Fig. 6 shows the results of the SK test performed to assess whether there is a significant difference between ensembles and solo techniques based on MAE. As can be seen from Fig. 6, expect for COCOMO81 dataset, all the datasets do not prefer any specific technique. Since, the test identified only one cluster, this means that there is no significant difference between techniques. Meanwhile, the SK test identified two clusters in COCOMO81 dataset (Fig. 6-c); the best one contains 27 techniques (10 solo techniques, 17 ensembles), we notice that all the ensembles that use IRW as combiner belong to the best cluster.

To conclude on which types of techniques are more accurate, we sorted the ones that belong to the best cluster for each dataset using Borda count. Table 10 presents the raking of Classical Analogy ensembles and the top ten solo Classical Analogy techniques (the SK test has identified only one cluster for all dataset expect for COCOMO81 dataset). As can be seen from Table 10, except for COCOMO81 and Miyazaki datasets, solo techniques are ranked far behind ensemble techniques. For instance, in COCOMO81 dataset the first solo technique appeared at the 14th position (MA1IR solo technique). As for Miyazaki dataset, the solo technique MA4WD was ranked at the third position. However, for the remaining datasets the top 18 techniques are all ensembles techniques.

However, some solo techniques outperformed some ensemble techniques. This can be due to the used combination rule or the lack of diversity of solo techniques:

(1) Some solo technique combinations prefer a certain rule as in Desharnais dataset, the ensemble CIRW3 (e.g. ensemble based on 3 variants of Classical Analogy and IRW as combiner) was ranked at the 7th position whereas the CMED3 (e.g. ensemble based on 3 variants of Classical Analogy and median as combiner) was the worst one in this dataset since it was ranked 28th (last ensemble in this dataset); the same comment may be made on the Albrecht dataset, the first ensemble is CAVR4 while CMED4 is at the 10th position.

(2) The lack of diversity of solo techniques may occur since these solo techniques have small variations of AE (see Fig. 4).

Concerning the ensembles, as can be seen from Table 10, there is no best ensemble across all datasets. For example, the ensemble ranked first in Albrecht dataset (CAVR4), was ranked 26th in China, 6th in COCOMO81, 10th in Desharnais, 13th in ISBSG, 2nd in Kemerer and 11th in Miyazaki datasets: this means that the ensembles based on Classical Analogy variants are not stable across all datasets. In fact, the same ensemble may behave differently from one dataset to another. Thus, each dataset favors a specific ensem-

Please cite this article in press as: A. Idri, et al., Improved estimation of software development effort using Classical and Fuzzy Analogy ensembles, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.08.012

Table 8
Top ten solo Fuzzy Analogy techniques based on Borda count.

Rank	Dataset						
	Albrecht	China	COCOMO81	Desharnais	ISBSG	Kemerer	Miyazaki
1	TP312	TP715	TP713	TP213	TP614	TN515	TP515
2	TN413	TP714	TP712	TN214	TN714	TN314	TP514
3	TP413	TN713	TN713	TP214	TN712	TP315	TN513
4	TN312	TN714	TN714	TN213	TP613	TN315	TN514
5	TN411	TN715	TP714	TN212	TN615	TP314	TN515
6	TP313	TN614	TN611	TP212	TP615	TN313	TP513
7	TN514	TP612	TP612	TP315	TN715	TN211	TN615
8	TP415	TP613	TN711	TP711	TN613	TN311	TP415
9	TP714	TN613	TP715	TN313	TN612	TP211	TP512
10	TP414	TN612	TN712	TP611	TP714	TP311	TN614

Table 9
SA and effect size of Classical Analogy ensembles.

Dataset	Albrecht		China		COCOMO81		Desharnais		ISBSG		Kemerer		Miyazaki	
$SA_{5\%}$	0.2931		0.082		0.158		0.1554		0.1304		0.331		0.3357	
Ensemble	SA	Δ	SA	Δ	SA	Δ	SA	Δ	SA	Δ	SA	Δ	SA	Δ
CAVR2	0.7402	−3.773	0.7255	−13.98	0.9616	−9.789	0.467	−4.971	0.6026	−7.115	0.4979	−1.952	0.5428	−1.957
CAVR3	0.7652	−3.901	0.7288	−14.044	0.9616	−9.789	0.4648	−4.947	0.6117	−7.222	0.4964	−1.946	0.5406	−1.949
CAVR4	0.757	−3.858	0.725	−13.971	0.9616	−9.789	0.4632	−4.93	0.6147	−7.257	0.5056	−1.982	0.5406	−1.949
CAVR5	0.7524	−3.835	0.7286	−14.039	0.9611	−9.783	0.459	−4.885	0.6126	−7.232	0.5023	−1.969	0.5402	−1.947
CAVR6	0.7488	−3.817	0.7289	−14.045	0.943	−9.6	0.4561	−4.855	0.6153	−7.265	0.5028	−1.971	0.5372	−1.936
CAVR7	0.7434	−3.79	0.7284	−14.036	0.9272	−9.439	0.4521	−4.812	0.6111	−7.216	0.502	−1.968	0.5349	−1.928
CAVR8	0.743	−3.787	0.7273	−14.014	0.9139	−9.303	0.4519	−4.81	0.6096	−7.197	0.4997	−1.959	0.5351	−1.929
CAVR9	0.7423	−3.784	0.7268	−14.006	0.9033	−9.196	0.4553	−4.846	0.6072	−7.169	0.5027	−1.971	0.5343	−1.926
CAVR10	0.7373	−3.758	0.7262	−13.993	0.8949	−9.11	0.4533	−4.824	0.6179	−7.295	0.511	−2.004	0.5346	−1.927
CMED2	0.7402	−3.773	0.7255	−13.98	0.9616	−9.789	0.467	−4.971	0.6026	−7.115	0.4979	−1.952	0.5428	−1.957
CMED3	0.7402	−3.773	0.7282	−14.031	0.9616	−9.789	0.4466	−4.753	0.6026	−7.115	0.4982	−1.953	0.5421	−1.954
CMED4	0.7441	−3.793	0.7249	−13.968	0.9616	−9.789	0.4629	−4.927	0.6154	−7.266	0.502	−1.968	0.5392	−1.944
CMED5	0.7451	−3.798	0.7284	−14.035	0.9616	−9.789	0.4577	−4.872	0.6133	−7.241	0.5102	−2	0.5346	−1.927
CMED6	0.7426	−3.785	0.7309	−14.084	0.9616	−9.789	0.4584	−4.879	0.6127	−7.234	0.5032	−1.973	0.5369	−1.935
CMED7	0.7455	−3.8	0.7283	−14.034	0.9616	−9.789	0.4565	−4.858	0.6135	−7.243	0.4948	−1.94	0.538	−1.939
CMED8	0.7412	−3.778	0.726	−13.988	0.962	−9.793	0.4511	−4.801	0.5998	−7.081	0.4953	−1.942	0.5365	−1.934
CMED9	0.735	−3.746	0.7254	−13.979	0.9621	−9.794	0.4553	−4.846	0.6157	−7.27	0.4999	−1.96	0.5345	−1.927
CMED10	0.7421	−3.783	0.726	−13.989	0.9361	−9.529	0.4535	−4.826	0.6056	−7.15	0.5002	−1.961	0.5335	−1.923
CIRW2	0.7402	−3.773	0.7254	−13.978	0.9616	−9.789	0.4671	−4.971	0.6026	−7.115	0.4979	−1.952	0.5428	−1.957
CIRW3	0.7649	−3.899	0.7288	−14.043	0.9616	−9.789	0.4648	−4.947	0.6116	−7.221	0.4965	−1.946	0.5407	−1.949
CIRW4	0.7569	−3.858	0.7252	−13.974	0.9616	−9.789	0.4633	−4.931	0.6146	−7.257	0.5053	−1.981	0.5406	−1.949
CIRW5	0.7525	−3.836	0.7285	−14.037	0.9612	−9.784	0.4593	−4.888	0.6128	−7.235	0.5022	−1.969	0.5403	−1.948
CIRW6	0.749	−3.818	0.7292	−14.052	0.9467	−9.637	0.4565	−4.858	0.6154	−7.266	0.5028	−1.971	0.5374	−1.937
CIRW7	0.7439	−3.792	0.7285	−14.037	0.9346	−9.514	0.4527	−4.818	0.6114	−7.219	0.502	−1.968	0.5352	−1.929
CIRW8	0.7435	−3.79	0.7275	−14.019	0.9247	−9.413	0.4525	−4.816	0.6099	−7.202	0.4998	−1.96	0.5353	−1.93
CIRW9	0.7428	−3.786	0.7272	−14.013	0.9173	−9.338	0.4555	−4.848	0.6077	−7.175	0.5026	−1.97	0.5346	−1.927
CIRW10	0.7384	−3.764	0.7269	−14.007	0.9117	−9.28	0.4537	−4.829	0.6173	−7.289	0.5102	−2	0.5349	−1.928

ble configuration. Indeed, as can be seen from Table 11, each dataset prefers a specific combination rule as well as a specific number of solo techniques that construct the ensemble.

However, Table 11 shows that Classical Analogy ensembles using average or IRW as combiner rule were ranked first in 3 datasets each (Ensembles using AVR: Albrecht, COCOMO81, ISBSG; Ensembles using IRW: China, Desharnais, and Miyazaki). Ensemble using median as combination rule was ranked first in Kemerer dataset. With regards to the number of solo techniques, Table 11 shows the number of techniques constructing the ensemble ranked first in each dataset. The main observation is that except for China and COCOMO81 datasets (same number of solo techniques: 6), no number of solo techniques occurred twice.

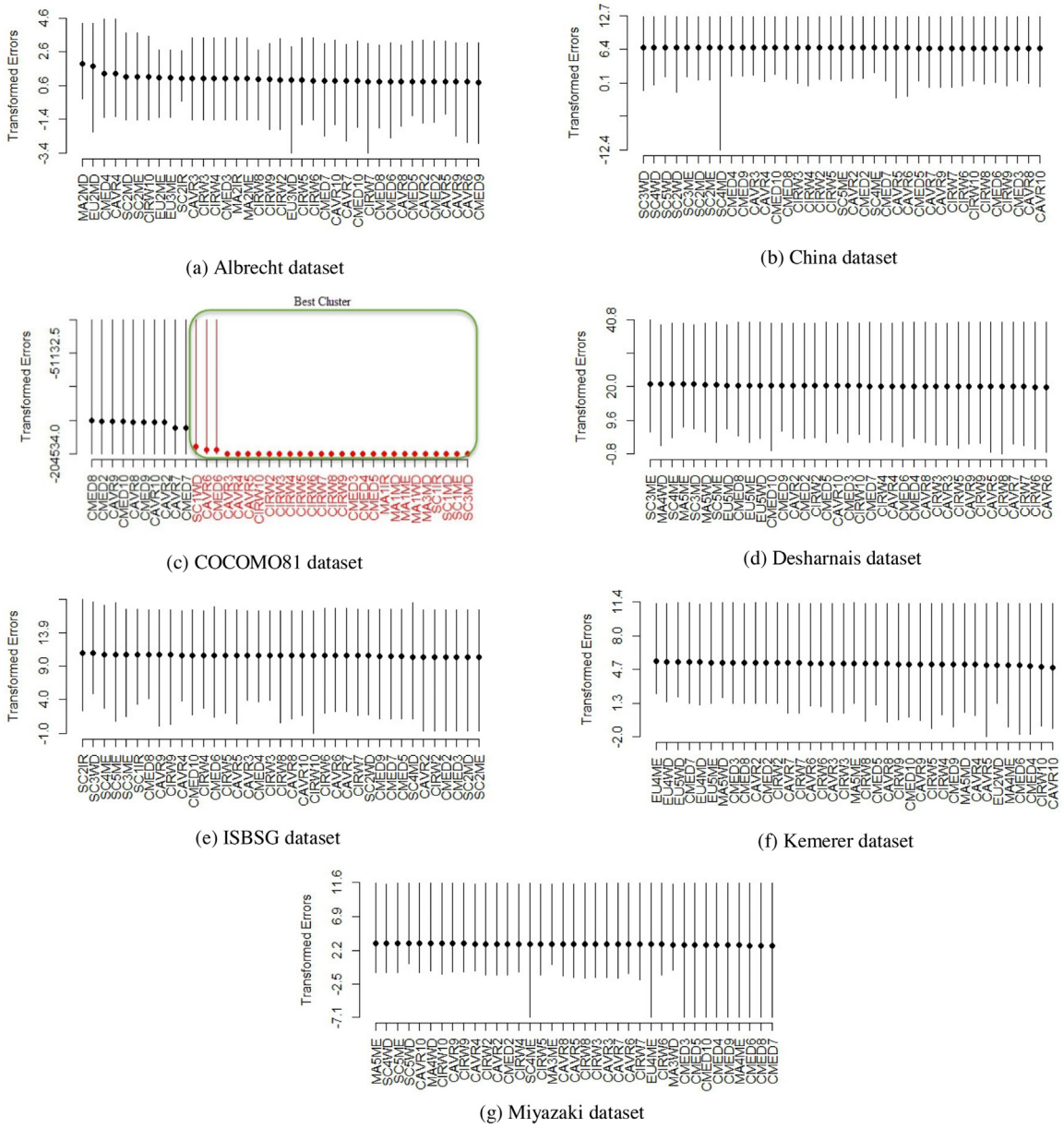
To summarize, Classical Analogy ensembles outperformed solo Classical Analogy techniques. This result was supported by 6 out of 7 selected datasets. Further, Classical Analogy ensembles behave differently from one dataset to another since each dataset favors a special configuration of the best ensemble.

5.3.2. Fuzzy Analogy ensembles

Table 12 presents the summary of SA values and effect size of all constructed Fuzzy Analogy ensembles over the seven datasets where the baseline method is random guessing. We notice that all ensembles over all datasets fall beyond the 5% quantile of random guessing (2nd row), which is not the case of solo Fuzzy Analogy techniques especially for Kemerer dataset. In summary, these ensembles perform better than random guessing since they are constructed using only the solo techniques that performed better than random guessing too. Further, all ensembles show a large improvement over guessing, since the values of effect size are larger than 0.8 (e.g. $\Delta > 0.8$).

As for Classical Analogy techniques, we conducted the SK test in order to identify the best cluster of techniques for each dataset. Fig. 7 presents the results obtained by the SK test; as it can be seen, 5 datasets (Albrecht, Kemerer, Miyazaki, Desharnais, and ISBSG) do not favor any technique since all of them (solo techniques and ensembles techniques) belong to one cluster. Furthermore, 4 clusters were identified by SK in COCOMO81 dataset and 2 clusters in

Please cite this article in press as: A. Idri, et al., Improved estimation of software development effort using Classical and Fuzzy Analogy ensembles, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.08.012



(a) Albrecht dataset
(b) China dataset
(c) COCOMO81 dataset
(d) Desharnais dataset
(e) ISBSG dataset
(f) Kemerer dataset
(g) Miyazaki dataset

Fig. 6. Plot of SK tests of top ten solo Classical Analogy techniques and 27 Classical Analogy ensembles in each dataset. The x-axis represents the selected techniques stored where the better positions start from the right side. The y-axis represents the transformed AEs, each vertical line shows the variation of transformed AEs for each technique, and the small circle represents the mean of transformed AEs. Green box means best cluster. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

China dataset. In fact, the best cluster in COCOMO81 dataset contains 12 techniques (4 solo techniques and 8 ensembles). Further, the ensembles that belong to the best cluster used MED as combiner. As for China dataset, all techniques belong to the best cluster, expect the FIRW5 ensemble.

We stored the best techniques that were identified by SK in order to identify the most accurate ones. The sorting is performed by Borda count. Table 13 lists, for each dataset, the ranking obtained of all best clusters techniques identified by SK. As it can be seen from Table 13, ensembles techniques are ranked first in each dataset. Further, on the top 18, solo techniques appeared twice (TP213 and TP214 techniques in 10th and 18th positions respectively) in Desharnais, and once in Albrecht and Kemerer (TP312 in 13th and TN515 in 11th positions respectively). In COCOMO81 dataset, the first solo technique occupied the 8th position, and the last ensemble technique was ranked 9th. However, for the three remaining datasets the top 18 are all ensemble techniques. The lack of diversity may be the reason behind the superiority of some solo Fuzzy Analogy techniques over Fuzzy Analogy ensembles since, as it can be seen from Fig. 5, not all solo techniques have a large variation of transformed AEs.

Table 13 shows that there is no best Fuzzy Analogy ensemble across all datasets. For example, the FAVR7 ensemble was ranked first in China and ISBSG datasets whereas it is ranked 26th in Albrecht, 32nd in Desharnais, 8th in Kemerer and 13th in Miyazaki datasets, and did not belong to the best cluster in COCOMO81 dataset. As for Classical Analogy ensembles, the ensemble construction based on Fuzzy Analogy variants may behave differently from a dataset to another. As it can be seen from the first row of Table 14, each dataset prefers a specific combination rule as well as a specific number of solo techniques that belong to the ensemble.

Please cite this article in press as: A. Idri, et al., Improved estimation of software development effort using Classical and Fuzzy Analogy ensembles, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.08.012

Table 10
Ranking of Classical Analogy ensembles and the top ten solo Classical Analogy techniques based on Borda count. Ensemble methods are highlighted in bold.

Rank	Dataset						
	Albrecht	China	COCOMO81	Desharnais	ISBSG	Kemerer	Miyazaki
1	**CAVR4**	**CIRW6**	**CAVR6**	**CIRW2**	**CAVR10**	**CMED5**	**CIRW3**
2	**CAVR5**	**CAVR6**	**CIRW6**	**CMED2**	**CIRW10**	**CAVR4**	**CAVR3**
3	**CIRW4**	**CMED6**	**CAVR5**	**CAVR2**	**CMED9**	**CIRW4**	MA4WD
4	**CIRW5**	**CAVR5**	**CIRW5**	**CMED6**	**CIRW8**	**CAVR5**	MA4ME
5	**CMED5**	**CIRW7**	**CAVR3**	**CIRW5**	**CAVR8**	**CAVR10**	**CIRW2**
6	**CAVR6**	**CIRW5**	**CAVR4**	**CIRW4**	**CIRW6**	**CIRW5**	**CMED3**
7	**CIRW6**	**CAVR7**	**CIRW2**	**CIRW3**	**CIRW7**	**CMED6**	**CMED2**
8	**CMED7**	**CMED5**	**CIRW3**	**CAVR5**	**CAVR6**	**CIRW10**	**CIRW4**
9	**CAVR3**	**CIRW8**	**CIRW4**	**CMED7**	**CMED4**	**CMED4**	**CAVR2**
10	**CMED4**	**CIRW3**	**CMED3**	**CAVR4**	**CAVR7**	**CAVR6**	**CIRW5**
11	**CIRW7**	**CMED7**	**CMED4**	**CAVR3**	**CMED7**	**CIRW6**	**CAVR4**
12	**CMED6**	**CAVR8**	**CMED5**	**CIRW6**	**CIRW4**	**CIRW9**	**CMED7**
13	**CAVR7**	**CAVR3**	**CMED6**	**CAVR6**	**CAVR4**	**CAVR9**	**CAVR5**
14	**CIRW3**	**CIRW9**	MA1IR	**CMED5**	**CIRW5**	**CIRW7**	**CIRW6**
15	**CIRW8**	**CIRW10**	MA1MD	**CIRW7**	**CMED10**	**CAVR7**	**CMED4**
16	**CAVR8**	**CAVR9**	MA1ME	**CMED4**	**CMED5**	**CIRW8**	**CAVR6**
17	**CMED8**	**CAVR10**	MA1WD	**CAVR7**	**CIRW9**	**CAVR8**	**CMED6**
18	**CMED10**	**CMED3**	**CIRW7**	**CIRW8**	**CIRW3**	**CMED10**	**CMED8**
19	**CAVR9**	**CMED8**	**CIRW8**	**CIRW9**	**CAVR3**	EU4ME	**CIRW8**
20	**CIRW9**	**CIRW2**	MA3MD	**CMED9**	**CAVR9**	EU4WD	**CIRW7**
21	**CIRW10**	**CAVR2**	**CIRW9**	**CIRW10**	**CMED8**	**CMED9**	**CAVR8**
22	**CAVR10**	**CMED2**	**CIRW10**	**CAVR9**	**CAVR5**	**CAVR3**	**CAVR7**
23	**CAVR2**	**CMED4**	SC3MD	**CMED8**	**CMED6**	**CIRW3**	**CIRW9**
24	**CIRW2**	**CIRW4**	SC1IR	**CAVR8**	SC2ME	EU4MD	**CAVR9**
25	**CMED2**	SC4ME	SC1MD	**CAVR10**	SC2MD	**CMED3**	**CIRW10**
26	**CMED3**	**CAVR4**	SC1ME	**CMED10**	**CMED3**	**CMED2**	**CMED5**
27	SC2MD	**CMED10**	SC1WD	SC3MD	**CMED2**	**CIRW2**	**CAVR10**
28	SC2ME	SC5ME		**CMED3**	**CIRW2**	**CAVR2**	MA3WD
29	SC2IR	**CMED9**		EU5MD	**CAVR2**	EU5WD	**CMED9**
30	**CMED9**	SC3ME		EU5WD	SC3ME	EU5ME	SC4WD
31	EU3MD	SC5WD		SC3ME	SC4ME	**CMED8**	SC4ME
32	EU3ME	SC2MD		MA5WD	SC2IR	MA5ME	**CMED10**
33	MA2IR	SC2ME		EU5ME	SC4MD	**CMED7**	EU4ME
34	EU2MD	SC4WD		SC4ME	SC5ME	MA4ME	SC5WD
35	EU2ME	SC3WD		SC5ME	SC2WD	MA5WD	SC5ME
36	MA2MD	SC4MD		MA5ME	SC1WD	EU2WD	MA3ME
37	MA2ME	SC2WD		MA4WD	SC3WD	MA5MD	MA5ME

Table 11
Combination rule and number of solo techniques for the best Classical Analogy ensemble in each dataset.

Dataset	Albrecht	China	COCOMO81	Desharnais	ISBSG	Kemerer	Miyazaki
Rule	AVR	IRW	AVR	IRW	AVR	MED	IRW
Number of solo techniques	4	6	6	2	10	5	3

In fact, Fuzzy Analogy ensembles using average as combiner rule were ranked first in 4 datasets each. (China, Desharnais, ISBSG and Kemerer datasets). Ensembles using IRW were ranked first in two datasets (Albrecht and Miyazaki). Ensemble using median as combination rule was ranked first in COCOMO81 dataset. The number of techniques that composed the first Fuzzy Analogy ensemble in each dataset is given in second row of Table 14. The ensemble based on top 7 solo techniques was ranked first in China, COCOMO81 and ISBSG datasets. The remaining datasets favor different number of solo techniques (3 for Albrecht, 10 for Desharnais, 9 for Kemerer and 2 for Miyazaki).

The main conclusion is that there is no relationship between the size of datasets and number of solo techniques that construct the Fuzzy Analogy ensembles, since the small dataset such as Kemerer that contains 15 projects favors the ensemble which is based on 9 solo techniques, while the medium dataset such as Miyazaki that contains 48 projects prefers an ensemble which is based on 2 solo techniques. Furthermore, the large datasets such as China, COCOMO81 and ISBSG favor a combination of 7 techniques.

Summing up, Fuzzy Analogy ensembles outperform solo Fuzzy Analogy techniques. There is strong evidence of the superiority of Fuzzy Analogy ensembles, since solo Fuzzy Analogy techniques are usually ranked last. Moreover, each dataset prefers a specific combination rule as well as a specific number of solo techniques that construct the best ensemble.

5.4. Classical Analogy ensembles vs Fuzzy Analogy ensembles

In this section, we present a comparison between Classical Analogy ensembles and Fuzzy Analogy ensembles based on: (1) SA measure, (2) four accuracy measures (MAE, LSD, MBRE and MIBRE) using Borda count, and (3) SK test.

Fig. 8 shows the comparisons between Classical Analogy and Fuzzy Analogy ensembles (with three combination rules and nine solo techniques) over all datasets in terms of SA. Concerning Albrecht dataset, Fig. 8-a shows that Fuzzy Analogy ensembles outperformed Classical Analogy ensembles in all situations (e.g.: different rules, different number of solo techniques). Moreover, Fuzzy Analogy ensembles using median as combiner gave the best results (SA = 82.32% for FMED3) while Classical Analogy ensembles using AVR and IRW gave the best results among all the Classical Analogy ensembles (SA = 76.52% for CAVR3, SA = 76.48% for CIRW3). Note that CAVR, CMED and CIRW behave almost the same with 7, 8 and 10 solo techniques. As for China dataset, Fig. 8-b shows that

Please cite this article in press as: A. Idri, et al., Improved estimation of software development effort using Classical and Fuzzy Analogy ensembles, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.08.012

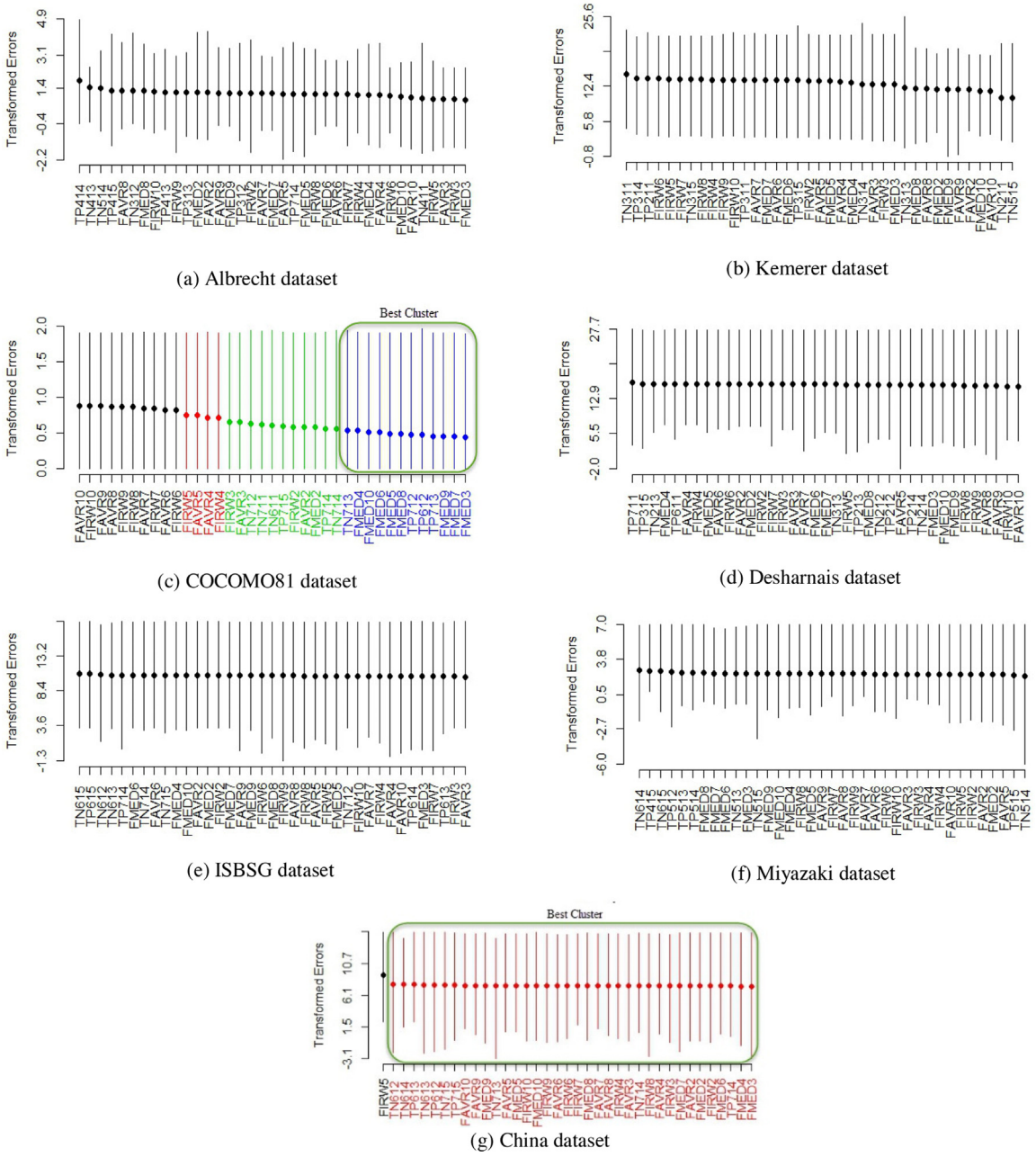
Table 12
SA and effect size of Fuzzy Analogy ensembles.

Dataset	Albrecht		China		COCOMO81		Desharnais		ISBSG		Kemerer		Miyazaki	
$SA_{5\%}$	0.2931		0.082		0.158		0.1554		0.1304		0.331		0.3357	
Ensemble	SA	Δ	SA	Δ	SA	Δ	SA	Δ	SA	Δ	SA	Δ	SA	Δ
FAVR2	0.8019	−4.088	0.71283	−13.735	0.9821	−9.998	0.4597	−4.893	0.6061	−7.157	0.509	−1.995	0.5717	−2.061
FAVR3	0.7993	−4.074	0.71605	−13.798	0.9801	−9.977	0.46	−4.896	0.6115	−7.22	0.5008	−1.963	0.5628	−2.029
FAVR4	0.7799	−3.975	0.71845	−13.844	0.978	−9.956	0.4592	−4.888	0.6058	−7.153	0.4948	−1.94	0.5578	−2.011
FAVR5	0.7992	−4.074	0.71484	−13.774	0.978	−9.956	0.4584	−4.879	0.6033	−7.123	0.4907	−1.924	0.5586	−2.014
FAVR6	0.7949	−4.052	0.71593	−13.795	0.9779	−9.955	0.4574	−4.868	0.6072	−7.169	0.4858	−1.905	0.5581	−2.012
FAVR7	0.7847	−4	0.72375	−13.946	0.9761	−9.936	0.4521	−4.811	0.6152	−7.263	0.5033	−1.973	0.5599	−2.019
FAVR8	0.7729	−3.94	0.71886	−13.852	0.9755	−9.93	0.4693	−4.995	0.6133	−7.242	0.5009	−1.964	0.5568	−2.007
FAVR9	0.7836	−3.994	0.71986	−13.871	0.9752	−9.927	0.466	−4.96	0.6103	−7.206	0.5106	−2.002	0.5581	−2.012
FAVR10	0.7771	−3.961	0.7181	−13.837	0.9748	−9.923	0.4714	−5.017	0.6157	−7.269	0.5052	−1.981	0.5648	−2.036
FMED2	0.8019	−4.088	0.71283	−13.735	0.9821	−9.998	0.4597	−4.893	0.6061	−7.157	0.509	−1.995	0.5717	−2.061
FMED3	0.8233	−4.197	0.71201	−13.72	0.985	−10.03	0.4605	−4.901	0.593	−7.22	0.4853	−1.902	0.5556	−2.003
FMED4	0.7999	−4.077	0.71425	−13.763	0.981	−9.986	0.4594	−4.889	0.5929	−7.153	0.481	−1.886	0.5486	−1.977
FMED5	0.8323	−4.242	0.71484	−13.774	0.9821	−9.997	0.4585	−4.88	0.5972	−7.123	0.478	−1.874	0.5413	−1.951
FMED6	0.8259	−4.21	0.7162	−13.8	0.9833	−10.01	0.4584	−4.878	0.592	−7.169	0.4812	−1.886	0.5442	−1.962
FMED7	0.8239	−4.2	0.71861	−13.847	0.9848	−10.03	0.4586	−4.881	0.5968	−7.263	0.484	−1.898	0.5483	−1.977
FMED8	0.8141	−4.15	0.71644	−13.805	0.984	−10.02	0.4593	−4.888	0.5983	−7.242	0.4851	−1.902	0.5438	−1.96
FMED9	0.8143	−4.151	0.71382	−13.755	0.984	−10.02	0.4615	−4.912	0.5984	−7.206	0.4859	−1.905	0.5468	−1.971
FMED10	0.8076	−4.117	0.70969	−13.675	0.9827	−10	0.4623	−4.92	0.5975	−7.269	0.4927	−1.932	0.5512	−1.987
FIRW2	0.8023	−4.09	0.71283	−13.735	0.9616	−9.789	0.4597	−4.893	0.6062	−7.157	0.5094	−1.997	0.5719	−2.062
FIRW3	0.7996	−4.076	0.71615	−13.8	0.9616	−9.789	0.46	−4.896	0.6116	−7.22	0.5013	−1.965	0.5631	−2.03
FIRW4	0.7808	−3.98	0.7185	−13.845	0.9616	−9.789	0.4592	−4.888	0.6061	−7.153	0.4954	−1.942	0.5581	−2.012
FIRW5	0.7993	−4.074	0.33735	−6.5005	0.9611	−9.783	0.4584	−4.879	0.6038	−7.123	0.4914	−1.927	0.5588	−2.015
FIRW6	0.7951	−4.053	0.71649	−13.806	0.943	−9.6	0.4575	−4.869	0.6073	−7.169	0.4868	−1.909	0.5584	−2.013
FIRW7	0.7854	−4.003	0.72342	−13.94	0.9272	−9.439	0.4524	−4.815	0.6145	−7.263	0.5029	−1.972	0.5601	−2.019
FIRW8	0.7743	−3.947	0.71947	−13.863	0.9139	−9.303	0.4689	−4.99	0.613	−7.242	0.5007	−1.963	0.5571	−2.008
FIRW9	0.7841	−3.997	0.72035	−13.88	0.9033	−9.196	0.4659	−4.958	0.6104	−7.206	0.5092	−1.996	0.5584	−2.013
FIRW10	0.7781	−3.966	0.71894	−13.853	0.8949	−9.11	0.4713	−5.016	0.6149	−7.269	0.5047	−1.979	0.5645	−2.035

Table 13
Fuzzy Analogy ensembles and the top ten solo Fuzzy Analogy techniques based on Borda count.

Rank	Dataset						
	Albrecht	China	COCOMO81	Desharnais	ISBSG	Kemerer	Miyazaki
1	FIRW5	FAVR7	FMED7	FAVR10	FAVR7	FAVR9	FIRW2
2	FIRW3	FIRW7	FMED8	FIRW10	FAVR10	FIRW9	FMED2
3	FAVR5	FIRW8	FMED9	FAVR8	FIRW7	FIRW2	FAVR2
4	FMED5	FIRW9	FMED3	FIRW8	FIRW10	FAVR2	FIRW3
5	FMED3	FIRW10	FMED6	FAVR9	FIRW9	FMED2	FAVR3
6	FAVR3	FIRW4	FMED10	FIRW9	FIRW8	FIRW10	FIRW4
7	FMED6	FAVR9	FMED5	FMED9	FAVR9	FAVR10	FMED3
8	FIRW2	FAVR8	TP713	FMED10	FAVR8	FAVR7	FIRW5
9	FMED4	FAVR4	FMED4	FMED8	FIRW3	FIRW7	FAVR4
10	FAVR2	FAVR10	TP712	TP213	FAVR3	FIRW3	FAVR5
11	FMED2	FIRW3	TN713	FMED2	FIRW6	TN515	FIRW7
12	FMED7	FMED7	TP612	FIRW2	FIRW4	FAVR3	FIRW6
13	TP312	FIRW6		FAVR2	FAVR6	FAVR8	FAVR7
14	FIRW6	FMED6		FMED4	FAVR4	FIRW8	FIRW10
15	FIRW9	FAVR3		FIRW3	FIRW5	FIRW4	FAVR10
16	FIRW4	FAVR6		FAVR3	FAVR5	FAVR4	FMED10
17	FAVR9	FAVR5		FIRW4	FMED2	FMED10	FMED7
18	FAVR6	FMED4		TN214	FAVR2	FIRW5	FAVR6
19	FAVR4	FMED5		FMED3	FIRW2	FAVR5	FMED9
20	FIRW10	FMED3		FAVR4	FMED3	FIRW6	FIRW9
21	FMED9	FMED8		TP214	FMED5	FMED9	FIRW8
22	FAVR10	FAVR2		FMED7	FMED8	FAVR6	FMED4
23	FMED8	FIRW2		FMED5	FMED9	FMED3	FAVR9
24	FMED10	FMED2		TN213	FMED4	FMED8	FAVR8
25	FIRW7	FMED9		FMED6	TP614	FMED7	FMED6
26	FAVR7	FMED10		FIRW5	FMED7	TN211	FMED5
27	FIRW8	TP715		FAVR5	FMED6	TP211	FMED8
28	FAVR8	TP714		FIRW6	FMED10	TN314	TP512
29	TN413	TN713		TP711	TN714	FMED4	TP415
30	TP413	TN714		FAVR6	TN712	TP315	TP513
31	TN411	TN715		FIRW7	TP613	FMED6	TN614
32	TN312	TN614		FAVR7	TN615	TN311	TN615
33	TP714	TP612		TN212	TN715	FMED5	TN514
34	TP313	TN613		TP212	TN613	TN313	TN513
35	TP415	TP613		TP611	TP615	TN315	TP514
36	TP414	TN612		TP315	TN612	TP314	TN515
37	TN514			TN313	TP714	TP311	TP515

Please cite this article in press as: A. Idri, et al., Improved estimation of software development effort using Classical and Fuzzy Analogy ensembles, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.08.012



(a) Albrecht dataset
(b) Kemerer dataset
(c) COCOMO81 dataset
(d) Desharnais dataset
(e) ISBSG dataset
(f) Miyazaki dataset
(g) China dataset

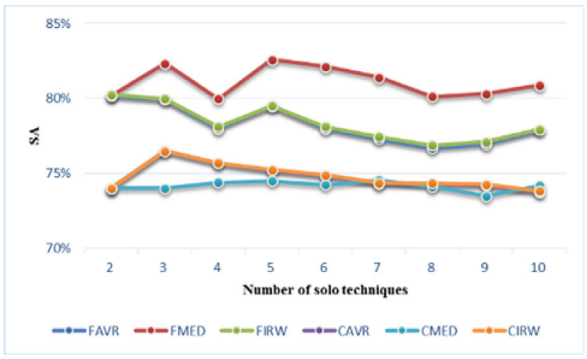
Fig. 7. Plot of SK tests of top ten solo Fuzzy Analogy variants and 27 Fuzzy Analogy ensembles in each dataset. The x-axis represents the selected techniques stored where the better positions start from the right hand side. The y-axis represents the transformed AEs, each vertical line shows the variation of transformed AEs for each technique, and the small circle represents the mean of transformed AEs. Green box means best cluster. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 14
Combination rule and number of solo techniques for the best Fuzzy Analogy ensemble in each dataset.

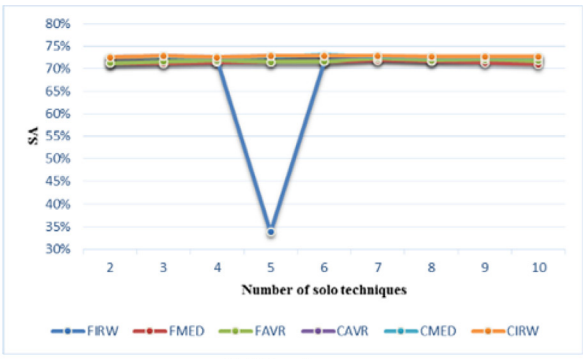
Datasets	Albrecht	China	COCOMO81	Desharnais	ISBSG	Kemerer	Miyazaki
Rule	IRW	AVR	MED	AVR	AVR	AVR	IRW
Number of solo techniques	3	7	7	10	7	9	2

all Classical and Fuzzy Analogy ensembles behave almost the same for all configurations except for FIRW5 which presents the worst SA with 33.73%. As shown in Fig. 8-c, in COCOMO81 dataset, the Fuzzy Analogy ensembles outperformed Classical Analogy regardless of the rules and number of solo techniques. In fact, FAVR, FMED and FIRW behave almost the same with slight improvement for FMED in large number of solo techniques (SA = 98.26% for FMED10, SA = 97.58% for FIRW10 and SA = 97.47 for FAVR10). CAVR, CMED and CIRW behave the same when using 2, 3, 4 or 5 solo techniques. Meanwhile, for large number of solo techniques, CMED gave the best performance (SA = 93.6% for CMED10). Note that the SA for CIRW and CAVR decreases as the number of solo techniques increases beyond 5. For Desharnais dataset, Fig. 8-d shows that Classical Analogy ensembles outperformed Fuzzy Anal-

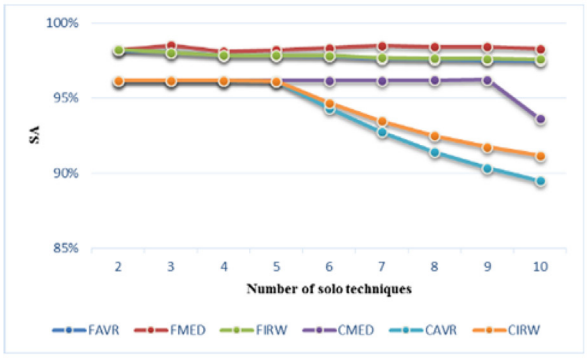
Please cite this article in press as: A. Idri, et al., Improved estimation of software development effort using Classical and Fuzzy Analogy ensembles, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.08.012



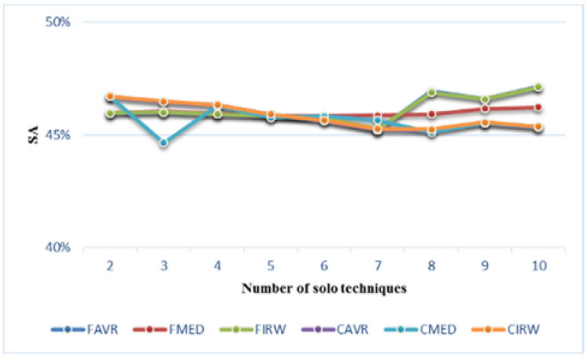
(a) Albrecht dataset, the line of CIRW covers the line of CAVR, the line of FIRW covers the line of FAVR.



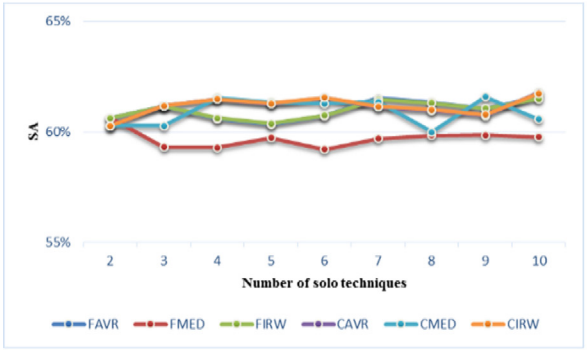
(b) China dataset.



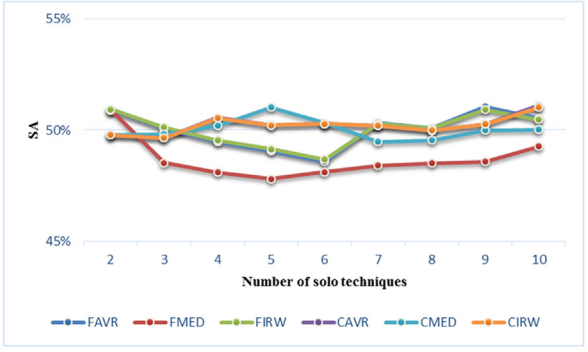
(c) COCOMO81 dataset, the line of FIRW covers the line of FAVR.



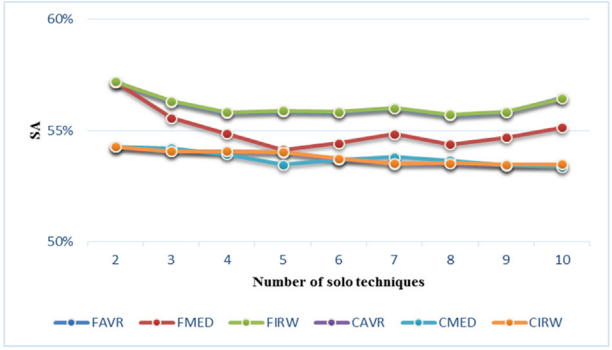
(d) Desharnais dataset, the line of CIRW covers the line of CAVR, the line FAVR covered by the line of FMED and the line of FIRW when the number of solo techniques is lower or higher than 6 respectively.



(e) ISBSG dataset, the line of CIRW covers the line of CAVR, the line of FIRW covers the line of FAVR.



(f) Kemerer dataset, the line of CIRW covers the line of CAVR, the line of FIRW covers the line of FAVR.



(g) Miyazaki dataset, the line of CIRW covers the line of CAVR, the line of FIRW covers the line of FAVR.

Fig. 8. SA values of Classical and Fuzzy Analogy ensembles with three combination rules and ten solo techniques. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

ogy ensembles when the number of solo techniques is under 5, while Fuzzy Analogy ensembles outperformed Classical Analogy ensembles otherwise. Regarding Classical Analogy ensembles, they generally behaved the same (SA = 46.71% for CIRW2 and SA = 46.70% for CAVR2 and CMED2); their SA values decrease when the number of solo techniques increases (SA = 45.32% for CAVR10, SA = 45.34%

Please cite this article in press as: A. Idri, et al., Improved estimation of software development effort using Classical and Fuzzy Analogy ensembles, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.08.012

Table 15
Classical and Fuzzy Analogy ensembles based on Borda count.

Rank	Datasets						
	Albrecht	China	COCOMO81	Desharnais	ISBSG	Kemerer	Miyazaki
1	FIRW5	CIRW6	FMED7	CIRW2	FAVR10	FAVR9	FIRW2
2	FIRW3	CAVR6	FMED8	CAVR2	FAVR7	FIRW9	FAVR2
3	FAVR5	CMED6	FMED9	CMED2	FIRW10	FIRW2	FMED2
4	FMED5	CAVR5	FMED3	FAVR10	FIRW7	FMED2	FIRW3
5	FMED3	CIRW7	FMED10	FIRW10	FIRW8	FAVR2	FAVR3
6	FAVR3	CIRW5	FMED6	FAVR8	FAVR8	CMED5	FIRW4
7	FMED6	CAVR7	CAVR6	CMED6	FIRW3	FAVR7	FMED3
8	FIRW2	CMED5	FMED5	FIRW8	FIRW9	FIRW10	FAVR4
9	FMED4	CIRW8	CIRW6	CIRW4	FAVR3	FAVR10	FIRW5
10	FMED7	CMED7	CAVR5	CIRW3	FAVR9	FIRW7	FAVR5
11	FAVR2	CIRW3	FIRW2	CIRW5	FIRW4	CAVR4	FIRW7
12	FMED2	CAVR8	FMED2	CAVR4	CAVR10	FIRW3	FIRW6
13	FIRW6	CAVR3	FAVR2	CMED7	CIRW10	CIRW4	FAVR7
14	FIRW9	CIRW9	FMED4	CAVR3	FAVR4	FAVR3	FIRW10
15	FAVR9	CIRW10	CIRW5	CAVR5	FIRW6	CAVR10	FMED10
16	FIRW4	CAVR9	CIRW7	FAVR9	FAVR6	FAVR8	FAVR10
17	FAVR6	CAVR10	FIRW3	FIRW9	FIRW5	FIRW8	FMED7
18	FAVR4	CMED3	FIRW4	CIRW6	FAVR5	CIRW10	FAVR6
19	FIRW10	CMED8	CMED8	CAVR6	FAVR2	CAVR5	FMED9
20	FAVR10	CIRW2	CAVR7	CMED5	FMED2	CIRW5	FIRW9
21	FMED9	CAVR2	CMED9	FMED10	FIRW2	CMED6	FIRW8
22	FIRW7	CMED2	FAVR3	FMED9	CMED9	CMED4	FAVR9
23	FMED8	CIRW4	CMED10	CMED4	CIRW6	CAVR6	FMED4
24	FMED10	CMED4	CIRW2	CIRW7	CAVR6	FIRW4	CIRW3
25	FAVR7	CMED10	CIRW3	FMED8	CAVR8	CIRW6	CAVR3
26	CAVR4	CAVR4	CIRW4	FMED2	CIRW8	FAVR4	FAVR8
27	FIRW8	CMED9	CMED2	FAVR2	CIRW7	FMED10	CIRW2
28	CIRW4	FAVR7	CMED3	FIRW2	FMED3	CIRW9	CAVR2
29	FAVR8	FIRW7	CMED4	CAVR7	CAVR7	CAVR9	CMED2
30	CAVR5	FIRW9	CMED5	FMED4	CIRW4	CIRW7	CMED3
31	CMED5	FIRW8	CMED6	FIRW3	CMED4	FIRW5	CIRW4
32	CIRW5	FIRW10	CMED7	FAVR3	CAVR4	CAVR7	CIRW5
33	CAVR6	FIRW4	CAVR2	CIRW8	FMED5	FAVR5	FMED6
34	CMED7	FAVR9	CAVR3	FIRW4	CMED7	CIRW8	CAVR4
35	CIRW6	FAVR8	CAVR4	FAVR4	FMED8	FIRW6	CAVR5
36	CAVR3	FAVR4	FAVR4	CMED8	FMED4	CAVR8	CMED7
37	CMED4	FAVR10	CIRW8	CMED9	CIRW9	FMED9	FMED8
38	CIRW7	FIRW3	FIRW5	CIRW9	FMED9	CMED10	CIRW6
39	CMED6	FMED7	FIRW7	CIRW10	CMED10	FAVR6	CAVR6
40	CIRW3	FIRW6	CIRW9	CAVR8	CIRW5	FMED3	CMED4
41	CAVR7	FMED6	FAVR5	CAVR9	CMED5	FMED8	FMED5
42	CIRW8	FAVR3	FAVR7	FMED5	CAVR3	CMED9	CMED6
43	CAVR8	FAVR6	FIRW6	FMED7	CIRW3	CAVR3	CIRW7
44	CMED8	FAVR5	CAVR8	FMED3	CAVR9	CIRW3	CIRW8
45	CMED10	FMED5	CIRW10	CAVR10	CAVR5	FMED7	CMED8
46	CAVR9	FMED4	FIRW8	CMED10	FMED7	FMED4	CAVR7
47	CIRW9	FMED3	CAVR9	FMED6	CMED8	FMED6	CAVR8
48	CIRW10	FMED8	FAVR6	FIRW5	FMED6	CMED3	CIRW9
49	CAVR10	FIRW2	CAVR10	FAVR5	CMED6	CAVR2	CAVR9
50	CAVR2	FAVR2	FIRW9	FIRW6	FMED10	CMED2	CIRW10
51	CMED2	FMED2	FIRW10	FAVR6	CAVR2	CIRW2	CMED5
52	CMED3	FMED9	FAVR8	FIRW7	CMED2	FMED5	CAVR10
53	CIRW2	FMED10	FAVR9	CMED3	CMED3	CMED8	CMED9
54	CMED9	FIRW5	FAVR10	FAVR7	CIRW2	CMED7	CMED10

for CMED10 and SA = 45.36% for CIRW10). As for Fuzzy Analogy ensembles, they behaved the same until the number of solo techniques reaches 6; after that FIRW and FAVR outperformed FMED (SA = 47.1% for both FAVR10/FIRW10 and SA = 46.22% for FMED10).

For ISBSG and Kemerer datasets, Fig. 8-e and 8-f show that no ensemble distinctly outperformed the others. However, FMED gave the worst performance in both datasets. (SA = 59.2% for ISBSG and SA = 47.80% for Kemerer with 6 and 5 solo techniques respectively).

For Miyazaki dataset, Fig. 8-g shows that Fuzzy Analogy outperformed Classical Analogy regardless of the combiner rule and the number of solo techniques. FAVR and FIRW behaved the same (SA = 56% for both FAVR7/FIRW7). The CAVG, CMED, and CIRW ensembles gave the same performance regardless of the number of solo techniques.

In order to provide more confidence in these results, we sorted Classical and Fuzzy Analogy ensembles according to 4 error measures using Borda count. Table 15 lists the ranking for each dataset and shows that Fuzzy Analogy ensembles are more accurate than Classical Analogy ensembles in five datasets (Albrecht, ISBSG, Miyazaki, Kemerer and COCOMO81 datasets). In fact, for Albrecht, ISBSG and Miyazaki datasets all top ten positions were occupied by Fuzzy Analogy ensembles. As for Kemerer dataset only one Classical Analogy ensemble (CMED5) was occurred in top ten positions (ranked at sixth position). Concerning COCOMO81 dataset, the top ten was occupied by 7 Fuzzy Analogy ensembles and 3 Classical Analogy ensembles. Meanwhile, Classical Analogy ensembles outperformed the Fuzzy Analogy ones in China dataset, since all Classical Analogy ensembles were ranked before the Fuzzy Analogy ones. As for Desharnais dataset, 6 Classical Analogy ensembles appeared in top ten positions. Furthermore, as can be seen in Fig. 9, there is no significant difference between Fuzzy and Classical Analogy ensembles, except for China and COCOMO81 datasets (Fig. 9-b

Please cite this article in press as: A. Idri, et al., Improved estimation of software development effort using Classical and Fuzzy Analogy ensembles, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.08.012

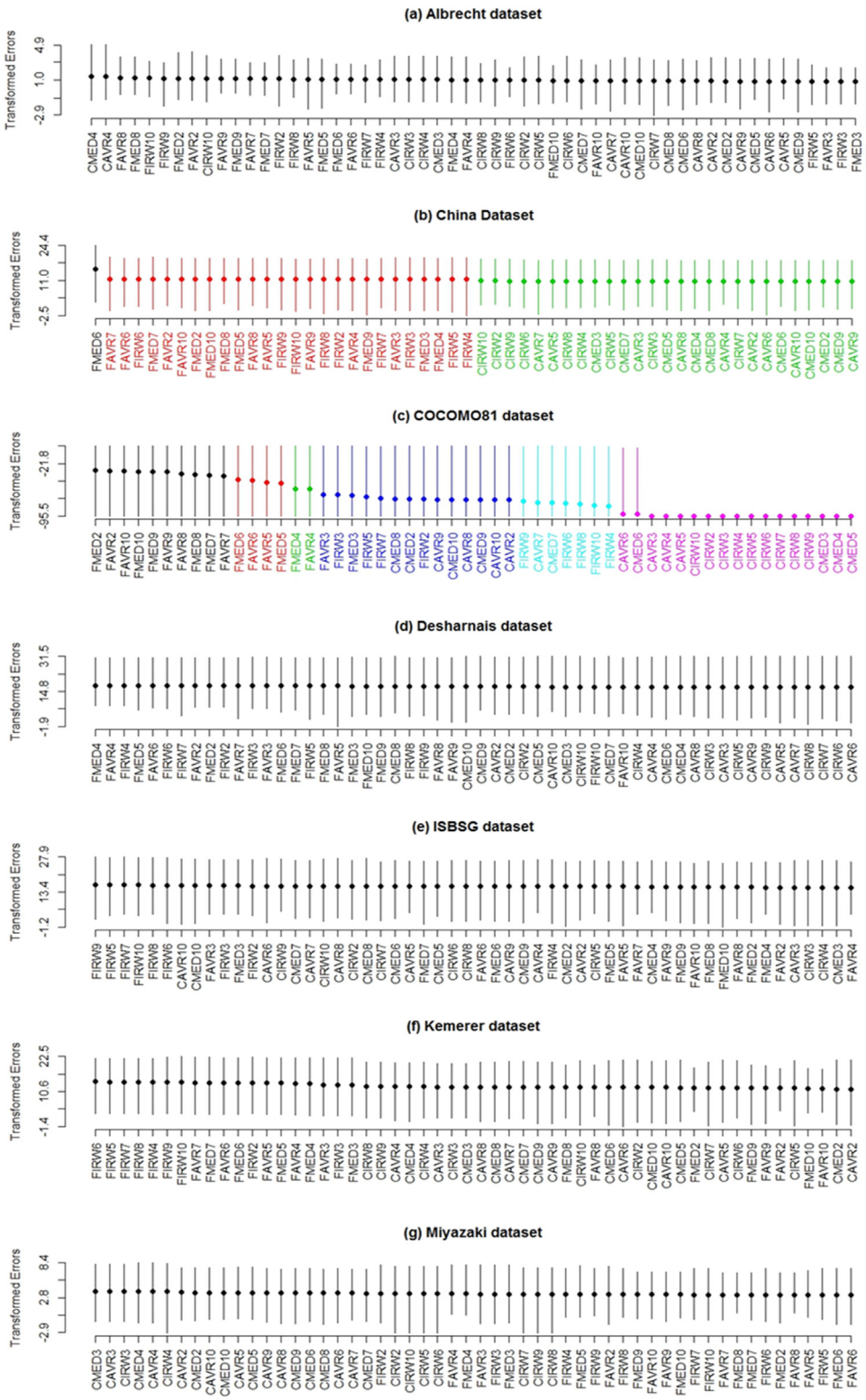


Fig. 9. SK results of Fuzzy and Classical Analogy ensembles on each dataset. The x-axis represents the selected techniques stored where the better positions start from the right side. The y-axis represents the transformed AEs, each vertical line shows the variation of the transformed AEs for each technique, and the small circle represents the mean of the transformed AEs. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

and 9-c) in which Classical Analogy ensembles are favored rather than the Fuzzy Analogy ones.

The main findings of this comparison are:

(1) Fuzzy Analogy ensembles generally outperform Classical Analogy ensembles, in terms of SA and the four performance criteria, since solo Fuzzy Analogy techniques outperformed solo Classical Analogy too [27,29,30,71].
(2) FAVR and FIRW generally behave the same.
(3) There is no conclusion about the number of solo techniques that construct the best ensemble.

Please cite this article in press as: A. Idri, et al., Improved estimation of software development effort using Classical and Fuzzy Analogy ensembles, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.08.012

(4) There is no evidence about the best rule used as combiner in either Fuzzy or Classical Analogy ensembles.
(5) There is no best ensemble across all datasets.

However, these empirical results report two challenges for EEE techniques that are related to: (1) the number of solo techniques, and (2) the combiner rule.

1. The first challenge is how many solo techniques should be combined to construct an accurate ensemble? In this study, the experiments conducted show that each dataset favors a specific number of solo techniques which are chosen from the top ten solo techniques.
2. As for the second challenge, we used three linear rules to aggregate the solo estimates, and none of them was proved to be the best rule across all selected datasets; examining other rules (e.g. non-linear rules) may help to deal with this second challenge.

Furthermore, the results of this study show that ensemble techniques outperform the best solo techniques. However, as for solo techniques, we cannot specify the best ensemble in all circumstances, since the ensembles behave differently from one dataset to another. So, specifying the best one in a specific context seems to be more beneficial than looking for the absolute best one.

6. Threats to validity

This section describes threats to this paper's validity, with respect to internal, external and construct validity.

Internal validity: in this paper one evaluation method was used: the Jackknife validation (LOOCV). The main reason behind using this method over cross validation is that LOOCV generates lower bias than cross validation, and it produces a higher variance estimate [36]. Also, LOOCV can generate the same results in a particular dataset if the evaluation is replicated, which is not the case for cross validation [107].

External validity: regards whether the results obtained in this study can be generalized to other circumstances. This study used seven datasets which contain 1063 projects in total. These projects are collected from different countries and organizations, and they are diverse in terms of their features: this makes them adequate for evaluating the EEE techniques. Some prior works evaluated their proposed ensembles with only one dataset [43,47,108]. However, it will be a good benefit to replicate this study using other software projects datasets. Another external threat may concern the choice of the ensemble members in practice especially when it is not possible to use test-error to choose top solo techniques. Moreover, practitioners may prefer, in addition to accurate prediction, a technique that is easy to understand and simple to use. Hence, developing a well-documented tool that computes all the construction steps of ensembles may be beneficial for practitioners.

Construct validity: aims at answering the question about the reliability of the performances measured through this study. Since this study focuses only on the accuracy of estimates, four criteria were used (MAE, LSD, MIBRE and MBRE) beside SA and effect size. The main reasons behind this choice are that these performance criteria are unbiased and less vulnerable to asymmetry assumption. We recall that the MMRE criterion was not used in this study since it was criticized by many SDEE researchers [53,78,99]. The final conclusion was obtained by drawing a collective decision from the four performance measures using Borda count voting system with equal weights. This strategy was adopted to make sure that we do not favor a particular measure.

7. Conclusion and future work

This paper investigated the potential of Fuzzy and Classical Analogy ensemble techniques in SDEE. For that we used 40 and 100 variants of Fuzzy Analogy and Classical Analogy respectively. We assessed these techniques using SA and effect size over seven datasets through a LOOCV validation, and clustered them using the Scott-Knott test. Thereafter, the EEE selected techniques were sorted according to the Borda count, based on four error measures. The constructed ensembles were based on the top ten techniques for each dataset. Thus, each dataset was used to assess 27 fuzzy and Classical Analogy ensembles. The findings to each of the research questions of this study are as follows:

(RQ1): *Is there evidence that the ensembles based on Classical Analogy outperform solo Classical Analogy techniques?*

Classical Analogy ensembles outperformed solo Classical Analogy techniques; this result was supported by 6 out of 7 selected datasets. Further, the Classical Analogy ensembles presented the same weakness of solo Classical Analogy techniques: they behave differently from one dataset to another. Hence each dataset favors a special configuration of Classical Analogy ensembles.

(RQ2): *Is there evidence that the ensembles based on Fuzzy Analogy outperform solo Fuzzy Analogy techniques?*

Fuzzy Analogy ensembles outperform solo Fuzzy Analogy techniques. There is strong evidence of the superiority of Fuzzy Analogy ensembles since solo Fuzzy Analogy techniques are usually ranked as last. Moreover, each dataset prefers a specific combination rule as well as a specific number of solo techniques that construct the best Fuzzy Analogy ensemble.

(RQ3): *Among the three linear rules used in this study, which of them provides a better accuracy for the Classical Analogy ensembles and Fuzzy Analogy ensembles?*

For both Classical and Fuzzy Analogy ensembles, no particular rule proved to be significantly better than the others. Each dataset prefers a particular combination rule.

(RQ4): *Do Fuzzy Analogy ensembles outperform Classical Analogy ensembles*?

This study found that Fuzzy Analogy ensembles generally outperform Classical Analogy ensembles in terms of SA whatever combiner rule and number of solo techniques used to construct the ensembles. Furthermore, the ranking obtained by Borda count method which is based on 4 performance criteria shows that except for China and Desharnais datasets, Fuzzy Analogy ensembles are better ranked than the Classical ones.

Ongoing work focus on investigating other combination rules especially the non-linear ones. Moreover, replication studies using other datasets are required to confirm or not the findings of this study. Also, it will be interesting to evaluate ensembles based on both solo Classical Analogy and Fuzzy Analogy techniques. This paper dealt only with numerical attributes: hence, investigating datasets with both numerical and categorical data would give more confidence to our results.

Appendix A. Attributes of the seven datasets

Table A1
Albrecht dataset attributes.

Attributes	Description
Input	function points of input
Output	function points of external output
Inquiry	function points of external enquiry
File	function points of internal logical files or entity references
FPAdj	Adjusted function points
RawFPcounts	Total number of rows

Please cite this article in press as: A. Idri, et al., Improved estimation of software development effort using Classical and Fuzzy Analogy ensembles, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.08.012

Table A2
China dataset attributes.

Attributes	Description
AFP	adjusted function points
Input	function points of input
Output	function points of external output
Enquiry	function points of external enquiry
File	function points of internal logical files or entity references
Interface	points of external interface added
Added	function points of new or added functions
Changed	function points of changed functions
Deleted	function points of deleted functions
PDR_UFP	normalized level 1 productivity delivery rate norm
NPDR_AFP	normalized productivity delivery rate
NPDU_UFP	productivity delivery rate (adjusted function points)
Resource	Team type
Dev.Type	development type
Duration	total elapsed time for the project

Table A3
COCOMO81 dataset attributes.

Attributes	Description
SIZE	Software Size
DATA	Database Size
TIME	Execution Time Constraint
STOR	Main Storage Constraint
VIRTMIN, VIRT MAJ	Virtual Machine Volatility
TURN	Computer Turnaround
ACAP	Analyst Capability
AEXP	Applications Experience
PCAP	Programmer Capability
VEXP	Virtual Machine Experience
LEXP	Programming Language Experience
SCED	Required Development

Table A4
Desharnais dataset attributes.

Attributes	Description
ExpEquip	Team experience measured in years
ExpProjMan	Team manager experience measured in years
Transactions	Transactions is a count of basic logical transactions in the system (function points)
Entities	Entities is the number of entities in the systems data model (function points)
Adj_Factor	Function point complexity adjustment factor (Total Processing Complexity)
RawFPs	Unadjusted function points

Table A5
Miyazaki dataset attributes.

Attributes	Description
KSLOC	The number of COBOL source lines in thousands.
SCRN	number of different input or output
FORM	number of different (report) forms
FILE	number of different record formats
ESCRN	total number of data elements in all the screens
EFORM	total number of data elements in all the forms
EFILE	total number of data elements in all the files

Table A6
Kemerer dataset attributes.

Attributes	Description
KSLOC	Kilo Line of Code
AdjFP	Adjusted Function Points
RawFP	Unadjusted Function points
Duration	Duration of project
Language	Programming language
Hardware	Hardware Resources

Table A7
ISBSG dataset attributes.

Attributes	Description
UBBU	User Base – Business Units (Number of business units that the system services)
UBL	User Base – Locations (Number of physical locations being serviced/supported by the installed system).
UBCU	User Base – Concurrent Users (Number of users using the system concurrently).
VAF	Value Adjustment Factor;
MTS	Max Team Size
IC	Input Count
OC	Output Count
EC	Enquiry Count
FC	File Count
IFC	Interface Count

Appendix B. SA and effect size measure for all variants of classical and Fuzzy Analogy techniques

Please cite this article in press as: A. Idri, et al., Improved estimation of software development effort using Classical and Fuzzy Analogy ensembles, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.08.012

Table B1
SA and Effect Size of 100 variants of Classical Analogy technique (Red cell indicates that the variant is greater than random guessing).

Datasets	Albertch		China		COCOMO81		Desharnais		ISBSG		Kemerer		Miyazaki	
SA$_{5\%}$	0.2931		0.082		0.158		0.1554		0.1304		0.331		0.3357	
Variants	SA	Δ	SA	Δ	SA	Δ	SA	Δ	SA	Δ	SA	Δ	SA	Δ
CH1IR	0.643	-3.277	0.5934	-11.435	0.7503	-7.638	0.2737	-2.913	0.4884	-5.766	0.2649	-1.039	0.4938	-1.78
CH1MD	0.643	-3.277	0.5934	-11.435	0.7503	-7.638	0.2737	-2.913	0.4884	-5.766	0.2649	-1.039	0.4938	-1.78
CH1ME	0.643	-3.277	0.5934	-11.435	0.7503	-7.638	0.2737	-2.913	0.4884	-5.766	0.2649	-1.039	0.4938	-1.78
CH1WD	0.643	-3.277	0.5934	-11.435	0.7503	-7.638	0.2737	-2.913	0.4884	-5.766	0.2649	-1.039	0.4938	-1.78
CH2IR	0.6027	-3.072	0.5547	-10.689	0.5094	-5.186	0.225	-2.395	0.4528	-5.346	0.2645	-1.037	0.4753	-1.713
CH2MD	0.6092	-3.105	0.6014	-11.589	0.6695	-6.816	0.3435	-3.656	0.5241	-6.188	0.3814	-1.495	0.5048	-1.82
CH2ME	0.6092	-3.105	0.6014	-11.589	0.6695	-6.816	0.3435	-3.656	0.5241	-6.188	0.3814	-1.495	0.5048	-1.82
CH2WD	0.5944	-3.03	0.5908	-11.385	0.6307	-6.42	0.3436	-3.657	0.4974	-5.873	0.3749	-1.47	0.4996	-1.801
CH3IR	0.532	-2.712	0.5072	-9.7731	0.4525	-4.606	0.1158	-1.233	0.3629	-4.285	0.2969	-1.164	0.4534	-1.635
CH3MD	0.5127	-2.614	0.5932	-11.431	0.6389	-6.504	0.331	-3.523	0.481	-5.679	0.4357	-1.708	0.476	-1.716
CH3ME	0.5448	-2.777	0.6	-11.562	0.5601	-5.701	0.3612	-3.844	0.5142	-6.071	0.4404	-1.727	0.5025	-1.812
CH3WD	0.5249	-2.676	0.5902	-11.372	0.486	-4.947	0.3535	-3.762	0.4889	-5.772	0.4431	-1.737	0.4999	-1.802
CH4IR	0.5307	-2.705	0.482	-9.2879	0.3041	-3.096	0.1363	-1.45	0.401	-4.735	-0.076	0.2982	0.4553	-1.641
CH4MD	0.5005	-2.551	0.586	-11.293	0.5413	-5.51	0.374	-3.981	0.5242	-6.189	0.4771	-1.87	0.4839	-1.744
CH4ME	0.5217	-2.659	0.5966	-11.496	0.474	-4.826	0.3754	-3.996	0.5344	-6.309	0.2893	-1.134	0.4962	-1.789
CH4WD	0.4973	-2.535	0.5863	-11.297	0.395	-4.021	0.3728	-3.967	0.504	-5.951	0.2593	-1.016	0.4904	-1.768
CH5IR	0.4906	-2.501	0.4571	-8.8071	0.2329	-2.371	0.1049	-1.116	0.4526	-5.344	-0.062	0.2431	0.4394	-1.584
CH5MD	0.4714	-2.403	0.5697	-10.977	0.451	-4.591	0.3739	-3.979	0.4975	-5.874	0.421	-1.651	0.477	-1.719
CH5ME	0.5139	-2.62	0.5916	-11.4	0.4454	-4.535	0.3837	-4.083	0.5437	-6.42	0.3369	-1.321	0.4798	-1.73
CH5WD	0.4864	-2.48	0.5813	-11.2	0.3751	-3.819	0.384	-4.087	0.5217	-6.16	0.3263	-1.279	0.4693	-1.692
EU1IR	0.6956	-3.546	0.6263	-12.068	0.8292	-8.441	0.2666	-2.838	0.5519	-6.517	0.3649	-1.431	0.4737	-1.708
EU1MD	0.6956	-3.546	0.6263	-12.068	0.8292	-8.441	0.2666	-2.838	0.5519	-6.517	0.3649	-1.431	0.4737	-1.708
EU1ME	0.6956	-3.546	0.6263	-12.068	0.8292	-8.441	0.2666	-2.838	0.5519	-6.517	0.3649	-1.431	0.4737	-1.708
EU1WD	0.6956	-3.546	0.6263	-12.068	0.8292	-8.441	0.2666	-2.838	0.5519	-6.517	0.3649	-1.431	0.4737	-1.708
EU2IR	0.7197	-3.668	0.6373	-12.281	0.6825	-6.948	0.1853	-1.972	0.4136	-4.883	0.4397	-1.724	0.4941	-1.781
EU2MD	0.7279	-3.71	0.6655	-12.824	0.8363	-8.513	0.3462	-3.685	0.4895	-5.78	0.4653	-1.824	0.5071	-1.828
EU2ME	0.7279	-3.71	0.6655	-12.824	0.8363	-8.513	0.3462	-3.685	0.4895	-5.78	0.4653	-1.824	0.5071	-1.828
EU2WD	0.7198	-3.669	0.6632	-12.779	0.8252	-8.4	0.3425	-3.646	0.4565	-5.389	0.4789	-1.877	0.5063	-1.825
EU3IR	0.7131	-3.635	0.6065	-11.686	0.6595	-6.714	0.1108	-1.179	0.4221	-4.983	0.2698	-1.058	0.4815	-1.736
EU3MD	0.6528	-3.327	0.6463	-12.454	0.9097	-9.261	0.3911	-4.162	0.4943	-5.836	0.4393	-1.722	0.4957	-1.787
EU3ME	0.6906	-3.52	0.6644	-12.803	0.733	-7.461	0.3951	-4.205	0.5282	-6.236	0.4329	-1.697	0.5228	-1.884
EU3WD	0.6667	-3.399	0.6554	-12.63	0.66	-6.719	0.3925	-4.177	0.5071	-5.988	0.4125	-1.617	0.5237	-1.888
EU4IR	0.651	-3.318	0.576	-11.1	0.5715	-5.818	0.0567	-0.603	0.4569	-5.394	0.3012	-1.181	0.4716	-1.7
EU4MD	0.5926	-3.021	0.6459	-12.446	0.7567	-7.703	0.411	-4.374	0.5506	-6.501	0.4916	-1.928	0.5145	-1.855
EU4ME	0.6151	-3.135	0.6629	-12.774	0.6459	-6.576	0.4055	-4.315	0.5512	-6.507	0.4855	-1.903	0.5352	-1.929
EU4WD	0.5797	-2.955	0.6525	-12.573	0.5688	-5.79	0.399	-4.247	0.526	-6.211	0.4867	-1.908	0.5312	-1.915
EU5IR	0.6011	-3.064	0.5463	-10.527	0.494	-5.028	0.0193	-0.205	0.4585	-5.414	0.3035	-1.19	0.465	-1.676
EU5MD	0.5409	-2.757	0.6171	-11.891	0.6906	-7.03	0.4561	-4.854	0.4736	-5.592	0.4437	-1.739	0.5032	-1.814
EU5ME	0.5868	-2.991	0.6456	-12.441	0.5868	-5.973	0.4427	-4.711	0.5397	-6.372	0.4995	-1.958	0.5172	-1.865
EU5WD	0.5523	-2.815	0.6336	-12.209	0.5126	-5.218	0.4417	-4.701	0.5114	-6.038	0.5023	-1.969	0.5127	-1.848
MA1IR	0.7415	-3.78	0.6799	-13.1	0.9616	-9.789	0.2871	-3.055	0.5339	-6.303	0.3481	-1.365	0.4719	-1.701
MA1MD	0.7415	-3.78	0.6799	-13.1	0.9616	-9.789	0.2871	-3.055	0.5339	-6.303	0.3481	-1.365	0.4719	-1.701
MA1ME	0.7415	-3.78	0.6799	-13.1	0.9616	-9.789	0.2871	-3.055	0.5339	-6.303	0.3481	-1.365	0.4719	-1.701
MA1WD	0.7415	-3.78	0.6799	-13.1	0.9616	-9.789	0.2871	-3.055	0.5339	-6.303	0.3481	-1.365	0.4719	-1.701
MA2IR	0.7334	-3.739	0.6764	-13.034	0.7699	-7.837	0.1948	-2.073	0.4542	-5.362	0.4156	-1.629	0.4627	-1.668
MA2MD	0.7281	-3.712	0.6976	-13.442	0.9461	-9.631	0.3578	-3.808	0.5252	-6.201	0.4386	-1.719	0.4818	-1.737
MA2ME	0.7281	-3.712	0.6976	-13.442	0.9461	-9.631	0.3578	-3.808	0.5252	-6.201	0.4386	-1.719	0.4818	-1.737
MA2WD	0.7166	-3.653	0.6901	-13.297	0.9403	-9.572	0.3448	-3.67	0.494	-5.833	0.4467	-1.751	0.4814	-1.735
MA3IR	0.6981	-3.558	0.6729	-12.966	0.6868	-6.992	0.1205	-1.283	0.4989	-5.89	0.2416	-0.947	0.4848	-1.747
MA3MD	0.6213	-3.167	0.6665	-12.842	0.9586	-9.758	0.355	-3.779	0.5254	-6.204	0.4437	-1.74	0.5078	-1.83
MA3ME	0.6668	-3.399	0.6939	-13.37	0.8848	-9.007	0.3599	-3.83	0.5681	-6.708	0.4409	-1.729	0.5289	-1.906
MA3WD	0.6392	-3.258	0.6849	-13.197	0.8593	-8.747	0.3505	-3.73	0.5505	-6.499	0.3991	-1.565	0.5331	-1.922
MA4IR	0.6486	-3.306	0.6596	-12.711	0.6754	-6.875	0.0743	-0.791	0.4863	-5.741	0.2591	-1.016	0.4942	-1.782
MA4MD	0.5976	-3.046	0.6691	-12.892	0.9161	-9.326	0.4007	-4.264	0.5515	-6.512	0.4534	-1.777	0.5092	-1.836
MA4ME	0.6244	-3.183	0.6941	-13.375	0.7628	-7.765	0.4156	-4.424	0.5662	-6.685	0.4787	-1.877	0.5422	-1.954
MA4WD	0.5891	-3.003	0.686	-13.218	0.6992	-7.118	0.4232	-4.504	0.5431	-6.413	0.4554	-1.785	0.5433	-1.959
MA5IR	0.6041	-3.079	0.6505	-12.534	0.6632	-6.751	0.0684	-0.728	0.4626	-5.462	0.2538	-0.995	0.4813	-1.735
MA5MD	0.5479	-2.793	0.6422	-12.375	0.9067	-9.229	0.4258	-4.532	0.4894	-5.778	0.4715	-1.849	0.4861	-1.752
MA5ME	0.5943	-3.029	0.6782	-13.067	0.7004	-7.129	0.4362	-4.643	0.5459	-6.445	0.4923	-1.93	0.5271	-1.9
MA5WD	0.5571	-2.84	0.6668	-12.849	0.6301	-6.414	0.4409	-4.692	0.5198	-6.138	0.4903	-1.922	0.5256	-1.895
MI1IR	-2.16	11.01	-0.217	4.185	-0.769	7.829	0.2426	-2.582	0.3636	-4.293	0.1261	-0.494	0.383	-1.381
MI1MD	-2.16	11.01	-0.217	4.185	-0.769	7.829	0.2426	-2.582	0.3636	-4.293	0.1261	-0.494	0.383	-1.381
MI1ME	-2.16	11.01	-0.217	4.185	-0.769	7.829	0.2426	-2.582	0.3636	-4.293	0.1261	-0.494	0.383	-1.381
MI1WD	-2.16	11.01	-0.217	4.185	-0.769	7.829	0.2426	-2.582	0.3636	-4.293	0.1261	-0.494	0.383	-1.381
MI2IR	-3.518	17.932	-0.258	4.96352	-1.072	10.91	-0.007	0.078	0.348	-4.108	-0.052	0.2022	0.1836	-0.662
MI2MD	-2.352	11.988	0.0247	-0.4768	-0.604	6.147	0.2064	-2.196	0.337	-3.979	0.1387	-0.544	0.3174	-1.144
MI2ME	-2.352	11.988	0.0247	-0.4768	-0.604	6.147	0.2064	-2.196	0.337	-3.979	0.1387	-0.544	0.3174	-1.144
MI2WD	-2.352	11.988	0.0247	-0.4768	-0.604	6.147	0.2064	-2.196	0.337	-3.979	0.1387	-0.544	0.3174	-1.144
MI3IR	-2.508	12.785	-0.023	0.43588	-0.695	7.077	0.1718	-1.829	0.1485	-1.753	-1.637	6.4169	0.2311	-0.833
MI3MD	-2.42	12.336	0.2292	-4.417	-0.442	4.497	0.2358	-2.51	0.3647	-4.306	0.0683	-0.268	0.3875	-1.397
MI3ME	-1.29	6.5757	0.234	-4.5095	-0.212	2.153	0.3307	-3.519	0.3013	-3.558	-0.653	2.5617	0.388	-1.399
MI3WD	-1.29	6.5757	0.234	-4.5095	-0.212	2.153	0.3307	-3.519	0.3013	-3.558	-0.653	2.5617	0.388	-1.399
MI4IR	-2.039	10.394	-0.031	0.60679	-0.473	4.818	0.1311	-1.396	0.1856	-2.192	-1.238	4.8518	0.1983	-0.715
MI4MD	-0.964	4.9147	0.2764	-5.3265	0.0413	-0.421	0.2926	-3.114	0.3639	-4.297	0.2824	-1.107	0.389	-1.402
MI4ME	-0.849	4.3259	0.2568	-4.9489	-0.026	0.268	0.33	-3.512	0.3294	-3.889	-0.281	1.1016	0.3886	-1.401
MI4WD	-0.849	4.3259	0.2568	-4.9489	-0.026	0.268	0.33	-3.512	0.3294	-3.889	-0.281	1.1016	0.3886	-1.401
MI5IR	-1.8	9.1737	-0.025	0.4851	-0.278	2.828	0.1678	-1.786	0.2395	-2.828	-1.167	4.5748	0.1721	-0.62
MI5MD	0.0677	-0.345	0.3159	-6.0863	0.3943	-4.014	0.3284	-3.495	0.3646	-4.305	0.0092	-0.036	0.3902	-1.406
MI5ME	-0.646	3.2913	0.275	-5.2989	0.0998	-1.016	0.3355	-3.571	0.3471	-4.098	-0.179	0.701	0.3885	-1.401
MI5WD	-0.646	3.2913	0.275	-5.2989	0.0998	-1.016	0.3355	-3.571	0.3471	-4.098	-0.179	0.701	0.3885	-1.401
SC1IR	0.6316	-3.219	0.686	-13.218	0.8188	-8.336	0.3779	-4.022	0.6025	-7.114	0.1992	-0.781	0.4829	-1.741
SC1MD	0.6316	-3.219	0.686	-13.218	0.8188	-8.336	0.3779	-4.022	0.6025	-7.114	0.1992	-0.781	0.4829	-1.741
SC1ME	0.6316	-3.219	0.686	-13.218	0.8188	-8.336	0.3779	-4.022	0.6025	-7.114	0.1992	-0.781	0.4829	-1.741
SC1WD	0.6316	-3.219	0.686	-13.218	0.8188	-8.336	0.3779	-4.022	0.6025	-7.114	0.1992	-0.781	0.4829	-1.741
SC2IR	0.738	-3.762	0.6678	-12.868	0.6577	-6.695	0.2758	-2.936	0.5307	-6.266	0.2055	-0.806	0.5046	-1.819
SC2MD	0.7402	-3.773	0.7091	-13.663	0.8157	-8.303	0.4089	-4.352	0.6026	-7.115	0.3305	-1.296	0.5099	-1.838
SC2ME	0.7402	-3.773	0.7091	-13.663	0.8157	-8.303	0.4089	-4.352	0.6026	-7.115	0.3305	-1.296	0.5099	-1.838
SC2WD	0.7191	-3.666	0.6927	-13.348	0.8002	-8.146	0.3942	-4.195	0.563	-6.647	0.3504	-1.374	0.5056	-1.823
SC3IR	0.6947	-3.541	0.6542	-12.605	0.5836	-5.941	0.12	-1.277	0.5394	-6.368	0.205	-0.804	0.4814	-1.735
SC3MD	0.6064	-3.091	0.6964	-13.42	0.8522	-8.676	0.4534	-4.825	0.5498	-6.492	0.2661	-1.043	0.4909	-1.77
SC3ME	0.6652	-3.391	0.7223	-13.917	0.6959	-7.084	0.4228	-4.5	0.6076	-7.173	0.4055	-1.59	0.5237	-1.888
SC3WD	0.6314	-3.219	0.7054	-13.593	0.5351	-5.447	0.3859	-4.107	0.5662	-6.685	0.4385	-1.719	0.5206	-1.877
SC4IR	0.6391	-3.258	0.6337	-12.21	0.4902	-4.99	0.0301	-0.32	0.5052	-5.965	0.17	-0.666	0.4901	-1.767
SC4MD	0.6021	-3.069	0.7021	-13.53	0.6693	-6.814	0.4124	-4.389	0.5746	-6.784	0.3917	-1.535	0.5121	-1.846
SC4ME	0.6578	-3.353	0.7274	-14.016	0.6065	-6.174	0.4222	-4.493	0.595	-7.025	0.4443	-1.742	0.5313	-1.915
SC4WD	0.6188	-3.154	0.7097	-13.675	0.449	-4.57	0.4012	-4.27	0.5402	-6.378	0.4686	-1.837	0.5327	-1.92
SC5IR	0.5608	-2.859	0.6239	-12.021	0.4404	-4.483	-0.023	0.2399	0.4702	-5.552	0.0743	-0.291	0.4704	-1.696
SC5MD	0.5824	-2.969	0.6821	-13.143	0.5585	-5.686	0.4298	-4.574	0.5158	-6.09	0.4274	-1.676	0.5021	-1.81
SC5ME	0.65	-3.313	0.7214	-13.9	0.5809	-5.913	0.4189	-4.459	0.5811	-6.861	0.4343	-1.703	0.5192	-1.872
SC5WD	0.6065	-3.092	0.7072	-13.627	0.4498	-4.579	0.3968	-4.224	0.524	-6.186	0.4224	-1.656	0.5175	-1.865

Please cite this article in press as: A. Idri, et al., Improved estimation of software development effort using Classical and Fuzzy Analogy ensembles, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.08.012

Table B2
SA and Effect size of 60 variants of Fuzzy Analogy technique (Red cell indicates that the variant is greater than random guessing).

Datasets	Albrecht		China		COCOMO81		Desharnais		ISBSG		Kemerer		Miyazaki	
$SA_{5\%}$	0.2931		0.082		0.158		0.1554		0.1304		0.331		0.3357	
Variants	SA	Δ	SA	Δ	SA	Δ	SA	Δ	SA	Δ	SA	Δ	SA	Δ
TN211	0.7018	-3.577	0.4612	-8.887	0.8496	-8.648	0.4047	-4.307	0.5036	-5.946	0.4807	-1.885	0.4749	-1.712
TN212	0.6075	-3.097	0.4873	-9.39	0.7364	-7.496	0.4512	-4.802	0.512	-6.045	0.4085	-1.602	0.477	-1.72
TN213	0.5981	-3.049	0.4874	-9.391	0.6664	-6.783	0.4569	-4.863	0.5047	-5.959	0.4119	-1.615	0.4776	-1.722
TN214	0.5606	-2.858	0.4931	-9.501	0.5801	-5.905	0.4605	-4.901	0.5081	-5.999	0.3668	-1.438	0.4879	-1.759
TN215	0.5249	-2.676	0.4642	-8.945	0.417	-4.245	0.3862	-4.111	0.457	-5.396	0.344	-1.349	0.4696	-1.693
TN311	0.6371	-3.247	0.5227	-10.07	0.8956	-9.117	0.3817	-4.062	0.6036	-7.126	0.4683	-1.836	0.5052	-1.821
TN312	0.7012	-3.574	0.5819	-11.21	0.827	-8.419	0.4051	-4.311	0.6198	-7.318	0.408	-1.6	0.5026	-1.812
TN313	0.6518	-3.323	0.5839	-11.25	0.8258	-8.407	0.4142	-4.408	0.5922	-6.992	0.4616	-1.81	0.513	-1.849
TN314	0.633	-3.226	0.5844	-11.26	0.7495	-7.629	0.3816	-4.062	0.5412	-6.39	0.4824	-1.891	0.5124	-1.847
TN315	0.6747	-3.439	0.5654	-10.89	0.6122	-6.232	0.3868	-4.116	0.4813	-5.682	0.4768	-1.869	0.4994	-1.8
TN411	0.7909	-4.032	0.5941	-11.45	0.9435	-9.604	0.3304	-3.516	0.5616	-6.631	0.329	-1.29	0.4636	-1.671
TN412	0.6341	-3.232	0.6197	-11.94	0.9202	-9.367	0.3556	-3.785	0.572	-6.754	0.3866	-1.516	0.4902	-1.767
TN413	0.7434	-3.789	0.6008	-11.58	0.9259	-9.426	0.3678	-3.914	0.541	-6.387	0.3727	-1.461	0.5051	-1.821
TN414	0.599	-3.053	0.6187	-11.92	0.9268	-9.434	0.3643	-3.877	0.528	-6.234	0.3883	-1.523	0.506	-1.824
TN415	0.6811	-3.472	0.5947	-11.46	0.8572	-8.726	0.3939	-4.193	0.5411	-6.388	0.383	-1.502	0.5129	-1.849
TN511	0.6198	-3.159	0.6816	-13.13	0.9625	-9.798	0.3037	-3.232	0.5861	-6.92	0.2735	-1.072	0.4924	-1.775
TN512	0.6388	-3.256	0.6513	-12.55	0.9562	-9.733	0.2959	-3.149	0.5162	-6.095	0.4214	-1.652	0.5161	-1.861
TN513	0.643	-3.277	0.6672	-12.86	0.9551	-9.723	0.3682	-3.919	0.5288	-6.244	0.3027	-1.187	0.5426	-1.956
TN514	0.7178	-3.659	0.6554	-12.63	0.956	-9.732	0.399	-4.246	0.5008	-5.913	0.3444	-1.35	0.5414	-1.952
TN515	0.7713	-3.932	0.6538	-12.6	0.9417	-9.587	0.3821	-4.067	0.5128	-6.054	0.5075	-1.99	0.5437	-1.96
TN611	0.523	-2.666	0.6791	-13.09	0.9738	-9.913	0.3286	-3.497	0.5318	-6.279	0.3551	-1.392	0.5136	-1.852
TN612	0.5821	-2.967	0.6949	-13.39	0.9674	-9.848	0.3537	-3.765	0.5861	-6.92	0.372	-1.459	0.5335	-1.923
TN613	0.5796	-2.954	0.6771	-13.05	0.9517	-9.688	0.3305	-3.518	0.5567	-6.573	0.3554	-1.393	0.5224	-1.883
TN614	0.6997	-3.567	0.683	-13.16	0.9417	-9.586	0.4095	-4.358	0.5579	-6.587	0.4115	-1.613	0.5395	-1.945
TN615	0.773	-3.94	0.6566	-12.65	0.9593	-9.765	0.3244	-3.453	0.5799	-6.847	0.327	-1.282	0.5183	-1.868
TN711	0.5978	-3.047	0.6992	-13.47	0.9698	-9.873	0.3129	-3.331	0.5174	-6.109	0.3672	-1.44	0.5352	-1.929
TN712	0.548	-2.793	0.6695	-12.9	0.968	-9.854	0.3387	-3.604	0.584	-6.895	0.3531	-1.384	0.5068	-1.827
TN713	0.5807	-2.96	0.6977	-13.44	0.9742	-9.917	0.3461	-3.683	0.574	-6.777	0.1687	-0.661	0.4983	-1.796
TN714	0.6319	-3.221	0.698	-13.45	0.9691	-9.865	0.4404	-4.688	0.5926	-6.997	0.3386	-1.327	0.478	-1.723
TN715	0.7823	-3.988	0.6838	-13.18	0.9705	-9.879	0.2456	-2.614	0.5909	-6.977	0.3197	-1.254	0.5089	-1.835
TP211	0.6996	-3.566	0.4638	-8.937	0.8496	-8.649	0.3801	-4.046	0.4937	-5.83	0.4806	-1.884	0.5029	-1.813
TP212	0.6105	-3.112	0.4882	-9.407	0.7367	-7.499	0.4512	-4.802	0.5119	-6.044	0.4085	-1.601	0.477	-1.72
TP213	0.5983	-3.05	0.4872	-9.389	0.6677	-6.797	0.4589	-4.884	0.5047	-5.959	0.4119	-1.615	0.4776	-1.722
TP214	0.5606	-2.857	0.4928	-9.496	0.5801	-5.905	0.4605	-4.901	0.5081	-5.999	0.3668	-1.438	0.4879	-1.759
TP215	0.5249	-2.676	0.4557	-8.781	0.417	-4.245	0.421	-4.481	0.457	-5.396	0.344	-1.349	0.4696	-1.693
TP311	0.7734	-3.942	0.5397	-10.4	0.9113	-9.276	0.3792	-4.035	0.5961	-7.038	0.4509	-1.768	0.4072	-1.468
TP312	0.841	-4.287	0.5801	-11.18	0.9044	-9.207	0.3759	-4.001	0.6243	-7.371	0.2921	-1.145	0.5041	-1.817
TP313	0.7624	-3.886	0.5851	-11.27	0.8482	-8.635	0.389	-4.14	0.5985	-7.066	0.3734	-1.464	0.5118	-1.845
TP314	0.6197	-3.159	0.5954	-11.47	0.7593	-7.729	0.3891	-4.141	0.5386	-6.359	0.4741	-1.859	0.5102	-1.839
TP315	0.6121	-3.12	0.5821	-11.22	0.6377	-6.492	0.3991	-4.247	0.4984	-5.885	0.4844	-1.899	0.5132	-1.85
TP411	0.7866	-4.009	0.608	-11.71	0.9661	-9.835	0.3567	-3.796	0.5505	-6.5	0.3977	-1.559	0.5023	-1.811
TP412	0.8069	-4.113	0.6059	-11.67	0.9657	-9.831	0.342	-3.639	0.5592	-6.602	0.3648	-1.43	0.5225	-1.883
TP413	0.7634	-3.891	0.6235	-12.01	0.9608	-9.78	0.3945	-4.199	0.5608	-6.621	0.3736	-1.465	0.528	-1.903
TP414	0.698	-3.558	0.6111	-11.77	0.9336	-9.504	0.4145	-4.412	0.5171	-6.106	0.3824	-1.499	0.5192	-1.872
TP415	0.6872	-3.503	0.6174	-11.9	0.9208	-9.374	0.4511	-4.801	0.5342	-6.307	0.3776	-1.48	0.5271	-1.9
TP511	0.6956	-3.546	0.6703	-12.92	0.9316	-9.484	0.392	-4.172	0.5683	-6.709	0.3359	-1.317	0.5332	-1.922
TP512	0.7437	-3.791	0.654	-12.6	0.9676	-9.85	0.3298	-3.51	0.4484	-5.294	0.4105	-1.61	0.5188	-1.87
TP513	0.7137	-3.638	0.6568	-12.66	0.9627	-9.8	0.2778	-2.956	0.4679	-5.524	0.4354	-1.707	0.5284	-1.905
TP514	0.6885	-3.51	0.6559	-12.64	0.9568	-9.74	0.3614	-3.846	0.4528	-5.346	0.2705	-1.06	0.5499	-1.982
TP515	0.7699	-3.924	0.6407	-12.34	0.957	-9.742	0.3154	-3.357	0.4977	-5.876	0.4364	-1.711	0.5837	-2.104
TP611	0.6232	-3.177	0.6767	-13.04	0.9494	-9.664	0.4252	-4.525	0.5329	-6.292	0.4488	-1.759	N	N
TP612	0.6554	-3.341	0.7093	-13.67	0.9615	-9.788	0.3397	-3.615	0.527	-6.223	0.1745	-0.684	0.4653	-1.677
TP613	0.5563	-2.836	0.6787	-13.08	0.9691	-9.866	0.2992	-3.184	0.5658	-6.68	0.4489	-1.76	0.4776	-1.722
TP614	0.7738	-3.944	0.6525	-12.57	0.9416	-9.585	0.2557	-2.721	0.5981	-7.062	0.431	-1.69	0.5084	-1.833
TP615	0.7579	-3.863	0.6493	-12.51	0.9682	-9.856	0.3258	-3.468	0.5783	-6.828	0.3574	-1.401	0.4989	-1.798
TP711	0.7136	-3.638	0.6828	-13.16	0.9376	-9.545	0.4093	-4.356	0.536	-6.328	0.3658	-1.434	0.5103	-1.84
TP712	0.7142	-3.64	0.6719	-12.95	0.9794	-9.97	0.3696	-3.934	0.5274	-6.227	0.3846	-1.508	0.5003	-1.804
TP713	0.7066	-3.602	0.6735	-12.98	0.9837	-10.01	0.3953	-4.207	0.5532	-6.532	0.3615	-1.417	0.4576	-1.649
TP714	0.7249	-3.695	0.7057	-13.6	0.9773	-9.948	0.4203	-4.473	0.5612	-6.625	0.3547	-1.391	0.4443	-1.602
TP715	0.7301	-3.721	0.7026	-13.54	0.9704	-9.879	0.3287	-3.499	0.5603	-6.615	0.2819	-1.105	0.4544	-1.638

References

[1] J. Wen, S. Li, Z. Lin, Y. Hu, C. Huang, Systematic literature review of machine learning based software development effort estimation models, Inf. Softw. Technol. 54 (2012) 41–59, http://dx.doi.org/10.1016/j.infsof.2011.09.002.

[2] L.L. Minku, X. Yao, Software effort estimation As a multiobjective learning problem, ACM Trans. Softw. Eng. Methodol. 22 (35) (2013) 1–35, http://dx.doi.org/10.1145/2522920.2522928 (32).

[3] P. Brooks Frederick Jr., The Mythical Man-Month: Essays on Software Engineering (1975).

[4] I.F. de Barcelos Tronto, J.D.S. da Silva, N. Sant'Anna, An investigation of artificial neural networks based prediction systems in software project management, J. Syst. Softw. 81 (2008) 356–367, http://dx.doi.org/10.1016/j.jss.2007.05.011.

[5] R.T. Hughes, Expert judgement as an estimating method, Inf. Softw. Technol. 38 (1996) 67–75, http://dx.doi.org/10.1016/0950-5849(95)01045-9.

[6] B. Boehm, Software engineering economics, IEEE Trans. Softw. Eng. 10 (1984) 4–21.

[7] L.H. Putnam, A general empirical solution to the macro software sizing and estimating problem, IEEE Trans. Softw. Eng. SE-4 (1978) 345–361, http://dx.doi.org/10.1109/TSE.1978.231521.

[8] J.E. Albrecht, Software function, source lines of code, and development effort prediction: a software science validation, IEEE Trans. Softw. Eng SE-9 (1983) 639–648, http://dx.doi.org/10.1109/TSE.1983.235271.

[9] A. Idri, A. Abran, T.M. Khoshgoftaar, Estimating software project effort by analogy based on linguistic values, Proc Eighth IEEE Symp. Softw. Metrics (2002) 21–30, http://dx.doi.org/10.1109/METRIC.2002.1011322.

[10] M. Jorgensen, M. Shepperd, A systematic review of software development cost estimation studies, IEEE Trans. Softw. Eng. 33 (2007) 33–53, http://dx.doi.org/10.1109/TSE.2007.256943.

[11] M. Shepperd, C. Schofield, Estimating software project effort using analogies, IEEE Trans. Softw. Eng. 23 (1997) 736–743, http://dx.doi.org/10.1109/32.637387.

[12] A. Heiat, Comparison of artificial neural network and regression models for estimating software development effort, Inf. Softw. Technol. 44 (2002) 911–922, http://dx.doi.org/10.1016/S0950-5849(02)00128-3.

[13] B. Baskeles, B. Turhan, A. Bener, Software effort estimation using machine learning methods, Proc 22nd Int. Symp. Comput. Inf. Sci. (2007) 1–6, http://dx.doi.org/10.1109/ISCIS.2007.4456863.

[14] P.C. Pendharkar, G.H. Subramanian, J.A. Rodger, A probabilistic model for predicting software development effort, IEEE Trans. Softw. Eng. 31 (2005) 615–624, http://dx.doi.org/10.1109/TSE.2005.75.

Please cite this article in press as: A. Idri, et al., Improved estimation of software development effort using Classical and Fuzzy Analogy ensembles, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.08.012

[15] A.L.I. Oliveira, Estimation of software project effort with support vector regression, Neurocomputing 69 (2006) 1749–1753, http://dx.doi.org/10.1016/j.neucom.2005.12.119.
[16] K. Srinivasan, D. Fisher, Machine learning approaches to estimating software development effort, IEEE Trans. Softw. Eng. 21 (1995) 126–137, http://dx.doi.org/10.1109/32.345828.
[17] C.J. Burgess, M. Lefley, M. Le, Can genetic programming improve software effort estimation? A comparative evaluation, Inf. Softw. Technol. 43 (2001) 863–873, http://dx.doi.org/10.1016/S0950-5849(01)00192-6.
[18] S. Bibi, I. Stamelos, L. Angelis, Combining probabilistic models for explanatory productivity estimation, Inf. Softw. Technol. 50 (2008) 656–669, http://dx.doi.org/10.1016/j.infsof.2007.06.004.
[19] E. Mendes, A comparative study of cost estimation models for web hypermedia applications, Empir. Softw. Eng. 8 (2003) 163–196, http://dx.doi.org/10.1023/A:1023062629183.
[20] F. Walkerden, R. Jeffery, An empirical study of analogy-based software effort estimation, Empir. Softw. Eng. 158 (1999) 135–158, http://dx.doi.org/10.1023/A:1009872202035.
[21] N. Mittas, L. Angelis, LSEbA. Least squares regression and estimation by analogy in a semi-parametric model for Software Cost Estimation, Empir. Softw. Eng. 15 (2010) 523–555, http://dx.doi.org/10.1007/s10664-010-9128-6.
[22] J.W. Keung, B.A. Kitchenham, D.R. Jeffery, Analogy-X. Providing statistical inference to analogy-based software cost estimation, IEEE Trans. Softw. Eng. 34 (2008) 471–484, http://dx.doi.org/10.1109/TSE.2008.34.
[23] M. Azzeh, D. Neagu, P. Cowling, Software effort estimation based on weighted fuzzy grey relational analysis, Proc 5th Int. Conf. Predict. Model. Softw. Eng. – PROMISE ' 09 (2009) 8:1–8:10, http://dx.doi.org/10.1145/1540438.1540450.
[24] J. Li, G. Ruhe, A. Al-Emran, M.M. Richter, A flexible method for software effort estimation by analogy, Empir. Softw. Eng. 12 (2007) 65–106, http://dx.doi.org/10.1007/s10664-006-7552-4.
[25] A. Idri, T.M. Khoshgoftaar, A. Abran, Investigating soft computing in case-based reasoning for software cost estimation, Eng. Intell. Syst. Electr. Eng. Commun. 10 (2002) 147–157.
[26] A. Idri, A. Zahi, A. Abran, Generating fuzzy term sets for software project attributes using fuzzy C-Means and real coded genetic algorithms, in: Int Conf. Inf. Commun. Technol. Muslim World, Malaysia, 2006, pp. 120–127.
[27] F.A. Amazal, A. Idri, A. Abran, An analogy-Based approach to estimation of software development effort using categorical data, Jt Conf. Int. Work. Softw. Meas. Int. Conf. Softw. Process Prod. Meas. (2014) 252–262, http://dx.doi.org/10.1109/IWSM.Mensura.2014.31.
[28] A. Idri, A. Zahi, A. Abran, Software cost estimation by Fuzzy Analogy for web hypermedia applications, in: Proc Int. Conf. Softw. Process Prod. Meas., Cadiz, Spain, 2006, pp. 53–62.
[29] A. Idri, A. Zahi, Software cost estimation by classical and Fuzzy Analogy for Web Hypermedia Applications: a replicated study, in: Proc IEEE Symp. Comput. Intell. Data Min., Singapore, 2013, pp. 207–213, http://dx.doi.org/10.1109/CIDM.2013.6597238.
[30] A. Idri, F. azzahra amazal, A., abran, accuracy comparison of analogy-Based software development effort estimation techniques, Int. J. Intell. Syst. 0 (2015) 1–25, http://dx.doi.org/10.1002/int.
[31] A. Idri, F.A. Amazal, Software cost estimation by fuzzy analogy for ISBSG repository, in: Proc 10th Int. FLINS Conf. Uncertain. Model. Knowl. Eng. Decis. Mak., Istanbul, Turkey, 2012.
[32] L.A. Zadeh, From computing with numbers to computing with words — from manipulation of measurements to manipulation of perceptions, IEEE Trans. Circuits Syst. 45 (1999) 105–119.
[33] A. Idri, F.A. Amazal, A. Abran, Analogy-based software development effort estimation: a systematic mapping and review, Inf. Softw. Technol. 58 (2015) 206–230, http://dx.doi.org/10.1016/j.infsof.2014.07.013.
[34] M.O. Elish, T. Helmy, M.I. Hussain, Empirical study of homogeneous and heterogeneous ensemble models for software development effort estimation, Math. Prob. Eng. 2013 (2013), http://dx.doi.org/10.1155/2013/312067.
[35] M.J. Shepperd, G. Kadoda, Comparing software prediction techniques using simulation, IEEE Trans. Softw. Eng. 27 (2001) 1014–1022, http://dx.doi.org/10.1109/32.965341.
[36] E. Kocaguneli, T. Menzies, J.W. Keung, On the value of ensemble effort estimation, IEEE Trans. Softw. Eng. 38 (2012) 1403–1416, http://dx.doi.org/10.1109/TSE.2011.111.
[37] M.O. Elish, Assessment of voting ensemble for estimating software development effort, in: IEEE Symp Comput. Intell. Data Min., Singapore, 2013, pp. 316–321, http://dx.doi.org/10.1109/CIDM.2013.6597253.
[38] Y. Kultur, B. Turhan, A. Bener, Ensemble of neural networks with associative memory (ENNA) for estimating software development costs, Knowl. Based Syst. 22 (2009) 395–402, http://dx.doi.org/10.1016/j.knosys.2009.05.001.
[39] G. Seni, J.F. Elder, Ensemble Methods in Data Mining: Improving Accuracy Through Combining Predictions (2010), http://dx.doi.org/10.2200/S00240ED1V01Y200912DMK002.
[40] L. Breiman, Bagging predictors, Mach. Learn. 26 (1996) 123–140, http://dx.doi.org/10.1023/A:1018054314350.
[41] Y. Liu, X. Yao, Ensemble learning via negative correlation, Neural Netw. 12 (1999) 1399–1404, http://dx.doi.org/10.1016/S0893-6080(99)00073-8.
[42] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten, The WEKA data mining software, ACM SIGKDD Explor. 11 (2009) 10–18, http://dx.doi.org/10.1145/1656274.1656278.
[43] P. Braga, A. Oliveira, G. Ribeiro, S. Meira, Bagging predictors for estimation of software project effort, Proc Int. Jt. Conf. Neural Networks (2007) 14–19, http://dx.doi.org/10.1109/IJCNN.2007.4371196.
[44] M. Azzeh, A.B. Nassif, L.L. Minku, An empirical evaluation of ensemble adjustment methods for analogy-based effort estimation, J. Syst. Softw. 103 (2015) 36–52, http://dx.doi.org/10.1016/j.jss.2015.01.028.
[45] D. Wu, J. Li, Y. Liang, Linear combination of multiple case-based reasoning with optimized weight for software effort estimation, J. Supercomput. 64 (2013) 898–918, http://dx.doi.org/10.1007/s11227-010-0525-9.
[46] L.L. Minku, X. Yao, locality. Ensembles, Insight on improving software effort estimation, Inf. Softw. Technol. 55 (2013) 1512–1528, http://dx.doi.org/10.1016/j.infsof.2012.09.012.
[47] D. Azhar, P. Riddle, E. Mendes, N. Mittas, L. Angelis, Using ensembles for web effort estimation, in: 2013 ACM/IEEE int, Symp. Empir. Softw. Eng. Meas. 2013 (2016) 173–182, http://dx.doi.org/10.1109/ESEM.2013.25.
[48] A. Chandra, X. Yao, Ensemble learning using multi-Objective evolutionary algorithms, J. Math. Model. Algorithms 5 (2006) 417–445, http://dx.doi.org/10.1007/s10852-005-9020-3.
[49] A. Idri, M. Hosni, A. Abran, Systematic literature review of ensemble effort estimation, J. Syst. Softw. 118 (2016) 151–175, http://dx.doi.org/10.1016/j.jss.2016.05.016.
[50] N. Mittas, M. Athanasiades, L. Angelis, Improving analogy-based software cost estimation by a resampling method, Inf. Softw. Technol. 50 (2008) 221–230, http://dx.doi.org/10.1016/j.infsof.2007.04.001.
[51] C.-J. Hsu, N.U. Rodas, C.-Y. Huang, K.-L. Peng, A study of improving the accuracy of software effort estimation using linearly weighted combinations, in: Proc 34th IEEE Annu. Comput. Softw. Appl. Conf. Work, Seoul, 2010, pp. 98–103, http://dx.doi.org/10.1109/COMPSACW.2010.27.
[52] E. Kocaguneli, Y. Kultur, A.B. Bener, Combining multiple learners induced on multiple datasets for software effort prediction, Proc Int. Symp. Softw. Reliab. Eng. (2009) http://www.issre2009. org/papers/issre2009_245.pdf.
[53] M. Shepperd, S. MacDonell, Evaluating prediction systems in software project estimation, Inf. Softw. Technol. 54 (2012) 820–827, http://dx.doi.org/10.1016/j.infsof.2011.12.008.
[54] T.M. Khoshgoftaar, P. Rebours, N. Seliya, Software quality analysis by combining multiple projects and learners, Softw. Qual. J. 17 (2009) 25–49, http://dx.doi.org/10.1007/s11219-008-9058-3.
[55] J. Keung, Software development cost estimation using analogy: a review, in: Proc. 20th Aust. Softw. Eng. Conf., Gold Coast, QLD, 2009, pp. 327–336, http://dx.doi.org/10.1109/ASWEC.2009.32.
[56] M. Azzeh, D. Neagu, P.I. Cowling, Analogy-based software effort estimation using Fuzzy numbers, J. Syst. Softw. 84 (2011) 270–284, http://dx.doi.org/10.1016/j.jss.2010.09.028.
[57] M. Azzeh, D. Neagu, P. Cowling, Improving analogy software effort estimation using fuzzy feature subset selection algorithm, Proc 4th Int. Work. Predict. Model. Softw. Eng. — PROMISE' 08 (2008) 71–7178, http://dx.doi.org/10.1145/1370788.1370805.
[58] L. Angelis, I. Stamelos, A simulation tool for efficient analogy based cost estimation, Empir. Softw. Eng. 5 (2000) 35–68, http://dx.doi.org/10.1023/A:1009897800559.
[59] G.K. Michelle, M. Cartwright, L. Chen, M.J. Shepperd, Experiences Using Case-Based Reasoning to Predict Software Project Effort, Proc 4th Int. Conf. Empir. Assess. Eval. Softw. Eng. (2000) 1–22 (http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.34.6648; http://dec.bournemouth.ac.uk/ESERG/Technical_Reports/TR00-01/TR00-01.pdf).
[60] M.-Á. Sicilia, J.-J. Cuadrado-Gallego, J. Crespo, E. García-Bariocanal, Software cost estimation with fuzzy inputs: fuzzy modelling and aggregation of cost drivers, Kybernetika 41 (2005) 249–264.
[61] A. Idri, A. Zahi, A. Abran, Generating fuzzy term sets for software project attributes using fuzzy C-means and real coded genetic algorithms, Inf. Commun. Technol. Muslim World (2005).
[62] J.C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, New York, 1981. 10.1007/978-1-4757-0450-1.
[63] J.C. Dunn, A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters, J. Cybern. 3 (1973) 32–57, http://dx.doi.org/10.1080/01969727308546046.
[64] F. Herrera, M. Lozano, A.M. Sánchez, A taxonomy for the crossover operator for real-coded genetic algorithms: an experimental study, Int. J. Intell Syst. 18 (2003) 309–338, http://dx.doi.org/10.1002/int.10091.
[65] D. Mühlenbein, H. Schlierkamp-Voosen, Predictive models for the breeder genetic algorithm I. continuous parameter optimization, Evol. Comput. 1 (1993) 25–49.
[66] A. Idri, A. Abran, A fuzzy logic based set of measures for software project similarity: validation and possible improvements, in: Proc Seventh Int. Softw. Metrics Symp., London, 2001, pp. 85–96, http://dx.doi.org/10.1109/METRIC.2001.915518.
[67] N.-H. Chiu, S.-J. Huang, The adjusted analogy-based software effort estimation based on similarity distances, J. Syst. Softw. 80 (2007) 628–640, http://dx.doi.org/10.1016/j.jss.2006.06.006.
[68] I. Stamelos, L. Angelis, M. Morisio, E. Sakellaris, G.L. Bleris, Estimating the development cost of custom software, Inf. Manage. 40 (2003) 729–741, http://dx.doi.org/10.1016/S0378-7206(02)00099-X.

Please cite this article in press as: A. Idri, et al., Improved estimation of software development effort using Classical and Fuzzy Analogy ensembles, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.08.012

[69] S. Cha, Comprehensive survey on Distance/Similarity measures between probability density functions, Int. J. Math. Model. Methods Appl. Sci. 1 (2007) 300–307.
[70] M. Azzeh, Y. Elsheikg, Learning best K analogies from data distribution for case-Based software effort estimation, Seventh Int. Conf. Softw. Eng. Adv. (2012) 341–347 http://www.thinkmind.org/index.php?view=article&articleid=icsea_2012_12_40_10085.
[71] F.A. Amazal, A. Idri, A. Abran, Software development effort estimation using classical and fuzzy analogy: a cross-Validation comparative study, int, Int. J. Comput. Intell. Appl. 13 (2014) 1450013, http://dx.doi.org/10.1142/S1469026814500138.
[72] I. Ozkan, I.B. Turksen, Upper and lower values for the level of fuzziness in FCM, Inf. Sci. (Ny) 177 (2007) 5143–5152, http://dx.doi.org/10.1016/j.ins.2007.06.028.
[73] H. Choe, J.B. Jordan, On the optimal choice of parameters in a fuzzy c-means algorithm, in: IEEE Int. Conf. Fuzzy Syst., San Diego, CA, 1992, pp. 349–354, http://dx.doi.org/10.1109/FUZZY.1992.258640.
[74] K. Zhou, C. Fu, S. Yang, Fuzziness parameter selection in fuzzy c-means: the perspective of cluster validation, Sci. China Inf. Sci. 57 (2014) 1–8, http://dx.doi.org/10.1007/s11432-014-5146-0.
[75] P. Agarwal, M.A. Alam, R. Biswas, Issues challenges tools of clustering algorithms, Int. J. Comput. Sci. Issues 8 (2011) 523–528.
[76] P. Berkhin, A survey of clustering data mining, in: group, Multidimens. Data 2006 (2016) 25–71, http://dx.doi.org/10.1007/3-540-28349-8_2.
[77] I. Myrtveit, E. Stensrud, M. Shepperd, Reliability and validity in comparative studies of software prediction models, IEEE Trans. Softw. Eng. 31 (2005) 380–391, http://dx.doi.org/10.1109/TSE.2005.58.
[78] T. Foss, E. Stensrud, B. Kitchenham, I. Myrtveit, A simulation study of the model evaluation criterion MMRE, IEEE Trans. Softw. Eng. 29 (2003) 985–995, http://dx.doi.org/10.1109/TSE.2003.1245300.
[79] Y. Miyazaki, Method to estimate parameter values in software prediction models, Inf. Softw. Technol. 33 (1991) 239–243, http://dx.doi.org/10.1016/0950-5849(91)90139-3.
[80] L.L. Minku, X. Yao, An analysis of multi-objective evolutionary algorithms for training ensemble models based on different performance measures in software effort estimation, Proc 9th Int. Conf. Predict. Model. Softw. Eng. – PROMISE ' 13 (2013) 1–1-10, http://dx.doi.org/10.1145/2499393.2499396.
[81] J. Cohen, A power prime, Psychol. Bull. 112 (1992) 155–159, http://dx.doi.org/10.1037/0033-2909.112.1.155.
[82] A.M.H. Quenouille, Notes on bias in estimation, Biometrika 43 (1956) 353–360 (10.1093%2Fbiomet%2F43.3-4.353).
[83] A.J. Scott, M. Knott, A cluster analysis method for grouping means in the analysis of variance, Biometrics 30 (1974) 507–512 http://www.jstor.org/stable/2529204.
[84] I.T. Jollife, Cluster analysis as multiple comparison method, Appl. Stat. (1975) 159–168.
[85] D.R. Cox, E. Spjøtvoll, On partitioning means into groups, Scand. J. Stat. 9 (1982) 147–152 http://www.jstor.org/stable/4615870.
[86] T. Calinski, L.C.A. Corsten, Clustering means in ANOVA by simultanous testing, Biometrics 41 (1985) 39–48 http://www.jstor.org/stable/2530641.
[87] S. Bony, N. Pichon, C. Ravel, A. Durix, F. Balfourier, J. Guillaumin, The relationship between mycotoxin synthesis and isolate morphology in fungal endophytes of Lolium perenne, New Phytol. 152 (2001) 125–137, http://dx.doi.org/10.1046/J.0028-646x.2001.00231.X.
[88] D. a. Bisognin, D.S. Douches, K. Jastrzebski, W.W. Kirk, Half-sib progeny evaluation and selection of potatoes resistant to the US8 genotype of Phytophthora infestans from crosses between resistant and susceptible parents, Euphytica 125 (2002) 129–138, http://dx.doi.org/10.1023/A:1015763207980.
[89] J. Sharma, L.W. Zettler, J.W. Van Sambeek, M.R. Ellersieck, C.J. Starbuck, Symbiotic seed germination and mycorrhizae of federally threatened platanthera praeclara (Orchidaceae), Am. Midl. Nat. 149 (2003) 104–120, http://dx.doi.org/10.1674/0003-0031(2003)149[0104:SSGAMO]2.0.CO;2.
[90] L. Borges, D. Ferreira, I. Power type, errors rate of Scott-Knott, Tukey and Newman-Keuls tests under normal and no-normal distributions of the residues, Rev. Matemática E Estatística. 21 (2003) 67–83.
[91] G. Tsoumakas, L. Angelis, I. Vlahavas, Selective fusion of heterogeneous classifiers, Intell. Data Anal. 9 (2005) 511–525.
[92] N. Mittas, L. Angelis, Ranking and clustering software cost estimation models through a multiple comparisons algorithm, IEEE Trans. Softw. Eng. 39 (2013) 537–551, http://dx.doi.org/10.1109/TSE.2012.45.
[93] N. Mittas, I. Mamalikidis, L. Angelis, A framework for comparing multiple cost estimation methods using an automated visualization toolkit, Inf. Softw. Technol. 57 (2015) 310–328, http://dx.doi.org/10.1016/j.infsof.2014.05.010.
[94] H.W. Lilliefors, On the Kolmogorov-Smirnov test for normality with mean and variance unknown, J. Am. Stat. Assoc. 62 (1967) 399–402, http://dx.doi.org/10.1080/01621459.1967.10482916.
[95] G.E.P. Box, D.R. Cox, An analysis of transformations, J. R. Stat. Soc. 26 (1964) 211–252 http://www.jstor.org/stable/2984418.
[96] T. Menzies, B. Caglayan, E. Kocaguneli, J. Krall, F. Peters, B. Turhan, The promise repository of empirical software engineering data terapromise.csc.ncsu.edu. 2012.
[97] J. Deharnais, Analyse Statistique De La Productivitie Des Projects De Development En Informatique Apartir De La Technique Des Points Des Fontion, Quebec university, 1989.
[98] C.F. Kemerer, An empirical validation of software cost estimation models, Commun. ACM 30 (1987) 416–429, http://dx.doi.org/10.1145/22899.22906.
[99] Y. Miyazaki, M. Terakado, K. Ozaki, Robust regression for developing software estimation models, J. Syst. Softw. 27 (1994) 3–16, http://dx.doi.org/10.1016/0164-1212(94)90110-4.
[100] C. Lokan, T. Wright, P. Hill, M. Stringer, Organizational benchmarking using the ISBSG data repository, IEEE Softw. 18 (2001) 26–32, http://dx.doi.org/10.1109/52.951491.
[101] Y.F. Li, M. Xie, T.N. Goh, A study of project selection and feature weighting for analogy based software cost estimation, J. Syst. Softw. 82 (2009) 241–252, http://dx.doi.org/10.1016/j.jss.2008.06.001.
[102] D. Milios, I. Stamelos, C. Chatzibagias, Global optimization of analogy-Based software, Proc EANN/AIAI 2011 (2011) 350–359.
[103] J. Wen, S. Li, L. Tang, Improve analogy-based software effort estimation using principal components analysis and correlation weighting, Proc. Asia Pac. Softw. Eng. Conf. APSEC (2009) 179–186, http://dx.doi.org/10.1109/APSEC.2009.40.
[104] B.M. Byrne, Structural Equation Modeling with AMOS, New York, 10.4324/9781410600219, 2009.
[105] A. Idri, A. Abran, L. Kjiri, COCOMO cost model using fuzzy logic, in: Proc. 7th Int. Conf. Fuzzy Theory Tech., Atlantic New Jersy, 2000, pp. 1–4.
[106] The R Project for Statistical Computing, (n.d.). https://www.r-project.org/.
[107] E. Kocaguneli, T. Menzies, Software effort models should be assessed via leave-one-out validation, J. Syst. Softw. 86 (2013) 1879–1890, http://dx.doi.org/10.1016/j.jss.2013.02.053.
[108] V. Vinaykumar, M.C.K. Ravi, Software cost estimation using soft computing approaches, in: Handbook of Research on Machine Learning Applications and Trends (Ed.) Handb. Res. Mach. Learn. Appl. Trends, IGI-global, 2009, pp. 499–518, http://dx.doi.org/10.4018/978-1-60566-766-9.

Please cite this article in press as: A. Idri, et al., Improved estimation of software development effort using Classical and Fuzzy Analogy ensembles, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.08.012