Alternative methods using similarities in software effort estimation

Makrina Viola Kosti  
Department of Informatics  
Aristotle University  
54124 Thessaloniki, Greece  
+302310991930  
mkosti@csd.auth.gr

Nikolaos Mittas  
Department of Informatics  
Aristotle University  
54124 Thessaloniki, Greece  
+302310991930  
nmittas@csd.auth.gr

Lefteris Angelis  
Department of Informatics  
Aristotle University  
54124 Thessaloniki, Greece  
+302310998230  
lef@csd.auth.gr

ABSTRACT
A large variety of methods has been proposed in the literature about Software Cost Estimation, in order to increase accuracy when predicting the effort of developing new projects. Estimation by Analogy is one of the most studied techniques in this area the last 20 years. The popularity of the methodology can be explained by its accordance to human problem thinking and solving, the straightforward interpretation and the usually comparable accuracy to other methodologies. Furthermore, the methodology is essentially a special case of non-parametric regression, easily implementable and free of theoretical assumptions, based on the notion of “similarity” which is used to define “neighbors”. All of these reasons led us to study the technique in more depth, considering alternative ways to exploit similarities, in order to assign weights to neighbors. In this paper, our aim is to review the existing weighting practices and explore some new iterative procedures from matrix algebra, which transform a similarity matrix to a bi-stochastic matrix (a matrix with row and column summing to 1). Specifically, we apply algorithms such as the Sinkhorn–Knopp and the Bregmanian Bi-Stochastication to similarity matrices of well-known software cost datasets in order to derive matrices that assign weights to the neighbors used for effort estimates. We investigate the sensitivity of the results with respect to the similarity function, focusing on a Gaussian kernel matrix with a tuning parameter. The promising results show that the new methods deserve a more thorough investigation and can be considered as generalization of the Estimation by Analogy method.

Categories and Subject Descriptors
D.2.9 [Software Engineering]: Management – cost estimation

General Terms
Algorithms, Management, Measurement, Performance, Design, Experimentation

Keywords
Software effort estimation, Estimation by Analogy, iterative algorithms, Bregmanian bi-stochastication, Sinkhorn – Knopp.

1. INTRODUCTION
The last few years we observe a continuous technological growth, straightforwardly reflected in the importance software has taken in our everyday life. As human activity becomes more and more complicated, the need of even more exigent systems seems vital. A demanding system, however, has challenging requirements regarding the development procedure. A project manager is the one who has to deal with the issue in many perspectives, either has it to do with risk management matters, i.e. defect prediction, or effort and time requirements. Effort requirements in means of effort estimation is a task known as Software Cost Estimation (SCE), which is essential in the early stages of the development and may be performed in any stage of the life cycle of the under estimation project. To cope with this important task, a plethora of methods has been proposed during the previous decades [1]. These estimation methods range from expert judgment to statistical models and complicated machine learning algorithms. In expert judgment the calculation of effort is based on human experience and reasoning but due to its instinctive nature, the method is difficult to be fully analyzed ([2], [3], [4], [5]). Statistical models, like COCOMO [6] and Function Points [7], require the application of a cost model, which is expressed in the form of mathematical equations, estimated through statistical data analysis. Machine learning techniques are trained on data and their principles are also based on statistics. The latter, despite the fact that usually offer intelligent algorithms for optimizing prediction, they often work as “black boxes” [8] with difficult interpretation of their results.

One of the most popular and intuitively appealing techniques for predicting the cost of a new project is Estimation by Analogy (EbA), a form of Case Based Reasoning (CBR) ([9], [10], [11]) which can be considered as a systematic and empirical approach, closely related to the expert judgment process. Main aspect of the technique is to find among historical projects the ones which are most similar to a new one. The effort of these “most similar” projects is the basis for the estimation of the new project.

To understand EbA in a nutshell, we can portray it as a procedure that follows three main steps: First, the project to be estimated is characterized by a set of features, common to the ones describing the projects in the historical database. Second, one or more projects from the historical dataset are found according to a similarity criterion. Third, the cost values of the selected projects (usually referred to as neighbors or analogies) are combined to produce the final estimate. As discussed by Shepperd and Schofield [10], there are certain advantages that appoint EbA more attractive in respect with other methodologies, since it is free of distribution assumptions and it is easily applied to all types of data, either numerical or categorical. Furthermore, EbA can be considered as a special case of non-parametric regression ([12], [13]). Other empirical studies [14] show that the predictive power
of EbA is significantly comparable and sometimes superior to the classical parametric regression models. In fact, users are more willing to accept solutions from analogy based systems since they are congruent to human problem thinking and solving, in direct opposition with the awkward chains of complicated “black box” procedures [8].

A characteristic of EbA is that it treats all neighbors in the same manner, i.e. it only chooses a number of neighbors and estimates the effort of the new project as the average of their efforts. The facility of application and comprehension of the method stimulates further investigation of one of its aspects; the utilization of similarities in order to assign weights to neighbors. This can be achieved by simple arithmetic methods (e.g. [15]) which use elements of a similarity matrix. However, in mathematics, and specifically in the field of linear algebra, we can find techniques which can broaden this approach by transforming the similarity matrix as a whole. These can reveal better comprehension of the methods based on similarities or even introduce prospects of possible improvements. Towards this direction, we examine bi-stochastication techniques as a possible way to translate similarities into weights. The output of this translation is a bi-stochastic matrix. Specifically, the transformation of a similarity matrix to a bi-stochastic matrix that maintains the neighbor structure is achieved by applying two iterative procedures from matrix algebra: the Sinkhorn–Knopp (SK) [16] and the Bregmanian Bi-Stochastication (BBS) [17] algorithms. In this paper, we explore the possibility of using these bi-stochastication methods to effort estimation and we especially explore the properties of the resulting matrices and their effect on prediction. Their application on two datasets indicated promising results but also the need for future and in-depth investigation.

The rest of the paper is structured as follows: In Section 2, related work to the weighting methodologies is presented. In Section 3, we introduce the theoretical and practical principles of the new methodologies. Sections 4 and 5 present the datasets the algorithms that applied to and the details of the application procedure. Also, a thorough presentation of results is performed, in order to provide a better understanding of the methods. Finally, in Section 6 we conclude this investigative paper, summarizing the prediction results and discussing future work issues.

2. RELATED WORK

There is a variety of approaches regarding EbA extensions and improvements in the literature [18]. A considerable amount of research work is driven from the idea of weighting either project features or analogies, having as final scope the determination of the most contributing ones (either features or analogies) for optimizing estimation accuracy.

Considering the problem of weighting analogies, which is the main problem addressed in the current study, there have been some adaptation techniques [20]. The techniques presented by Azzeh [20] are in accordance to the methods proposed in our work and try to overcome the hypothesis of equal influence of analogies to the final outcome (cost estimation). A related approach is the inverse rank weighted mean ([15], [21]) which gives to closest analogies higher weights, so they can influence the estimation more than the others.

In his assessment, Azzeh [20] also presents a variety of adjustment categories, according to the procedure each of them follows. The Linear Similarity Adjustment category includes types of adaptation techniques, which adjust retrieved analogous efforts. The adjustment is based on local or global similarity degrees of the historical projects with the target project. Local similarities can be calculated based on one of the project features, while the global is the aggregated average of each local. This kind of adjustment is reflected in Eq. 1.

$$\text{Effort}(p_i) = \sum_{j=1}^{k} \left[ \text{Sim}(p_j, p_i) \times \text{Effort}(p_j) \right] / \sum_{j=1}^{k} \text{Sim}(p_j, p_i)$$

where $\text{Sim}$ is the calculated global similarity between $p_i$ and $p_j$ (target project), and $k$ is the number of analogies. The adjustments based on the abovementioned idea have been used in estimation models such as AQUA [22] and Fuzzy Grey Relational Analysis (FGRA) [23]. This category also contains another adaptation strategy which is based on heuristic searching methods, such as Genetic Algorithms (GA). Chiu and Huang [24], in particular, examined the application of GA to optimize the coefficients $a_j$ (see Eq. 2) for each feature dissimilarity.

$$\text{Effort}(p_i) = \text{Effort}(p_i) + \sum_{j=1}^{k} a_j \times (f_j - f_{j,i})$$

where $f_j$ is the $j^{th}$ feature value of the target project and $f_{j,i}$ is the $j^{th}$ feature value of the nearest analogy.

Furthermore, various studies were based on Linear Size Feature Inference [20] to adjust effort estimation. These studies initiated depending on several research work that showed significant strong correlation between size and effort ([25], [26], [27]). More specifically, Walkerden and Jeffery [25] developed a function which used the nearest neighbor for estimation. The cost of the closest analogy was adjusted using linear extrapolation of the function points attribute (see Eq. 3).

$$\text{Effort}(p_i) = \frac{\text{Effort}(p_i)}{FP(p_i)} \times FP(p_j)$$

where $FP(p_i)$ and $FP(p_j)$ are the sizes of the nearest analogy and the target case correspondingly, measured in Function Points (FP). The fact that the size of the project has to be measured in FP’s, was a restriction to the method and led to another linear size adaptation proposed by Mendes et al. [21] applied to web datasets. Its form is depicted in Eq. 4.

$$\text{Effort}(p_i) = \frac{1}{k} \sum_{j=1}^{k} \frac{1}{m} \sum_{j=1}^{m} f_j \times \text{Effort}(p_i)$$

where $k$ is the number of selected analogies, $m$ the number of size features describing each project and $f_j$ is the $j^{th}$ feature value of historical project $i$.

In addition, Azzeh [20] refers to another two categories, namely Linear Productivity Adjustment (LPA) and Non-linear Distance Adjustment (NLDA). Regarding LPA, Jorgensen et al. [27] explored the use of Regression Towards the Mean (RTM) to adjust effort estimation with the adjusted productivity of the target project and the productivity of the closest analogies. This approach can be considered as an extension of Walkerden and Jeffery [25]. The idea is depicted in Eq.5.

$$\text{Effort}(p_i) = FP(p_i) \times \left[ \text{PR}(p_i) + (M - \text{PR}(p_i)) \times (1 - r) \right]$$
where \( PR(p_a) = \frac{\text{Effort}(p_a)}{FP(p_a)} \), is the productivity of the closest analogy, \( M \) is the average productivity of the similar projects and \( r \) is the historical correlation between the non-adjusted analogy based productivity and the actual productivity. This approach, taking also into account a replicated study by Shepperd and Cartwright [28], did not result in particular increase of predictive accuracy.

To conclude this literature review, Li et al. [19], underlining the non-normal distribution the historical datasets follow, suggest that applying linear techniques might not be an optimal solution. Consequently, they proposed in their study a model based on Neural Networks (NN) (see, Eq. 6),

\[
Effort(p_i) = Effort(p_{\text{a}}) + f(S_i, S_{\text{a}})
\]

where \( f(S_i, S_{\text{a}}) \) is the output of the NN model, \( S_i \) is the feature vector of the target project and \( S_{\text{a}} \) is the feature matrix of the top analogies.

In this paper, we want to extend the research regarding weighting analogies by presenting new methods which are based on the appropriate transformation of the similarity matrix. The methods we investigate here are different from the ones presented in the literature so far. The basic idea is that once the similarity matrix is transformed to a bi-stochastic matrix, its elements are ready to be used for weighting all projects in the dataset, without the need to determine how many and which analogies (neighbors) to use for each one of the projects. Therefore, the methods we propose here provide a solid, non-parametric model which is fitted to the available dataset.

Our first goal is to explain and familiarize readers with the general idea of bi-stochastic matrices, which can be potentially applied to provide neighbors with different impact degrees for the final calculation of SCE. Therefore, a detailed presentation of the algorithms presented in the Introduction, is given in Section 3.

3. THEORETICAL APPROACH

We begin our description of the methods by presenting EbA as a result of a weighting estimation technique. Informally, EbA can be easily portrayed as a procedure conducted in three steps: (1) Characterization of the target project with attributes same to the historical ones, (2) calculation of similarities between the target and the historical projects and (3) choice of a number of the most similar historical project to the target and combination of their values to produce estimation.

However, Mittas et al. [13] showed that EbA can be also expressed in the form of a mathematical model. In the statistical literature, the method is known as nearest neighbor non-parametric regression [12] and can be considered as an alternative choice to the more traditional parametric regression models. The k-nearest neighbor (k-NN) estimate is a weighted mean in a neighborhood. According to the same paper [13], given a dataset with \( n \) observations (projects), a k-NN estimate of the unknown effort function \( f(X_i) + \varepsilon \) can be evaluated by Eq. 7,

\[
f(X_i) = \sum_{j=1}^{n} W_{ij}(X_j) Y_j
\]

where \( \{W_{ij}(X_j)\}_n \) is a sequence of weights defined by the set of indices \( S_{\text{seq}} = \{j : X_j \text{ is one of the } k \text{ nearest neighbors of } X_i\} \). The simplest k-NN sequence of weights is constructed by Eq. 8, which gives to the selected neighbors equal weights, all summing to 1:

\[
W_{ij}(X_j) = \begin{cases} 1/k & \text{if } j \in S_{\text{seq}} \\ 0 & \text{otherwise} \end{cases}
\]

This assumption implies that each one of the selected neighbors withholds equally important information and has the same impact on the SCE procedure. Nevertheless, this is a simplified assumption which can be extended to a more realistic and generic concept, i.e. the weighted contribution of analogies according to their similarities. Indeed, the idea behind our approach is that the higher the similarity of a historical project to a target case, the higher is the probability to address this project in order to retrieve information from it. According to this idea, similarity can be interpreted as a probability of directing towards a neighbor, which in turn can be used as a weight in a mathematical formula. The mathematical background for such an interpretation is provided by the theory of stochastic matrices and Markov processes [29].

The problem is therefore to approximate the structure of a similarity matrix by a bi-stochastic matrix, which keeps the symmetric structure and where each element represents a transition probability so that its rows and columns are summed to 1. The elements of the resulting matrix can be used directly as weights in Eq. 7 in order to estimate the effort of any project by the convex combination of all other efforts. In Section 3.2 we present two algorithms that can convert similarity matrices to bi-stochastic ones: the Sinkhorn–Knopp and the Bregman bi-stochastication procedures.

3.1 Calculation of similarity matrices

Before explaining the two algorithms, it is necessary to describe the procedure we follow for calculating a similarity matrix.

Let \( X_{\text{proj}} \) be a data matrix, which comprises \( n \) historical projects \{\( P_1, \ldots, P_n \)\}, each of them having \( p \) attributes. First, we calculate the \( n \times n \) distance matrix \( D \), which contains the pairwise distances of the historical projects. As historical datasets contain various types of variables, we have to use a distance metric that takes into account the mixed-type of data. An important procedure is the standardization (between 0 and 1) of each dimension so that every attribute has the same degree of influence and the method is immune to the choice of units.

Hence, we used a special dissimilarity coefficient suggested by Kaufman and Rousseew [30]. The distance or dissimilarity of projects \( P_i \) and \( P_j \) is computed by their vectors of attributes \( X_i = (x_{i1}, \ldots, x_{ip}) \) and \( X_j = (x_{j1}, \ldots, x_{jp}) \) respectively, and is given by the following expressions:

\[
d(i,j) = \sum_{m=1}^{p} \delta_{i,j}^{(m)} d_{i,j}^{(m)} \sum_{m=1}^{p} \delta_{i,j}^{(m)}
\]

where:

\[
\delta_{i,j}^{(m)} = \begin{cases} 1 & \text{if } x_{im}, x_{jm} \text{ nonmissing} \\ 0 & \text{otherwise} \end{cases}
\]

If the \( m \)th variable is binary or nominal,
\[ d_{ij}^{(m)} = \begin{cases} \frac{1}{x_{im} - x_{jm}} & \text{if nonmissing} \\ 0 & \text{if } x_{im} = x_{jm} \end{cases} \quad (11) \]

If the \( m^{th} \) variable is of interval or ratio scale:

\[ d_{ij}^{(m)} = \frac{1}{R_m} \quad (12) \]

where \( R_m = \max(x_{im}) - \min(x_{im}) \) is the range of the variable.

Finally, if the \( m^{th} \) variable is ordinal:

- The values \( x_{im} \) are replaced by their ranks \( r_{im} \in \{1, \ldots, M_m\} \).
- The ranks are transformed to values in the interval \([0,1]\):
  \[ z_{im} = \left( r_{im} - 1 \right) / (M_m - 1) \]
- The \( z_{im} \) values are treated as interval-scaled.

Next, the dissimilarity matrix \( D \) is converted to a similarity matrix \( K \) which will be subsequently used as an input to the bi-stochastic algorithms. The conversion from dissimilarity to similarity is achieved in our paper by taking the Gaussian kernel matrix with a tuning parameter \( \gamma \) (gamma) [17] (Eq. 13).

\[ K = \exp\left( -\frac{1}{\gamma} D \right) \quad (13) \]

In Section 3.2, we describe two approaches for gaining a stochastic matrix from \( K_{nn} \).

### 3.2 The Sinkhorn–Knopp and the Bregman bi-stochastic algorithms

Sinkhorn and Knopp [31] introduced a method for converting square \((n \times n)\) nonnegative matrices to bi-stochastic matrices. The general idea is that for every non-negative symmetric matrix of order \( n \), like our similarity matrix \( K \), there exists a diagonal matrix \( H = \text{diag}(z) \) such that the matrix \( G = HKH \) is bi-stochastic [32].

Specifically, the algorithm converts a nonnegative square matrix into a bi-stochastic matrix by alternately scaling the row sums and the column sums [16]. The algorithm places all the elements of the matrix into the interval \([0,1]\) while preserving the underlying similarity structure. Additionally, if the original matrix is symmetric, the new bi-stochastic matrix is also symmetric.

The Sinkhorn-Knopp (SK) algorithm performs a series of iterations, where a vector \( z \) is updated from the following recursive formula:

\[ z^{(t)} = \left( \text{diag}(Kz^{(t-1)}) \right)^{-1} I_n \quad (14) \]

where \( t \) is initially set to 1, \( z^{(1)} = KI_n \) and \( I_n \) is an \( n \times 1 \) column vector of ones. After a large number of iterations the algorithm converges to a stable vector \( z \) (see details in [16]) which in turn is used to compute \( H = \text{diag}(z) \). The resulting bi-stochastic matrix is given by \( G = HKH \).

Considering the second method, the so called Bregmanian bi-stochastic (BBS), Wang et al. [17] derived a bi-stochastic matrix \( G \) from an \( n \times n \) similarity matrix \( K \) by solving the optimization problem:

\[
\begin{aligned}
\min_{G} C_p(G,K) &= \sum_{ij} C_p(G_{ij},K_{ij}) \\
\text{s.t. } &G \geq 0, \ G = G^T, \ G_{ii} = 1
\end{aligned}
\]

where \( C_p(x,y) = \phi(x) - \phi(y) - \nabla \phi(y)(x-y) \) is known as the Bregman divergence between \( x \) and \( y \), and depends on a strictly convex function \( \phi \). The BBS algorithm proposed by Wang et al. [17] used the function \( \phi(x) = x^2/2 \). Note that by \( G^T \) we denote the transpose of matrix \( G \).

The algorithm is again an iterative procedure, initiated by setting \( t = 1 \) and \( G^{(1)} = K \). If we denote by \( I_n \) the \( n \times n \) identity matrix (a matrix with ones in its diagonal and zeros elsewhere) and by \( J_n \) the \( n \times n \) matrix of ones, then the following recursive formula (Eq. 16)

\[
G^{(t)} = \left[ G^{(t-1)} + \frac{1}{n} \left( I_n - G^{(t-1)} + \frac{J_nG^{(t-1)}}{n} \right) J_n - \frac{1}{n} J_nG^{(t-1)} \right]^+ \quad (16)
\]

converges after a large number of iterations to a bi-stochastic matrix \( G \). The \( +^\dagger \) symbol in Eq. 16 denotes the positive part of the matrix, i.e. only its positive elements.

In our implementation, we initiated both algorithms (SK and BBS), by a similarity matrix \( K \) which has the diagonal elements equal to zero. The resulting bi-stochastic matrix \( G \) has also zero diagonal elements. This is essential for achieving an unbiased estimation procedure, since it prevents the participation of the under estimation project in the whole process. Once we obtain the bi-stochastic matrix \( G \), either by SK or by BBS, we can directly calculate the cost estimation of each project from all the others.

Let \( Y_d \) be the \( n \times 1 \) vector representing the actual efforts \( y_{ij} \) of each project \( P_i, i = 1, \ldots, n \) and \( G \) the bi-stochastic matrix resulting from transforming the similarity matrix of the independent variables by SK or BBS. Then, the \( n \times 1 \) vector of effort estimations \( Y_{d'} \) is computed simply by Eq. 17:

\[ Y_{d'} = G Y_d \quad (17) \]

Therefore, since \( G \) has zero diagonal elements, each effort is estimated as the weighted mean of all other efforts. As we will see from the applications, the SK and BBS transformations put more weight on large similarities while they minimize or even eliminate the contribution of small similarities.

### 4. STATISTICAL METHODS USED IN APPLICATIONS

In this section, we present the statistical setup for the application and evaluation of the under investigation methods to real datasets. The SK and BBS-based methods described in the previous section were applied and compared to the simple EbA. We chose EbA since it is the most known technique using similarities, in order to have a reference point for comparisons.

The predictive accuracy was evaluated through the leave-one-out cross-validation (LOOCV) procedure, which estimates the cost of each project from all the others [33]. At each iteration, a project \( i \) with actual effort \( Y_{di} \) is removed from the dataset (test case) and
The remaining projects (training set) are the basis for the estimation of the effort value \( Y_{E_1} \).

The set of predictions \( Y_{E_1} \) obtained from LOOCV in contrast to the actual values \( Y_{A_1} \) can be evaluated by functions of “local error”. The thorough discussion on the advantages and limitations of error functions [34], shows that there is not a universal solution concerning the most appropriate one, in order to compare alternative models. Taking into account the literature review of SCE, we decided to utilize two error functions, which measure two different aspects of the prediction performance: (a) the magnitude of relative error (MRE) and (b) the absolute error (AE).

These errors values are in turn, the basis for the evaluation of a “global” indicator describing the overall predictive performance of a SCE model via a central tendency statistic. The mean value is a possible choice and the derived indicators have the acronyms MMRE and MAE, respectively. However, it is essential for meaningful comparisons to realize what each of the abovementioned functions of error really measures [35]. So, AE and MAE are used in order to measure the accuracy of models. On the other hand, MRE is the most known and used error function ([34], [35]) but there is also a severe criticism about the inability of MMRE to select the “best” model. Kitchenham et al. [35] show that MMRE describes the spread of the z-values obtained by dividing the estimated cost by the actual cost which is clearly related to the distribution of the residuals.

Finally, we also decided to use the Spearman correlation coefficient [36] between the actual and estimated values obtained from each prediction model. According to Mukhopadhyay et al. [9], a correlation coefficient can be used as a measure of the consistency of a method.

For a systematic and meaningful analysis, one should utilize formal statistical procedures in order to reinforce the inferential process and justify the selection of a prediction technique ([34], [37], [38]). A well-known parametric test suitable for comparisons of mean values is the paired-sample t-test which requires samples from normal distributions. However, a common problem encountered in the comparisons with statistical tests is that the error samples are highly-skewed with many outliers and far from being normal ([35], [37]).

In order to overcome the limitations of traditional parametric tests and obtain more reliable results, we decided to use a statistical simulation method, namely the permutation tests [39]. Moreover, we decided not to use a non-parametric test (e.g. the Wilcoxon signed rank test), since it is totally based on the ranks of the sample values, causing the loss of information that is hidden in ratio-scaled measurements, and tests whether there is a significant difference between the medians of two paired samples. Summarizing, the skewness of the error data and our objective (comparison of mean values) led us to make use of permutation tests for mean values.

Describing briefly, the method is based on a number of iterations, where the paired-samples of errors are permuted randomly [37]. At each iteration, the statistic under consideration (the difference of means between the paired-samples in our case) is computed. Finally, the statistical significance (or \( p \)-value) of the hypothesis test is evaluated through the placement of the original statistic on the permutation distribution. The latter is a desired property, since the significance of the test is computed directly from the original data without worrying about the distribution of the error functions. These resampling techniques seem to attract the interest of SCE community; while researchers point out that their usage presents certain benefits with significant potential value [38].

The comparison of the alternative methodologies with the permutation test was based on the distribution of AE values similarly to other studies ([40], [41]). In all tests, we consider as statistically significant a difference with \( p \)-value (significance) smaller than 0.05. All the tests conducted are two-tailed (non-directional) in the sense that the alternative hypothesis is that the measures tested are not equal.

Regarding the tuning of parameters, SK and BBS require the determination of the gamma (\( \gamma \)) parameter in Eq. 13 which however seems to affect the prediction accuracy of the final results. Indeed, we noticed from various experiments that the performance of the methods on different datasets depends on the choice of \( \gamma \). We therefore ran a sort of training experiments in order to decide the value of \( \gamma \) for each dataset. Specifically, we tried a range of values and we finally used for our comparisons the value that gave the smallest MMRE. Regarding EbA, the parameter that has to be decided is the number \( k \) of the nearest neighbors (or analogies). The right choice of \( k \) has been the subject of research (see indicatively, [10], [11], [21]), since there is no rule of thumb and someone has to calibrate it on the available data. Since in our comparisons we use EbA as a simple reference point, we decided to follow an approach similar to other studies ([20], [21]) and make use of few different \( k \) values ranging from 1 to 3.

Moreover, graphical comparison of the methods through boxplots is incorporated, so as to identify interesting properties of the distribution of errors [35]. Following the recommendations of Kitchenham et al. [35], visualization analysis is totally based on the residuals (or errors) that provide a manner to measure the bias accounting for underestimation or overestimation of a prediction system.

Finally since one of our goals is to study the properties of the matrices resulting from the bi-stochastication procedures we use a type of image representation tool for matrices, known as “heat maps” [42] (see Section 5.3) in order to visually compare the weights assigned by the SK and BBS algorithms and by the simple EbA.

5. APPLICATION TO DATASETS

In this section, we present the results of the conducted case studies in order to investigate the predictive potentials of the under investigation methodologies.

For the application of the introduced algorithms, we used two widely known datasets, namely: the Abran-Robillard (AR) and the Desharnais (DES) dataset. The AR dataset contains 20 projects [43] from a major Canadian financial organization with 10 independent continuous attributes and a dependent continuous variable (actual effort days). The second dataset, (DES dataset), contains 77 completed software projects from a Canadian Software house [44] with 4 independent cost predictors (3 of which are continuous and 1 is nominal) and a dependent variable (effort) measured in person-hours.
5.1 The Abran–Robillard dataset
The general results regarding the predictive performance of the comparative techniques are presented in Table 1. Concerning the SK algorithm, we can observe that it gives better results than the traditional EbA procedure for all $k$ – values. This is also true for the BBS model which produces again, noticeably superior results than the classical approach for all $k$ – values, since it presents the lowest global indicators of error (MMRE and MAE). The percentages inside parentheses in the EbA columns show the improvement in the corresponding measures achieved by the best prediction model (which is highlighted with bold fonts), compared to that of EbA. The improvements are generally remarkable ranging from 10.76% up to 44.32%.

The Spearman correlation coefficients in Table 2 clearly portray that all models achieve a strong and statistically significant positive correlation between the actual and the predicted values. The highest value of Spearman's rho is derived from BBS where $\rho = 0.892$.

The boxplots of the residual values (Fig. 1a) indicate that BBS has generally better prediction behavior than EbA. The box length (or inter-quartile range) and the whiskers of BBS are significantly smaller than those of other models. Also, the median value is very close to zero, meaning that the predictions are unbiased and not prone to under- or over-estimations.

The abovementioned findings indicate that there is an indication of improvement from using SK and BBS. In order to examine the significance of the accuracy improvement, we perform the permutation test for matched pairs (Table 3). Indeed, the tests reveal that there is a statistically significant difference between the mean values of AEs derived from both SK and BBS and the corresponding local errors derived from all EbA variants. Finally, we have also to observe that despite the small divergences between the measures of SK and BBS, the permutation test did not indicate a statistically significant difference.

5.2 The Desharnais dataset
The accuracy indicators (Table 1) illustrate that for the DES dataset, SK presents a slightly better predictive performance than EbA with small divergences for the case of MMRE, whereas the opposite is true for the MAE comparison. Concerning the MMRE and MAE indicators, the improvement ranges from 5.12% up to 12.67%.

The derived results concerning the consistencies (Spearman correlation coefficients) of SK and BBS algorithms are comparable. Both algorithms yield predictions that are positively and highly correlated with the actual values of effort with SK achieving the highest correlation coefficient ($\rho = 0.728$ in Table 2). In contrast, the variants of EbA produce less consistency in estimations, i.e. less correlated with the actual values of effort. As a final point, we have also to remark that the boxplots of residuals show that the variation of SK and BBS is lower than EbA variants (Fig. 1b).

The permutation tests (Table 3) show that only BBS has statistically significant difference from EbA for two out of three cases.

The conclusions from the second dataset are that both bi-stochastication methodologies give in general better results than EbA, however only BBS gives statistically significant improvements.

<table>
<thead>
<tr>
<th>Table 1. Predictive performance measures</th>
</tr>
</thead>
<tbody>
<tr>
<td>AR dataset</td>
</tr>
<tr>
<td>MMRE (improvement %)</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

| DES dataset                              |
| MMRE (improvement %)                     | 0.5518 | 0.5731 | 0.6250 (8.30%) | 0.6518 (12.07%) | 0.6040 (5.12%) |
| MAE                                       | 2256.28 | 2196.65 | 2510.52 (12.50%) | 2515.32 (12.67%) | 2420.81 (9.26%) |

<table>
<thead>
<tr>
<th>Table 2. Spearman correlation coefficient between actual and estimated effort</th>
</tr>
</thead>
<tbody>
<tr>
<td>SK</td>
</tr>
<tr>
<td>EbA ($k=1$)</td>
</tr>
<tr>
<td>EbA ($k=2$)</td>
</tr>
<tr>
<td>EbA ($k=3$)</td>
</tr>
<tr>
<td>AR</td>
</tr>
<tr>
<td>DES</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 3. Significance of the permutation tests for differences of MAE between methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>SK</td>
</tr>
<tr>
<td>EbA ($k=1$)</td>
</tr>
<tr>
<td>EbA ($k=2$)</td>
</tr>
<tr>
<td>EbA ($k=3$)</td>
</tr>
<tr>
<td>AR</td>
</tr>
<tr>
<td>DES</td>
</tr>
</tbody>
</table>
Figure 1. Boxplots of the residuals for (a) AR and (b) DES datasets

5.3 Properties of Bi-stochastic matrices

The results of the applications we presented in the previous section concern the predictive potentials of Eq. 17, which is based on bi-stochastic matrices that are computed in two different ways, either by the SK or the BBS algorithm. Although the methodology needs a thorough theoretical and practical investigation, the accuracy of predictions as shown in Tables 1, 2, and 3 seems encouraging. The algorithms are comparable to simple EbA and reduce the overall error while the improvement is in certain cases statistically significant. It is therefore necessary to explore the mechanism behind the methodology, regarding the properties of matrix $G$, in order to explain the quality of results.

Bi-stochastic matrices that are generated by the SK and BBS algorithms have the ability to preserve the neighborhood structure of the similarity matrix. This is achievable, since the resulting bi-stochastic matrix is essentially an approximation of the similarity matrix. The property is very important, since the neighborhood information is necessary for the estimation procedure. This reveals that the algorithms essentially "translate" that information by projecting each row of the similarity matrix from points in the real Euclidean space to points inside an $n$-simplex, i.e. an $n$-dimensional polytope. Furthermore, this translation has a particularly intuitive appealing interpretation in the case of effort estimation: if we want to estimate the effort of project $i$ then the $(i,j)$ element of matrix $G$ can be interpreted as the probability that project $j$ can provide correct estimation. In EbA this probability is the same for a set of chosen neighbors, but in our case it depends on similarities which are tuned all together. This probabilistic interpretation is related to the properties of bi-stochastic matrices which are known from Markov processes, where they are used as matrices of transition probabilities between states of a system [29].

It is important to note here that both algorithms, when they transform similarities to the $[0,1]$ interval, tend to assign very small values (practically rounded to zero) to small similarities (distant neighbors). Therefore, the algorithms essentially define the neighborhood of each project. Relatively to EbA, we found out that, in the majority of cases, bi-stochastication gives large weights to the same projects that EbA uses as analogies. Moreover, SK and BBS assign smaller weights to other projects that are not considered as nearest neighbors, but can be useful for estimation, contributing analogously to their similarity ranking. This means that bi-stochastication builds a neighborhood separately for each project under estimation, with different number of analogies, and therefore it can be considered as a technique for dynamic selection of analogies [18].

A nice graphical representation of the bi-stochastic matrices resulting from SK and BBS algorithms can be given by "heat maps", which represent the elements of a matrix as pixels shaded according to the corresponding values. So black pixels represent values zero or close to zero, while pixels with lighter shades, of grey or white, represent higher values. Figures 2a, 2b and 2c contain three heat maps for the case of the AR dataset and show how neighborhoods are formed respectively for the SK and BBS algorithms and for EbA (with $k=3$).

In order to highlight the dynamic adaptation of neighbors, suppose that we aim to predict the cost of the $1$st project (last row of the heat maps). The simple EbA approach with $k=3$ analogies indicates that the neighborhood of the $1$st project consists of three historical projects ($7\text{th}$, $4\text{th}$, and $2\text{nd}$) ordered according to the similarity with the new project. Hence, the estimated cost should be evaluated by these three projects through Eq. 7 and Eq. 8., where the traditional approach uses a sequence of equal weights $\{0.33, 0.33, 0.33\}$ for Eq. 8 summing to one. The limitation of this approach is that we do not exploit meaningful information concerning the degree of similarity, since a group of arbitrarily chosen neighbors contribute in the same manner to the final estimation.

On the contrary, the proposed algorithms build dynamic neighborhoods by assigning different weights to the closest projects, while they annihilate the contribution of furthest projects. For example, the BBS algorithm for the abovementioned $1$st project of the AR dataset reveals a neighborhood containing six past projects ($7\text{th}$, $4\text{th}$, $3\text{rd}$, $5\text{th}$ and $6\text{th}$) with $\{0.28, 0.26, 0.19, 0.13, 0.1, 0.03\}$ weights, respectively. Therefore, we observe that in this case, the algorithm not only preserves the ranking of the projects but also indicates that there are three more projects ($3\text{rd}$, $5\text{th}$ and $6\text{th}$) that should be taken into consideration for the evaluation of the final effort value.
Looking again at Fig 2a and Fig 2b, we can realize that the shading of pixels shows the communality of the corresponding project to the final estimation. In this example, the 7th project has the highest communality in the estimation of project 1 (white pixel), whereas the 6th has the lowest one (dark grey pixel). The communalities of the other projects are very close to zero so their pixels are black. Finally, we have also to notice that the size of the neighborhood (number of projects with high communality) is different for each specific project of the dataset, indicating a sophisticated manner for dynamic selection of analogies. Analogous results are derived by the SK bi-stochastication method. In general, both algorithms give higher weights to the most similar projects. However, their weighting mechanisms are not the same. We also have to point out that the neighborhood pattern we observe across the diagonal of the three heat maps is related to the sorting of projects according to their actual effort, i.e. project 1 has the smallest effort, and so on.

At this point it is essential to emphasize that bi-stochastication algorithms do not keep exactly the same ranking of each and every row of the original similarity matrix. Their mechanism is approximative, in the sense that they try to rebuild the similarity matrix subject to the desired property of rows and columns summing to one. Therefore, it is mathematically impossible to keep the ranking of all neighbors of each project 100% intact. Practically, SK and BBS operate in a holistic manner, trying to reform the entire similarity matrix and not each row separately. This is the main difference of the present methodologies with respect to all other weighting schemes reviewed in Section 2. After all, this is the concept behind the most known statistical methods and models. A rather trivial example of this concept is least squares regression analysis. It is impossible to have a line passing through all of the data points, but we can build a line that fits to our data by minimizing a certain global criterion measure, i.e. the sum of squared errors. So the concept is to treat all data points as a whole.

A final issue that has to be mentioned is the computational effort of applying the algorithms. Although the recursive formulas in Eq. 14 and Eq. 18 can be terminated by a convergence criterion (for example when the result of an iteration differs by a negligible amount from the result of the previous one), we found convenient to run the formulas in MATLAB for a large number of iterations (>10,000). For the relatively small datasets used in SCE and the available computational power, convergence was achieved in just
a few seconds. Therefore the algorithms and the estimation procedure are very easily implementable.

Concluding this section, we point out that the above mentioned holistic manner of treating similarities, the easiness of implementing the algorithms and the elegant estimation formula in Eq. 17 make the methodologies attractive for experimentation and further investigation.

6. DISCUSSION
In this paper we investigate the possibility of using similarities as weights for effort estimation by introducing bi-stochastication, which extends the notion of Estimation by Analogy. Our primary goal was to explore the applicability and predictability of two algorithms using well-known effort estimation datasets. The methods we describe, at first transform the similarity matrix of the projects to a bi-stochastic matrix, ready to be used for effort weighting. To the best of our knowledge, this is the first time such techniques are applied in the field of SCE and more generally in nearest neighbor estimation.

With EbA being one of the most studied techniques in this area, it was only natural to enrich it with the idea of stochastication, aiming further exploration of the principles of EbA regarding the use of similarities. Thereupon, we were based on two algorithms that can derive stochastic matrices, namely the Sinkhorn - Knopp (SK) and the Bregmanian Bi-Stochastication (BBS).

In order to examine the predictive power of our approaches, we experimented on two well-known datasets, the Abran-Robillard (AR) and the Desharnais (DES) dataset. The results were compared with EbA for \( k = 1,2,3 \) neighbors. Regarding the AR dataset, the use of SK and BBS for assigning weights to analogies, produced better and statistically significant estimation results than the simple EbA procedure for all three values of \( k \). In general, referring to the AR dataset, BBS gave the best accuracy measures (MMRE and AE). In relation to the DES dataset, the accuracy indicators showed us that SK performed slightly better, not only compared to BBS, but also compared to the traditional EbA for all three \( k \) values. These case studies allowed us to have an insight of the mechanism and the properties of the methods which are discussed in Section 5.3, focusing on how bi-stochastication maintains the information from the similarity matrix and uses it in order to perform estimation based on it.

On one hand, these positive results sustain the scope of this exploratory study, which was to broaden the range of possible ways for choosing analogies that are more likely to give us useful information about a new, under estimation, project. In this sense, we may say that the methods we studied address the problem of dynamic selection of analogies. On the other hand, results indicate that we have to perform an in more depth analysis of the methods in order to fully understand their advantages, drawbacks and potentials under different datasets, towards their generalization and large-scale exploitation. This is the main direction of our future research.

Additionally, there is a variety of the methodologies' aspects that affect our algorithm and require further study. For instance we can highlight the choice of the similarity function and the tuning of its parameters (like the gamma parameter we used in Eq. 13) for every dataset.

Regarding the threats to validity of our study, it is important to point out that we chose to experiment with two historical datasets in order to explore whether the methods are meaningful and applicable, in an attempt to understand the behavior of the procedures, having as a reference point EbA. Therefore, we firstly emphasize that it is not our scope to show that the methods we study are generally better than any other method and secondly that the results should not be generalized to any population of projects. Our aim is to show that bi-stochastication provides new perspectives in the research related to analogies and deserves the attention of researchers in the field.

However, even for this exploratory study on the specific datasets we used several statistical methods, appropriate for evaluation of the prediction accuracy. Due to the small or medium sizes of datasets, we decided to use leave-one-out cross-validation that gives unbiased estimates but often with high variability compared with other techniques. Concerning the statistical hypothesis testing, we used permutation tests for mean values, since the error samples are usually non-normally distributed and highly-skewed. Despite the fact that permutation tests are subject to a source of bias due to the randomness of the repeated sampling, we consider the extra variability by increasing the number of the permuted samples.

The scope of this work is mainly centered in the introduction of a considerable method that can be alternatively used to estimate the cost of a new project. As the idea is still new, it lacks of easiness of application from industry, mainly because of its difficult mathematical concepts and procedures. Therefore, our objective is to implement a suitable interface, in order to make the method approachable to software engineering professionals. At the same time, we intend to expand this study by implementing larger-scale experiments, in order to compare it with other methods and on a wider range of datasets. It is essential to understand the factors affecting the accuracy and the limitations of these new methods. Also, it would be especially interesting to study bi-stochastication in combination with other improving techniques like feature selection or outlier detection.

7. REFERENCES
for software effort estimation. MIS Quarterly, 16 (2), 155-171.


