Global Optimization of Analogy-Based Software Cost Estimation with Genetic Algorithms

Dimitrios Milios[1], Ioannis Stamelos[2], and Christos Chatzibagias[2]

[1] School of Informatics, University Of Edinburgh, Edinburgh UK EH8 9AB
[2] Department Of Informatics, Aristotle University Of Thessaloniki, Thessaloniki
Greece 54124

Abstract. Estimation by Analogy is a popular method in the field of software cost estimation. A number of research approaches focus on optimizing the parameters of the method. This paper proposes an optimal global setup for determining empirically the best method parameter configuration based on genetic algorithms. We describe how such search can be performed, and in particular how spaces whose dimensions are of different type can be explored. We report results on two datasets and compare with approaches that explore partially the search space. Results provide evidence that our method produces similar or better accuracy figures with respect to other approaches.

1 Introduction

One of the most challenging, yet essential, tasks in software industry is cost estimation. The term cost refers to the human effort spent in every phase of the development life cycle, including analysis, design, coding and testing. The most widely researched estimation methods are expert judgement, algorithmic/parametric estimation and analogy based estimation [4]. The first one involves consulting one or more experts who provide estimates using their own methods and experience. Although this practice produces estimates easily, the dependency on human factor has certain drawbacks, like subjectivity or availability of the expert. Algorithmic estimation models, such as COCOMO [4] and Function Points Analysis [1], have been proposed in order to generate estimates in a more repeatable and controllable manner. They are based on mathematical models that produce a cost estimate as a function of a number of variables. Several researchers have investigated the accuracy of such models. For example, Kemerer has suggested in [14] that the efficiency of such models is strongly dependent on the calibration of these models to local conditions. Finnie et al [9] conducted a comparison between algorithmic cost models, Artificial Neural Networks and Case-Based Reasoning for software cost estimation. It was verified that algorithmic models do not effectively model the complexities involved in software development projects. Both CBR and ANN produced more accurate estimations for the effort, however, ANN do not adequately explain their predictions. More recent research has reinforced such findings, providing further

L. Iliadis et al. (Eds.): EANN/AIAI 2011, Part II, IFIP AICT 364, pp. 350–359, 2011.

evidence in support of calibration. For example, in [16] it was found that estimates based on *within-company* projects are more accurate w.r.t. estimates based on *cross-company* historical data. In addition, research efforts have been dedicated to approaches that aim to explain better cost estimates to users and managers [3].

Estimation by Analogy (or EbA) is a case-based reasoning technique, in which cost estimation is produced by comparing the characteristics of the target project with the ones of old cases. These old cases are projects with known costs, also called *analogies*. It is assumed that the cost of the project to be estimated will be similar to that of the most similar projects. The similarity between two projects can be specified using the distance between them in the N-dimensional space defined by their characteristics. In most of EbA recent implementations [12][15][17], Euclidean distance is adopted for measuring project dissimilarity. In this work, the following distance metrics are also considered: *Euclidean*, *Manhattan*, *Minkowski*, *Canberra*, *Czekanowski*, *Chebychev* and *Kaufman-Rousseeuw*.

Projects are classified differently for each one of the metrics above, thus method's performance is affected. The choice of attributes that describe the projects also affect this classification, since the attributes are not of the same importance [21]. Attribute importance can be expressed either as the subset of attributes that best describes the given dataset, or as a vector of weights that are applied to the attributes. Furthermore, it has been already stated that the prediction accuracy is sensitive to the number N of analogies [21]. One more issue is how to combine their values in order to produce an estimate. In this paper the following statistics are used for the adjustment of the efforts of the most similar projects: *Mean*, *Median*, *Size Adjusted Mean* and *Size Adjusted Median*.

The objective of the current paper is to propose an automated strategy to select a combination of the parameters described above that result in the best performance for EbA. What we have is an optimization problem featuring a huge search space. The width of the possible solutions as well as the lack of an underlying mathematical model imposes the use of approximate search algorithms. In this domain, a popular optimization method is Genetic Algorithms (or GA).

A brief overview of related work is presented in Sect. 2. Section 3 describes how global optimization can be implemented with GA. Section 4 provides some experimental results. Finally, Section 5 concludes and provides future research ideas.

2 Related Work

Estimation by Analogy, was considered as valid alternative to conventional methods long ago [4], although it was not presented as estimation methodology until 1997 by Shepperd and Schofield [21]. The efficiency of EbA is dependent on numerous parameters that affect the way of retrieval of similar projects and producing estimations. In [21], it was found that the exclusion of the least important attributes results in improvement of prediction accuracy. In the same paper, the number of analogies was marked as another parameter that has to be carefully determined, since it has a strong effect on final estimates. In [2], different distance metrics were also investigated and it was proposed that interval estimates be produced for EbA.

The optimization of certain parameters in applying analogy was among the goals of AQUA method [18]. More specifically, a learning phase was applied in order to find the number of analogies and the threshold for the degree of similarity that granted the maximum prediction accuracy. Exhaustive search was used, by applying jackknife [8] for each number of analogies and similarity threshold. It was observed that although, in general, high numbers of analogies tend to produce increased values of MMRE, best performance is achieved for a different number of analogies for each database. Furthermore, the optimum number of analogies varied for different values of the threshold.

The importance of the project attributes was also discussed in various papers. For example in [15], a stepwise procedure based on statistical analysis was used, in order to select the most important attributes. However, it is more common that weighting factors are used to adjust the influence of each attribute in correspondence with its own significance. In [12], a GA approach was proposed as a search method for identifying a combination of weights that maximizes the accuracy of the EbA. Further improvement was possible after applying both linear and nonlinear equations as weighting factors. The identification of the optimum weights was the objective of a second version of AQUA that was published in 2008 [17], establishing a way of attribute weighting based on rough set analysis.

In 2007, Chiu and Hung [5] proposed an adjustment of the efforts of the closest analogies, based on similarity distances. According to experimental results, this way of adjustment outperformed the typical adjustment methods. It is also important that more than one distance metrics were used in these experiments, including Euclidean, Manhattan and Minkowski. It is interesting, however, that there was no distance metric found that was globally superior to others. In fact, performance of each distance metric varied both for different ways of adjustment and different datasets.

From what has been discussed so far, there is no optimization proposal that considers all parameters involved in EbA. Consequently, although genetic algorithms have already been adopted in the field of cost estimation [12][19], the current paper focuses on a fitness function and a problem representation that enable the search to be expanded to any application parameter available. The challenge is to combine the variables that represent the alternatives in applying EbA with the N-dimensional space derived from the weighting factors of project attributes.

3 Global Optimization by Genetic Algorithms

GA [11] is a search technique inspired from the genetic evolution in nature. Given an optimization problem, populations of possible solutions are considered, from which the fittest ones are going to survive and get combined so as to form a probably better population. The new set of solutions, or else the next generation, goes through the same process. It has been observed that the average fitness of each generation is increased. So, it is quite possible that the population will converge to one of the local optima.

In the case of EbA optimization, the individuals of the population are different combinations of the alternatives for applying the EbA method. In general, most GAs consist of the following components [10]:

- Encoding refers to the assignment of a unique string to each solution. The characteristics of a solution are translated into *genes*. Genes are parts of a larger string, the chromosome, in which the entire solution is encoded.
- Genetic operators are applied to the chromosomes. Their purpose is to produce the genes of the chromosomes of the next generation.
- Fitness evaluation function assigns to each solution-individual a value that corresponds to its relative fitness.
- Selection algorithm is applied to the evaluated solutions. The selection favors the fittest among the individuals, which is basic principal of evolution theory.

However, only two of these components are problem dependent: the encoding and the evaluation function.

3.1 Selection and Reproduction

Starting from the selection algorithm, the roulette-wheel selection is a typical choice in GA textbooks [10]. The probability of selecting an individual to participate in the process of reproduction is proportional to its fitness. This feature captures the main principle of evolution, the survival of the fittest.

As we will see in Sect. 3.3, the chromosomes are encoded in a binary form, so we can use some of the simplest, yet effective reproduction operators that can be found in the literature [10]. One-point binary crossover randomly combines parents' chromosomes, producing children that inherit most of the genes of their parents, while there is still a probability to create random genes with unknown effect. In order to prevent premature convergence, a mutation operator is used with much less probability e.g. about 5%. Binary mutation involves the random inversion of any bit of a chromosome, destroying its properties.

3.2 Evaluation

As already mentioned, individuals are combinations of EbA parameters. The fitness of any combination can be expressed as an accuracy measure of its application on the known projects of an available dataset. Accuracy measures, as defined in [6], make use of the Magnitude Relative Error:

$$MRE = \frac{|act - est|}{act} \tag{1}$$

where *act* is the project's actual effort, and *est* is the estimated value produced by applying EbA.

A widely accepted accuracy measure is the Mean MRE (or MMRE). The purpose of the algorithm when using this criterion is to minimize the error. Another accuracy measure that is also considered is Pred(0.25), which computes the number of projects that have been estimated with an MRE of less than 25%. The fitness of an individual is directly proportional to the second criterion. Moreover, the median of MRE values, MdMRE, is also used as it is less likely to be affected by extreme values. Eventually, in our approach these criteria are combined using the following ad-hoc fixed formula:

$$\mathit{fitness} = -0.4\mathit{MMRE} - 0.3\mathit{MdMRE} + 0.3\mathit{Pred}(\mathit{0.25}) \tag{2}$$

The purpose of the GA is to maximize Equation (2) for the projects of some training dataset. A jackknife method is used consisting of the following steps:

1. For each project with known effort that belongs to the database
2. Remove the current project from the database
3. Apply EbA with the selected configuration, in order to produce an estimate
4. Push the project back to its original dataset

The procedure above will be repeated for each one of the individuals of a population, which correspond to different configurations of EbA.

3.3 Encoding

We have chosen to encode EbA configuration parameters into binary strings. This is a convenient choice, since we can then apply the simple genetic operators described in Sect. 3.1. Moreover, this choice allows to easily explore a massive search space whose dimensions are essentially of different type.

Application Parameters. The use of one or another parameter for applying EbA should be translated into binary words (i.e. genes). Provided that genes have fixed size, each one should be long enough to store all the possible options for a parameter. The number of bits needed to encode a parameter will be:

$$\mathit{bits} = \lceil \log_2 S \rceil \tag{3}$$

where S is the number of the possible values of the parameter encoded to the specified gene. In this paper, we have considered three parameters for EbA: the distance metric, the number of analogies, and the analogies' adjustment method. It is straightforward to allocate an appropriate number of bits to represent the possible values for these parameters in a binary string. This policy of encoding is quite flexible in the case we want to examine the effect of more parameters.

Project Attributes. Project attributes can be either numerical or categorical. In the first case, they are normalized between 0 and 1, in order to neutralize the side-effects of different units. This normalization is described by the type below:

$$\mathit{normalized}_{xy} = \frac{\mathit{attr}_{xy} - \mathit{min}_x}{\mathit{max}_x - \mathit{min}_x} \tag{4}$$

Where $attr_{xy}$ is the value of the x-th numerical attribute for the y-th project of the dataset, min_x is the minimum value observed for the x-th attribute in the current dataset, while max_x is respectively the maximum value observed. In the case of categorical attributes, only values selected from a limited set can be assigned. As there is no order imposed among possible values, a '0' indicates that two compared values are the same, while '1' is used when they are different. In this way, the definition of similarity is compatible with that of numerical values.

Encoding of the Attribute Subsets. In the simplest case, an attribute will either participate in distance calculation or not. Just one bit is enough for each attribute indicating its presence in calculations. Finally, we have as many one-bit genes as the maximum of attributes available in the dataset, with each gene corresponding to a certain attribute.

Non-Normalized Weights Encoding (NNWE). Any real number could be applied as weighting factor to an attribute. However, we do not have to consider negative weights, since there is no use in counting the degree of unsuitability. It is also reasonable to allow values only in the range $[0, 1)$.

The next step is to choose an appropriate level of accuracy for the weighting factors. Two decimal digits are considered enough to represent all significant variations in attributes' importance. So, the length of each attribute gene will be determined by Equation (3). This Non-Normalized Weights Encoding (or NNWE) is used in the majority of the experiments run in Sect. 4. Although it performs well, it is not considered as optimum, due to the existence of weight vectors that have same effect.

An alternative encoding is also proposed, that allows weighting vectors that are linearly independent only. However, NNWE is still preferred. In the next paragraph, there is a detailed description about the need for excluding linearly dependent vectors and new problems that arise.

Normalized Weighting Vector Encoding (NWVE). In general, when applying a weighting vector to a set of attributes, the N-dimensional space defined by these attributes is modified. Weighting vectors that are linearly dependent produce the same space, in a different scale however. This means that distance between any two points will be different in each one of the modified spaces, but the relative position of these points will be the same. Thus, we should use only normalized vectors as candidate solutions, which means that a new representation has to be specified. It is difficult to generate binary words from normalized vectors, as any bit inversion or interchange would probably destroy the unary length property.

Instead, we could take advantage of the fact that all these vectors define an $(N-1)$-sphere in an N-dimensional space. Attribute weights are normally expressed in Cartesian coordinates. Equations (5), which are adapted from [20], show the relationship between hyperspherical and Cartesian coordinates in an N dimensional space, such as the one defined by the weights of N attributes.

$$
\begin{aligned}
x_1 &= r \cos \phi_1 \\
x_2 &= r \sin \phi_1 \cos \phi_2 \\
&\dots \\
x_{n-1} &= r \sin \phi_1 \dots \sin \phi_{n-2} \cos \phi_{n-1} \\
x_n &= r \sin \phi_1 \dots \sin \phi_{n-2} \sin \phi_{n-1}
\end{aligned} \tag{5}
$$

Vectors in the hyperspherical system include a radius r and N-1 angles ϕ_i of the vector with each one of the N-1 axes. Since we are interested in the portion of space where all weights are non-negative, we can restrict the range of all $N-1$ angles to $[0, \pi/2]$. Moreover, as far as only normalized vectors are used, radius will always be equal to 1, so it can be considered as a constant. In this way, any combination of N-1 ϕ_i angles represents a vector of unary norm. Each one of them has unique influence in EbA, since linear independence among different individuals is guaranteed. The vector of angles can easily be encoded into binary strings that can be recombined without destroying the vector's unary norm. Each angle corresponds to a gene that consists of a numbers of bits. Therefore, the hyperspherical system is used to encode the attribute weights into genes, while Equation (5) is used to transform to Cartesian coordinates to calculate distances.

In this way, there is a unique bit string representation for any weighting vector. However, there is a critical drawback that is derived from the way of computation of Cartesian coordinates. In Equations (5), we can see that a weighting factor x_k, where $1 \leq k \leq n-1$, is dependent on k sines or cosines, while x_{n-1} is dependent on $n-1$ sines. It is evident that for large values of k, x_k tends to be zero, as it is a product of a large set of numbers whose absolute values are smaller than 1.

On the other hand, the existence of few attributes could probably diminish that bias. So, given a dataset described by a small number of attributes, it is worth exploring the possibility of further improvement in estimation accuracy, when using Normalized Weighting Vector Encoding (or NWVE). Hence, we are going to perform experiments for this encoding too. In any case however, NWVE needs to be revised so as to be generally applicable.

4 Experiments

The experiments are conducted in two phases. In the first phase, the optimization method is applied in Desharnais dataset [7]. This is sort of a sanity check of the method, trying to measure the improvement, if any, of EbA after applying a global optimization with GA. In the second phase, the goal is to produce comparative results with other EbA approaches [12][17]. ISBSG is a popular database for project estimation and benchmarking [13], and there is already evidence of the performance of various methods using this dataset. In both cases, the datasets were randomly split into training and validation sets. The 2/3 of each dataset are used for the training phase, which involves applying the GA.

The results presented correspond to the remaining 1/3 of the projects. It is noted that the datasets have been split several times and the results presented are the averages of the different dataset instances. Is it also noted that any project with one or more missing values is excluded from EbA optimization.

4.1 Results on Desharnais Dataset

The global optimization method is applied using three different encodings for project attributes, as described in Sect. 3.3. Table 1 summarizes the results of the simple EbA and the three different encodings. The MMRE, MdMRE and Pred(0.25) values presented are the average values out of ten different trials.

The first thing we observe is a significant improvement in all of the evaluation criteria, when comparing the simple EbA with the GA optimized ones. Moreover, the MMRE for NNWE has been further reduced to 44.33%, compared to the 46.18% of Subset Encoding. This is due to the fact that weighting factors provide a more refined way of regulating the influence of each attribute in distance calculations.

Table 1. Results of Global Optimization approaches in Desharnais dataset

	EbA	Global Optimization with GA		
		Subset Encoding	NNWE	NWVE
MMRE %	73.02	46.18	44.43	40.67
MdMRE %	48.5	37.14	38.11	36.8
Pred(0.25) %	24.8	39.2	38.4	38.8

Finally, NWVE produced the best results in terms of MMRE and MdMRE, 40.67% and 36.8% respectively. This seems to confirm that attribute weights should be handled as normalized vectors. However, as reported in Sect. 3.3, a large number of project attributes would introduce serious bias towards certain attributes. That is the reason why we use NNWE to measure methods efficiency at the sections that follow, as ISBSG dataset makes use of a large number of attributes.

4.2 ISBSG Results and Comparison

In this test case, we have adopted the assumption made in AQUA+, including projects with effort between 10000 and 20000, as representatives of medium size projects [17]. Such an assumption is realistic, since it is possible for an experienced manager to define a priori the size of a project in general terms. After eliminating projects with missing values, a total of 65 projects remain in effort range [10000 − 20000]. Table 2 presents the results of AQUA+ for the same estimation range and the best case obtained by attribute weighting by genetic algorithms [12].

It is clear that our method outperformed Attribute Weighting GA. Actually, this was expected to happen, since the search has been expanded to parameters that had not been taken into account in the past. It also outperforms by a

Table 2. Results of EbA approaches in ISBSG dataset

	AQUA+	Attribute Weighting GA	Global Optimization GA
MMRE%	26	54	23
MdMRE%	-	34	23.4
Pred(0.25)%	72	45	59.4

limited amount AQUA+ MMRE. However, it is important to note that AQUA+ had a higher value for Pred(0.25). It should be noted that AQUA+ used a larger collection of projects as a result of tolerating missing values. In any case, method's efficiency is promising, however the elimination of a number of projects due to missing values seems to be an issue.

5 Conclusions and Future Work

The purpose of the current work was to explore the unified search space that emerges from the combination of many parameters which are incompatible at first sight. Genetic Algorithms were selected to conduct this search, due to their ability to approach global optima, no matter what the properties of the search space are. However, the key element was the encapsulation of variables that are different in nature, such as the distance metric to be used and the attribute weights, into a single encoding.

Summarizing the results for Desharnais dataset, optimization based on attribute weights outperformed attribute selection. Two different kinds of encoding were used for attribute weights. Both of them resulted in considerable improvement in prediction accuracy, while NWVE seems to be superior to NNWE. This fact supports the hypothesis that there should be a single representation for each weighting vector. However, because of some computational issues, as reported in Sect. 3.3, NWVE is not used in the rest of the experiments in the current paper. Further adjustment for dealing with this problem is subject of future research. Finally, the results in ISBSG dataset seem to support the concept of broadening the search space. By comparing with optimization attempts in earlier researches [12][17], evidence was found that the optimum configuration is obtained with combinations of multiple parameters.

The generation of the current experimental data involved distance metrics, analogy numbers, attribute weights and a small set of adjustment methods only. Of course, there are more options than the ones reported. Nevertheless, the encoding adopted can be easily expanded to contain more alternatives for any parameter. Such an expansion demands no change in the rationale of the encoding. An even more extensive search is an interesting subject of future work. Another issue to be looked into is to deal with missing values. Moreover, Equation (2), which was adopted for fitness evaluation, is ad-hoc and needs further investigation.

References

1. Albrecht, A.: Software function, source lines of code, and development effort prediction: a software science validation. IEEE Trans. Softw. Eng. 9(6), 639–648 (1983)
2. Angelis, L., Stamelos, I.: A simulation tool for efficient analogy based cost estimation. Empirical Softw. Engg. 5(1), 35–68 (2000)
3. Bibi, S., Stamelos, I., Angelis, L.: Combining probabilistic models for explanatory productivity estimation. Inf. Softw. Technol. 50(7-8), 656–669 (2008)
4. Boehm, B.: Software engineering economics, vol. 1. Prentice-Hall, Englewood Cliffs (1981)
5. Chiu, N., Huang, S.: The adjusted analogy-based software effort estimation based on similarity distances. J. Syst. Softw. 80(4), 628–640 (2007)
6. Conte, S.D., Dunsmore, H.E., Shen, V.Y.: Software Engineering Metrics and Models. Benjamin-Cummings Publishing Co., Inc. (1986)
7. Desharnais, J.M.: Analyse statistique de la productivitie des projets informatique a partie de la technique des point des fonction (1989)
8. Efron, B., Gong, G.: A leisurely look at the bootstrap, the jackknife, and cross-validation. The American Statistician 37(1), 36–48 (1983)
9. Finnie, G.R., Wittig, G.E., Desharnais, J.M.: A comparison of software effort estimation techniques: using function points with neural networks, case-based reasoning and regression models. J. Syst. Softw. 39(3), 281–289 (1997)
10. Goldberg, D.: Genetic algorithms in search, optimization, and machine learning. Addison-Wesley, Reading (1989)
11. Holland, J.: Adaptation in natural and artificial systems. The University of Michigan Press, Ann Arbor (1975)
12. Huang, S., Chiu, N.: Optimization of analogy weights by genetic algorithm for software effort estimation. Inf. Softw. Technol. 48(11), 1034–1045 (2006)
13. International Software Benchmark and Standards Group. ISBSG Data Release 10
14. Kemerer, C.: An empirical validation of software cost estimation models. Communications of the ACM 30(5), 416–429 (1987)
15. Keung, J., Kitchenham, B.: Analogy-X: Providing statistical inference to analogy-based software cost estimation. IEEE Trans. Softw. Eng. 34(4), 471–484 (2008)
16. Kitchenham, B., Mendes, E., Travassos, G.: Cross versus within-company cost estimation studies: a systematic review. IEEE Trans. Softw. Eng. 33(5), 316–329 (2007)
17. Li, J., Ruhe, G.: Analysis of attribute weighting heuristics for analogy-based software effort estimation method AQUA+. Empirical Softw. Engg. 13(1), 63–96 (2008)
18. Li, J., Ruhe, G., Al-Emran, A., Richter, M.: A flexible method for software effort estimation by analogy. Empirical Softw. Engg. 12(1), 65–106 (2007)
19. Oliveira, A., Braga, P., Lima, R., Cornélio, M.: GA-based method for feature selection and parameters optimization for machine learning regression applied to software effort estimation. Inf. Softw. Technol. 52(11), 1155–1166 (2010)
20. Schweizer, W.: Numerical quantum dynamics. Kluwer Academic Publishers, Dordrecht (2001)
21. Shepperd, M., Schofield, C.: Estimating software project effort using analogies. IEEE Trans. Softw. Eng. 23(11), 736–743 (1997)