Decision Support Analysis for Software Effort Estimation by Analogy

Jingzhou Li
Software Engineering Decision Support Laboratory, University of Calgary, Calgary AB, Canada, T2N1N4
jingli@ucalgary.ca

Guenther Ruhe
Software Engineering Decision Support Laboratory, University of Calgary, Calgary AB, Canada, T2N1N4
ruhe@ucalgary.ca

Abstract

Effort estimation by analogy (EBA) is an established method for software effort estimation. For this paper, we understand EBA as a meta-method which needs to be instantiated and customized at different stages and decision points regarding a specific context. Some example decision problems are related to the selection of the similarity measures, the selection of analogs for adaptation or the weighting and selection of attributes.

This paper proposes a decision-centric process model for EBA by generalizing the existing EBA methods. Typical decision-making problems are identified at different stages of the process as part of the model. Some existing solution alternatives of the decision-making problems are then studied. The results of the decision support analysis can be used for better understanding of EBA related techniques and for providing guidelines for implementation and customization of general EBA. An example case of the process model is finally presented.

1. Introduction

The idea of offering decision support always arises when decisions have to be made in complex, uncertain and/or dynamic environments. The process of software development is an ambitious undertaking, in which many decisions have to be made concerning processes, products, tools, methods and techniques. From a decision-making perspective, all these questions are confronted by different objectives and constraints, a huge number of variables under dynamically changing requirements, processes, actors, stakeholders, tools and techniques. Rules of thumb, intuition, tradition or gut feeling are no longer applicable to make such important decisions as reuse of artifacts, termination of testing, selection of techniques, or allocation of resources as part of software development.

Software Engineering Decision Support (SEDS) [1] is an integrated effort to aid decision-making in software development process. It can also be seen as an extension and continuation of the experience factory [2] and learning software engineering approaches [1]. In addition to collecting, retrieving and maintaining models, knowledge, and experience in the form of lessons learned, SEDS generates new insights from pro-active analysis and evaluation of possible decision scenarios.

Typically, a concrete decision support system is focused on a relatively narrow problem domain, in which the emphasis is on providing a methodology for pro-active generation, evaluation, prioritization and selection of solution alternatives. However, the results of this process can only be as good as the underlying models and experience. This is exactly the main purpose of Empirical Software Engineering: to incrementally establish a body of empirically validated knowledge about existing or new phenomena [3].

In this paper, we are specifically interested in exploring the decision problems in Effort Estimation by Analogy (EBA), which is one of the frequently used methods for software effort estimation. We conduct an analysis for decision support to the application and customization of EBA that is considered as a research agenda to establish the necessary knowledge from a series of empirical studies to be conducted in the future.

The rest of the paper is organized as following. After a brief introduction to EBA, a decision-centric process model of EBA is proposed in section 2. The major decision problems are identified from the tasks in the process model and some existing solution alternatives for the decision problems are studied in section 3. An example case in section 4 demonstrates how this process model can guide practitioners to apply and customize EBA. Section 5 discusses the current problem of the comparison between variants of EBA in general and a possible way to solve the problem. Finally, the conclusion and future work are presented in section 6.

2. Decision-centric process model for effort estimation by analogy

2.1 Effort estimation by analogy

Software effort estimation by analogy [4, 5] is mainly a data-driven method. It compares the project under consideration (target project) with similar historical projects through their common attributes, and determines the effort of the target project as a function of the known efforts from similar historical projects. EBA can be used for effort estimation for objects at levels of project, feature, or requirement, given corresponding historical data sets. There are three basic steps of EBA to estimate the effort for a given object under estimation:

Step 1. Retrieve analogs (or similar objects) of the given object from the historical data set through a set of common attributes using certain similarity measures;

Step 2. Determine the closest analogs of the given object;

Step 3. Predict the effort of the given object by adapting the effort information of the closest analogs, which is referred to as analogy adaptation.

For this paper, we understand EBA as a meta-method which needs to be instantiated at different stages and decision points. Example decisions are:

- What similarity measures are the appropriate ones for use in which conditions?
- How many analogs should be chosen for adaptation?
- How can one determine the appropriate number of analogs?
- Which adaptation strategy should be used under which conditions?
- What if there are missing values in the given data set?

Some researchers have proposed various methods as alternative answers regarding some of the above questions. Nonetheless, there has not been any systematic discussion yet about what all the possible decision-making problems are and how to compare and choose among possible alternative solutions for EBA methods.

Answering the above questions actually means to provide decision support for the application and customization of EBA. What are the basic decision problems in EBA? What are the existing solution alternatives of the decision problems? This paper is not intended to provide a systematic evaluation methodology, but to identify the space of possible decision problems and solution alternatives for EBA.

2.2 Decision-centric process model

As we just demonstrated that although there are only three basic steps of EBA, one must answer some questions before starting with the three steps. From a decision-making point of view, tasks related to answering questions as mentioned above are organized into a decision-centric process model of EBA. To begin with the process model, the following assumption is made in this paper.

Assumption: EBA is applied based on given data sets. Definition of attributes and collection of data is a prerequisite for EBA.

Therefore, when we refer to EBA, a data set is always assumed to have been given. All the related tasks of EBA are performed based on the given data set. As for the determination of the most relevant attributes for a data set, we recommend using the Goal-Metric-Question paradigm [6]. We suggest that attributes measuring software product, process, and resource should be considered.

Based on the three basic steps of EBA and our research experience in EBA, we propose the decision-centric process model as illustrated in

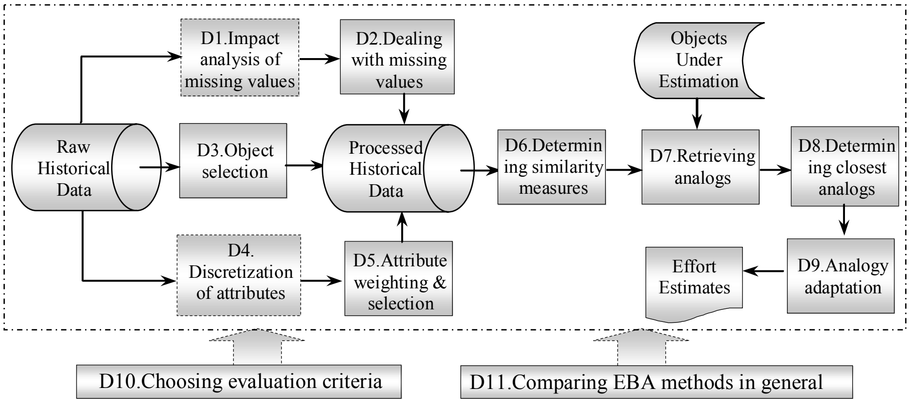
D1.Impact analysis of missing values
D2.Dealing with missing values
Objects Under Estimation
Raw Historical Data
D3.Object selection
Processed Historical Data
D6.Determining similarity measures
D7.Retrieving analogs
D8.Determining closest analogs
D4. Discretization of attributes
D5.Attribute weighting & selection
Effort Estimates
D9.Analogy adaptation
D10.Choosing evaluation criteria
D11.Comparing EBA methods in general

Figure 1. Decision-centric process model of EBA

Figure 1, which contains the major tasks of EBA, labeled as D1 to D11, and the possible input and output of each tasks. Because this is a high level description of a general EBA process, some inputs and outputs local to a task are not illustrated at this level.

The given data set for EBA is represented as a data base called Raw Historical Data that is composed of objects and attributes describing the objects using <*Attribute, Value*> pairs. In order to obtain better prediction accuracy, the Raw Historical Data can be processed in three ways, simultaneously or separately, to produce the Processed Historical Data: dealing with missing values, attribute weighting and selection, and object selection. D1 and D2 handle the missing values in the data set. D1 performs the impact analysis of the percentage of missing values in a data set on the prediction accuracy to provide practitioners with the knowledge about how EBA is applicable to the data set, given a certain percentage of missing values in the data set. D1 is represented as a dotted box because it is needed only to provide relevant knowledge in general. D2 deals with missing values in different ways such as deletion or imputation. D3 helps select a subset of the objects from the data set such that the subset of objects may produce better prediction accuracy than the whole data set. D4 and D5 perform the tasks for attribute weighting and selection. If Rough Set Analysis (RSA) based attribute weighting method is used, the continuous attributes in the data set must be discredited through D4 using certain discretization techniques according to attribute types. Because D4 is needed for RSA-based attribute weighting methods only, it is represented as a dotted box.

Types of similarity measures are determined through D6 based on the types of attributes in the Processed Historical Data. D7, D8, and D9 are corresponding to the aforementioned three steps of EBA to complete the estimation and generate the outputs of Effort Estimates with the given inputs of Objects Under Estimation.

In addition to the major tasks in the process model, the strategy for choosing evaluation criteria and comparison of EBA methods in general are necessary for the evaluation of an EBA method and the comparison between different EBA methods. Therefore, D10 and D11 are listed as supporting tasks for the whole process.

3. Decision problems in EBA and solution alternatives

The tasks in the process model actually cover the major decision problems in EBA. All these decision problems along with some existing solution alternatives are summarized in Table 1. For each decision problem, the solution alternatives must be compared in order to decide which one is most appropriate in a specific situation. These alternatives may be used independently or in combination depending on the decision problems.

In what follows, brief explanations of the decision problems and solution alternatives are given, in the order of logical dependencies between the decision problems.

Table 1. Decision problems of EBA and solution alternatives

ID	Decision problems	Solution alternatives
D1	Impact analysis of missing values	Preliminary knowledge
D2	Dealing with missing values	Deletion and imputation techniques; NULL value
D3	Object selection	Hill climbing, simulated annealing, forward and backward sequential selection algorithms
D4	Discretization of continuous attributes	For RSA-based attribute weighting; Based on interval, frequency, or both; other machine learning techniques
D5	Attribute weighting and selection	RSA-based, Wrapper, hill-climbing, genetic algorithm
D6	Determining similarity measures	Distance-based, local-global similarity principle
D7	Retrieving analogs	Using similarity measures or rule-based heuristics
D8	Determining closest analogs	Fixed number of analogs without considering similarity measure; through learning process
D9	Analogy adaptation strategy	Mean, weighted mean, linear extrapolation
D10	Choosing evaluation criteria	Some conventional criteria: e.g. MMRE, Pred
D11	EBA comparison methods in General	Accuracy-based methods

D2—Dealing with missing values: It is very often that there are missing values in the data sets for EBA. There are three major types of techniques for dealing with missing values in a data set. Let Si.j represent the j^{th} solution alternative of decision problem Di:

S2.1—Deletion [7, 8, 9]

S2.2—Imputation [7, 8, 9]

S2.3—Toleration [10, 11]

Deletion methods just ignore missing values by listwise deletion, which may result in the loss of data. Imputation methods replace missing values with estimates that are obtained based on available data. Some basic imputation methods are Mean imputation, Hot-Deck imputation, Cold-Deck imputation, Regression imputation, and Multiple imputation methods. Details about the evaluation and application of deletion and imputation methods to software cost estimation can be found in [9].

Different from the deletion and imputation techniques, missing values can be tolerated as well. This idea is used in collaborative filtering [12] and is applied in effort estimation in [11]. A specific value *NULL* along with its semantics and operations is defined for representing the missing value in [10] such that analogs are retrieved by tolerating the missing values.

D1—Impact analysis of missing values: In addition to applying the techniques dealing with missing values, knowing the impact of the amount of missing values in a data set on the eventual prediction accuracy is helpful in answering questions like whether the EBA method is applicable if there are 55% missing values in a data set in order for practitioners to apply EBA. Some preliminary knowledge has been obtained in our recent empirical study [13], in which 40% is determined to be the upper limit of the amount of missing values in a data set in order to safely apply our proposed EBA method AQUA [11].

D5—Attribute weighting and selection: Each attribute may have different degrees of relevance to the effort for implementing an object. There are mainly two ways to reflect this degree of relevance of each attribute to the effort: attribute weighting and attribute selection.

Typical techniques for attribute selection include Brute-force [14] [15] and WRAPPER [16, 17], in which attributes are added (or removed) and then tested by a learner one at a time until no better results are achieved. Attribute selection using rough sets was discussed in [18, 19], where heuristics were provided to generate the optimal subset of attributes according to certain fitness functions.

Techniques for attribute weighting include methods for quantitative data sets, such as regression [5, 20, 21] and genetic algorithm [22]. A RSA-based method of attribute weighting and selection for data sets that may contain non-quantitative attributes was proposed in [23]. All the above attribute weighting and selection alternatives are summarized as following:

S5.1—Brute-force attribute selection [15]

S5.2—WRAPPER attribute selection [16, 17]

S5.3—Rough Sets based attribute selection [18, 19]

S5.4—Attribute weighting using regression [5, 20, 21]

S5.5—Attribute weighting using genetic algorithm [22]

S5.6—Attribute weighting and selection using Rough Sets [23]

D4—Discretization of continuous attributes: Discretization is the process of converting a continuous quantitative attribute into a discrete attribute by partitioning the continuous domain of the quantitative attribute into a finite number of intervals and associating with each interval a discrete value. For RSA-based attribute weighting and selection techniques, continuous attributes in the data set must be discretized for the purpose of attribute weighting and selection only. Various discretization techniques are proposed in machine learning domain and classified in different ways, such as global or local, supervised or unsupervised. Some typical discretization techniques are summarized as following:

S4.1—Equal Frequency Intervals[24, 25]

S4.2—Equal Width Intervals [24, 25]

S4.3—Maximum entropy [26],

S4.4—Decision trees algorithm [27]

S4.5—Information-gain [28],

S4.6—Statistics-based algorithms [29],

S4.7—Class-attribute interdependency algorithms like CADD algorithm [30]

S4.8—Clustering-based algorithms like K-means discretization algorithm [31]

Some of the discretization techniques are available in the data mining tool Weka [32].

D3—Object selection: Similar to attribute selection, not all objects in a data set are necessarily useful for EBA. We believe that there exist some subsets of objects that will lead to optimal predictions. The process of searching the subsets of objects is called object selection, also known as case selection [15]. Techniques for feature selection are also applicable to case selection:

S3.1—Different variants of hill climbing algorithms

S3.2—Simulated annealing algorithms

S3.3—Forward and backward sequential selection algorithms

The above algorithms were studied in [15] over two data sets. The combination of feature selection and case selection was also studied. Suggestions were made from the results of these data sets regarding when feature selection and case selection can work well. However, further studies are necessary for more data sets in order to get more general conclusions about whether feature selection or/and case selection are useful for which type of data sets using which type of algorithms.

D6—Determining similarity measures: Basically, there are two categories of similarity measures: distance-based measure and composite measure that is based on local-global principle [33, 34]. Given that there are *m* objects described by *n* attributes in a data set. For distance-based method, each object is regarded as a vector in an *n*-dimensional space with each dimension corresponding to a participating attribute that has numerical value. Therefore, distance measures between the vectors, such as Euclidian and Hamming distance, are used to measure the similarity between objects. The shorter the distance between two objects, the more similar the objects are. Euclidian distance and weighted Euclidian distance are the most frequently used method [4, 5, 20, 21]. This technique is only applicable for data sets that have quantitative values and no missing values.

Local-global principle means that the global similarity between two objects is derived from the local similarity measures about the comparison of values between individual attributes using an amalgamation function. There are many ways to aggregate the local similarities into the global similarity, such as minimum, maximum, or *k*-maximum [33]. Weighted average of local similarity measures is used as the global similarity measure in [11]. This technique can be used for data sets that have non-quantitative attributes for which different local similarity measures can be defined. As a result, decision problems, such as which local similarity measure should be used for which type of attribute in what conditions, still need to be studied.

S6.1—Un-weighted Euclidian distance [4, 5]

S6.2—Weighted Euclidian distance [20, 21]

S6.3—Local similarity measures: depending on types of attributes

S6.4—Global similarity measures: aggregation of local similarity measures using minimum, maximum, or k-maximum [33],

S6.5—Global similarity measures using weighted average [10] of local similarity measures

D7—Retrieving analogs: Based on given similarity measures, analogs are then retrieved from the data set. In addition to similarity measures, rule-based heuristic were used in [4] for retrieving analogs based on domain-specific knowledge.

S7.1—Similarity based

S7.2—Rule based heuristics

D8—Determining closest analogs: A small number of analogs, say 1 to 3, were suggested to use in many EBA methods such as [4, 5, 20]. *k* = 2 was found to be the optimum value for small data sets in [35]. The closest analog and two virtual analogs were used in [21]. A more general method was proposed in [10] to determine the number of analogs for a given object under estimation through a learning process based on the knowledge of the accuracy distribution of the current data set.

S8.1—A fixed number of analogs, say 1 to 3 [4, 5, 20, 35]

S8.2—Virtual analogs [21]

S8.3—Through a learning process [10]

D9—Analogy adaptation strategy: Assume that *k* closest analogs are selected for adaptation to predict the effort for the given object. Techniques for analogy adaptation include:

S9.1—Mean or weighted mean of the effort of the k analogs [10]

S9.2—Production rules [4]

S9.3—Linear extrapolation along the dimension of a single attribute, a size metric strongly correlated with effort such as function points (FP) [36]: $Effort_{TARGET}=Effort_{ANALOG}*(FP_{ANALOG}/FP_{TARGET})$

D10—Choosing evaluation criteria: In order to assess the performance of EBA methods and the reliability of the estimation results, corresponding evaluation criteria must be defined. Some frequently-used criteria are *MRE* (Magnitude of Relative Error), *MMRE* (Mean MRE), and *Pred*(*l*) (prediction accuracy at level *l*) [5, 10, 20, 21]; and *MMRE*≤0.25 and *Pred*(0.25)≥0.75 are considered acceptable as proposed in [37]. Other statistical criteria such as maximum, minimum, and median of *MMRE*, absolute residual, standard deviation, correlation between the estimated and actual values, and statistical significance are also often used. These criteria are used not only for EBA, but also for other effort estimation techniques; and provide evaluation of the performance of estimation from different perspectives. Very often, different researchers choose to use different criteria. As a result, it is quite difficult to compare different estimation techniques. Furthermore, misuse of criteria in comparative studies may lead to unreliable results as indicated in [38]. How to choose evaluation criteria to achieve better evaluation and reliable comparative studies remains as a decision making problem for the application of EBA.

S10.1—MRE, MMRE, absolute residual, standard deviation, maximum, minimum, and median of MMRE [4, 5, 11, 20, 21]

S10.2—Pred(l) [4, 5, 11, 20, 21]

S10.3—Correlation between the estimated and actual values [39]

S10.4—Statistical significance [39]

S10.5—Aggregated criteria [40]

D11—EBA comparison methods in General: EBA methods are normally compared through prediction accuracy criteria. This alternative is labeled as S11.1. There are not general comparison methods for EBA. This issue will be discussed in section 5.

4. Decision support in an example EBA method

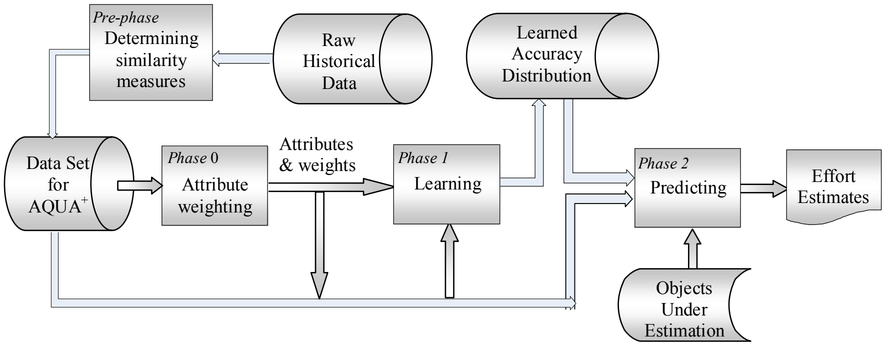

Figure 2. An instance of the decision-centric process model of EBA

As an example case, let's look at the EBA method AQUA$^+$ [23] to see how the proposed decision-centric process model can help in understanding EBA methods and guiding the application and customization of them. The four major phases of AQUA$^+$ that are proposed in [23] are shown in Figure 2. The given historical data set is called Raw Historical Data.

The correspondence between the four phases of AQUA$^+$ and the tasks in the proposed decision-centric process model is illustrated in Table 2.

Table 2. Correspondence between AQUA$^+$ phases and the tasks in the process model

Phases in AQUA$^+$	Tasks in the EBA process model
Pre-phase	D1, D2, D6
Phase 0	D4, D5
Phase 1	D8
Phase 2	D7, D8, D9

The Pre-phase prepares the Data Set for AQUA$^+$ by determining the attribute types and corresponding local similarity measures in accordance with the definitions in AQUA$^+$; and carries out tasks such as dealing with missing values and error filtering. Phase 0 performs attribute weighting and selection, including discretization of continuous attributes, using RSA-based weighting method that was proposed in AQUA$^+$. The discretization uses a combination of Equal Frequency and Equal Width intervals. The selected attributes and corresponding weights from Phase 0 are then used in the Learning phase to determine the Learned Accuracy Distribution that is used to determine the closest analogs that are likely to produce the best accuracy for analogy adaptation. The Predicting phase actually includes the three basic steps of EBA: retrieving analogs, determining the closest analogs, and analogy adaptation. We can see that decision support in AQUA$^+$ is provided mainly for attribute weighting (D1, D2) and determining the closest analogs (D8).

The solution alternatives that are used in AQUA$^+$ are listed in Table 3.

Table 3. Solution alternatives used in AQUA$^+$

ID	Decision problems	Solution alternatives used in AQUA$^+$
D1	Impact analysis of missing values	Not used in AQUA$^+$
D2	Dealing with missing values	S3: NULL value
D3	Object selection	Not supported by AQUA$^+$
D4	Discretization of continuous attributes	S4.1 and S4.2: Combination of Equal Frequency and Equal Width intervals
D5	Attribute weighting and selection	S5.6: RSA-based, four heuristics H0 to H4
D6	Determining similarity measures	S6.3, S6.5: Local-global similarity principle
D7	Retrieving analogs	S7.1: Using similarity measures
D8	Determining closest analogs	S8.2: Through a learning process
D9	Analogy adaptation strategy	S9.1: Weighted mean
D10	Choosing evaluation criteria	S10.1, S10.2, S10.5: *MMRE*, *Pred*, *Strength*, and aggregated criteria
D11	EBA comparison methods in general	S11.1: Accuracy-based method

4.3 Example of an EBA comparison method

Although only RSA-based attribute weighting and selection is employed in AQUA$^+$, there are still four heuristics to generate the weights based on different information from RSA. The four heuristics are H0—All attributes with equal weights, H1—Reducts based heuristic, H2—Core attributes based heuristic, H3—Decision rule based heuristic using the number of occurrences, and H4—Decision rule based heuristic using relative strength. With other solution alternatives fixed, the prediction accuracy of AQUA$^+$ using the four attribute weighting heuristics over five data sets was compared in [40]. The brief description of the five data sets is given in Table 4, where "#Objects" represents the number of objects in the data set, "#Attributes" represents the number of condition attributes, "%Missing values" represents the percentage of missing values, and "%Non-Quantitative attributes" represents the percentage of non-quantitative attributes. The results of the comparison are shown in Table 5. For detailed description of the five data sets, the four weighting heuristics, and the aggregated criteria *AccuH*[*i*], please refer to [40].

H1 performs the best (greater values of *AccuH*[*i*]) among the four heuristics. There is little difference between H3 and H4 over data sets ISBSG04-2, Mends03, and Kem87. Because core attributes do not always exist for any given data set, the core attribute based heuristic H2 is not recommended to be applied. For detailed discussion about the lessons learned from this comparative study, readers are directed to [40].

Table 4. Summary of the data sets

Name	#Objects	#Attri-butes	%Missing values	%Non-quantitative attributes	Source
USP05-FT	121	14	2.54	71	[10]
USP05-RQ	76	14	6.8	71	[10]
ISBSG04-2	158	24	27.24	63	[41]
Mends03	34	6	0	0	[42]
Kem87	15	5	0	40	[43]

Table 5. Comparison of weighting heuristics over five data sets for AQUA$^+$

(*AccuH*[*i*])	H0	H1	H2	H3	H4
USP05-RQ	0.22	0.42	-1.53	0.52	0.37
USP05-FT	-0.79	0.03	-	0.62	0.15
ISBSG04-2	0.16	1.81	-2.62	0.30	0.35
Mends03	-0.09	0.15	-	-0.05	-0.05
Kem87	-0.48	1.42	1.42	-0.47	-0.47

We can see from this example case that how different the prediction accuracy is, when different solution alternatives are chosen for an EBA method. Therefore, methods for the comparison between different EBA variants are necessary for practitioners to choose an appropriate EBA method in a specific situation.

5. Discussion about the comparison of EBA methods in general

Considering all the decision problems and solution alternatives as discussed in section 2 and 3, combination of the alternatives along the EBA process may result in quite different EBA methods. How to compare and then choose an appropriate EBA method for a certain type of data sets needs to be studied as well. Currently, different EBA methods are normally compared only through criteria of prediction accuracy, such as *MMRE* and *Pred* as used in [5, 10], over the same data sets.

A method, called COSEEKMO, regarding the selection of best practices of mathematical model-based software effort estimation is proposed in [39]. In addition to *MMRE* and *Pred*(0.3), the statistical significance, the correlation between the estimated effort and the actual effort, the standard deviation, and the number of attributes used in a model are also used as rejection rules to select possible best estimation method according to different types of data sets.

However, COSEEKMO is currently applicable for model-based estimation methods only, e.g. COCOMO. Furthermore, we have observed that two aspects of COSEEKMO are not applicable to EBA methods.

One aspect is the rejection rule, Rule 5, which "rejects treatments that have similar performance but use more variables", meaning that the less number of attributes in an estimation model, the better the model. This may be true for mathematical models, e.g. using only a metric of size for effort estimation, but not for EBA methods. One advantage of EBA over model-based methods is that it is able to deal with poorly understood domains that are difficult to model. The more (relevant) attributes are used in the data set, the more complete the objects are described from different aspects; and the more attributes are involved in similarity measuring, the more likely to find true analogs. One of the preliminary results from our recent empirical study [13] indicates that the more attributes, as well as objects, in a data set, the more applicable of our EBA method AQUA when there are increasing number of missing values that are replaced by the *NULL* value. This is also an indicator of the requirement of more attributes by EBA.

Another aspect is the principle regarding the size of test sets: "*Effort estimators should be assessed via small test sets since that is how they will be assessed in practice*." This principle may apply in the case of effort estimation at project level, when there are not so many projects under estimation at the same time. However, the EBA method proposed in [10] can also be applied for effort estimation at feature or requirement levels, in which cases there may be hundreds of objects under estimation at the same time. Small test sets are thus not applicable in this situation for EBA.

Consequently, techniques for comparing among different EBA methods and choosing the best-fitted one for a data set, an organization or an application domain are still required. Criteria and a process were defined in [40] for comparison between different attribute weighting heuristics using our proposed EBA method AQUA$^+$. These criteria and the comparison process may be used as a basis for comparing and choosing among different EBA methods with certain enhancement.

In addition, there is ongoing research addressing the methodology for systematic evaluation of the performance of artificial intelligent systems. One example is the methodology proposed in [44] for systematic evaluation of the performance of case-based reasoning or machine learning systems in dependent of different design decisions and domain characteristics.

In principle, this kind of methodology could be applied to the decision problems and solution space identified in this paper. The evaluation results could also be used as the basis for our decision support framework for EBA. However, the difficulty of applying this type of methodology in our situation is the huge problem and solution space. In addition, the requirement of the methodology that a mathematical model should be built to describe to relationship between the performance of the system (numerical dependent variable) and all the affecting factors (numerical independent variables) is hard to satisfy. In fact, it is impractical to test all the combinations of the possible options of the independent variables in the case of EBA methods.

As concluded in [45], we attempt to find some simple options that work well in most cases for a given data set, instead of evaluating the whole space of options. Particularly, our methodology will be focused on the decision support aspects, other than systematic evaluation of EBA methods. This is one of the goals of our future research.

6. Conclusions and future work

Based on the three basic steps of traditional EBA methods, i.e. retrieving analogs, determining the closest analogs, and analogy adaptation, this paper proposes a decision-centric process model for EBA. This process model covers the major tasks for applying and customizing EBA methods for specific situations. The typical decision problems in EBA are then identified from the tasks in the process model. The existing solution alternatives of the decision problems are also presented and analyzed to guide the application of EBA.

The analysis of an example case demonstrated that the proposed decision-centric process model can help practitioners of EBA to identify what decision problems are missing and what other solution alternatives are available for a decision problem.

Empirical studies are an effective way to providing the knowledge for decision support of many of the decision-making problems in EBA. The decision support analysis in this paper is considered a starting point for a more comprehensive empirical evaluation of all the alternatives at different stages of the EBA process model.

With a focus on the decision support for applying and customizing AQUA$^+$, our future work will be directed towards a methodology for comparing and choosing between different alternative solutions for the identified decision-making problems through systematic empirical studies. Our next goal is a decision support system for EBA that is able to provide recommendations on which alternative should be taken in dependence of the problem context and the given data set.

Acknowledgements

The authors would like to thank the Alberta Informatics Circle of Research Excellence (iCORE) for its financial support of this research. Thanks are also given to anonymous reviewers for their valuable comments.

References

[1] G. Ruhe, "Software Engineering Decision Support—A New Paradigm for Learning Software Organizations", *Advances in Learning Software Organization*, Lecture Notes In Computer Science, Vol. 2640, Springer 2003, pp 104-115.

[2] V.R. Basili, G. Caldiera, and H.D. Rombach, "Experience Factory", *Encyclopedia of Software Engineering* (Eds. J. Marciniak), Vol. 1, 2001, pp 511-519.

[3] G. Ruhe, "Software Engineering Decision Support and Empirical Investigations - A Proposed Marriage", *The Future of Empirical Studies in Software Engineering* (A. Jedlitschka, M. Ciolkowski, Eds.), Workshop Serious on Empirical Studies in Software Engineering, Vol. 2, 2003, pp 25-34.

[4] T. Mukhopadhyay, S. Vicinanza, and M.J. Prietula, "Examining the Feasibility of a Case-based Reasoning Model for Software Effort Estimation", *MIS Quarterly*, Vol. 16, No. 2, 1992, pp 155-171.

[5] M. Shepperd, C. Schofield, "Estimating Software Project Effort Using Analogies", *IEEE Transactions on Software Engineering*, Vol. 23, No. 12, 1997, pp 736-743.

[6] V.R. Basili, G. Caldiera, and H.D. Rombach, "The Goal Question Metric Approach", *Encyclopedia of Software Engineering* (Eds. J. Marciniak), Vol. 1, 2001.

[7] K. Strike, et al., "Software Cost Estimation with Incomplete Data", *IEEE Transactions on Software Engineering*, Vol. 27, 2001, pp 890-908.

[8] M. Cartwright, M. Shepperd, and Q. Song, "Dealing with Missing Software Project Data", *Proceedings of the 9th International Symposium on Software Metrics*, 2003, pp 154-165.

[9] I. Myrtveit, E. Stensrud, and U. H. Olsson, "Analyzing Data Sets with Missing Data: An Empirical Evaluation of Imputation Methods and Likelihood-Based Methods", *IEEE Transactions on Software Engineering*, Vol. 27, 2001, pp 999-1013.

[10] J.Z. Li, G. Ruhe, A. Al-Emran, and M.M. Ritcher, "A Flexible Method for Effort Estimation by Analogy", *Empirical Software Engineering*, Vol. 12, No. 1, 2007, pp 65-106.

[11] N. Ohsugi, et al., "Applying Collaborative Filtering for Effort Estimation with Process Metrics", *PROFES'04: 5th International Conference on Product Focused Software Process Improvement*, LNCS 3009, Japan, 2004.

[12] B. Sarwar, et al., "Item-Based Collaborative Filtering Recommendation Algorithms", *ACM WWW'01: Proceedings of the tenth international conference on World Wide Web*, Hong Kong, 2001, pp 285-295.

[13] J.Z. Li, A. Ahmed, G. Ruhe, "Impact Analysis of Missing Values on the Prediction Accuracy of Analogy-based Software Estimation Method AQUA", *Technical Report TR067-2007*, Software Engineering Decision Support Laboratory, University of Calgary, Calgary, Canada, January 2007.

[14] G. Kadoda, C. Michelle, L. Chen, M. Shepperd, "Experiences Using Case-Based Reasoning to Predict Software Project Effort", *Proceedings of EASE'2000 - Fourth International Conference on Empirical Assessment and Evaluation in Software Engineering*, January 2000.

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

[16] R. Kohavi, G.H. John, "Wrappers for Feature Subset Selection", *Artificial Intelligence*, Vol. 97, 1997, pp 273–324.

[17] Z. Chen, B. Boehm, T. Menzies, and D. Port, "Finding the Right Data for Software Cost Modeling", *IEEE Software*, Vol. 22, 2005, pp 38-46.

[18] M. Zhang, J.T. Yao, "A Rough Sets Based Approach to Feature Selection", *Proceedings of the 23rd International Conference of NAFIPS*, June 2004, pp 434-439.

[19] N. Zhong, J. Dong, "Using Rough Sets with Heuristics for Feature Selection", *Journal of Intelligent Information Systems*, Vol. 16, 2001, pp 199-214.

[20] E. Mendes, I. Watson, T. Chris, M. Nile, C.A. Steve, "Comparative Study of Cost Estimation Models for Web Hypermedia Applications", *Empirical Software Engineering*, Vol. 8, 2003, pp 163-196.

[21] H.K.N. Leung, "Estimating Maintenance Effort by Analogy", *Empirical Software Engineering*, Vol. 7, 2002, pp 157-175.

[22] S.J. Huang, N.H. Chiu, "Optimization of analogy weights by genetic algorithm for software effort estimation", *Information and Software Technology*, Vol. 48, No. 11, 2006, pp 1034-1045.

[23] J.Z. Li, G. Ruhe, "Software Effort Estimation by Analogy Using Attribute Weighting Based on Rough Sets", *International Journal of Software Engineering and Knowledge Engineering*, To appear.

[24] M.R. Chmielewski, J.W. Grzymala-Busse, "Global Discretization of Continuous Attributes As Preprocessing for Machine Learning", *Third International Workshop on Rough Sets and Soft Computing*, 1994, pp 294-301.

[25] J. Dougherty, R. Kohavi, and M. Sahami, "Supervised and Unsupervised Discretization of Continuous Features", *Proceedings of International Conference on Machine Learning*, 1995, pp. 194-202.

[26] A.K.C. Wong, D.K.Y. Chiu, "Synthesizing Statistical Knowledge from Incomplete Mixed-Mode Data", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.9, 1987, pp. 796-805.

[27] J.R. Quinlan, *C4.5 Programs for Machine learning*, Morgan-Kaufmann, 1993.

[28] X. Wu, "A Bayesian Discretizer for Real-Valued Attributes", *The Computer Journal*, Vol. 39, 1996.

[29] R. Kerber, ChiMerge, "Discretization of Numeric Attributes", *Proceedings of AAAI-91: the 9th International Conference on Artificial Intelligence*, pp 123-128, 1992.

[30] J.Y. Ching, A.K.C. Wong, and K.C.C. Chan, "Class-Dependent Discretization for Inductive Learning from Continuous and Mixed Mode Data", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.17, No.7, pp 641-651, 1995.

[31] J.T. Tou, R.C. Gonzalez*, Pattern Recognition Principles*, Addison-Wesley, 1974.

[32] I.H. Witten, E. Frank, *Data Mining: Practical machine learning tools and techniques* (2nd Edition), Morgan Kaufmann, San Francisco, 2005.

[33] M.M. Richter, "On the Notion of Similarity in Case-Based Reasoning", *Mathematical and Statistical Methods in Artificial Intelligence* (Eds. G. Della Riccia, et al.), Springer Verlag, 1995.

[34] M.M. Richter, "Similarity", *Case-Based Reasoning on Signals and Images* (Eds. P. Perner), Series on Computational Science, Springer Verlag, 2007.

[35] C. Kirsopp, M. Shepperd, "Making inferences with small numbers of training sets", International Conference on Empirical Assessment of Software Engineering, Keele University, UK, 2002.

[36] F. Walkerden, R. Jeffery, "An Empirical Study of Analogy-based Software Effort Estimation", *Empirical Software Engineering*, Vol. 4, No. 2, 1999, pp 135-158.

[37] S.D. Conte, H. Dunsmore, and V.Y. Shen, *Software engineering metrics and models*, Benjamin-Cummings Publishing Co. Inc., 1986.

[38] I. Myrtveit, E. Stensrud, and M. Shepperd, "Reliability and Validity in Comparative Studies of Software Prediction Models", *IEEE Transactions on Software Engineering*, Vol. 31, No. 5, 2005, pp 380-391.

[39] T. Menzies, Z.H. Chen, J. Hihn, and K. Lum, "Selecting Best Practices for Effort Estimation", *IEEE Transactions on Software Engineering*, Vol. 32, No. 11, 2006, pp 1-13.

[40] J.Z. Li, G. Ruhe, "A Comparative Study of Attribute Weighting Heuristics for Effort Estimation by Analogy", *Proceedings of the ACM-IEEE International Symposium on Empirical Software Engineering (ISESE'06)*, September 2006, Brazil.

[41] ISBSG, Data R8, *International Software Benchmark and Standards Group*, www.isbsg.org, October 18, 2005.

[42] E. Mendes, I. Watson, T. Chris, M. Nile, and C. Steve, "A Comparative Study of Cost Estimation Models for Web Hypermedia Applications", *Empirical Software Engineering*, Vol. 8, No. 2, 2003, pp 163-196.

[43] C.F. Kemerer, "An Empirical Validation of Software Cost Estimation Models", *Communication of the ACM*, Vol. 30, No. 5, 1987, pp 436-445.

[44] J.C. Santamaria, A. Ram, "Systematic Evaluation of Design Decisions in Case-Based Reasoning Systems", *Case-Based Reasoning: Experiences, Lessons, and Future Directions*, (David B. Leake Eds.), AAAI Press/MIT Press, 1996.

[45] R.C. Holte, "Very Simple Classification Rules Perform Well on Most Commonly Used Dataset", *Machine Learning*, Vol. 11, No. 1, 1993, pp 63-91.