Effort Estimation of Business Process Modeling through Clustering Techniques

Brunno Silveira, Felipe Klussmann, Fernanda Baião, Kate Revoredo

Department of Applied Informatics
Federal University of the State of Rio de Janeiro (UNIRIO)
Av Pasteur 458 – CCET – Rio de Janeiro – RJ – Brazil

{brunno.silveira, felipe.klussmann, fernanda.baiao, katerevoredo}@uniriotec.br

Abstract. *A critical activity in project planning, especially in Business Process Modeling (BPM) projects, is effort estimation. It involves several dimensions such as business domain complexity, team and technology characteristics, turning estimation into a difficult and inaccurate task. In order to reduce this difficulty, background knowledge about past projects is typically applied; however, it is too costly to be carried out manually. On the other hand, Data Mining enables the automatic extraction of new nontrivial and useful knowledge from existing data. This paper presents a new approach for BPM project effort estimation using data mining through clustering technique. This approach was successfully applied to real data.*

1. Introduction

Business Process Modeling (BPM) [Indulska *et al.*, 2009] has been adopted by organizations in many different contexts, such as for reviewing the organization practices, guiding the definition of strategies for Information Technology (IT) or driving the specification of information systems artifacts [Weston, Chatha, Ajaefobi, 2004], [Ferreira *et al.*, 2011]. In fact, BPM has already been pointed as one of the phases for developing an information system, and recent researches suggest that a business process model stands as an important starting point both for the identification of adequate services to support a Service Oriented Architecture (Service Oriented Architecture – SOA) [Woodley and Gagnon, 2005], [Azevedo *et al.*, 2009], [Diirr *et al.*, 2012] and for adequately designing a Datawarehouse in a business intelligence approach [Linden *et al.*, 2010], [Oliveira, 2010], [Sekine *et al.*, 2009]. This broad spectrum of scenarios turns the business process model into a valuable and critical resource in different situations, in which the cost of its elaboration should be defined as precisely as possible.

According to Mutschler and Reicher [2013], even though BPM has become a success-critical instrument for improving overall business performance, there is a need for a comprehensive approach enabling BPM professionals to systematically investigate the costs of BPM projects. Planning is one of the first activities in every BPM project, and it aims at estimating the required effort to elaborate the set of models desired. From this effort, it is possible to estimate the project duration, its financial cost, risks that may impact on the project and the resources required. However, despite its importance, it is still difficult to determine the effort required for the modeling phase of a BPM project due to the many variables involved, as well as to the subjectivity of some tasks such as process elicitation.

One of the most frequent approaches to estimate the effort required for process modeling is based on the knowledge gathering from past projects since, under similar conditions and with a well defined methodology, the modeling phase of a BPM project tends to require a similar effort to be carried out, with minor variations. These approaches search for behavioral patterns, or correlations between the variables that characterize a modeling task and its required effort. However, finding these relations is a complex and laborious task, demanding a deep knowledge and time from the specialists. In the approach proposed by Cappelli *et al.* [2010], a detailed analysis of historical BPM projects suggested three classifications according to the nature of the project: administrative projects (ADM), technical-operational projects (TOP) and technical-managerial projects (TMP). The authors then defined a mathematical model for estimating the effort of projects of each type. As expected, the mathematical equations proposed are strongly based on the number of process activities to be modeled. Their work evidenced that both the definition of the equations and the classification of projects require a lot of expertise from the specialists, reflecting the influence of human capabilities in this task [Dutra *et al.*, 2012]. This vast amount of background knowledge of the specialists comprise characteristics about the modeling team profile, the average productivity of each participant and the probability of having to replace one of the team members, the tools used, the extent to which existing process models could be reused, the number of interviews that will have to be conducted to gather information from process executors, and so on.

A frequent difficulty faced by BPM project managers when estimating the project effort for modeling a process is that most of this information is unknown if he/she did not participate in previous BPM projects in the same scenario. From a different perspective, a very long experience in a specific scenario is required from the project manager in order for him/her to adequately accurately estimate the effort of a future process modeling project. If this tacit knowledge from the specialist is not available, he/she has to rely on historical data recorded from past projects. In these cases, however, when this knowledge repository gets larger, it is not viable to analyze it manually, with no systematics [Mutschler and Reichert, 2013].

On the other hand, data mining techniques [Witten and Frank, 2011] have been successfully used in a vast amount of scenarios for automatically discovering patterns. In the context of project effort estimation, Tronto *et al.* [2007] applied a clustering technique to estimate the effort required to implement a software development project. There is, however, no work in the literature applied to the domain of a BPM project and its specificity, as shown in Dutra *et al.* [2012].

In this paper, we propose an alternative to the conventional approaches for automatically estimating the effort required for modeling a process during a BPM project, through clustering technique. This approach reduces dependency on experts and enables the analysis of large datasets. We have evaluated our proposal in a real dataset from an Oil and Gas company in Brazil.

This paper is divided as follows. Section 2 describes the main concepts of data mining, as well as the clustering technique. In Section 3, our method is presented. In

Section 4, we analyze the results obtained with the application of our method in a real dataset, which is the main contribution of this article. Section 5 analyzes and compares with related work and, finally, Section 6 presents the conclusions and future work.

2. Knowledge Discovery and Data Mining

Knowledge Discovery in Databases (KDD) [Witten and Frank, 2011], [Fayyad *et al.*, 1996] is the process of extracting patterns from datasets by combining methods from statistics and artificial intelligence with dataset management. Figure 1 illustrates the KDD process, which is an iterative process where the main step is the data mining. The three steps before are responsible for the preparation of the data, so the data mining algorithms can be applied. This preparation includes, collecting and analyzing the data, removing noisy and selecting relevant features (attributes from the dataset). The last step corresponds to the interpretation and evaluation of the patterns found. If the analyses demonstrated that the patterns found are useful then the process is done and a new knowledge found. Otherwise, previews steps can be redone in order to improve the results initially found.

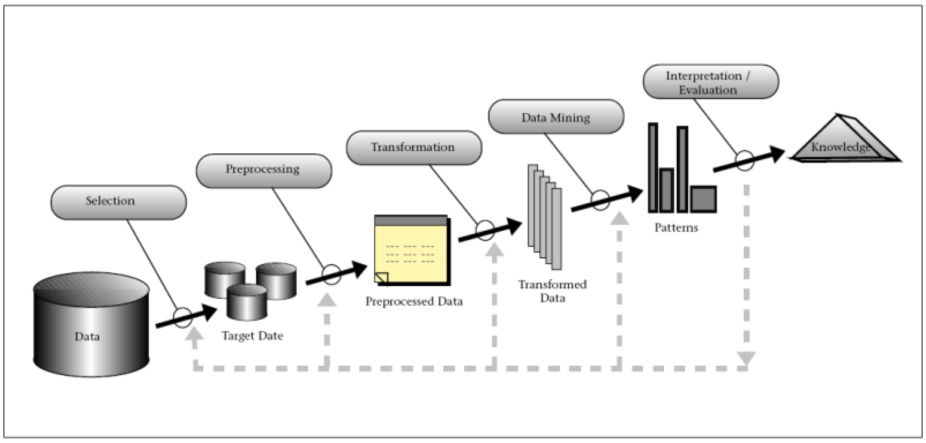

Figure 1 - Operational stages of the KDD process (Source: [Fayyad *et al.*, 1996])

Data mining is considered the most important stage of the KDD process, since machine learning algorithms performed in it are responsible for finding useful patterns [Mitchell, 1997]. The choice of which algorithm one must use depends on the task to be performed. Even though the “data mining” terminology is very broad, the knowledge discovered may have distinct natures depending on the perspective chosen as the focus of analysis. For example, in a scenario where we want to discover a process (a flow of activities executed to reach a goal), a process mining task should be considered [Aalst, 2012]. When we want to discover co-relations among the characteristics describing a set of historical data, or when we want to classify or predict the value of a new instance based on historical data (which is the case of our problem scenario), we should rely on traditional data mining tasks.

In general, data mining tasks are divided into two categories: predictive tasks and descriptive tasks. The objective of the former is to predict the value of a particular attribute (target) based on the values of other attributes (dependent variables).

Algorithms addressing these tasks aim at building a model for the target as a function of the dependent variables. An example would be a model that is able to predict whether a patient has a particular disease based on the results of medical tests. The objective of the latter (descriptive task) is to derive patterns that summarize the underlying relationships in the dataset, such as strongly associated characteristics (association analysis) or closely related instances (clustering analysis). In some scenarios, in order to find a predictive model, it is necessary to first explore the data, thus performing a descriptive task. For instance, a clustering algorithm can be used to group the data and then a predictive algorithm is applied to find a model that suits each group. Since the problem addressed in this paper concerns estimating the effort for a process modeling project based on historical data of past projects, and assuming that the historical data contains a broad spectrum of projects, some of them similar and some of them different from the project which effort is to be estimated, the proposed approach combines a clustering algorithm (descriptive task) with a predictive algorithm. In the rest of this section we describe in more details the algorithms used in our approach.

Clustering algorithms group data into clusters, where the elements of a cluster must share common properties that maximally distinguish them from the elements of other clusters. Thus, intracluster similarity is maximized while intercluster similarity minimized. Different similarity measures can be used, being the most common the Euclidean distance, or Euclidean metric.

Given the set of clusters found when a new data is known it is possible to associate it to the cluster most similar to it. Thus, the data in this cluster are the most similar data to the new one received. Another algorithm known as k-nearest neighbor (KNN) [Witten and Frank, 2011] can also be used to find the most similar data to a new one received, in this case the k most similar. However, different from the clustering algorithms, when a new data arises it must be compared to all the data in the dataset in order to decide which k data are most similar to it. Considering the large dataset and frequently new data arising, it is cost to go through the whole dataset every time a new data arises. Therefore, in this situation clustering algorithms are a better choice.

The most common clustering algorithm divides the dataset into k clusters, where k is defined by the user. Initially, the algorithm chooses k points in R^n, where n is the number of characteristics (attributes), as centroids of the clusters. The data is then divided between those clusters according to the similarity measure adopted and considering the centroids as reference. After this initial division, the data inside each cluster define the new centroids, which will be a new reference for calculating the similarity. There are different ways for defining centroids. A centroid can be the element closest to the center of gravity of the cluster (this element is called the medoid) or the average of the data belonging to the cluster in question. This latter approach is known as k-means, the name the one of the most common clustering algorithms. The algorithm is an iterative process, determining if a given data cluster should change or not to maintain a high similarity between the data intercluster and low among data intracluster.

There are many variations of the k-means algorithm, which generally differ in the selection of centroids, in the calculation of similarity, or in the strategy to calculate the average of the clusters. Among the main algorithms are: k-medoids [Theodoridis and Koutroumbas, 2006], k-modes [Huang, 1998] and SimpleKMeans [Witten and Frank, 2011]. The last two deal with categorical attributes. The SimpleKMeans will be adopted in this paper as the clustering algorithm used.

3. Applying Clustering Techniques to Estimate the Effort of a BPM project

BPM projects have a number of variables which characterize them, such as the amount of activities and processes involved. In order to estimate the effort of a new BPM project an expert usually uses knowledge about past BPM projects. The expert analyses the dataset with completed BPM projects, searching for the ones with similar characteristics compared with the new BPM project in his hand. Then, he uses his expertise to estimate the effort of this new project based on the information from the set of past BPM projects most similar to it. However, finding these set of related BPM projects is not a simple task, requiring a lot from the expert and therefore susceptible to errors. Moreover, being a manual task, there is a restriction about the number of projects already completed that the expert can analyze manually. Our proposal is to automatically identify the set of relevant completed projects that the expert must analyze, i.e. the most similar projects.

In this paper, we propose an alternative to conventional approaches of BPM projects effort estimation, where we apply data mining. Since a set of relevant completed projects may share common characteristics, we apply a descriptive algorithm in order to find sets of similar instances. Therefore, clustering algorithms are a natural choice for this task. Thus, a clustering algorithm is used to cluster the dataset with past BPM projects and then associate the new BPM project to the most similar cluster. Afterwards, the projects inside the chosen cluster predict the effort of the new BPM project. As a consequence, we reduce the dependency on the expert, thus fostering the possibility of analyzing large amounts of data on projects and obtaining more precise estimation.

The proposed method consists of two phases. In the first one, called training phase, the KDD process is used, considering a clustering algorithm in the data mining step, in order to find a set of clusters of historical BPM projects that best fits a quality metric. Therefore, the task for the training phase can be defined as:

- **Given**: a dataset with historical information of BPM projects
- **Find**: the best set of clusters according to some quality metric

In the second phase, the clusters found in the training phase are used to estimate the effort of a new BPM project. The task for this phase is defined as:

- **Given**: a set of clusters, a new BPM project and a similarity measure
- **Find**: effort estimation for the new BPM project.

When a new BPM project arises a similarity measure is used to indicate the cluster that this new project is most similar to. This is done by comparing the characteristics of the new BPM project with the characteristics consolidated in the centroid of each cluster. Then, effort is estimated considering some consolidate metric about the cost of all BPM project grouped in the chosen cluster. Different metrics can be used to help expert decide on the effort for the new BPM project received. A possible one, used in this work, is the average.

Considering that new projects arise frequently, as mentioned in Section 2, clustering algorithm is preferable compared to algorithms such as k-nearest neighbor. On the other hand, the training phase can be repeated when projects are completed, thereby adding more information that will help estimating the effort for new projects.

4. Case Study

The proposed approach was applied to a real dataset containing information about 48 BPM projects, carried out in a large Brazilian company in the Oil and Gas domain. To allow an analysis of the proposed method, 4 out of the 48 projects were separated as the test base. Therefore, the first phase of our method considered a dataset with 44 BPM projects, while the second phase considered 4 new BPM projects. Due to information confidentiality reasons, we cannot describe details about project instances; however, the attributes collected for each instance and considered in the mining process are described in details.

4.1. Scenario Description

Each project instance contained 15 attributes (characteristics) deemed relevant by experts to determine the effort of the modeling task during a BPM project. These attributes reflected characteristics about (i) the complexity of the process being modeled, (ii) the business in which the process is inserted, and (iii) the level of customer participation during the process modeling project. Additionally, one of the attributes indicated the actual modeling effort required by the project, representing the class attribute. Each attribute is described below.

- **Attributes related to the domain**
  - Number of Business Rules. This attribute indicates the number of business rules involved in the execution of the business process, mentioned by users during the elicitation phase. It is worth mentioning that the number of business rules reflects the complexity of a business process model. For example, business rules of the type action condition assertion (so called "if-then-else" rules) are reflected in the structure of the process model through logical connectors (or, xor, and) and decision points;
  - Number of Business Requirements. This attribute indicates the number of business requirements mentioned by users during the elicitation phase;
  - Number of Indicators. This attribute indicates the number of performance indicators associated to the business process, mentioned by users during the elicitation phase;
  - Number of Risks. This attribute indicates the number of risks involved in the execution of the process, mentioned by users during the elicitation phase. It is worth mentioning that the complexity of the process model increases as the number of risks increases, since this will require a greater level of detail in the process model;
- **Attributes related to the technology supporting the process**
  - Number of Systems. This attribute indicates the number of information systems that support the business process. As you increase the number of systems, the process becomes more complex, always seeking to maintain systems integrated

  - with the process and vice versa. Moreover, more technical details should typically be included in the process model so as to represent the IT perspective;
  - Number of System Interfaces. This attribute indicates the number of system interfaces the process executor interacts with when instantiating the process;
  - Number of Equipments. This attribute indicates the number of equipments needed to perform the process.
- **Attributes related to the complexity of the project scope**
  - Number of Processes. This attribute indicates the number of processes to be modeled in the project;
  - Number of EPCs. This attribute indicates the number of EPC (Event-driven Process Chain) diagrams composing the business process being modeled. An EPC diagram details the flow of activities composing a sub-process, that is, the logic of combining its activities [Scheer, 2000];
  - Number of FADs. This attribute indicates the number of FAD (Function Allocation Diagram) diagrams composing the business process being modeled. The FAD diagram details all elements of each process activity in the highest degree of detail. FAD diagrams represent, for example, the roles involved in performing the activity, input data, output data, equipments and / or systems used during its implementation and business rules observed during its implementation;
  - Number of Interface Diagrams, indicating the number of interfaces between the processes modeled in the project. The interfaces between processes indicate the invocation of another business process during (or at the end of) the execution of the current process;
  - Number of Elements, indicating the average number of elements in each FAD, including input and output data and products;
- **Attributes related to the customer profile**
  - Customer Participation, indicating the degree of customer participation (categorized as LOW, MEDIUM or HIGH) during the design process modeling.
  - Customer Factor, reflecting the dependence of the modeling team about the availability of the customer during the project. Processes in which knowledge is mostly tacit, "in the head" of users, this customer factor is HIGH, while in processes which have available updated documentation, the client factor is LOW.
- **Attribute reflecting the effort for completing the project:**
  - Normalized Cost attribute. This attribute indicates the project effort (the total number of person-hour, or P.H.). This value is normalized according to the proportional participation of each modeler in the project team.

4.2. First Phase: Learning the Clusters

In this section we describe the results obtained with the application of our proposal on the dataset considered. Since some of the attributes of this dataset are categorical we applied the SimpleKMeans clustering algorithm [Witten and Frank, 2011].

The application of SimpleKMeans required the definition of a similarity function, the number of clusters to be generated (k) and the choice of a quality metric. The similarity function used was the Euclidean distance. Furthermore, the algorithm automatically normalizes numerical attributes, between 0 and 1, when doing distance computations. To define k, we conducted a parameter tuning approach testing several values according to the Fibonacci sequence. Thus, we conducted tests considering k={2, 3, 5, 8, …} (k=1 was discarded for obvious reasons). The whole idea behind this approach is that the Fibonacci sequence adequately reflects the relation between the uncertainty of effort estimation and the project length: effort estimation uncertainty gets higher when the project length increases and, on the other way around, more precise estimates are expected for short projects. The Fibonacci sequence was previously used in agile software development projects [Cohn, 2005]. Finally, the sum of squared errors (SSE) was the quality metric adopted.

Figure 2 depicts the SSE values obtained when tuning k. As expected, SSE decreases as k increases, since the smaller the number of clusters, the higher the chance of grouping instances of low similarity within the same cluster. However, as the number of clusters increases, the chance of overfitting also increases. In our case study, k=8 led to overfitting, since two of the clusters comprised only one instance each. This made it useless to test values greater than 8 (considering the Fibonacci sequence, the next value would be k = 13). Looking at results for k = {2, 3, 5, 8}, the best result was obtained by considering five clusters, (k = 5).

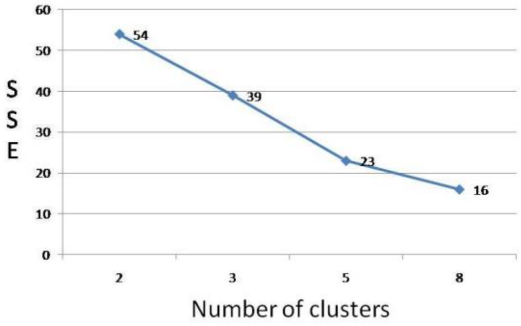

Figure 2 - SSE value as the number of clusters increases

4.3. Second Phase: Estimating the effort of new modeling projects

After learning k and grouping the project instances from the training dataset in 5 distinct clusters, we focused on learning how the combined values for all project characteristics could automatically lead to the estimation of the effort required for a new project instance.

Given a new project instance *p*, the cluster most similar to it is found considering a similarity measure over all attributes except the Normalized cost attribute, since this is the one that must be estimated. The effort for *p* is then estimated as the average normalized cost of all instances of the cluster to which *p* has the highest similarity.

We evaluated our approach using the test dataset previously defined. The values from the Normalized Cost attribute and the estimated values of the four real projects from the test base are shown in Table 1. The actual Normalized Cost contains, for each instance, the sum of working hours conducted by the team of the corresponding project, according to the spreadsheets kept by the project manager along the time. The values were normalized according to the proportional participation of each modeler in the project team, since some of the team members worked part-time. The last column indicates whether our approach was able to find the most similar value among all 5 clusters or not.

Table 1 - Clustering estimation result

Test instance	Actual normalized cost	Estimated normalized cost	Absolut difference	Closest value?
1	64,1	30,6	33,5	No
2	39,7	30,6	9,1	Yes
3	22,0	29.49	7,49	Yes
4	209,5	141.5	68	Yes

In Table 1, the actual and the estimated Normalized Cost values and the difference are presented in months. The "closest value" indicates that the instance is assigned to the cluster that has the average Normalized Cost value closest to its actual effort, meaning that the algorithm was right. The test accuracy of this approach was 75%.

4.4 Analyzing the Results

Instances 2 and 3 were grouped in the cluster with the closest Normalized Cost, with a small difference between the actual and the estimated if compared with the other two instances. The fourth instance was associated with the closest cluster, but the difference was great. One possible reason for this is that the fourth instance represented the project with the third highest Normalized Cost of the dataset, so when considering the average value of the Normalized Cost attribute for the instances belonging to the cluster in question, the estimated effort varies a lot due to the grouping with shorter projects.

The first instance, the only one that was not assigned to the closest cluster (and therefore did not have a good estimated effort) can also be analyzed individually. It represented the project with the second highest effort of its cluster, that is, with the second highest Normalized Cost value among the instances of the same cluster. Therefore, it is difficult to have an accurate estimation because projects similar to this in question have occurred only a few times. The work of Cappelli *et al.* [2010] already quoted that the models generated with the base available did not suffice to estimate complex projects.

5. Related Work

In [Mutschler and Reichert, 2013], the authors propose EcoPOST, a framework composed by evaluation models describing the interplay of technological, organizational and project-specific BPM cost factors. The framework covers the whole BPM life-cycle, including the implementation of process-aware information systems (which are based on process models) and the evolution of process models. In particular, with regard to process modeling, they model the correlation among a set of variables empirically identified as being relevant for estimating the cost of this task. Each variable in the set is related to the complexity, to the size of the process, or to the knowledge about the process. The variables related to the process size are the number of activities, events, arcs and connectors, which are also considered in our proposed approach. The other perspectives are not even described due to their complexity. In fact, the authors of EcoPOST point out the difficulty to apply their proposed framework in practice, due to the large number of evaluation concepts and tools that need to be learned by the process manager.

As mentioned in Section 1, Cappelli *et al.* [2010] proposed a method to estimate the effort of BPM projects. This method analyzed 48 completed projects, and manually classified them according to their nature: Administrative Projects (ADM), Technical management projects (TMP) and Operational Projects (TOP). This classification is not trivial and required much effort by the experts involved. They further defined formulas for estimating the effort of BPM projects considering each of the three classifications. These formulas have a strong dependence of activities involved in BPM design, that is, the expert concluded that the effort of a process modeling task is closely related to the amount of activities from the process in question. Our approach represented a valid automatic mechanism to adequately estimate the effort for modeling a process within a BPM project, in the same scenario.

Two comparison experiments were performed. The first one sought to evaluate whether the attribute type design really represents the most important characteristics for estimating project effort, as pointed out by Cappelli *et al.* [2010]. In the second experiment we evaluated whether it is possible to automatically infer such classification in project types and the relevance attributed to a characteristic of BPM projects. The analysis considered k = 3 for the SimpleKMeans algorithm, that is, we assume that BPM modeling tasks are classified into three distinct groups. So, the expectation was to verify if the manually set values defined by Cappelli *et al.* [2010] for the Project Type attribute was reflected in the clusters automatically discovered.

The first experiment assumed the 12 instances classified in [Cappelli *et al.*, 2010] as being of the ADM type. Two out of these 12 instances were separated for the test base. Table 2 shows the distribution of such ADM projects among the three clusters.

Table 2 - ADM Projects distributed in three clusters (k = 3)

Cluster	# of instances	Normalized Cost	# of FADs
0	2	16.15	19
1	5	40.28	44.4
2	3	19.33	58.67

The result shows three distinct project groups. The first, represented by the cluster 0, are small projects, with few FADs and less effort. However, analysis of clusters 1 and 2 is not so simple. If the number of FADs represents the number of activities and the activities of a process are a determining factor in the conclusion effort, it is strange that the group formed by cluster 2 have an average number of FADs considerably higher (~ 32% more) and the Normalized Cost value, representing the effort required to complete the project, considerably smaller (~ 52% lower) than the cluster 1. Consequently, this analysis suggests that the relevance of such characteristic for estimating the effort of each project is not so important, or at least not alone, but has been considered decisive in the work of Cappelli *et al*. [2010]. So, the method proposed in this article is useful in that sense, since it automatically considers all the characteristics of a BPM project reflecting it on the clusters to estimate the effort for a new project.

In a second experiment we have considered all 44 projects from the training base and 4 were reserved forming the test basis, as mentioned in Section 4. The expectation was that by considering k = 3, the clusters were formed only by a type of classification of projects. Figure 4 shows the projects distribution among the 3 clusters.

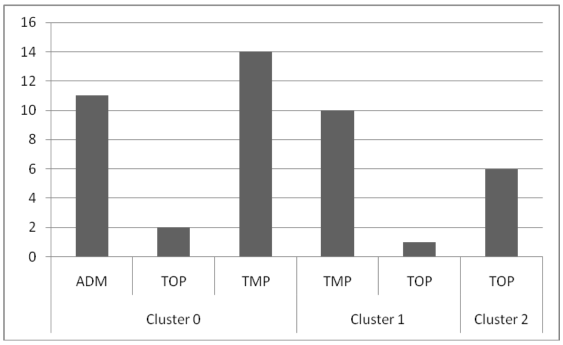

Figure 4 - Distribution of projects by classification

Figure 4 shows that there was a tendency to group projects of the same type as the cluster 0 contains all the ADM projects and cluster 2 assigned only TOP projects. It shows that the application of our method was able to define automatically the project classification. Moreover, our approach found similar projects from different categories, which was not possible in a manual approach.

As described in [Cappelli et al., 2010], experts argue that TMP projects are in the middle ground when it comes to complexity and detail. Our experiment divided the TMP projects fairly between the ADM and TOP, evidencing that we were able to automatically confirm this background knowledge.

In the Software Engineering area, methods such as the Constructive Cost Model (COCOMO) [Boehm, 1981] aim to remove the expert subjectivity during an estimation task. Joseph and Ravichandran [2012] also presented a comparative evaluation of software effort estimation, and proposed a new version of COCOMO to better handle missing information. The results were compared to Rep Tree (a decision tree learner) and K* (an instance-based learner), revealing that K* is well suited for such estimation problems.

Deng and Purvis [2009] used another instance-based learner (KNN) for software effort estimation, and confirmed the difficulty of this task due to unique factors for each project considered. They also concluded that it is possible for algorithms to make new findings. In the context of process automation, Aysolmaz *et al*. [2013] propose a model that takes behavioral, organizational, functional and informational perspectives into account to effectively predict the automation effort. Moreover, the works of Menzies *et al*. [2006] and Dejaeger *et al*. [2012] compared different estimation methods, however focusing on software effort estimation in particular domains.

6. Conclusion

This paper has presented an automatic approach for effort estimation of Business Process Modeling projects through clustering techniques. The proposal was applied in a real scenario, proving its viability in practice. Also, since it relies on existing mining algorithms that are well used in organizations, once the historical data is already collected, the expected difficulty of applying it in practice (that is, using the learned model) is low. The experiments performed with a real dataset showed the ability of grouping similar BPM projects that were already completed. An important side contribution of the proposed approach, besides providing the organization managers with a means to estimate process modeling projects more accurately, is to automatically find the most relevant data that impact on the cost of a new Business Process Modeling project.

Our approach enabled the automatic discovery of patterns without a manual analysis of a large dataset, depending less on the ability of experts and people involved in the scenario of the process. On the other hand, it is important to point out that this approach relies on the existence of historical data collected from the completed projects, which requires the organization management team to establish systematic procedures to collect relevant data from its projects. We argument, however, that the information required from our approach is very simple and feasible to obtain. Moreover, our results show that the approach is beneficial even in scenarios where amount of historical data is not very large; in fact, as more data is considered, more accurate predictions may be obtained.

This work reinforces the tendency in current organizations for establishing management paradigms that rely on historical information and results, such as Business Intelligence [Lim *et al.*, 2013] and Knowledge Management, this last one more specifically towards the Business Process-Oriented Knowledge Management (BPO-KM) [Harmon, 2010].

Through the clustering technique, data is not classified in predefined classes and the creation of the training base takes all the attributes into account, without having to establish the Normalized Cost attribute in advance. The experiments have shown the ability of our approach to automatically find more relevant characteristics than those previously described by experts in literature. We observed that projects considered by experts as being of different classifications did have similarities that are not often perceived in a trivial way. Also, patterns were detected among projects of different expert's classifications. The algorithm has automated the analysis stage, which is not trivial, and led to an automatic and more accurate result in a scenario with real data.

A continuous use of this approach in practice will produce better results, since the concluded projects can feed the training base, increasing the comparative basis. Thus, the technique is expected to produce better results as more evidences are observed.

Perspectives for future work include the consideration of techniques for automatic selection of attributes in order to evaluate how well a particular subset of attributes (project variables) suffices to estimate the effort of BPM projects. In addition, different metrics other than k-means may be applied, such as the cluster median or medoid. An instance-based learner, such as KNN, could also be used to determine to which instance from the cluster is most similar to the new project, aiming to reduce the absolute difference from the actual and the estimated Normalized Cost. It would be a mix of predictive and descriptive tasks. A future comparative work with the model generated by Cappelli *et al.* [2010] is also considered, with a richer dataset and possible combinations between the two approaches.

References

Aysolmaz, B., Iren, D., Demirörs, O. (2013). An Effort Prediction Model Based on BPM Measures for Process Automation. BMMDS/EMMSADm pp. 154-167

van der Aalst, W, M. P. (2012). Process mining. Communications of ACM (8), pp. 76-83

Azevedo, L. G.; Baiao, F.; Santoro, F.; Souza, J. F.; Revoredo; Pereira; Herlain (2009). Identificação de Serviços a partir da Modelagem de Processos de Negócio. In: V Simpósio Brasileiro de Sistemas de Informação, Brasília. V Simpósio Brasileiro de Sistemas de Informação.

Boehm, B.W., (1981). Software Engineering Economics, Prentice-Hall

Cappelli, C.; Santoro, F.M.; Nunes, V.T.N.; Barros, M.O.; Dutra, J.R. (2010). An Estimation Procedure to Determine the Effort Required to Model Business Processes. In: International Conference on Enterprise Information Systems (ICEIS), Funchal, Portugal, pp. 178-184.

Cohn, M. (2005). Agile Estimating and Planning. Prentice Hall.

Dejaeger, K.; Verbeke, W.; Martens, D.; Baesens, B. (2012), Data Mining Techniques for Software Effort Estimation: A Comparative Study. Software Engineering, IEEE Transactions on , vol.38, no.2, pp.375,397, March-April 2012

Diirr, T. ; Azevedo, L. ; Baião, F. ; Santoro, F. M. ; Faria, (2012). F. Practical Approach for Service Design and Implementation. In: IADIS Information Systems, Berlim. Proceedings of IADIS Information Systems. v. 1. p. 1-1.

Dutra, J.R. ; Barros, M. O. ; Santoro, F. M.; Magalhaes, A.; Cappelli, C. ; Nunes, V. T.; Klussmann, F. (2012) The Influence of Human Capabilities in Effort Estimation of Business Processes Modeling Projects. In: Moller, C.; Chaudhry, S. (eds.), Advances in Enterprise Information Systems II, CRC Press

Fayyad, U. M.; Piatetsky-Shapiro, G.; Smyth, P.; Uthurusamy, R. (1996). Advances in Knowledge Discovery & Data Mining. 1 ed. American Association for Artificial Intelligence, Menlo Park, Califórnia.

Ferreira, J. ; Araujo, R. M. ; Baião, F. (2011). Identifying Ruptures in Business-IT Communication through Business Models. Enterprise Information Systems (Print), v. 73, p. 311-325.

Harmon, P. (2010) Business process change: a guide for business managers and BPM and sixsigma professionals. Morgan Kaufmann.

Huang Z. (1998). Extensions to the k-means algorithm for clustering large data sets with categorical values. — Data Mining Knowl. Discov., Vol. 2, No. 2, pp. 283–304.

Indulska, M., Recker, J. C., Rosemann, M., Green, P. (2009). Business process modeling: current issues and future challenges. International Conference on Advanced Information Systems, 8-12, The Netherlands.

Joseph, K. Suresh; Ravichandran, T. (2012). A Comparative evaluation of Software Effort Estimation using REPTree and K* in Handling with Missing Values. Australian Journal of Basic & Applied Sciences; Jul 2012 6(7), pp. 312

Lim, E; Chen, H; Chen, G. (2013). Business Intelligence and Analytics: Research Directions. ACM Transactions on Management Information Systems (TMIS) 3(4):17

Linden, M.; Neuhaus, S.; Kilimann, D.; Bley, T.; Chamoni, P. (2010), Event-Driven Business Intelligence Architecture for Real-Time Process Execution in Supply Chains. 280-290

Purvis, M. and Deng, J. (2009). Software Effort Estimation: Harmonizing Algorithms and Domain Knowledge in an Integrated Data Mining Approach. The Information Science, Discussion Paper Series, Number 2009/05, June 2009, ISSN 1177-455X.

Menzies, T.; Chen, Z.; Hihn, J.; Lum, K. (2006), Selecting Best Practices for Effort Estimation. Software Engineering, IEEE Transactions on , vol.32, no.11, pp.883,895, Nov. 2006

Mike Cohn (2005). Agile Estimating and Planning. Prentice Hall PTR.

Mitchell, T. (1997). Machine Learning, McGraw Hill.

Mutschler, B, and Reichert, M, (2013) *Understanding the Costs of Business Process Management Technology.* In: Business Process Management - Theory and Applications. Studies in Computational Intelligence (444). Springer, pp. 157-194.

Olivera M. (2010). The Importance of Process Thinking in Business Intelligence. IJBIR 1(4):29-46

Scheer, A.-W, (2000). ARIS - Business Process Modelling. Springer, Berlin.

Sekine, J.; Suenaga, T.; Ano, J.; Nakagawa, K; Yamamoto, S. (2009) A Business Process-IT Alignment Method for Business Intelligence. BMMDS/EMMSAD 46-57

Theodoridis, S., Koutroumbas, K. (2006). Pattern Recognition 3rd ed., pp. 635.

Tronto I.F.B., Silva J.D.S., Sant'anna N. (2007) "Comparison of artificial neural network and regression models in software effort estimation" In: Proceedings of International Joint Conference on Neural Networks, Orlando, p.12-17

Weston, R.H., Chatha, K.A., Ajaefobi, J.O. (2004) "Process thinking in support of system specification and selection". Advanced Engineering Informatics, 18(4), pp. 217-229.

Witten, I. Frank, E. (2011). Data Mining: Practical Machine Learning Tools and Techniques, 3nd ed., Morgan Kaufmann.

Woodley, T., Gagnon, S. (2005). BPM and SOA: Synergies and Challenges, In: Proceedings of 6th International Conference on Web Information Systems Engineering, LNCS, vol. 3806, pp. 679–688.F, J.