Discovering Fails in Software Projects Planning Based on Linguistic Summaries

Iliana Pérez Pupo[1(✉)], Pedro Y. Piñero Pérez[1], Roberto García Vacacela[2], Rafael Bello[3], and Luis Alvarado Acuña[4]

[1] Departamento de Investigaciones en Gestión de Proyectos, Universidad de las Ciencias Informáticas, 54830 La Habana, CP, Cuba
{iperez, ppp}@uci.cu

[2] Facultad de Especialidades Empresariales, Universidad Católica De Santiago de Guayaquil, Guayaquil, Ecuador
roberto.garcia@cu.ucsg.edu.ec

[3] Centro de Investigación en Informática, Universidad Central Marta Abreu de las Villas, Santa Clara, Cuba
rbellop@uclv.edu.cu

[4] Departamento de Ingeniería de la Construcción, Universidad Católica del Norte, Antofagasta, Chile
lualvar@ucn.cl

Abstract. Linguistic data summarization techniques help to discover complex relationships between variables and to present the information in natural language. There are some investigations associated to algorithms to build linguistic summaries. But the literature does no report investigations concerned with combination linguistic data summarization techniques and outliers' mining applied to planning of software project. In particular, outliers' mining is a datamining technique, useful in errors and fraud detection. In this work authors present new algorithms to build linguistic data summaries from outliers in software project planning context. Besides, authors compare different outliers' detection algorithms in software project planning context. The main motivation of this work is to detect planning errors in projects, to avoid high cost and time delays. Authors consider that the combination of outliers' mining and linguistic data summarization support project managers to decision-making process in the software project planning. Finally, authors present the interpretation of obtained summaries and comment about its impact.

Keywords: Linguistic data summarization · Outliers mining · Project management · Software project planning

1 Introduction

During the planning of software project, managers continuously have to take decisions to avoid delays and the elevation of project's cost. There are standards and authors that reflect best practices in project management. Some of them stand out: The Capability Maturity Model Integration (CMMI) [1], the guide of Project Management Body of

R. Bello et al. (Eds.): IJCRS 2020, LNAI 12179, pp. 365–375, 2020.
https://doi.org/10.1007/978-3-030-52705-1_27

Knowledge (PMBOK) [2], the ISO 21500 [3], Pressman [4] and Wilson Padua [5]. Despite the existence of these guides, there are still numerous difficulties that are reflected in successful, failed and renegotiated projects. The indexes of successful, failed and renegotiated projects have moved slightly around 29%, 19% and 52% respectively.

The main causes in project failings include planning errors, errors in human resources management and low control and monitoring level [6, 7]. In organizations that develop software projects, planning errors often appear, such as:

- Errors in the cost estimate.
- Errors in the estimation of resources.
- Errors in the estimation of the duration of activities.

Errors manual detection in software project planning constitutes a high time consuming work [8], which affects the projects correct operation. Automatic or semiautomatic detection of errors helps to reduce the cost during projects execution, projects planning and the total cost at the project end.

Planning errors can be identified as derived data from the projects plans. In this sense, it is identified in this investigation early detection of software project planning errors and linguistic data summarization techniques with using outlier mining, will help project managers to correct difficulties. In general, different authors have given their outliers definition [9, 10] among which Hawkins' definition stands out. Hawkins defines in page 2 of [11] that "Outlier is an observation that deviates greatly from the rest of the observations, appearing as a suspicious observation that could have been generated by mechanisms different from the rest of the data" [12].

Nevertheless, it should be perceived that there are not enough publications about outliers' mining in software project planning. In addition, errors presentation and negative impact factors in projects in natural language leads project managers to a better situations understanding and making quick decisions [13].

The objective of this work is to present different algorithms for detect errors in software project planning and construction of linguistic summaries that represent the errors' behavior in this discipline. The work is organized in sections as follows: Second section presents a brief analysis of outliers mining and linguistic data summarization art state. In third section, authors present linguistic data summarization algorithms based on outliers' mining in software project planning processes. The four section aims at the results obtained by the application of proposed algorithms in software project planning environments. Last section presents the conclusions.

2 Algorithm for Discovering Fails in Software Projects Planning Based on Linguistic Summaries

2.1 Brief Analysis of Linguistic Summaries and Outlier Mining

Most of the authors classify the outliers in three categories: punctual outlier's values, collective outliers or contextual outliers [9, 14]. On the other hand, authors classify the outlier detection algorithms following different criteria. In this work, the authors

consider the approach proposed by Aggarwal [9], who establishes the following categories for outliers' detection: supervised, unsupervised and semi-supervised methods.

Unsupervised methods include: statistical techniques, techniques based on proximity and spatial data analysis [15]. Methods based on statistical techniques are based on: descriptive statistics [16], linear regression [17] and in the principal components' analysis [18]. These methods are not efficient when increasing the data set or dimensionality. Proximity-based methods include: distance-based on methods [19], clusters [20] and density-based methods [19].

Distance-based methods usually establish a ranking where the first elements in ranking represent data with high probability of being outliers [21]. In distance-based methods the distance function has a high relevance; for example, different authors refer that Mahalanobis-distance reports better results than Euclidean distance. But data sceneries are different in each case. Authors should test with different methods to discover the best technique. Density-based methods focus on identifying regions of space as a function of their data density, and they are very useful for their interpretation. Among the best known methods of this approach are: local anomalous data factor (LOF) method [22] and local integral correlation method (LOCI). Finally, clustering methods are further subdivided into hierarchical methods, partition-based methods, grid-based methods and constraint-based methods [17]. In this context, the question "what is the best method: cluster algorithm or proximity-based method?" does not have a unique answer. Researchers should analyze data nature in most of the situations and apply empirical tests in every one of the sceneries in order to recognize the algorithms with best results.

On the other hand, supervised methods in outliers mining represent traditional approach based on objects classification by having objects previously classified. In this sense different approaches are presented such as: decision trees, vector support techniques [23], rule-based systems [24], neural networks [25] and the use of metaheuristics [26]. However, these methods usually do not report the best results in outlier's detection because the outliers' mining usually represent a problem with unbalanced classes or with completely unknown classes. For this reason, supervised methods are frequently combined with unsupervised techniques.

In this paper, summaries are generated from outliers. The authors of this work discuss different linguistic data summarization techniques. In [13] defined summary as "using few words to give the most important information about something".

Kacprzyk and Zadrożny are recognized authors in Linguistic data summarization techniques. They define a set of six protoforms that describe linguistic summaries structure and the queries for their search [27]. In this paper, the authors group six protoforms into two basic structures [27, 28] in order to build the linguistic summaries. The elements contained in summary are described in Table 1. Examples:

Table 1. Elements contained in summary.

Elements	Meaning
Q	Represents quantifiers such as: most, some, a few, etc.
R	Represents filters for example: "high planned material resources"
y	Represent the object of study for example "outlier projects"
S	Represents summarizer such as: "very high"
T	Represents measures to evaluate the linguistic summaries quality

First: summaries without filters *Qy's are S*, representing relationships such as:

$$\mathrm{T}\,(\text{Most employees have low pay}) = 0.7$$

Second: summaries with filters *QRy´s are S*, describes relationships such as:

$$\mathrm{T}\,(\text{Most young employees have low pay}) = 0.7$$

There are different approaches to generate linguistic summaries; the simplest protoforms can be obtained by combining fuzzy logic with descriptive statistics or by combining fuzzy logic with sql database query language [29]. But in this work, authors concentrate on summaries generation that represent more complex protoforms and associated to outlier's detection. In this context, basic techniques are not appropriated. More complex protoforms can be built by using mining of fuzzy association rules, Kacprzyk [30] or by using genetic algorithms [31]. These strategies focus on linguistic summaries that represent most of the objects in database. Nevertheless, in this paper authors are in focus of outliers, rare elements and hard difficult detecting elements by using association rules or meta heuristics. For this reason, authors propose a new algorithm in next section.

2.2 A New Algorithm for Generating Summaries from Outliers in Software Project Planning

In this section, an algorithm is proposed for the construction of linguistic summaries from outliers. The following is a hybrid algorithm that combines clustering techniques with distance-based methods to detect outliers and to build linguistic summaries from the outliers detected.

Algorithm's name: *Outlier_Hybrid_LDS.*

Notation

O: outliers set.

B_0: threshold based on the b_0 compact assembly concept.

Ranking_outlier(*S*): returns elements from *S* set, sorted in descending order according to distance.

R: set of linguistic summaries obtained.

SetFuzzyVar: set of linguistic variables, one for each attribute that describes data behavior.

Inputs:

D: data set associated to software project planning.

C: seeded center sets;

Distance (*d*,*S*): distance function from *d* to the set of points *S*.

P: percentile used for the determination of the outliers (the 0.92 percentile was taken).

Q: linguistic variable that describes the quantifiers of the summaries.

Threshold: threshold (ε) is used for the calculation of the *T* and for quantifying the default value as 0.3

ParT-S_norm: Aggregation operators, T-norm pair and S-norm.

begin

1. $O = \{\}$;
2. $clusters$ = Cluster(*D*, *centers*=*C*)
3. *centers* = clusters.centers
4. For each $cluster_i$ in clusters, make
   - 4.1 B_0 = Calculate_threshold($clusters_i$)
   - 4.2 O = clusters.out_centers_B_o

   End of the cycle
5. O = Ranking_outlier(*O*, *P*)
6. O_f = Transforms elements in *O*, into linguistic values by using the *SetFuzzyVar* variables
7. $R = \{\}$ //initializing rule base
8. For each O_{fi} in O_f
   - 8.1 If does not exist rule in *R* that cover O_{fi} (see Definition 1) then
   - 8.2 R_k = Build rule from O_{fi}
   - 8.3 $R = R \cup \{R_k\}$

   End of the cycle
9. S = Build a summary from each rule in *R*
10. S_f = Complete summaries *S* with quantifiers *Q*
11. Calculate truth grade *T* for each summary in S_f
12. Refine summaries S_f using active learning techniques
13. Return S_f sorted, considering *T* values calculated

End

Definition 1: An object X is cover by a rule $G = (P, C)$ with P antecedents and C consequent if and only if for each attribute of $x_i \in X$, $\exists$ ($P_k \in P$ or $C_t \in C$): $x_i \equiv P_k$ or $x_i \equiv C_t$ (operator $\equiv$ means equivalents).

This algorithm could be applied with different clustering methods. Selection of appropriate clustering algorithm depends on data nature. For example, for numerical data could be used *kmeans* cluster algorithm; although, the use of *kmeans* themselves create clusters forming hyper spheres. In each cluster the objects furthest from the center can represent potential outliers.

These objects are detected by using distance methods. In this sense, algorithm can be implemented by using different distance methods, with different threshold values too. In step 10, outlier's data are transformed into linguistic values by using the *SetFuzzyVar* variables defined for each variable and the maximum membership principle. The algorithm continues creating fuzzy rules from detected outliers, and for each fuzzy rule, it creates a candidate linguistic summary. After that, each candidate linguistic summary is completed with quantifiers calculated.

3 Application, Results and Discussion

This algorithm was applied to help projects' managers in software project planning, and to understand projects evolution and projects' human resources behavior. Authors was compared different combinations of algorithms in multiple project management databases. The algorithms are compared by analyzing their performance with the following databases: "mul_plan", "mul_rate", "mul_mix", "alone_rate" and "col_mix" from "170905_gp_eval_proy_fuzzy" Research Database Repository of Project Management Research Group [32]. Each database contains 8430 records with 19 attributes. Different attributes are modified to convert them in outliers. The modification is applied following a supervised way. Later, during test, authors calculate the quality of each algorithm setting in outlier, see Table 2.

Table 2. Description of the databases used in the experimentation.

Database	Meaning	Percent of outliers
alone_rate	rate_rrhh	5% of the modified
mul_plan	serv_plan_quantity, rrhh_plan_quantity, eqp_plan_quantity, inf_plan_quantity, mat_plan_quantity	5% of the modified
mul_rate	rate_equipment, rate_rrhh, rate_service, rate_material	5% of the modified
mul_mix	rate_rrhh, rrhh_plan_quantity, rate_material, mat_plan_quantity, rrhh_plan_quantity, rrhh_real_quantity	5% of the modified
col_mix	rate_rrhh, rrhh_plan_quantity, rate_material	95% of the records in each project transformed to be collective outlier

Table 3. Comparison of multiple algorithms respect to efficacy in outlier detection (numbers joined to each algorithm name, represent parameters).

Group	col_mix	Alone	mult_mix	mul_rate	mult_plan
a	Distance_Mahalanobis_3_0.92 Outlier_Hibrid_LDS_0.92	Outlier_Hibrid_LDS_0.92 Distance_Mahalanobis_3_0.92	Outlier_Hibrid_LDS_0.92	Outlier_Hibrid_LDS_0.92 Kmodr_3	Outlier_Hibrid_LDS_0.92
b	Angle_5_0.95 Kmodr_3_0	Kmodr_3_0 Angle_5_0.95	Distance_Mahalanobis_3_0.92 Angle_5_0.95	Angle_5_0.95 Distance_Mahalanobis_3_0.92	Kmodr_3_0
c	Crossclustering_5_3 Distance_Euclidean_9_0.92	Crossclustering_5_3 Distance_Euclidean_9_0.92	Kmodr_3 Distance_Euclidean_9_0.92 Crossclustering_5_3	Crossclustering_5_3_0	Angle_5_0.95 Distance_Mahalanobis_3_0.92
d				Distance_Euclidean_9_0.92	Crossclustering_5_3 Distance_Euclidean_9_0.92

For each of these databases, 20 partitions are built using cross validation techniques. The algorithms are then compared using non-parametric test of Wilcoxon for two samples related to 95% confidence interval. The following algorithms were used in comparisons: Angle algorithm [33] based on the spatial data analysis approach, cross-clustering algorithm [34] based on partial clustering with automatic estimation of clusters number and outliers' identification, Kmodr algorithm [35] and *Outlier_Hybrid_LDS* based on kmeans (with k = 5), *Distance_Mahalanobis* [36] and *Distance_Euclidean* [9]. Table 3 resumes the comparisons result among the algorithms.

In the comparison, the algorithms groups are organized according to results quality, such as "group a" > "group b" > "group c" > "group d". The algorithms in the same group have no significant differences between them. In most of these databases, *Outlier_Hybrid_LDS* algorithm obtained good results except in the collective anomalous database (col_mix), where *Distance_Mahalanobis* algorithm is slightly superior. The worst result was *Distance_Euclidean_9_0.92*. Regarding efficiency, the best results are found with distance-based methods.

Outlier_Hybrid_LDS detected 450 outliers, representing 95.27% of real outlier's total number. This algorithm generates 44 rules that were unified by considering logical relations and finally 11 linguistic summaries were generated. All summaries were evaluated by using active learning techniques, by project management specialists. The following 5 summaries were identified as the most relevant for project management decisions:

1. Around 50% "outlier projects" have a "very high human resources' plan". T (0.76, 0.44, 0.69, 0.22, 1, 0.62).
2. Around 30% "outlier projects" have "Very high rate of human resources". T (0.5, 0.86, 0.55, 0.06, 1, 0.59).
3. Around 30% "outlier projects" have "Very high material resources' plan". T (0.53, 0.26, 0.27, 0.15, 1, 0.44).
4. Some "outlier projects" with "High material resources plan" have "High rate equipment resource". T (0.78, 0.16, 0.79, 0.33, 1, 0.61).
5. Around 30% "outlier projects" with "High human resources' plan" have "Very high human resources real plan". T (0.95, 0.49, 0.44, 0.23, 1, 0.62).

T vector means the evaluation of summaries by considering the traditional T values defined by Zadeh [37]. In order to get more legible linguistic summaries, algorithm introduces English language words such as "with" and "have" to connect filters and summarizers.

First linguistic summary means, around 50% of "outlier projects" have over-planned the human resources required. The second summary represents that human resources cost of around 30% "outlier projects" are over-planned. The third and fourth summaries represent that some "outlier projects" have over-planned the material resources, and some of them, have over-planned equipment cost rate. From the fifth summary it is interpreted that, in some cases, the number of human resources was planned below the actual number of human resources used. All these summaries help projects managers, correct errors in project management and scheduling.

4 Conclusions

From the results of this investigation, we can reach the following conclusions:

- In used databases, the most detected outliers deal with overestimation of human resources in project tasks.
- Around 30% of outlier projects incur higher costs for using more resources than planned.
- Around 30% "outlier projects" over-planned material resources and some of them contains over-planned equipment's cost-rate.
- Summaries detected from outliers help to project managers to fix errors on project scheduling and to detect project's over-cost.
- In used database the best outlier detection algorithm was the combination of "Kmeans" method with Mahalanobis distance.
- Mahalanobis distance method reports better results than the Euclidean distance in the context of this investigation.
- The experimentation demonstrated that is possible the errors' detection in software project planning from combination of techniques, such as outliers mining and linguistic data summarization.

References

1. Chrissis, M.B., Konrad, M., Shrum, S.: CMMI Guidlines for Process Integration and Product Improvement. Addison-Wesley Longman Publishing Co., Inc., Boston (2003)
2. PMI: A guide to the project management body of knowledge (PMBOK guide) Sixth Edition/Project Management Institute. Project Management Institute, Inc. Newtown Square, Pennsylvania 19073-3299, USA (2017)
3. Grau, N., Bodea, C.-N.: ISO 21500 Project Management Standard: Characteristics, Comparison and Implementation. VShaker Verlag GmbH, Germany (2014)
4. Pressman, R.: Ingeniería del Software Un Enfoque Práctico, 7ma edn. University of Connecticut, Storrs (2010)
5. Pádua, W.: Measuring complexity, effectiveness and efficiency in software course projects. In: Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering, vol. 1, pp. 545–554. IEEE (2010)
6. Gimeno Alonso, J.Á.: Fallos en proyectos: investigación sobre causas generales. In: XVI Congreso Internacional de Ingeniería de Proyectos (2012)
7. Hussain, A., Mkpojiogu, E.O., Kamal, F.M.: The role of requirements in the success or failure of software projects. Int. Rev. Manage. Mark. **6**, 306–311 (2016)
8. Pérez Pupo, I., García Vacacela, R., Piñero Pérez, P., Sadeq, G., Peña Abreu, M.: Experiencias en el uso de técnicas de soft-computing en la evaluación de proyectos de software. Rev. Invest. Oper. **41**(1), 106–117 (2015)
9. Aggarwal, C.H.C.: Datos anómalos analysis. IBM T.J.: Watson Research Center Yorktown Heights, New York, USA (2013)
10. Gupta, M., Gao, J., Aggarwal, C.C., Han, J.: Outlier detection for temporal data: a survey. IEEE Trans. Knowl. Data Eng. **26**, 2250–2267 (2014)

11. Williams, G., Baxter, R., He, H., Hawkins, S., Gu, L.: A comparative study of RNN for outlier detection in data mining. In: Proceedings of 2002 IEEE International Conference on Data Mining, ICDM 2003, pp. 709–712. IEEE (2002)
12. Hawkins, D.M.: Identification of Outliers. Springer, Heidelberg (1980). https://doi.org/10.1007/978-94-015-3994-4
13. Degtiarev, K.Y., Remnev, N.V.: Linguistic resumes in software engineering: the case of trend summarization in mobile crash reporting systems. Proc. Comput. Sci. **102**, 121–128 (2016)
14. Castro Aguilar, G.F., Pérez Pupo, I., Piñero Pérez, P.Y., Martínez, N., Crúz Castillo, Y.: Aplicación de la minería de datos anómalos en organizaciones orientadas a proyectos. Rev. Cubana Ciencias Inf. **10**, 195–209 (2016)
15. Hubert, M., Rousseeuw, P.J., Segaert, P.: Multivariate functional outlier detection. Stat. Methods Appl. **24**, 177–202 (2015)
16. Templ, M., Gussenbauer, J., Filzmoser, P.: Evaluation of robust outlier detection methods for zero-inflated complex data. J. Appl. Stat. 1–24 (2019)
17. Patel, S.P., Shah, V., Vala, J.: Outlier detection in dataset using hybrid approach. Int. J. Comput. Appl. (2015)
18. Bro, R., Smilde, A.K.: Principal component analysis. In: Analytical Methods, pp. 2812–2831 (2014)
19. Kamble, B., Doke, K.: Outlier detection approaches in data mining. J. Eng. Technol. (IRJET) **4**, 634–638 (2017). International Research
20. Ranga Suri, N.N.R., Murty, M.N., Athithan, G.: Research issues in outlier detection. In: Ranga Suri, N.N.R., Murty, M.N., Athithan, G. (eds.) Outlier Detection: Techniques and Applications: A Data Mining Perspective, vol. 155, pp. 29–51. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-05127-3_3
21. Radovanović, M., Nanopoulos, A., Ivanović, M.: Reverse nearest neighbors in unsupervised distance-based outlier detection. IEEE Trans. Knowl. Data Eng. **27**, 1369–1382 (2015)
22. Mishra, S., Chawla, M.: A comparative study of local outlier factor algorithms for outliers detection in data streams. In: Abraham, A., Dutta, P., Mandal, J.K., Bhattacharya, A., Dutta, S. (eds.) Emerging Technologies in Data Mining and Information Security. Advances in Intelligent Systems and Computing, vol. 183, pp. 347–356. Springer, Singapore (2019). https://doi.org/10.1007/978-981-13-1498-8_31
23. Abdulalla, F.Q., Abduljabar, A.S., Shaker, S.H.: A survey of human face detection methods. J. Al-Qadisiyah Comput. Sci. Math. 108–117 (2018)
24. Rajeswari, A., Sridevi, M., Deisy, C.: Outliers detection on educational data using fuzzy association rule mining. In: Proceedings of International Conference on Advanced in Computer Communication and Information Science (ACCIS-14), pp. 1–9 (2014)
25. Jain, L.C., Martin, N.: Fusion of Neural Networks, Fuzzy Systems and Genetic Algorithms: Industrial Applications. CRC Press, Boca Raton (2020)
26. Stützle, T., López-Ibáñez, M.: Automated design of metaheuristic algorithms. In: Gendreau, M., Potvin, J.-Y. (eds.) Handbook of Metaheuristics, vol. 272, pp. 541–579. Springer, Cham (2019)
27. Kacprzyk, J., Zadrożny, S.: Linguistic summarization of the contents of web server logs via the Ordered Weighted Averaging (OWA) operators. Fuzzy Sets Syst. **285**, 182–198 (2016)
28. Donis-Diaz, C., Muro, A., Bello-Pérez, R., Morales, E.V.: A hybrid model of genetic algorithm with local search to discover linguistic data summaries from creep data. Expert Syst. Appl. **41**, 2035–2042 (2014)
29. Kacprzyk, J., Zadrożny, S.: Fquery for access: fuzzy querying for a Windows based DBMS. In: Bosc, P., Kacprzyk, J. (eds.) Fuzziness in Database Management Systems. Studies in Fuzziness, vol. 5, pp. 415–433. Springer, Heidelberg (1995). https://doi.org/10.1007/978-3-7908-1897-0_18

30. Kacprzyk, J., Zadrozny, S.: Linguistic summarization of data sets using association rules. In: The 12th IEEE International Conference on Fuzzy Systems, FUZZ 2003, pp. 702–707. IEEE (2003)
31. Donis-Diaz, C.A., Bello, R., Kacprzyk, J., et al.: Linguistic data summarization using an enhanced genetic algorithm. Czasopismo Tech. 3–12 (2014)
32. Pérez, P.P., Pupo, I.P., Rivero Hechavarría, C.C., Lusardo, C.R., Sosa, R.G., López, S.T.: Repositorio de datos para investigaciones en gestión de proyectos. Rev. Cubana Ciencias Inf. 176–191 (2019). https://gespro.uci.cu/projects/
33. Jimenez, J.: Angle-based outlier detection (2016). https://cran.r-project.org/
34. Hennig, C., Meila, M., Murtagh, F., Rocci, R.: Handbook of Cluster Analysis. CRC Press, Boca Raton (2015)
35. Howe, D.C.: K-Means with simultaneous outlier detection (2016)
36. Rakhe, S.S., Vaidya, A.S.: Enhanced outlier detection for high dimensional data using different neighbor metrics. Int. J. Eng. Sci. (2016)
37. Zadeh, L.A.: A computational approach to fuzzy quantifiers in natural languages. Comput. Math. Appl. 149–184 (1983)