DOI: 10.15514/ISPRAS-2023-35(1)-9

Software project estimation using smooth curve methods and variable selection and regularization methods using a wedge-shape form database

F. Valdés-Souto, ORCID: 0000-0001-6736-0666 <fvaldes@ciencias.unam.mx>
L. Naranjo-Albarrán, ORCID: 0000-0002-9078-6363 <lizbethna@ciencias.unam.mx>
Universidad Nacional Autónoma de México
Ciudad Universitaria, Coyoacán, 04510 Mexico City, Mexico

Abstract. *Context:* The impact of an excellent estimation in planning, budgeting, and control, makes the estimation activities an essential element for the software project success. Several estimation techniques have been developed during the last seven decades. Traditional regression-based is the most often estimation method used in the literature. The generation of models needs a reference database, which is usually a wedge-shaped dataset when real projects are considered. The use of regression-based estimation techniques provides low accuracy with this type of database. *Objective:* Evaluate and provide an alternative to the general practice of using regression-based models, looking if smooth curve methods and variable selection and regularization methods provide better reliability of the estimations based on the wedge-shaped form databases. *Method:* A previous study used a reference database with a wedge-shaped form to build a regression-based estimating model. This paper utilizes smooth curve methods and variable selection and regularization methods to build estimation models, providing an alternative to linear regression models. *Results:* The results show the improvement in the estimation results when smooth curve methods and variable selection and regularization methods are used against regression-based models when wedge-shaped form databases are considered. For example, GAM with all the variables show that the R-squared is for Effort: 0.6864 and for Cost: 0.7581; the MMRE is for Effort: 0.1095 and for Cost: 0.0578. The results for the GAM with LASSO show that the R-squared is for Effort: 0.6836 and for Cost: 0.7519; the MMRE is for Effort: 0.1105 and for Cost: 0.0585. In comparison to the R-squared is for Effort: 0.6790 and for Cost: 0.7540; the MMRE is for Effort: 0.1107 and for Cost: 0.0582 while using MLR.

Keywords: Generalized additive models; LASSO; Software estimation; Effort estimation; Cost estimation; Functional size; COSMIC method

For citation: Valdés-Souto F., Naranjo-Albarrán L. Software project estimation using smooth curve methods and variable selection and regularization methods using a wedge-shape form database. Trudy ISP RAN/Proc. ISP RAS, vol. 35, issue 1, 2023. pp. 123-140. DOI: 10.15514/ISPRAS-2023-35(1)-9

Оценка программного проекта с использованием методов гладких кривых и методов выбора переменных и их регуляризации с использованием базы данных клиновидной формы

Ф. Вальдес-Суто, ORCID: 0000-0001-6736-0666 <fvaldes@ciencias.unam.mx>
Л. Наранхо-Альбарран, ORCID: 0000-0002-9078-6363 <lizbethna@ciencias.unam.mx>

Национальный автономный университет Мексики
Мексика, 04510 Мехико, Койоакан, Университетский городок

Аннотация. *Контекст:* влияние правильной оценки на планирование, составление бюджета и контроль делает действия по оценке важным элементом успеха программного проекта. За последние семь десятилетий было разработано несколько методов оценки. В литературе чаще всего используется традиционный метод оценки, основанный на регрессии. Для создания моделей требуется справочная база данных, которая при рассмотрении реальных проектов обычно представляет собой набор данных клиновидной формы. Использование методов оценки на основе регрессии для этого типа базы данных обеспечивает низкую точность. *Цель:* Оценить и предоставить альтернативу общепринятой практике использования моделей на основе регрессии, выяснив, обеспечивают ли методы гладких кривых и методы регуляризации переменных более высокую надежность оценок, основанных на базах данных клиновидной формы. *Метод:* В предыдущем исследовании использовалась эталонная база данных клиновидной формы для построения модели оценки на основе регрессии. В этой статье используются методы гладких кривых, а также методы выбора переменных и регуляризации для построения моделей оценки, которые представляют собой альтернативу моделям линейной регрессии. *Результаты:* результаты показывают улучшение результатов оценки при использовании методов сглаженной кривой и регуляризации переменных по сравнению с моделями на основе регрессии с использованием клиновидных баз данных.

Ключевые слова: обобщенные аддитивные модели, LASSO, оценка программного обеспечения, оценка усилий, оценка стоимости, функциональный размер, метод COSMIC

Для цитирования: Вальдес-Суто Ф., Наранхо-Альбарран Л. Оценка программного проекта с использованием методов гладких кривых и методов выбора переменных и их регуляризации с использованием базы данных клиновидной формы. Труды ИСП РАН, том 35, вып. 1, 2023 г., стр. 123-140. DOI: 10.15514/ISPRAS–2023–35(1)–9

1. Introduction

Since the appearance of effort estimation in the 50s [1], it has been a relevant topic for researchers in the academy and managers in the industry.

Estimation is one of the crucial activities in software projects [2] It has been identified that inaccurate estimates in the software development industry are one of the most severe problems that cause the failure of software projects [3] because project estimation has an impact on several aspects like planning, budgeting, control, and success of the software projects [4, 5].

Regression-based estimation approaches dominate the literature, as was mentioned by several authors [5-8]. Although other authors have identified a frequent situation in the literature, the regression techniques are not applied correctly, [5, 9-11].

In order to create a regression-based estimation model, a reference database is required; when the database conforms to a broad set of real projects, a wedge-shaped form is presented very often [5, 12] in this type of database, while the x-axis increases, a greater dispersion is observed in the y-axis [12]. This type of dataset was for the first time by [13], presenting high data dispersion, providing low accuracy. Abran [12] mentions that some of the causes that generate the wedge-shaped dataset are, i.e.: “The project data come from organizations with distinct production processes with correspondingly distinct productivity behavior, or the project data represent the development of

software products with major differences, in terms of software domains, nonfunctional requirements, and other characteristics."

This paper explores the use of some smooth curve methods and variable selection and regularization methods like Generalized Additive Models (GAM) and Least Absolute Shrinkage and Selection Operator (LASSO). A comparison of their performance is made, looking to improve the accuracy of the regression-based model developed in the previous study based in the Mexican Software Metrics Association (AMMS) reference database.

The outline of this paper is as follows. Section 2 shows a literature review of software estimation techniques and problems described directly in the models or the database integration. In section 3, the introduction of the fundamental statistical elements used in the paper. Section 4 presents the case study, estimating the effort and cost of the database from AMMS using the Generalized Additive Models (GAM) and Least Absolute Shrinkage and Selection Operator (LASSO). Section 5 discusses the main results of the case study. Finally, the conclusions are discussed in Section 6.

2. Background

2.1 Software Estimation

For more than 70 years since software estimation appeared [1], it has generated interest in the scientific community and the industry as it is a fundamental piece for the success of software projects [1, 2, 14] and has a crucial impact on the planning and budgeting of software projects [15].

After more than seven decades the software estimation research, it is still an open question [16] and presents many difficulties [16]. However, a great variety of estimation techniques [17-19], estimation methods classifications [1, 5, 7, 8, 9], and estimation process topologies [10, 11] have been created. Each statistic technique has specific features that should be considered to make it proper to solve specific problems [11].

The base to create an estimation model is the reference database that should represent the projects to be estimated. Any estimation model possesses a strong relationship with the input data employed to generate the model: "No cost estimation model (or any other model, come to that) will predict well if it is asked to predict effort for projects that are substantially different in nature to the projects on which the model was built" [9]. A lot of weakness in the databases has been identifying in the literature by several authors [6, 9, 15[.

When an estimation model is generated, there is a need to integrate a reliable reference database based on past completed projects. This database allows identifying relationships between different variables [19] (cost drivers) corresponding to the information of the project. According to Carbonera et al. [15], "most studies (71.67%) use multiple cost-drivers rather than priorate a specific one".

Even when several cost drivers are used, several authors identify the functional size as a critical factor to be included in the reference database [1, 20-24]. This situation makes a sound because "nowadays, the only feature of the software that could be defined in a consensual mode and, in consequence, measured in a standard way is the functional size" [25].

It is important that except for the functional size, most of the other drivers are descriptive or qualitative rather than quantitative, p.e. programming language, primary database, primary operating system, software life cycle, etc. In consequence, the estimation-based on functional size does not represent all the cost estimations for the projects and includes an uncertainty degree derived from the other cost drivers.

When a database is integrated over real projects using as independent variables the functional size, a wedge-shaped dataset is usually observed. In a wedge-shaped dataset, a greater dispersion is observed in the y-axis while the x-axis increases (see Fig. 1), some of the causes that generate the wedge-shaped dataset identified by Abran [12] are "The project data come from organizations with distinct production processes with corresponding distinct productivity behavior, or the project data

represent the development of software products with major differences, in terms of software domains, nonfunctional requirements, and other characteristics."

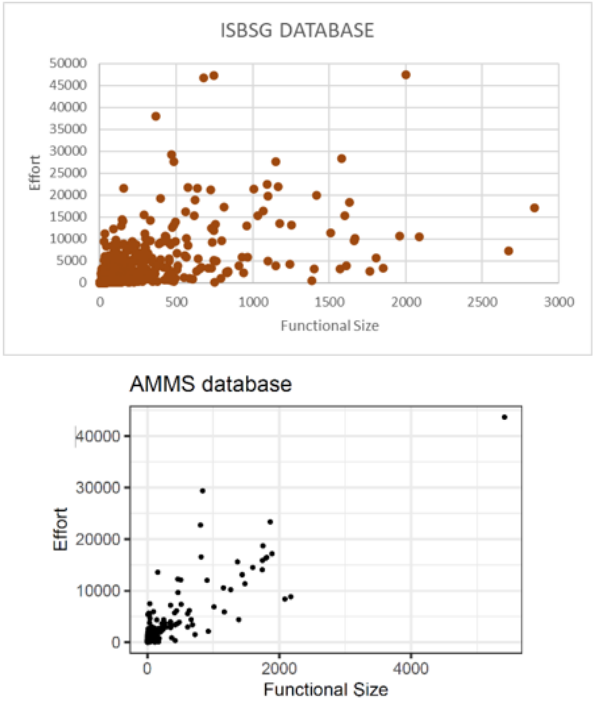

Fig. 1. Wedge-shaped dataset

As there is a high dispersion in a wedge-shaped dataset, the specific features for regression-based models are not accomplished [5]. The use of regression-based estimation techniques may provide low accuracy frequently or may present a cut-off for the accuracy, especially if methods are not adequately applied.

In particular, the software estimation literature reviewed it is not founding the use of smooth curve methods and variable selection and regularization methods. This paper introduces and compares these methods to evaluate their performance with a wedge-shaped form dataset.

2.2 Estimated models performance comparison

In the literature, it has been sought to have a quantitative way of evaluating the performance of estimated models, mainly based on the differences between the real values and the estimated values. Different criteria have been used that determine the confidence of the models used [26-29]. Among the most used criteria in the literature are:

- Coefficient of Determination (R2),
- R2 adjusted,
- Mean Magnitude of Relative Error (MMRE),
- Median Magnitude of Relative Error (MdMRE),
- Standard Deviation of MRE (SDMRE), and
- Prediction level, PRED (x%).

2.2.1 Cross-Validation

A cross-validation framework is considered to validate the results. Specifically, the dataset is randomly split into a training subset composed of 80% of the software projects, and the remaining 20% of the software projects constitute the testing subset. This procedure is repeated independently 500 times, and the results are then averaged.

3. Statistical fundamental elements in estimation

3.1 Smooth Curve Methods

Linear regression models have been studied in Software estimation [5-9], as in many other areas. In Software estimation, the effect of Functional Size on effort or cost is often not linear. In this section, we give a general overview of some statistical methods that allow smooth curve approximations and their properties; we focus on generalized additive models (GAM) and use the regularization and variable selection method LASSO, see [30-33], among others.

3.2 Generalized additive models (GAM)

GLM was proposed by Nelder and Wedderburn (1972) [40], and they extended the multiple linear regression model (MLR) or linear model to include models for binaries and counts data, among others. The GLM is defined by three components [35]:

1) First, the random component Y, with mean $E[Y] = \mu$, where the variable Y has a distribution in the exponential family.
2) Second is the systematic component, where the variables $x_1, x_2, \dots, x_p$ produce a linear predictor $\eta = \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p$.
3) Third, the link function $g(\cdot)$, that link the random and systematic components, $g(\mu) = \eta$. The required properties of $g(\cdot)$ are strict monotonicity and being twice differentiable in the range of μ.

In GAM, the systematic component η is defined as a sum of smooth functions of the independent variables, $\boldsymbol{x} = (x_1, x_2, \dots, x_p)$:

$$\eta = f_1(x_1) + f_2(x_2) + \cdots + f_p(x_p)$$

Usually, the intercept is included as $f_1(x_1) = \beta_0$ because the f_k are centered for identifiability purposes. The effects of the covariates are assumed additive. The functions f_k are estimated by smoothers.

In the particular case of Y being a random variable with normal distribution, $Normal(\mu, \sigma^2)$, the GAM reduces to the additive model, where the relationship between the mean $E[Y] = \mu$ and the linear predictor $\eta = f_1(x_1) + f_2(x_2) + \cdots + f_p(x_p)$ is defined by the identity link function $\mu = \eta$.

Note that the additive model reduces to the MLR model when the smothers are defined as $f_k(x_k) = \beta_k x_k$.

3.2.1 Smoothing Methods

The smoother functions f_k allow to extend the linear predictor to other sophisticated non-linear curves, the most common are the following, see more details in [36]:

1) *Polynomial regression* extends linear regression and adds extra predictors by raising each one to a power.
2) *Step functions* cut the range of x into k distinct regions producing a quantitative variable, and then fitting a piecewise constant function.
3) *Basis function* consists of having a family of functions or transformations that are applied to x.

4) *Regression splines* involve dividing the range of x into k distinct regions, and within each one, a polynomial function is fitted.
5) *Smoothing splines* are similar to regression splines; they result from minimizing a residual sum of squares criterion subject to a smoothness penalty.
6) *Local regression* is similar to splines, but the regions are allowed to overlap in a smooth way.

3.2.2 Inference and Prediction

In order to fit the generalized additive models, the criterion is to maximize a penalized log-likelihood, or equivalently, minimize a penalized of the least squared errors.

3.3 Least Absolute Shrinkage and Selection Operator (LASSO)

The common selection variable methods retain a subset of the predictor variables and discard the rest; however, this subset selection often exhibits high variance, and it doesn't reduce the prediction error of the full model. Shrinkage methods, consisting of regularization and selection variables, do not suffer as much from high variability.

The LASSO (least absolute shrinkage and selection operator) is a shrinkage method. The LASSO coefficients are defined by

$$\boldsymbol{\beta} = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum\nolimits_{j=1}^{p} \beta_j x_{ij} \right)^2,$$

$$\text{subject to } \sum_{j=1}^{p} |\beta_j| \leq t.$$

Making t sufficiently small will cause some of the coefficients to be exactly zero, making LASSO like a selection variable method. Choosing t larger than $\sum_{j=1}^{p} |\hat{\beta}_j|$, where $\hat{\beta}_j$ is the least-squares estimates, then the LASSO results in these $\hat{\beta}_j$'s. See [36] and [32] for more details.

When the independent variables belong to predefined groups, for instance, a collection of dummy variables representing the levels of a categorical variable is desirable to shrink and select the group members, to have all coefficients within a group become nonzero or zero simultaneously. The algorithm needed for these cases is the Group LASSO method [37].

In GAM is possible to apply regularization and variable selection methods, see, and particularly to use LASSO, see [38].

4. Case Study

In this section, the analysis of the Effort and Cost estimation models based on the Mexican reference database is described. For detail information about the database conformation see Table 1.

Table 1. Summary of database information.

Variable or Drivers	Effort N = 390
Effort (N = 390)	454.1 (184.8 – 1457.9)
Cost (N = 387, with 3 missing data)	71,067 (28,522 – 226,468)
Functional size	16.70 (6.96 – 125.42)
Type of organization: • private (reference) • governmental	 302 (77.44%) 88 (22.56%)

Development: • maintenance (reference) • new	 323 (82.82%) 67 (17.18%)
Capacity of development: • area inter of systems (reference) • outsourcing or project of key on hand	 315 (80.77%) 75 (19.23%)
Architecture: • client/server (reference) • development web • multilayers • other	 171 (43.85%) 122 (31.28%) 86 (22.05%) 11 (2.82%)
Language: • C# or PHP (reference) • JAVA/J2EE • C++ • other or non-specified • ASP.NET • VisualBasic6	 103 (26.41%) 76 (19.49%) 105 (26.92%) 47 (12.05$) 2 (0.51%) 57 (14.62%)
Operative System: • windows XP or Linux (reference) • UNIX, windows NT, or other • windows 7/8, windows mobile, or windows vista • windows	 215 (55.13%) 59 (15.13%) 46 (11.79%) 70 (17.95%)
Data base: • POSTGRESTSQL, MySQL, or non-specified (reference) • INFORMIX • ORACLE • SQLSERVER	 178 (45.64%) 96 (24.62%) 17 (4.36%) 99 (25.38%)
Process framework: • CMMI (reference) • MAAGTICSI or RUP • other	 326 (83.59%) 9 (2.31%) 55 (14.10%)
Life cycle: • cascade (reference) • Iterative/agile	 328 (84.10%) 62 (15.90%)
Certification of quality model: • yes (reference) • no	 348 (89.23%) 42 (10.77%)
Size of organization: • ≥ 500 employees (reference) • 251-500 employees • Micro and small	 332 (85.13%) 43 (11.02%) 15 (3.85%)

4.1 GAM with LASSO

In the GAM, for the predictor or independent variables we used the functional size, and other categorical variables. As response or dependent variables, we used Effort and Cost in two different analyses. We used the logarithm transformation for the functional size, effort, and cost. Considering

the multicollinearity and the existence of not significant variables, we applied variable selection methods using LASSO looking to integrate some categories.

The models' generation was made using the software R, defining specific code to calculate all the statistic values. The R libraries *mgcv* [38], *grplasso* [39] and *plsmselect* were used for the GAM, the LASSO linear regression for categorical variables, and the GAM with LASSO, respectively.

For the Effort, the results showed that the statistically significant variables are the logarithm of Functional size, Development, Architecture, Language, Operative System, Data Base, Certification of the quality model, and Size of Organization. On the other hand, the variables identified as statistically no significant like: Organization, Capacity of development, Process framework, and Cycle of life, were deleted, i.e., the model is the following:

$$\log(Effort) = \beta_0 + f(\log(Functional\ size)) + \beta_{devlop} Development + \beta_{archi} Architecture + \beta_{lang} Language + \beta_{os} Operative\ system + \beta_{dbase} Data\ base + \beta_{certif} Certification + \beta_{sizeorg} Size\ of\ organization.$$

For the Cost, the results showed that the statistically significant variables are the logarithm of Functional size, Type of organization, Capacity of development, Architecture, Language, Operative system, Data base, and Certification of the quality model. On the other hand, the variables identified as statistically no significant like: Development, Process framework, Cycle of life, and Size of organization were deleted, i.e. the model is the following:

$$\log(Cost) = \beta_0 + f(\log(Functional\ size)) + \beta_{typeorg} Type\ of\ organization + \beta_{capdevelop} Capacity\ of\ development + \beta_{archi} Architecture + \beta_{lang} Language + \beta_{os} Operative\ system + \beta_{dbase} Data\ base + \beta_{certif} Certification.$$

The models' estimated parameters are shown in Tables 2 and 3, respectively, for Effort and Cost. The first columns show the category or variable names associated with the corresponding parameter. Column two shows the estimates parameters. The last column refers to the standard errors. The fourth columns display the p-values related to the test $H_0: \beta = 0$ vs. $H_a: \beta \neq 0$ for each parameter.

Table 2. Summary of the estimated parameters for Effort by using GAM with LASSO.

Response: Effort Coefficients:	**Estimate**	**Standard Error**	**p-value**
Intercept	6.1488	0.1110	<0.0001
Development: new (ref.: maintenance)	0.1854	0.1423	0.1935
Architecture: (reference: client/server or multilayer)	‘’	‘’	‘’
Architecture: development web or other	-0.2226	0.1151	0.0538
Language (reference: C# or PHP or other or non-specified)	‘’	‘’	‘’
Language: JAVA/J2EE or ASP.NET	0.2631	0.1383	0.0579
Language: C++	-0.0418	0.1125	0.7100
Language: Visual Basic 6	-0.1314	0.1746	0.4521
Operative System (reference: windows XP or Linux)	‘’	‘’	‘’
Operative System: UNIX, windows NT, or other	-0.5843	0.1716	0.0007
Operative System: windows 7/8, windows mobile, or windows vista	0.6007	0.1514	<0.0001
Operative System: windows	-0.5830	0.1469	<0.0001
Data base (reference: POSTGRESTSQL, MySQL, SQLSERVER or non-specified)	‘’	‘’	‘’
Data base: INFORMIX	0.5669	0.1206	<0.0001
Data base: ORACLE	0.1568	0.2540	0.5374
Certification of quality model: no (ref: yes)	0.6068	0.2771	0.0291
Size of organization (reference: ≥ 500 employees)	‘’	‘’	‘’
Size of organization: 251-500 employees	0.2664	0.2378	0.2632
Size of organization: Micro and small	-0.1828	0.2549	0.4735

In Table 3, column 2, line 12, the logarithm of the cost decreases by 0.7181 units (estimated parameters equal to -0.7181) if the operative system is UNIX or Windows NT compared to the operative system LINUX or Windows XP, which is the reference for this categorical variable (line 11). In contrast, Table 3, column 2, line 18, increases 0.8341 units (estimated parameter equal to 0.8341) if the certification of the quality model is "No" in comparison to the "Yes" (category of reference for this variable).

Table 3. Summary of the estimated parameters for Cost by using GAM with LASSO.

Response: Cost Coefficients:	**Estimate**	**Standard Error**	**p-value**
Intercept	11.1182	0.1128	<0.0001
Type of organization: governmental (ref.: private)	0.7876	0.2850	0.0060
Capacity of development: outsourcing or project of key on hand (ref.: area inter of systems)	0.6206	0.4121	0.1329
Architecture: (reference: client/server or multilayer)	‘’	‘’	‘’
Architecture: development web or other	-0.4002	0.1179	0.0007
Language (reference: C# or PHP or other or non-specified)	‘’	‘’	‘’
Language: JAVA/J2EE or ASP.NET	0.2504	0.1421	0.0788
Language: C++	-0.0671	0.1166	0.5649
Language: VisualBasic6	-0.0375	0.1834	0.8380
Operative System (reference: windows XP or Linux)	‘’	‘’	‘’
Operative System: UNIX, windows NT, or other	-0.7181	0.2822	0.0113
Operative System: windows 7/8, windows mobile, or windows vista	0.6960	0.1556	<0.0001
Operative System: windows	-0.5297	0.1519	0.0005
Data base (reference: POSTGRESTSQL, MySQL, SQLSERVER or non-specified)	‘’	‘’	‘’
Data base: INFORMIX	0.6397	0.1239	<0.0001
Data base: ORACLE	0.0451	0.2879	0.8753
Certification of quality model: no (ref: yes)	0.8341	0.2341	0.0004

Additionally, there are no differences in the logarithm of the cost for the category C++ of language since the p-value is 0.5649 (Table 3, column 4, line 9), which means that there are significant differences between the language C++ and the language C# or PHP (reference category). However, there are no differences if the architecture development web or other (Table 3, column 4, line 5) in comparison to the architecture client/server or multilayer since the p-value is 0.0007.

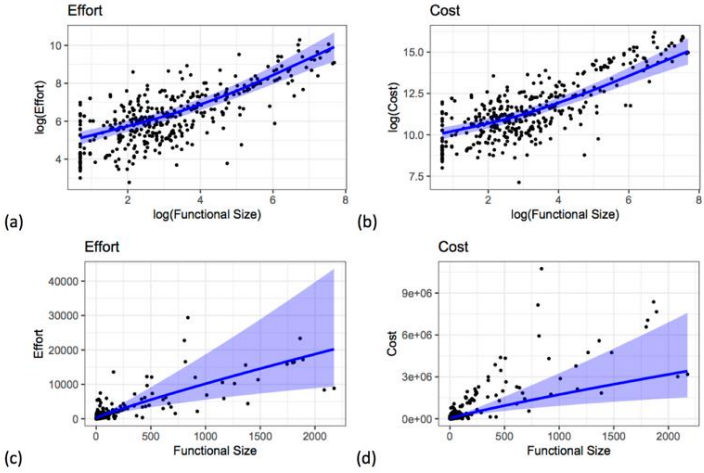

Fig. 2. Fitted line and 95% confidence intervals (shades) for (a) Effort and (b) Cost in logarithmic scale, and (c) Effort and (d) Cost in real scale, by using GAM with LASSO

Fig. 2 shows the fitted lines for reference categories for the categorical independence variables. The shades on the graph provide the 95% pointwise confidence interval for the fitted. To return to the original scale of Y, an exponential function is applied to the predicted values obtained from the model.

From results in Table 2, the estimated model for Effort is:

$$\begin{aligned}\log(Effort) = {} & 6.1488 + s(\log(Functional\ size), 2.46) + 0 * Devlop(maintenance) \\ & + 0.1854 * Devlop(new) + 0 * Archi(client\ server\ \ or\ multilayer) \\ & - 0.2226 * Archi(develpment\ web\ or\ other) + 0 \\ & * Lang(C\#, PHP\ or\ other) + 0.2631 * Lang(JAVA\ J2EE\ or\ ASP.NET) \\ & - 0.0418 * Lang(C++) - 0.1314 * Lang(Visual\ Basic\ 6) + 0 \\ & * OS(windows\ XP\ or\ Linux) - 0.5843 * OS(UNIX, windows\ NT\ or\ other) \\ & + 0.6007 * OS(windows\ 7{,}8, windows\ mobile\ or\ windows\ vista) - 0.5830 \\ & * OS(windows) + 0 \\ & * Dbase(POSTGRESTSQL, MySQL, SQLSERVER\ or\ non\ specified) \\ & + 0.5669 * Dbase(INFORMIX) + 0.1568 * Dbase(ORACLE) + 0 \\ & * Certif(yes) + 0.6068 * Certif(no) + \ 0 * Sizeorg(\geq 500\ employees) \\ & + 0.2664 * Sizeorg(251 - 500\ employees) - 0.1828 \\ & * Sizeorg(Micro\ or\ small).\end{aligned}$$

From results in Table 3, the estimated model for Cost is:

$$\begin{aligned}\log(Cost) = {} & 11.1182 + \ s(\log(Functional\ size), 2.5) + 0 * Typeorg(private) + 0.7876 \\ & * Typeorg(governmental) + 0 * Capdevelop(are\ inter\ of\ systems) \\ & + 0.6206 * Capdevelop(outsourcing\ or\ project\ of\ key\ on\ hand) + 0 \\ & * Archi(client\ server\ or\ multilayer) - 0.4002 \\ & * Archi(development\ web\ or\ other) + 0 * Lang(C\#, PHP\ or\ other) \\ & + 0.2504 * Lang(JAVA\ J2EE\ or\ ASP.NET) - 0.0671 * Lang(C++) \\ & - 0.0375 * Lang(Visual\ Basic\ 6) + \ 0 * OS(windows\ XP\ or\ Linux) \\ & - 0.7181 * OS(UNIX, windows\ NT\ or\ other) + \ 0.6960 \\ & * OS(windows\ 7{,}8, windows\ mobile\ or\ windows\ vista) - 0.5297 \\ & * OS(windows) + 0 \\ & * Dbase(POSTGRESTSQL, MySQL, SQLSERVER\ or\ non\ specified) \\ & + \ 0.6397 * Dbase(INFORMIX) + 0.0451 * Dbase(ORACLE) + 0 \\ & * Certif(yes) + \ 0.8341 * Certif(no).\end{aligned}$$

Table 4 and Table 5 depict the results related to the smooth function $f(\log(Functional\ size))$, in the same format as Tables 2 and 3. Fig. 3 shows the estimated effect of the functional size in the logarithmic scale, as a solid curve, with its 95% confidence limit as dashed lines. Note that the degree of smoothness of the corresponding $f(functional\ size)$ is 2.46 for Effort and 2.5 for Cost. This means that in both cases, the dimension of the smoother is around 2.5.

Table 4. Spline-based smooths for Effort using GAM with LASSO

Approximate significance of smooth terms			
	Effective Degrees of freedom	**F statistic test**	**p-value**
s(log (Functional size))	2.46	100.5	<0.0001

Table 5. Spline-based smooths for Cost using GAM with LASSO

Approximate significance of smooth terms			
	Effective Degrees of freedom	**F statistic test**	**p-value**
s(log (Functional size))	2.5	87	<0.0001

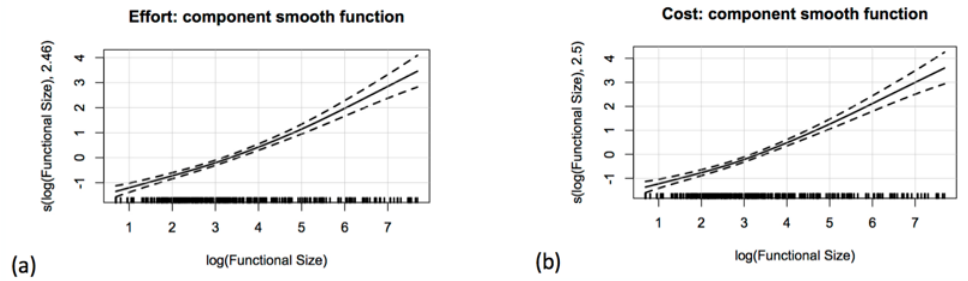

Fig. 3. Component smooth function for the logarithm of Functional Size for the (a) Effort and (b) Cost by using GAM with LASSO.

In GAM, a diagnostic of the residuals, similar to linear regression, must be done. After fitting the model, a diagnostic of the residuals is done to check if the fitted model and assumptions are consistent with the observed data. We used rescaled residuals and graphs to identify homoscedasticity, normality, and influential outliers. Fig. 4 shows the graphs for the residuals for the fitted model for productivity and cost. The quantile-quantile normal plots graphs (a) and (c) visually indicate normality because most dots follow the identity line pattern. The fitted values against the residuals (graphs (b) and (d) in the right) show evidence of constant variance because the dots do not show patterns, which means they show homoscedasticity (constant variance). Moreover, there is no evidence of outliers since there are no residuals with larger values.

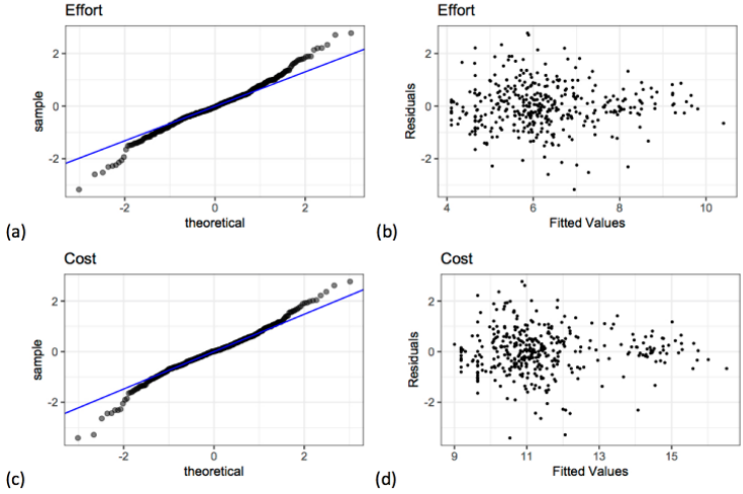

Fig. 4. Residuals for Effort (a, b) and Cost (c, d) by using GAM with LASSO. (a) and (c) quantile-quantile plots to review normal distribution. (b) and (d) residuals vs. fitted values plots to review constant variance

5. Discussion

Three pairs of estimation model techniques were evaluated for comparison purposes, considering the raw data in wedge-shaped form, functional size as the independent variable, and the effort and cost as dependent variables (Fig. 5). The previous study developed the first technique [5], applying a linear regression model (MLR) considering the correct statistical principles and assumptions. The second technique was applying a smooth curve method known as the generalized additive model (GAM). The third technique improved the second approach, using variable selection and regularization methods LASSO (GAM with LASSO), aiming to avoid variables that may be

redundant or irrelevant for predicting the dependent variable, in consequence gathering sparse or simpler models.

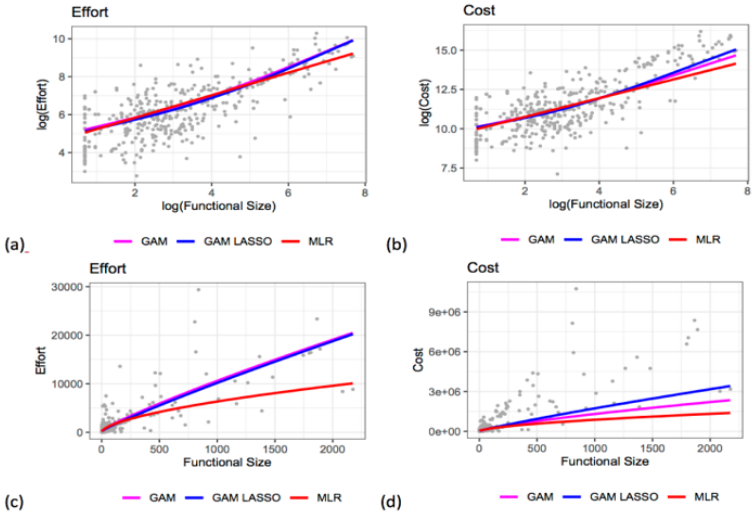

Fig 5. Comparison of the fitted lines for different models. (a) Effort and (b) Cost in logarithmic scale, and (c) Effort and (d) Cost in real scale, by using MLR, GAM, and GAM with LASSO

Tables 6 and 7 show some criteria used to compare the multiple linear regression (MLR), GAM, and GAM with LASSO. Notice that results from GAM and GAM with LASSO are similar. However, GAM with LASSO does not include some variables that are not statistically significant, getting reduced models only with the significant variables.

Table 6. Summary of the criteria for Effort

EFFORT	MLR	GAM	GAM with LASSO
In logarithmic Scale			
R2	0.6790	**0.6864**	0.6836
R2 adjusted	0.6579	0.6645	**0.6705**
MAE	0.6290	**0.6204**	0.6282
MMRE	0.1107	**0.1095**	0.1105
MdMRE	0.0737	0.0743	**0.0730**
SDMRE	0.1356	**0.1340**	0.1346
PRED 25%	0.9025	**0.9051**	**0.9051**
In original Scale			
MAE	773.3	**735.4**	745.0
MMRE	0.8768	0.8594	**0.8557**
MdMRE	0.4353	**0.4345**	0.4434
SDMRE	1.2119	**1.1302**	1.1558
PRED 25%	0.2743	**0.3051**	0.2948

The results for the GAM with all the variables show that the R-squared is for Effort: 0.6864 and for Cost: 0.7581; the R-squared adjusted is for Effort: 0.6645 and for Cost: 0.7413; MAE is for Effort: 0.6204 and for Cost: 0.6355; the MMRE is for Effort: 0.1095 and for Cost: 0.0578. The results for the GAM with LASSO show that the R-squared is for Effort: 0.6836 and for Cost: 0.7519; the R-

squared adjusted is for Effort: 0.6705 and for Cost: 0.7422; MAE is for Effort: 0.6282 and for Cost: 0.6404; the MMRE is for Effort: 0.1105 and for Cost: 0.0585.

Table 7. Summary of the criteria for Cost

COST	MLR	GAM	GAM with LASSO
In logarithmic Scale			
R2	0.7540	**0.7581**	0.7519
R2 adjusted	0.7377	0.7413	**0.7422**
MAE	0.6401	**0.6355**	0.6404
MMRE	0.0582	**0.0578**	0.0585
MdMRE	0.0414	**0.0415**	0.0434
SDMRE	0.0756	**0.0749**	0.0759
PRED 25%	0.9896	**0.9896**	0.9870
In original Scale			
MAE	259498.2	256972.6	**211025.5**
MMRE	0.9113	**0.8993**	0.9575
MdMRE	0.4269	**0.4301**	0.4468
SDMRE	2.2944	2.3511	**1.3845**
PRED 25%	0.2894	**0.3023**	0.2945

5.1 Cross Validation

A cross-validation framework is considered to validate the results. Specifically, the models are fitted using the training subset, and the testing subset is used to predict. Different criteria are computed for the estimated curves in the training subset and then calculated for the predictions in the testing subset. This procedure is independently repeated 500 times, and the results are then averaged.

Table 8. Means and standard deviations of criteria by using MLR, GAM and GAM with LASSO, under the cross-validation scheme for Effort.

Training Data Set			
EFFORT	**MLR**	**GAM**	**GAM with LASSO**
EFFORT	***In logarithmic Scale***	***In logarithmic Scale***	***In logarithmic Scale***
R2	0.6827 ±0.0167	**0.6915 ±0.0169**	0.6869 ±0.0168
R2 adjusted	0.6621 ±0.0178	0.6699 ±0.0180	**0.6703 ±0.0175**
MAE	0.6255 ±0.0153	**0.6145 ±0.0153**	0.6232 ±0.0150
MMRE	0.1100 ±0.0029	**0.1085 ±0.0029**	0.1096 ±0.0028
MdMRE	0.0748 ±0.0032	0.0740 ±0.0034	**0.0740 ±0.0030**
SDMRE	0.1346 ±0.0032	**0.1327 ±0.0033**	0.1337 ±0.0032
PRED 25%	0.9065 ±0.0084	0.9073 ±0.0086	**0.9074 ±0.0088**
EFFORT	***In original scale***	***In original scale***	***In original scale***
MAE	767.2 ±54.4	**717.8 ±51.8**	735.3 ±49.75
MMRE	0.8691 ±0.0407	0.8460 ±0.0416	**0.8447 ±0.0402**
MdMRE	0.4401 ±0.0183	**0.4355 ±0.0185**	0.4402 ±0.0189
SDMRE	1.1930 ±0.1073	**1.0997 ±0.1055**	1.1314 ±0.0996
PRED 25%	0.2934 ±0.0173	**0.3040 ±0.0168**	0.2926 ±0.0180
Testing Data Set			
EFFORT	**MLR**	**GAM**	**GAM with LASSO**
EFFORT	***In logarithmic Scale***	***In logarithmic Scale***	***In logarithmic Scale***
R2	0.6085 ±0.0893	0.6104 ±0.0941	**0.6426 ±0.0728**
R2 adjusted	0.4806 ±0.1185	0.4697 ±0.1302	0.5513 ±0.0925
MAE	0.6819 ±0.0617	0.6750 ±0.0621	**0.6625 ±0.0574**

MMRE	0.1185 ±0.0122	0.1177 ±0.0121	**0.1161 ±0.0118**
MdMRE	0.080 ±0.0107	0.0801 ±0.0109	**0.0793 ±0.0105**
SDMRE	0.1472 ±0.0146	0.1467 ±0.0152	**0.1408 ±0.0125**
PRED 25%	0.8886 ±0.0321	0.8853 ±0.0320	**0.8951 ±0.0314**
EFFORT	***In original scale***	***In original scale***	***In original scale***
MAE	1016.1 ±587.2	946.5 ±384.9	**826.4 ±215.9**
MMRE	1.0057 ±0.2668	0.9834 ±0.2451	**0.9270 ±0.2190**
MdMRE	0.4715 ±0.0547	0.4686 ±0.0576	**0.4678 ±0.0572**
SDMRE	1.8631 ±2.6708	1.6669 ±1.4335	**1.2266 ±0.3749**
PRED 25%	0.2754 ±0.0477	**0.2828 ±0.0478**	0.2758 ±0.0466

Tables 8 and 9 present the summary of the criteria for the Effort and Cost, respectively. Note. that for the training subset, some criteria show better results for GAM, but others show better results for GAM with LASSO; that means that the best estimates result from the GAM or GAM with LASSO compared to the MLR. However, the best results are from the GAM with LASSO for the testing subset. That means GAM with LASSO has the highest predictive capability with this database.

Table 9. Means and standard deviations of criteria by using MLR, GAM and GAM with LASSO, under the cross-validation scheme for Cost

Training Data Set			
COST	**MLR**	**GAM**	**GAM with LASSO**
	In logarithmic Scale	***In logarithmic Scale***	***In logarithmic Scale***
R2	0.7553 ±0.0133	**0.7607 ±0.0132**	0.7536 ±0.0139
R2 adjusted	0.7393 ±0.0141	**0.7439 ±0.0140**	0.7414 ±0.0145
MAE	0.6414 ±0.0162	**0.6347 ±0.0164**	0.6385 ±0.0161
MMRE	0.0582 ±0.0015	**0.0577 ±0.0015**	0.0583 ±0.0015
MdMRE	0.0420 ±0.0019	**0.0420 ±0.0020**	0.0424 ±0.0018
SDMRE	0.0753 ±0.0018	**0.0745 ±0.0019**	0.0756 ±0.0019
PRED 25%	0.9888 ±0.0025	**0.9885 ±0.0027**	**0.9885 ±0.0027**
	In original scale	***In original scale***	***In original scale***
MAE	285293.5 ±34154.5	275634.4 ±33629.2	**210678.5 ±19335.1**
MMRE	0.9194 ±0.0457	**0.9012 ±0.0468**	0.9497 ±0.0550
MdMRE	0.4432 ±0.0194	**0.4468 ±0.0194**	0.4512 ±0.0168
SDMRE	2.4133 ±0.5503	2.3547 ±0.5500	**1.3576 ±0.1892**
PRED 25%	0.2864 ±0.0172	**0.2936 ±0.0174**	0.2928 ±0.0156
Testing Data Set			
COST	**MLR**	**GAM**	**GAM with LASSO**
	In logarithmic Scale	***In logarithmic Scale***	***In logarithmic***
R2	0.6997 ±0.0683	0.7012 ±0.0682	**0.7239 ±0.0603**
R2 adjusted	0.5999 ±0.0911	0.5919 ±0.0936	**0.6583 ±0.0752**
MAE	0.6948 ±0.0656	0.6939 ±0.0667	**0.6712 ±0.0626**
MMRE	0.0625 ±0.0062	0.0625 ±0.0062	**0.0611 ±0.0062**
MdMRE	0.0448 ±0.0065	0.0454 ±0.0065	**0.0449 ±0.0061**
SDMRE	0.0820 ±0.0087	0.0818 ±0.0087	**0.0786 ±0.0076**
PRED 25%	0.9857 ±0.0130	0.9849 ±0.0133	0.9864 ±0.0120
	In original scale	***In original scale***	***In original scale***

MAE	459287.5 ±540631.3	450147.3 ±594499.7	238069.9 ±82057.5
MMRE	1.0683 ±0.3164	1.0532 ±0.3019	**1.0286 ±0.2935**
MdMRE	0.4672 ±0.0591	0.4767 ±0.0615	**0.4754 ±0.0579**
SDMRE	4.0525 ±6.4314	3.8541 ±6.0017	**1.4729 ±0.6324**
PRED 25%	0.2683 ±0.0456	0.2747 ±0.0479	**0.2825 ±0.0481**

6. Conclusions

Software cost/effort estimation has been a relevant topic for more than 60 years in research because of its impact on the industry.

Regression-based estimation approaches have been the more often used technique in the literature, focusing on the estimation model performance comparison. Although, many times, the regression techniques principles are not accomplished.

Additionally, when a database is integrated over real projects and a wedge-shaped form dataset is present, high data dispersion is shown, usually because the project data come from distinct organizations or the project data represent software products with major differences in its characteristics, providing low accuracy in the prediction models generated from the database.

This paper evaluates and provides an alternative to the general practice of using regression-based models. The proposed approach has not been identified in the literature reviewed; it focuses on some smooth curve methods and variable selection and regularization methods like: Generalized Additive Models (GAM) and Least Absolute Shrinkage and Selection Operator (LASSO).

The approach proposed was compared, then the wedge-shaped form database used in a previous study was considered. The performance of the methods generated was evaluated, aiming to improve the accuracy of the MLR model based on the Mexican Software Metrics Association (AMMS) reference database.

A case study is presented to demonstrate how the application of GAM and LASSO over the Mexican Software Metrics Association (AMMS) reference database (wedge-shaped) improves the estimation based on traditional regression-based models (MLR).

In the case of additive models (GAM with normal distribution), the assumptions behind the model are similar to those in multiple linear regression (MLR): residuals must be distributed as Gaussian, being non-correlated and having a constant variance.

This paper used logarithmic transformation to correct problems about normal distribution, constant variance, and influential outliers.

In GAM, the smoother methods extend the linear predictor of generalized linear models (GLM) to other more flexible and non-linear curves, making a more representative model for the data considered in a wedge-shaped database.

The results in this paper show the improvement, providing better accuracy of the generalized additive models (GAM) in comparison to the multiple linear regression (MLR). Moreover, in the cross-validation task, the improvement of the GAM with LASSO on its predictive capability is highest for both dependent variables, Effort, and Cost.

The main contribution of this article is focusing on the generation of estimation models that work better, that is, that offer better precision than those traditionally used, such as simple or multiple linear regression when there are wedge-shaped databases. Additionally, they consider additional drivers, qualitative or quantitative, and optimize them concerning their impact, resulting in simpler models.

Additionally, the explanatory variables should not be correlated to avoid multicollinearity problems and that there are no influential outliers.

Applying the LASSO algorithm, the independent variables are selected, avoiding multicollinearity problems, and choosing those statistically significant, making a sparse model easy to manage and use.

The results show the improvement in the estimation results when smooth curve methods (GAM) and variable selection and regularization methods (LASSO) are used against regression-based models (MLR) when wedge-shaped form databases are considered.

References / Список литературы

[1] Fedotova O., Teixeira L., Alvelos A.H. Software effort estimation with multiple linear regression: Review and practical application. Journal of Information Science and Engineering, vol. 29, issue 5, 2013, pp. 925–945.

[2] Sharma P., Singh J. Systematic literature review on software effort estimation using machine learning approaches. In Proc. of the International Conference on Next Generation Computing and Information Systems (ICNGCIS), 2017: pp. 43-47.

[3] Oliveira A.L.I. Estimation of software project effort with support vector regression. Neurocomputing, vol. 69, issues 13-15, 2006, pp. 1749-1753.

[4] Papadopoulos H., Papatheocharous E., Andreou A.S. Reliable confidence intervals for software effort estimation. In Proc. of the Workshops of the 5th IFIP Conference on Artificial Intelligence Applications & Innovations (AIAI-2009), 2009: pp. 211-220.

[5] Valdés-Souto F., Naranjo-Albarrán L. Improving the Software Estimation Models Based on Functional Size through Validation of the Assumptions behind the Linear Regression and the Use of the Confidence Intervals When the Reference Database Presents a Wedge-Shape Form. Programming and Computer Software, vol. 47, issue 8, 2021, pp. 673-693.

[6] Jørgensen M., Shepperd M. A systematic review of software development cost estimation studies. IEEE Transactions on Software Engineering, vol. 33, no. 1, 2007, pp. 33-53.

[7] Braga P.L., Oliveira A.L.I., Meira S.R.L. Software Effort Estimation using Machine Learning Techniques with Robust Confidence Intervals. In Proc. of the 7th International Conference on Hybrid Intelligent Systems (HIS 2007), 2007, pp. 352-357.

[8] Shin M., Goel A.L. Empirical Data Modeling in Software Engineering Using Radial Basis Functions. IEEE Transactions on Software Engineering, vol. 26, no. 6, 2000, pp. 567-576.

[9] Kitchenham B., Mendes E. Why comparative effort prediction studies may be invalid. In Proc. of the 5th International Conference on Predictor Models in Software Engineering, 2009, article no. 4, 5 p.

[10] Bilgaiyan S., Sagnika S. et al. A systematic review on software cost estimation in Agile Software Development. Journal of Engineering Science and Technology Review, vol. 10, issue 4, 2017, pp. 51-64.

[11] Jørgensen M. Regression Models of Software Development Effort Estimation Accuracy and Bias. Empirical Software Engineering, vol. 9, issue 3, 2004, pp. 297-314.

[12] Abran A. Software Project Estimation: The Fundamentals for Providing High Quality Information to Decision Makers, 1st ed. John Wiley & Sons, 2015, 288 p.

[13] Kitchenham B., Taylor N. Software cost models, ICL Technical Journal, vol. 4, issue 1, 1984, pp. 73-102.

[14] Lee T.K., Wei K.T., Ghani A.A.A. Systematic literature review on effort estimation for Open Sources (OSS) web application development, In Proc. of the Future Technologies Conference (FTC), 2016, pp. 1158-1167.

[15] Carbonera C.E., Farias K., Bischoff V. Software development effort estimation: A systematic mapping study. IET Software, vol. 14, issue 4, (2020, pp. 328-344.

[16] Yadav N., Gupta et al. Comparison of COSYSMO Model with Different Software Cost Estimation Techniques. In Proc. of the International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT), 2019, pp. 1-5.

[17] Gray A.R., MacDonell S.G. A Comparison of Techniques for Developing Predictive Models of Software Metrics. Information and Software Technology, vol. 39, issue 6, 1997, pp. 425-437.

[18] Silhavy R., Prokopova Z., Silhavy P. Algorithmic optimization method for effort estimation. Programming and Computer Software, vol. 42, issue 3, 2016, pp. 161-166 / Сильхавы Р., Попова З., Сильхавы П. Алгоритмический метод оптимизации оценки трудозатрат. Программирование, том 42, вып. 3, 2016

г., стр. 64-71.

[19] Durán M., Juárez-Ramírez R. et al. User Story Estimation Based on the Complexity Decomposition Using Bayesian Networks. Programming and Computer Software, vol. 46, issue 8, 2020, pp. 569-583 / Дуран М., Хуарес-Рамирес Р. и др. Оценка пользовательских историй на основе декомпозиции сложности с использованием байесовских сетей. Труды ИСП РАН, том 33, вып. 2, 2021 г., стр. 77-92. DOI: 10.15514/ISPRAS–2021–33(2)–4.

[20] Bourque P., Oligny S. et al. Developing Project Duration Models in Software Engineering. Journal of Computer Science and Technology, vol. 22, 2007, pp. 348-357.

[21] Laird L.M., Brennan M.C. Software Measurement and Estimation: A Practical Approach, Jonh Wiley & Sons, 2006, 280 p.

[22] Koch S., Mitlöhner J. Software project effort estimation with voting rules, Decision Support Systems, vol. 46, issue 4, 2009, pp. 895-901.

[23] De Lucia, Pompella E., Stefanucci S. Assessing effort estimation models for corrective maintenance through e A.mpirical studies, Information and Software Technology, vol. 47, issue 1, 2005, pp. 3-15.

[24] Hill J., Thomas L.C., Allen D.E. Experts' estimates of task durations in software development projects, International Journal of Project Management, vol. 18, issue 1, 2000, pp. 13-21.

[25] ISO/IEC 14143-1:2007 Standard. Information technology — Software measurement — Functional size measurement — Part 1: Definition of concepts. 2007.

[26] Shepperd M., MacDonell S. Evaluating prediction systems in software project estimation. Information and Software Technology, vol. 54, issue 8, 2012, pp. 820-827.

[27] Foss T., Stensrud E. et al, A simulation study of the model evaluation criterion MMRE. IEEE Transactions on Software Engineering, vol. 29, issue 11, 2003, pp. 985-995.

[28] Myrtveit I., Stensrud E., Shepperd M. Reliability and validity in comparative studies of software prediction models. IEEE Transactions on Software Engineering, vol. 31, issue 5, 2005, pp. 380-391.

[29] Jørgensen M., Halkjelsvik T., Liestøl K. When should we (not) use the mean magnitude of relative error (MMRE) as an error measure in software development effort estimation? Information and Software Technology, vol. 143, 2022, article no. 106784, 5 p.

[30] Hastie T., Tibshirani R., Friedman J. The Elements of Statistical Learning, Data Mining, Inference, and Prediction, 2nd ed. Springer New York, 2009, 745 p.

[31] Yee T.W. Vector Generalized Linear and Additive Models. With an Implementation in R, Springer,, 2015, 613 p.

[32] Hastie T., Tibshirani R., Wainwright M. Statistical Learning with Sparsity The Lasso and Generalizations. Routledge, 2015, 367 p.

[33] Wood S.N. Generalized Additive Models, 2nd ed. Chapman and Hall/CRC, 2017, 476 p.

[34] Hastie T.J., Tibshirani R.J., Sasieni P. Generalized additive models, Routledge, 1990, 352 p.

[35] McCullagh P., Nelder J.A., Enderlein G. Generalized linear models. 2nd ed. Chapman and Hall/CRC, 1989, 532 p.

[36] James G., Witten D. et al, An Introduction to Statistical Learning with Applications in R, 1st ed., Springer, 2013. 440 p.

[37] Yuan M., Lin Y. Model selection and estimation in regression with grouped variables. Journal of the Royal Statistical Society, Series B (Statistical Methodology), vol. 68, issue 1, 2005, pp. 49-67.

[38] Groll A., Hambuckers J. et al. LASSO-type penalization in the framework of generalized additive models for location, scale and shape, Computational Statistics and Data Analysis, vol. 140, 2019, pp. 59-73.

[39] Meier, L., van de Geer S., Bühlmann P., The Group Lasso for Logistic Regression, Journal of the Royal Statistical Society, Series B (Statistical Methodology), vol. 70, issue 1, 2008, pp. 53-7.1

[40] Nelder J., Wedderburn R. Generalized linear models, Journal of the Royal Statistical Society. Series A (General), vol. 135, issue 3, 1972, pp. 370-384.

Information about auth.ors / Информация об авторах

Francisco VALDÉS-SOUTO, Ph.D. in Software Engineering, Associate Professor. Areas of interest: Software Engineering, Software Measurement, Software Estimation, Software, Economy, Software Project Management.

Франсиско ВАЛЬДЕС-СУТО, кандидат наук в области программной инженерии, доцент. Области интересов: разработка программного обеспечения, измерение программного обеспечения, оценка программного обеспечения, программное обеспечение, экономика, управление программными проектами.

Lizbeth NARANJO-ALBARRÁN, Ph.D. in Mathematics, Professor. Areas of interest: Bayesian statistics, biostatistics, generalized linear models, and measurement error models.

Лизбет НАРАНХО-АЛЬБАРРАН, кандидат математических наук, профессор. Области интересов: байесовская статистика, биостатистика, обобщенные линейные модели и модели ошибок измерения.