An Empirical Study of Process–related Attributes in Segmented Software Cost–Estimation Relationships

Juan J. Cuadrado–Gallego [a] Miguel-Ángel Sicilia [a]
Miguel Garre [a] Daniel Rodríguez [b]

[a] *Computer Science Department. Polytechnic School.*
University of Alcalá. Ctra. Barcelona km. 33.6
28871 – Alcalá de Henares, Madrid (Spain)
{jjcg, msicilia, miguel.garre}@uah.es

[b] *Computer Science Department.*
RG6 6AY, University of Reading, UK
d.rodriguez-garcia@rdg.ac.uk

Abstract

Parametric software effort estimation models consisting on a single mathematical relationship suffer from poor adjustment and predictive characteristics in cases in which the historical database considered contains data coming from projects of a heterogeneous nature. The segmentation of the input domain according to clusters obtained from the database of historical projects serves as a tool for more realistic models that use several local estimation relationships. Nonetheless, it may be hypothesized that using clustering algorithms without previous consideration of the influence of well–known project attributes misses the opportunity to obtain more realistic segments. In this paper, we describe the results of an empirical study using the ISBSG–8 database and the EM clustering algorithm that studies the influence of the consideration of two process–related attributes as drivers of the clustering process: the use of engineering methodologies and the use of CASE tools. The results provide evidence that such consideration conditions significantly the final model obtained, even though the resulting predictive quality is of a similar magnitude.

Key words: Parametric software effort estimation, clustering algorithms, software cost drivers, EM algorithm.

Preprint submitted to Elsevier Science *18 April 2005*

1 Introduction

The *Parametric Estimating Handbook* (PEH) (PEI, 2000) defines parametric estimation as "a technique employing one or more cost estimating relationships (CERs) and associated mathematical relationships and logic". These techniques are nowadays widely used to measure and/or estimate the cost associated with software development (Boehm et al., 2000). CERs are mathematical devices that obtain numerical estimates from main *cost drivers* that are known to affect the effort or time spent in development. According to the PEH, these drivers are the controllable system design or planning characteristics that have a predominant effect on system cost. Parametrics uses the few important parameters that have the most significant cost impact on the software being estimated. Nonetheless, even though the final CERs should use only the most significant parameters, it is often also useful to consider other parameters as a foundation for the logics of deriving the mathematical relationships from empirical data. The notion of "cost realism" as described in the PEH clearly points out to this dimension of reasonable and justified usage of data.

One important aspect of the process of deriving models from databases is that of the heterogeneity of data. Heteroscedasticity (non–uniform variance) is known to be a problem affecting data sets that combine data from heterogeneous sources (Stensrud et al., 2002). When using such databases, traditional application of curve regression algorithms to derive a single mathematical model results in poor adjustment to data and subsequent potential high deviations. This is due to the fact that a single model can not capture the diversity of distribution of different segments of the database points. As an illustrative example, the straightforward application of a standard least squares regression algorithm to the points used in the Reality tool of the ISBSG 8 database [1] distribution results in measures of MMRE=2.8 and PRED(.3)=23% (MMRE and PRED measures are discussed later in this paper), which are poor figures of quality of adjustment.

The use of clustering techniques has been described as a solution to provide more realism to parametric models by decomposing the model in a number of sub–models, one per segment, that are used to estimate points that are near them (Garre et al., 2004). Related work includes the use of different clustering approaches to several aspects of software management, including software estimation, software quality and software metrics. Concretely, Xu and Khoshgoftaar (Xu and Khoshgoftaar, 2004) use the *fuzzy c-means* algorithm for variable, the partitioning of the data into a number of clusters based on experiences. Pedrycz and Succi (Pedrycz and Succi, 2005) also use fuzzy c-

[1] http://www.isbsg.org/

means as a tool to derive prototypes related to software code measurements. Dick et al. (Dick et al., 2004) use the same algorithm for a similar setting in a knowledge discovery study. Nonetheless, these approaches do not deal with the heterogeneity of the project databases they use. Lung, Zaman, and Nandi (Lung et al., 2004) have used the numerical taxonomy method for the clustering of software components at several development phases, but these analysis are driven by the structure of the code, which is rarely available in public historical software project databases. Oligny et al (Oligny, 2000) approach estimation studies by the partitioning of the project database into "more homogeneous subsets". This study can be considered as supporting evidence for the segmentation approach described in this paper, even though the partitioning of the data is carried out without using a clustering algorithm. Preliminary data for the use of clustering following the same considerations is described in (Garre et al., 2004).

In spite of the scattered available evidence regarding the practical usefulness of partitioning historical databases, the use of clustering techniques for the problem described has to date the drawback of not being explicative of the composition of the segments, i.e. the many concrete factors regarding characteristics of the development process and context are not considered as determinants of the resulting segments. On the contrary, these techniques apply a "blind" approach to characteristics that may be considered as relevant *a priori*.

In this paper, we explore an alternative technique in which some parameters selected purposely are used *a priori* to drive the process of subsequent clustering. Concretely, a case study using the ISBSG 8 database is described, which evaluates the influence of the pre–configuration of segments according to the two process–related parameters of using methodologies (METHO) and using support tools (CASET). These factors are considered in classical public models of parametric estimation to have an influence in the estimation process, e.g. the COCOMO 81 model considers them under the "modern programming practices" (MODP) and "use of software tools" (TOOLS) attributes Boehm (1981). Its update version COCOMO–II (Boehm et al., 2000) also considers software tools as a cost driver, even though the rating levels have changed due to changes in development technology (Boehm, 1995). In contrast, the definition of "modern programming practices" has even evolved into a broader "mature software engineering practices" term exemplified by the Software Engineering Institute Capability maturity Model (Paulk, 1993) and comparable models such as ISO 9000-3 and SPICE. The cost estimation effects of this broader set of practices are addressed in COCOMO 2.0 [2] via the "Process Maturity" exponent driver.

[2] COCOMO 2.0 was the name of the COCOMO-II model before its definitive release.

Existing proprietary parametric models also take into account these two factors. For example, SEER-SEM [3] describes a feature to optimize the estimations called "development standard" which reflects the use a methodology in module development, and it also uses a parameter that takes into account the use of tools in the development. Other extended proprietary model, PRICE-S [4] also consider these two factors in order to correct the initial estimations. PRICE-S CPLX1 variable captures the CASET parameter together with other factors as personnel skills or familiarity of the personnel with the product. PRICE-S also considers the use of several process models as Waterfall, Spiral, Incremental development and US DoD MIL-STD-2167a process, as an implicit consideration of development methods.

The segmented parametric estimation model described in this paper demonstrates that the partitioning of the project database before the calibration process results in better predictive quality, in addition to constituting by itself an important step into increased cost realism, since it considers the divergences in variance as a characteristic of the model. In addition, the the consideration of specific well–known software attributes enhances the explicative properties of the segmented model. These two elements are relevant as a complement to existing calibration techniques for large and heterogeneous project databases.

The rest of this paper is structured as follows. Section 2 describes conditional parametric models for discrete values. Then, the concrete evaluation of the technique is reported in Section 3. Finally, Section 4 provides conclusions and further directions for research.

2 Segmented Cost–Estimation Relationships

A standard parametric model is obtained from the entire project database using conventional curve regression techniques relating effort or schedule predictions to a number of cost drivers $c_i \in \mathcal{C}$. Expression (1) shows one the most usual concrete models for the relationship between size (expressed in function point estimates) and total effort measured for example in total hours or effort spent.

$$e = a \cdot fp^b \text{ generally } e = f(c_i) \;\; \mathcal{C} = \{c_i\} \tag{1}$$

Segmented models replace the single–equation approach with a collection of mathematical models f_j, each of them associated to the definition of a segment

[3] http://www.galorath.com
[4] http://www.pricesystems.com/

$s_i \in \mathcal{S}$, as expressed in (2).

$$e = f_j(c_i) \;\; j = \gamma(c_i) \;\; \text{with} \;\; \mathit{segment}(f_j) = s_j \tag{2}$$

Segment definitions may be expressed in different ways, depending on the clustering technique used with the project database. The mapping function $\gamma(c_i)$ is responsible for selecting the function for each particular project being estimated, and it proceeds by finding out the segment (cluster) that best characterizes the project under consideration. The use of this model was demonstrated in (Garre et al., 2004), using the EM clustering algorithm with the ISBSG–8 database without any previous explicit breakdown of the data. Our hypothesis in this paper is that this "blind" approach could be improved by a prior decomposition of the historical database of projects according to a set of parameters that are known or believed to affect the effort spent. This kind of procedures (we well call them "tailored (conditional) segmented models" from here on) serves the objective of assessing the actual influence of the selected parameters in the model, and they may eventually lead to clusters that better reflect the database characteristics, i.e. they are a tool to improve the "cost realism" of the models.

A set $\mathcal{P}$ of parameters are selected as factors determining the derivation and use of the parametric model. In this paper we only consider variables with discrete, nominal values, but the approach could be extended to discretized numerical attributes as well.

$$\mathcal{P} = \{p_1 \ldots p_n\} \quad p_i \notin \mathcal{C} \quad \mathit{type}(p_i) = (v_1^i, \ldots v_{k_i}^i) \tag{3}$$

Then, the clustering process is applied to partitions of the original project database obtained from the possible combination of discrete values for all the elements in $\mathcal{P}$, i.e. any of the elements in the cartesian product $p_1 \times \ldots p_n$. Obviously, the number of parameters and labels considered should be kept small to make the procedure viable. In practice, the parameters considered would require a previous assessment of relevance and of their relevance to produce significant partitionings.

3 Evaluation

The parameters selected for the case study are the utilization of CASE tools (CASET) and the application of a methodology (METHO) in the project, which are factors considered in existing estimation models (Boehm, 1981).

For example, Baik et al. (Baik et al., 2002) estimates the influence of CASE tool usage in final effort of 1.5 in productivity.

The ISBSG–8 project database was used for the empirical study. A correlation analysis of these two variables with regards to effort and size measured in function points yield values below ± 0.1, so that there is no evidence of strong dependency. The correlation coefficient CASET-METHO is positive 0.12, which can not be interpreted as a strong dependency either.

It should be noted that the model used here could also be used with other existing defined parametric models. For example, if the COCOMO-II model was selected, the clustering phase should take place before the calibration process is carried out. Moreover, the calibration should be carried out for each of the clusters obtained, and the analysis of influence of cost drivers could be used to discard or include some of them in the final model.

The study reported here attempts to gather evidence about the influence of well–known process or project attributes in the creation of segmented software estimation models. The empirical method proceeds by carrying out two variants of the clustering process. On the one hand, all the projects in the database were used as input to a one–step "blind" clustering process corresponding to expression (2), without considering any process or project attribute. On the other hand, the same process was applied to selected subsets of data ("tailored") corresponding to the pairs of label values of variables METHO and CASET, as expressed in (3). The rest of this section reports the results and discussion.

3.1 Data preparation

The entire ISBSG-8 database [5] containing information about 2028 projects was used as the project database. The database contained information about size, effort and many other project characteristics. The first cleaning step was that of removing the projects with null or invalid numerical values for the fields effort ("Summary Work Effort" in ISBSG-8) and size ("Function Points"). Then, the projects with "Recording method" for total effort other than "Staff hours" were removed. The rationale for this is that the other methods for recording were considered to be subject to subjectivity. For example, "productive time" is a rather difficult magnitude to assess in a organizational context.

Since size measurements were considered the main driver of project effort, the database was further cleaned for homogeneity in such aspect. Concretely, the

[5] http://www.isbsg.org/

projects that used other size estimating method ("Derived count approach") than IFPUG, NESMA, Albretch or Dreger were removed, since they represented smaller portions of the database. The differences between IFPUG and NESMA methods are considered to have a negligible impact on the results of function point counts (NESMA, 1996). Counts based on Albretch techniques were not removed since in fact IFPUG is a revision of these techniques, similarly, the Dreger method refers to the book (Dreger, 1989), which is simply a guide to IFPUG counts.

3.2 Procedure

The unsupervised EM clustering algorithm (Dempster et al., 1977) was selected based on the evidence of its appropriateness for the task reported elsewhere (Garre et al., 2004). Nonetheless, the standard EM implementation used (the one included in the WEKA Java libraries [6]) assumes independent input variables, which seems not justified given the various interrelationships that common project attributes hold with each other. To overcome this potential problem, a variant of the EM algorithm was coded as a C program, introducing correlation matrices in the process of clustering.

The procedure for two parts of the study consisted in the following steps:

- In the "blind" clustering process, the entire database was given to the modified version of the EM algorithm.
- In the "tailored" clustering process, four partitions were prepared from the database, corresponding to each of the possible boolean value combination of the parameters considered. Projects with missing values in any or both of these parameters were not considered, which resulted in a significantly smaller databases. In the case of CASET, only the "Upper CASE" attribute was used. The reason for this was twofold. First, the other CASE categories resulted in small sized categories. And second, the CASE categories were not considered similar enough to consider them together, since "Analysis and Design" tools provide a very different kind of automated support than, for example, coding support tools.

In both parts of the study, the models obtained from regression techniques were subject to cross–validation following standard practices. The data assigned to each cluster was randomly split into two sets called training (t) and validation (v), respectively containing a 70% and a 30% of the data. Then, the measures for each clusters are computed on both sets, as a standard means to validate the goodness of adjustment. The measures of prediction accuracy used were standard MMRE and PRED(.3) which are commonly accepted measures that

[6] http://www.cs.waikato.ac.nz/ ml/weka/

	MMRE	PRED(.3)	a	b
with c.v.	2.81	0.23	7.6	1.07
without c.v.	0.88	0.027	14.5	0.4615

Table 1
Characteristics of the model for the entire database (without clustering)

reflect different aspects of the models (Dolado, 2001). Mean magnitude of relative error (MMRE) is defined as (Conte et al. , 1996):

$$MMRE = \frac{1}{n}\sum_{i=1}^{n}|\frac{e_i - \widehat{e}_i}{e_i}| \qquad (4)$$

where e_i is the actual value of the variable and $\widehat{e}_i$ its corresponding estimate, and n is the number of observations. Thus if MMRE is small, then the predictions can be considered as good.

Prediction at Level p where p is a percentage, is defined as the quotient of number of cases in which the estimates are within the p absolute limit of the actual values, divided by the total number of cases. For example, PRED(0.2)= 70 means that 70% of the cases have estimates within the 20% range of their actual values.

Additionally, a small number of outliers have been removed after checking of the distance from the mean of the clusters, as it is also common practice.

For comparison purposes, an overall model (1) was obtained from the entire ISBSG–8 database. The measures of adjustment for this model with and without cross–validation are showed in Table 6.

As it can be appreciated in the numbers in Table 6, the predictive properties of a single-relationship model justifies the search for alternative parametric approaches. Discussions on heterodestacity (Stensrud et al., 2002) point out that clustering algorithms that deal with measures related to variance could be candidates to break down the problem according to data characteristics.

3.3 Results and Discussion

Figure 1 depicts in loglinear scale the clusters obtained with the "blind" procedure, along with the overall non–cross-validated curve which parameters are provided in Table 6. Table 2 provides partial and average measures for each of the eight clusters. Globally, it can be appreciated that it provides much better adjustment than overall models. However, it should be noted that the clustering process could be applied recursively in several steps to improve adjustment, as described in (Garre et al., 2004), but this is not relevant for our present comparative study.

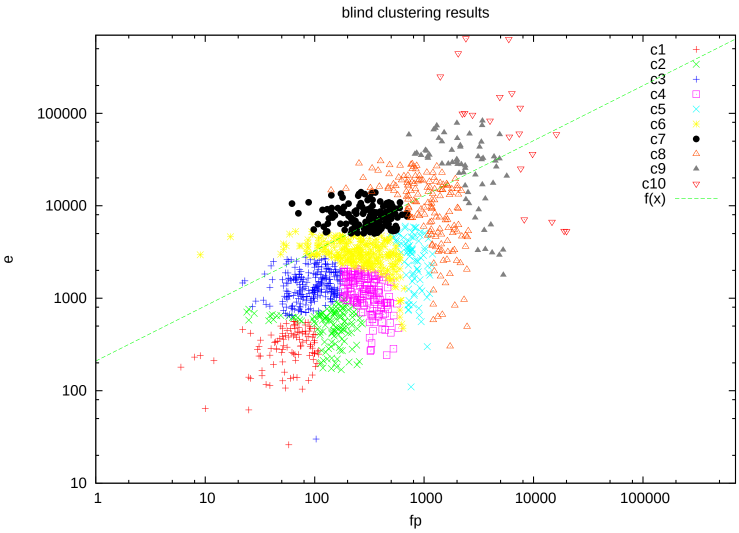

Fig. 1. Clusters obtained through the “blind” procedure

Figure 3–4 depicts in loglinear scale the clusters obtained with the “tailored” procedure, and Table 3 provides the adjustment measures for the “tailored” procedure, organized according to (METHO, CASET) pairs. Comparing the results by cluster in both cases, it can be appreciated that the tailored technique results in clusters with a size below or equal 10. These are precisely the clusters that have the worst adjustment properties. In the case of the clusters for the set (nm, c), the two clusters can be merged, resulting in curve parameters $a = 54.49$ and $b = 0.6765$, with MMRE[t/v]=0.66/0.53 and PRED(.3)[t/v]=0.27/0.66. In the case of the (m,c) number 3 cluster, the cluster can be simply discarded, since the other two clusters in the set still provide two divergent characterizations. The resulting overall measure after these two changes is showed in Table 3.3 with the label Overall$^{(*)}$. In addition, it should be noted that the average predictive quality for the (nm, nc) set is significantly better.

An important qualitative issue that should also be considered is that the category (nm, c) represents projects in which an upper-CASE tool was used, but no methodology was followed (including methods developed “in–house”). This seems an uncommon configuration for projects, and in fact, the number of projects registered is small, and their high dispersion results in poor adjustment. This has lead us to discard this category from the analysis. The (m, c) category has also a small number of points which also result in small clusters and worse adjustment measures. Nonetheless, the parametric models obtained are fairly different to those of the other clusters.

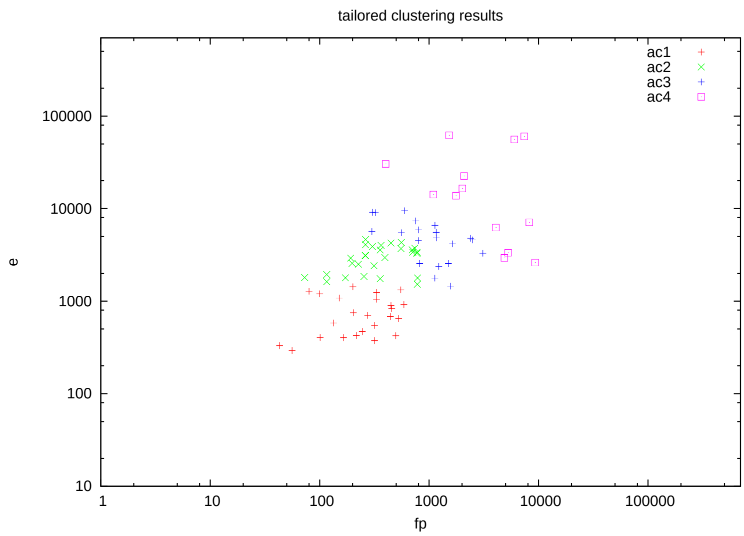

Fig. 2. Clusters obtained through the “tailored” procedure with (nm, nc)

Cluster #	# points	MMRE [t/v]	PRED(.3) [t/v]	a	b
1	152	.5/.51	.49/0.55	88.37	0.2883
2	183	.38/.45	.6/.67	1058	-0.1557
3	343	.3/.33	.6/.52	601.5	0.1825
4	226	.36/.48	.5/.56	7.552e4	-0.7427
5	115	.87/.89	.43/.32	3.451e6	-1.09
6	365	.26/.67	.75/.73	12440	-0.2624
7	209	.23/.21	.67/.71	9213	-0.03311
8	228	.97/.96	.35/.36	4.443e6	-0.8956
9	96	.66/.59	.26/.37	2.528e8	-1.218
10	29	.76/.68	.1/.28	3.210e10	-1.539
Overall		.53/.58	.48/.51		

Table 2
Blind clustering results and adjustment coefficients

The average measures of adjustment for the two studies can be considered as of a comparable magnitude. In consequence, a first quantitative conclusion is that the introduction of *a priori* knowledge result in similar overall predictive characteristics, in the case of METHO and CASET (even though the size of the database used in the second study is significantly). Nonetheless, an analysis of the form of the clusters obtained leads to a different view on the data.

The first important finding of the comparison is that the clusters from dif-

METHO, CASET	Cluster #	# points	MMRE [t/v]	PRED(.3) [t/v]	a	b
nm, nc	1	33	.39/.3	.375/0.55	229.7	0.2009
nm, nc	2	39	.26/.3	.71/.45	873.9	0.2014
nm, nc	3	28	.42/.31	.45/.625	87640	-0.4326
nm, nc	4	19	.57/.7	.23/.33	3.423e6	-0.7347
nm, c	1	10	.31/.4	0/.33	142.7	0.3985
nm, c	2	9	.42/1.2	.5/0	100.7	0.7064
m, nc	1	61	.76/.46	.35/.3	178.4	0.1841
m, nc	2	47	.66/.74	.41/.46	3497	0.09716
m, nc	3	116	.26/.23	.6/.49	699.9	0.202
m, nc	4	78	.27/.19	.62/.6	1356	0.2508
m, nc	5	69	.46/.66	.43/.44	35120	-0.2503
m, nc	6	29	.37/.87	.4/.33	4.565e5	-0.4006
m, c	1	18	.51/.66	.33/.16	0.147	1.527
m, c	2	20	.54/.64	.47/.4	2297	0.05899
m, c	3	9	.26/1.71	.66/0	1.572	1.242
Overall			.43/.56	.44/.36		
Overall$^{(*)}$			.49/.51	.44/.43		

Table 3
Tailored clustering results and adjustment coefficients

ferent *a priori* partitions have a considerable amount of overlapping, which is consistent with the hypothesis that there is not a correlation between size and use of CASE tools or methodologies, i.e. they are revealing underlying aspects of the process that are significant to the creation of realistic parametric models. This consideration raises the need for a systematic study of the potential impact of other variables, since the cost realism of the models could be improved through such studies.

In second place, if the "blind" and "tailored" procedure would have resulted in clusters with large degrees of overlapping, it may be argued that the differences are not so relevant. Figure 5 provides an illustration that evidences that this is not the case. In Figure 5, the clusters c1-c5 are the five first "direct" clusters, and they are put into contrast with the three first clusters of the (m, nc) "tailored" category. The rectangles are computed by obtaining the average of each cluster, and adding and subtracting the standard deviation to that "center value", thus roughly characterizing the area of the points that would be considered as part of the cluster. It can be appreciated that clusters c1 and c2 are to a great extent included in cluster 1 of (m, nc). This may be interpreted as a similarity between the two procedures, since the mnc1 cluster could be subject to an additional clustering, resulting in two partitions roughly equivalent to c1 and c2. Nonetheless, the same pattern does not occur for the rest of the clusters. It is particularly relevant the case of cluster c5, which overlaps to some extent mnc2 but also mnc3 and others, and it does so with a different configuration with respect to deviations on effort. An analysis of the degree of overlapping of clusters using the respective $\gamma(c_i)$ procedures leads to the conclusion that not in all the cases the segments can be considered as similar or as decompositions of other.

Another interesting analysis in the same direction can be appreciated in Figure 3. The curve fitted to cluster (m, nc) number 4 is compared with the overlapping "direct" cluster number 6 (both of them with reasonably good adjustment measures as can be appreciated in the Tables above). The differences between the two models are inverse in their growth, which indicates divergent considerations on economies of scale (Dolado, 2001). This aspect provides an stronger evidence that the use of *a priori* parameters bring up aspects that are not considered in the blind approach.

In summary, the actual models that would be used in each of the approaches differ significantly. This points out that the use of variables actually has a significant influence in the resulting models.

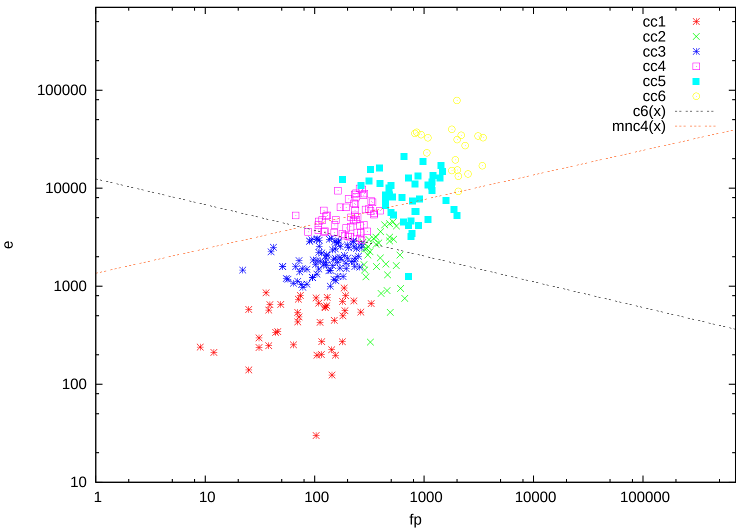

Fig. 3. Clusters obtained through the "tailored" procedure with (m, nc)

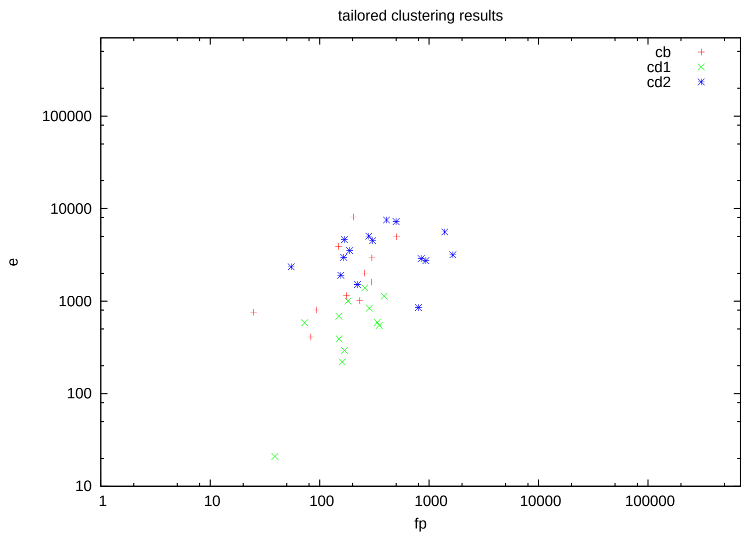

Fig. 4. Clusters obtained through the “tailored” procedure with (m, c)

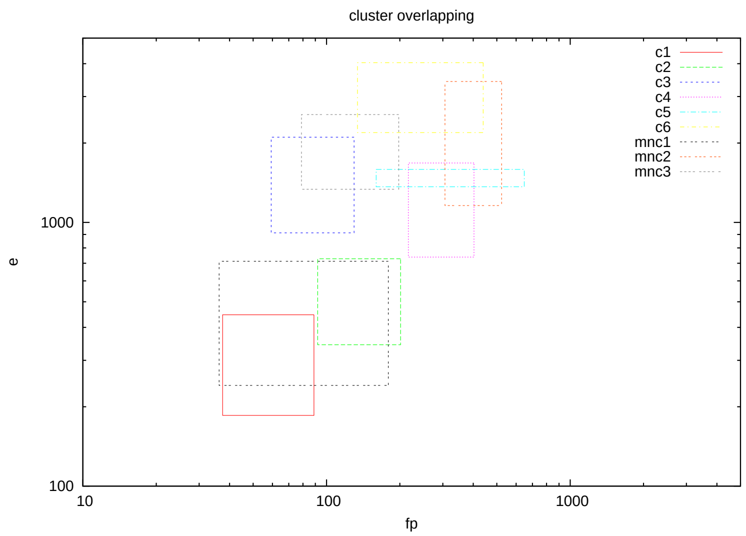

Fig. 5. Overlapping of “direct” clusters 1-6 and “tailored” clusters (m, nc) 1-3

4 Conclusions and Future Work

The use of segmented models in parametric software cost estimation provides an alternative to single–relation models for input domains that are large and heterogenous. Segments can be obtained through clustering procedures that consider project distributions and divergences in variance. Even though size is considered to be the main driver of effort in software development, other factors have also a significant influence. The comparative study described in this paper has provided evidence about the influence of considering two well–known process–related attributes as determinants of the clustering process. Even though the measures of adjustment in the “direct” and “tailored” approaches do not differ significantly, the properties and form of the clusters and models obtain are not assimilable to the same underlying characteristics. This suggests that segmented models should first proceed by assessing potential drivers of the clustering process to obtain more realistic estimation frameworks.

From a pragmatic perspective, the approach described here provides two advantages over existing models. On the one hand, it provides a way to produce parametric estimation models with improved predictive quality without neglecting the consideration of specific relevant process attributes. And on the other hand, the empirical analysis procedure followed can be used to study and gain understanding on the influence of some project attributes for the history of projects available.

There are two main directions for continuing the research presented here. On the one hand, a comprehensive and detailed evaluation of the influence of well–known cost drivers should be carried out to gather additional evidence on the divergences between the two approaches. And on the other hand, other clustering schemes or algorithms combining several of them should be experimented with. The systematic study of cost drivers would provide insights on the actual influence of them, serving as a means for assessing the appropriateness of including them in parametric models. Furthermore, techniques to compare the results of the clusters (e.g. their degree of overlapping) should be applied in an attempt to gain insight on the actual similarities of clusters obtained with and without consideration of pre-selected attributes. With regards to continuing the work on the application of clustering techniques, in addition to using other algorithms, a more thorough cross-validation procedure should be used whenever enough volume of project data is available.

Further work is ongoing in experimenting other clustering algorithms for the same problem. Concretely, algorithms that attempt to extract relationships characterizing clusters like the M5' could provide additional insight in obtaining realistic and interpretable shapes for project clusters.

5 Acknowledgements

This work has been supported by the project MCYT TIN2004-06689-C03 (IN2GESOFT).

References

Baik, J., Boehm, B., Steece, B., 2002. Disaggregating and calibrating the CASE Tool variable in COCOMO II. IEEE Transactions on Software Engineering 28 (11), 1009–1022.

Boehm, B., Abts, C., Chulani. S., 2000. Software Development Cost Estimation approaches – a survey. USC-CSE-2000-505. Center for Software Engineering. University of Southern California.

Boehm, B., 1981. Software Engineering Economics. Prentice Hall.

Boehm, B., Clark, B., Horowitz, E., Madachy, R., Shelby, R., Westland, C., 1995. Cost Models for Future Software Life Cycle Processes: COCOMO 2.0. Annals of Software Engineering 1, 57-94

Boehm, B., Abts, C., Winsor, A., Chulani, S., Clark, B., Horowitz, E., Madachy, R., Reifer, D., Steece, B., 2000. Software Cost Estimation with Cocomo II. Prentice Hall.

Conte, S., Dunsmore, H., Shen, H., 1986. Software Engineering Metrics and Models. Benjamin/Cummings.

Dempster, A., Laird, N., Rubin, D., 1977. Maximum-likelihood from incomplete data via the em algorithm. Journal Royal Statistics Society. Ser. B 39, 1-38.

Dolado, J.J., 2001. On the problem of the software cost function. Information and Software Technology. 43 (1), 61–72.

Dreger, J., 1989. Function Point Analysis. Prentice Hall.

Garre, M., Cuadrado-Gallego, J., Sicilia, M.A., 2004. Recursive segmentation of software projects or the estimation of development effort. In: Proceedings of the Workshop on Decision Support in Software Engineering, Malaga, Spain, November. CEUR Workshop proceedings Vol. 120. Available at http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-120.

Lung, C., Zaman, M., Nandi, A., 2004. Applications of clustering techniques to software partitioning, recovery and restructuring. Journal of Systems and Software 73 (2), 227-244.

NESMA, 1996. NESMA FPA Counting Practices Manual 2.0. Nesma Association.

Oligny, S., Bourque, P., Abran, A., Fournier, B., 2000. Exploring the Relation Between Effort and Duration in Software Engineering Projects. In: Proceedings of the World Computer Congress, Beijing, China, August, pp. 175-178.

Paulk, M., Curtis, B., Chrissis, M., Weber, C., 1993. Capability Maturity Model for Software, Version 1.1. CMU-SEI-93-TR-24. Software Engineering Institute.

PEI, 1999. Parametric estimating handbook, 2nd edition. Parametric Estimating Initiative.

Stensrud, E., Foss, T., Kitchenham, B. and Myrtveit, I., 2002. An Empirical Validation of the Relationship Between the Magnitude of Relative Error and Project Size. In: Proceedings of the Eighth IEEE Symposium on Software

Metrics, Ottawa, Canada, June, pp. 3-12.
Pedrycz, W., Succi, G., 2005.Genetic granular classifiers in modeling software quality. Journal of Systems and Software 76 (3), 277285.
Dick, S., Meeks, A., Last, M., Bunke, H.,Kandel, A., 2004. Data mining in software metrics databases Fuzzy Sets and Systems. 145 (1) 81-110.
Xu,Z., Khoshgoftaar, T., 2004. Identification of fuzzy models of software cost estimation. Fuzzy Sets and Systems. 145 (1), 141-163.