Assessing the Reliability of a Human Estimator

Gary D. Boetticher, Nazim Lokhandwala
University of Houston – Clear Lake
Boetticher@uhcl.edu, Nazim.Lokhandwala@sprint.com

Abstract

Human-based estimation remains the predominant methodology of choice [1]. Understanding the human estimator is critical for improving the effort estimation process. Every human estimator draws upon their background in terms of domain knowledge, technical knowledge, experience, and education in formulating an estimate. This research uses estimator demographic information to construct over 4000 classifiers which distinguish between the best and worst types of estimators. Various attribute techniques are applied to determine most significant demographics. Best case models produce accuracy rates ranging from 74 to 80 percent. Some of the best case models are presented for gaining insight into how demographics impact effort estimation.

1. Introduction

Rather than build more predictor models based on effort estimates, or refine current algorithmic models, why don't empirical software engineer researchers focus on the humans making the estimates?

Of all the effort estimation techniques available, human-based estimation remains the most popular due to its simplicity and flexibility in estimating input and time spent on producing estimates. Various studies compiled by Jorgenson [1], show that human-based estimation is the preferred technique over algorithmic or machine-learning approaches about 77.4% of the time. Furthermore, there is no substantial evidence supporting any claim that any algorithmic or machine-learning method guarantees better estimates than humans make [2].

Algorithmic-based estimation approaches are based on human subjectivity. The post-architecture intermediate COCOMO model has 23 parameters which includes 5 scale factors and 17 effort multipliers that require the modeler to discriminate between classes and to weigh or consolidate different sub-terms within one parameter. For example, the Process Maturity scale factor is a consolidation of 18 key process areas. Finally, the COCOMO model relies upon an accurate size estimate. This size metric may be based on a source lines of code (SLOC) estimate, or a Function Point estimate. Deriving Function Points requires the user to assign values for the 14 Global System Characteristics.

Defining all the factors and coming up with the estimate using the model does not conclude the estimating effort. The estimator must calibrate the results from estimating models to current projects and organization environments in order to achieve potentially accurate results [3]. Even Function Point Analysis is highly subjective in that judgments are made on the General System Characteristics terms. Thus, algorithmic techniques depend heavily upon human, preferably expert, intervention. It seems that human-based estimation is unavoidable.

Machine Learning (ML) based estimation requires many human decisions in terms of which metrics to collect, number of samples to collect, which learner to apply, and how to interpret the results. The black box nature of some ML, such as neural networks, introduces an additional learning curve that might discourage estimators from using it, until they have successfully tried it several times in order to build their confidence in it.

A challenge that exists in human estimation revolves around nature of human life. Assume a person starts working in the software field at the age of 24 and retires at the age of 65. The typical length of a software project is two years. On average, a person would encounter approximately 21 projects during their career. This implies that there are relatively few benchmark points on which to base current estimates. Furthermore, as software development increases in complexity and spans over more complex and dynamic domains, it becomes harder to apply historical domain knowledge in the current domains with newer technologies.

There has been a continuous effort to enhance algorithmic models by calibrating them in order to measure the impact of various inputs on the accuracy of outputs received from the algorithmic models [4]. At the PROMISE 2006 workshop [5], the authors developed a series of machine learners using Genetic Programs along with statistical models which associate human demographics with their estimation ability. This paper extends that research in several ways:

- **Larger sample set.** The 2007 PROMISE dataset contains 56 more samples than the earlier dataset for a total of 178 samples. This offers the luxury of eliminating outliers and noisy data.
- **Many learners.** This paper conducts 4142 experimental trials using 51 different classifiers.
- **Attribute analysis.** Various attribute techniques are applied in multiple experimental contexts to determine which demographic attributes contribute to effective estimation.
- **Simpler models.** This research focuses primarily on classifiers. The intent is to produce more human readable models than the previous research.

This current research examines the influence of different human demographics on the estimation process to determine the impact of demographics in the estimation process. To assess human estimators, a survey is developed which gathers user demographic and requests the respondent to estimate the time needed to complete 28 separate components. 178 samples are collected.

Two types of experiments are conducted. The first compares the worst under-estimators with the best estimators using 51 different classifiers. The second experiment repeats the process, but compares the worst over-estimators with the best estimators. The best models from each experiment serve as evaluators in reducing the number of original attributes. The best reduced attribute models produce accuracy rates ranging from 74 to 80 percent accuracy.

There are significant reasons for addressing this topic. The knowledge of the programmers' demographics can be used to identify the best and worst estimators.

2. Related Research

There have been a relatively few studies on expert estimation. Gray [6] examines a set of expert-derived estimates for the effort required to develop a collection of modules from a large health-care system. Statistical tests suggest a clear relationship between the screen or report type and characteristics of modules and the likelihood of the associated development effort being underestimated, approximately correct, or over-estimated.

Connolly [7] compares Decomposed versus Holistic Estimates of Effort Required for Software Writing Tasks. He reports that the actual effort used to solve programming tasks falls inside the 98% confidence effort prediction intervals for only 60% of the tasks. Explicit attention to training in establishing good minimum and maximum effort values increases the proportion inside the prediction interval to about 70%. He suggests that expert estimates get more accurate when including risk analysis in the estimation process.

Jorgensen [8] randomly selected 109 maintenance tasks and assigns them to people with varying experiences after providing details regarding task specifications. The study reports no clear correlation between length of experience and prediction accuracy of own work among software maintainers.

Studies from other domains indicate several interesting characteristics of expert judgment that can probably be relevant to software effort estimation. Hoch [9] in his study on decision support systems suggests that experts performed better than models in a highly predictable environment, but worse in a less predictive environment. MacGregor's [10] study on aids for quantitative estimation suggests that decomposition of a task for estimation purposes could activate too much information processing and lead the expert estimator astray. Braun [11] compares expert judgment with model forecasts suggests that experts outperformed models in shorter-term business forecasting, whereas models outperformed experts in long term forecasting. The application and relation of these characteristics in software effort estimation have not been found.

3. Survey-Based Empirical Data

A Web-based survey serves as the data collection mechanism. This survey consists of four sections. The first section describes the software project that the respondents are assessing. The second section gathers demographic information about the respondent. In the third section, the respondent assesses the amount of effort, in hours, for a set of 28 modules. The fourth section provides statistical feedback to the respondent. This survey in its entirety is available at: http://nas.cl.uh.edu/boetticher/EffortEstimationSurvey.html

The survey was marketed to several Yahoo Groups related to software engineering; to a mailing list of about 800 members of the Project Management Institute; and to graduate students enrolled in a software metrics course over several semesters. About one third of the survey was completed by graduate students.

3.1. Project Description

In order to provide a background foundation for the survey, the first section describes the project's software requirements in a narrative format. This description also talks about the environment in which the actual project was created.

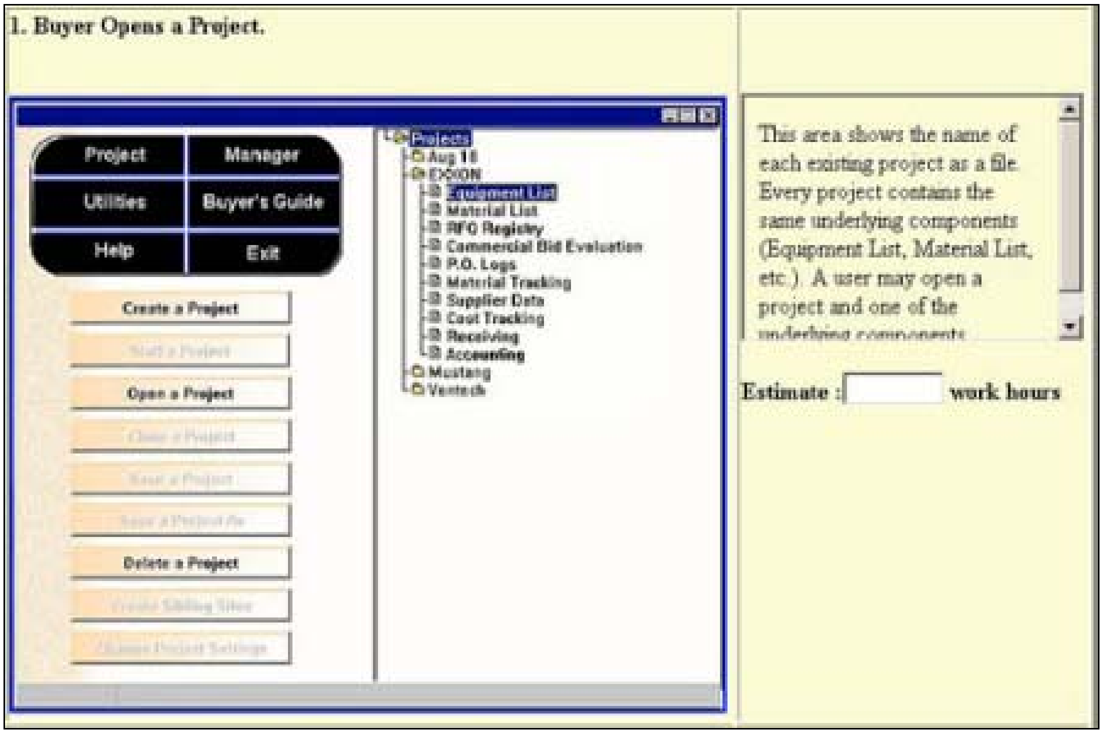
3.2. Demographic Information

Participant demographics include: *Year Of Birth*, *Gender*, *Nationality*, *Highest Academic Degree Achieved*, *The Number of Courses taken at the Undergraduate and Graduate level* such as Computer Science, Computer Information Systems, Computer Hardware, or Management Information Systems. The *Number of Workshops and Conference* in the areas of Computer Science, Computer Information Systems, Computer Hardware, Management Information Systems, Project Management, Project Metrics, or Software Engineering. The *Number of Years of Industrial Experience* for a specific programming language. The respondent is asked to consider 18 different programming languages. The *Years of Work Experience* in Hardware and Software Industry, *Years of Experience as a Project Manager* in Hardware and Software Industry, *Number of Projects* estimated in Hardware and Software Industry, and *Average Size of Software Projects* estimated.

3.3. Component Analysis

In the third phase the respondent must provide effort estimates for a set of 28 different modules. These modules originated in an eCommerce project developed by one of the authors (Boetticher) in the late 1990s. Rigorous effort estimation data was logged per module during the development process. To insure the survey could be completed within a reasonable amount of time, a representative sample of modules from the project are included in the survey.

The survey provides extensive help in the form of an overall description of the whole project along with context sensitive help per module. Figure 1 illustrates one of the modules from the survey.


1. Buyer Opens a Project.

This area shows the name of each existing project as a file. Every project contains the same underlying components (Equipment List, Material List, etc.). A user may open a project and one of the underlying components

Estimate : ______ work hours

Figure 1: Screenshot of one of the modules

This section closes with questions regarding the respondent's domain experience in the procurement and process industry.

3.3. Survey Results

After assessing the 28 modules, the respondent receives feedback regarding their estimates. Figure 2 shows a graph from a survey where the results are sorted by effort. Ideally, a respondent's estimates would overlap the actual values. This graph also provides a Pred(25) count, which represents the number of estimates within 25% of the actual values.

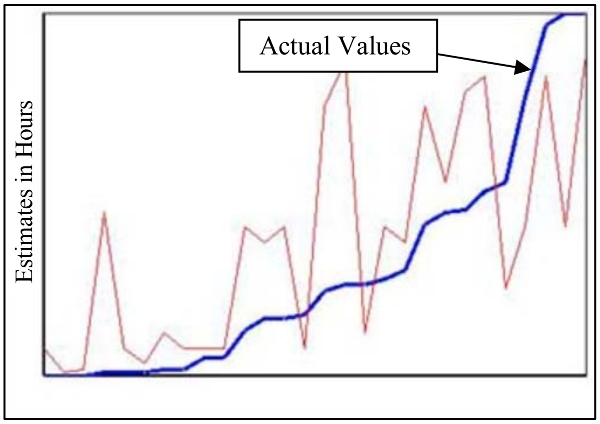


Figure 2. Estimates Sorted by Actual Values

The survey also provides project-based feedback to the respondent in terms of how their accumulative estimates compare to previous participants. Figure 3 shows the Mean Absolute Relative Error (MARE) of all the participants plotted in ascending order. In this example, the participant was in the top 80% profile.

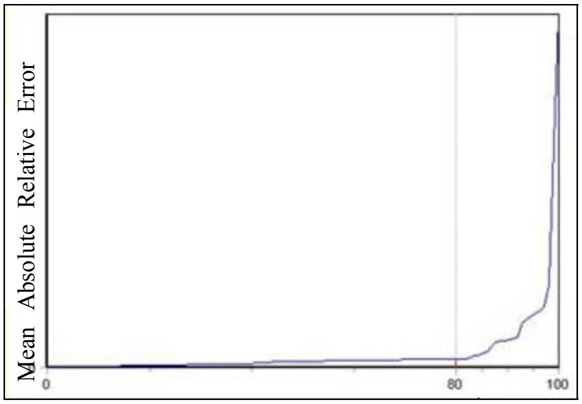


Figure 3. Respondent's Estimates Relative to Other Participants in terms of MARE.

The module and project feedback offer immense value and serves to motivate the user to complete the survey

4. Data Demographics

The data set consists of 178 samples that were collected from 2001 through 2005. The average age is 31.43. There are 148 male and 30 female respondents. Academically, in terms of highest degree held, 1% held a Ph.D., 24% held a Master's 72% held a Bachelor's degree, and 5% held a High School degree. Citizens from 25 different countries completed the survey with 42% from India, 32% from the United States, 6% from Romania, and 4% from Vietnam. Table 1 summarizes each participant's work, estimation, and domain experience.

Table 1: Summary of Experience of Participants

	Ave. Years	**Max.**	**Std. Dev.**
Years of Experience as a			
Hardware Proj. Mgr.	1.0169	25	3.0633
Software Proj. Mgr.	1.6967	15	2.4757
No. of Projects estimated			
Hardware Projects	1.4382	25	4.4390
Soft. Projects	3.6692	28	5.3856
Domain Experience			
Procurement & Billing	1.4382	25	4.4391
Process Industry	3.6629	28	5.3856

5. Experiments

Prior to conducting any experiment, outliers are removed from the data set. In this case, anyone who over-(under-) estimates by a factor of 10x or greater, is removed. Three groups of 25 samples are extracted from the remaining 163 tuples. These groups are the 25 worst under-estimators; the 25 best estimators, and the 25 worst over-estimators. Extracting the best and worst samples seeks to eliminate noisy data in the form of average estimators.

Two sets of experiments are conducted. The first compares the *under-estimators* to the *best* estimators. The second experiment assesses the worst *over-estimators* to the *best* group. The intent is to identify distinguishing characteristics between each of the two groups.

The data is assessed using the Waikato Environment for Knowledge Analysis (WEKA) tool. WEKA is an open-source, Java-based application that contains very many classifier, clustering, association, and data analysis algorithms. It is developed by the University of Waikato in New Zealand and is available at: http://www.cs.waikato.ac.nz/~ml/weka/

Each experiment initially runs 51 sub-experiments with a different classifier each time. There are 4 trials per sub-experiment using a random seed values of 1, 10, 20, and 40 per trial. Those classifiers which produce the best results serve as the evaluator classifier for reducing the number of attributes. After reducing the number of attributes, the next phase conducts another set of sub-experiments against the classifiers using 4 trials per sub-experiment. All sub-experiments use a 10-fold cross validation with the default settings of each classifier.

5.1. Under-Estimators versus Best-Estimators

The average accuracy for the 51 classifiers described above is 48.22 percent. Table 2 lists the classifiers with the best results.

Table 2: Best Classifiers

Classifier	Accuracy
PART	76%
J48	68%
Logistic	64%
ThresholdSelector	64%
VFI	64%

In an effort to improve accuracy and reduce complexity, the Wrapper attribute reduction algorithm is applied with a threshold of 0.30. Classifiers from Table 2 serve as the evaluators in the Wrapper algorithm. All 5 cases use the Exhaustive search.

Table 3: Attribute. reduction by class. (Thresh. = 0.30)

	Evaluator Classifier				
Demographic	J48	Logistic	PART	Thresh.	VFI
Domain Experience	Y			Y	Y
Hardware Project Management Experience	Y	Y	Y	Y	
Mgmt Grad. Courses		Y			
Mgmt Undergrad Courses	Y		Y		Y
# of Hardware Proj. Est.	Y	Y	Y	Y	
Level of College		Y	Y		
Software Proj. Mgmt Exp.		Y			Y
Tech Undergrad Courses		Y			
Total Conferences					Y
Total Workshops		Y		Y	Y
Total Lang Experience		Y		Y	Y

Table 3 shows the results of reducing attributes. A 'Y' signifies that the Wrapper attribute algorithm reduced to the specified demographic feature. For example, the Wrapper algorithm using the J48 evaluator reduces the attribute set from 15 down to 4 attributes: *Domain Experience*, *Hardware Project Management Experience, Mgmt Grad. Courses*, *Mgmt Undergrad Courses*, and *# of Hardware Proj. Estimated.*

The reduced set of attributes is applied to each of the 51 classifiers with 4 trials per classifier. Table 4 shows the best models for each of the reduced sets of attributes.

Table 4: Best models from reduced attrib. (Thresh=0.3)

Classifier	Evaluator	Accuracy
ADTree	PART	78%
ThresholdSelector	ThresholdSelector	76%
Bagging	J48	74%
LogitBoost	J48	74%
J48	J48	74%
PART	J48	74%
VFI	VFI	70%
Logistic	Logistic	68%

A second Wrapper attribute reduction algorithm is applied, but with a threshold of 0.01. The classifiers from table 2 are used as the evaluators in conjunction with the Wrapper algorithm. All 5 cases use the Exhaustive search.

Table 5 shows the results of reducing attributes. A 'Y' signifies that the Wrapper attribute algorithm reduced to the specified demographic feature.

Table 5: Attribute reduction by class. (Thresh. = 0.30)

	Evaluator Classifier				
Demographic	J48	Logistic	PART	Thresh.	VFI
Domain Experience	Y		Y	Y	Y
Hardware Project Management Experience	Y	Y	Y	Y	
Mgmt Grad. Courses	Y	Y			Y
Mgmt Undergrad Courses	Y		Y	Y	
# of Hardware Proj. Est.	Y	Y	Y	Y	
# of Software Proj. Est.					Y
Level of College		Y		Y	Y
Software Proj. Mgmt Exp.		Y			Y
Tech Undergrad Courses		Y	Y		
Total Conferences			Y		Y
Total Workshops		Y			Y
Total Lang Experience		Y			Y

The reduced set of attributes is applied to each of the 51 classifiers with 4 trials per classifier. Table 6 shows the best models for each of the reduced sets of attributes.

Table 6: Best models from reduced attrib (Thres=0.01)

Classifier	Evaluator	Accuracy
ADTree	ThresholdSelector	76%
J48	PART	74%
PART	PART	74%
ADTree	J48	74%
PART	J48	74%
VFI	VFI	70%
Logistic	Logistic	68%

The PART and J48 models are presented. The ADTree is omitted since it is much more complex in structure than the PART and J48 models.

The PART algorithm, 74% accuracy, generates the following four rules for distinguishing between the *best*, Class *A*, and *under*, Class *F*, estimators:

```
1)  Domain Exp <= 3 AND,
    Hardware Proj Mgmt Exp <= 1 AND
    # Of Hardware Proj Estimated <= 4 AND
    MgmtUGCourses <= 0: A (23.0/8.0)
```

If a human estimator satisfies the conditions posed in rule 1, then there is about a 35 percent chance (8/23) of being classified as a best estimator.

```
2)  Domain Exp <= 3 AND,
    Hard. Proj Mgmt Exp <= 1: F (14.0/1.0)
```

The second rule repeats the first two clauses of the first rule. Now there are no restrictions on number of projects estimated and the number of undergraduate management courses. Those respondents that satisfy rule 2, but not rule one, have about 93 (13/14) percent of being a very good estimator.

```
3)  # Of Hard Proj Estimated <= 4: A (8.0)
```

Those respondents that do not satisfy rules one and two, but satisfy the third rule, are classified as under-estimators 100 percent of the time. This rule does not claim that a person has more than one year experience since it is possible to have 0 years of hardware project management experience and more than 3 years of domain experience.

```
4)  F (5.0/1.0)
```

The remaining individuals are considered under-estimators 17 percent of the time (1/6).

The J48 algorithm, 74% accuracy, generates:

```
    Domain Exp <= 3
    | No Of Hardware Proj Estimated <= 4
    | | Hardware Proj Mgmt Exp <= 1
    | | | MgmtUGCourses <= 0: A (23.0/8.0)
    | | | MgmtUGCourses > 0: F (13.0/1.0)
    | | Hard. Proj Mgmt Exp > 1: A (5.0)
    | No Of Hard. Proj Est. > 4: F (5.0)
    Domain Exp > 3: A (4.0)
```

This rule is almost functionally equivalent to the PART rule described above. The confusion matrix for both rules is as follows:

```
  A  F   <-- classified as
 21  4 |  A
  9 16 |  F
```

The confusion matrix means that best estimators are correctly classified 84 percent (21/25) of the time and under-estimators are correctly classified 64 percent (16/25) of the time.

5.2. Over-Estimators versus Best-Estimators

The next set of sub-experiments compares the 25 best estimators with the 25 poorest over-estimators where estimate is less than 10X. The average accuracy for the 51 classifiers described above is 42.86 percent. Table 7 lists the classifiers with the best results.

Table 7: Best Classifiers

Classifier	Accuracy
RandomTree	66%
Decorate	62%
RandomCommittee	60%
ThresholdSelector	60%
Ridor	60%

Apply the Wrapper attribute reduction algorithm as in section 5.1 with a threshold of 0.30 and an *Exhaustive* search reduces the attributes as shown in Table 8. The Decorate algorithm is not included since it did not eliminate any attributes. Also, RandomTree eliminated all the attributes.

Table 8: Attribute reduction by class. (Thresh.= 0.30)

Demographic	Random Committee	Ridor	Threshold Selector
Hardware Project Management Experience	Y	Y	Y
Mgmt Grad. Courses	Y	Y	Y
Mgmt Undergrad Courses			Y
# of Software Proj. Est.			Y
Software Proj. Mgmt Exp.	Y		Y
Tech Undergrad Courses			Y
Total Conferences	Y		Y
Total Workshops		Y	
Total Lang Experience	Y		Y

Applying the reduced set of attributes from Table 8 to each of the 51 classifiers with 4 trials per classifier yields the following best models as depicted in Table 9.

Table 9: Best models from reduced attrib. (Thrsh=0.30)

Classifier	Evaluator	Accuracy
IB1 (Instance Base Learner)	Ridor	80%
Random Committee	RandomCommittee	72%
ThresholdSelector	ThresholdSelector	66%
ADTree	ThresholdSelector	62%

Since several of the learners eliminated all the attributes during the reduction process. The experiment is repeated, with a threshold of 0.01 instead of 0.30. Table 10 shows the results. Once again, the RandomTree classifier eliminated all the attributes.

Table 10: Attribute reduction by class (Thresh.= 0.01)

Demographic	Decorate	Random Comm.	Ridor	Thresh.
Domain Experience				Y
Hardware Project Management Experience	Y	Y		Y
Mgmt Grad. Courses	Y			
Mgmt Undergrad Courses	Y		Y	
# of Hardware Proj. Est.		Y		
Level of College		Y		
Procurement Industry Exp			Y	Y
Software Proj. Mgmt Exp.			Y	
Tech Grad Courses			Y	Y
Tech Undergrad Courses		Y		
Total Workshops			Y	
Total Lang Experience	Y		Y	Y

Applying the reduced set of attributes from Table 10 to each of the 51 classifiers with 4 trials per classifier yields the following best models as depicted in Table 11.

Table 11: Best models from reduced attrib.(Thrsh=0.01)

Classifier	Evaluator	Accuracy
RandomCommittee	RandomCommittee	80%
RandomTree	RandomCommittee	80%
IB1 (Instance Base Learner)	Decorate	74%
IBk	Decorate	74%
RandomForest	Decorate	74%
RandomCommittee	Decorate	72%
NNge	Decorate	72%
PART	Decorate	72%
NNge	ThresholdSelector	66%
ThresholdSelector	Ridor	64%
Ridor	ThresholdSelector	62%
Ridor	Ridor	62%

The next column shows the rule for the random committee which has an accuracy of 80 percent. The confusion matrix for this rule is:

```
  A  F   <-- classified as
 23  2 |  A
  8 17 |  F
```

Thus, best-estimators are correctly classified 92 percent (23/25) of the time and over-estimators are correctly classified 68 percent (17/25) of the time.

```
TechUGCourses < 45.5
| Hardware Proj Mgmt Exp < 6
| | No Of Hardware Proj Estimated < 4.5
| | | No Of Hardware Proj Estimated < 3
| | | | TechUGCourses < 23
| | | | | Hardware Proj Mgmt Exp < 0.75
| | | | | | TechUGCourses < 18
| | | | | | | Hardware Proj Mgmt Exp < 0.13
| | | | | | | | TechUGCourses < 0.5
| | | | | | | | | TechUGCourses < -1 : F (1/0)
| | | | | | | | | TechUGCourses >= -1
| | | | | | | | | | Degree < 3.5 : A (4/0)
| | | | | | | | | | Degree >= 3.5 : A (5/2)
| | | | | | | | TechUGCourses >= 0.5
| | | | | | | | | TechUGCourses < 5.5
| | | | | | | | | | Degree < 3.5 : F (5/0)
| | | | | | | | | | Degree >= 3.5
| | | | | | | | | | | TechUGCrses < 2 : A (1/0)
| | | | | | | | | | | TechUGCrses >= 2 : F (1/0)
| | | | | | | | | TechUGCrses >= 5.5
| | | | | | | | | | Degree < 3.5
| | | | | | | | | | | TechUGCrs < 10.5 : A (3/0)
| | | | | | | | | | | TechUGCrses >= 10.5
| | | | | | | | | | | | TechUGCrs<12.5 : F (3/0)
| | | | | | | | | | | | TechUGCrses >= 12.5
| | | | | | | | | | | | | TechUGCrs<16: A (2/0)
| | | | | | | | | | | | | TechUGCrs>15 : A (2/1)
| | | | | | | | | | Degree >= 3.5 : F (1/0)
| | | | | | | HardProjMgmt Exp >= 0.13 : A (2/0)
| | | | | | TechUGCourses >= 18 : A (2/0)
| | | | | Hard Proj Mgmt Exp >= 0.75 : F (1/0)
| | | | TechUGCourses >= 23 : F (5/0)
| | | No Of Hardware Proj Est >= 3 : F (1/0)
| | No Of Hardware Proj Est >= 4.5 : A (5/0)
| Hardware Proj Mgmt Exp >= 6 : F (4/0)
TechUGCrses >= 45.5 : A (2/0)
```

6. Discussion

It is interesting that hardware, not software; project management experience appears most often in the reduced attribute sets and appears in most of the rules presented in section 5.1. Perhaps the fact that hardware had higher standard deviations for the demographic attributes, as depicted in Table 1, may be the reason in that there is a wider range of values for distinguishing between two groups.

The last rule generated in section 5.2, although accurate, is quite complex. However, it was able to create many pure leaf nodes. Attempts to simplify the tree by reducing the height led to drops in accuracy. So, this strategy was abandoned.

Comparing the rules generated from sections 5.1 and 5.2 hardware project management experience is the only attribute that occurs frequently in both sets of rules. The rules in section 5.1 use *Domain Experience* and *Undergraduate Management Courses* some and the rule in section 5.2 rule makes extensive use of *Technical Undergrad Courses* and *Degree*. Further research may shed more light on this observation.

7. Conclusions

Traditionally, issues like model-based versus expert-based effort estimation have served to divide the software engineering discipline into two camps, the expert-based versus the model-based, which, for the most part, ignore each other. This paper shows why this division is artificial and therefore should be rejected.

Thousands of classification models are constructed which result in watermark accuracy rates ranging from 72 to 80 percent. This paper presents some of more human-readable rules from these accurate models and offers some corresponding observations.

8. Future Directions

As time permits, an additional set of experiments will be conducted that combine both worst cases and compares them to the best case.

One of the challenges in reducing features in order to characterize good estimators is to produce consistent a consistent set of features for various learners. Adding more samples might be a method for reducing the variability in features when reducing the number of attribute.

It is worth noting that all 28 modules in this study come from one project. It is entirely possible that some of the worst under-estimators (over-estimators) might be the best estimators on a different project data. Thus, it would be desirable to conduct further studies with different projects.

9. References

The authors would like to thank all the reviewers for their insightful comments.

10. References

[1] Jorgensen, M., “A review of studies on Expert Estimation of Software Development Effort,” Journal of Systems and Software, 2004.

[2] Jorgensen, M., “Top-down and Bottom-Up Expert Estimation of Software Development Effort,” Journal of Information and Software Technology, 2004.

[3] Jeffery, D. G. and G. Low, “Calibrating estimation tools for software development,” Software Engineering Journal, vol. 5, no 4, Pp. 215..221.

[4] Boehm, B., Clark, B., Horowitz, E., Westland, C., Madachy, R. and R. Selby, “Cost models for future software life cycle processes: COCOMO 2.0,” *Annals of Software Engineering*, Vol. 1, 1995, Pp. 57..94.

[5] Boetticher, G., Lokhandwala, N., James C. Helm, “Understanding the Human Estimator,” Second International Predictive Models in Software Engineering (PROMISE) Workshop co-located at the 22nd IEEE International Conference on Software Maintenance, Philadelphia, PA, September, 2006.

[6] Gray, A. R., MacDonell, S. G. and M. J. Shepperd, “Factors systematically associated with errors in subjective estimates of software development effort: the stability of expert judgement,” Proceedings of the *Sixth International Software Metrics Symposium*, 1999.

[7] Connolly, T., and D. Dean, “Decomposed versus holistic estimates of effort required for software writing tasks,” *Management Science*, vol. 43, 1997, Pp. 1029..1045.

[8] Jorgensen, M., Sjoberg, D. and G. Kirkeboen, “The Prediction Ability of Experienced Software Maintainers,” *4th European Conference on Software Maintenance and Reengineering*, Zurich, 2000.

[9] Hoch, S. J. and D.A. Schkade, “A psychological approach to decision support systems,” *Management Science*, vol. 42, 1996, Pp. 51..64.

[10] MacGregor, D. G., and S. Lichtenstein, “Problem structuring aids for quantitative estimation,” *Journal of behavioral decision making*, vol. 4, 1991, Pp. 101..116.

[11] Braun, P.A., and I. Yaniv, “A case study of expert judgement: Economists’ probabilities versus base rate model forecasts,” *Journal of Behavioral Decision Making*, vol. 5, 1992 Pp. 217..231.