Presented at the 2011 ISPA/SCEA Joint Annual Conference and Training Workshop - www.iceaaonline.com

Cost Analysis using Random Forest Prediction : Estimating the Cost of Simulation-based Experimentation Projects

Karen Mourikas, Denise Nelson, James Schimert

Submitted to ISPA / SCEA User Workshop
June 2011

This document does not contain technical data within the definition contained in the International Traffic in Arms Regulations (ITAR) and the Export Administration Regulations (EAR), as such is releasable by any means to any person whether in the U.S. or abroad. The Export Compliance log number for this document is RBE32511-NT (assigned IAW PRO-4527, PRO 3439).

Abstract

What do you do when your cost data is not well suited for linear regression analysis? One solution is to employ Random Forests, a computer intensive learning algorithm. Random Forests are a collection of Decision Trees. Trees can capture non-linear relationships and interactions among predictors. Individual decision trees are interpretable, however, not necessarily great predictors. By suitably fitting a collection of trees, and averaging the tree predictions, Random Forests results compete with the best machine learning predictors.

Because our dataset has a small number of observations relative to the number of mostly categorical predictors, creating a design matrix for linear regression results in more columns than rows. This forces a strategy of selecting variables, and artificially converting categorical data to numeric values. An alternative is to consider adaptive, nonparametric procedures such as Support Vector Machines and Random Forests. We chose Random Forests, which can handle small datasets with numerous predictors, as well as a mix of categorical and numerical data. We used a model selection strategy, evaluated the goodness of fit using R-squared, and assessed performance on future data sets using prediction error and prediction intervals. This paper introduces the basic concepts of Random Forest prediction and discusses how Random Forests are applied to estimate costs of simulation-based experimentation projects.

1.0 Introduction

There are many methods to determine how much an engineering or design project will cost. Typical methods include analogies, rules of thumb, engineering judgment, level of effort, and parametric estimates. Of these, parametric estimates, or those based on cost estimating relationships (CERs), are often a preferred method since the methodology can be quick to apply and resulting estimates can be easier to defend. Parametric equations representing CERs are derived from existing, historical data by employing analytic methods such as linear regression (i.e. y = ax+b, where y is the estimated cost), or cost factors (i.e. y = 0.21x where x is the cost of another product or effort) to estimate project cost.

In our research at The Boeing Company, we addressed the question "How much do simulation-based experimentation projects cost?" Estimating the cost of a simulation-based experiment project differs from estimating the cost of an acquisition program which includes distinct development, production, and operations and support phases and ultimately could result in the procurement of a product or system. Simulation-based experimentation projects are often research and development (R&D) type of endeavors, involving significant systems engineering effort. The end result of simulation-based experimentation is an experimentation environment, experiment data, and findings or insights of analysis which address key customer questions and concerns. At the onset of our research, cost estimates for experimentation projects were determined by several methods mentioned above: analogies, rules of thumb, and level of effort. This resulted in inconsistent methods which lead to estimates that were often difficult to predict and defend.

We investigated parametric methods to estimate the costs of simulation-based experimentation projects, primarily regression techniques; however, we ran into some issues with the dataset that discouraged us from pursuing regression analysis. Looking for an alternative approach, we experimented with Random Forest prediction, which is an example of using a computer intensive learning algorithm instead of assuming a parametric model. This paper introduces the basic concepts of Random Forest Prediction and summarizes our approach and progress in predicting the costs of simulation-based experimentation projects using Random Forest Prediction methods.

Organization of Paper

This paper is organized as follows. In Section 2.0 we will briefly describe what we mean by experimentation in general, as well as simulation-based experimentation. Section 3.0 provides descriptions of the processes used and data collected in our attempts to identify the significant parameters that describe the scope of the projects in terms of tasks, work products, and effort.

Furthermore, we discuss shortcomings of our dataset with respect to linear regression analysis and possible alternative approaches to analyze our data. In Section 4.0, we introduce the concept of Random Forest prediction and provide technical background on adaptive, non-parametric regression. Our approach to implement Random Forest Prediction to estimate simulation-based experimentation costs is described in Section 5.0. This includes data preparation, variable selection, model development and validation, prediction intervals, and accuracy of the results. Comparisons of prediction with actual costs of completed projects are described in Section 6.0. Finally, we conclude with potential adaptations to the Random Forest model approach to predict costs of other types of projects, as well as suggestions of future work.

2.0 Simulation-based Experimentation

In order for the reader to grasp the complexities of simulation-based experimentation, we provide some general background information on experimentation projects. For additional information, refer to Appendix A which includes what constitutes an experiment, types of experiments and typical aspects of an experimentation project.

First off, let's define the term "experiment". From the online dictionary, www.thefreedictionary.com, the first definition of an *experiment* is listed as "A test under controlled conditions that is made to demonstrate a known truth, examine the validity of a hypothesis, or determine the efficacy of something previously untried." For our purposes, the second and third phrases correspond to the types of simulation-based experiments that we address, namely examining the validity of a hypothesis, and evaluating something previous untried. In addition, from the same source, the term *experimentation* is defined as "The act, process, or practice of experimenting." We further generalize the above definition of experimentation to include gathering and examining data, exploring questions with analyses, and providing insights and observations related to these questions.

Simulation-based experimentation consists of designing, developing and executing experiments with the use of computer models and simulations representing real systems, with or without operators-in-the-loop. A Constructive Analysis (CA) experiment, for our purposes, refers to an experiment conducted solely with computer simulations, in other words, without real operators affecting the outcome of the experiment. In these types of experiments, automated systems simulate the actions of real people and systems. Conversely, a Virtual Operator-in-the-loop (OITL) experiment involves human interaction, such as decision making, which affects the outcome of the experiment The term "Virtual" indicates people operating simulated systems, such as operators flying a simulated aircraft, sometimes in an environment representing real world situations (i.e. as in a dome-like cockpit simulator). In addition, experiments can involve real, live assets, such as pilots flying actual aircraft which provides data feeds to the experiment.

For our analysis, we focus on CA and OITL experiments only, and exclude any experiments involving live assets. Throughout this paper, the authors interchange the terms experimentation and simulation-based experimentation and refer to the terms synonymously.

3.0 Experimentation Estimating Processes and Data

In order to consistently estimate the costs of experimentation projects a standard cost estimating methodology and process needs to be determined. In addition to predicting estimates in a consistent manner, the process must produce accurate results, be easy-to-use, and, above all, be value-added for the project leads.

We consulted standard processes for both cost estimating *and* experimentation, from an industry standpoint, as well as internal company standards. Standard practices from industry societies, such as the International Society of Parametric Analysts (ISPA), and the Society of Cost Analysis (SCEA), as well as experimentation guidelines from industry, the Guide for Understanding and Implementing Defense Experimentation (GUIDEx) (TTCP, 2006) and the Code of Best Practice for Experimentation (COBP), (Alberts & Hayes, 2002) provided a basis to which we incorporated Boeing Cost, Affordability and Experimentation processes. In addition, we based the results on internal historical data since this would more accurately reflect the type and scope of our projects.

The objective of the cost analysis was to derive the main cost drivers, or the scope parameters that most heavily influence the cost of the project. It was anticipated (hoped) that the cost drivers be as minimally subjective as possible to prevent user bias. Additional objectives were to capture cost savings due to re-use, by leveraging existing work products, such as models, data, and computing environment, and to quantify and apply savings due to learning, by accounting for existing relationships, team interaction and experience, and programmatic aspects.

Based on the constraints and guidelines described above, we developed the experimentation cost estimating process consisting of five steps: *Identification, Collection, Analysis, Modeling, and Prediction*. The *Identification* step consists of two parts: identifying the types of data to collect to help scope the projects, and identifying past, current, and future experimentation projects along with the point of contact (POC). In the *Collection* step, the POCs, or project leads, are interviewed regarding the scope and cost for their projects. The *Analysis* step consists of statistical analysis: identifying pertinent variables, investigating outlier data points, and determining the relationships between the cost, and the predictors. The analyzed data is then applied during the *Model* step, which includes calibrating the model to the dataset, (i.e. determining proper correction factors, see Section 6.0 for more detail) and validating the results. (i.e. ensure that the results are reasonable.) Finally, in the *Prediction* step, the model is run with new data resulting in an estimate of the cost for the new project. Standard reports of

summary and detailed information are produced. The resulting prediction is then used as an estimate to allocate budget or in the determination of the business case of the experiment.

Data

In order to accurately estimate experimentation, the cost drivers must be identified and collected. We developed a questionnaire to collect qualitative as well as quantitative information from the project lead about each experimentation project. The questionnaire evolved over time as we learned more about the work efforts included in an experimentation project. The questionnaire consisted of three main sections: Identification, Scope, and Re-use and Learning. The identification section contained information such as project name, POC, and period of performance. The scope data consisted of the technical and complexity details as well as descriptive information. Examples of technical/complexity data include number of interacting systems and number of measures of effectiveness or measures of performance. Descriptive fields include type of project (i.e. constructive analysis, or virtual operator-in-the-loop experiment), and follow-on status (i.e. new project or continuation / expansion of existing project.) The third section of the questionnaire captured work / effort accomplished on previous projects, including level of redesign / rework to be performed. This section primarily focused on model development and integration, simulation environment, and data collection and analysis. It was assumed that a typical project would benefit from some nominal amount of reuse, and hence would not be starting from scratch in every area. In addition, learning factors were captured reflecting effectiveness of the team, familiarity with tools, and individual experience.

As the number of interviews increased, it became apparent that 1) not everyone spoke the same language, and 2) that data collected reflected only the cost and effort controlled by the project lead regardless of additional contribution from other groups. If/when relevant, the project leads for these additional contributions were interviewed and the data was marked as a separate project. To address the language consistency issue, clear and precise definitions were formed of the terms used to scope the effort. Look-up tables provided additional guidance as well as examples to help the project lead scope the effort.

After collecting approximately 70 data points, we began statistical analysis of the data. Conducting linear regression on the dataset provided disappointing results. Foremost, our dataset contained many potential predictors (~20) compared to the number of observations (~70.) In order to capture non-linear or interacting relationships, the degrees of freedom required a much larger dataset. Hence, we were limited to simple additive linear regression. Determining which variables to analyze and how many variables to include proved to be a daunting and inconclusive task. Another hurdle was the fact that many of the potential predictors were categorical or ordinal data. One method to address this issue was to convert the qualitative data into quantitative data by artificially assigning numerical ratings or ranges to each qualitative level.

This can skew results and furthermore may not be statistically valid. Due to the size of the dataset and the types of data, we investigated other forms of regression analysis.

4.0 Random Forests

Random Forests are a collection or ensemble of Decision Trees. The idea behind any ensemble method is to combine several models rather than try to pick one *best* model -- especially when many models give comparable performance. Ensemble methods aim to achieve superior accuracy by harnessing the strengths of several models, taking advantage of all attempts to learn from the data. In particular, tree ensembles are excellent predictors. In contrast, a single Decision Tree is valued for its interpretability, but often does not predict as accurately as tree ensembles or other methods.

The general prediction problem is to predict a *response y* using some *predictors x*. The variable *y* may be a class (such as good/bad or Republican/Democrat/Independent/etc), probability of a class (such as 73% good, 27% bad), or numeric valued as in regression problems. Here we apply regression since our response *y* is cost, which is numeric valued. Formally, the goal is to find a function *f* such that $y = f(x)$. The function *f* may be a linear function such as used in linear regression, but *f* may also be more broadly defined, such as a "smooth" function for example. Such adaptive nonparametric statistical procedures remove the need to specify a parametric functional form, and "let the data speak for themselves".

Decision Trees

Decision Trees are an example of an adaptive nonparametric statistical procedure. In building the cost prediction model, the tree-based regression algorithm starts by splitting the data into two groups of similar costs. For example, it tries to put all experiments with "high" costs into one group, and "low" costs into another group. It accomplishes this partition by using a rule based on a predictor. An example of a rule is "Is the number of integrated systems < 10". Those experiments satisfying the rule go into one group; the rest go into another. The algorithm finds this rule by searching over the predictors and possible split values that best partition the data into "high" and "low" cost groups. It repeats this partitioning on each group, in a process called *recursive partitioning*. This procedure stops when the final regions (terminal nodes) contain only a few observations, typically less than five. The average of the y response variables in each terminal node are used for prediction. To predict the cost of a new observation, obtain each tree prediction by "walking it down the tree", i.e. successively apply the rule at each node until the observation arrives at a terminal node. The terminal node has an associated prediction which, as described above, is the average of the training set observations in that node. The terminal nodes

form a partition of the data which is conveniently represented as a binary tree as in **Error! Reference source not found.**.

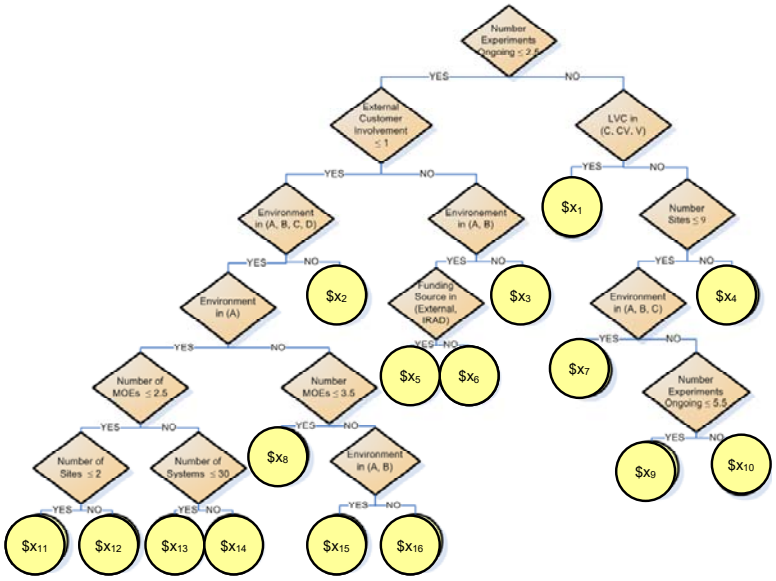


Figure 1: Binary Tree showing rules, split nodes, and terminal nodes

Trees offer a number of advantages including being easy to interpret, even by those with no statistical expertise. In fact, the tree representation may mimic the way that many scientists think about data. Trees can capture not only linear relationships, but also non-linear and interaction behavior among predictors. However, while a single decision tree is generally easier to interpret than a collection of trees, a single tree is not as accurate in predicting as an ensemble of trees. In addition, single trees tend to be *unstable* in the sense that small changes in data can lead to changes in the tree structure.

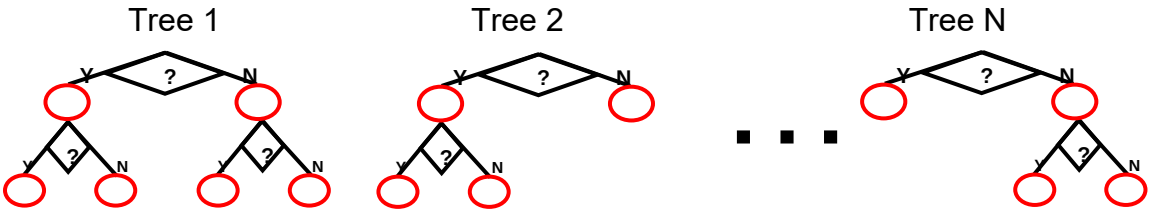


Figure 2: Ensemble of Decision Trees can be more accurate than a single tree

Random Forests exploit this instability by deliberately perturbing the data before building each tree, to get a variety of tree structures in the ensemble. This is done by bootstrapping, i.e. for each tree, use a training set that results from sampling the original training data with replacement. Some data points will be included more than once in a training set, while roughly 1/3 of the data is left out. This data is considered “out of bag” (OOB) for building a particular tree, and, as discussed below, can be used to help evaluate the random forest accuracy.

In addition, when splitting a node, the algorithm searches only over a randomly chosen set of predictors. This allows variables to occur in a tree model that might not otherwise appear when searching over all the predictors. While it may seem counter-intuitive, the extra randomness helps achieve greater accuracy. In ensembles, each tree should have both good accuracy and little correlation with other trees. Low correlation between trees means that the different trees describe different aspects of the data. Random forests work because they decrease correlation between trees (Murua, 2002).

Any regression model approach consists of two phases: (1) training the model and (2) using the model for prediction. The key goal of prediction modeling is to predict future data well. Using the same data to both fit the model and estimate prediction error will lead to over-optimism about future accuracy. Using test samples or techniques like bootstrapping and cross validation are ways to achieve accurate performance estimates.

Another alternative is to use out-of-bag (OOB) estimates (Wolpert and Macready (1996), Breiman (1996)). The idea is that, on average, each bootstrap sample will contain about 63% of the original data. Therefore, for each tree in the ensemble, train on the data on the bootstrap sample, and predict on the data left out of the bootstrap sample ("out-of-bag" or OOB observations). After repeating this for all the trees in the ensemble, obtain an OOB prediction by averaging over the predictions produced whenever an observation is OOB. This simulates predicting new data with an ensemble of trees. Estimate accuracy/error of the ensemble by comparing the OOB predictions with the known response for each observation. Breiman (1996) provides empirical evidence to show that the OOB estimate is as accurate as using a test set of the same size as the training set.

5.0 Modeling with Random Forests

Because our dataset has a small number of observations relative to the number of mostly categorical predictors, creating a design matrix for linear regression would result in more columns than rows. This forces a strategy of selecting variables, and artificially converting categories to numeric values. An alternative is to consider adaptive, nonparametric procedures such as Random Forests, which can handle small datasets with a large numbers of predictors, as well as a mix of categorical and numerical data. This section discusses how we used OOB error to fit a model. In addition, we illustrate variable importance and partial dependence, two concepts that aid in interpretation of the mode, and a model selection strategy, as well as goodness of fit evaluation using R-squared and residual plots. Furthermore, we discuss assessing performance on future data sets using prediction error and prediction intervals.

Fitting the Random Forest

In practice, fitting a Random Forest mainly requires choosing (1) the number of trees and (2) the number of variables to randomly select at each node of a tree. To pick the number of trees in the forest, the strategy is to monitor the OOB errors as trees are added, then stop when the OOB error no longer decreases. Typically, the OOB misclassification rate will fall sharply, and then decrease at a slower rate, eventually leveling off (see

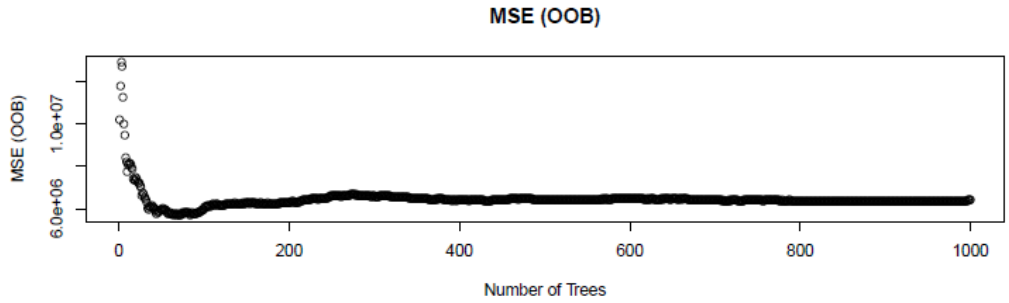


Figure 3: Example Plot of OOB Error versus Number of Trees. The typical pattern is that the error decreases quickly and then levels off.

). In our work we chose to use 1000 trees, although based on the OOB error, we could have used a much smaller number. Similarly, one could choose the optimal number of variables to randomly select at each node, choosing the number that gives the smallest OOB error. In this work, though, we used a recommended number, which is the square root of the number of predictors. Since our data set contained 18 predictors, we used four as the candidate number of variables at each node.

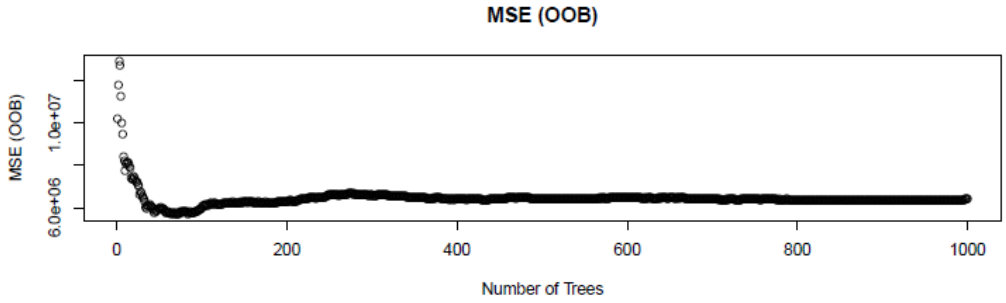


Figure 3: Example Plot of OOB Error versus Number of Trees. The typical pattern is that the error decreases quickly and then levels off.

Interpretation

Although accurate predictions are the main goal in developing a cost estimating model, user acceptance may depend upon understanding the underlying phenomenon. Variable importance measures and partial dependence plots are two techniques that aid in interpreting forests.

One way to measure the importance of a variable is simply to add up the improvements in error achieved when that variable is used to split a node. The more nodes a variable splits, and the more it improves error by splitting, the more important it is (Breiman *et al.*, 1984). However, in node splitting, a variable may be selected for reasons other than information content, and hence introduce bias in measures of variable importance. For example, there is a preference for predictors that have fewer missing values, more distinct numerical values, or more categories in a categorical variable. In this work, we corrected for such bias in variable importance by using an auxiliary set of variables which are permuted in order to give a baseline variable importance (Sandri and Zuccolotto ,2008),

The variable importance plot in **Error! Reference source not found.** identifies the variables that contribute the most to the model. The importance is scaled, with the higher numbers representing more importance. The importance seems to level off after nine predictors, although one could argue to use the top six or even the top four variables.

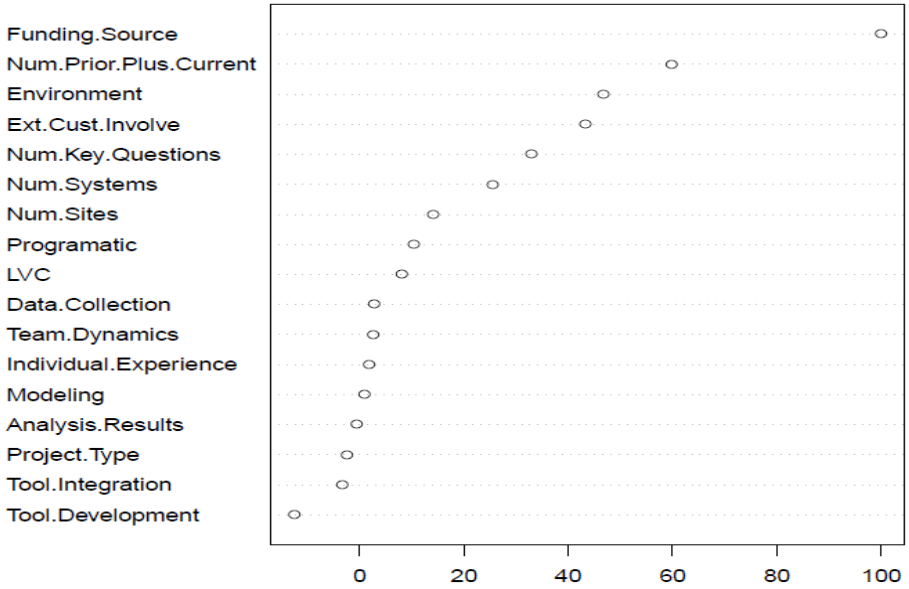


Figure 4: Example Plot of Variable Importance. Higher values indicate greater importance.

The OOB error can also help determine the optimal number of variables. We fit a sequence of models, each with the *k* most important predictors, *k= 1, 2, …18*. **Error! Reference source not found.** indicates that the minimum OOB error occurs when nine variables are included in the model. Hence our final model uses the nine most important predictors.

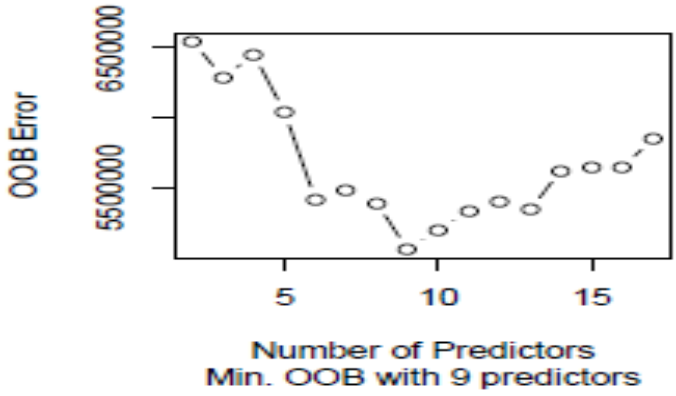


Figure 5: OOB Error vs. Number of Predictors. The Minimum OOB Error is at 9 predictors.

Partial dependence plots can also give insight into the relationships between cost and the predictors. Visualization is a powerful tool for interpretation, but unfortunately many visualization methods fail in high dimensions. Instead, a collection of low dimensional plots can give clues on how cost relates to each predictor. The partial dependence plots in **Error! Reference source not found.** indicate how costs relate to three predictors, when other predictors are held constant.

Error! Reference source not found.b, which is an example of numeric data, shows cost increases as the number of follow-on experiments rises. We expected that as more follow-on experiments are executed, the more complex the experiment becomes, hence the cost increases. However, at the same time, we also assumed that the amount of reuse would increase which would decrease the cost. These plots do not explain the combined effect. Hence we cannot claim that costs increase only on the number of follow-on experiments, but rather that in general costs *tend* to increase when more follow-on experiments are performed but may be affected by other predictors alone or in combination. Also, **Error! Reference source not found.**c, which represents ordinal data, shows little change for the first three levels (A, B, and C), then increases for the fourth level (D) and slightly decreases on the fifth level (E). As the level of effort increases with respect to the environment variable, we would expect the cost to increase. What this plot indicates is that perhaps levels A, B, and C are not significantly different in terms of cost, and therefore could be combined into one level in future work.

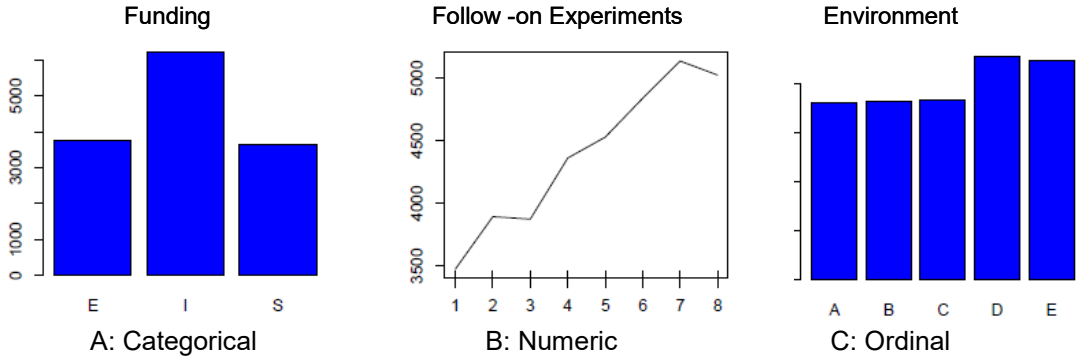


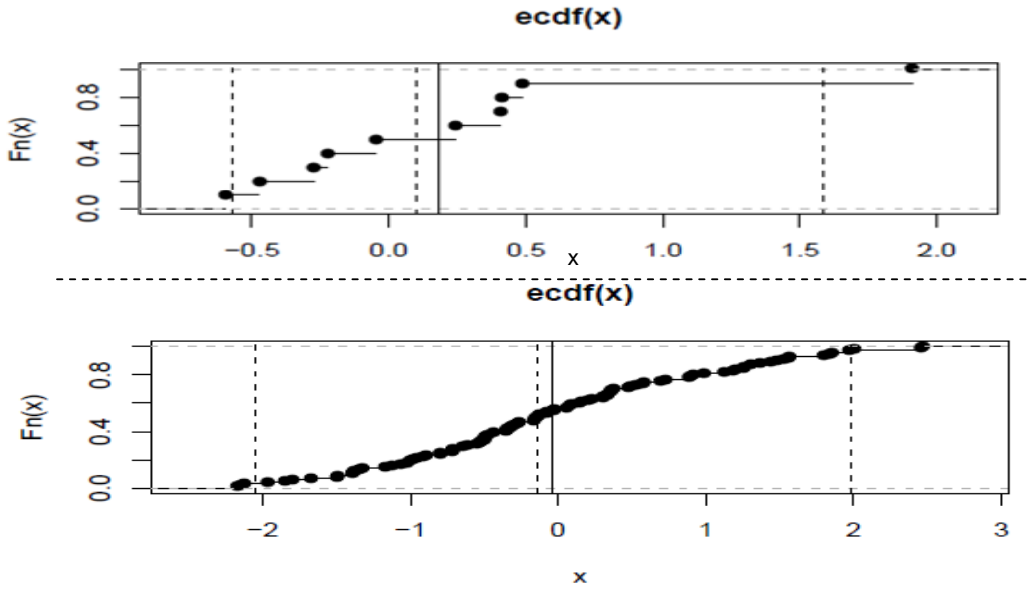
Figure 6: Example Partial Dependence plots of high importance

Prediction intervals

A prediction interval estimates within a certain probability where future observations will fall, based on what has already been observed. For example, one interpretation of a 90% prediction interval is as follows: for a given value of predictors, if you run the model a large number of times, the value will fall within the interval 90% of the time.

When using linear models under the assumption of a Gaussian error distribution, quantiles of a t-distribution are used for prediction intervals. Random Forests do not make such an assumption, and so in this work we applied a nonparametric approach described in Meinshausen (2006).

Recall that the prediction from a tree model is the average of the training set observations in a terminal node. The average is one characteristic of the conditional distribution of the response (cost in this case), given the predictors. Meinshausen (2006) shows that the training set observations in a terminal node may be used to estimate quantiles of the distribution, which in turn can be used to estimate prediction intervals. Terminal node size can be critical. As



shows, an empirical cumulative distribution function is better estimated with increasing amounts of data. Using a larger sample (of 100) provides a smoother chart as shown in the bottom graph. This is relevant to choosing terminal node size for calculating the quantiles nonparametrically.

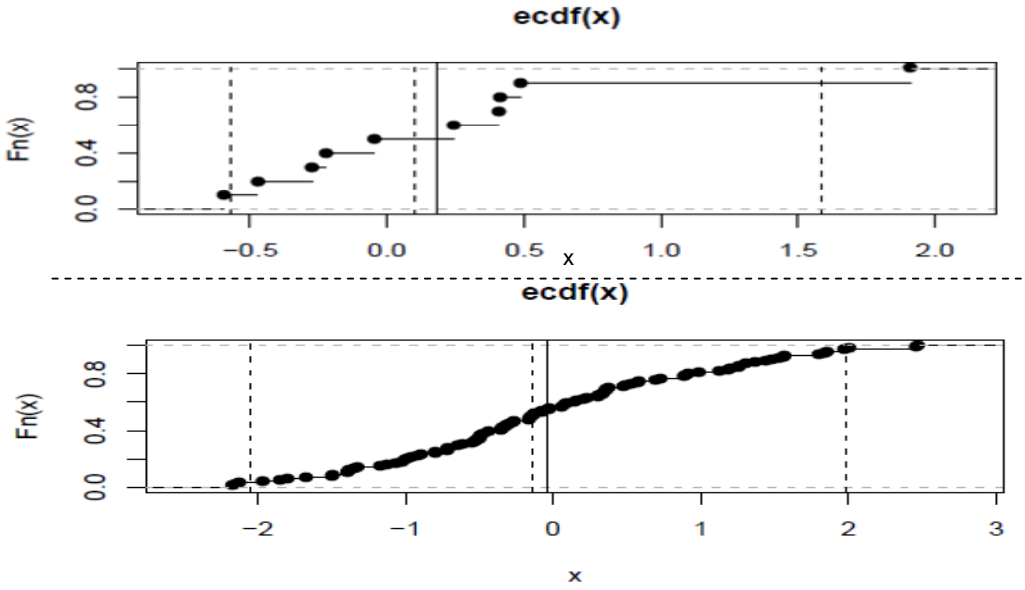


Figure 7: Empirical cumulative distribution plots using 10 observations (top chart) and 100 observations (bottom chart)

Moreover, with estimating experimentation, we found the prediction intervals are shorter after using variable selection versus using all the predictors. In addition, coverage (the percent of intervals that include the observations) is closer to the nominal percent intervals when intervals are calculated using OOB observations. As shown in **Error! Reference source not found.**, the

number of observations that fall outside of the prediction interval is approximately 30%, 20% and 10% for the 70th percentile, the 80th percentile and the 90th percentile prediction intervals.

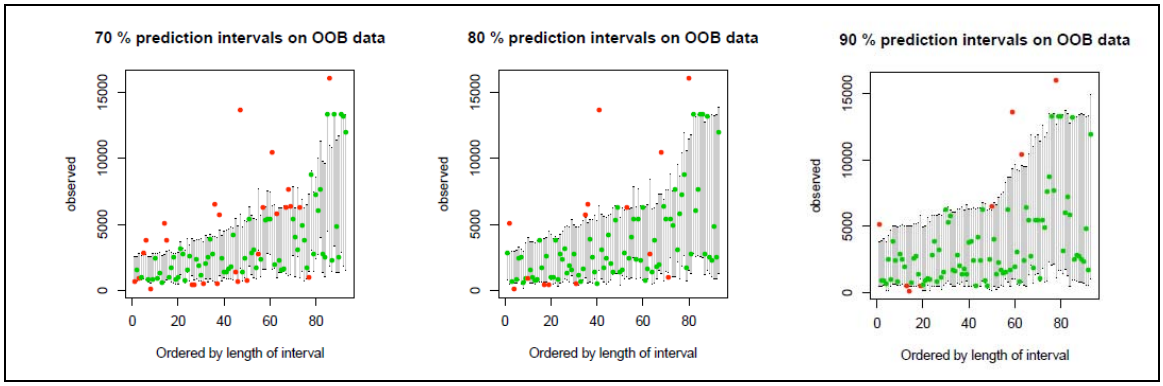
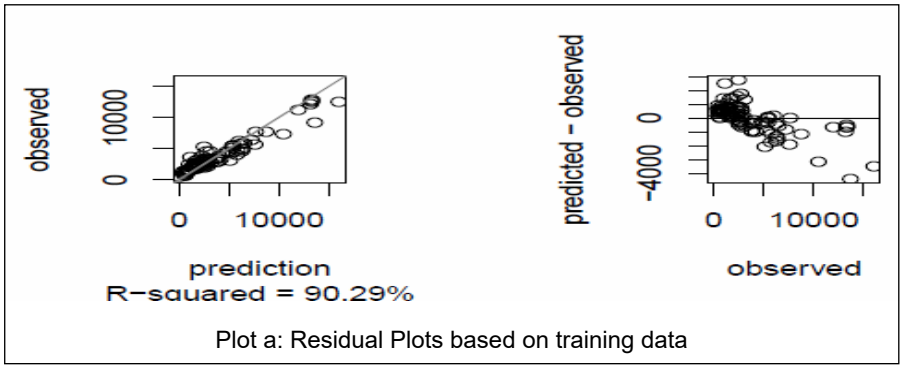


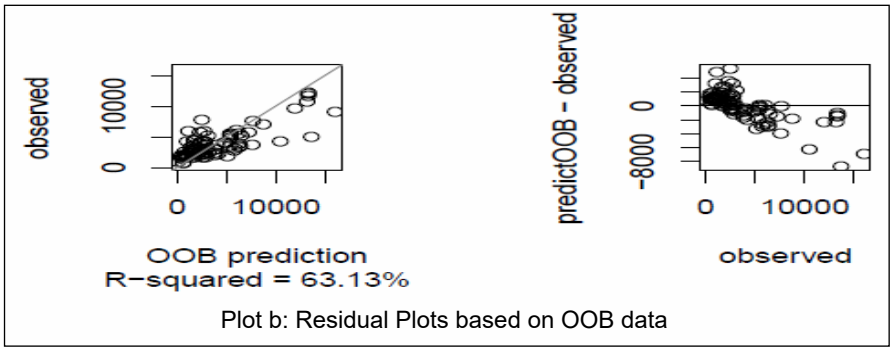
Figure 8: 70, 80, 90% prediction intervals ordered by length of interval, based on OOB data

R^2 Diagnostic Plots

A classic way to assess model fit is to look at diagnostic plots. In an "observed versus predicted" plot, the points from a good fit lie close to a diagonal line. Plotting residuals (predicted minus observed) shows a good fit if the points form a horizontal band. In **Error! Reference source not found.**a, the left chart plots observed cost versus predicted cost. The right chart plots the residuals versus observed values. The plots are based on using the training data and indicate goodness of fit with an R^2 of 90.29%. **Error! Reference source not found.**b shows similar charts, except based on OOB predictions. As noted previously, using the training data both to fit the model and assess future performance (prediction accuracy) results in overly optimistic estimates of accuracy; i.e. the true accuracy on future data is typically lower. In a similar way, assessing fit with OOB data typically results in lower measures of goodness of fit (i.e. lower R^2), but tend to be more accurate. We show both here, because usually the R^2 based on training data is presented. In addition, the charts show that predictions for low cost experiments are more accurate than for high cost. This may be due to having more data points for lower cost experiments than higher cost experiments. Additional investigation is warranted such as analyzing these data points as possible outliers or developing two separate cost models: one for "low" cost projects and one for "high" cost projects.



Plot a: Residual Plots based on training data



Plot b: Residual Plots based on OOB data

Figure 9: Diagnostic Plots used to measure goodness of fit

6.0 Prediction Results

The current Random Forest model was developed based on 70 data points with 18 possible predictors. An additional 34 data points, newly completed projects, were collected to evaluate the model's prediction accuracy. This is an important step in gaining acceptance of the model from the user community. The Random Forest estimate was compared to the actual costs of these projects. On average, the estimates differed from the actual costs by 26% (higher). Since the model tended to overestimate lower-cost projects and underestimate higher-cost projects, as noted in the diagnostic charts, we adjusted for model bias. This adjustment was determined via linear regression on the residual values. While this adjustment may correct for model bias of existing data, there is no guarantee that the adjustment will correct for newly predicted data. With the model bias adjustment, the average difference in estimated versus actual costs was 13%, (lower), an improvement over the original Random Forest model.

Each year, we expect to collect approximately 30 new data points. Hence, each year, we intend to re-visit the Random Forest model, incorporating the new datapoints. In addition, only seven out of 18 potential predictors are included in the current Random Forest model. We plan to investigate how the other potential predictors may influence the cost, and develop post-Random

Forest processing. The effect of these predictors, while not significant in the Random Forest model, may help to fine tune the overall result.

Appendix A: Experimentation Overview

Types of Experimentation

An experiment will take different approaches depending on the type of experiment being performed. Three types of experiments are briefly discussed here: *Discovery, Hypothesis Testing,* and *Technology Demonstrations*.

Discovery experiments are designed to understand the effects of some innovation, such as technical, conceptual, or organizational, and to determine benefits or utility. One famous example of a Discovery experiment comes from World War II. "Perhaps the most famous initial discovery experiments were those conducted by the Germans to explore the tactical use of short range radios before World War II. They mimicked a battlespace (using Volkswagens as tanks) in order to learn about the reliability of the radios and the best way to employ the new communications capabilities and information exchanges among the components of their force." (Alberts & Hayes, 2002).

Hypotheses testing experiments are designed to address specific hypotheses and their expected impacts and take the form of if/then statements with limiting conditions. In one example of hypotheses testing the experiment Objective was to investigate camera-only capabilities for identification and tracking, addressing the question "Does camera tracking software XYZ provide sufficient target recognition and cueing to be used without radars?" The hypothesis was set up to compare target recognition with or without the camera tracking software:

If camera tracking software XYZ used (If A)
then increased Target Recognition (Then B)
without radars (Under conditions C).

Technology Demonstration experiments are designed to showcase, display or validate a technology, concept, or solution under specific conditions often as the result of a discovery or hypothesis experiment.

Phases of Experimentation

In addition to these three types of experiments, some references, such as the "Joint Concept Development and Experimentation Development Process (JOpsC-DP)", (CJCSI 2007), include war games and exercises, as well as seminars, symposiums, and workshops as experiments. Hence, experimentation can range from simple experiments such as a spreadsheet analysis to war game type exercises that last several days and involve hundreds of operators or

warfighters. The scope of our investigation of experimentation projects is focused on Discovery and Hypothesis Testing experiments.

Experimentation typically consists of six functional phases as described in industry: *Discovery* or Customer Interaction, *Problem Formulation, Experiment Design, Experiment Development, Experiment Execution*, and *Analysis*, which are described below. Experimentation starts with initial discovery and customer interaction and continues through to the analysis of the experiment results and observations. The Experimentation process may not necessarily be a linear process; instead it is an iterative process in which the entire process, or phases or tasks within the process, may be repeated as needed. In addition, the phases may overlap and/or be performed in parallel to varying degrees.

During the *Discovery* Phase, working with the customer, the goal is to understand the customers' needs, capability gaps, and issues. In the *Problem Formulation* phase, the customer and experimentation team work to identify the specific problem and to scope the effort, producing a high level plan of action. Typically this phase occurs in conjunction with the *Discovery* Phase. Next, during the *Design* phase, the problem is decomposed into more detail: the experiment objective is refined, requirements for models, simulations, and analysis are identified, and architecture products are developed. During the *Development*, or implementation, phase, the simulation scenario and vignettes are defined; models and simulation capabilities are developed and integrated; and preliminary metrics are collected to answer the key customer questions. Once the experiment environment has been tested, the *Execution* phase begins. This phase consists of the actual conduct of the experiment, running the various cases of the test matrix and collecting run-time data and observations. The *Analysis* phase consists of pre-experiment planning and preparation, as well as analysis of data during and after the experiment execution, during which the experiment data is analyzed to address the key customer questions, and findings, observations and insights are recorded.

References

[1] Alberts, D. and Hayes, R., et al (2002) Code of Best Practice for Experimentation, http://www.dodccrp.org/html4/books_downloads.html
[2] Breiman, L. and Friedman, J. and Olshen, R. and Stone, C. (1984), Classification and Regression Trees, Wadsworth, Monterey, CA.
[3] Breiman, L. (1996), "Bagging predictors", Machine Learning, volume 26, Pages 123—140.
[4] CJCSI 3010.02B Series, "Joint Concept Development and Experimentation Development Process (JOpsC-DP)", Appendix D, Current as of 4 Dec 07
[5] Meinshausen, Nicolai (2006). Quantile Regression Forests, Journal of Machine Learning Research, Vol. 7, pages 983 – 999.
[6] Murua, A (2002), "Upper bounds for error rates associated to linear combination of classifiers", IEEE Transactions on Pattern Analysis and Machine Intelligence.
[7] Sandri M., and Zuccolotto P. (2008), "A Bias Correction Algorithm for the Gini Variable Importance Measure in Classification Trees", Journal of Computational and Graphical Statistics, Volume 17, Number 3, pp611-628.

[8] TTCP Guide for Understanding and Implementing Defense Experimentation GUIDEx (2006), http://www.dtic.mil/ttcp/guidex.htm

[9] Wolpert, D. H. and W. G. Macready (1996), "An efficient method to estimate bagging's generalization error", Santa Fe Institute Technical Report, http://www.santafe.edu/research/publications/wpa