WORKING
PAPER

Comparing Algorithms for Scenario Discovery

ROBERT J. LEMPERT, BENJAMIN P. BRYANT,
STEVEN C. BANKES

WR-557-NSF

March 2008

Prepared for the National Science Foundation

RAND INFRASTRUCTURE, SAFETY, AND ENVIRONMENT

Comparing Algorithms for Scenario Discovery

Robert J. Lempert, Benjamin P. Bryant, Steven C. Bankes

RAND Corporation
1776 Main Street
Santa Monica, CA 90407

February 2008

Abstract

While attractive in many ways, traditional scenarios have lacked an appropriate analytic foundation for inclusion in quantitative decision analyses. In previous work, we have proposed to remedy this situation with a systematic, analytic process we call "scenario discovery" that has already proved useful in a wide variety of applications. This study aims to evaluate alternative algorithms needed to implement this novel scenario discovery task, in which users identify concise descriptions of the combinations of input parameters to a simulation model that are strongly predictive of specified policy-relevant results. This study offers three measures of merit -- coverage, density, and interpretability - and uses them to evaluate the capabilities of PRIM, a bump-hunting algorithm, and CART, a classification algorithm. The algorithms are first applied to datasets containing clusters of known and easily visualized shapes, then to datasets with unknown shapes generated by a simulation model used previously in a decision analytic application. We find both algorithms can perform the required task, but often imperfectly. The study proposes statistical tests to help evaluate the algorithms' scenarios and suggests simple modifications to the algorithms and their implementing software that might improve their ability to support decision analysis with this scenario discovery task.

1. INTRODUCTION

Scenarios provide a commonly used and intuitively appealing means to characterize and communicate uncertainty in a wide variety of practical decision support applications. Unfortunately, no entirely satisfactory approach exists for choosing scenarios (Parson et. al.2006) in situations where computer simulation models are available to provide valuable information to support the decision, but where a high number of potentially important dimensions of uncertainty obscure key driving factors. In many instances, analysts may informally explore the model's implications with a handful of "what if" cases chosen by intuition. Alternatively, some organizations run extensive, formal processes involving many stakeholders to define a small number of key cases that ought to be used in simulation model runs. Neither approach satisfactorily addresses two key methodological challenges – how to choose three or four scenarios that can effectively summarize what is effectively a very wide range of uncertainties and how to best include probabilistic information with such scenarios. In previous work, we have proposed a systematic, analytic method for addressing these challenges by identifying policy-relevant regions in databases of simulation model-generated results (Groves and Lempert, 2007; Lempert et. al. 2006). This study aims to evaluate alternative algorithms needed to implement this task we call "scenario discovery."

The goal of scenario discovery is to find descriptions of the combinations of a small number of input parameters to a simulation model that are most strongly predictive of certain classes of results. In brief, a computer simulation model is run many (hundreds to millions) of times over space defined by combinations of values for the uncertain model input parameters. Some criterion of interest is used to distinguish a subset of the cases as particularly policy relevant, typically by applying a threshold to one of the model's output parameters. Statistical or data-mining algorithms applied to the resulting multi-dimensional database then find the combinations of constraints on a small number of input parameters that best predict these policy-relevant cases. These simply-defined regions of input space can then be usefully considered as scenarios for decision analytic applications. As described in detail in previous work, this approach provides a systematic justification for the choice of these (as opposed to some other) scenarios as well as quantitative measures of merit for their quality. Because the scenarios describe regions in a multi-dimensional space rather than single points, they can be more easily combined with probabilistic judgments.

As an example, a recent study used a computer simulation model to assess the key tradeoffs among alternative roles the U.S. Federal government might play in the provision of insurance against terrorist attacks (Dixon et. al. 2007). The model forecast the costs to taxpayers for each of several thousand combinations of 17 uncertain input parameters representing different types of attacks, the factors that influence the pre-attack distribution of insurance coverage, and any post-attack compensation decisions by the Federal government. An analysis of the resulting model-generated database suggested that the current U.S. government policy would *always* cost the taxpayers the same or less than alternative policies except in a specific scenario where the losses from a terrorist attack exceed some very large monetary threshold, largely independent of the value of the other uncertain parameters. While estimates exist for probability density functions that can be used to calculate the likelihood of terrorist attacks exceeding some monetary loss threshold, such estimates remain highly contentious. Upon identifying this large-attack scenario, the analysis was able to demonstrate that the expected cost to taxpayers for the current U.S. policy was lower than that of the alternatives over a very wide range of plausible assumptions about the likelihood of a large terrorist attack. This surprising result may prove useful to the policy debate over whether to extend the current program.

Identifying an appropriate scenario to support this terrorism insurance analysis proved straightforward because the important cases depended almost entirely on a single uncertain input parameter. But in general discovering such policy-relevant scenarios will prove a more difficult task. Regions containing interesting cases in a multi-dimensional space of model-generated results may have complex shapes and may depend on non-trivial interactions among different model input parameters. The measures of merit for assessing the quality of a given set of scenarios will depend not only on their ability to explain the cases of interest but also on less tangible measures of interpretability and policy-relevance. The most useful algorithms may in general need to work interactively with human users who can help adjudicate the tradeoffs among these multi-attribute measures most appropriate for the policy problem at hand.

To our knowledge no algorithms have been designed specifically to perform the tasks required for scenario discovery. Our past work has used Friedman and Fisher's (1999) Patient Rule Induction Method (PRIM). We have used this bump-hunting algorithm to identify one or more policy-relevant "boxes" -- (hyper)-rectangular regions within the space of uncertain model input parameters which contain a high fraction of policy-relevant points. PRIM has seemed to perform adequately. But to date, there has been no systematic assessment of PRIM's or any other algorithms' strengths and weaknesses for finding such scenarios or, indeed, any formal analysis of what might constitute proper criteria for such an assessment.

Scenario discovery may have broad applicability to a wide range of policy challenges. The approach can enhance the increasingly common practice of "exploratory modeling" (Bankes 1993), also called "data farming"(Brandstein and Horn, 1998), where a computer simulation model is run many times over a carefully chosen set of different combinations of input parameters and the resulting database is analyzed with a combination of interactive visualization, statistical analysis, and computer search in order to discover and test alternative hypothesis relevant to some decision problem. For instance, robust decision making (RDM) is an exploratory modeling based approach to decision making under conditions of deep uncertainty (Lempert and Collins, 2007; Lempert, Popper, Bankes 2003).[i] A key RDM step requires identifying combinations of simulation model input parameters for which a candidate robust strategy performs poorly compared to alternative strategies. As described by Lempert et al. (2006) and in Section 5 below, this step is equivalent to scenario discovery. It should be noted that scenario discovery differs from more traditional sensitivity analysis (Saltelli et. al. 2000) because it seeks to describe regions in databases of model results that have particular properties, rather than rank the importance of different inputs in explaining the variance in the model output around a small number of points. Scenario discovery also differs from prediction-oriented statistical data mining because the former aims to maximize interpretability in addition to accuracy in prediction.

This study assesses algorithms that could potentially support a wide range of scenario discovery applications. It aims to not only compare the performance of two specific algorithms, but also to provide a guide to the use of such algorithms in decision support applications. The study first proposes a formal process for scenario discovery and a set of criteria for assessing the results. It then compares two candidate algorithms – PRIM and CART (Classification and Regression Tree) that seem to possess promising yet different abilities to perform the required task. The algorithms are first applied to datasets containing clusters of known and easily visualized shapes, then to datasets with unknown shapes generated by a simulation model used previously in a decision analytic application. The study finds that both algorithms prove useful for identifying policy-relevant scenarios in the cases considered, but that new and different means of exercising them are necessary to capitalize on their unique strengths and overcome their unique weaknesses. The study concludes with proposed methods to test the statistical significance of the scenarios generated by these algorithms and with suggestions for simple modifications that might enhance their scenario-discovery abilities.

2. MEASURES OF MERIT FOR SCENARIO DISCOVERY

Figure 1 shows the proposed process for scenario discovery. The user creates an experimental design over the space defined by the uncertain input parameters to the simulation model and at each point calculates the value of some policy-relevant model output. A threshold value can then distinguish policy-relevant from less interesting cases, thereby creating a binary output. A scenario-discovery algorithm applied to the resulting database then suggests one or more alternative sets of boxes that characterize these policy-relevant cases and provides the users with measures of the quality of each box set. The user then chooses that box set which appears most useful for the decision analytic application. This paper assesses the ability of alternative algorithms to successfully support this process.

Formally, we imagine a computer simulation model $Y = f(\mathbf{x})$ that calculates some output Y over an D-dimensional space of continuous model input parameters x_j , $j = 1,...,D$ (variables may in practice also be categorical or discrete, but for simplicity in presentation we assume continuity). Each of these varies over a plausible range that, without loss of generality, we can normalize to lie in the range $0 \leq x_j \leq 1$. The decision-maker defines some threshold performance level Y^T that distinguishes the set of policy-relevant cases $\boldsymbol{I} = \{\mathbf{x}^I \mid f(\mathbf{x}^I) \geq Y^T\}$ from other less-interesting cases $\{\mathbf{x}^U \mid f(\mathbf{x}^U) < Y^T\}$. We wish to discover a useful description of ***I*** based upon a finite sampling of $f(\mathbf{x})$, where usefulness depends upon both the accuracy of the description and its interpretability. To this end, consider a set of limiting restrictions on a set of dimensions L that has size $\mathrm{d} \leq \mathrm{D}$. This takes the form $B_i = \bigcap_{k \in L} a_k \leq x_k \leq b_k$ on the ranges of a subset of the input parameters, while dimensions not in L remain unrestricted. We call each individual set of restrictions B_i a box and a set of boxes a box set. We desire an algorithm to suggest a box set that contains mainly policy relevant cases, and also captures most of the policy relevant cases in the dataset.

Choosing among box sets, either algorithmically or through user judgment, requires measures of the quality of any box and box set. Such measures are also useful for comparing the performance of alternative algorithms. The traditional scenario planning literature emphasizes the need to employ a small number of scenarios, each explained by a small number of "key driving forces," lest the audience for the scenarios be confused or overwhelmed by complexity (Schwartz 1996). In addition to this desired simplicity, the quantitative algorithms employed here seek to maximize the explanatory power of the boxes, that is, their ability to accurately differentiate among policy-relevant and less interesting cases in the database.

These characteristics suggest three useful measures of merit for scenario discovery. To serve as a useful aid in decision-making, a scenario-discovery algorithm should suggest a box set that captures a high proportion of the total number of policy-relevant cases (high *coverage*), captures primarily policy-relevant cases (high *density*), and proves easy to understand (high *interpretability*),. We define and justify these criteria as follows:

Coverage gives the ratio of policy-relevant cases contained within a box set to the total number of policy-relevant cases in the database. Thus for a given box set *B* composed of individual boxes $B_i \in B$ and set of policy-relevant points $X^I = \{\mathbf{x}_j^I\}$,

$$Cvr(B;X^I) = \left|\bigcup_i (\mathbf{x}_j^I \mid \mathbf{x}_j^I \in B_i)\right| \Big/ \left|X^I\right| \tag{1}$$

where $|X^I|$ represents the total number of policy-relevant cases in the database. Eq (1) includes the possibility that boxes may overlap. When the boxes in a set are disjoint, $Cvr(B) = \sum_i Cvr(B_i)$. Decision makers should find this measure important because they would like the scenarios to explain as many of the policy-relevant cases as possible.

Density gives the fraction of policy-relevant points within the box set, relative to the total number of points *in the box set*. Thus, this has the same numerator as coverage, but the denominator is a function of the box definition and the complete set of points X = $\{\mathbf{x}_j\}$:

$$Dens(B;X;X^I) = \left|\bigcup_i (\mathbf{x}_j^I \mid \mathbf{x}_j^I \in B_i)\right| \Big/ \left|\bigcup_i (\mathbf{x}_j \mid \mathbf{x}_j \in B_i)\right| \tag{2}$$

Decision makers should find this measure important because they would like each scenario to be a strong predictor of policy-relevant points.

Interpretability suggests the ease with which decision makers can understand a box set and use the boxes to gain insight about their decision analytic application. This measure is thus highly subjective, but we can nonetheless quantitatively approximate it by reporting the number of boxes B_N in a box set and the maximum number of dimensions *n* restricted by any box. Based on the experience reported by the traditional scenario planning literature (Schwartz 1996), a highly interpretable box set should consist of on the order of three or four boxes, each with on the order of two or three restricted dimensions.

An ideal set of scenarios would combine high density, coverage, and interpretability. Unfortunately, these measures are generally in tension, so that increasing one typically comes at the expense of another. For instance, increasing interpretability by restricting fewer dimensions can increase coverage but typically decreases density. Achieving the same goal by reducing the number of boxes will necessarily decrease either density or coverage. For a given dataset, these measures define their own multi-dimensional efficiency frontier. The process in Figure 1 envisions that users interactively employ a scenario-discovery algorithm to generate alternative box sets at different points along this frontier and then choose that set most useful for their decision analytic application.

3. PRIM AND CART ALGORITHMS

To our knowledge, no existing algorithm performs tasks identical to that required for scenario-discovery. The interpretability measure poses requirements distinct from most other applications. In addition, while many algorithms seek to maximize coverage, which is equivalent to the success-oriented quantification of the Type II error rate, few consider density, which is related to but does not neatly correspond to the Type I error rate because the denominator in Eq (1) refers to the box rather than the overall dataset.

Among existing algorithms, the scenario-discovery task appears most similar to classification and bump-hunting approaches. For datasets with binary output, classification algorithms partition the input space into regions of high purity, that is, regions that contain predominantly one output class. Bump-hunting algorithms find regions of input space with a comparatively high mean output value. In this study we assess the ability of one example of each approach – CART (Classification and Regression Tree) and PRIM (Patient Rule Induction Method) – to serve as scenario-discovery algorithms.

Both PRIM and CART optimize measures related, but not identical, to those for scenario discovery. CART minimizes misclassification rates in order to divide the space into regions of high purity. CART's false negative rate equals one minus coverage, but the false positive rate employs a different denominator so is not equivalent to density. PRIM strives for an optimal balance between density and support, typically with a user bias towards density. PRIM's support objective is non-negatively correlated with but not equivalent to coverage, because the former refers to all points in the database and not just those considered policy-relevant.

3.1 Patient Rule Induction Method (PRIM)

Our previous scenario-discovery work used PRIM (Friedman and Fisher, 1999), an algorithm designed for the relatively novel task termed “bump-hunting” by its creators and “activity region finding” by others (Amaratunga and Cabrera, 2004). This study considers PRIM because it is one of few available bump-hunting algorithms and because it provides interactive features that in our application enhance users’ ability to achieve a desired balance among coverage, density, and interpretability.

PRIM employs an iterative, two-step process to find regions within the input space where the mean of the output is significantly higher than the mean of the output over the entire dataset. The algorithm begins with the entire input space and none of the input dimensions restricted. It then creates a series of increasingly smaller and denser (higher mean) boxes. As shown in Fig 2a, PRIM finds each new box by removing a thin low density slice from whichever face of the current box will most increase the mean inside the new (remaining) box. PRIM’s developers call this series of boxes a “peeling trajectory.”

At this stage, PRIM is designed to receive user input. The algorithm presents the user with a visualization of the peeling trajectory, plotting the density of each box against its “support,” the fraction of total points contained within the box. The user is then asked to choose the box that best balances the competing goals of high density and high support for their particular application. As we describe later, the correspondence between support and coverage is less than ideal for scenario-discovery.

These candidate boxes can then be expanded in a process called pasting, in which other dimensions that were perhaps unnecessarily restricted are allowed to expand. While conceptually optimal, PRIM’s authors found that in general pasting plays little role in improving box quality.

Once the user has chosen a box from the pasted candidates, he or she can iterate PRIM using a process called “covering.” As shown in Figure 2b, the algorithm removes all the data points from the dataset inside the first box and replicates the peeling/pasting process with the remaining data. The user may repeat this process of box selection followed by covering until he or she determines the algorithm has exhausted its ability to generate useful boxes.

As one key feature, PRIM peels away only a small fraction of data at each step. This "patience" improves the algorithm's ability to determine the most important input parameters before it runs out of data and helps minimize the adverse consequences of any suboptimal step. In contrast, CART splits the data at every step, therefore limiting itself to an average of $\log_2(N)$ splits, where N is the number of points in the dataset. While CART might fail to restrict important input parameters before it exhausts the data, PRIM's patience may, as shown below, lead the algorithm to inappropriately clip the ends of boxes that would otherwise extend the length of some parameter range.

3.2 Classification and Regression Trees (CART)

This study considers CART because it is one of the few commonly implemented classification algorithms that returns output easily translated into the boxes useful for scenario discovery. CART typically provides output in the form of a decision tree, which provides a hierarchical set of splitting criteria for determining the output class associated with given input combinations (Brieman et al, 1984). However, because the bounds of CART regions are necessarily orthogonal and aligned with the variable axes, a small amount of post-processing can describe each terminal node of CART trees as a box identical in form to those of PRIM.

In our application, CART considers a database whose outputs belong to one of two classes. The algorithm successively divides the input space one parameter split at a time (repeated splits on the same parameter are permitted) with the aim of creating multiple regions that contain outputs of a single class. Unlike PRIM, CART is designed to work without user input. As shown in Figure 3, the algorithm partitions the input space with a sequence of binary splits, each designed to minimize impurities on either side. With appropriate parameter settings, the resulting tree can be grown until it describes entirely pure regions. However for most applications the resulting tree would be intractably complicated and any noise in the data may cause over-fitting and suboptimal performance (though in our case, the simulation model generated datasets may not have noise). After growing an initial, highly complicated tree, CART generally 'prunes' it by recombining some nodes in order to find the tree with the best predictive power, accounting for statistical noise. A single CART run thus can generate several trees – the initial, most complicated one and several alternative pruned versions. The user can then choose the tree with the desired balance of predictive performance and interpretability.

CART always partitions the entire input space into disjoint regions in contrast to PRIM whose boxes may overlap. CART's approach may proliferate the number of boxes needed to capture a nuanced shape, but it can also easily maximize density and coverage simultaneously.

4. TESTING THE ALGORTIHMS ON KNOWN SHAPES

We can now test the scenario-discovery abilities of PRIM and CART. This section compares the algorithms on datasets containing simple, known shapes. The next section will compare them on the more realistic challenge posed by datasets with more complex, unknown shapes generated from the results of a computer simulation model.

4.1 Test Procedures

The process shown in Figure 1 envisions using the algorithms interactively. Such user input may enhance an algorithms' usefulness because it will both allow users to participate in their construction and, during the course of the analysis, allow them to make subjective judgments about which box sets might contribute the most to the particular decision support application. PRIM is designed and in practice implemented as an interactive algorithm. Typical implementations of CART offer minimal interaction, although users can select among pruned trees rather than have the algorithm choose them automatically. This difference in interactivity complicates a fair comparison of the algorithms' performance and requires us to employ a test procedure that reduces interactivity relative to that envisioned for an actual scenario-discovery process.

For each test case, we first create a dataset by sampling an input space containing one or more truncated ellipsoidal shapes. As described below, the output at each input point is either 1 or 0 depending on whether a point lies inside or outside the shapes. We run CART and PRIM on the dataset one or more times. Each algorithm generates a set of boxes. We then choose and compare the set from each algorithm that provides the best balance between coverage and density while also providing a good match between the number of boxes and their dimensions and that of the known shapes. Note that we evaluate the quality of box *sets*, rather than individual boxes. Ideally, all boxes in a set should address the measures of merit independently and simultaneously. But comparing only box sets provides sufficient richness for this initial study.

One CART run on the dataset yields a single complicated tree and up to tens of less-complicated subtrees generated by the pruning process. We select the parameters governing CART's performance to generate the most complicated possible initial tree, limited in our case by the amount of data available. We identify each subtree as a candidate box set, and identify each terminal node classifying regions with output equal to 1 as a box. Confining ourselves to those box sets that come closest to matching the number and dimensions of the known shapes in the dataset, we choose that set with the best balance of coverage and density.

In contrast to CART, PRIM can run repeated iterations on the dataset with the user asked after each to choose a box from among many offered. To facilitate a consistent comparison with CART in our testing, we eliminate the need for this interactivity by running four separate groups of PRIM iterations for each dataset. After each PRIM iteration (which generates a peeling trajectory as described in Section 3.1), we choose the box with the best balance of coverage and density by maximizing $Density^{w} Coverage^{1-w}$ with w = {0.5, 0.75, 0.9, 0.99} for each of the four groups of runs. The resulting iterations in each group yield a sequence of boxes {B_1, B_2, …, B_N}. We truncate each sequence to produce a box set that best matches the number and dimension of the shapes in the dataset and then choose the best of the four box sets as our final PRIM result.

4.2 Creating a Results Database

Our test datasets offer six combinations of three-dimensional ellipsoidal shapes of varying number and orientation in a three and a ten dimensional input space. These datasets were chosen to have obvious desired box forms and to provide a range of challenges for CART and PRIM.

Consider two input parameter spaces {x_1, x_2, …, x_n}, $0 \le x_j \le 1$, with n=3 or 10, and four sets of ellipsoids, each truncated by the unit cube or hypercube:

$$\text{Upright Barrel:}\quad 16\left(x_1^{'}\right)^2 + 16\left(x_2^{'}\right)^2 + \left(x_3^{'}\right)^2 < 1 \tag{3a}$$

$$\text{Tilted Barrel:}\ \frac{25}{4}\left(x_1^{'}\right)^2 + 16\left(x_3^{'}\right)^2 + \frac{25}{9}\left(x_3^{'}\right)^2 + 5\left(x_1^{'}x_2^{'} + x_1^{'}x_3^{'} + x_2^{'}x_3^{'}\right) < 1 \tag{3b}$$

$$\text{Crossed Barrels:}\quad \begin{array}{l} 25\left(x_1^{'}\right)^2 + 25\left(x_2^{'}\right)^2 + \frac{9}{4}(x_3^{'})^2 < 1 \\ \frac{9}{4}\left(x_1^{'}\right)^2 + 25\left(x_2^{'}\right)^2 + 25(x_3^{'})^2 < 1 \end{array} \tag{3c}$$

$$\text{Disjoint Barrels:}\quad \begin{array}{l} 25\left(x_1^{'}\right)^2 + 25\left(x_2^{'}\right)^2 + \frac{9}{4}(x_3^{'} - 0.25)^2 < 1 \\ \frac{9}{4}\left(x_1^{'} - 0.25\right)^2 + 25\left(x_2^{'}\right)^2 + 25(x_3^{'})^2 < 1 \end{array} \tag{3d}$$

where $x_i^{'} = x_i - 0.5$ so that each shape is centered in the middle of the space.

As shown in Figures 4, 5, and 6 we consider six combinations of these shapes and spaces: the upright and tilted barrels in the three-dimensional space only and the crossed and disjoint barrels in both the three- and ten-dimensional spaces. We include the ten-dimensional spaces to test the algorithms' ability to distinguish extraneous dimensions. We create the six results datasets for the algorithms by running a 1,000 point Latin hypercube sample over each combination of shapes and spaces, assigning a 1 to each point inside a shape and a 0 otherwise. The underlying shapes range in precise volume from .163 to .281 in an input space of volume 1, so each sample has approximately 160 to 280 interesting points, with a variance consistent with the standard error introduced by the stochasticity of the Latin Hypercube experimental design. For these simply defined shapes, computational limitations would present few restrictions to generating larger samples. However, we confine ourselves to 1,000 points to facilitate comparison with the samples sizes convenient for the simulation model dataset discussed in the next section.

4.2 Results from Simple Test Cases

The upright and tilted barrels in the three dimensional input space are the simplest shapes in our test suite. Figure 4 shows these ellipsoids and the best box sets generated by PRIM and CART from the datasets generated by these shapes. The PRIM results were generated from the w =0.5 group of iterations. The figure also reports the coverage, density, number of boxes, and the total number of dimensions restricted by each algorithm. The leftmost column reports these measures for the optimal box of given form, generated by running a custom greedy hill-climbing algorithm on the dataset from hundreds to thousands of starts in box-boundary space. This algorithm uses a pre-specified number of boxes and dimensions to optimize the term $Density^{0.5}Coverage^{0.5}$. Such an optimization algorithm proves useful for benchmarking our test cases whose shapes are known a priori, but infeasible for finding more complicated, unknown shapes in data.

Fig 4 shows that both PRIM and CART describe the upright and tilted barrels reasonably well. Both algorithms successfully identify box sets with the correct number of boxes (one) and restricted dimensions (two). For the upright barrel, both algorithms generate density and coverage similar to one another and close to that of the optimal box. A rectangular box provides a poorer description (lower coverage and density) for the tilted than for the upright barrel. PRIM's box for the tilted barrel closely matches the optimal box, while CART's box provides a different balance of coverage and density. In this case, PRIM is closer to the optimal because the objective function used to choose a box from PRIM's peeling trajectory under our test procedure is identical to that used to calculate the optimal box.

The crossed and disjoint barrels present a more complicated challenge. Figure 5 shows these two shapes in a three dimensional space, the best box sets generated by PRIM and CART, and the performance measures for the optimal box set. The PRIM results were generated with w=0.5 for the disjointed barrels and w=0.75 for the crossed barrels. In contrast to the previous single-box cases, the algorithms differ more significantly with each other and with the optimal boxes.

For both shapes, CART generates more boxes than necessary. This proliferation of boxes owes to the algorithm's iterative disjoint partitioning. For the disjoint barrel, the partition requires the splits defining the upper and lower surfaces of the horizontal barrel to extend through the entire space and effectively cut the box that would otherwise describe the vertical barrel into three sections. Though this partitioning degrades CART's performance on the interpretability measure, the algorithm does capitalize on the additional degrees of freedom provided by the extra boxes to generate a better density-coverage combination than the optimal two-box solution.

PRIM generates the correct number of boxes for both the disjoint and crossed barrels. The ability to create overlapping boxes for the crossed barrels improves the description of these shapes. However, for the cross shape, the algorithm yields significantly less coverage than the optimum because it slices off the ends of the barrels. This behavior owes to a combination of PRIM's "patient" peeling trajectory and the stochastic variation in a sparse sample of points. PRIM starts its peeling trajectory by taking small slices from the edges of the input space and can increase its density-coverage performance if the ends of the barrels are insufficiently populated in the experimental design.

Increasing the dimensionality of the space causes some problems for PRIM but less for CART. Figure 6 shows the projection into $\{x_1, x_2, x_3\}$ space of the disjoint and crossed barrels in a ten dimensional space $\{x_1, \ldots, x_{10}\}$, the $\{x_1, x_2, x_3\}$ projection of the best box sets generated by PRIM and CART, and the performance measures for the optimal box set. The PRIM results were generated with w=0.99 for the disjoint shapes and 0.5 for the cross shape. This case challenges the algorithms with seven extraneous dimensions, similar to actual scenario-discovery applications where many of the input dimensions may prove less relevant than others to describing the space of interesting results.

The extra dimensions have little effect on CART's performance. The algorithm generates box sets of similar form, and for the disjoint case they are exactly the same. However, even though CART does not restrict extraneous dimensions in the cases shown, the extra dimensions do cause a slight deterioration in box quality for the 10-d cross. Specifically, the extra dimensions prevent CART from making the additional split that led to the higher density in the 3-d case.

The extra dimensions have a more significant effect on PRIM. For both shapes the algorithm restricts at least one extraneous dimension. This behavior owes to PRIM's patience combined with the sparseness of sampling. When the space is not infinitely filled with points and the underlying shapes have non-orthogonal boundaries, it is possible to increase the density-coverage measure by restricting dimensions that do not play a role in defining the shape. PRIM's patience makes it more likely than

CART to capitalize on these spurious density gains, since it is generally able to fragment the data much more slowly. However, we did observed cases where CART also restricts extraneous dimensions when asked to generate a higher number of small boxes to more precisely cover the shape.

PRIM also creates a box in the disjoint case that inappropriately spans the two barrels. Because the 3-d data set used for these examples is identical to the 10-d set in the first three dimensions, this odd behavior is most likely caused by a combination of the extraneous dimensions and low sampling density. Increasing 10-d sampling density to 50,000 points caused the problem to disappear in multiple trials, though it sometimes appeared at a density of 10,000 points. CART did not appear to suffer this specific problem in our test cases, most likely thanks to its partitioning mechanism.

4.3 Appropriate Use of Objective Functions for Scenario Discovery

The PRIM and CART boxes shown in Figures 4, 5, and 6 were chosen with knowledge of the underlying shapes. But each algorithm generates multiple box sets and in actual practice the user would have to chose among them without such knowledge. In principle, an objective function with the appropriate weightings for coverage, density, and interpretability might determine the best box set and eliminate the difficulty and subjectivity that accompanies the inclusion of user interactivity in the scenario discovery process. It appears, however, that formalized objective functions prove practically difficult to implement without significant extra knowledge of the specific dataset.

Figure 7 provides two examples of the potential difficulties of relying solely on an objective function for scenario discovery. The box sets generated by the algorithms and displayed in the figure seem undesirable as descriptions of the underlying shapes, but with some coverage/density weightings could score better than those in Figure 6.

In the first example, CART generates a box set with asymmetries not found in the underlying function, leaving the top of the vertical barrel uncovered. Random fluctuations in the sampling density leave this upper portion of the barrel less densely populated with points so that CART's best three-box set excludes this region. As shown in Figures 5 and 6, PRIM can also generate asymmetric box sets, but the asymmetries are likely to be less dramatic thanks to the algorithm's patience in peeling.

In the second example, PRIM generates extra boxes due to needless nesting. The algorithm's covering process allows it to improve coverage significantly with a minor loss in density by describing the single horizontal barrel with two boxes. In many applications, the gain in coverage or density may not be worth the resulting loss of subjective interpretability.

These examples suggest two general types of problems that arise when choosing among box sets. In general we prefer box sets with fewer boxes, but should not penalize a set that uses multiple boxes to capture an equivalent number of genuinely disjoint box-shaped regions. We might however wish to penalize a box set that uses two boxes to capture one rectilinear shape such as the upright barrel.

We also prefer box sets that restrict a small number of parameters. But to appropriately weight penalties for high dimensionality, one must identify the threshold at which gains in density outweigh the loss of interpretability, and possibly coverage, from restricting an additional dimension. Because these judgments are subjective, there is no reason to think the threshold will be the same for all datasets, or even independent of the number of boxes in a particular set.

These considerations suggest that while an objective function may prove useful in eliminating egregiously dominated box sets, one should not rely on it to return the most desired box set. Indeed, we tested a range of objective functions each combining a term balancing converge and density with terms that penalize box sets containing a number of boxes or dimensions in excess of some specified thresholds. After densely sampling many combinations of thresholds, penalties, and coverage/density weightings, we were unable to find a single objective function that reproduced all the desired box sets shown in section 4.2. Thus, an objective function may best serve the scenario discovery task as part of an interactive approach that assists users in generating and selecting box sets that lie along an efficient frontier.

5. APPLICATION TO AN UNKNOWN SHAPE

We now test the scenario discovery algorithms under conditions relevant to an actual decision analytic application. We use the algorithms to identify policy-relevant regions in model-generated data containing shapes initially unknown to the user.

We have chosen for this test a simple but highly non-linear simulation model we have explored extensively in other work (Lempert and Collins 2007). The model, based on work by Peterson et al (2003), considers how much pollution a decision maker can safely add to a lake without causing a shift to an undesirable and potentially irreversible eutrophic state. The decision maker gains utility by adding pollution, but suffers large utility loss if the lake becomes eutrophic. The model tracks the lake's pollution concentration over time as it changes due to human and natural flows. The lake becomes eutrophic if the concentration exceeds some threshold value initially unknown to the decision maker. The decision maker can learn about the location of the threshold over time. The model assesses the desirability of alternative strategies for adding pollution to the lake, contingent on eleven potentially uncertain input parameters describing the critical threshold, the lake's natural pollution flows, the lake's ability to absorb pollution, the decision maker's ability to learn about the location of the critical threshold, and the costs of the eutrophic state.[ii]

Lempert and Collins (2007) conducted an RDM analysis with this model to identify a promising emissions strategy, one that begins with very low emissions that gradually increase over time as the decision maker learns more about the location of critical threshold. The emissions level off at a value that keeps the pollution concentration safely below the critical threshold once the decision maker understands where that threshold lies. This emissions strategy is robust in the sense that its expected utility is close to the optimum expected utility for any value of the critical threshold.

This previous analysis, however, considered only one of the model input parameters as uncertain, that which describes the location of the critical threshold. The other parameters were held constant at their nominal values. We can thus test the scenario discovery algorithms by using them to examine the robustness of the candidate emissions strategy over variations of all eleven model input parameters.

5.1 PRIM and CART Boxes

To create a dataset on which to test the PRIM and CART algorithms, we first construct a 1000-point Latin hypercube sample over the 11-dimensional input parameter space. Each point in this space represents one plausible state of the world relevant to the decision problem. Because this study focuses on the scenario-discovery algorithms, not the model or the decision, we normalize all input parameters to lie on the interval [0,1] and label the input parameters X_i, where $i = 1,...,11$, rather than use their more substantively evocative names. At each point in the dataset we calculate the regret of the candidate strategy, defined as the difference between the optimum utility and the utility of the candidate strategy for that state of the world. This regret provides a convenient measure of the strategy's robustness. We apply a satisficing criterion that distinguishes between states of the world where the decision maker finds the candidate strategy's regret acceptably low or undesirably high. We find that 152 of the 1000 points in the dataset have such high regret. Our scenario discovery algorithms can now address the following policy-relevant question: which combinations of which model input parameters best describe those regions of the input space where the candidate strategy has high regret?

PRIM's answer is shown in Figure 8. Using PRIM's diagnostic features and applying human judgment to mediate tradeoffs between the measures of merit, we identified two boxes described by restrictions on three parameters each, with one dimension common to both. Together the boxes cover 73% (111) of the 152 high regret points and have a density of 54%. The latter is about 3.6 times higher than the density of high regret points in the entire space. This coverage and density performance is significantly below that for the three-dimensional ellipsoids in a ten dimensional space seen in Fig 6, which is not surprising since there is no reason to believe that the high regret regions for this model have such a simple shape.

The first PRIM box contains 74 points of which 60 have high regret, for density of 81% and coverage of 39%. By itself, the second PRIM box contains 152 points of which 71 have high regret, for a density of 47% and coverage of 47%. The two boxes overlap with 22 points, of which 20 have high regret.

Figure 9 shows the analogous results suggested by CART, after running it automatically and choosing the two-box set from among multiple box sets returned. The algorithm generates two boxes, the first described by restrictions on three and the second on five parameters. The boxes cover 57% (87) of the high regret points and have a density of 76%, about five times that of the high regret points in the entire space. The first CART box is almost identical to the first PRIM box. The second has similar restrictions on the X_1 and X_7 parameters, but also restricts X_4, X_9, and X_{10} while the second PRIM box only additionally restricts X_2.

The first CART box contains 82 points of which 63 have high regret, for density of 77% and coverage of 41%. The second CART box contains 32 points of which 24 have high regret, for a density of 75% and coverage of 16%. As mentioned in Section 3, CART boxes will never overlap, and these two boxes are indeed disjoint.

Since they are so different, it is useful to examine the extent to which the second PRIM and CART boxes capture different high regret points or whether they offer different descriptions of the same set of points. We note that the CART box with its 32 points is much smaller and has higher density than the PRIM box, with 130 points and 47% density. The PRIM box contains 20 of the CART box's 32 points, 17 of which are high regret. Thus, the second CART box only provides 7 new high regret cases over the PRIM box. We thus conclude that the second CART box is primarily capturing a small subset of the PRIM box (sacrificing coverage for density), with the second PRIM box essentially encompassing the CART box.

5.2 Do the Boxes Provide Significant and Reproducible Summaries of the Regions?

The differences between the PRIM and CART boxes, and their coverage and density performance, suggests neither provides a perfect description of the candidate strategy's high regret regions. In part, rectilinear boxes may only imperfectly approximate these regions. Sampling density, the algorithms' idiosyncrasies, or similar factors may also influence the PRIM and CART boxes. It is thus useful to test the statistical significance and reproducibility of the four boxes in Figures 8 and 9. We perform two types of tests, those estimating the statistical significance of each restriction on an input parameter and those examining the similarities among box sets generated from bootstrapped data. This analysis suggests that the first PRIM and CART boxes provide stable descriptions of the rectilinear core of a policy-relevant region while the second PRIM and CART boxes provide two different, imperfect approximations to the remaining and highly non-rectilinear regions of high regret cases.

We can estimate the statistical significance of PRIM and CART's proposed restrictions on each parameter by assuming a known distribution of high regret points within the boxes. Consider a given box *B* with restricted dimension *d* and the larger box *B'* with no restriction on *d*. Assuming a binomial distribution of high regret points, we can estimate the probability of finding a box of size B within *B'* which has density greater than or equal to *B*. Applying this test suggests that all the parameters restricted by all four boxes are significant at the 5% level with three possible exceptions. The X_2 restriction in PRIM's second box is just barely insignificant, with a probability 5.2%. More interestingly, removing the X_4 restriction from CART's second box actually *improves* the mean, up to 77.5%.[iii] Finally, the most noticeable difference between the first PRIM and CART boxes is the former's restriction on the high values of X_3. The analysis in Section 4 suggests that PRIM can inappropriately clip the ends of parameter ranges when the sampling density proves too low. Assuming a binomial distribution of high regret points in the first PRIM box but with no upper restriction for X_3, we estimate a 7.5% likelihood that PRIM would restrict the upper end by chance.

We can test the reproducibility of the PRIM and CART boxes by comparing box sets generated from pseudo-bootstrapped resamplings of the original data. Table 1 shows the frequency with which each parameter was restricted by each algorithm in at least one box over ten such samples. Each sample has roughly 620 of the original 1000 cases, due to elimination of duplicated samples.[iv] Only the X_1, X_4, X_9, and X_{11} parameters are treated similarly in all cases, the first two restricted by both algorithms in all ten samples and the last two never restricted in any sample by either algorithm. In general, the algorithms restrict each parameter with similar frequency, with the largest differences arising for X_3 and X_8.

We can also test for reproducibility by comparing the number of points common to two box sets generated on different resamplings relative to the number of points in their union. For distinct consistent box sets this intersection to union ratio should lie very close to one. Running this test pairwise on each box set we find that CART's box sets have an average intersection to union ratio of 0.40 ± 0.14 while PRIM's boxes have an insignificantly lower average ratio of 0.35 ±.09.

These tests all suggest that the first PRIM and CART box in each set provide a reasonable description of a box-like region of high regret for the candidate strategy, while the second PRIM and CART boxes struggle to describe the remaining high regret points which are poorly approximated as a rectilinear region. First, both PRIM and CART produce nearly identical first boxes with high density, 81% and 77%, respectively, compared to different second boxes, where CART's high density box (75%) is largely contained inside PRIM's low density (47%) box. Second, the significance tests suggest that, with the possible exceptions of PRIM's restriction of the high end of X_3, the parameter descriptions on the first boxes are all statistically significant while some second box restrictions are not. Third, the reproducibility tests suggest that for both algorithms two of the three parameters defining the first boxes are restricted in every box set on the resampled data while the parameters describing the second PRIM and CART boxes are only one of many possible descriptions across the resampled data. Finally, the intersection to union ratio across these samples, roughly 40%, approximates the coverage for the first CART and PRIM boxes, consistent with the interpretation that one box from each algorithm in every sample captures the same rectilinear set of high regret points.

One of the most important, if not the most subjective, questions about the boxes generated by PRIM and CART reflects whether or not they provide useful insights about the candidate strategy as described by the model. It is beyond the scope of this paper to provide a sufficient description of the model and its behavior to enable readers to judge for themselves whether the boxes in Figures 8 and 9 provide such unique and novel information. However, one of the authors had worked extensively with the simulation model before exploring its output with PRIM and CART, was surprised by the boxes, and believes that the scenario discovery algorithms did indeed suggest scenarios that would enrich a policy analysis.

6. IMPROVING THE ALGORITHMS FOR SCENARIO DISCOVERY

Our tests on simple, known shapes and on simulation model-derived data suggest that both PRIM and CART can perform the scenario discovery task, but often imperfectly. PRIM can restrict too many dimensions and span disjoint regions with single boxes. CART can generate too many and inappropriately asymmetric boxes. Neither algorithm does well when presented with non-rectilinear regions. We have identified several modifications that might enhance the algorithms' performance: adding visualizations that specifically report measures of merit for scenario discovery, modifying the algorithms' objective functions to focus on these measures, and employing a post-processing step that amalgamates multiple small boxes into a single box.

Friedman's publicly available PRIM software package[v] assists the user in choosing boxes with visualizations (see the support line in Fig 10) that show the "peeling trajectory," a series of boxes of increasing density and decreasing support. While support provides an appropriate measure for PRIM's original "bump-hunting" purpose, it is not identical to the coverage measure appropriate for scenario discovery. In particular, support always decreases with increasing density because the former measures the number of points in the box. In contrast, density can increase with no effect on coverage if constricting a box excludes only non-policy-relevant points. Thus plotting density against coverage, as opposed to density against support, can reveal convex corners that indicate particularly good choices for scenarios. As shown in Fig 10, such corners can reveal the smallest box that captures all the policy-relevant points. In some cases, vertical regions of density-coverage plots may also reveal boxes that lie at the boundaries of disjoint regions of policy relevance. Providing such visualizations might help users identify boxes with better coverage/density performance.

More generally, PRIM and CART software might more effectively assist scenario discovery by displaying for users the tradeoffs among any combination of the multiple scenario measures. For instance, 3-d or contour plots relating density, coverage and various interpretability measures would allow users to look for convex corners among these scenario attributes. The interpretability axis could usefully represent box number (for box sets), number of dimensions restricted (for a one box case), or some combination of the two. Alternatively, a 3-d visualization could display multiple interpretability measures – for example box number and total number of dimensions restricted -- by combining density and coverage with a single index measure for analytic quality. Adding such visualizations represents an entirely feasible computational and algorithmic extension to the PRIM and CART software packages.

In addition to new visualizations, modifications to PRIM and CART's objective functions might help them offer users scenario boxes that expand the density-coverage-interpretability frontier. PRIM's peeling criterion may be modified to incorporate interpretability features. For example, the required gains in density from restricting a new dimension could be made higher than the threshold required to further constrain a dimension that has already entered the box definition. Similarly, CART might incorporate a combined coverage-density measure, rather than using misclassification rates or the Gini index as a criterion for judging its splits.

CART's tendency towards too many boxes and both algorithms' difficulty with non-rectilinear regions might be addressed by post-processing their output to amalgamate multiple smaller boxes into a single larger object. For instance, such an amalgamation algorithm might meld the three CART boxes describing the vertical barrel in the upper left-hand panel of Figures 5 and 6 into a single box which minimally bounds all three. Alternatively, several boxes might be combined into a non-rectilinear yet conceptually simple object (such as a wedge) to describe the region of policy-relevant points poorly approximated by the second PRIM and CART boxes in Figures 8 and 9. Such amalgamation will in general improve the interpretability measure of the resulting box set, but require some sacrifice of coverage and/or density.

This amalgamation process might prove valuable for both algorithms, allowing them to provide relatively concise descriptions of non-rectilinear regions with combinations of boxes. But the process might prove most useful for CART. On the simpler datasets, CART generates better coverage/density performance than PRIM. But we have generally found PRIM more useful for actual policy analysis applications because for complex datasets CART often fails to reach sufficient levels of coverage before requiring an infeasible large number of boxes. In principle, amalgamation would eliminate this difficulty though perhaps at significant computational expense. While amalgamation may prove a valuable tool for scenario discovery, the algorithms needed to perform this task are not yet obvious to the authors.

7. CONCLUSIONS

While attractive in many ways, traditional scenarios have lacked an appropriate analytic foundation for inclusion in quantitative decision analyses. This study helps to remedy this situation by examining the ability of two types of statistical algorithms to systematically identify policy-relevant scenarios in simulation model-generated data and proposing a framework for assessing the results of such a scenario discovery process.

The study offers an interactive process and three measures of merit for scenario discovery: coverage, density, and interpretability. Using these measures, the study evaluates the ability of PRIM, a bump-hunting algorithm, and CART, a classification algorithm, to find low-dimensional, hyper-rectangular combinations of input parameters with high predictive power for interesting cases in a database of model results. We find both algorithms can perform this scenario discovery task, but often imperfectly. PRIM can restrict too many dimensions and span disjoint regions with single boxes. CART can generate too many and inappropriately asymmetric boxes. The study proposes statistical tests to help identify such problems and suggests some simple modifications to the algorithms and their implementing software that might improve their scenario discovery capabilities.

This study does not consider the effects of sampling density on the algorithms' performance. But sample size should in general have an important influence on the quality and reproducibility of the boxes they generate. In practical applications, the computer time required to create a large results-database from complicated simulation models may prove an important constraint. The question of sampling density and the best experimental designs for scenario discovery thus represent an important topic for future research.

PRIM and CART may offer only an initial step in the search for good algorithms for scenario discovery. Both attempt to describe the policy-relevant regions in a model-results database as rectilinear, not necessarily the best approximation in many cases. Algorithms that generate other shapes could prove useful, though with a possible loss of interpretability. While PRIM offers more interactivity than CART, both are designed for fixed datasets, so neither aggressively exploits the ability to generate additional data on demand that is available with simulation models. New machine learning approaches based on active learning (Bryan, et.al., 2005; Cohn, et.al., 1995) may address sampling density challenges by iteratively requesting new simulation model runs that add cases to the database in only those specific regions where they are most needed to improve the performance of the scenario discovery algorithms.

Despite these yet-untapped opportunities and current shortcomings, the initial methods proposed here should support a wide range of useful decision analytic applications and may help provide a new quantitative foundation for thinking about scenarios.

REFERENCES

Amaratunga, D. and J Cabrera. 2004. Data mining to find subsets of high activity. Journal of Statistical Planning and Inference. 122 23 – 41

Bankes S.C. 1993. Exploratory Modeling for Policy Analysis. Operations Research 41 (3) 435-449.

Brandstein, A. and G. Horne. 1998. Data Farming: A Meta-Technique for Research in the 21st Century. *Maneuver Warfare Science 1998*, Marine Corps Combat Development Command Publication, Quantico, Virginia.

Breiman, L., J.H. Friedman, R.A. Olshen, and C.J. Stone, 1984. Classification and Regression Trees. Chapman & Hall, London.

Bryan, B, J. Schneider, R. C. Nichol, C. J. Miller, C. R. Genovese, and L. Wasserman, 2005, "Active Learning For Identifying Function Threshold Boundaries", Advances in Neural Information Processing Systems, volume 17, The MIT Press.

Cohn, D. A. , Z. Ghahramani, and M. I. Jordan. 1995. Active learning with statistical models. In G. Tesauro, D. Touretzky, and T. Leen, editors, Advances in Neural Information Processing Systems, volume 7, pages 705–712. The MIT Press,.

Dixon L, R. Lempert, T. LaTourrette, R..T. Reville, and P. Steinberg. 2007. Tradeoffs among alternative government interventions in the market for terrorism insurance. RAND. http://www.rand.org/pubs/documented_briefings/DB525/

Friedman, J.H. and N.I. Fisher. 1999. Bump hunting in high-dimensional data. Statistics and Computing 9, 123–143.

Groves, D.G. and R.J. Lempert. 2007. A new analytic method for finding policy-relevant scenarios. Global Environmental Change, 17 73-85

Lempert, R.J. and M.T. Collins. 2007. Managing the Risk of Uncertain Threshold Responses: Comparison of Robust, Optimum, and Precautionary Approaches. Risk Analysis. in press.

Lempert, R.J., D.G. Groves, S.W. Popper, and S.C. Bankes. 2006. A general, analytic method for generating Robust strategies and narrative scenarios. Management Science 52 (4), 514–528.

Lempert, R. J., S. W. Popper, and S. C. Bankes. 2003. Shaping the Next One Hundred Years: New Methods for Quantitative, Long-Term Policy Analysis, RAND.

Parson, E.A., Burkett, V., Fischer-Vanden, K., Keith, D., Mearns, L., Pitcher, H., Rosenweig, C., and Webster, M., 2006. Global-Change Scenarios: Their Development and Use, Synthesis and Assessment Product 2.1b. US Climate Change Science Program.

Peterson, G.D., S.R. Carpenter, and W.A. Brock. 2003. Uncertainty and the management of multistate ecosystems: An apparently rational route to collapse, Ecology, *84* (6), 1403-1411.

Saltelli, A., K. Chan, and M. Scott. 2000. Sensitivity Analysis. Wiley, New York, NY.

Schwartz, Peter, 1996: The Art of the Long View. New York, Double Day.

[i] Our previous work has defined deep uncertainty as the situation where decision makers do not know or do not agree on the system model relating actions to outcomes and/or the prior probability distributions over the key inputs to the the system model(s).

[ii] See Lempert and Collins (2007) for a complete description of this model

[iii] This error suggests a potential advantage of PRIM's pasting step, which searches for such improvements.

[iv] Because our purpose is gauging stability on datasets with unique observations rather than improving model fit, we eliminated duplicate points in the bootstrap sampling, which one would normally retain in a bagging or boosting analysis.

[v] http://www-stat.stanford.edu/~jhf/SuperGEM.html

TABLES

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}
PRIM	100	40	60	100	10	60	10	70	0	30	0
CART	100	40	30	100	0	50	20	30	0	20	0

Table 1. Percentage of bootstrapped datasets from test simulation model results in which each algorithm restricts each model input parameter

FIGURES

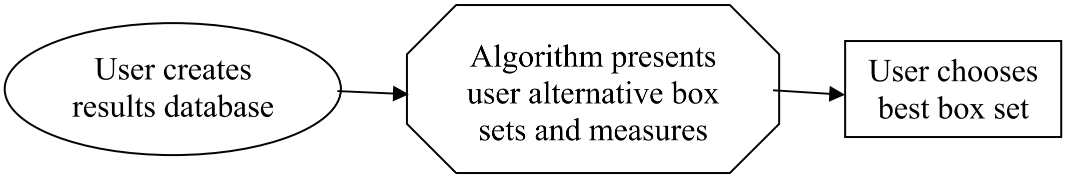

User creates results database
Algorithm presents user alternative box sets and measures
User chooses best box set

Figure 1: Proposed scenario discovery process

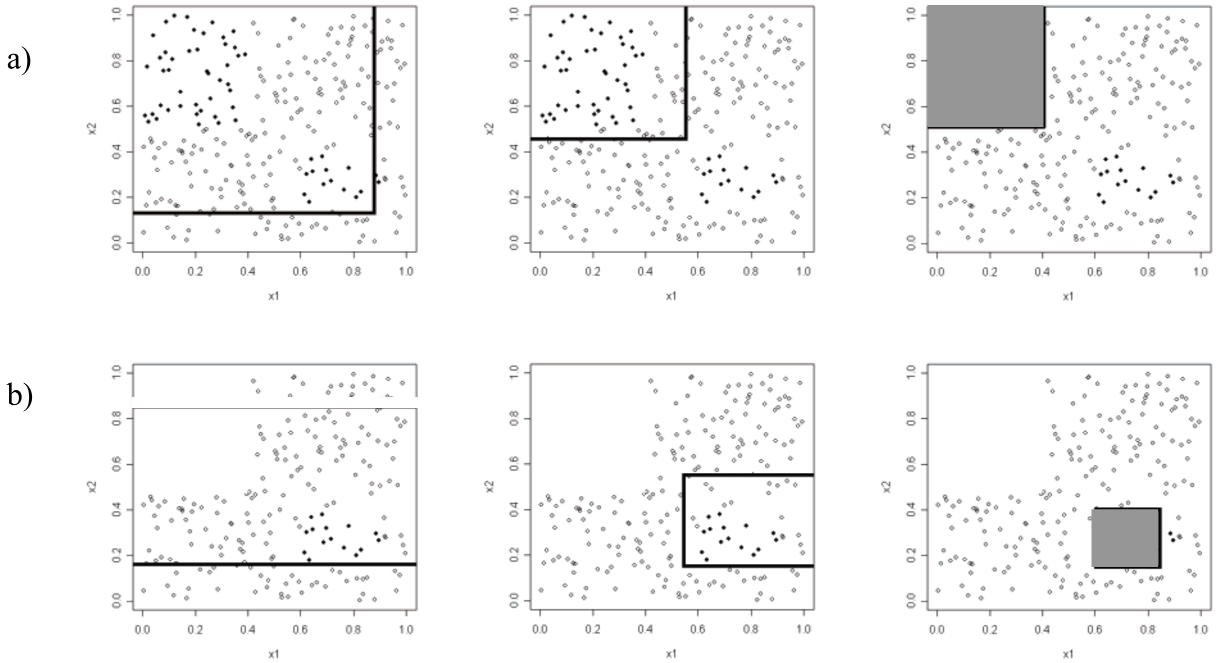


Figure 2: Conceptual illustration of PRIM's: a) peeling and b) covering process.

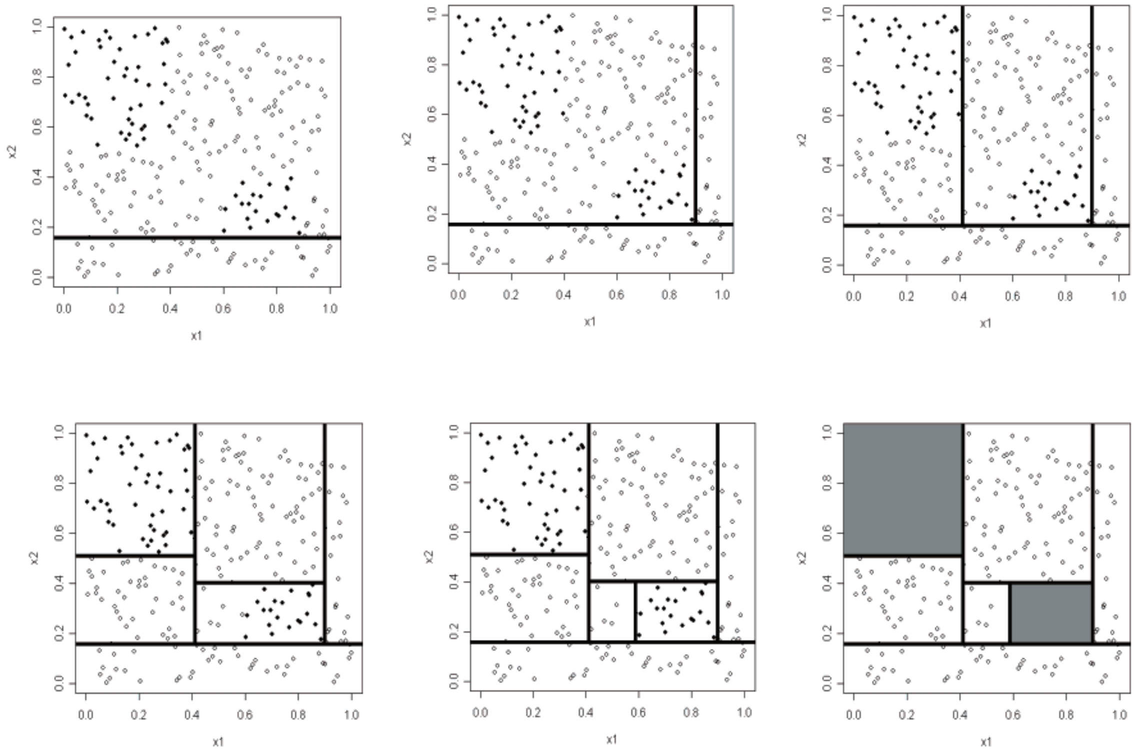


Figure 3: Conceptual illustration of CART's partitioning process

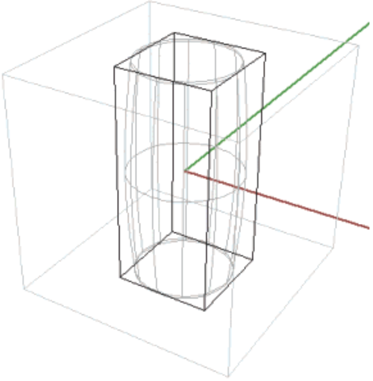
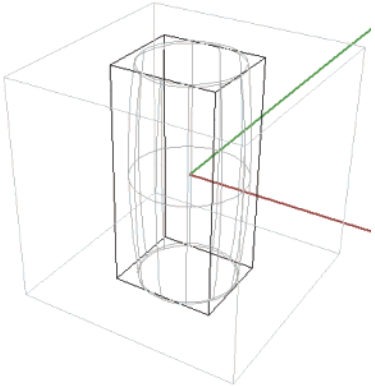
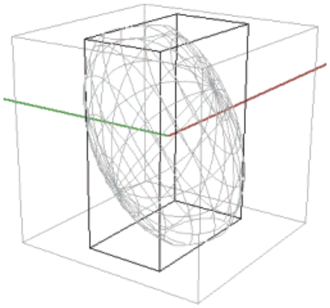
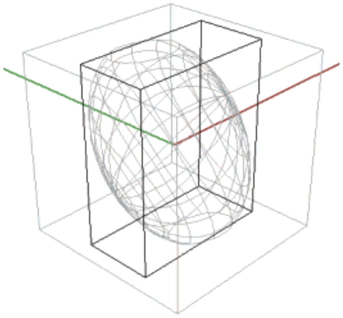
OPTIMUM	CART	PRIM
Upright barrel in 3D space Density: 92 Coverage: 94 Max box dim: 2 Boxes: 1	 Density: 89% Coverage 95% Max box dim: 2 Boxes: 1	 Density: 92% Coverage 90% Max box dim: 2 Boxes: 1
Tilted barrel in 3D space Density: 78 Coverage: 89 Max box dim: 2 Boxes: 1	 Density: 81% Coverage 84% Max box dim: 2 Boxes: 1	 Density: 77% Coverage 87% Max box dim: 2 Boxes: 1

Figure 4: Optimum, CART, and PRIM results for upright and tilted barrels.

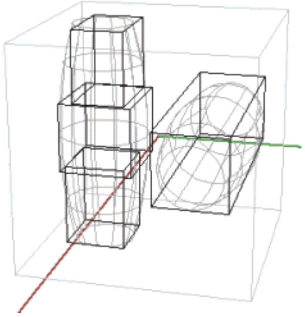
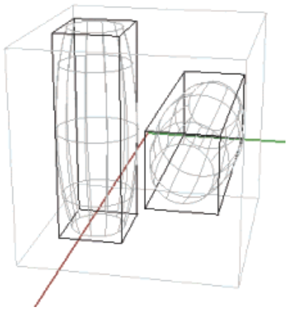
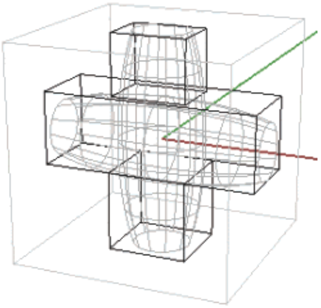
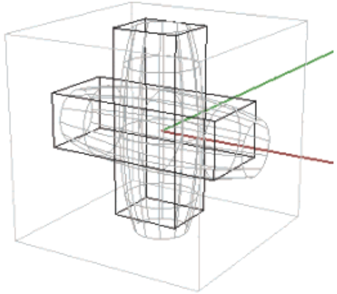
OPTIMUM	CART	PRIM
Disjoint barrels in 3D space Density: 91 Coverage: 95 Max box dim: 2 Boxes: 2	 Den: 93% Coverage 93 % Max box dim: 3 Boxes: 4	 Den: 88% Coverage 94% Max box dim: 2 Boxes: 2
Crossed barrels in 3D space Density: 92 Coverage: 93 Max box dim: 2 Boxes: 2	 Den: 84% Coverage 97% Max box dim: 3 Boxes 3	 Den: 94% Coverage 84% Max box dim: 3 Boxes: 2

Figure 5: Optimum, CART, and PRIM results for disjoint and crossed barrels in three-dimensional input space.

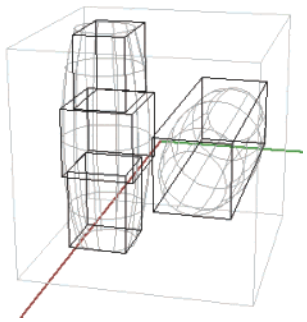
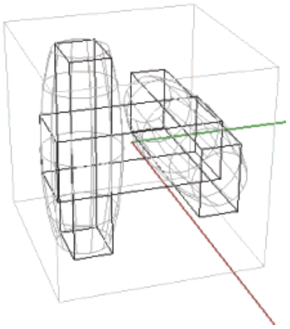
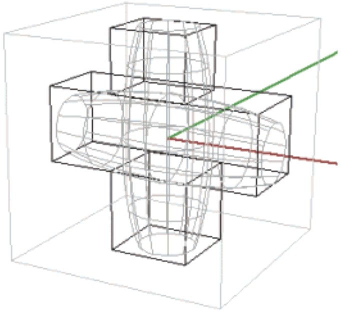
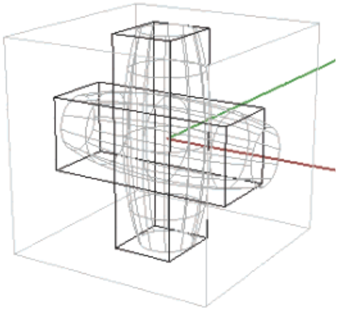
OPTIMUM	CART	PRIM
Disjoint barrels in 10D space Density: 91 Coverage: 95 Max dims: 2 Boxes: 2	 Den: 93% Coverage 93% Max box dim: 3 Boxes: 4	 Den: 100% Coverage 72% Max box dim: 6 Boxes: 3
Crossed barrels in 10D space Density: 92 Coverage: 93 Max dims: 2 Boxes: 2	 Den: 82% Coverage 97% Max box dim: 3 Boxes: 3	 Den: 90% Coverage 89% Max box dim: 4 Boxes: 2

Figure 6: Optimum, CART, and PRIM results for disjoint and crossed barrels in ten-dimensional input space.

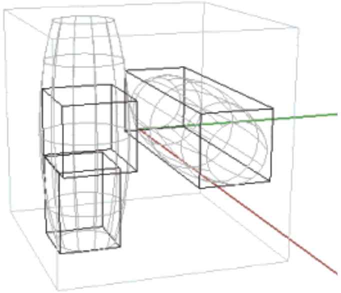
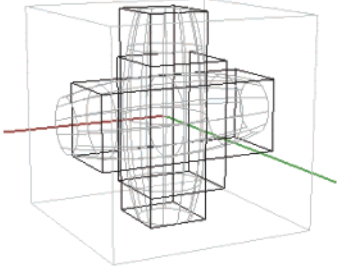
CART	PRIM
Disjoint barrels in 3D space  Den: 87 Coverage: 87 Max box dim: 3 Boxes: 3	Crossed barrels in 10D space  Den: 87 Coverage: 97 Max box dim: 4 Boxes: 3

Figure 7: Examples of undesirable box sets generated by CART and PRIM.

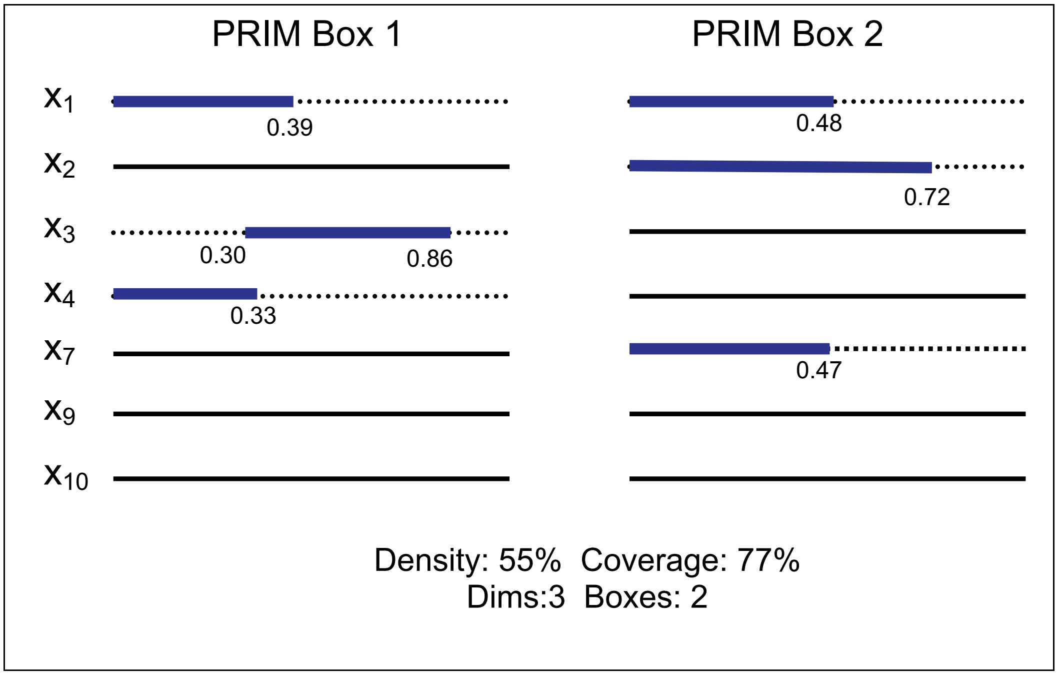


Figure 8: PRIM boxes for test simulation model results

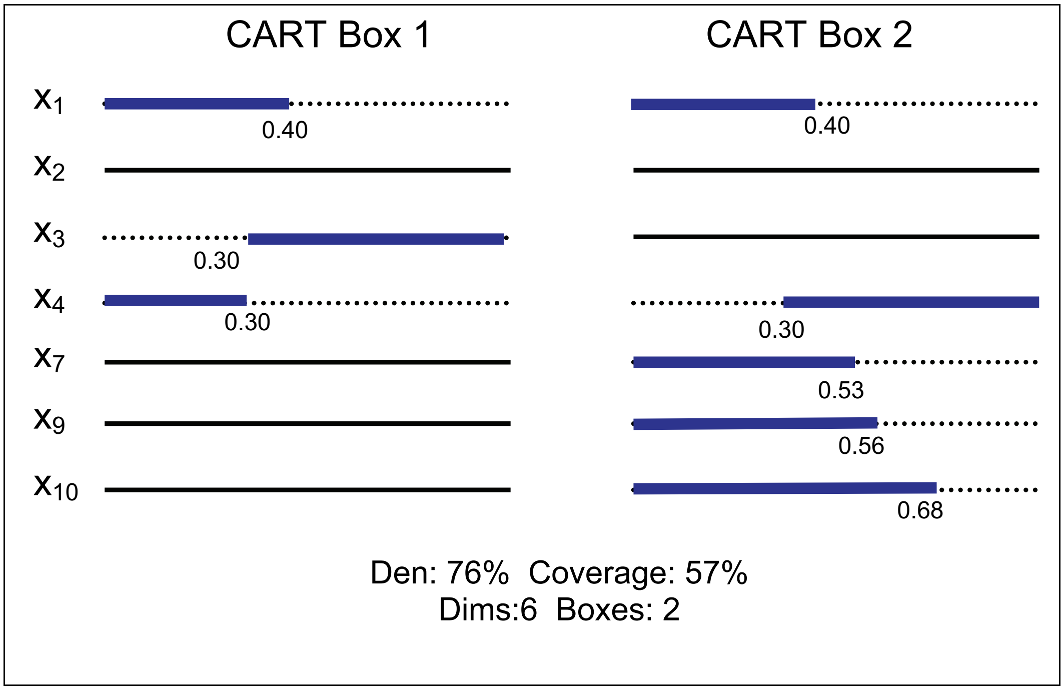


Figure 9: CART boxes for test simulation model results

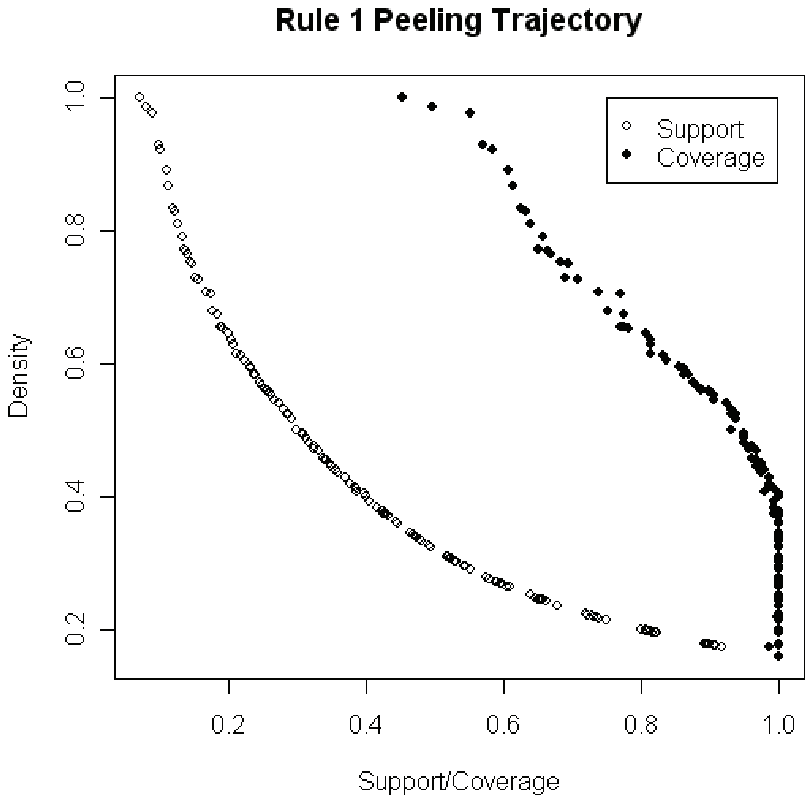


Fig 10: Comparative visualizations of PRIM peeling trajectory showing density against support and coverage