SANDIA REPORT

SAND2013-2534
Unlimited Release
Printed February 2013

Builtin vs. Auxiliary Detection of Extrapolation Risk

M. Arthur Munson, W. Philip Kegelmeyer

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Approved for public release; further dissemination unlimited.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online ordering: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2013-2534
Unlimited Release
Printed February 2013

Builtin vs. Auxiliary Detection of Extrapolation Risk

M. Arthur Munson
Sandia National Laboratories
P.O. Box 969, MS 9159
Livermore, CA 94551
mamunso@sandia.gov

W. Philip Kegelmeyer
Sandia National Laboratories
P.O. Box 969, MS 9159
Livermore, CA 94551
wpk@sandia.gov

Abstract

A key assumption in supervised machine learning is that future data will be similar to historical data. This assumption is often false in real world applications, and as a result, prediction models often return predictions that are extrapolations. We compare four approaches to estimating extrapolation risk for machine learning predictions. Two *builtin* methods use information available from the classification model to decide if the model would be extrapolating for an input data point. The other two build *auxiliary* models to supplement the classification model and explicitly model extrapolation risk. Experiments with synthetic and real data sets show that the auxiliary models are more reliable risk detectors. To best safeguard against extrapolating predictions, however, we recommend combining builtin and auxiliary diagnostics.

Acknowledgment

We thank Levi Lloyd and Ken Chiang for help with collecting data, and Tammy Kolda, Margot Kimura, and Todd Plantenga for feedback and interesting discussions.

Contents

1 Introduction . . . 7
2 Extrapolation Risk Measures . . . 9
2.1 Risk Definition . . . 9
2.2 Builtin: Margin Risk . . . 9
2.3 Builtin: Forest Dispersion Risk . . . 10
2.4 Auxiliary: CERT Forest . . . 11
2.5 Auxiliary: Chaos Forest . . . 12
2.6 Baseline: Box Risk . . . 12
3 Implementation Details . . . 13
4 Comparing Builtin vs Auxiliary . . . 15
4.1 Synthetic Data . . . 15
4.2 Detecting an Unknown Class . . . 19
5 Discussion . . . 22
References . . . 24

Appendix

A Extra Information about Extrapolation Risk Experiments . . . 26

Figures

1 Example 2D mixture of ridge densities. . . . 17
2 The CERT learning algorithm can stop tree growing prematurely. . . . 23

Tables

1 Pruning is not needed for CERT forests. . . . 13
2 Average error for synthetic Gaussian mixture data. . . . 16
3 Average error for synthetic ridge mixture data. . . . 18
4 Data sets used in unknown class experiment. . . . 20
5 AUC for detecting class not seen during training. . . . 21

1 Introduction

A key assumption in supervised machine learning is that future data will be similar to labeled historical data. Without this assumption, there would be no basis for models to make predictions. Real world applications often violate this assumption to a greater or lesser extent: some future data may be outliers, a novel class might appear, or maybe an adversary changes tactics. In these cases, the learned prediction model is not qualified to make a prediction; any prediction is an extrapolation beyond its experience (that is, its training data). For domains where mistakes incur high costs (e.g., medicine), it is important to detect when a model's prediction is an extrapolation so that domain experts can intervene and provide a qualified analysis.

We consider the problem of quantifying extrapolation risk for domains containing **mixed predictor variable types**; that is, some numeric and some categorical input variables, with the numeric variables potentially having different scales (e.g., a person's age vs. their height). Prediction models can be built automatically from mixed data with no pre-processing and without specifying variable distributions (e.g., using decision tree ensemble learning (Breiman, 1996, 2001)); we seek a similarly automatic method for assessing extrapolation risk. While there are many methods available for density estimation, outlier detection, which could be leveraged to estimate extrapolation risk, most of them require data preprocessing or specifying variable distributions before they can be applied to mixed data. Notable exceptions are tree-based density estimators (Hooker, 2004; Schmidberger & Frank, 2005; Ram & Gray, 2011; Ting *et al.*, 2011) which can naturally handle the same mixed variable types as classification and regression trees; our work uses one of them as a building block in a forest density estimator. While the methods we consider could be extended to regression modeling, this paper only looks at detecting extrapolation risk for classification models.

In this paper we compare four approaches to estimating extrapolation risk for mixed data. The first two approaches are based on information in an ensemble of decision trees trained for classification, so we call them **builtin** risk detectors. The second two approaches construct an external, **auxiliary** model to explicitly model risk. Briefly, the four approaches are:

- **Margin Risk**: when the gap between the likelihood of the first and second place classes is small, the prediction is risky.
- **Forest Dispersion Risk**: an extension of forest proximity to find outliers (Breiman & Cutler, 2003) that uses the structure of the tree ensemble to measure proximity.
- **CERT Forest**: an ensemble of confidence and extrapolation risk trees.
- **Chaos Forest**: an ensemble of purely random decision trees.

Using synthetic and real data sets, we compare how well the four methods estimate extrapolation risk. Our main findings and contributions are:

- Extending forest proximity for extrapolation risk detection.

- Showing that pruning is not needed for CERT forests, but it *is* for chaos forests.
- Thoroughly testing forest-based outliers and CERT, resulting in a better understanding of how the algorithms behave.
- Demonstrating that neither builtin nor auxiliary approaches are a complete solution. Both strategies should be used to safeguard against extrapolation.

2 Extrapolation Risk Measures

We will shortly define extrapolation risk and present four methods for estimating this risk. We use the following notation. Data points are d-dimensional random variables $X = (X_1, X_2, \ldots, X_d)$ sampled from the feature space $\mathscr{X}$ according to some unknown distribution. Each X_i is either a real-valued variable or a categorical variable with a finite domain. An observation for X is denoted as $x = (x_1, x_2, \ldots, x_d)$. Let $\mathscr{Y} = \{y_1, y_2, \ldots, y_c\}$ denote the set of possible classes, and associate with each input random variable X an output variable $Y = h(X)$. In supervised classification learning, we are given a training set T of n input-output pairs

$$T = \left\{ (x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(n)}, y^{(n)}) \right\}$$

with the goal of estimating the unknown function h mapping from $\mathscr{X}$ to $\mathscr{Y}$. We assume that the learned classifier $\hat{h}$ is a probability estimator that predicts the probability of x belonging to each of the c classes; i.e., $\hat{h}(x) = (o_1, o_2, \ldots, o_c)$ with $o_i \in [0,1]$ and $\sum_i o_i = 1$.

2.1 Risk Definition

We adopt Hooker's (2004) definition for extrapolation risk. Intuitively, the risk of extrapolation is high for input x if it is more likely that x was sampled from the uniform distribution than from the training data distribution. Formally, the extrapolation risk is

$$R(x) = \frac{f_U(x)}{f_U(x) + f_T(x)} \tag{1}$$

where $f_U(x)$ and $f_T(x)$ are the data density at x assuming a uniform distribution and the training data distribution, respectively. $R(x)$ has range $[0,1]$, with 1 indicating maximum extrapolation risk.

In order for $f_U(\cdot)$ to be well-defined, all real-valued X_j must have bounded domains. Although the lower and upper bounds for numeric X_j are usually not known for sure, a domain expert can often provide ranges that will cover typical values. Alternatively, the range for X_j can be assumed to be the range observed in the training set.

When a point x' has a value that falls outside the range of the corresponding variable, we define $R(x') = 1$.

Directly computing (1) is problematic because the true density function for the training data, $f_T(\cdot)$, is unknown and must be estimated from data. We next consider four proxies for (1).

2.2 Builtin: Margin Risk

The first proxy for risk is margin risk. The **margin** for a prediction is the gap in the likelihood between the first and second place classes. Let o_i and o_j be the largest and second largest predicted

probabilities in $\hat{h}(x) = o$, respectively. Then the prediction margin for x is simply $o_i - o_j$. We define **margin risk** as

$$MR(x) = 1 - \text{margin}(x) \tag{2}$$

Because margin risk is derived from the classification model's output, it is a builtin measure of risk.

Margin risk can be computed for any classifier that predicts class membership probabilities. In this paper we use either bagged decision trees or random forests Breiman (1996, 2001) as the classification model because tree ensembles can handle mixed variable types. To predict class membership probabilities for an input, The ensemble averages the probability predictions from the trees in the ensemble.[1]

Admittedly, classifier confidence — and therefore, margin risk — seems like a poor substitute for $R(x)$ because the classifer can be 100% confident in its prediction and be 100% wrong if it is extrapolating. We include it in the comparison because classifier confidence is widely used in practice for filtering out unreliable predictions and because it is free to compute.

2.3 Builtin: Forest Dispersion Risk

The forest dispersion risk (FDR) score uses **forest proximity** Breiman & Cutler (2003) to measure the similarity between data points x and x'. The proximity of x and x' to each other in a tree ensemble is the fraction of trees in which they land in the same leaf:

$$\text{prox}(x, x') = \frac{1}{\#\text{ trees}} \sum_{t \in \text{Forest}} \text{I}\left[t.\text{leaf}(x) = t.\text{leaf}(x')\right]$$

Here, $\text{I}[\cdot]$ denotes the indicator function that returns 1 if the argument is true and 0 otherwise.

Breiman and Cutler (2003) used forest proximiy to derive an outlier score for finding outliers in training data as follows. The raw, unnormalized outlierness of point x is inversely proportional to its total proximity to other points with the *same class label*. Formally,

$$\text{rawOut}_c(x) = \left(\sum_{x' \in D_c - \{x\}} \text{prox}(x, x')^2 \right)^{-1}$$

where c is the class label for x and D_c is the subset of training data labeled as c. These raw scores are then normalized to be comparable across classes and always non-negative:

$$\text{outlier}_\text{c}(x) = \max\left(0, \frac{\text{rawOut}_c(x) - \mu_c}{\sigma_c}\right)$$

[1] Alternatively, the ensemble can use the vote distribution as the class membership probabilities. E.g., if 65 trees predict class y_1 (out of an ensemble containing 100 trees) as the most probable class, then the predicted probability o_1 is 0.65. This removes the requirement that the base classifier be able to predict probabilities. In preliminary experiments we found that this approach yielded coarser probabilities and was less sensitive for detecting risk.

with μ_c denoting the median raw scores for D_c and and σ_c similarly denoting the mean absolute deviation around μ_c.

We extend this idea to be able to measure the risk of a test point z in comparison to the training data. Note that no class label is available for z. We define **forest dispersion risk** (FDR) as the outlier score for the most similar class:

$$\text{FDR}(z) = \min_{c \in C} \text{outlier}_c(z) \tag{3}$$

The normalization statistics μ_c and σ_c are computed on the training data as before. Intuitively, $\text{FDR}(z)$ is high when z is widely dispersed in forest proximity space: its neighbors (those training data landing in the same leaf as z) vary from tree to tree.

Finally, it is not necessary to keep the training data available to compute (3). Instead we save the normalization statistics and a table recording which leaf each of the training points lands in. Entry (i, j) in the table stores the leaf ID for data point i in tree j.

2.4 Auxiliary: CERT Forest

The first auxiliary risk estimation method we study is building an ensemble of confidence and extrapolation risk trees (CERT, Hooker, 2004); each CERT model explicitly tries to predict $R(x)$. The resulting CERT forest is an auxiliary model that supplements the classification model with estimates of extrapolation risk.

CERT models are classification trees that are trained to discriminate between two classes: the foreground class and the background class. The **foreground class** consists of all observations in T; the class labels in T are ignored. The **background class** is the uniform distribution $f_U(\cdot)$. In this setting, the classifier is learning where the observed data are likely (the foreground class), and where no data are observed (the background class). A CERT model predicts the probability that a data point belongs to the background class; this probability is a direct estimate for $R(x)$. Thus, instead of estimating $f_T(\cdot)$ to be able to compute $R(x)$, we instead directly approximate $R(x)$ as the probability that x belongs to the background class.

The CERT learning algorithm proposed by Hooker is identical to standard classification tree learning except for one detail: the input for CERT learning only contains samples from the foreground class.

Hooker showed that sampling examples from the background class is problematic for feature spaces with more than a few dimensions. The problem is that sampling coverage is necessarily sparse in high dimensions, and this leads to high variance in the examples from the background class. Therefore, the CERT learning algorithm uses the known density function for the background class (uniform) instead of requiring a sample from the background class.

To compute the quality of a potential split, we need to know the number of data points from each class present before the split (i.e., in the parent node) and after the split (i.e., in each child

node). Armed with these quantities, we can compute the information gain from the split, or how much the Gini impurity has decreased, etc. The expected number of background points present at each node can be computed analytically. Each tree node corresponds to a subregion of feature space (the root node corresponds to the full feature space). The expected number of background points in a subregion r is

$$\text{count}(r) = \frac{\text{volume}(r)}{\text{volume}(\text{root})} \times \text{count}(\text{root})$$

where $\text{volume}(\cdot)$ is the hyper-volume of the given region, and $\text{count}(\text{root})$ is the number of background points at the root node. Usually, $\text{count}(\text{root})$ is set equal to the number of foreground data points in the training set.

2.5 Auxiliary: Chaos Forest

The last risk measure we consider is an ensemble of purely random decision trees (Fan *et al.*, 2003). We call this a **chaos forest** to emphasize that the model is constructed completely randomly and to distinguish it from Breiman's random forests (Breiman, 2001). We originally experimented with chaos forests to diagnose if CERT Forests were making random decisions. When they performed surprisingly well, we added them to set of risk measures being compared.

In a chaos forest, trees are grown randomly: the variable to split on at each node is chosen randomly, and if a threshold is needed (for numeric variable splits), it too is chosen randomly. Trees are grown until data runs out or until a maximum tree depth is reached. The ensemble of purely random trees makes predictions by averaging their predicted probabilities. Chaos forests, like random forests, often provide good estimates class memberhip probabilities (Fan *et al.*, 2005).

To estimate extrapolation risk with chaos forests, we construct the models to predict the probability of the foreground class vs the probability of the background class. This is identical to training CERT forests except that tree splits are chosen randomly.

2.6 Baseline: Box Risk

For comparison we include a simple baseline called BOX RISK. The BOX RISK detector predicts risk of 1 if the data point is outside of the expected bounding box and 0 otherwise. BOX RISK shows what performance can be achieved with a simple bounding box and lets us check if the risk measures are successful at detecting risk *within* the bounding box.

On the synthetic data problems (§ 4.1), we set the bounding box based to match the support of the known data distributions. For the real data sets (§ 4.2), the bounding box is set automatically from the range of values in the training data.

3 Implementation Details

The builtin risk measures (margin risk and forest dispersion risk) require first training classification tree ensembles. In our experiments we train ensembles of C4.5-style classification trees Quinlan (1993) with trees grown to full size. Trees are not pruned because larger trees generally produce more accurate ensembles Bauer & Kohavi (1999), particularly when the goal is to estimate class membership probabilities Provost & Domingos (2003); Chawla & Cieslak (2006). The predicted probability distribution from an ensemble is the average of the probabilities predicted by the trees in the ensemble. Probabilities are estimated at each tree leaf from the frequency of classes in the leaf, with Laplace smoothing applied to correct for small sample sizes Provost & Domingos (2003).

Our CERT forest implementation also grows full-size trees that predict class membership probabilities (background vs. foreground), but it uses information gain Quinlan (1986) as the split criterion and does not use Laplace smoothing for the probability estimation. We explored other split criteria but they did not make a significant difference. Similarly, preliminary experiments with Laplace smoothing suggested that the smoothing introduced a strong bias towards predicting probability 0.5.

When we first started testing single CERT models, we found that tree pruning was essential to getting reasonable risk estimates. Accordingly, we tested whether reduced error pruning Quinlan (1987) would improve the risk estimates from CERT forests. We found that pruning slightly improved risk estimates for 2-dimensional problems, and significantly hurt risk estimates for higher dimensional problems (Table 1). The rest of our experiments use unpruned CERT models in ensembles.

Table 1. Pruning is not needed for CERT forests. CERT forests, with and without tree pruning, were applied to twenty random synthetic problems with 2, 5, and 10 input variables (mixture of ridge distributions; see Section 4.1 for data details). On each problem, we measured the root mean squared error of the risk estimates compared to the true risk. The average RMSE (over twenty problems) is shown below. A * indicates statistically sig. lower error according to the Wilcoxon signed ranks test at 95% confidence level Demšar (2006).

# VAR	# TRAIN	PRUNED	UNPRUNED
2	1000	0.0944*	0.1094
5	4000	0.1916	0.1434*
10	4000	0.2436	0.1423*

For chaos forests, half of the training data in each bootstrap sample are used for constructing the tree structure. When a threshold needs to be chosen for a numeric variable, it is selected randomly from the set of midpoints between sorted values (for the variable in question). In addition to the midpoint splits, edge splits can also be selected. That is, lower and upper bound thresholds —

falling below and above all values in the current node — are included in the set of candidate splits. Allowing edge splits noticeably improved the performance of chaos forests on the synthetic data sets in the next section.[2]

Like CERT forests, the trees in chaos forests are grown to full-size. Unlike CERT forests, preliminary experiments showed that chaos forests were significantly more accurate if the trees were pruned. Specifically, the half of the bootstrap sample not used for tree construction is used to estimate the error of the tree's probability predictions according to the Brier score (Brier, 1950). Starting from the bottom of the tree, reduced error pruning deletes leaves when that would lower the Brier score.

[2]Interestingly, edge splits statistically significantly *hurt* the performance of CERT forests. The problem seems to be that a CERT tree can more easily overfit data when edge splits are allowed. This is not a problem in a chaos forest because an edge split is only selected at random (vs. to maximize information gain).

4 Comparing Builtin vs Auxiliary

In this section we compare how well the different risk measures detect extrapolation risk. First, we generate synthetic data where we know what the true extrapolation risk is; this gives us ground truth for evaluating the different methods. Second, on real data sets, we test whether the methods can detect unknown, novel classes that were not present during training.

4.1 Synthetic Data

We generate two types of synthetic problems: mixtures of multivariate Gaussians distributions, and mixtures of ridge distributions. For both types we vary the dimensionality d to see how results change with different numbers of input variables. For each of $d \in \{2,5,10\}$, we create twenty different random synthetic problems and average the results. Training sets contain 1000 data points ($d = 2$) or 4000 data points ($d = 5$ and $d = 10$) falling in a hyper-cube bounded by (-10,10) in all dimensions. Complete results for each of the generated problems can be found in Appendix A (Tables A.3-A.8).

The builtin risk detectors, $\text{MR}(x)$ and $\text{FDR}(x)$ are not natively calibrated to produce a probability of extrapolation. FDR uses scores on the range $[0,\infty)$, and while MR uses the range $[0,1]$, the scores are not well-calibrated. To be able to measure their quality using root mean squared error, we apply isotonic regression (Zadrozny & Elkan, 2002) to calibrate their outputs to be probabilities. *This calibration uses the true risk scores from the test set.* As a result, the numbers reported below for the builtin risk detectors represent a best possible scenario. Future work should evaluate quality with a ranking metric to remove this optimistic bias.

For the synthetic experiments, we use 100 rounds of bagging (Breiman, 1996) to create the ensembles for $\text{MR}(x)$, $\text{FDR}(x)$, CERT forests, and chaos forests.

Mixtures of Multivariate Gaussians

For the Gaussian mixture problems, we generate k mixture components per class. Thus, data points for class A can be generated from any of k different Gaussians (and likewise for points of type class B). Each multivariate Gaussian component has a mean sampled uniformly from the hyper-cube defined over $[-5,5]$ on all dimensions and a random covariance matrix Σ:

$$\Sigma = S^T S$$

where S is a d-by-d matrix with entries S_{ij} sampled from a univariate standard normal distribution. To generate a training data point, we randomly select one of the $2k$ components and draw a sample from the component's distribution. We set k to 2, 4, or 7 when d equals 2, 5, and 10, respectively.

Previously, CERT (Hooker, 2004) was evaluated by fitting a model to Gaussian mixture data and then tested with data sampled from a uniform distribution and with data sampled from the

Gaussian mixture. *In all but two dimensions, this test data is too easy.* With high probability, the test data from the uniform distribution will be far from all mixture components and have true risk close to 1. Similarly, test data from the mixture distribution will be near the center of the generating component and have true risk close to 0. Such test data does not let us validate how methods behave across the full risk spectrum.

We generate test data with a full range of true risk scores. For the two dimensional problems we exhaustively sample test data from a fine-grained grid across the entire input space (creating 13,000 test points on average). For higher dimensional problems we randomly pick a target true risk uniformly from $[0, 1]$ and then generate a test point with that risk. To find a point with the target risk, we generate a random uniform point and a random mixture point and search along the line between them for an interpolation point with the target risk. The line search can fail (e.g., if the risk of the mixture point is larger than the target risk), and in this case we try again (generating a new target risk and new end points for the line). We repeat this process until the test set contains 3,000 data points.

Table 2. Average root mean squared error for synthetic Gaussian mixture data.

TRIAL #	BOX RISK	MARGIN RISK	FDR	CERT	CERT FOREST	CHAOS FOREST
2D	0.6943	0.2244	0.2267	0.1026	0.1209	0.0968
5D	0.5801	0.2818	0.2826	0.3026	0.4538	0.3805
10D	0.5790	0.2873	0.2877	0.3505	0.5457	0.5208

Table 2 shows that all tested risk detection methods perform poorly on the Gaussian mixture data. The only exceptions are CERT, CERT FOREST, and CHAOS FOREST for two dimensional problems. The problem is that characterizing the boundaries of a Gaussian distribution requires jointly examining all of the dimensions. Meanwhile, the decision trees myopically test a single dimension at a time. In two dimensions, ensembles of trees do a reasonably good job of defining the mixture distributions, but struggle when dimensionality increases to five. Because all the risk detectors involve decision trees, they all suffer this problem for 5- and 10-dimensional problems. Consequently, the Gaussian mixture data sets do not illuminate the different strengths and weaknesses of the algorithms: they are all bad in this setting.

Mixtures of Ridge Distributions

The second set of synthetic problems is designed to create problems where decision trees can reasonably approximate the true function, but are also hard enough to show demonstrate differences between the risk detection algorithms. Our guiding design principle is to first create problems where testing an input variable in isolation reveals significant information and then mix in some statistical interactions between variables.

The mixtures of ridge distributions are defined as follows. A **ridge distribution** is the product of a one-dimensional Gaussian distribution in the ridge dimension i and uniform distributions for all other dimensions. For two dimensional spaces, this turns the Gaussian density function into a ridge-like surface. Formally, the ridge density function is:

$$f_{\text{ridge}(i)}(x) = g(x_i) \prod_{j \neq i} f_U(x_j) \tag{4}$$

where $g(x_i)$ is a univariate Gaussian density function and $f_U(x_j)$ is the uniform density defined over the domain of variabe X_j. If all variables are numeric, with lower and upper bounds l_j and u_j respectively,

$$f_{\text{ridge}(i)}(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}} \prod_{j \neq i} \frac{1}{u_j - l_j} \tag{5}$$

A **mixture of ridge distributions** is defined by adding together multiple component ridge distributions. For our experiments each ridge component has equal probability in the mixture.

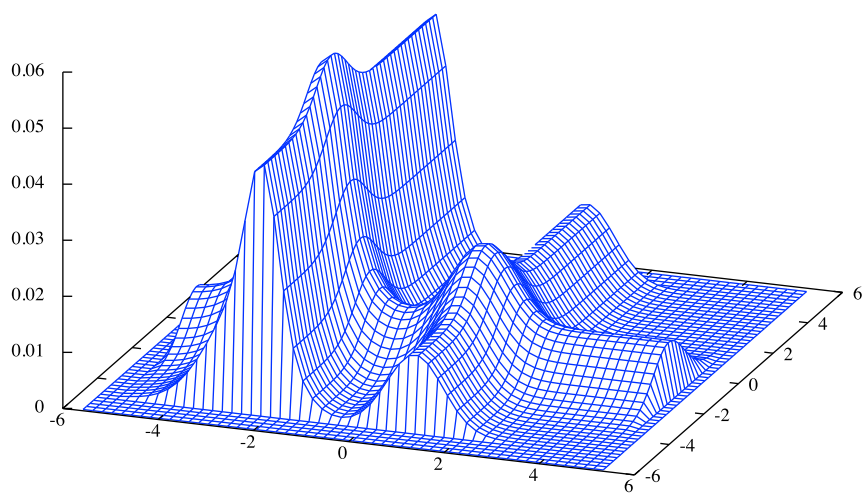

Figure 1. Example 2D mixture of ridge densities.

Figure 1 shows the density surface for an example mixture of ridge distributions defined over a two variable input space. This distribution is formed from four ridge components. (Two ridges have nearly identical means, which creates the taller ridge.) By construction, each ridge runs orthogonally to one of the input dimensions. Consequently, the density function can be well-characterized by recursively subdividing the input space via thresholds on single variables. As it turns out, such simple distributions are too easy for our purposes because they are *too* perfect for decision trees.

To make the problems more challenging we rotate the basis space of the mixture distribution so that some of the ridges are no longer orthogonal to variable axes. Specifically, the rotation is

formed by a sequence of planar rotations over pairs of input variables. The plane defined by two randomly selected dimensions is rotated around the origin via a Givens rotation. Any ridges in the plane remain orthogonal to each other but are diagonal to the input variable axes. Because the rotations are orthogonal, distances within the plane are preserved.

For our experiments we generate ridge mixture models with k components and r planar rotations. For $d = 2, 5$, or 10, k is set to 5, 8, and 14, respectively, and r is set to 1, 2, and 3.[3] The angle of rotation is chosen randomly, with all angles equally likely. Each ridge component is randomly labeled as class A or B.

To generate a training data point, we randomly select one of the k ridge components and draw a sample from its distribution in the non-rotated basis space and then rotate the data point into the appropriate basis space. Test data are generated in the same way as in the synthetic mixture of Gaussians experiment.

Table 3. Average root mean squared error for synthetic ridge mixture data.

TRIAL #	BOX RISK	MARGIN RISK	FDR	CERT	CERT FOREST	CHAOS FOREST
2D	0.6231	0.2293	0.2208	0.1074	0.1275	0.1073
5D	0.5856	0.1860	0.1838	0.1577	0.2678	0.1434
10D	0.5565	0.1512	0.1492	0.1375	0.3094	0.1423

There are clear differences between the risk detection algorithms for this family of synthetic problems (Table 3). The risk scores from both CERT FOREST and CHAOS FOREST have low root mean squared error for all dimensionalities tested. Both forest methods outperform CERT, and dramatically so for five and ten dimensional problems. Somewhat surprisingly, chaos forests perform virtually as well as CERT forests, despite the fact that its trees are grown randomly. The two methods are statistically tied for the 2- and 10-dimensional problems; for the 5-dimensional problems, the CERT forests have statistically smaller error (although the difference is small).[4] Comparing all algorithms across all 60 synthetic ridge problems, we find that the two forest methods are statistically significantly better than the others, and that CERT, MARGIN RISK, and FDR are tied with each other but better than the baseline BOX RISK.[5]

A surprising result in Table 3 is that the error for the builtin risk methods decreases as dimensionality increases. Generally, data analysis becomes harder as dimensionality increases. We do not currently have an explanation for why MARGIN RISK and FDR appear to improve. One concern

[3]We experimented with larger numbers of rotations, and we found that problem difficulty increases quickly with the number of rotations. Intuitively, if variable X_1 participates in two planar rotations (say with variables X_5 and X_8), then there is a three-term statistical interaction between X_1, X_5, and X_8. Standard decision tree learning has trouble detecting interactions between more than two numeric variables.

[4]Statistical significance tested with Wilcoxon signed ranks test at $\alpha = 0.05$ level (Demšar, 2006).

[5]Friedman test rejects null hypothesis for $\alpha = 0.05$, and Nemenyi post-hoc test shows critical difference between groups (Demšar, 2006).

we have is that these methods both are benefitting from score calibration that adjusts their output to better fit the true risk scores on the test data. This is an unfair advantage over the other methods, and it is possible that calibration is obscuring the true quality of MARGIN RISK and FDR.

4.2 Detecting an Unknown Class

While synthetic data has the advantage of knowing the true extrapolation risk, the conclusions we draw from it are only valid to the extent the synthetic problems are good proxies for real data problems. This section tests whether the risk detection algorithms are effective on real data.

One scenario in which a prediction model risks extrapolation is when the test data contain an unknown class not seen during model training. This can happen because the training sample has incomplete coverage (perhaps a rare class is overlooked or the data collection process is biased) or because the world changes and a novel class is introduced in the future. Assuming the unknown class lies in a distinct region of input space, data points from the unknown class will have high extrapolation risk.

We simulate this scenario with the following experiment. One class is designated as the unknown class. We train the risk detection measures using training data with all examples of the unknown class removed. The test set contains the unknown class along with all of the other classes. We then evaluate the risk detectors on whether they correctly rank examples from the unknown class as higher risk than examples from the other classes.

We ran the unknown class experiment on eight public data sets of varying sizes, dimensionality, and mixes of numeric and categorical variables (Table 4). The unknown class was either chosen because it seemed fundamentally different from the others (based on our human intuition) or because it was an infrequent class with at least a few dozen examples.

All data sets except nyt-topics are available from the UCI Machine Learning repository and are documented there. The nyt-topics data set contains New York Times articles published in 2003 (Sandhaus, 2008), using the section labels provided as topic labels. We used the topics *NY Region*, *Paid Death Notices*, and *Real Estate* (48.9%, 48.6%, and 2.4% of our data set, respectively). Based on prior research (Stevens *et al.*, 2012), we extracted 9 features using latent semantic analysis.

On each data set we repeated 2-fold cross-validation five times and measured the average area under the ROC curve (AUC). The AUC measures ranking quality, and it equals the probability that the model ranks a random positive example above a random negative example (Fawcett, 2006). An AUC of 1 is perfect, and 0.5 equals random guessing. For purposes of computing AUC, the unknown class was designated the positive class, and all other classes were lumped together as a single negative class.

For this experiment, we used random forest learning (Breiman, 2001) with 250 iterations to generate all ensembles. For each tree node, $\lceil \log_2 d \rceil + 1$ variables were randomly selected (from the d available inputs) and considered as candidates for the tree split. (Chaos forests were able to

Table 4. Data sets used in unknown class experiment.

NAME	# DATA	# VAR.†	DESCRIPTION
annealing‡	898	38 (6,32)	Classify objects into 6 possible classes. Class 2 withheld during training (11.0% of data set).
arrhythmia‡	420	260 (197,63)	Determine type of arrhythmia from ECG recordings. Type 10 (out of 13 in data; 11.4% of data) not seen during training.
dermatology‡	358	34 (33,1)	Differentiate between 6 types of skin disease. Type 5 withheld during training (13.4% of data).
glass‡	214	9 (9,0)	Predict type of glass from chemical properties. Types 5-7 combined into *NonWindow* class and were omitted during training (23.8% of data).
isolet‡	7797	617 (617,0)	Predict which letter-name was spoken. Vowels and the letter Y were omitted during training (23.1% of data).
nyt-topics	22926	9 (9,0)	Classify New York Times articles into 3 topics. *Real Estate* topic not seen during training (2.4% of data).
page-blocks‡	5473	10 (10,0)	Decide if block of a document page is *text*, an *image*, or a *vertical* or *horizontal* line. *Image* class withheld during training (2.6% of data).
splice-junction‡	3190	60 (0,60)	Decide if window of 60 DNA sequence elements contains a splice junction (EI or IE site, or neither). *Neither* class omitted during training (51.9% of data).

† Number of numeric and categorical variables listed in parentheses.
‡ Downloaded from UCI Machine Learning Repository (Frank & Asuncion, 2010).

randomly select any of the d variables.)

To determine the best possible detection rate, we also evaluated how well the unknown class could be detected if it were known. Specifically, we trained a random forest classifier *with training data that contained the unknown class* and then ranked the test data by the predicted probability of belonging to the unknown class. This upper bound is listed as CHEAT in the results.

The risk detectors are generally able to detect the unknown class, although the best detector varies from problem to problem (Table 5). Overall there is no clear best method,[6] although the two forest methods are more often than not the top performers.

[6]The Friedman test with $\alpha = 0.05$ accepted the null hypothesis that margin risk, FDR, CERT, CERT Forest, and chaos forest were tied.

Table 5. AUC for detecting class not seen during training.

DATA SET	BOX RISK	MARGIN RISK	FDR	CERT	CERT FOREST	CHAOS FOREST	CHEAT
anneal	0.6487	0.6970	0.6109	0.5450	0.6719	0.7275	0.9996
arrhythmia	0.6296	0.6235	0.5931	0.6350	0.7019	0.6808	0.9339
dermatology	0.8312	0.9687	0.9725	0.8602	0.8590	0.8693	1.0000
glass	0.7984	0.7415	0.7868	0.8208	0.8837	0.8811	0.9781
isolet	0.5604	0.7737	0.8314	0.5268	0.5104	0.5880	0.9949
nyt-topics	0.5566	0.3053	0.4480	0.7817	0.9073	0.9021	0.9877
page-blocks	0.6114	0.9057	0.7654	0.7206	0.8253	0.8078	0.9885
splice-junction	0.4995	0.9210	0.6626	0.9443	0.9926	0.9762	0.9945

Interestingly, the builtin risk measures (MARGIN RISK and FDR) are sometimes worse than the BOX RISK straw man (anneal, arrhythmia, glass, nyt-topics). In other words, it would be better to use a simple bounding box around the training data than to rely on classifier confidence (margin risk) or forest dispersion risk. This is particularly pronounced for the nyt-topics data set where they are worse than random guessing (0.5 AUC) — it would be better invert their risk rankings! We hypothesize that the builtin risk measures primarily assign high risk to regions where known classes overlap. For some data sets (like nyt-topics) this causes them to rank regions of high data density as risky; usually, high data density regions ought to have low extrapolation risk. But there are also clearly data sets where one or both of the builtin risk measures do a better job of finding the unknown class (dermatology, isolet, page-blocks).

The most important conclusion from Table 5 is that both builtin and auxiliary risk measures are useful. Future research should investigate how to combine their complementary strengths.

5 Discussion

We empirically compared four algorithms for estimating the extrapolation risk of a machine learning classifier. If one were to choose a single method to use, either CERT forests or chaos forests would be the best choice. These auxiliary risk detectors performed best on the synthetic data problems, and on average were more reliable than the builtin risk detectors in the unknown class experiment. Whereas both margin risk and FDR occassionally performed worse than the box risk baseline, the two forest were always better than the baseline and often the best at ranking the unknown class as high risk. Compared to distance based or density based approaches for finding outliers and detecting extrapolation risk, these methods are easier for users to apply because they work directly with both numerical and categorical variables and do not require the user to normalize or scale data to get reasonable results.

However, our recommendation is to combine a builtin detector with an auxiliary detector. The two strategies performed well on different data sets, and the safest way to guard against potential extrapolation risk would be combining their strengths. More research is needed to better characterize when margin risk and FDR can be trusted. One possibility is that they are actually bad at detecting extrapolation risk, but for identifying test points that are ambiguous because they are in regions of class overlap.

While it was expected that the CERT forest would outperform a singe CERT model, it was surprising that chaos forests performed so well. An interesting open question is to understand how averaging random trees can perform so well. We have two thoughts in this regard. First, it might be possible to make a connection between chaos forests and using k-nearest neighbor for density estimation (an established non-parametric approach to density estimation). Lin and Jeon 2006 have already shown that random forests are a type of adaptive nearest neighbor algorithm. Extending their result to chaos forests could help explain their success.

A second possible reason behind the success of chaos forests is a weakness we have sometimes observed with CERT forests. In certain conditions a CERT tree will stop growing before it has fully characterized (bounded) a local cluster of data (Figure 2). Because chaos forests install tree splits randomly, it is possible that they are more robust to the problem of running out of data early that causes premature stopping for CERT models.

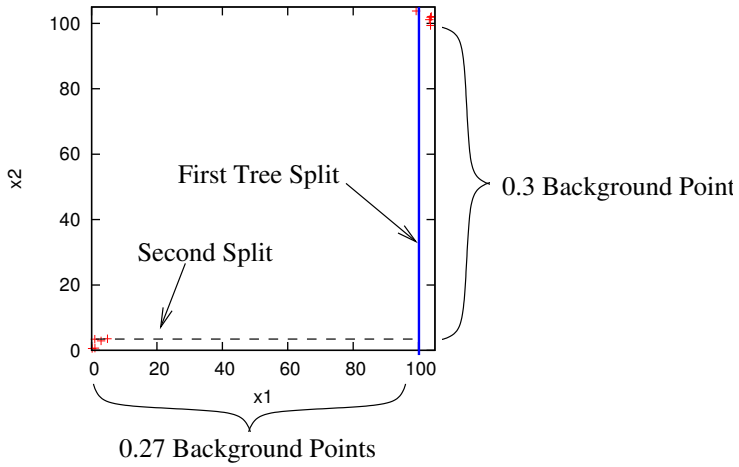

Figure 2. The CERT learning algorithm can stop tree growing prematurely if splits create very small regions of feature space. In this two dimensional example, there are 10 foreground data points (red plus signs) contrasted with a uniform background distribution of equal size. The first learned tree split creates a narrow vertical region containing 4 foreground points and 0.3 background points (in expectation), with the remaining 6 foreground and 9.7 background points on the left of the line. The vertical region is now pure, and it is not split further. The estimated extrapolation risk for the region is $\frac{0.3}{4+0.3} \approx 0.07$. Similarly, the second split creates a narrow horizontal region containing 4 foreground and 0.27 background points with estimated risk of 0.06. *Predictions for data near location (100, 0) will therefore be considered safe (low extrapolation risk) despite that location being far from all foreground data.*

References

Bauer, E. & Kohavi, R. (1999) An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning*, **36**, 105–139.

Breiman, L. (1996) Bagging predictors. *Machine Learning*, **24**, 123–140.

Breiman, L. (2001) Random forests. *Machine learning*, **45**, 5–32.

Breiman, L. & Cutler, A. (2003) *Manual–Setting Up, Using, and Understanding Random Forests v4.0*. UC Berkeley. Online at `http://oz.berkeley.edu/users/breiman/Using_random_forests_v4.0.pdf`.

Brier, G.W. (1950) Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, **78**, 1–3.

Chawla, N.V. & Cieslak, D.A. (2006) Evaluating probability estimates from decision trees. In *AAAI Workshop on the Evaluation Methods in Machine Learning*, pp. 18–23.

Demšar, J. (2006) Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, **7**, 1–30.

Fan, W., Greengrass, E., McCloskey, J., Yu, P.S. & Drummey, K. (2005) Effective estimation of posterior probabilities: Explaining the accuracy of randomized decision tree approaches. In *ICDM'05*.

Fan, W., Wang, H., Yu, P.S. & Ma, S. (2003) Is random model better? on its accuracy and efficiency. pp. 51–58.

Fawcett, T. (2006) An introduction to ROC analysis. *Pattern Recognition Letters*, **27**, 861–874.

Frank, A. & Asuncion, A. (2010) UCI machine learning repository. University of California, Irvine, School of Information and Computer Sciences, URL `http://archive.ics.uci.edu/ml`.

Hooker, G. (2004) Diagnosing extrapolation: Tree-based density estimation. In *KDD'04*, pp. 569–574.

Lin, Y. & Jeon, Y. (2006) Random forests and adaptive nearest neighbors. *Journal of the American Statistical Association*, **101**, 578–590.

Provost, F. & Domingos, P. (2003) Tree induction for probability-based ranking. *Machine Learning*, **52**, 199–215.

Quinlan, J.R. (1986) Induction of decision trees. *Machine Learning*, **1**, 81–106.

Quinlan, J.R. (1987) Simplifying decision trees. *Intl. Journal of Man-Machine Studies*, **27**, 221–234.

Quinlan, J.R. (1993) *C4.5: Programs for Machine Learning*. Morgan Kaufmann.

Ram, P. & Gray, A.G. (2011) Density estimation trees. In *KDD'11*, pp. 627–635.

Sandhaus, E. (2008) The New York Times Annotated Corpus. Linguistic Data Consortium, Philadelphia.

Schmidberger, G. & Frank, E. (2005) Unsupervised discretization using tree-based density estimation. In *PKDD'05*, pp. 240–251.

Stevens, K., Kegelmeyer, P., Andrzejewski, D. & Buttler, D. (2012) Exploring topic coherence over many models and many topics. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics, Stroudsburg, PA, USA.

Ting, K.M., Washio, T., Wells, J.R. & Liu, F.T. (2011) Density estimation based on mass. In *ICDM'11*, pp. 715–724.

Zadrozny, B. & Elkan, C. (2002) Transforming classifier scores into accurate multiclass probability estimates. In *KDD'02*, pp. 694–699.

A Extra Information about Extrapolation Risk Experiments

A place to document interesting ideas we tried that did not work. Probably not in conference version, but useful for SAND report.

- edge splits
- fractal splits, combined with statistical test to decide if split worth attempting

A.1 Edge Splits

Table A.1. Edge splits hurt (surprising). Details: synthetic data (product model with planar rotations; see section ...). Unpruned CERT forests. * indicates statistically sig. better according to Wilcoxon signed ranks test, $\alpha = 0.05$.

	EDGE SPLITS?	
# ATTR	YES	NO
2	0.2164	0.1094
5	0.2740	0.1434
10	0.2469	0.1423

A.2 Fractal Splitting

Table A.2. Fractal splits hurt (not that surprising). Details: synthetic data (product model with planar rotations; see section ...). Unpruned CERT forests. * indicates statistically sig. better according to Wilcoxon signed ranks test, $\alpha = 0.05$.

	EDGE SPLITS?	
# ATTR	YES	NO
2	0.1398	0.1094
5	0.2351	0.1434
10	0.2252	0.1423

A.3 Complete Results for Synthetic Data Experiments

Table A.3. Results for 2D synthetic gaussian mixture data.

TRIAL #	BOX RISK	MARGIN RISK	FDR	CERT	CERT FOREST	CHAOS FOREST
1	0.6649	0.2154	0.2281	0.1007	0.1278	0.1045
2	0.6945	0.2163	0.2102	0.0887	0.1075	0.0862
3	0.7288	0.2207	0.2225	0.1165	0.1426	0.1097
4	0.6436	0.2428	0.2446	0.1003	0.1170	0.1071
5	0.6818	0.2180	0.2171	0.1067	0.1174	0.0971
6	0.6989	0.2427	0.2431	0.1005	0.1201	0.0939
7	0.6823	0.2362	0.2376	0.0962	0.1085	0.0916
8	0.7442	0.2178	0.2172	0.0994	0.1270	0.0883
9	0.7302	0.2198	0.2196	0.1096	0.1406	0.1112
10	0.6633	0.2121	0.2064	0.1038	0.1040	0.0928
11	0.6736	0.2181	0.2172	0.0914	0.1019	0.0812
12	0.7353	0.2317	0.2353	0.1230	0.1463	0.1147
13	0.6335	0.2239	0.2246	0.1078	0.1149	0.1015
14	0.6970	0.2079	0.2210	0.1028	0.1148	0.0891
15	0.7166	0.2467	0.2484	0.1030	0.1131	0.0885
16	0.7463	0.2242	0.2317	0.0994	0.1308	0.0853
17	0.7311	0.1966	0.2078	0.1085	0.1499	0.1082
18	0.7167	0.2395	0.2400	0.1079	0.1183	0.0902
19	0.6734	0.2323	0.2352	0.0927	0.1091	0.0956
20	0.6305	0.2262	0.2263	0.0933	0.1066	0.0989
AVG	0.6943	0.2244	0.2267	0.1026	0.1209	0.0968

Table A.4. Results for 5D synthetic gaussian mixture data.

TRIAL #	BOX RISK	MARGIN RISK	FDR	CERT	CERT FOREST	CHAOS FOREST
1	0.5821	0.2807	0.2812	0.2928	0.4480	0.3664
2	0.5781	0.2817	0.2828	0.2990	0.4578	0.3782
3	0.5761	0.2853	0.2860	0.3265	0.4730	0.4118
4	0.5862	0.2851	0.2853	0.3120	0.4683	0.4001
5	0.5778	0.2845	0.2849	0.3164	0.4615	0.3951
6	0.5805	0.2765	0.2785	0.2979	0.4597	0.3823
7	0.5815	0.2866	0.2877	0.3076	0.4588	0.3954
8	0.5759	0.2841	0.2846	0.2850	0.4266	0.3467
9	0.5786	0.2790	0.2802	0.2977	0.4456	0.3711
10	0.5742	0.2827	0.2823	0.2960	0.4483	0.3779
11	0.5706	0.2804	0.2818	0.2960	0.4402	0.3622
12	0.5843	0.2807	0.2816	0.2934	0.4289	0.3652
13	0.5747	0.2800	0.2814	0.2953	0.4402	0.3628
14	0.5831	0.2800	0.2809	0.2702	0.4395	0.3536
15	0.5842	0.2822	0.2828	0.2932	0.4572	0.3728
16	0.5832	0.2817	0.2839	0.3050	0.4568	0.3733
17	0.5835	0.2765	0.2780	0.2953	0.4497	0.3639
18	0.5850	0.2853	0.2859	0.3009	0.4536	0.3867
19	0.5893	0.2828	0.2838	0.3558	0.4949	0.4409
20	0.5729	0.2794	0.2793	0.3163	0.4667	0.4041
AVG	0.5801	0.2818	0.2826	0.3026	0.4538	0.3805

Table A.5. Results for 10D synthetic gaussian mixture data.

TRIAL #	BOX RISK	MARGIN RISK	FDR	CERT	CERT FOREST	CHAOS FOREST
1	0.5788	0.2863	0.2867	0.3578	0.5491	0.5286
2	0.5780	0.2897	0.2901	0.3561	0.5447	0.5192
3	0.5759	0.2890	0.2891	0.3513	0.5448	0.5217
4	0.5757	0.2885	0.2889	0.3437	0.5388	0.5198
5	0.5850	0.2891	0.2894	0.3628	0.5546	0.5327
6	0.5829	0.2857	0.2859	0.3539	0.5470	0.5212
7	0.5841	0.2827	0.2829	0.3552	0.5514	0.5279
8	0.5706	0.2856	0.2865	0.3435	0.5401	0.5153
9	0.5836	0.2879	0.2890	0.3454	0.5491	0.5185
10	0.5751	0.2863	0.2867	0.3476	0.5421	0.5158
11	0.5696	0.2892	0.2900	0.3456	0.5370	0.5142
12	0.5833	0.2884	0.2887	0.3630	0.5563	0.5306
13	0.5793	0.2866	0.2870	0.3360	0.5394	0.5116
14	0.5840	0.2892	0.2900	0.3498	0.5471	0.5190
15	0.5819	0.2832	0.2834	0.3371	0.5448	0.5133
16	0.5826	0.2894	0.2895	0.3629	0.5498	0.5267
17	0.5777	0.2862	0.2867	0.3523	0.5457	0.5233
18	0.5733	0.2878	0.2880	0.3446	0.5336	0.5093
19	0.5802	0.2879	0.2886	0.3470	0.5487	0.5224
20	0.5785	0.2870	0.2871	0.3545	0.5505	0.5248
AVG	0.5790	0.2873	0.2877	0.3505	0.5457	0.5208

Table A.6. Results for 2D synthetic ridge mixture data.

TRIAL #	BOX RISK	MARGIN RISK	FDR	CERT	CERT FOREST	CHAOS FOREST
1	0.6331	0.2504	0.2535	0.1009	0.1056	0.1098
2	0.6006	0.2176	0.2091	0.0988	0.1354	0.1017
3	0.6623	0.2432	0.2344	0.1087	0.1476	0.1035
4	0.6381	0.2452	0.2312	0.0964	0.1164	0.1042
5	0.5991	0.2171	0.2033	0.1133	0.1266	0.1128
6	0.5642	0.1992	0.1906	0.1143	0.1236	0.1156
7	0.6419	0.2396	0.2257	0.1119	0.1437	0.1087
8	0.6524	0.2382	0.2416	0.1128	0.1483	0.1079
9	0.6242	0.2351	0.2360	0.1209	0.1528	0.1121
10	0.5310	0.1761	0.1729	0.1197	0.1040	0.1208
11	0.6742	0.2325	0.2311	0.1052	0.1406	0.0975
12	0.6118	0.2474	0.2218	0.1075	0.1211	0.1142
13	0.6284	0.2334	0.2181	0.1040	0.1309	0.1054
14	0.6549	0.2496	0.2250	0.1117	0.1435	0.1054
15	0.6479	0.2226	0.2363	0.1021	0.1280	0.0984
16	0.6045	0.2246	0.2130	0.1058	0.0740	0.1114
17	0.5853	0.2239	0.2228	0.1104	0.0913	0.1127
18	0.6498	0.2285	0.1979	0.0901	0.1376	0.0923
19	0.6311	0.2173	0.2136	0.0873	0.1245	0.0896
20	0.6269	0.2435	0.2390	0.1264	0.1552	0.1217
AVG	0.6231	0.2293	0.2208	0.1074	0.1275	0.1073

Table A.7. Results for 5D synthetic ridge mixture data.

TRIAL #	BOX RISK	MARGIN RISK	FDR	CERT	CERT FOREST	CHAOS FOREST
1	0.6100	0.1980	0.1980	0.1793	0.3117	0.1601
2	0.5898	0.1693	0.1678	0.1264	0.2553	0.1280
3	0.5916	0.1906	0.1904	0.1480	0.2399	0.1299
4	0.6079	0.1915	0.1842	0.1673	0.3120	0.1427
5	0.5755	0.1770	0.1736	0.1466	0.2749	0.1352
6	0.5587	0.1695	0.1679	0.1470	0.2786	0.1339
7	0.6060	0.1967	0.1952	0.1660	0.2653	0.1416
8	0.6005	0.1932	0.1902	0.1916	0.3402	0.1699
9	0.5554	0.1823	0.1822	0.1429	0.1728	0.1589
10	0.5577	0.1753	0.1699	0.1596	0.2953	0.1511
11	0.5934	0.1876	0.1877	0.1543	0.2427	0.1148
12	0.5817	0.1648	0.1640	0.1634	0.3341	0.1448
13	0.5905	0.2046	0.2032	0.1900	0.3393	0.1818
14	0.6099	0.1912	0.1877	0.1946	0.3316	0.1500
15	0.6124	0.2008	0.2012	0.1521	0.2321	0.1372
16	0.5813	0.1872	0.1859	0.1453	0.2458	0.1322
17	0.5778	0.1935	0.1818	0.1686	0.2608	0.1310
18	0.5703	0.1750	0.1750	0.1359	0.2321	0.1357
19	0.5579	0.1792	0.1791	0.1188	0.1940	0.1262
20	0.5830	0.1922	0.1919	0.1558	0.1966	0.1621
AVG	0.5856	0.1860	0.1838	0.1577	0.2678	0.1434

Table A.8. Results for 10D synthetic ridge mixture data.

TRIAL #	BOX RISK	MARGIN RISK	FDR	CERT	CERT FOREST	CHAOS FOREST
1	0.5803	0.1714	0.1714	0.1527	0.3070	0.1475
2	0.5760	0.1514	0.1465	0.1331	0.3365	0.1277
3	0.5556	0.1558	0.1557	0.1484	0.3313	0.1440
4	0.5696	0.1511	0.1451	0.1309	0.3485	0.1564
5	0.5442	0.1307	0.1270	0.1218	0.2439	0.1242
6	0.5341	0.1409	0.1397	0.1333	0.3551	0.1758
7	0.5363	0.1325	0.1299	0.1186	0.3498	0.1615
8	0.5558	0.1502	0.1491	0.1346	0.2696	0.1247
9	0.5609	0.1609	0.1607	0.1443	0.3221	0.1484
10	0.5527	0.1500	0.1486	0.1435	0.3574	0.1732
11	0.5572	0.1554	0.1511	0.1522	0.2540	0.1477
12	0.5543	0.1416	0.1386	0.1290	0.3520	0.1525
13	0.5657	0.1791	0.1779	0.1566	0.2973	0.1385
14	0.5644	0.1590	0.1571	0.1437	0.3236	0.1412
15	0.5690	0.1653	0.1659	0.1564	0.2285	0.1455
16	0.5649	0.1601	0.1597	0.1457	0.2988	0.1335
17	0.5405	0.1444	0.1446	0.1318	0.2872	0.1259
18	0.5423	0.1294	0.1258	0.1087	0.3101	0.1221
19	0.5543	0.1445	0.1427	0.1285	0.2986	0.1235
20	0.5524	0.1501	0.1464	0.1366	0.3164	0.1328
AVG	0.5565	0.1512	0.1492	0.1375	0.3094	0.1423

DISTRIBUTION:

1	MS 0899	Technical Library, 8944 (electronic copy)
1	MS 0359	D. Chavez, LDRD Office, 1911

Sandia National Laboratories