Journal of Machine Learning Research 20 (2020) 1-31

Stable Classification

Dimitris Bertsimas DBERTSIM@MIT.EDU
Sloan School of Management and Operations Research Center
Massachusetts Institute of Technology
Cambridge, MA 02139, USA

Jack Dunn JACK@INTERPRETABLE.AI
Interpretable AI
Cambridge, MA 02139, USA

Ivan Paskov IPASKOV@MIT.EDU
Operations Research Center
Massachusetts Institute of Technology
Cambridge, MA 02139, USA

Editor:

Abstract

We address the problem of instability of classification models: small changes in the training data leading to large changes in the resulting model and predictions. This phenomenon is especially well established for single tree based methods such as CART, however it is present in all classification methods. We apply robust optimization to improve the stability of five of the most commonly used classification methods: Random Forests, Logistic Regression, Support Vector Machines, CART, and Optimal Classification Trees. Through experiments on 30 datasets with sizes ranging between 10^2 and 10^4 observations and features, we show that our approach (a) leads to improvements in both performance and stability compared to the original methods, with the gains in stability being particularly significant (even, surprisingly, for those methods that were previously thought to be stable, such as Random Forests) and (b) has computational times comparable with (and indeed in some cases even faster than) the original methods allowing the method to be very scalable.

Keywords: Stability, Optimal Decision Trees, Robustness, Interpretability, Logistic Regression, Support Vector Machines, CART, Classification

1. Introduction

We address the problem of instability of classification models: small changes in the training data leading to large changes in the resulting model and predictions. Such instability arises due to two primary sources: (a) Training Instability: variability arising due to the

choice of training/validation split, and (b) Temporal Instability: variability arising due to receiving new data over time. Decision tree based methods such as CART are well known to exhibit both such forms of instability and high variance. Indeed, it was this very issue that motivated Breiman (1996a) to develop Bagging and Breiman (2001) to further refine Bagging with Random Forests, which are explicitly designed to reduce such instability via averaging. While certainly more stable than CART, the cost of increasing stability was high: Random Forests is by and large uninterpretable, and Breiman (1996b) asks"whether there is a more stable single-tree version of CART."

In this paper, we answer this question in the affirmative. Moreover, despite Random Forests being more stable with respect to the choice of training/validation split, it still suffers from temporal instability, and in general it is still an open question whether its overall stability can be improved. This too we answer in the affirmative in this paper. More precisely, we generalize the robust optimization based approach for constructing stable linear regression models, developed in Bertsimas and Paskov (2020), to general classification methods. Specifically, we develop a methodology for building classification models that are robust to the specific dataset that was used to build them. We apply this approach to five popular classification methods: Random Forests (RF), Logistic Regression (LR), Support Vector Machines (SVM), CART, and Optimal Classification Trees (OCT). Through experiments on 30 datasets with sizes ranging between 10^2 and 10^4 observations and features, we show that our approach (a) leads to improvements in both performance and stability compared to the original methods, with the gains in stability being particularly significant (even, surprisingly, for those methods that were previously thought to be stable, such as Random Forests) and (b) has computational times comparable with the original methods allowing the method to be very scalable.

Literature

The idea of using optimization (over randomization) to build regression models that are robust to the subsample of data they are trained upon, was first developed in Bertsimas and Paskov (2020) building on the theme of using optimization versus optimization in machine learning models, see Chapters 15-18 in Bertsimas and Dunn (2019). We extend these ideas to classification problems: RF, LR, SVM, CART and OCT introduced in Breiman (2001), Cox (1966), Vapnik and Lerner (1963), Breiman et al. (1984) and Bertsimas and Dunn (2017), respectively. Other attempts at producing stable, tree based methods can be found via the approaches of bagging developed by Breiman (1996a), boosting developed by Freund and Schapire (1995) and Random Forests, developed by Breiman (2001). All three of these methods work by combining multiple models to produce more accurate and stable trees. While more stable than CART, these methods are by and large uninterpretable. Breiman (1996b) proposed averaging as means of stabilizing any general method, albeit at the cost of interpretability. Still another approach, developed by Last et al. (2002) attempts to use statistical significance testing and pruning to produce stable trees. While more stable than CART, their approach unfortunately suffers from poor accuracy. Thus, none of the competing approaches have succeeded in being at once interpretable, accurate, and stable.

Finally, to the best of our knowledge, no work exists attempting to stabilize RF, SVM, or LR, likely because these methods are already widely believed to be stable. Indeed RF was explicitly designed to further stabilize the bagging procedure by averaging uncorrelated trees, see Breiman (2001) for more detail, and SVM are considered so stable as to actually be usable in live image stabilization, as detailed in Dong et al. (2011).

Contributions and Structure

In this paper, we extend the approach of Bertsimas and Paskov (2020) to general classification problems. We develop a robust optimization framework for stabilizing any classification method, and apply it to RF, LR, SVM, CART and OCT. We present three approaches: Robust Counterpart, Cutting Planes and Monte Carlo. Through experiments on 30 datasets, we show that the stable methods improve both in performance and stability compared to the original methods, with the gains in stability being particularly significant. We also demonstrate empirically that surprisingly this approach benefits methods that are generally thought of as stable already, such as Random Forests.

In Section 2, we describe the general stable methodology, as well as how to quantify the stability of a method. In Section 3, we discuss how to efficiently compute stable solutions. In Section 4, we present computational results comparing five classification methods to their stable counterparts. In Section 6, we summarize our results and report our conclusions.

2. The Stable Methodology

In this section, we describe a way to quantify the stability of a method, and then use this measure to derive the general stable methodology.

2.1 Output Stability

An important measure of the stability of a method is the variability of its output. We next describe how to quantify output stability in the context of both classification and regression.

1. Assuming the data has already been split into training/testing sets, build a model on the training data, and record the prediction assigned to each point.
2. Repeat this procedure s times, each time keeping track of the prediction assigned to each point.
3. This will yield, for each point, an empirical confidence interval, capturing the range of predictions produced by the s models constructed from the training data. Clearly, the narrower the interval the better, as that indicates the method is producing more consistent predictions.

4. For regression problems with n observations, we let $\widehat{y}_{ij}$ be the prediction of model $j \in [s] = \{1, \ldots, s\}$ for point $i \in [n]$ and define

$$\text{RegressionStabilityScore} = \frac{1}{n} \frac{1}{s-1} \sum_{i=1}^{n} \sum_{j=1}^{s} \left(\widehat{y}_{ij} - \frac{1}{s} \sum_{j=1}^{s} \widehat{y}_{ij} \right)^2.$$

5. For classification problems on K classes with n observations, we let $\widehat{p}_{ik}$ the proportion of assignments to class $k \in [K]$ of point $i \in [n]$ among the s models and define

$$\text{ClassificationStabilityScore} = \frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{K} \widehat{p}_{ik}(1 - \widehat{p}_{ik}).$$

Both scores quantify the stability of predictions made by a method with lower scores leading to higher stability.

2.2 Structural Stability

In the case of parameterized models (such as LR and SVM, but not CART, OCT and RF), it is also possible to define a second notion of stability: structural stability that measures the standard deviation of the parameters in the model over s models. This gives some indication of the degree to which the underlying model is changing, a useful counterpart to output stability described above. When it is possible to compute this measure (i.e., for Stable LR and Stable SVM) we report it alongside output stability.

2.3 The Stable Methodology

With a measure of stability defined, we now proceed to derive a methodology for building stable models. At a high-level, what we would like to do is construct a model that is robust to the specific dataset that is used to build the model. One way to think about this, is to view the training dataset as a sample from the true data distribution, and then require that the resulting model be robust to the specific sample that was received. Viewing the partitioning of the data into training/validation sets as a sampling mechanism from this true data distribution (because for a given choice of split, we get one training set), we desire to build models that are robust to every subset in the data.

A way to achieve this is to associate each observations (x_i, y_i) to a binary variable z_i, $i \in [n]$ that indicates whether or not (x_i, y_i) participates in the training set. We can then train a given classification algorithm over all possible allocations of these z_i's, resulting in a model that is explicitly built to do well not just over one training set, as is typical, but over all possible training sets. We now formalize this.

Table 1: List of the Model, Model Class, Loss Function, and Algorithmic details for the five methods considered in this paper.

Method	**Model** m	**Model Class** $\mathcal{M}$	**Loss** $f(m,x,y)$	**Algorithmic Comments**
SVM	(β, β_0)	$\mathbb{R}^p \times \mathbb{R}$	$\max\{0, 1 - y_i(\beta^T x_i + \beta_0)\}$	Linear Optimization Problem
LR	(β, β_0)	$\mathbb{R}^p \times \mathbb{R}$	$\log(1 + e^{-y_i(\beta^T x_i + \beta_0)})$	Convex Optimization Problem
CART	tree of fixed depth	set of all tree models	$\mathbb{1}\{\text{predict}(m,x) \neq y\}$	Solved Greedily
OCT	tree of fixed depth	set of all tree models	$\mathbb{1}\{\text{predict}(m,x) \neq y\}$	Solved to Optimality
RF	set of trees of fixed depth	set of all tree models	$\mathbb{1}\{\text{predict}(m,x) \neq y\}$	Bagging De-correlated Trees

We begin by considering a general model formulation:

$$\min_{m \in \mathcal{M}} \sum_{i=1}^{n} f(m, x_i, y_i), \tag{1}$$

where m is a model optimized over a class of models $\mathcal{M}$, and $f(m,x,y)$ gives the cost of applying model m to a given datapoint (x,y). We list in Table 1 the corresponding model class and loss function for the five classification problems considered in this paper.

Now, we would like to find a model that is robust to the particular data it is trained on. A way to achieve this is to associate to each observation a binary variable z_i that indicates whether or not that specific point will participate in the training set. We can then train a given classification algorithm over all possible allocations of these z_i's, resulting in a model that is explicitly built to do well not just over one training set, as is typical, but over all possible training sets. We now formalize this as

$$\min_{m \in \mathcal{M}} \max_{z \in \mathcal{Z}} \sum_{i=1}^{n} z_i f(m, x_i, y_i), \tag{2}$$

where $\mathcal{Z}$ is the so-called uncertainty set in the language of robust optimization. In this way, we must now optimize a model that minimizes the worst-case training error across elements of $\mathcal{Z}$.

A natural choice of uncertainty set is all subsets of size k:

$$\mathcal{Z} = \left\{ z : \sum_{i=1}^{n} z_i = k, \quad z_i \in \{0,1\},\ i \in [n] \right\}.$$

At an optimal solution of (2), each z_i will be equal to either 0 or 1, with the interpretation that if $z_i = 1$, then point (x_i, y_i) is assigned to the training set, otherwise it it is assigned to the validation set. The number k indicates the desired proportion between the size of the training and validations sets. Namely, by setting $k = 0.7n$ we recover the typical 70/30 training/validation split and by setting $k = 0.5n$ we recover the 50/50 training/validation split, etc.

It is clear why the above formulation is a faithful translation of our earlier intuition: find a model m that does the best against the hardest subset of size k in the data. In Section 3, we discuss how to solve Problem (2).

3. Computing Stable Solutions

In this section, we describe how to compute stable solutions. As we described in the previous section, our formulation belongs to the class of robust optimization (RO) problems. The two most frequently described methods in the literature for solving such problems are reformulation to a deterministic optimization problem (often called the robust counterpart) or an iterative cutting-plane method. Bertsimas et al. (2015) show that both approaches are tractable. In this section, we also develop a third approach based on Monte Carlo simulation that applies widely (in particular to all five problems we consider), while remaining competitive in terms of performance.

In what follows, we first derive the robust counterpart for (2). We then describe how to apply the cutting plane algorithm for (2). Finally, we introduce our third approach for solving RO problems and show how to apply it to (2).

3.1 Tractable Robust Counterpart

Consider again the stable formulation:

$$\min_{m\in\mathcal{M}} \max_{z\in\mathcal{Z}} \sum_{i=1}^{n} z_i f(m, x_i, y_i) \quad \text{with} \quad \mathcal{Z} = \left\{ z : \sum_{i=1}^{n} z_i = k, \quad z_i \in \{0,1\},\ i \in [n] \right\}. \tag{3}$$

As the inner maximization problem is linear in z, the problem is equivalent to optimizing over the convex hull of $\mathcal{Z}$

$$\text{conv}(\mathcal{Z}) = \left\{ z : \sum_{i=1}^{n} z_i = k, \quad 0 \le z_i \le 1,\ i \in [n] \right\}.$$

Thus, Problem (3) is equivalent to

$$\min_{m\in\mathcal{M}} \max_{z\in\text{conv}(\mathcal{Z})} \sum_{i=1}^{n} z_i f(m, x_i, y_i) \quad \text{with} \quad \text{conv}(\mathcal{Z}) = \left\{ z : \sum_{i=1}^{n} z_i = k, 0 \le z_i \le 1,\ i \in [n] \right\}. \tag{4}$$

Problem (4) belongs to the class of robust optimization problems, see Bertsimas et al. (2011) for a review. We leverage techniques from RO to solve Problem (4) efficiently. Namely, to alleviate the multiplication of variables (i.e., the product of z_i with $f(m, x_i, y_i)$) we take the linear optimization dual of the inner maximization problem

$$\max_{z_i} \sum_{i=1}^{n} z_i f(m, x_i, y_i) \quad \text{subject to} \quad \sum_{i=1}^{n} z_i = k, \quad 0 \le z_i \le 1,\ i \in [n]$$

by introducing the dual variable θ for the first constraint and the dual variables $u_i,\ i \in [n]$ for the second set of constraints to arrive at:

$$\min_{\theta, u_i} \ k\theta + \sum_{i=1}^{n} u_i \quad \text{subject to} \quad \theta + u_i \ge f(m, x_i, y_i), \quad u_i \ge 0,\ i \in [n].$$

Substituting this minimization problem back into the outer minimization we arrive at the following problem:

$$\min_{\substack{m\in\mathcal{M};\\ \theta,u_i\in\mathbb{R}}} k\theta+\sum_{i=1}^{n} u_i \quad \text{subject to} \quad \theta+u_i\geq f(m,x_i,y_i), \quad u_i\geq 0,\ i\in[n]. \tag{5}$$

This is a convex optimization problem for $f(\cdot)$ convex, and hence can be solved by commercial optimization software in very high dimensions. Using the formulas for $f(\cdot)$ from Table 1 we have that the stable robust counterparts for SVM and LR

$$\min_{\beta,\beta_0,\theta,u_i} k\theta+\sum_{i=1}^{n} u_i \quad \text{subject to} \quad \theta+u_i\geq max\{0,1-y_i(\beta^T x_i+\beta_0)\}), \quad u_i\geq 0,\ i\in[n], \tag{6}$$

$$\min_{\beta,\beta_0,\theta,u_i} k\theta+\sum_{i=1}^{n} u_i \quad \text{subject to} \quad \theta+u_i\geq \log(1+e^{-y_i(\beta^T x_i+\beta_0)}), \quad u_i\geq 0,\ i\in[n], \tag{7}$$

respectively. Note that the robust counterpart of Stable SVM (6) is a linear optimization problem, easily solvable for very large dimensions, see Bertsimas and Tsitsiklis (1997) for more details, while the robust counterpart of Stable LR (7) is a convex optimization problem, easily solvable for large dimensions, see Boyd and Vandenberghe (2004) for more details. We remark that the robust counterpart method only applies for SVM and LR.

3.2 Cutting Plane Algorithm

We next describe how to apply the cutting plane algorithm to Problem (2). We start with the stable formulation:

$$\min_{m\in\mathcal{M}}\max_{z\in\mathcal{Z}}\sum_{i=1}^{n} z_i f(m,x_i,y_i) \quad \text{with} \quad \mathcal{Z}=\left\{z:\ \sum_{i=1}^{n} z_i=k, \quad z_i\in\{0,1\},\ i\in[n]\right\}.$$

Re-expressing this in an equivalent epigraph formulation we obtain

$$\min_{\substack{m\in\mathcal{M};\\ t\in\mathbb{R}}} t \quad \text{s.t.} \quad t\geq\max_{z\in\mathcal{Z}}\sum_{i=1}^{n} z_i f(m,x_i,y_i),\ \mathcal{Z}=\left\{z:\ \sum_{i=1}^{n} z_i=k, \quad z_i\in\{0,1\},\ i\in[n]\right\}, \tag{8}$$

which is equivalent to:

$$\min_{\substack{m\in\mathcal{M};\\ t\in\mathbb{R}}} t \quad \text{s.t.} \quad t\geq\sum_{i=1}^{n} z_i f(m,x_i,y_i),\ \forall z\in\mathcal{Z}=\left\{z:\ \sum_{i=1}^{n} z_i=k, \quad z_i\in\{0,1\},\ i\in[n]\right\}. \tag{9}$$

We now begin with some random subset $\mathcal{Z}_1\subset\mathcal{Z}$ and solve

$$\min_{\substack{m\in\mathcal{M};\\ t\in\mathbb{R}}} t \quad \text{s.t.} \quad t\geq\sum_{i=1}^{n} z_i f(m,x_i,y_i) \quad \forall z\in\mathcal{Z}_1. \tag{10}$$

We let m_1^*, t_1^* denote minimizers of (10) and search for a violated constraint in the original problem by computing: $\max_{z \in \mathcal{Z}} \sum_{i=1}^n z_i f(m_1^*, x_i, y_i)$. Denote the optimum value of this c^* and the maximizing z by z^*. If $t_1^* \geq c^*$, then m_1^* is optimal for the original problem and we are done. If $t_1^* < c^*$, then the constraint $t \geq \sum_{i=1}^n z_i^* f(m_1^*, x_i, y_i)$ is violated in the original problem. In this case, we need to add this constraint to (10) and repeat, i.e., let $\mathcal{Z}_2 = \mathcal{Z}_1 \cup \{z^*\}$ and then solve:

$$\min_{\substack{m \in \mathcal{M}; \\ t \in \mathbb{R}}} t \quad \text{s.t.} \quad t \geq \sum_{i=1}^n z_i f(m, x_i, y_i) \quad \forall z \in \mathcal{Z}_2, \tag{11}$$

and then repeat this procedure until we find an optimum solution. The algorithm converges as discussed in Fletcher and Leyffer (1994). The method applies to all five classification problems we consider in this paper.

3.3 Monte Carlo

While the cutting plane algorithm described in the previous section is theoretically guaranteed to eventually discover the optimal solution, in practice in may be very slow, especially if the optimization problem (10) is not easy to solve, as is the case with OCT. The reason for the difficulty is the need to solve nested versions of (10) in a loop potentially many times. Instead, we introduce the idea to randomly sample a number ζ of points without replacement from $\mathcal{Z}$, denote this collection $\mathcal{Z}_\zeta$ and solve:

$$\min_{\substack{m \in \mathcal{M}; \\ t \in \mathbb{R}}} t \quad \text{s.t.} \quad t \geq \sum_{i=1}^n z_i f(m, x_i, y_i) \quad \forall z \in \mathcal{Z}_\zeta, \tag{12}$$

and return the resulting $m \in \mathcal{M}$. The method was introduced in Calafiore and Campi (2006) and Campi et al. (2018), where probabilistic guarantees are derived for the solution to be feasible with high probability.

The advantages of this approach are:

(a) it is very fast as we only need to solve (12) once;

(b) it applies to all five classification methods we consider in this paper;

(c) its performance is comparable with the robust counterpart and the cutting planes methods.

While the solution is random as it is dependent on the random sample chosen, we can eliminate the randomness in the solution by employing a scheme similar to that derived in Wyner (1967), wherein the user constructs deterministic sequences to model uniformly distributed points.

4. Computational Experiments:

In this section, we present computational results comparing the five classification methods to their stable counterparts. For SVM and LR we implement all three methods, while for RF, OCT and CART we implemented the Cutting Plane and the Monte Carlo method. We compare these methods along the metrics of accuracy and stability. For accuracy, we report misclassification rate (we also computed Area Under the Curve (AUC) and saw that the results were similar). For stability, we report output stability. For SVM and LR, we additionally report structural stability. We include average results averaged across the 30 datasets. The full results at the individual dataset level can be found in the appendix.

4.1 Testing Methodology

To compare the classification methods to their stable counterparts, we employ the following methodology:

1. We first collected 30 datasets from the UCI Machine Learning Repository (Dua and Taniskidou (2017)). The exact list of datasets can be found in the appendix.
2. For a given dataset, we took a random 10% subset of the data as the testing set. We then applied the following procedure 100 times: we train SVM, LR, CART, RF and OCT on the remaining 90% of the data, and do the same for their stable counterparts.
3. For the Robust Counterpart method for SVM and LR we solved Problems (6) and (7), respectively.
4. For the Cutting Plane approach, we ran the method until convergence, and tracked the performance after each iteration. The performance reported is that of the model found after convergence.
5. For the Monte Carlo approach, we sought to illustrate its performance as a function of the number ζ of samples drawn from $\mathcal{Z}$. Towards that end, we solve Problem (12) for each $\zeta \in \{1, \ldots, 20\}$. The entire trajectory of results is then reported in the Figures below while the final result (for $\zeta = 20$) is reported in the Tables.
6. We report misclassification rate (ACC) and output stability for all methods. For SVM and LR we additionally report structural stability.

4.2 Support Vector Machines

In Table 2, we report the misclassification rate (ACC), output stability, and structural stability for SVM, Stable-Monte Carlo (SMC), Stable - Cutting Plane (SCP), and Stable - Robust Counterpart (SRC). Each entry in Table 2 represents the average metric value for the corresponding method/metric pair over the 30 datasets from the UCI Machine Learning

Table 2: Comparison of Misclassification Rate (ACC), Output Stability, and Structural Stability for Original, SMC, SCP, and SRC versions of SVM.

	ACC	Output Stability	Structural Stability
Original	0.809 (3.4)	0.083 (3.9)	0.025 (3.9)
SMC	0.836 (2.2)	0.018 (2.5)	0.005 (2.2)
SCP	0.828 (2.2)	0.001 (2.0)	0.007 (2.5)
SRC	0.834 (2.1)	0.000 (1.6)	0.000 (1.4)

Table 3: Comparison of Misclassification Rate (ACC), Output Stability, and Structural Stability for Original, SMC, SCP, and SRC versions of LR.

	ACC	Output Stability	Structural Stability
Original	0.815 (3.4)	0.072 (3.9)	0.038 (3.7)
SMC	0.832 (2.1)	0.022 (2.7)	0.030 (2.8)
SCP	0.833 (2.4)	0.016 (1.8)	0.022 (2.5)
SRC	0.836 (2.1)	0.000 (1.6)	0.000 (1.0)

Repository. For accuracy higher numbers are desirable as they indicate greater predictive accuracy. For output stability and structural stability for misclassification rate (ACC), output stability, and structural stability lower numbers are desirable as they indicate greater stability. We also include (in parenthesis) the average rank achieved by that method/metric pair across the 30 datasets, where lower numbers are desirable.

Table 2 indicates that the stable methodology improves both the accuracy of the original method as well as its stability; indeed we see improvements across all metrics, with the improvement in stability being particularly strong. As expected SRC is the strongest method, followed closely by SCP. SMC achieves close results. Figure 1 depicts the evolution of the performance of SRC, SCP and SMC as a function of the number of iterations. "Iteration" for SMC means the number of samples drawn from $\mathcal{Z}$, the intent being to illustrate its performance as a function of the number of samples drawn from $\mathcal{Z}$, and in particular to show how few samples are needed before performance comparable with the SCP and SRC approaches is achieved. After five iterations we observe that SCP and SMC find high quality solutions. We experience strongly diminishing returns on further computation.

4.3 Logistic Regression

In Table 3 , we report results on LR. Table 3 indicates that the stable methodology improves both the accuracy of the original method as well as its stability. As in SVM, SRC is the strongest method, followed closely by SCP. SMC achieves close results. Figure 2 depicts the evolution of the performance of SRC, SCP and SMC as a function of the number of iterations.

Figure 1: Comparison of Original, SMC, SCP, and SRC versions of SVM across average ACC(a), Output Stability(b), and Structural Stability(c) over the 30 datasets as a function of the number of iterations.

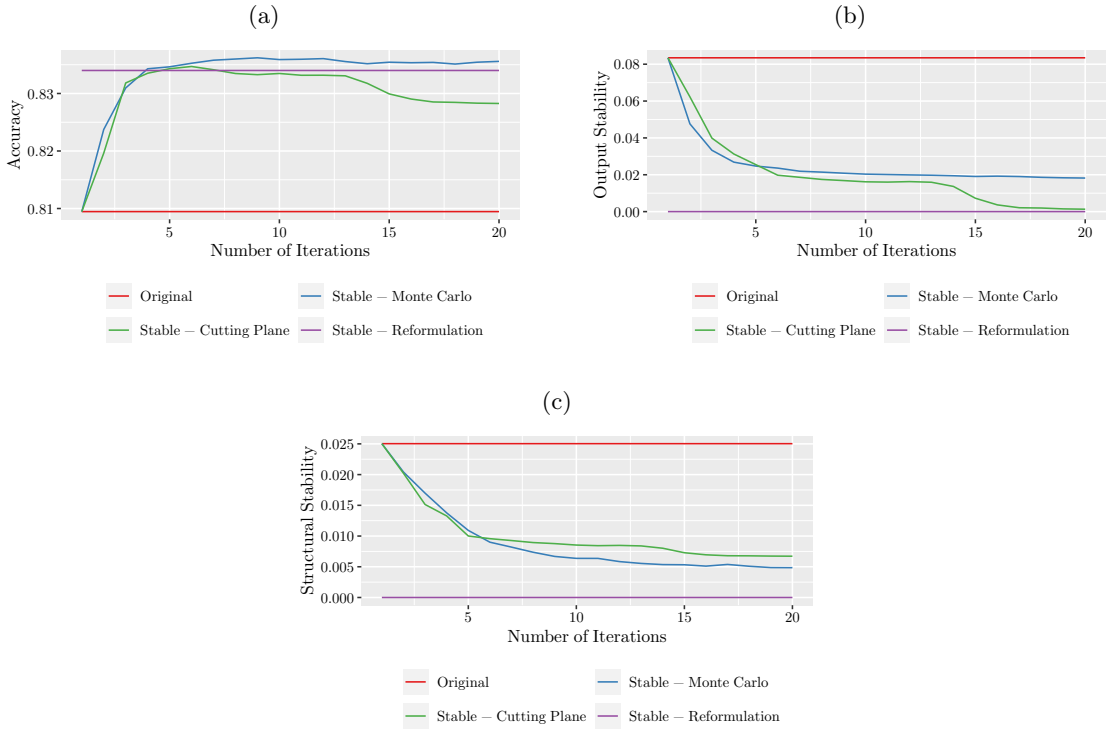


Figure 2: Comparison of LR, SCP, SMC and SRC for LR across average ACC(a), Output Stability(b), and Structural Stability(c) over the 30 datasets as a function of the number of iterations.

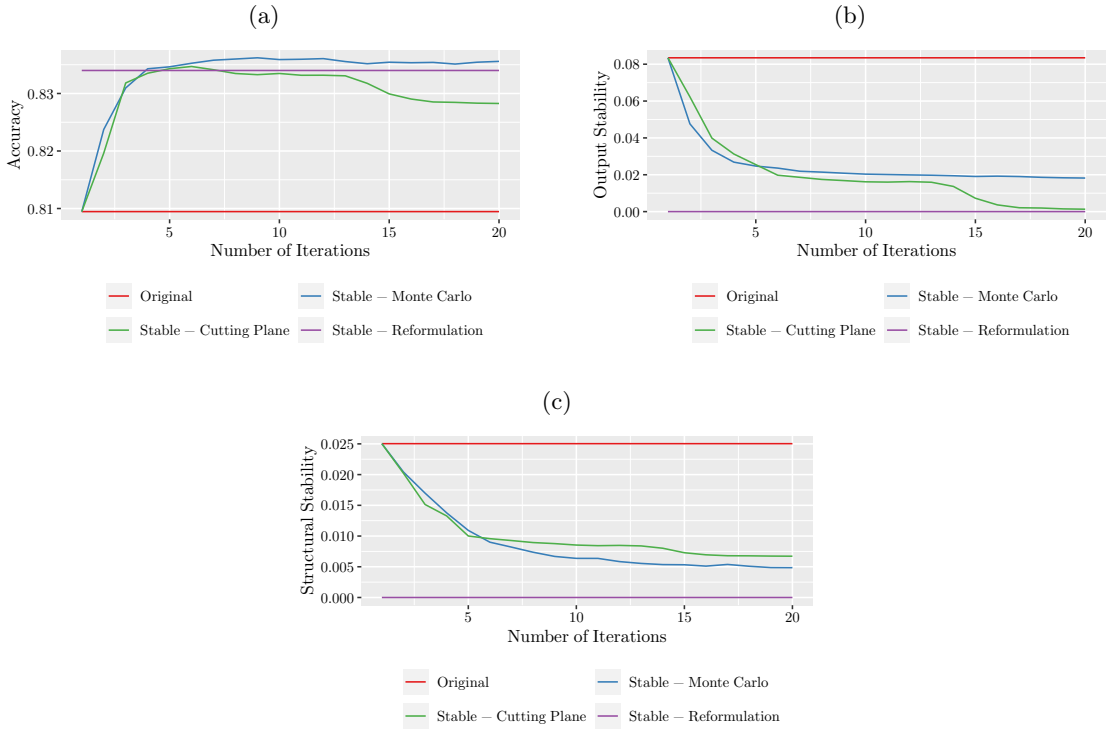
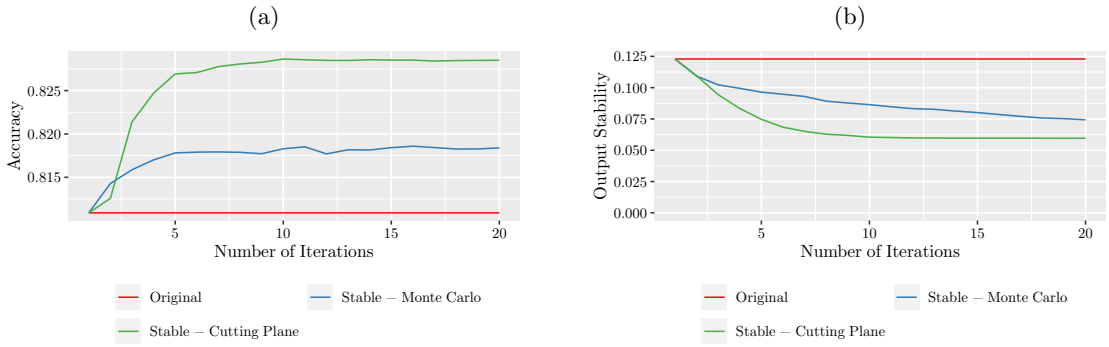


Table 4: Comparison of Misclassification Rate (ACC) and Output Stability for Original, SMC, and SCP versions of CART.

	ACC	Output Stability
Original	0.811 (2.4)	0.123 (3.0)
SMC	0.818 (1.8)	0.074 (1.8)
SCP	0.828 (1.8)	0.060 (1.3)

Figure 3: Comparison of Original, SMC, and SCP versions of CART across average ACC(a) and Output Stability(b) over the 30 datasets as a function of the number of iterations.



4.4 CART

In Table 4, we report results on CART. We observe that the stable solutions improve both the accuracy of the original method as well as its stability; As expected SCP has a small edge over SMC, but SMC is quite close. Figure 3 suggests that we can find high quality solutions for both SCP and SMC quickly.

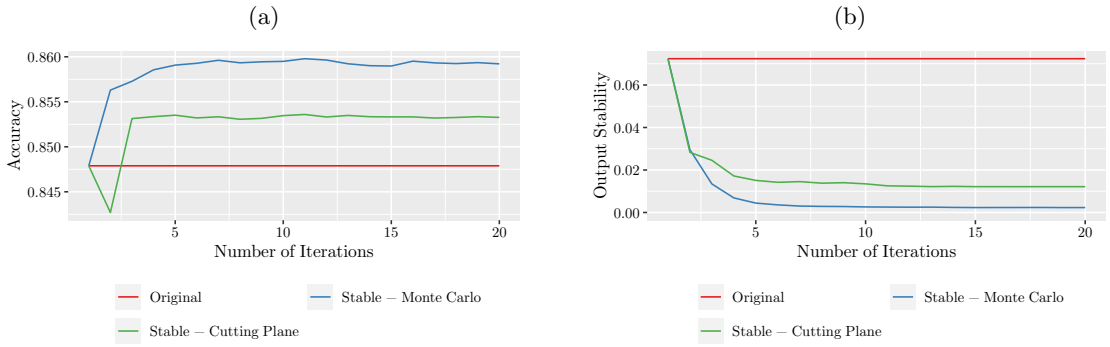
4.5 Random Forests

In Table 5, we report results on RF. Remarkably, the stable solutions improve both the accuracy of the original method and its stability, with the improvement in stability being particularly strong. This latter point is particularly surprising because RF is considered to be quite stable as it was explicitly designed for this purpose. This shows that there is considerable room for stability improvement, even for methods like RF previously thought to be stable. As expected SCP has a small edge over SMC, but SMC is quite close. Figure 4 suggests that we can find high quality solutions for both SCP and SMC quickly.

Table 5: Comparison of Misclassification Rate (ACC) and Output Stability for Original, SMC, and SCP versions of RF.

	ACC	Output Stability
Original	0.848 (2.5)	0.072 (2.9)
SMC	0.859 (1.5)	0.002 (1.4)
SCP	0.853 (1.9)	0.012 (1.6)

Figure 4: Comparison of Original, SMC, and SCP versions of RF across average ACC(a) and Output Stability(b) over the 30 datasets as a function of the number of iterations.

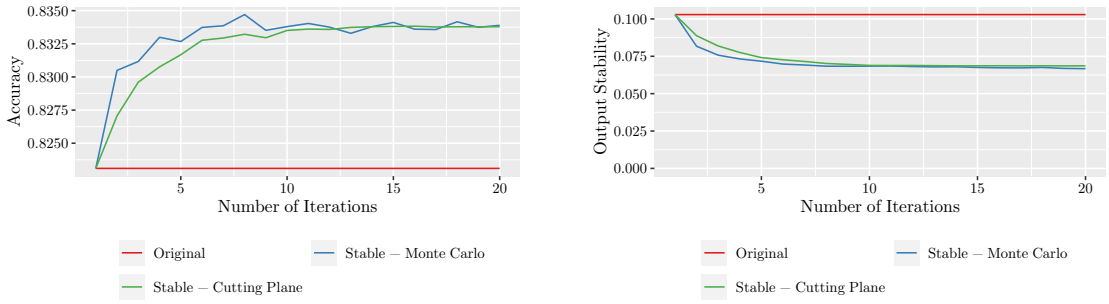


4.6 Optimal Classification Trees

In Table 6, we report results on OCT. The stable solutions improve both the accuracy of the original method and its stability, with the improvement in stability being particularly strong. Interestingly, this time SMC has a slight edge over SCP. Considering that OCT are the most computationally demanding of all examined methods, this is a particularly positive result, as the SMC approach results in effectively no increase in the computational burden. Figure 5 suggests that we can find high quality solutions for both SCP and SMC fast.

Table 6: Comparison of Misclassification Rate (ACC) and Output Stability for Original, SMC, and SCP versions of OCT.

	ACC	Output Stability
Original	0.823 (2.8)	0.103 (2.9)
SMC	0.834 (1.5)	0.067 (1.4)
SCP	0.834 (1.7)	0.069 (1.7)



(a) (b)

Figure 5: Comparison of Original, SMC, and Stable - Cutting Plane versions of Optimal Classification Trees across average ACC(a) and Output Stability(b) over the 30 datasets as a function of the number of iterations.

5. Computational Times

In this section, we compare the computational times of the Original, SMC, SCP, and SRC versions of the five methods, averaged across the 30 datasets. We note that the hardware used for all the experiments was a computer equipped with an Intel Core i9-9900K processor, while for the Software we used Julia 1.3.1, Ipopt 3.13.2 for LR, and Gurobi 9.0.0 for SVM.

The results can be found in Table 7, which is organized as follows: each row corresponds to an implementation, each column to a classification method. Entry (i, j) then corresponds to the average computational time for implementation i of classification method j. Note that the times are first scaled so that the original method has time 1, so that the other method times indicate the overhead factor for that method, i.e., 2 means takes twice as long as the original method, 0.5 means takes half as long.

Overall, we note that the stable versions of each classification method have computational times comparable with the original methods, suggesting that the stable methodology is scalable. Indeed, we even see in a few cases the approach offers a speed improvement over the original (i.e., whenever a reformulation is possible, as well as for the SMC versions of all the methods except for RF and OCT). This was surprising, as one would typically expect the runtime to increase with additional constraints. We believe that the robust constraints make the optimal solution in some sense "more obvious" and able to be found faster. Finally, as expected, the SCP approach has the longest runtimes, in the worst case 8.9 times slower than the original, and in the best case 1.6 times slower than the original. While certainly still reasonable, the possibility of even speeding up the original method while retaining the performance qualities of the SCP clearly makes the SMC approach quite attractive.

	SVM	LR	CART	RF	OCT
Original	1.000	1.000	1.000	1.000	1.000
SMC	0.398	0.625	**0.989**	**3.149**	**2.176**
SCP	1.621	4.093	1.965	8.920	2.880
Stable - Reformulation	**0.312**	**0.488**	NA	NA	NA

Table 7: Comparison of the computational times of the Original, SMC, SCP, and SRC versions of the five methods, averaged across the 30 datasets. The best stable run time for each method has been bolded.

6. Discussion

Overall, the results strongly suggest that employing the stable methodology is a good idea. For all five classification methods we saw employing the stable methodology led to improvements in both accuracy and stability. Especially for stability, the improvement was quite substantial. This was particularly surprising to observe in methods such as RF which were already thought to be stable.

In the case of SVM and LR, we have the ability to derive tractable exact robust counterparts, and as expected, the results suggest that this approach should be taken as it leads to better performance over the SMC and SCP approaches, and surprisingly, faster run times than even the original method. In the case of CART, RF, and OCT, both the SCP and SMC approaches showed sizeable improvements, especially for stability. For these cases, however, the SMC approach has runtimes comparable to the original methods, while essentially retaining the performance benefits of the SCP approach, a result which makes the SMC approach particularly attractive.

The SMC's strong performance may have come as a surprise to the reader. Our explanation, is that its advantage comes from communicating to the optimization problem that its proposed solution will be graded against how it performs across a variety of scenarios, and thus to not over-optimize for one, as is usually the case. Intuition then would suggest that a healthy improvement is to be expected after even just showing the model a few new scenarios, after which new scenarios offer diminishing returns on performance. Indeed the results confirm this.

7. Conclusion

In this paper, we propose a robust optimization based framework for stabilizing any classification method and derive efficient algorithms that scale the approach to very large problem sizes. The approach is generally applicable to general classification problems. Through experiments on 30 datasets with sizes ranging between 10^2 and 10^4 observations and features, we show that our approach (a) leads to improvements in both performance and stability compared to the original methods, with the gains in stability being particularly significant

(even, surprisingly, for those methods that were previously thought to be stable, such as RF) and (b) has computational times comparable with (and indeed in some cases even faster than) the original methods allowing the method to be very scalable.

What is perhaps most exciting, is that all of these benefits accrue to even the simplest implementation of stability: the Monte Carlo approach. In this approach, practitioners have a conceptually simple prescription for how to train models that barely increases the computational complexity over their un-stabilized counterparts. The fact that it leads to improvements in both stability and accuracy suggest that perhaps the current approaches to training algorithms have been operating at an interior point with respect to the performance/stability pareto curve. The results suggest that we can in fact make significant improvements in both accuracy and stability, without paying much of a computational cost, leaving the practitioner little reason not to employ the methodology.

References

Dimitris Bertsimas and Jack Dunn. Optimal classification trees. *Machine Learning*, 106: 1039–1082, 2017.

Dimitris Bertsimas and Jack Dunn. *Machine Learning under a Modern Optimization Lens*. Dynamic Ideas, 2019.

Dimitris Bertsimas and Ivan Paskov. Stable regression: On the power of optimization over randomization in training regression problems. *Machine Learning*, 2020. under review.

Dimitris Bertsimas and John Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, 1st edition, 1997.

Dimitris Bertsimas, David B. Brown, and Constantine Caramanis. Theory and applications of robust optimization. *SIAM Review*, 53(3):464–501, 2011.

Dimitris Bertsimas, Iain Dunning, and Miles Lubin. Reformulation versus cutting-planes for robust optimization a computational study. *Computational Management Science*, 13 (2):195–217, 2015.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, USA, 2004.

Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996a.

Leo Breiman. Heuristics of instability and stabilization in model selection. *Annals of Statistics*, 24(6):2350–2383, 1996b.

Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

Leo. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth and Brooks, 1984.

Giuseppe Calafiore and Marco Campi. The scenario approach to robust control design. *IEEE Transactions on Automatic Control*, 51(5):742–753, 2006.

Marco Campi, Simone Garatti, and Federico Ramponi. A general scenario theory for non-convex optimization and decision making. *IEEE Transactions on Automatic Control*, 63 (12):4067–4078, 2018.

David R. Cox. Some procedures connected with the logistic qualitative response curve. *Research Papers in Probability and Statistics*, 1, 1966.

Wen-de Dong, Yue-ting Chen, Zhi-hai Xu, Hua-jun Feng, and Qi Li. Image stabilization with support vector machine. *Journal of Zhejiang University SCIENCE C*, 12:478–485, 06 2011.

Dheeru Dua and Efi Taniskidou. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

Roger Fletcher and Sven Leyffer. Solving mixed integer nonlinear programs by outer approximation. *Mathematical Programming*, 66(1):327–349, 1994.

Yoav Freund and Robert Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. *European Conference on Computational Learning Theory*, pages 23–37, 1995.

Mark Last, Oded Maimon, and Einat Minkov. Improving stability of decision trees. *IJPRAI*, 16:145–159, 03 2002.

Vladimir Vapnik and Anatoly Lerner. Pattern recognition using generalized portrait method. *Automation and Remote Control*, 24, 1963.

Aaron Wyner. Random packings and coverings of the unit n-sphere. *The Bell System Technical Journal*, 46(9):2111–2118, 1967.

Table 8: Comparison of Misclassification Rate for Original, SMC, SCP, and SRC versions of SVM. The results indicate that the stable methodology improves the misclassifcation rate. In particular, we see that original, SMC, SCP, and SRC versions of SVM achieve an average misclassification rate of 0.809, 0.836, 0.828, 0.834, respectively.

	Original	SMC	SCP	SRC
acute-inflammations-1	1.000	1.000	1.000	**1.000**
acute-inflammations-2	1.000	1.000	1.000	**1.000**
banknote-authentication	0.992	0.992	0.993	**0.993**
blood-transfusion-service-center	**0.765**	0.763	0.763	0.763
breast-cancer-wisconsin-diagnostic	0.927	**0.953**	0.941	0.936
breast-cancer-wisconsin-original	0.962	**0.964**	0.961	0.961
breast-cancer-wisconsin-prognostic	0.690	**0.752**	0.741	0.741
breast-cancer	0.741	0.753	0.756	**0.762**
climate-model-simulation-crashes	0.945	**0.949**	0.938	0.938
congressional-voting-records	0.957	0.971	**0.979**	0.971
connectionist-bench-sonar	0.695	0.677	0.752	**0.758**
credit-approval	0.851	0.864	0.870	**0.883**
fertility	0.811	**0.867**	0.833	0.833
haberman-survival	**0.743**	0.736	0.725	0.725
hepatitis	0.781	0.917	0.892	**0.917**
indian-liver-patient	0.695	0.713	0.713	**0.713**
ionosphere	0.835	0.879	0.895	**0.895**
mammographic-mass	0.825	0.831	0.831	**0.831**
monks-problems-1	0.740	0.729	0.769	**0.842**
monks-problems-2	0.579	**0.620**	0.380	0.380
monks-problems-3	0.798	0.811	**0.813**	0.811
parkinsons	0.754	0.846	0.879	**0.879**
planning-relax	0.695	0.709	0.709	**0.709**
qsar-biodegradation	0.847	**0.855**	0.849	0.849
seismic-bumps	0.934	0.934	0.934	**0.934**
spect-heart	0.631	0.659	0.649	**0.750**
spectf-heart	0.511	**0.708**	0.660	0.625
statlog-project-german-credit	0.770	0.783	**0.788**	0.787
thoracic-surgery	0.837	0.851	0.851	**0.851**
tic-tac-toe-endgame	0.971	0.980	0.983	**0.983**

8. Appendix

In this section, we present computational results at an individual data-set level comparing SVM, LR, CART, RF, and OCT to their stable counterparts. We apply SRC, SCP ad SMC for SVM and LR. We apply SCP ad SMC for CART, RF, and OCT. We compare these methods along accuracy and stability. For accuracy, we report misclassification rate (we also computed Area Under the Curve (AUC) and saw that the results were similar). For stability, we report output stability. For SVM's and LR, we additionally report structural stability. The best entry (i.e., for ACC, the largest, for Output and Structural Stability, the lowest) entry in each row has been made bold to aid readability.

Table 9: Comparison of Output Stability for Original, SMC, SCP and SRC versions of SVM. The results indicate that the stable methodology improves output stability. In particular, we see that original, SMC, SCP, and SRC versions of SVM achieve an average output stability of 0.083, 0.018, 0.001, 0.000, respectively.

	Original	SMC	SCP	SRC
acute-inflammations-1	0.000	0.000	0.000	**0**
acute-inflammations-2	0.000	0.000	0.000	**0**
banknote-authentication	0.004	0.001	0.000	**0**
blood-transfusion-service-center	0.005	0.000	0.000	**0**
breast-cancer-wisconsin-diagnostic	0.043	0.000	0.000	**0**
breast-cancer-wisconsin-original	0.013	0.005	0.000	**0**
breast-cancer-wisconsin-prognostic	0.196	0.068	0.000	**0**
breast-cancer	0.112	0.067	0.004	**0**
climate-model-simulation-crashes	0.034	0.009	0.000	**0**
congressional-voting-records	0.046	0.000	0.000	**0**
connectionist-bench-sonar	0.192	0.000	0.000	**0**
credit-approval	0.071	0.042	0.005	**0**
fertility	0.098	0.000	0.000	**0**
haberman-survival	0.051	0.000	0.000	**0**
hepatitis	0.129	0.000	0.000	**0**
indian-liver-patient	0.137	0.000	0.000	**0**
ionosphere	0.087	0.023	0.000	**0**
mammographic-mass	0.075	0.058	0.000	**0**
monks-problems-1	0.171	0.105	0.008	**0**
monks-problems-2	0.251	0.000	0.000	**0**
monks-problems-3	0.048	0.000	0.000	**0**
parkinsons	0.124	0.026	0.000	**0**
planning-relax	0.044	0.000	0.000	**0**
qsar-biodegradation	0.060	0.035	0.000	**0**
seismic-bumps	0.001	0.000	0.001	**0**
spect-heart	0.169	0.043	0.015	**0**
spectf-heart	0.159	0.000	0.000	**0**
statlog-project-german-credit	0.123	0.062	0.005	**0**
thoracic-surgery	0.050	0.000	0.002	**0**
tic-tac-toe-endgame	0.009	0.002	0.000	**0**

Table 10: Comparison of Structural Stability for Original, SMC, SCP and SRC versions of SVM. The results indicate that the stable methodology improves the structural stability. In particular, we see that original, SMC, SCP, and SRC versions of SVM achieve an average output stability of 0.025, 0.005, 0.007, 0.000, respectively.

	Original	SMC	SCP	SRC
acute-inflammations-1	0.000	0.000	0.000	**0**
acute-inflammations-2	0.000	0.000	0.000	**0**
banknote-authentication	0.000	0.000	0.000	**0**
blood-transfusion-service-center	0.087	0.000	0.049	**0**
breast-cancer-wisconsin-diagnostic	0.031	0.000	0.025	**0**
breast-cancer-wisconsin-original	0.021	0.007	0.000	**0**
breast-cancer-wisconsin-prognostic	0.027	0.003	0.000	**0**
breast-cancer	0.022	0.015	0.004	**0**
climate-model-simulation-crashes	0.004	0.000	0.000	**0**
congressional-voting-records	0.030	0.000	0.023	**0**
connectionist-bench-sonar	0.016	0.000	0.014	**0**
credit-approval	0.015	0.013	0.014	**0**
fertility	0.038	0.000	0.000	**0**
haberman-survival	0.068	0.000	0.000	**0**
hepatitis	0.050	0.000	0.036	**0**
indian-liver-patient	0.032	0.000	0.000	**0**
ionosphere	0.012	0.001	0.000	**0**
mammographic-mass	0.019	0.009	0.000	**0**
monks-problems-1	0.056	0.053	0.000	**0**
monks-problems-2	0.041	0.000	0.000	**0**
monks-problems-3	0.008	0.000	0.000	**0**
parkinsons	0.047	0.020	0.000	**0**
planning-relax	0.037	0.000	0.000	**0**
qsar-biodegradation	0.011	0.009	0.000	**0**
seismic-bumps	0.010	0.002	0.010	**0**
spect-heart	0.020	0.005	0.006	**0**
spectf-heart	0.021	0.000	0.011	**0**
statlog-project-german-credit	0.004	0.001	0.000	**0**
thoracic-surgery	0.025	0.006	0.009	**0**
tic-tac-toe-endgame	0.000	0.000	0.000	**0**

Table 11: Comparison of Misclassification Rate for Original, SMC, SCP and SRC versions of LR. The results indicate that the stable methodology improves the misclassification rate. In particular, we see that original, SMC, SCP, and SRC versions of LR achieve an average misclassification rate of 0.815, 0.832, 0.833, 0.836, respectively.

	Original	SMC	SCP	SRC
acute-inflammations-1	1.000	1.000	1.000	**1.000**
acute-inflammations-2	1.000	1.000	1.000	**1.000**
banknote-authentication	0.992	0.992	0.993	**0.993**
blood-transfusion-service-center	**0.779**	0.774	0.768	0.768
breast-cancer-wisconsin-diagnostic	0.926	0.931	0.931	**0.942**
breast-cancer-wisconsin-original	0.961	0.961	0.966	**0.966**
breast-cancer-wisconsin-prognostic	0.704	**0.784**	0.776	0.776
breast-cancer	0.766	**0.788**	0.785	0.786
climate-model-simulation-crashes	0.949	**0.961**	0.957	0.957
congressional-voting-records	0.980	0.986	0.985	**0.986**
connectionist-bench-sonar	0.730	0.760	**0.768**	0.758
credit-approval	0.853	0.861	0.861	**0.867**
fertility	0.810	0.862	0.867	**0.867**
haberman-survival	**0.765**	0.753	0.747	0.747
hepatitis	0.792	0.888	**0.902**	0.875
indian-liver-patient	0.689	**0.693**	0.678	0.678
ionosphere	0.847	0.872	0.876	**0.876**
mammographic-mass	0.827	0.827	0.835	**0.835**
monks-problems-1	0.708	0.730	0.789	**0.789**
monks-problems-2	0.556	0.572	0.515	**0.600**
monks-problems-3	0.775	0.811	0.795	**0.811**
parkinsons	0.774	**0.812**	0.810	0.810
planning-relax	0.597	0.658	0.709	**0.709**
qsar-biodegradation	0.850	0.858	0.858	**0.861**
seismic-bumps	0.930	**0.932**	0.929	0.929
spect-heart	**0.657**	0.624	0.626	0.625
spectf-heart	0.665	**0.679**	0.672	0.667
statlog-project-german-credit	0.775	0.786	0.795	**0.797**
thoracic-surgery	0.824	0.834	0.834	**0.837**
tic-tac-toe-endgame	0.972	**0.974**	0.972	0.972

Table 12: Comparison of Output Stability Rate for Original, SMC, SCP and SRC versions of LR. The results indicate that the stable methodology improves the output stability. In particular, we see that original, SMC, SCP, and SRC versions of LR achieve an average output stability of 0.072, 0.022, 0.016, 0.000, respectively.

	Original	SMC	SCP	SRC
acute-inflammations-1	0.000	0.000	0.000	**0**
acute-inflammations-2	0.000	0.000	0.000	**0**
banknote-authentication	0.004	0.002	0.000	**0**
blood-transfusion-service-center	0.011	0.006	0.000	**0**
breast-cancer-wisconsin-diagnostic	0.043	0.000	0.000	**0**
breast-cancer-wisconsin-original	0.012	0.004	0.000	**0**
breast-cancer-wisconsin-prognostic	0.197	0.059	0.000	**0**
breast-cancer	0.112	0.045	0.000	**0**
climate-model-simulation-crashes	0.036	0.011	0.000	**0**
congressional-voting-records	0.024	0.000	0.000	**0**
connectionist-bench-sonar	0.146	0.000	0.000	**0**
credit-approval	0.066	0.040	0.000	**0**
fertility	0.092	0.005	0.000	**0**
haberman-survival	0.033	0.017	0.000	**0**
hepatitis	0.118	0.000	0.000	**0**
indian-liver-patient	0.093	0.054	0.000	**0**
ionosphere	0.090	0.028	0.000	**0**
mammographic-mass	0.028	0.022	0.000	**0**
monks-problems-1	0.143	0.070	0.000	**0**
monks-problems-2	0.248	0.094	0.485	**0**
monks-problems-3	0.042	0.000	0.000	**0**
parkinsons	0.111	0.025	0.000	**0**
planning-relax	0.126	0.016	0.000	**0**
qsar-biodegradation	0.054	0.026	0.000	**0**
seismic-bumps	0.007	0.003	0.000	**0**
spect-heart	0.165	0.055	0.000	**0**
spectf-heart	0.007	0.000	0.000	**0**
statlog-project-german-credit	0.102	0.047	0.004	**0**
thoracic-surgery	0.054	0.021	0.000	**0**
tic-tac-toe-endgame	0.010	0.005	0.000	**0**

Table 13: Comparison of Structural Stability Rate for Original, SMC, SCP and SRC versions of LR. The results indicate that the stable methodology improves the structural stability. In particular, we see that original, SMC, SCP, and SRC versions of LR achieve an average output stability of 0.038, 0.030, 0.022, 0.000, respectively.

	Original	SMC	SCP	SRC
acute-inflammations-1	0.004	0.034	0.004	**0**
acute-inflammations-2	0.004	0.003	0.004	**0**
banknote-authentication	0.000	0.000	0.000	**0**
blood-transfusion-service-center	0.168	0.135	0.213	**0**
breast-cancer-wisconsin-diagnostic	0.033	0.014	0.011	**0**
breast-cancer-wisconsin-original	0.019	0.003	0.000	**0**
breast-cancer-wisconsin-prognostic	0.026	0.002	0.000	**0**
breast-cancer	0.016	0.010	0.013	**0**
climate-model-simulation-crashes	0.003	0.000	0.000	**0**
congressional-voting-records	0.008	0.000	0.000	**0**
connectionist-bench-sonar	0.015	0.002	0.005	**0**
credit-approval	0.022	0.021	0.010	**0**
fertility	0.058	0.038	0.042	**0**
haberman-survival	0.289	0.282	0.000	**0**
hepatitis	0.049	0.009	0.008	**0**
indian-liver-patient	0.055	0.066	0.051	**0**
ionosphere	0.006	0.000	0.000	**0**
mammographic-mass	0.073	0.076	0.079	**0**
monks-problems-1	0.002	0.021	0.001	**0**
monks-problems-2	0.025	0.012	0.055	**0**
monks-problems-3	0.003	0.000	0.000	**0**
parkinsons	0.047	0.020	0.000	**0**
planning-relax	0.067	0.031	0.057	**0**
qsar-biodegradation	0.021	0.012	0.000	**0**
seismic-bumps	0.036	0.036	0.036	**0**
spect-heart	0.015	0.009	0.000	**0**
spectf-heart	0.008	0.006	0.014	**0**
statlog-project-german-credit	0.015	0.014	0.015	**0**
thoracic-surgery	0.022	0.016	0.010	**0**
tic-tac-toe-endgame	0.034	0.032	0.029	**0**

Table 14: Comparison of Misclassification Rate for Original, SMC and SCP versions of CART. The results indicate that the stable methodology improves the misclassification rate. In particular, we see that original, SMC, and SCP versions of CART achieve an average misclassification rate of 0.811, 0.818, 0.828, respectively.

	Original	SMC	SCP
acute-inflammations-1	0.999	1.000	**1.000**
acute-inflammations-2	0.999	**1.000**	0.999
banknote-authentication	0.961	**0.968**	0.966
blood-transfusion-service-center	**0.780**	0.780	0.765
breast-cancer-wisconsin-diagnostic	0.921	0.925	**0.926**
breast-cancer-wisconsin-original	0.954	0.956	**0.960**
breast-cancer-wisconsin-prognostic	0.657	0.665	**0.674**
breast-cancer	**0.735**	0.735	0.728
climate-model-simulation-crashes	0.888	**0.932**	0.915
congressional-voting-records	0.986	**0.995**	0.986
connectionist-bench-sonar	0.748	0.781	**0.800**
credit-approval	0.868	**0.880**	0.874
fertility	**0.844**	0.844	0.837
haberman-survival	0.696	0.705	**0.711**
hepatitis	0.813	0.835	**0.866**
indian-liver-patient	0.638	0.661	**0.690**
ionosphere	**0.841**	0.816	0.817
mammographic-mass	0.839	**0.840**	0.820
monks-problems-1	0.762	0.731	**0.957**
monks-problems-2	0.583	**0.616**	0.591
monks-problems-3	**0.840**	0.829	0.838
parkinsons	0.846	**0.867**	0.861
planning-relax	0.572	0.633	**0.705**
qsar-biodegradation	**0.810**	0.804	0.803
seismic-bumps	0.925	0.928	**0.934**
spect-heart	0.732	0.746	**0.748**
spectf-heart	**0.696**	0.662	0.692
statlog-project-german-credit	0.712	**0.718**	0.707
thoracic-surgery	0.819	0.846	**0.850**
tic-tac-toe-endgame	**0.864**	0.855	0.834

Table 15: Comparison of Output Stability for Original, SMC and SCP versions of CART. The results indicate that the stable methodology improves output stability. In particular, we see that original, SMC, and SCP versions of CART achieve an average output stability of 0.123, 0.074, 0.060, respectively.

	Original	SMC	SCP
acute-inflammations-1	0.009	0.000	**0.000**
acute-inflammations-2	0.000	0.000	**0.000**
banknote-authentication	0.028	0.022	**0.019**
blood-transfusion-service-center	0.078	0.059	**0.024**
breast-cancer-wisconsin-diagnostic	0.049	0.028	**0.028**
breast-cancer-wisconsin-original	0.030	**0.004**	0.010
breast-cancer-wisconsin-prognostic	0.233	0.169	**0.166**
breast-cancer	0.128	0.079	**0.028**
climate-model-simulation-crashes	0.067	0.038	**0.020**
congressional-voting-records	0.038	**0.005**	0.013
connectionist-bench-sonar	0.229	0.122	**0.122**
credit-approval	0.111	0.085	**0.046**
fertility	0.124	0.030	**0.023**
haberman-survival	0.188	**0.146**	0.177
hepatitis	0.112	0.056	**0.022**
indian-liver-patient	0.226	0.229	**0.145**
ionosphere	0.070	**0.026**	0.027
mammographic-mass	0.067	0.058	**0.032**
monks-problems-1	0.195	0.171	**0.045**
monks-problems-2	0.338	**0.062**	0.334
monks-problems-3	0.044	0.027	**0.000**
parkinsons	0.112	**0.066**	0.071
planning-relax	0.303	0.203	**0.023**
qsar-biodegradation	0.121	**0.104**	0.112
seismic-bumps	0.030	0.019	**0.000**
spect-heart	0.117	0.017	**0.005**
spectf-heart	0.205	0.104	**0.103**
statlog-project-german-credit	0.172	0.125	**0.089**
thoracic-surgery	0.098	0.020	**0.001**
tic-tac-toe-endgame	0.164	0.155	**0.103**

Table 16: Comparison of Misclassification Rate for Original, SMC and SCP versions of RF. The results indicate that the stable methodology improves the misclassification rate. In particular, we see that original, SMC, and SCP versions of RFt achieve an average misclassification rate of 0.848, 0.859, 0.853, respectively.

	Original	SMC	SCP
acute-inflammations-1	1.000	1.000	**1.000**
acute-inflammations-2	1.000	1.000	**1.000**
banknote-authentication	0.991	0.993	**0.994**
blood-transfusion-service-center	0.749	0.745	**0.760**
breast-cancer-wisconsin-diagnostic	0.953	**0.959**	0.957
breast-cancer-wisconsin-original	0.971	0.973	**0.974**
breast-cancer-wisconsin-prognostic	0.710	**0.718**	0.707
breast-cancer	0.750	**0.760**	0.702
climate-model-simulation-crashes	0.936	**0.943**	0.914
congressional-voting-records	0.995	0.998	**0.998**
connectionist-bench-sonar	0.872	**0.912**	0.886
credit-approval	0.890	**0.896**	0.891
fertility	0.857	**0.875**	0.834
haberman-survival	0.666	0.671	**0.726**
hepatitis	0.902	**0.916**	0.888
indian-liver-patient	0.676	0.690	**0.712**
ionosphere	0.932	**0.936**	0.932
mammographic-mass	0.803	**0.809**	0.808
monks-problems-1	0.816	0.832	**0.853**
monks-problems-2	0.647	**0.702**	0.673
monks-problems-3	**0.829**	0.811	0.811
parkinsons	0.937	**0.961**	0.954
planning-relax	0.616	0.652	**0.706**
qsar-biodegradation	0.856	**0.869**	0.868
seismic-bumps	0.933	0.933	**0.934**
spect-heart	0.765	**0.775**	0.758
spectf-heart	0.816	**0.854**	0.840
statlog-project-german-credit	0.766	**0.773**	0.700
thoracic-surgery	0.840	0.844	**0.845**
tic-tac-toe-endgame	0.964	**0.977**	0.974

Table 17: Comparison of Output Stability for Original, SMC and SCP versions of RF. The results indicate that the stable methodology improves output stability. In particular, we see that original, SMC, and SCP versions of RF achieve an average output stability of 0.072, 0.002, 0.012, respectively.

	Original	SMC	SCP
acute-inflammations-1	0.000	0.000	**0.000**
acute-inflammations-2	0.000	0.000	**0.000**
banknote-authentication	0.006	0.000	**0.000**
blood-transfusion-service-center	0.103	0.018	**0.011**
breast-cancer-wisconsin-diagnostic	0.020	0.000	**0.000**
breast-cancer-wisconsin-original	0.015	0.000	**0.000**
breast-cancer-wisconsin-prognostic	0.115	**0.000**	0.097
breast-cancer	0.110	**0.006**	0.009
climate-model-simulation-crashes	0.035	0.000	**0.000**
congressional-voting-records	0.024	0.000	**0.000**
connectionist-bench-sonar	0.098	0.000	**0.000**
credit-approval	0.063	0.002	**0.002**
fertility	0.049	**0.000**	0.001
haberman-survival	0.142	**0.009**	0.030
hepatitis	0.072	**0.000**	0.018
indian-liver-patient	0.132	**0.000**	0.008
ionosphere	0.032	0.000	**0.000**
mammographic-mass	0.090	0.012	**0.009**
monks-problems-1	0.084	0.000	**0.000**
monks-problems-2	0.194	**0.000**	0.005
monks-problems-3	0.027	0.000	**0.000**
parkinsons	0.056	**0.000**	0.001
planning-relax	0.167	**0.000**	0.128
qsar-biodegradation	0.061	0.001	**0.001**
seismic-bumps	0.027	0.003	**0.000**
spect-heart	0.143	0.000	**0.000**
spectf-heart	0.114	**0.000**	0.007
statlog-project-german-credit	0.117	**0.010**	0.036
thoracic-surgery	0.059	0.008	**0.003**
tic-tac-toe-endgame	0.016	0.000	**0.000**

Table 18: Comparison of Misclassification Rate for Original, SMC and SCP versions of OCT. The results indicate that the stable methodology improves the misclassification rate. In particular, we see that original, SMC, and SCP versions of OCT achieve an average misclassification rate of 0.823, 0.834, 0.834, respectively.

	Original	SMC	SCP
acute-inflammations-1	1.000	1.000	**1.000**
acute-inflammations-2	0.985	**0.986**	0.985
banknote-authentication	0.975	**0.978**	0.973
blood-transfusion-service-center	0.767	**0.769**	0.765
breast-cancer-wisconsin-diagnostic	0.924	0.931	**0.931**
breast-cancer-wisconsin-original	0.958	0.962	**0.962**
breast-cancer-wisconsin-prognostic	0.651	0.675	**0.685**
breast-cancer	0.727	0.741	**0.747**
climate-model-simulation-crashes	0.899	**0.928**	0.918
congressional-voting-records	0.979	0.983	**0.983**
connectionist-bench-sonar	0.748	0.770	**0.773**
credit-approval	0.879	0.889	**0.890**
fertility	0.843	**0.881**	0.869
haberman-survival	0.663	0.695	**0.705**
hepatitis	0.845	0.839	**0.845**
indian-liver-patient	0.652	0.665	**0.679**
ionosphere	0.874	**0.885**	0.883
mammographic-mass	0.830	**0.833**	0.831
monks-problems-1	0.998	**1.000**	0.999
monks-problems-2	0.643	**0.657**	0.654
monks-problems-3	0.815	0.825	**0.830**
parkinsons	0.869	**0.886**	0.882
planning-relax	0.569	0.591	**0.616**
qsar-biodegradation	0.819	**0.821**	0.820
seismic-bumps	0.922	0.930	**0.934**
spect-heart	0.759	0.760	**0.760**
spectf-heart	0.683	**0.718**	0.717
statlog-project-german-credit	0.724	**0.726**	0.721
thoracic-surgery	0.809	0.821	**0.845**
tic-tac-toe-endgame	**0.880**	0.875	0.811

Table 19: Comparison of Output Stability for Original, SMC and SCP versions of OCT. The results indicate that the stable methodology improves output stability. In particular, we see that original, SMC, and SCP versions of OCT achieve an average output stability of 0.103, 0.067, 0.069, respectively.

	Original	SMC	SCP
acute-inflammations-1	0.000	0.000	**0.000**
acute-inflammations-2	0.005	0.000	**0.000**
banknote-authentication	0.010	**0.003**	0.006
blood-transfusion-service-center	0.078	0.052	**0.051**
breast-cancer-wisconsin-diagnostic	0.040	0.023	**0.020**
breast-cancer-wisconsin-original	0.027	0.018	**0.018**
breast-cancer-wisconsin-prognostic	0.218	0.142	**0.140**
breast-cancer	0.118	0.089	**0.078**
climate-model-simulation-crashes	0.065	**0.036**	0.045
congressional-voting-records	0.031	**0.010**	0.019
connectionist-bench-sonar	0.180	**0.090**	0.108
credit-approval	0.085	**0.066**	0.068
fertility	0.133	0.087	**0.063**
haberman-survival	0.177	**0.112**	0.117
hepatitis	0.085	**0.022**	0.056
indian-liver-patient	0.228	0.199	**0.166**
ionosphere	0.064	**0.028**	0.029
mammographic-mass	0.056	**0.048**	0.050
monks-problems-1	0.003	**0.000**	0.002
monks-problems-2	0.186	**0.134**	0.151
monks-problems-3	0.066	**0.035**	0.041
parkinsons	0.096	**0.032**	0.048
planning-relax	0.252	**0.202**	0.215
qsar-biodegradation	0.111	**0.097**	0.100
seismic-bumps	0.030	0.016	**0.000**
spect-heart	0.167	**0.080**	0.109
spectf-heart	0.167	**0.072**	0.088
statlog-project-german-credit	0.174	0.113	**0.086**
thoracic-surgery	0.095	0.050	**0.026**
tic-tac-toe-endgame	**0.142**	0.145	0.158