INFORMS Journal on Optimization

Robust Classification

Dimitris Bertsimas, Jack Dunn, Colin Pawlowski, Ying Daisy Zhuo

informs
http://pubsonline.informs.org/journal/ijoo/

INFORMS JOURNAL ON OPTIMIZATION
Vol. 1, No. 1, Winter 2019, pp. 2–34
ISSN 2575-1484 (print), ISSN 2575-1492 (online)

Robust Classification

Dimitris Bertsimas,[a] **Jack Dunn,**[a] **Colin Pawlowski,**[a] **Ying Daisy Zhuo**[a]

[a] Sloan School of Management and Operations Research Center, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139

Contact: dbertsim@mit.edu, https://orcid.org/0000-0002-1985-1003 (DB); jackdunn@mit.edu, https://orcid.org/0000-0002-6936-4502 (JD); cpawlows@mit.edu, https://orcid.org/0000-0003-2781-7507 (CP); zhuo@mit.edu, https://orcid.org/0000-0002-8579-932X (YDZ)

Received: October 17, 2017
Revised: February 11, 2018; May 7, 2018
Accepted: May 8, 2018
Published Online in Articles in Advance:
October 19, 2018

https://doi.org/10.1287/ijoo.2018.0001

Abstract. Motivated by the fact that there may be inaccuracies in features and labels of training data, we apply robust optimization techniques to study in a principled way the uncertainty in data features and labels in classification problems and obtain robust formulations for the three most widely used classification methods: support vector machines, logistic regression, and decision trees. We show that adding robustness does not materially change the complexity of the problem and that all robust counterparts can be solved in practical computational times. We demonstrate the advantage of these robust formulations over regularized and nominal methods in synthetic data experiments, and we show that our robust classification methods offer improved out-of-sample accuracy. Furthermore, we run large-scale computational experiments across a sample of 75 data sets from the University of California Irvine Machine Learning Repository and show that adding robustness to any of the three nonregularized classification methods improves the accuracy in the majority of the data sets. We observe the most significant gains for robust classification methods on high-dimensional and difficult classification problems, with an average improvement in out-of-sample accuracy of robust versus nominal problems of 5.3% for support vector machines, 4.0% for logistic regression, and 1.3% for decision trees.

History: Dick den Hertog served as the editor-in-chief for this paper.
Funding: This material is based on work supported by the National Science Foundation Graduate Research Fellowship [Grant 1122374].

Keywords: robust optimization • machine learning • classification problems

1. Introduction

Three of the most widely used classification methods are support vector machines (SVM), logistic regression, and classification and regression trees (CART) (Friedman et al. 2001). These classifiers are among the state-of-the-art machine learning methods, giving high out-of-sample accuracy on many real-world data sets and admitting tractable training algorithms for large-scale problems. However, in many scenarios, the training data are subject to uncertainty, which can negatively affect the performance of these classifiers. Regularization is a common technique for mitigating the effect of data uncertainty and addressing the problem of overfitting. In this paper, we propose a novel approach for developing improved classifiers using techniques from robust optimization to explicitly model uncertainty in the data in a principled manner.

Support vector machines were first introduced by Cortes and Vapnik (1995) and have gained popularity since then. SVM classifiers find a hyperplane that maximizes the margin of separation and use a hinge loss function when the data are not separable. Alternatively, the geometric concept of margin can be viewed as a form of regularization. Previous work has shown the equivalence between support vector machines and a robust formulation of the hinge loss classifier (Xu et al. 2009). In this paper, we develop new robust formulations for SVM and other classifiers, which lead to further gains in out-of-sample accuracy compared to nonrobust methods.

Logistic regression is one of the oldest and most widely used classification methods that models the probability of a response belonging to a certain class. The performance of logistic regression can be improved by introducing a regularization term to penalize model complexity, and the resulting problem can be solved efficiently for large-scale problems (Friedman et al. 2010). Decision trees, a family of classification methods, aim to partition the space recursively and make predictions based on the region into which the points fall. Popular methods such as CART (Breiman et al. 1984) construct the partitions with greedy heuristic methods, although recently methods have been developed that efficiently find globally optimal solutions to the decision tree problem (Bertsimas and Dunn 2017). In practice, scientists and researchers apply these methods to real-world problems using packages which have been developed in R and other programming languages. Methods for SVM, logistic regression, and CART are included in the R packages E1071, STATS, and RPART, respectively.

The model training problems for SVM, logistic regression, and decision trees can all be formulated and solved as traditional optimization problems and therefore can benefit from the systematic improvements in model formulation and solver speeds in this area. Recent studies have explored using modern mixed-integer optimization (MIO) methods to solve problems in classical statistics, such as the least quantile squares (Bertsimas and Mazumder 2014) and best subset selection problems (Bertsimas et al. 2016), and to create algorithmic approaches for fitting regression models (Bertsimas and King 2015, 2017). These methods have been successful in part due to dramatic increases in hardware and software computing power for MIO over the past 30 years.

One of the biggest challenges in the field of machine learning is to design models that avoid the issue of *overfitting*, where the model describes the noise instead of the underlying relationship. Strong models should take into consideration the noise structure during model estimation, and in many real-world problems, the data representing both the feature variable ($\mathbf{x}_i, i = 1, \ldots n$) as well as the label variable ($y_i, i = 1, \ldots, n$) are subject to error. For example, the "Wisconsin Diagnostic Breast Cancer" data set is widely used in the machine learning community. This data set involves classifying benign and malignant tumors, with features computed from digitized images including the radius, texture, symmetry, etc. of the cell nuclei. Even though the features in this data set are relatively precisely measured, the images are not free from noise, and the accuracy of the measurements depends on the precision of the recognition programs. More generally, in data sets with missing data that require imputation, uncertainties are also introduced.

As an example of label uncertainty, in the "Contraceptive Method Choice" data set from the University of California Irvine (UCI) Machine Learning Repository, women were surveyed to report their current contraceptive method choice, as well as demographic and socioeconomic characteristics. Because of the survey nature of the data, we may suspect that some respondents have reported dishonest answers to the questions about their choice of contraceptive method. In cancer clinical trials, caregivers determine whether or not each patient has achieved remission, and these labels are subjective and depend upon the accuracy of the tumor measurement. Another common source for such errors is the employment of labeling personnel to provide labels for the training set. Therefore, it seems natural to expect that some of the labels may be incorrect when training the classifier.

1.1. Related Work

To date, there has not been a principled way of modeling data uncertainty directly for classification problems in the literature. In this paper, we propose a framework based on robust optimization to address classification problems whose data (both in features and in labels) are subject to error. Robust optimization is a flexible framework for modeling uncertainty (Ben-Tal et al. 2009) and is arguably one of the fastest-growing areas of optimization in the last decade. For a wide variety of problems in domains such as finance, statistics, and health care, robust formulations have been shown to be computationally tractable and lead to improved solutions compared to the classical optimization formulations (Bertsimas et al. 2011). The key advantage of robust solutions is that they provide near-optimal solutions that remain feasible when problem parameters are perturbed and thus are attractive when the problem is subject to uncertainty.

In particular, robust optimization has been shown to lead to improvements for many statistics problems. In the machine-learning community, the success of SVM in classification and Lasso in regression has been largely attributed to their regularization terms that reduce data overfitting. Pant et al. (2011) demonstrate how robust classification can be used to handle situations with imbalanced training data, and Livni et al. (2012) derive classifiers protected against stochastic adversarial perturbations to the training data. Xu et al. (2009) establish that robustness is a key *reason* behind the strong performance of regularized methods, due to the generalization ability of robustness.

There has been prior work that considered robust optimization classifiers based upon SVM, first proposed in Bi and Zhang (2005) and Bhattacharyya et al. (2005). These approaches have dealt mainly with feature uncertainty. One of the robust classification methods proposed in this paper, namely, feature-robust SVM, closely resembles the linear optimization robust classifiers proposed by Trafalis and Gilbert (2007), except that these methods contain an additional regularizer term in the objective. This difference is important because, more recently, it has been shown that a robust optimization formulation of the maximum margin classifier is equivalent to the classical SVM; thus, methods derived as robust variations to classical SVM are "double-counting" the effect of robustness (Xu et al. 2009, Bertsimas and Copenhaver 2017). In addition, there have been previous attempts to model uncertainties in labels for SVM, although these methods are largely heuristic in nature and have been tested primarily on synthetic or contaminated data (Biggio et al. 2011, Natarajan et al. 2013). There has also been work on robustifying kernel SVM methods against feature uncertainty by Ben-Tal et al. (2012). The approach we present could be extended to kernel methods, but this is beyond the scope of the paper.

For logistic regression, regularized versions such as Elastic Net have been proposed (Zou and Hastie 2005), which consider adding a convex combination of the ℓ_1 and ℓ_2-norm penalty to the objective; however these regularized classifiers were not derived using tools from robust optimization. Using robust optimization, logistic regression models that are robust to feature uncertainty have been derived for various uncertainty sets (El Ghaoui et al. 2003, Harrington et al. 2010).

To our knowledge, no work has been done framing decision trees as a robust optimization problem. Because tractable formulations and solution methods for the optimal decision tree problem were proposed quite recently in Bertsimas and Dunn (2017), robust optimal decision trees have not been explored.

In summary, results from the literature indicate that ideas from robust optimization have the potential to add value to existing classification methods. Prior work on SVM establishes the equivalence between regularization and robustness for certain problems, and in some examples robust classifiers yield higher out-of-sample accuracy compared to nominal methods. However, these works have largely focused on theoretical derivations of robust methods, with limited testing on synthetic data. Without extensive computational experiments, we do not know whether these robust classifiers yield gains in out-of-sample accuracy in practice, especially in comparison with regularized methods.

We build upon these previous efforts to present a framework for robust classification that accommodates three of the most widely used classification methods: SVM, logistic regression, and CART. By considering a diverse variety of classifiers, we compare the impact of adding robustness to different models, and we evaluate the performance of these methods in practice through large-scale computational experiments.

1.2. Contributions

This paper shows how to incorporate robustness in classification problems generally. Under the framework of robust optimization, we systematically develop new robust methods that offer predictable improvements in out-of-sample accuracy over nominal classifiers. We summarize our contributions in this paper below:

1. We present a principled framework for robust classification, which combines ideas from robust optimization and machine learning, with an aim to build classifiers that model data uncertainty directly. Building on previous work for modeling feature uncertainty, we introduce an approach for modeling uncertainty in labels, as well as both features and labels simultaneously. By viewing machine-learning algorithms as a family of optimization problems, we show that the robustification of existing classification methods can be done in a unified and principled way. This leads to tractable problems with relatively small overhead compared to the original methods. In particular, we use this framework to derive counterparts to SVM, logistic regression, and CART that are robust to variations in features and labels in the data. In the case where we consider feature uncertainty only, the resulting robust formulations for SVM and logistic regression match previous results in the literature.

2. We demonstrate the advantage of robust formulations over regularized and nominal methods through synthetic data experiments with two classes divided by a separating hyperplane. Compared to nominal and regularized methods, the robust SVM and logistic regression methods recover the separating hyperplane classifiers closer to the truth, leading to gains in out-of-sample accuracy, especially in the worst-case analysis.

3. We demonstrate that robust classification improves out-of-sample accuracy in large-scale computational experiments across a sample of 75 data sets from the UCI Machine Learning Repository. Furthermore, we identify characteristics of classification problems for which robust methods lead to significant accuracy gains compared to nonrobust methods. Specifically, in problems with high-dimensional data and difficult separability, the value of robustness is even more prominent.

4. We provide a simple, empirically derived decision rule for machine-learning practitioners that predicts with high accuracy when robust methods can offer significant improvement over the nominal methods, with an average improvement in out-of-sample accuracy of 5.3% for SVM, 4.0% for logistic regression, and 1.3% for CART. Compared to regularized SVM or logistic regression, the average out-of-sample accuracy improvement of our principled approach to robustness is 2.1% over regularized SVM and 1.2% over regularized logistic regression when this rule is satisfied.

We would like to distinguish robust optimization in statistical problems from the field of robust statistics, developed by Huber (1981), which studies how an estimator performs under perturbation of the model. Even though both fields share the motivation to avoid unduly effects from outliers, the underlying methodologies are totally different and address the problems from separate angles. Although robust statistics passively evaluates the robustness properties of a given algorithm, robust optimization actively constructs models that take into account data uncertainty.

The structure of the paper is as follows. In Section 2, we present a selection of widely used classification methods. In Section 3, we give a brief introduction to robust optimization and introduce some terms and properties that will be used later. In Section 4, we demonstrate how to apply robust optimization to the classification methods to derive a family of classification methods that are robust to uncertainty in the features of the training data set. In Section 5, we repeat this process to develop methods that are robust to uncertainty in data set labels. In Section 6, we combine these approaches to develop classification methods that are robust to noise in both features and labels. In Section 7, we compare the performance of these robust classification methods to their nonrobust counterparts and regularized methods through a series of synthetic data experiments. In Section 8, we comprehensively compare the performance of our robust classifiers to their benchmark methods on a wide range of real data sets. We conclude in Section 9.

2. Overview of Classification Methods

In this section, we present a selection of widely used methods for classification. These are the methods to which we will later apply robust optimization techniques. For this section and in the rest of the paper, let $\{\mathbf{x}_i, y_i\}_{i=1}^n$ be the training data provided for the classification task, where $\mathbf{x}_i \in \mathbb{R}^p$ is the feature vector and $y_i \in \{-1, 1\}$ is the label for observation i.

2.1. Soft-Margin Support Vector Machines

Soft-margin support vector machines are a variation on the simpler maximal margin classifier that relax the requirement that the data be separable and instead allow for points to be incorrectly classified (Cortes and Vapnik 1995). Support vector machines use hinge loss as the loss function and balance the minimization of total loss and maximization of margin with parameter C that can be tuned via validation. This classifier can be formulated as the following problem:

$$\min_{\mathbf{w},b} \quad \tfrac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^n \max\{1 - y_i(\mathbf{w}^T\mathbf{x}_i - b), 0\}. \tag{1}$$

Problem (1) can equivalently be formulated as the following problem:

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \tfrac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^n \xi_i \\ \text{s.t.} \quad & y_i(\mathbf{w}^T\mathbf{x}_i - b) \geq 1 - \xi_i \quad i = 1, \ldots, n, \\ & \xi_i \geq 0 \qquad\qquad\qquad\quad i = 1, \ldots, n. \end{aligned} \tag{2}$$

The dual problem can be formulated through the use of Lagrange multipliers:

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & C\sum_{i=1}^n \alpha_i - \frac{1}{2}\sum_{i=1}^n\sum_{j=1}^n \alpha_i\alpha_j y_i y_j \mathbf{x_i}^T\mathbf{x_j} \\ \text{s.t.} \quad & 0 \leq \alpha_i \leq C \quad i = 1, \ldots, n, \\ & \sum_{i=1}^n \alpha_i y_i = 0. \end{aligned}$$

Both the primal and dual are convex quadratic optimization problems. Because the dual problem has fewer decision variables, and the majority of these variables tend to be equal to zero or the cost parameter C in the optimal solution, it is typically the problem solved in practice (Friedman et al. 2001). In addition, the dual form is advantageous because it allows us to do the kernel trick to learn nonlinear decision rules (Cortes and Vapnik 1995). Alternatively, we may modify the objective function of Problem (1) by changing the norm of the regularizer term from ℓ_2 to ℓ_1 (Zhu et al. 2004). The resulting classifier is formulated as follows:

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \|\mathbf{w}\|_1 + C\sum_{i=1}^n \xi_i \\ \text{s.t.} \quad & y_i(\mathbf{w}^T\mathbf{x}_i - b) \geq 1 - \xi_i \quad i = 1, \ldots, n, \\ & \xi_i \geq 0 \qquad\qquad\qquad\quad i = 1, \ldots, n. \end{aligned} \tag{3}$$

Problem (3), which we refer to as ℓ_1-regularized SVM, is equivalent to a linear optimization problem, which is efficiently solvable.

2.2. Logistic Regression

Logistic regression assumes the response variable Y follows a Bernoulli distribution with the probability depending on the $\mathbf{x}$ and the model parameter $\boldsymbol{\beta} \in \mathbb{R}^p, \beta_0 \in \mathbb{R}$

$$\mathbb{P}(Y = 1|\mathbf{X} = \mathbf{x}) = \frac{e^{\boldsymbol{\beta}^T\mathbf{x}+\beta_0}}{1 + e^{\boldsymbol{\beta}^T\mathbf{x}+\beta_0}},$$

$$\mathbb{P}(Y = -1|\mathbf{X} = \mathbf{x}) = \frac{1}{1 + e^{\boldsymbol{\beta}^T\mathbf{x}+\beta_0}}.$$

Concisely, the conditional probability can be written as

$$\mathbb{P}(Y = y_i|\mathbf{X} = \mathbf{x}) = \frac{1}{1 + e^{-y_i(\boldsymbol{\beta}^T\mathbf{x}_i+\beta_0)}}.$$

Logistic regression coefficients $\boldsymbol{\beta}$ and β_0 are typically fit using the maximum-likelihood method. The log-likelihood is

$$-\sum_{i=1}^{n} \log\left(1 + e^{-y_i(\boldsymbol{\beta}^T\mathbf{x}_i+\beta_0)}\right).$$

Therefore, the maximum-likelihood estimators $\boldsymbol{\beta}$ and β_0 aim to solve the following problem:

$$\max_{\boldsymbol{\beta},\beta_0} -\sum_{i=1}^{n} \log\left(1 + e^{-y_i(\boldsymbol{\beta}^T\mathbf{x}_i+\beta_0)}\right). \tag{4}$$

Problem (4) is a concave maximization problem that is efficiently solvable by methods such as coordinate descent or Newton's method (Bertsekas 1999).

Similar to the regularization techniques in the popular lasso regression (Tibshirani 1996) for variable selection and shrinkage, a regularization term can be added to the logistic regression likelihood function, giving

$$\max_{\boldsymbol{\beta},\beta_0} -\sum_{i=1}^{n} \log\left(1 + e^{-y_i(\boldsymbol{\beta}^T\mathbf{x}_i+\beta_0)}\right) - \lambda\|\boldsymbol{\beta}\|_q, \tag{5}$$

where $\|\cdot\|_q$ is a given ℓ_q norm.

2.3. Decision Trees and CART

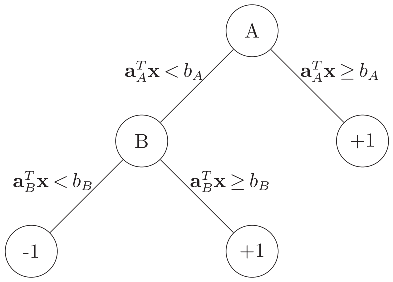
Decision trees are a family of classification methods that seek to recursively partition the feature space into disjoint regions and predict labels for new points based upon the region into which the point falls. The most widely used method for training decision trees is CART (Breiman et al. 1984), which takes a greedy heuristic approach to constructing the tree rather than posing the entire process as a single optimization problem.

However, in order to use robust optimization techniques to create robust decision trees, we require the formulation of the decision-tree training problem as a formal optimization problem. Optimal decision trees (Bertsimas and Dunn 2017) are a recent method that considers the entire decision-tree learning procedure as a single mixed-integer optimization problem and uses this to take a globally optimal view while constructing the tree. To create robust decision-tree methods, we will take the optimal decision tree problem and apply robust optimization.

Consider the problem of training a general decision tree. At each branch node in the tree, a split of the form $\mathbf{a}^T\mathbf{x} < b$ is applied. Points that satisfy this constraint will follow the left branch of the tree, whereas those that violate the constraint follow the right branch. Each leaf node is assigned a label, and each point is assigned the label of the leaf node into which the point falls. Figure 1 summarizes this for an example decision tree with two branch nodes, A and B, that apply splits $\mathbf{a}_A^T\mathbf{x} < b_A$ and $\mathbf{a}_B^T\mathbf{x} < b_B$ respectively. There are three leaf nodes that assign labels $\{-1\}$, $\{+1\}$, and $\{+1\}$ (from left to right in the figure).

Given that the tree contains K nodes, we define the sets $\mathcal{P}_k^l$, $\mathcal{P}_k^u$, and $\mathcal{P}_k$ for $k = 1, \ldots, K$ to capture the hierarchy of the tree.

Figure 1. An Example of a Decision Tree with Two Partition Nodes and Three Leaf Nodes


A
$\mathbf{a}_A^T\mathbf{x} < b_A$
$\mathbf{a}_A^T\mathbf{x} \geq b_A$
B
+1
$\mathbf{a}_B^T\mathbf{x} < b_B$
$\mathbf{a}_B^T\mathbf{x} \geq b_B$
-1
+1

- $\mathcal{P}_k^l$ = the ancestors of node k in the tree of which we have taken the left branch (a split of the form $\mathbf{a}_k^T\mathbf{x}_i < b_k$) to get to node k;
- $\mathcal{P}_k^u$ = the ancestors of node k of which we have taken the right branch (a split of the form $\mathbf{a}_k^T\mathbf{x}_i \geq b_k$) to get to node k;
- $\mathcal{P}_k = \mathcal{P}_k^l \cup \mathcal{P}_k^u$, that is, all ancestors of node k.

We will now state the optimal decision tree problem from Bertsimas and Dunn (2017) below as Problem (6) and then provide an explanation of the model:

$$\min \quad \sum_{k=1}^{K} f_k - \sum_{k=1}^{K} \lambda_k d_k \tag{6a}$$

$$\text{s.t.} \quad g_k = \sum_{i=1}^{n} \frac{1-y_i}{2} z_{ik} \qquad k = 1, \dots, K, \tag{6b}$$

$$h_k = \sum_{i=1}^{n} \frac{1+y_i}{2} z_{ik} \qquad k = 1, \dots, K, \tag{6c}$$

$$f_k \leq g_k + M[w_k + (1-c_k)] \qquad k = 1, \dots, K, \tag{6d}$$

$$f_k \leq h_k + M[(1-w_k) + (1-c_k)] \qquad k = 1, \dots, K, \tag{6e}$$

$$f_k \geq g_k - M[(1-w_k) + (1-c_k)] \qquad k = 1, \dots, K, \tag{6f}$$

$$f_k \geq h_k - M[w_k + (1-c_k)] \qquad k = 1, \dots, K, \tag{6g}$$

$$d_k = 1 \qquad k = \lceil K/2 \rceil, \dots, K, \tag{6h}$$

$$d_k \leq d_j \qquad k = 1, \dots, K, \forall j \in \mathcal{P}_k, \tag{6i}$$

$$d_k + \sum_{l=1}^{p} a_{kl} = 1 \qquad k = 1, \dots, K, \tag{6j}$$

$$\sum_{k=1}^{K} z_{ik} = 1 \qquad i = 1, \dots, n, \tag{6k}$$

$$z_{ik} \leq d_k \qquad i = 1, \dots, n, k = 1, \dots, K, \tag{6l}$$

$$z_{ik} \leq 1 - d_j \qquad i = 1, \dots, n, k = 1, \dots, K, \forall j \in \mathcal{P}_k, \tag{6m}$$

$$\sum_{i=1}^{n} z_{ik} \geq N c_k \qquad k = 1, \dots, K, \tag{6n}$$

$$c_k \geq d_k - \sum_{j \in \mathcal{P}_k} d_j \qquad k = 1, \dots, K, \tag{6o}$$

$$\mathbf{a}_j^T\mathbf{x}_i + \epsilon \leq b_j + M(1-z_{ik}) \qquad i = 1, \dots, n, k = 1, \dots, K, \forall j \in \mathcal{P}_k^l, \tag{6p}$$

$$\mathbf{a}_j^T\mathbf{x}_i \geq b_j - M(1-z_{ik}) \qquad i = 1, \dots, n, k = 1, \dots, K, \forall j \in \mathcal{P}_k^u, \tag{6q}$$

$$\mathbf{a}_k \in \{0,1\}^p \qquad k = 1, \dots, K, \tag{6r}$$

$$0 \leq b_k \leq 1 \qquad k = 1, \dots, K, \tag{6s}$$

$$z_{ik}, w_k, c_k, d_k \in \{0,1\} \qquad i = 1, \dots, n, k = 1, \dots, K. \tag{6t}$$

At each node $k = 1, \dots, K$ in the tree, we must decide whether to apply a split or set the node to be a leaf node. The binary variable d_k takes value 1 if no split is applied, and 0 otherwise.

If we choose to apply a split at a node k, the variables $\mathbf{a}_k$ and b_k are used to set a split of the form $\mathbf{a}_k^T\mathbf{x} < b_k$. To mirror the behavior of CART, we only consider univariate decision trees, and, hence, we only allow a single variable to be used in each split. This is achieved by the constraints (6r), which forces the components of $\mathbf{a}_k$ to be binary, and (6j) means we can only choose one of these variables at each node. Note that (6j) also forces $\mathbf{a} = \mathbf{0}$ if $d_k = 1$, so we cannot apply a split at a node that has been marked as a leaf node.

We use the binary variables z_{ik} to track which leaf node k each point $i = 1, \ldots, n$ in the training set is assigned. Constraints (6p) and (6q) ensure that points are assigned only to a node if they satisfy all required splits, whereas constraints (6l) and (6m) ensure that points can only be assigned to leaf nodes. Finally, (6k) ensures that each point is assigned to exactly one leaf node.

The objective is to minimize the number of misclassified points. The number of misclassified points in a node k is tracked by using the variable f_k. Note that it is always better to assign the leaf node a label that agrees with the most common label among points in the node. This means that the misclassification count is given by the size of the minority label. We use the variables g_k and h_k to count the number of points of each label in each node k, which is achieved with constraints (6b) and (6c). Constraints (6d)–(6g) set f_k to $\min\{g_k, h_k\}$ to count the misclassification in each node, and the objective sums this misclassification over all nodes.

CART imposes a constraint relating to the minbucket parameter, which requires each leaf node to contain at least this number of points. Constraints (6n) and (6o) enforce this restriction in the model for a supplied minbucket parameter N.

The small number of remaining constraints relates to ensuring the decision to split or not at each node is permitted by the structure of the tree. For example, no leaf node is permitted to have a child node. We omit the full details of these precedence constraints from this description of the model and instead refer the reader to Bertsimas and Dunn (2017) for the complete description.

This is a mixed-integer optimization problem that is practically solvable on real-world data sets and leads to results that are highly competitive with heuristic decision-tree methods like CART [see Bertsimas and Dunn (2017) for a comprehensive comparison].

3. Brief Overview of Robust Optimization

In this section, we give an overview of robust optimization and introduce the notions of uncertainty sets and dual norms that will be used later when applying robust optimization techniques to the unified classification framework.

Robust optimization is a means for modeling uncertainty in optimization problems without the use of probability distributions. Under this modeling framework, we construct deterministic *uncertainty sets* that contain possible values of uncertain parameters. We then seek a solution that is optimal for all such realizations of this uncertainty. Consider the general optimization problem:

$$\begin{aligned} \max_{\mathbf{x}\in\mathscr{X}} \quad & c(\mathbf{u}, \mathbf{x}) \\ \text{s.t.} \quad & \mathbf{g}(\mathbf{u}, \mathbf{x}) \le \mathbf{0}, \end{aligned}$$

where $\mathbf{x}$ is the vector of decision variables in feasible region $\mathscr{X}$, $\mathbf{u}$ is a vector of given parameters, c is a real-valued function, $\mathbf{g}$ is a vector-valued function, and $\mathbf{0}$ is the vector of all zeros. Relaxing the assumption that $\mathbf{u}$ is fixed, we assume instead that the realized values of $\mathbf{u}$ are restricted to be within some uncertainty set $\mathscr{U}$. We form the corresponding robust optimization problem by optimizing against the worst-case realization of the uncertain parameters across the entire uncertainty set:

$$\begin{aligned} \max_{\mathbf{x}\in\mathscr{X}} \quad & \min_{\mathbf{u}\in\mathscr{U}} c(\mathbf{u}, \mathbf{x}) \\ \text{s.t.} \quad & \mathbf{g}(\mathbf{u}, \mathbf{x}) \le \mathbf{0} \quad \forall \mathbf{u} \in \mathscr{U}. \end{aligned}$$

Despite typically having an infinite number of constraints, it is often possible to reformulate the problem as a deterministic optimization problem with finite size, depending on the choice of uncertainty set $\mathscr{U}$. The resulting deterministic problem is deemed the *robust counterpart*, which may be a problem of the same complexity as the nominal problem, depending on the structure of $\mathscr{U}$.

There is extensive evidence in the literature that robust solutions have significant advantages relative to nominal solutions. A case study of linear optimization problems from the NETLIB library found that in 13 out of 90 problems, the optimal nonrobust solution violates some of the inequality constraints by more than 50% of the right-hand-side values, when the uncertain coefficients are subject to small (0.01%) perturbations. On the other hand, robust solutions for these identical problems, which are feasible for all perturbations up to 0.1%, lead to objective values that are within 1% of the optimal (Ben-Tal and Nemirovski 2000).

3.1. Dual Norms

Let $\mathbf{x} = (x_1, \ldots, x_n)$ be a vector in $\mathbb{R}^n$. For any real number $q \geq 1$, we define the ℓ_q norm of $\mathbf{x}$ in the standard way, denoted by $\|\mathbf{x}\|_q$:

$$\|\mathbf{x}\|_q \triangleq \left(\sum_{i=1}^{n} |x_i|^q\right)^{\frac{1}{q}}.$$

A particular problem that is encountered frequently when using robust optimization is the so-called *dual-norm* problem:

$$\max_{\|\mathbf{x}\|_q \leq 1} \{\mathbf{a}^T\mathbf{x}\}.$$

When $q > 1$, the optimal solution to this problem is $\|\mathbf{a}\|_{q*}$, where $q^* = 1/(1 - 1/q)$. This ℓ_{q^*} norm is called the *dual norm* of the ℓ_q norm. In addition, when $q = 1$, it can be shown that the optimal solution to this problem is $\|\mathbf{a}\|_\infty$, where the ℓ_∞ norm of a vector $\mathbf{x} \in \mathbb{R}^n$ is defined by

$$\|\mathbf{x}\|_\infty \triangleq \lim_{q\to\infty} \|\mathbf{x}\|_q = \max\{|x_1|, |x_2|, \ldots, |x_n|\}.$$

A simple extension to this problem is when the norm of $\mathbf{x}$ is restricted by any number $\rho > 0$. In this case we have the following:

$$\max_{\|\mathbf{x}\|_q \leq \rho} \{\mathbf{a}^T\mathbf{x}\} = \max_{\|\mathbf{y}\|_q \leq 1} \{\mathbf{a}^T(\rho\mathbf{y})\} = \rho \cdot \max_{\|\mathbf{y}\|_q \leq 1} \{\mathbf{a}^T\mathbf{y}\}, \tag{7}$$

and the optimal solution to this problem is thus $\rho\|\mathbf{a}\|_{q^*}$.

4. Robustness Against Uncertainty in Features

In this section, we present the notion of robustifying classification methods against uncertainties in the features of the training set. Using an uncertainty set to model possible values of the features in reality, we then define and state the *feature-robust counterpart* for each of the classification methods. We note that the feature-robust counterparts for SVM and logistic regression are known in the literature, but we include their derivation here for completeness.

4.1. Motivating Feature-Robustness

Uncertainties in the features can arise from measurement errors during data collection and from input errors during data manipulation and missing value imputation. If left unaddressed, the trained model may be biased and severely influenced by inaccuracies in the data. Our goal is to train a *feature-robust* model that takes such uncertainties into account, which is stable and provides high accuracy in circumstances where data are perturbed.

With the robust approach, such uncertainties are taken into consideration when training the classifiers. To model uncertainty in the features of the training set, we assume that the data $\mathbf{x}_i$ are subject to additive perturbations $\Delta\mathbf{x}_i \in \mathbb{R}^p$, $i = 1, \ldots, n$. Let $\Delta\mathbf{X} = (\Delta\mathbf{x}_1, \Delta\mathbf{x}_2, \ldots \Delta\mathbf{x}_n)$ and define the following uncertainty set:

$$\mathcal{U}_x = \{\Delta\mathbf{X} \in \mathbb{R}^{n\times p} \mid \|\Delta\mathbf{x}_i\|_q \leq \rho, i = 1, \ldots, n\}, \tag{8}$$

where ρ is a parameter controlling the magnitude of the considered perturbations and, hence, the degree to which the features in the training set are able to deviate from their nominal values.

After introducing these perturbations, the features in the training set take values $\mathbf{x}_i + \Delta\mathbf{x}_i$, $i = 1, \ldots, n$. We now seek to construct a classifier that is robust to all such perturbations $\Delta\mathbf{X} \in \mathcal{U}_x$. To do this, we robustify against this uncertainty set of feature parameters in each of our classification methods. In practice, the parameter ρ can be chosen via validation, and the range to be searched over can be fixed if each feature in the data set is normalized. We also note that when $\rho = 0$, the problem is equivalent to the nominal problem, and so the nominal solution is a possible candidate to be considered during validation. This means that the feature-robust classifier will only be preferred over the nominal method when the validation score is better.

In addition, note that $\mathcal{U}_x$ is the Cartesian product of the sets $\{\boldsymbol{\Delta}\mathbf{x}_i \in \mathbb{R}^p \,|\, \|\boldsymbol{\Delta}\mathbf{x}_i\|_q \le \rho\}, i = 1, \ldots, n$. This structure enables us to derive tractable robust counterparts for all three classification methods. We may consider alternative uncertainty sets for the feature perturbations as well—for example, polyhedral or ellipsoidal uncertainty sets. Here, we consider the norm uncertainty set $\mathcal{U}_x$ because it admits a simple geometric interpretation and only requires tuning a single parameter ρ, which makes it tractable to evaluate in the computational experiments and to use in practice.

We present the reformulated robust counterparts below for soft-margin support vector machines, logistic regression, and optimal decision trees. For each method, we refer to the resulting deterministic optimization problem as the *feature-robust counterpart* of that classifier.

4.2. Soft-Margin Support Vector Machines

The regularized support vector machine problem in (2) has been shown by Xu et al. (2009) and Fertis (2009) to be equivalent to the robust counterpart of a nominal problem under a particular choice of uncertainty set in the features. These results suggest that the regularization term $\|\mathbf{w}\|_2^2$ is a by-product of feature robustness. Further discussion of the equivalence between classical SVM and feature-robust formulations is provided in Appendix A. In the following sections, to avoid double counting the effect of robustness, we consider the hinge loss classifier without the regularization term to be the nominal method for SVM:

$$\min_{\mathbf{w},b} \sum_{i=1}^{n} \max\{1 - y_i(\mathbf{w}^T\mathbf{x}_i - b), 0\}. \tag{9}$$

Robustifying Problem (9) against the uncertainty set $\mathcal{U}_x$ gives the following robust optimization problem:

$$\min_{\mathbf{w},b} \max_{\Delta\mathbf{X}\in\mathcal{U}_x} \sum_{i=1}^{n} \max\{1 - y_i(\mathbf{w}^T(\mathbf{x}_i + \boldsymbol{\Delta}\mathbf{x}_i) - b), 0\}. \tag{10}$$

We now derive the robust counterpart to Problem (10). Note that this is equivalent to theorem 3 in Xu et al. (2009).

Theorem 1. *The robust counterpart to Problem* (10) *is*

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \sum_{i=1}^{n} \xi_i \\ \text{s.t.} \quad & y_i(\mathbf{w}^T\mathbf{x}_i - b) - \rho\|\mathbf{w}\|_{q^*} \ge 1 - \xi_i && i = 1, \ldots, n, \\ & \xi_i \ge 0 && i = 1, \ldots, n. \end{aligned} \tag{11}$$

where ℓ_{q^*} is the dual norm of ℓ_q.

Proof. We can reformulate Problem (10) as

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \sum_{i=1}^{n} \xi_i \\ \text{s.t.} \quad & y_i(\mathbf{w}^T(\mathbf{x}_i + \boldsymbol{\Delta}\mathbf{x}_i) - b) \ge 1 - \xi_i \qquad \forall \boldsymbol{\Delta}\mathbf{X} \in \mathcal{U}_x && i = 1, \ldots, n, \\ & \xi_i \ge 0 && i = 1, \ldots, n. \end{aligned}$$

The first constraint must be satisfied for all $\boldsymbol{\Delta}\mathbf{X} \in \mathcal{U}_x$, thus the constraint is equivalent to

$$\min_{\boldsymbol{\Delta}\mathbf{X}\in\mathcal{U}_x} (y_i\mathbf{w}^T\boldsymbol{\Delta}\mathbf{x}_i) \ge 1 - \xi_i - y_i(\mathbf{w}^T\mathbf{x}_i - b) \quad i = 1, \ldots, n.$$

Here, for all $i = 1, \ldots, n$, the minimization term is equal to the objective value of the following optimization problem:

$$\begin{aligned} \min_{\boldsymbol{\Delta}\mathbf{x}_i} \quad & y_i\mathbf{w}^T\boldsymbol{\Delta}\mathbf{x}_i \\ \text{s.t.} \quad & \|\boldsymbol{\Delta}\mathbf{x}_i\|_q \le \rho. \end{aligned}$$

Because y_i is constant, we recognize this optimization problem as the dual-norm problem. Therefore, by (7), for any given value of $\mathbf{w}$, the objective value of this problem is $-\rho\|\mathbf{w}\|_{q^*}$, where ℓ_{q^*} is the dual norm of ℓ_q. Replacing the minimization term with this optimal value and rearranging yields (11). □

Depending on the choice of norm, the feature-robust counterpart of SVM can be solved efficiently by using various optimization methods. For example, when $q = q^* = 2$, feature-robust SVM can be solved by using second-order cone-optimization methods (Bertsekas 1999). When $q = 1$, $q^* = \infty$ or $q = \infty$, $q^* = 1$, feature-robust SVM can be reformulated as a linear optimization problem.

4.3. Logistic Regression

Robustifying Problem (4) against the uncertainty set $\mathcal{U}_x$ yields the following robust optimization problem:

$$\max_{\boldsymbol{\beta},\beta_0} \ \min_{\Delta\mathbf{X}\in\mathcal{U}_x} -\sum_{i=1}^{n} \log\left(1 + e^{-y_i\left(\boldsymbol{\beta}^T(\mathbf{x}_i + \Delta\mathbf{x}_i) + \beta_0\right)}\right). \tag{12}$$

Next, we determine the robust counterpart to Problem (12). We note that similar results on more specific uncertainty sets have been previously shown in El Ghaoui et al. (2003) and Harrington et al. (2010).

Theorem 2. *The robust counterpart to Problem* (12) *is*

$$\max_{\boldsymbol{\beta},\beta_0} -\sum_{i=1}^{n} \log\left(1 + e^{-y_i\left(\boldsymbol{\beta}^T\mathbf{x}_i + \beta_0\right) + \rho\|\boldsymbol{\beta}\|_{q^*}}\right), \tag{13}$$

where ℓ_{q^*} is the dual norm of ℓ_q.

Proof. Consider the inner minimization problem in (12), which is the following optimization problem:

$$\min_{\Delta\mathbf{X}\in\mathcal{U}_x} -\sum_{i=1}^{n} \log\left(1 + e^{-y_i\left(\boldsymbol{\beta}^T(\mathbf{x}_i + \Delta\mathbf{x}_i) + \beta_0\right)}\right). \tag{14}$$

Let $\omega_i = y_i(\boldsymbol{\beta}^T(\mathbf{x}_i + \Delta\mathbf{x}_i) + \beta_0)$, and define $g(\omega_i) = -\log(1 + e^{-\omega_i})$. The first-order derivative of g with respect to ω_i is

$$\frac{dg}{d\omega_i} = \frac{1}{1 + e^{\omega_i}},$$

which is strictly positive. Therefore, for each $i = 1, \ldots, n$, the solution to the inner minimization problem in (12) is the same as the solution of the problem

$$\begin{aligned} \min_{\Delta\mathbf{x}_i} \quad & y_i(\boldsymbol{\beta}^T(\mathbf{x}_i + \Delta\mathbf{x}_i) + \beta_0) \\ \text{s.t.} \quad & \|\Delta\mathbf{x}_i\|_q \le \rho. \end{aligned} \tag{15}$$

This is equivalent to the following problem:

$$\begin{aligned} y_i(\boldsymbol{\beta}^T\mathbf{x}_i + \beta_0) - \max_{\Delta\mathbf{x}_i} \quad & -y_i\boldsymbol{\beta}^T\Delta\mathbf{x}_i \\ \text{s.t.} \quad & \|\Delta\mathbf{x}_i\|_q \le \rho. \end{aligned}$$

We recognize this maximization term as the dual-norm problem. Therefore, by (7), the optimal solution is $\rho\|\boldsymbol{\beta}\|_{q^*}$, where ℓ_{q^*} is the dual norm of ℓ_q. We conclude that the optimal value to (15) is $y_i(\boldsymbol{\beta}^T\mathbf{x}_i + \beta_0) - \rho\|\boldsymbol{\beta}\|_{q^*}$. Substituting the optimal value into the inner minimization problem in (12), we obtain

$$-\sum_{i=1}^{n} \log\left(1 + e^{-y_i\left(\boldsymbol{\beta}^T\mathbf{x}_i + \beta_0\right) + \rho\|\boldsymbol{\beta}\|_{q^*}}\right).$$

Maximizing the above equation over $\boldsymbol{\beta}, \beta_0$ yields (13). □

If $q \ge 2$, the robust counterpart (13) is differentiable (as in the nominal problem) and thus is still solvable by using gradient and Newton methods. However, if $q \in \{1, \infty\}$, then Problem (13) becomes nondifferentiable, and we may solve it using subgradient methods. Alternatively, we may remodel the nonlinear terms to obtain a differentiable formulation with linear constraints, which is solvable by using gradient and Newton methods for constrained optimization (Bertsekas 1999).

Compared to the nominal case, the feature-robust counterpart of logistic regression has an additional $\rho\|\boldsymbol{\beta}\|_{q^*}$ term in the exponent of the logit function. It resembles the regularization term in regularized logistic regression, shown in Equation (5). However, the additional term from robustness penalizes model complexity in the logit, or log odds ratio, whereas the regularization term is a linear penalty on the entire likelihood. The connection between the two can be shown via a first-order Taylor series expansion of the objective function of the feature-robust counterpart, which gives the following:

$$-\sum_{i=1}^{n} \log\left(1 + e^{-y_i\left(\boldsymbol{\beta}^T \mathbf{x}_i + \beta_0\right)}\right) - \sum_{i=1}^{n} \frac{e^{-y_i\left(\boldsymbol{\beta}^T \mathbf{x}_i + \beta_0\right)}}{1 + e^{-y_i\left(\boldsymbol{\beta}^T \mathbf{x}_i + \beta_0\right)}} \rho \|\boldsymbol{\beta}\|_{q^*}.$$

In cases where $\rho\|\boldsymbol{\beta}\|_{q^*}$ is small and its coefficient is close to 1, robustification over features and regularization of logistic regression are approximately equivalent.

4.4. Optimal Decision Trees

Robustifying Problem (6) against the uncertainty set $\mathcal{U}_x$ gives a problem identical to Problem (6) except with the following constraints in place of the constraints (6p) and (6q):

$$\mathbf{a}_j^T(\mathbf{x}_i + \Delta\mathbf{x}_i) + \epsilon \leq b_j + M(1 - z_{ik}) \quad \forall \Delta\mathbf{X} \in \mathcal{U}_x, i = 1, \ldots, n, k = 1, \ldots, K, \forall j \in \mathcal{P}_k^l, \tag{16a}$$

$$\mathbf{a}_j^T(\mathbf{x}_i + \Delta\mathbf{x}_i) \geq b_j + M(1 - z_{ik}) \quad \forall \Delta\mathbf{X} \in \mathcal{U}_x, i = 1, \ldots, n, k = 1, \ldots, K, \forall j \in \mathcal{P}_k^u. \tag{16b}$$

We refer to this optimization problem as Problem (16).

Theorem 3. *The robust counterpart to Problem* (16) *is identical to Problem* (16) *except with the following constraints in place of constraints* (16a) *and* (16b):

$$\mathbf{a}_j^T \mathbf{x}_i + \rho + \epsilon \leq b_j + (1 - z_{ik}) \quad i = 1, \ldots, n,\ k = 1, \ldots, K,\ \forall j \in \mathcal{P}_k^l, \tag{17a}$$

$$\mathbf{a}_j^T \mathbf{x}_i - \rho \geq b_j + (1 - z_{ik}) \quad i = 1, \ldots, n,\ k = 1, \ldots, K,\ \forall j \in \mathcal{P}_k^l. \tag{17b}$$

Proof. Because constraint (16a) must hold for all $\Delta\mathbf{X} \in \mathcal{U}_x$, this constraint is equivalent to

$$\max_{\Delta\mathbf{X} \in \mathcal{U}_x} \{\mathbf{a}_j^T \Delta\mathbf{x}_i\} \leq b_j + M(1 - z_{ik}) - \mathbf{a}_j^T \mathbf{x}_i - \epsilon \;\; i = 1, \ldots, n,\ k = 1, \ldots, K,\ \forall j \in \mathcal{P}_k^l.$$

This maximization term is equal to the optimal value of the following problem:

$$\begin{aligned} \max \quad & \mathbf{a}_j^T \Delta\mathbf{x}_i \\ \text{s.t.} \quad & \|\Delta\mathbf{x}_i\|_q \leq \rho. \end{aligned}$$

We recognize this as the dual-norm problem, and by (7), the optimal value is $\rho\|\mathbf{a}_j\|_{q^*}$, where ℓ_{q^*} is the dual norm of ℓ_q. Moreover, if this constraint is to be nontrivial (which requires $z_{ik} = 1$), we know from (6m) that $d_j = 0$ for all ancestors $j \in \mathcal{P}_k^l$. Thus, from (6j) we have that $\sum_l \mathbf{a}_{jl} = 1$, and so together with (6r), we know that a single element of $\mathbf{a}_j$ is 1, with all other elements being 0. This means that $\|\mathbf{a}_j\|_{q^*} = 1$ for any q, so the value of the maximization term is simply ρ. Rearranging terms yields the constraint (17a). We use an identical approach to yield (17b) from (16b). □

This remains a linear mixed-integer optimization problem regardless of the original choice of q. The only difference compared to the nominal problem is the introduction of a margin of size ρ around each b_j. The problem is therefore practically solvable like the nominal problem.

5. Robustness Against Uncertainty in Labels

In this section, we introduce the notion of robustifying classification methods against uncertainties in the labels of the training set. We consider a discrete uncertainty set that limits the number of incorrect labels to be less than or equal to a fixed number Γ. We then define and state the *label-robust counterpart* for each of the classification methods.

5.1. Motivating Label Robustness

Uncertainties in data labels can occur naturally from errors in manual entries, self-reporting, or nonexact, nonobjective label definition. To model uncertainty in the labels of the training set, we consider a scenario where some number of the supplied labels is incorrect. We introduce variables $\Delta y_i \in \{0, 1\}$, where 1 indicates that the label was incorrect and has in fact been flipped, and 0 indicates that the label was correct. We consider the following uncertainty set:

$$\mathcal{U}_y = \left\{ \mathbf{\Delta y} \in \{0,1\}^n \,\middle|\, \sum_{i=1}^n \Delta y_i \le \Gamma \right\},$$

where Γ is an integer-valued parameter controlling the number of data points that we allow to be mislabeled. Observe that in contrast to the uncertainty set over the features, $\mathcal{U}_y$ cannot be decomposed as the Cartesian product of smaller uncertainty sets.

We can then model the true labels of the training set as $y_i(1 - 2\Delta y_i), i = 1, \ldots, n$. Applying robust optimization, we modify the training process so that our classifier is optimized against the worst-case realization $\mathbf{\Delta y} \in \mathcal{U}_y$ to obtain a classifier that is *label-robust*. In practice, the parameter Γ, which determines the size of our uncertainty set, is often modeled as a proportion of the total number of data points and can be chosen via validation. Note that when $\Gamma = 0$, the problem is the same as the nominal problem. In this sense, our validation can include the nominal case, so the best label-robust solution will only be preferred over the nominal case if it leads to an improvement in accuracy in validation.

As in Section 4, we present the reformulated robust counterparts below for logistic regression, SVM, and optimal trees. For each method, we refer to the resulting deterministic optimization problem as the *label-robust counterpart* of that classifier.

5.2. Soft-Margin Support Vector Machines

Robustifying Problem (2) against the uncertainty set $\mathcal{U}_y$ gives

$$\min_{\mathbf{w},b} \max_{\mathbf{\Delta y} \in \mathcal{U}_y} \sum_{i=1}^n \max\{1 - y_i(1 - 2\Delta y_i)(\mathbf{w}^T\mathbf{x}_i - b), 0\}. \tag{18}$$

Theorem 4. *The robust counterpart to Problem* (18) *is*

$$\begin{array}{lll}
\min & \displaystyle\sum_{i=1}^n \xi_i + \Gamma q + \sum_{i=1}^n r_i & \\
\text{s.t.} & q + r_i \ge \phi_i - \xi_i & i = 1, \ldots, n, \\
& \xi_i \ge 1 - y_i(\mathbf{w}^T\mathbf{x}_i - b) & i = 1, \ldots, n, \\
& \xi_i \le 1 - y_i(\mathbf{w}^T\mathbf{x}_i - b) + M(1 - s_i) & i = 1, \ldots, n, \\
& \xi_i \le M s_i & i = 1, \ldots, n, \\
& \phi_i \ge 1 + y_i(\mathbf{w}^T\mathbf{x}_i - b) & i = 1, \ldots, n, \\
& \phi_i \le 1 + y_i(\mathbf{w}^T\mathbf{x}_i - b) + M(1 - t_i) & i = 1, \ldots, n, \\
& \phi_i \le M t_i & i = 1, \ldots, n, \\
& r_i, \xi_i, \phi_i \ge 0 & i = 1, \ldots, n, \\
& q \ge 0, & \\
& \mathbf{s}, \mathbf{t} \in \{0,1\}^n. &
\end{array} \tag{19}$$

where M is a sufficiently large constant.

Proof. Fix $\mathbf{w}$ and b, and consider the inner maximization problem

$$\max_{\mathbf{\Delta y} \in \mathcal{U}_y} \sum_{i=1}^n \max\{1 - y_i(1 - 2\Delta y_i)(\mathbf{w}^T\mathbf{x}_i - b), 0\} \quad i = 1, \ldots, n. \tag{20}$$

Define the functions

$$f_i(\Delta y_i) = \max\{1 - y_i(1 - 2\Delta y_i)(\mathbf{w}^T\mathbf{x}_i - b), 0\}, \quad i = 1, \ldots, n.$$

Because $\Delta y_i \in \{0,1\}$ for all i, we observe

$$f_i(\Delta y_i) = [f_i(1) - f_i(0)]\Delta y_i + f_i(0) \quad i = 1, \ldots, n.$$

Let $\phi_i = f_i(1)$ and $\xi_i = f_i(0)$ for $i = 1, \ldots, n$. It follows that Problem (20) is equivalent to

$$\begin{array}{ll}
\max & \displaystyle\sum_{i=1}^n (\phi_i - \xi_i)\Delta y_i + \xi_i \\
\text{s.t.} & \mathbf{\Delta y} \in \mathcal{U}_y.
\end{array}$$

Next, consider the following polyhedron, which is the convex hull of $\mathcal{U}_y$:

$$\mathcal{P}_y = \left\{ \Delta\mathbf{y} \in \mathbb{R}^n \;\middle|\; 0 \le \Delta y_i \le 1, \sum_{i=1}^n \Delta y_i \le \Gamma \right\}.$$

Because the polyhedron $\mathcal{P}_y$ has integer extreme points, this problem is equivalent to its linear relaxation

$$\begin{aligned} \max \quad & \sum_{i=1}^n (\phi_i - \xi_i)\Delta y_i + \xi_i \\ \text{s.t.} \quad & 0 \le \Delta y_i \le 1 && i = 1, \ldots, n, \\ & \sum_{i=1}^n \Delta y_i \le \Gamma. \end{aligned}$$

By strong duality, this has the same objective value as its dual problem

$$\begin{aligned} \min \quad & \Gamma q + \sum_{i=1}^n r_i + \sum_{i=1}^n \xi_i \\ \text{s.t.} \quad & q + r_i \ge \phi_i - \xi_i && i = 1, \ldots, n, \\ & r_i \ge 0 && i = 1, \ldots, n, \\ & q \ge 0. \end{aligned}$$

Minimizing over $\mathbf{w}$ and b, this optimization problem becomes

$$\begin{aligned} \min \quad & \sum_{i=1}^n \xi_i + \Gamma q + \sum_{i=1}^n r_i \\ \text{s.t.} \quad & q + r_i \ge \phi_i - \xi_i && i = 1, \ldots, n, \\ & \xi_i = \max\{1 - y_i(\mathbf{w}^T\mathbf{x}_i - b), 0\} && i = 1, \ldots, n, \\ & \phi_i = \max\{1 + y_i(\mathbf{w}^T\mathbf{x}_i - b), 0\} && i = 1, \ldots, n, \\ & r_i \ge 0 && i = 1, \ldots, n, \\ & q \ge 0. \end{aligned}$$

Reformulating the problem to specify the values of the variables ξ_i, ϕ_i with linear constraints yields the desired result. □

Problem (19) is a mixed-integer optimization problem that is practically solvable.

5.3. Logistic Regression

Robustifying Problem (4) against the uncertainty set $\mathcal{U}_y$ gives

$$\max_{\beta,\beta_0} \min_{\Delta\mathbf{y}\in\mathcal{U}_y} -\sum_{i=1}^n \log\left(1 + e^{-y_i(1-2\Delta y_i)(\beta^T\mathbf{x}_i+\beta_0)}\right). \tag{21}$$

Theorem 5. *The robust counterpart to Problem* (21) *is*

$$\begin{aligned} \max_{\beta,\beta_0} \quad & -\sum_{i=1}^n \log\left(1 + e^{-y_i(\beta^T\mathbf{x}_i+\beta_0)}\right) + \Gamma\mu + \sum_{i=1}^n \nu_i \\ \text{s.t.} \quad & \mu + \nu_i \le \log\left(\frac{1 + e^{-y_i(\beta^T\mathbf{x}_i+\beta_0)}}{1 + e^{y_i(\beta^T\mathbf{x}_i+\beta_0)}}\right) && i = 1, \ldots, n, \\ & \nu_i \le 0 && i = 1, \ldots, n, \\ & \mu \le 0. \end{aligned} \tag{22}$$

Proof. Define the functions $f_i(\Delta y_i) = -\log\left(1 + e^{-y_i(1-2\Delta y_i)(\beta^T \mathbf{x}_i + \beta_0)}\right)$ for $i = 1, \ldots, n$. Because $\Delta y_i \in \{0, 1\}$, we can express $f_i(\Delta y_i)$ as

$$\begin{aligned} f_i(\Delta y_i) &= [f(1) - f(0)]\Delta y_i + f(0) \\ &= \log\left(\frac{1 + e^{-y_i(\beta^T \mathbf{x}_i + \beta_0)}}{1 + e^{y_i(\beta^T \mathbf{x}_i + \beta_0)}}\right)\Delta y_i - \log\left(1 + e^{-y_i(\beta^T \mathbf{x}_i + \beta_0)}\right). \end{aligned}$$

We can thus rewrite the inner minimization part of Problem (21) as

$$\min_{\Delta \mathbf{y} \in \mathcal{U}_y} \sum_{i=1}^n \left[\log\left(\frac{1 + e^{-y_i(\beta^T \mathbf{x}_i + \beta_0)}}{1 + e^{y_i(\beta^T \mathbf{x}_i + \beta_0)}}\right)\Delta y_i - \log\left(1 + e^{-y_i(\beta^T \mathbf{x}_i + \beta_0)}\right)\right]. \tag{23}$$

Because the convex hull of $\mathcal{U}_y$ has integer extreme points, Problem (23) has the same objective as its linear optimization relaxation (Bertsimas and Tsitsiklis 2008)

$$\begin{aligned} \min_{\Delta \mathbf{y}} \quad & \sum_{i=1}^n \left[\log\left(\frac{1 + e^{-y_i(\beta^T \mathbf{x}_i + \beta_0)}}{1 + e^{y_i(\beta^T \mathbf{x}_i + \beta_0)}}\right)\Delta y_i - \log\left(1 + e^{-y_i(\beta^T \mathbf{x}_i + \beta_0)}\right)\right] \\ \text{s.t.} \quad & 0 \le \Delta y_i \le 1 \qquad i = 1, \ldots, n, \\ & \sum_{i=1}^n \Delta y_i \le \Gamma. \end{aligned} \tag{24}$$

By strong duality, the optimal value to Problem (24) is equal to that of its dual problem

$$\begin{aligned} \max \quad & -\sum_{i=1}^n \log\left(1 + e^{-y_i(\beta^T \mathbf{x}_i + \beta_0)}\right) + \Gamma\mu + \sum_{i=1}^n \nu_i \\ \text{s.t.} \quad & \mu + \nu_i \le \log\left(\frac{1 + e^{-y_i(\beta^T \mathbf{x}_i + \beta_0)}}{1 + e^{y_i(\beta^T \mathbf{x}_i + \beta_0)}}\right) \qquad i = 1, \ldots, n, \\ & \nu_i \le 0 \qquad i = 1, \ldots, n, \\ & \mu \le 0. \end{aligned}$$

Substituting this back into Problem (21) in place of the inner minimization, it becomes a single maximization problem, giving the stated result. □

This problem has a twice continuously differentiable concave objective function and constraints, making it tractably solvable with an interior point method (Bertsekas 1999).

5.4. Optimal Decision Trees

Robustifying Problem (6a) against the uncertainty set $\mathcal{U}_y$ gives a problem identical to Problem (6) with the following constraints in place of constraints (6b)–(6g):

$$g_k = \sum_{i=1}^n \frac{1 - y_i(1 - 2\Delta y_i)}{2} z_{ik} \qquad k = 1, \ldots, K, \tag{25a}$$

$$h_k = \sum_{i=1}^n \frac{1 + y_i(1 - 2\Delta y_i)}{2} z_{ik} \qquad k = 1, \ldots, K, \tag{25b}$$

$$f_k \le g_k + M[w_k + (1 - c_k)] \qquad \forall \Delta \mathbf{y} \in \mathcal{U}_y,\ k = 1, \ldots, K, \tag{25c}$$

$$f_k \le h_k + M[(1 - w_k) + (1 - c_k)] \qquad \forall \Delta \mathbf{y} \in \mathcal{U}_y,\ k = 1, \ldots, K, \tag{25d}$$

$$f_k \ge g_k - M[(1 - w_k) + (1 - c_k)] \qquad \forall \Delta \mathbf{y} \in \mathcal{U}_y,\ k = 1, \ldots, K, \tag{25e}$$

$$f_k \ge h_k - M[w_k + (1 - c_k)] \qquad \forall \Delta \mathbf{y} \in \mathcal{U}_y,\ k = 1, \ldots, K. \tag{25f}$$

We refer to this optimization problem as Problem (25).

Theorem 6. *The robust counterpart to Problem* (25) *is identical to Problem* (25) *with the following constraints in place of constraints* (25a)–(25f):

$$g_k = \sum_{i=1}^n \frac{1-y_i}{2} z_{ik} \qquad k = 1, \dots, K, \tag{26a}$$

$$h_k = \sum_{i=1}^n \frac{1+y_i}{2} z_{ik} \qquad k = 1, \dots, K, \tag{26b}$$

$$f_k \le g_k - \Gamma\mu_{1,k} - \sum_{i=1}^n \nu_{1,ik} + M[w_k + (1-c_k)] \qquad k = 1, \dots, K, \tag{26c}$$

$$f_k \le h_k - \Gamma\mu_{2,k} - \sum_{i=1}^n \nu_{2,ik} + M[(1-w_k) + (1-c_k)] \qquad k = 1, \dots, K, \tag{26d}$$

$$f_k \ge g_k + \Gamma\mu_{3,k} + \sum_{i=1}^n \nu_{3,ik} - M[(1-w_k) + (1-c_k)] \qquad k = 1, \dots, K, \tag{26e}$$

$$f_k \ge h_k + \Gamma\mu_{4,k} + \sum_{i=1}^n \nu_{4,ik} - M[w_k + (1-c_k)] \qquad k = 1, \dots, K, \tag{26f}$$

$$\mu_{m,k} + \nu_{m,ik} \ge -y_i z_{ik} \qquad i = 1, \dots, n,\ k = 1, \dots, K,\ m = 1, 4, \tag{26g}$$

$$\mu_{m,k} + \nu_{m,ik} \ge y_i z_{ik} \qquad i = 1, \dots, n,\ k = 1, \dots, K,\ m = 2, 3, \tag{26h}$$

$$\mu_{m,k}, \nu_{m,ik} \ge 0 \qquad i = 1, \dots, n,\ k = 1, \dots, K,\ m = 1, \dots, 4. \tag{26i}$$

Proof. We can substitute (25a) into constraint (25c) to obtain

$$\sum_{i=1}^n \frac{1 - y_i(1-2\Delta y_i)}{2} z_{ik} \ge f_k - M[w_k + (1-c_k)] \qquad \forall \mathbf{\Delta y} \in \mathcal{U}_y,\ k = 1, \dots, K,$$

$$\sum_{i=1}^n \frac{1-y_i}{2} z_{ik} + \sum_{i=1}^n y_i z_{ik} \Delta y_i \ge f_k - M[w_k + (1-c_k)] \qquad \forall \mathbf{\Delta y} \in \mathcal{U}_y,\ k = 1, \dots, K.$$

Because this must hold for all $\mathbf{\Delta y} \in \mathcal{U}_y$, this is equivalent to the following constraint:

$$\sum_{i=1}^n \frac{1-y_i}{2} z_{ik} + \min_{\mathbf{\Delta y} \in \mathcal{U}_y} \left\{ \sum_{i=1}^n y_i z_{ik} \Delta y_i \right\} \ge f_k - M[w_k + (1-c_k)] \ \ k = 1, \dots, K.$$

The convex hull of $\mathcal{U}_y$ has integer extreme points, so the value of the minimization term is equivalent to the optimal value of its linear relaxation (for any fixed k)

$$\begin{aligned} \min \quad & \sum_{i=1}^n y_i z_{ik} \Delta y_i \\ \text{s.t.} \quad & 0 \le \Delta y_i \le 1 \qquad i = 1, \dots, n, \\ & \sum_{i=1}^n \Delta y_i \le \Gamma. \end{aligned}$$

By strong duality, this problem has the same optimal objective value as its dual

$$\begin{aligned} \max \quad & \Gamma\mu_{1,k} + \sum_{i=1}^n \nu_{1,ik} \\ \text{s.t.} \quad & \mu_{1,k} + \nu_{1,ik} \le y_i z_{ik} \qquad i = 1, \dots, n, \\ & \mu_{1,k}, \nu_{1,ik} \le 0 \qquad i = 1, \dots, n. \end{aligned}$$

Substituting this back into the original constraint gives

$$\begin{aligned}
&\sum_{i=1}^{n} \frac{1-y_i}{2} z_{ik} + \Gamma\mu_{1,k} + \sum_{i=1}^{n} \nu_{1,ik} \geq f_k - M[w_k + (1-c_k)] && k = 1,\ldots,K,\\
&\mu_{1,k} + \nu_{1,ik} \leq y_i z_{ik} && i = 1,\ldots,n,\\
&\mu_{1,k}, \nu_{1,ik} \leq 0 && i = 1,\ldots,n.
\end{aligned}$$

We substitute back for the original definition of g_k from (6b) and change the signs of μ and ν to get

$$\begin{aligned}
&g_k - \Gamma\mu_{1,k} - \sum_{i=1}^{n} \nu_{1,ik} \geq f_k - M[w_k + (1-c_k)] && k = 1,\ldots,K,\\
&\mu_{1,k} + \nu_{1,ik} \geq -y_i z_{ik} && i = 1,\ldots,n,\\
&\mu_{1,k}, \nu_{1,ik} \geq 0 && i = 1,\ldots,n.
\end{aligned}$$

We can rearrange this to obtain constraint (26c), as well as parts of constraints (26g) and (26i).

We repeat this entire process identically for constraints (25d)–(25f) to achieve the stated result. □

Similar to before, this remains a linear mixed-integer optimization problem, and so is practically solvable. The label robustification for optimal decision trees also has a simple geometric interpretation. Recall that in the model, g_k is the number of points in node k with label $y_i = +1$, h_k is the number of points in node k with label $y_i = -1$, and f_k is the number of points in node k that are misclassified, which in the nominal case is simply $\min\{g_k, h_k\}$. In the label-robust counterpart, the extra terms in these constraints require feasible solutions to have strict separation between f_k, g_k, and h_k. Indeed, we can obtain a feasible solution by setting $\mu_{m,k} = 1$ and $\nu_{m,ik} = 0$, which then requires $|g_k - h_k| \geq 2\Gamma$, and $f_k = \min\{g_k, h_k\} + \Gamma$. This means that a feasible label-robust solution requires the majority class in each node to be a strict majority, and the size of this required separation is controlled by the robustness parameter Γ. Increasing Γ has the effect of increasing the *label purity* of all nodes in the tree, because trees that do not have the required margin between g_k and h_k at every node k in the tree are treated as being infeasible for the label-robust problem.

6. Robustness in Both Features and Labels

In this section, we consider applying the methods of Sections 4 and 5 simultaneously to construct a new family of classifiers that are robust to uncertainty in both features and labels. We will refer to this family as *robust-in-both* classifiers. To develop these classifiers, we simply expose the classification problem to both feature uncertainty with uncertainty set $\mathcal{U}_x$, and label uncertainty with uncertainty set $\mathcal{U}_y$. This is a natural extension of our previous methods to handle classification problems that may have errors in both the features and labels of the training data. For example, in the contraceptive-method-choice data set considered in Section 5, survey data are used to obtain information on both the features (demographic and socioeconomic characteristics) and labels (contraceptive method choice), and both factors may be influenced by inaccurate reporting.

We present the reformulated robust counterparts below for soft-margin support vector machines, logistic regression, and optimal decisions trees, which we refer to as the *robust-in-both counterpart* for each method. The proofs are similar to the derivations of the robust counterparts in the previous two sections and are included in Appendix B.

Like both methods individually, the robust-in-both classifier has to select the robustness parameters ρ and Γ through validation. As per the individual cases, when we set $\rho = \Gamma = 0$, the problem reduces to the nominal problem. Note also that if only one of ρ/Γ is zero, the problem reduces to the label-robust/feature-robust problem respectively. This means that as part of the robust-in-both validation process, we consider the models from the nominal, feature-robust and label-robust classifiers in addition to the robust-in-both classifier and then select the classifier among these with the best validation accuracy. In this sense, the robust-in-both classifier is the strongest of all the robust classifiers, because it selects in validation the best-performing robust classifier of all those we have considered.

6.1. Soft-Margin Support Vector Machines

Robustifying Problem (1) against both $\mathcal{U}_x$ and $\mathcal{U}_y$ gives the following robust optimization problem:

$$\min_{\mathbf{w},b} \max_{\Delta\mathbf{y}\in\mathcal{U}_y} \max_{\Delta\mathbf{X}\in\mathcal{U}_x} \sum_{i=1}^{n} \max\{1 - y_i(1-2\Delta y_i)(\mathbf{w}^T(\mathbf{x}_i + \Delta\mathbf{x}_i) - b), 0\}. \tag{27}$$

Theorem 7. *The robust counterpart to Problem* (27) *is*

$$
\begin{array}{lll}
\min & \displaystyle\sum_{i=1}^{n} \xi_i + \Gamma q + \sum_{i=1}^{n} r_i & \\
\text{s.t.} & q + r_i \geq \phi_i - \xi_i & i = 1, \ldots, n, \\
& \xi_i \geq 1 - y_i(\mathbf{w}^T \mathbf{x}_i - b) + \rho \|\mathbf{w}\|_{q^*} & i = 1, \ldots, n, \\
& \xi_i \leq 1 - y_i(\mathbf{w}^T \mathbf{x}_i - b) + \rho \|\mathbf{w}\|_{q^*} + M(1 - s_i) & i = 1, \ldots, n, \\
& \xi_i \leq M s_i & i = 1, \ldots, n, \\
& \phi_i \geq 1 + y_i(\mathbf{w}^T \mathbf{x}_i - b) + \rho \|\mathbf{w}\|_{q^*} & i = 1, \ldots, n, \\
& \phi_i \leq 1 + y_i(\mathbf{w}^T \mathbf{x}_i - b) + \rho \|\mathbf{w}\|_{q^*} + M(1 - t_i) & i = 1, \ldots, n, \\
& \phi_i \leq M t_i & i = 1, \ldots, n, \\
& r_i, \xi_i, \phi_i \geq 0 & i = 1, \ldots, n, \\
& q \geq 0, & \\
& \mathbf{s}, \mathbf{t} \in \{0, 1\}^n. &
\end{array}
\tag{28}
$$

where ℓ_{q^*} is the dual norm of ℓ_q, and M is a sufficiently large constant.

The proof of Theorem 7 is straightforward, and it is provided in Appendix B.

Problem (28) is a mixed-integer optimization problem that is practically solvable.

6.2. Logistic Regression

Robustifying Problem (4) against both $\mathcal{U}_x$ and $\mathcal{U}_y$ gives the following robust optimization problem:

$$
\max_{\boldsymbol{\beta}, \beta_0} \min_{\Delta\mathbf{y} \in \mathcal{U}_y} \min_{\Delta\mathbf{X} \in \mathcal{U}_x} -\sum_{i=1}^{n} \log\left(1 + e^{-y_i(1 - 2\Delta y_i)(\boldsymbol{\beta}^T(\mathbf{x}_i + \Delta\mathbf{x}_i) + \beta_0)}\right). \tag{29}
$$

Theorem 8. *The robust counterpart to Problem* (29) *is*

$$
\begin{array}{lll}
\max & \displaystyle -\sum_{i=1}^{n} \log\left(1 + e^{-y_i(\boldsymbol{\beta}^T\mathbf{x}_i + \beta_0) + \rho\|\boldsymbol{\beta}\|_{q^*}}\right) + \Gamma\mu + \sum_{i=1}^{n} \nu_i & \\
\text{s.t.} & \mu + \nu_i \leq \log\left(\dfrac{1 + e^{-y_i(\boldsymbol{\beta}^T\mathbf{x}_i + \beta_0) + \rho\|\boldsymbol{\beta}\|_{q^*}}}{1 + e^{y_i(\boldsymbol{\beta}^T\mathbf{x}_i + \beta_0) + \rho\|\boldsymbol{\beta}\|_{q^*}}}\right) & i = 1, \ldots, n, \\
& \nu_i \leq 0 & i = 1, \ldots, n, \\
& \mu \leq 0. &
\end{array}
\tag{30}
$$

where ℓ_{q^*} is the dual norm of ℓ_q.

The proof of Theorem 8 can be found in Appendix B. It essentially applies the process in the proof for feature-robust logistic regression, followed by the process in the proof for label robustness to obtain the final robust counterpart.

Problem (30) is a maximization of a concave, twice continuously differentiable function in $\boldsymbol{\beta}$ and β_0 with constraints for any given ρ and Γ. Therefore, we can solve this problem using interior point methods (Bertsekas 1999).

6.3. Optimal Decision Trees

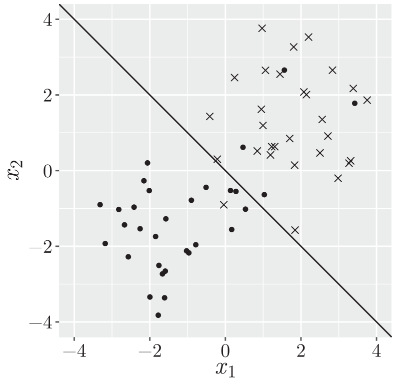
Robustifying Problem (6) against both $\mathcal{U}_x$ and $\mathcal{U}_y$ gives a problem identical to Problem (6) with the following exceptions:

- The constraints in (16) in place of constraints (6p) and (6q);
- The constraints in (25) in place of constraints (6b)–(6g).

Theorem 9. *The robust counterpart to the above problem is identical to Problem* (6) *with the following exceptions*:

- *The constraints in* (17) *in place of constraints* (6p) *and* (6q);
- *The constraints in* (26) *in place of constraints* (6b)–(6g).

Figure 2. Example of Synthetically Generated Data in Two Dimensions Alongside the True Generating Hyperplane



The proof of Theorem 9 is given in Appendix B, and the complete robust-in-both formulation is stated in full. This resulting problem is still a linear mixed-integer optimization problem, and so remains practically solvable.

7. Computational Experiments with Synthetic Data Sets

In this section, we evaluate the performance of robust methods on synthetically generated data sets in order to understand the relative performance of the different types of robustness and also how robust methods compare to the regularized methods used in practice. In these experiments, we run SVM and logistic regression methods to recover the separating hyperplane classifier on a synthetic example. We focus on SVM and logistic regression in this analysis because both of these classification models are suitable given the data-generation process and have widely used regularized methods to compare against.

7.1. Experimental Setup

The experiment uses data in $\mathbb{R}^2$. The data are generated synthetically in three parts:

1. Twenty-five points are generated as multivariate random normal, $N(1.5\mathbf{e}, \mathbf{I})$, where $\mathbf{e}$ is the vector of ones and $\mathbf{I}$ is the identity matrix. These points are given the label +1.

2. Twenty-five points are generated as multivariate random normal, $N(-1.5\mathbf{e}, \mathbf{I})$ and labeled −1.

3. Ten outlier points are introduced as multivariate random normal, $N(\mathbf{0}, 3\mathbf{I})$, where $\mathbf{0}$ is the vector of zeros. The labels are randomly generated as either −1 or +1.

We split these data 75%/25% into training and validation sets, which we used to tune the parameters for the regularized and robust methods. We included relatively few points in the training and validation sets to make the classification task nontrivial given the simple data-generation process. To create the test set, we generated 10,000 points in the same way as each major cluster of points (items 1 and 2 above).

An example of a data set generated according to this procedure is shown in Figure 2. We can see that there are two distinct clusters of points, with some scattered noise centered in the area between the two clusters. By the symmetry of this data-generation process, we can see that the true hyperplane separating the two clusters of points is given by the equation $\mathbf{e}^T\mathbf{x} = 0$, also shown in Figure 2. The goal of the experiment is to determine how closely the various methods can recover this truth in the data in the presence of added noise via the addition of these outlier points. In particular, we are interested in the following two measures:

- *Accuracy:* We measure accuracy by reporting the out-of-sample error of the trained classifiers on the larger test set.
- *Similarity:* To evaluate the ability of each method to recover the truth in the data, we measure the norm of the difference between the separating hyperplane generated by the methods and the true hyperplane ($\mathbf{e}^T\mathbf{x} = 0$).

7.2. Classification Methods

For these experiments, we considered SVM and logistic regression, as these both create classifiers with a single hyperplane, which matches the truth in the synthetic data. In both cases, we compared the nominal method, the regularized method, and all three robust methods (features, labels, and both). Each method was implemented in the Julia programming language, a rapidly maturing language designed for high-performance scientific computing (Bezanson et al. 2017). The optimization problems required by each method were formulated in JuMP, a state-of-the-art library for algebraic modeling and mathematical optimization (Lubin and Dunning 2015). The commercial solver Gurobi (Gurobi Optimization Inc. 2015) was used to solve the linear and mixed-integer optimization

Table 1. Performance Results for Synthetic Data Experiments

	Support vector machines		Logistic regression	
Method	Out-of-sample error (%)	Distance from truth	Out-of-sample error (%)	Distance from truth
Nominal	2.571 ± 0.021	0.357 ± 0.004	2.717 ± 0.023	0.388 ± 0.004
Regularized	2.643 ± 0.027	0.357 ± 0.004	2.694 ± 0.022	0.384 ± 0.004
Features	2.516 ± 0.020	0.345 ± 0.004	2.701 ± 0.023	0.386 ± 0.004
Labels	2.396 ± 0.018	0.320 ± 0.004	2.450 ± 0.019	0.332 ± 0.004
Both	2.363 ± 0.018	0.310 ± 0.004	2.436 ± 0.019	0.329 ± 0.004

Note. For each method, we report the mean and standard error over 2,000 runs for both the out-of-sample error and the distance of the generated classifier from the truth in the data.

problems for SVM, and the open source solver Ipopt (Wächter and Biegler 2006) was used to solve the convex optimization problems for logistic regression.

To ensure a fair comparison, we used the ℓ_1 norm in the regularized methods and set $q = \infty$ for the feature uncertainty set so that the norms in the robust methods are also ℓ_1 norms. For each method, the values of ρ and Γ were selected through validation when using the corresponding robust classifiers.

7.3. Results

This experiment was repeated 2,000 times, and we present the means and standard errors of the two measures for each method in Table 1. For SVM, the nominal and regularized methods have roughly the same power in recovering the truth in the data, after accounting for the standard errors. The feature-robust method improves upon the nominal method in both measures, and the label-robust method further improves upon both measures. The best performance in both measures is obtained when we consider both types of robustness simultaneously in the robust-in-both method, and this method improves significantly upon both methods that consider only a single type of robustness.

For logistic regression, we see that the nominal method performs the worst in both measures. The regularized method and our feature-robust method are roughly comparable, with the regularized method having a slight edge, and both offering a small improvement over the nominal method. As with SVM, the label-robust method offers significant improvement in both measures, and the robust-in-both method adds a further slight improvement on top of label-robust, showing that considering both types of robustness leads to additional power over considering just a single type.

In Table 2, we break down the results by percentile in out-of-sample error, and we report the $10^{th}, 20^{th}, \ldots, 90^{th}$ percentiles for each method. We find that robust methods match or outperform nominal and regularized methods across the board, and this relative improvement increases as the percentile increases. This follows our expectation that these robust methods reliably produce high-quality classifiers, which protects us from giving biased predictions in worst-case scenarios. In the worst-case scenario presented (90^{th} percentile out-of-sample error), robust-in-both SVM and logistic regression yield out-of-sample errors of 3.325% and 3.515%, whereas regularized methods give out-of-sample errors of 3.941% and 4.041%, respectively.

From these experiments on synthetic data, we conclude that our robust classifiers can effectively deal with data that have been contaminated with noise. For both SVM and logistic regression, we observe that the robust methods offer significant improvements over the nominal and regularized methods, both in their accuracy and in their

Table 2. Out-of-Sample Error Results by Percentile for Synthetic Data Experiments

		Percentile				
Classifier	Method	90^{th}	70^{th}	50^{th}	30^{th}	10^{th}
Support vector machines	Nominal	3.771	2.695	2.275	1.985	1.774
	Regularized	3.941	2.700	2.235	1.975	1.775
	Features	3.651	2.650	2.225	1.965	1.755
	Labels	3.381	2.460	2.125	1.915	1.755
	Both	3.325	2.425	2.100	1.890	1.740
Logistic regression	Nominal	4.096	2.940	2.400	2.050	1.795
	Regularized	4.041	2.910	2.385	2.045	1.795
	Features	4.050	2.917	2.393	2.043	1.790
	Labels	3.552	2.565	2.175	1.928	1.745
	Both	3.515	2.550	2.165	1.920	1.745

ability to correctly recover the truth in the data. Furthermore, we found that the robust-in-both methods that combine robustness in the features and labels performed stronger than the feature-robust and label-robust methods individually, demonstrating that there is value in considering both types of uncertainty simultaneously.

8. Computational Experiments with Real-World Data Sets

In this section, we report on a series of comprehensive computational benchmarks to compare robust methods to their nominal counterparts. We also explore problem characteristics that influence the performance gain of robust methods and derive a simple decision rule recommending when robust classification should be applied.

8.1. Experimental Setup

In order to comprehensively report performance of the robust classification methods on real data sets, we tested the accuracy of these methods on a selection of 75 problems from the UCI Machine Learning Repository (Lichman 2013). The data sets were selected to give a variety of problem sizes and difficulties to form a representative sample of real-world problems, with the largest data set having $n = 245,057$ observations and the highest number of features being $p = 857$.

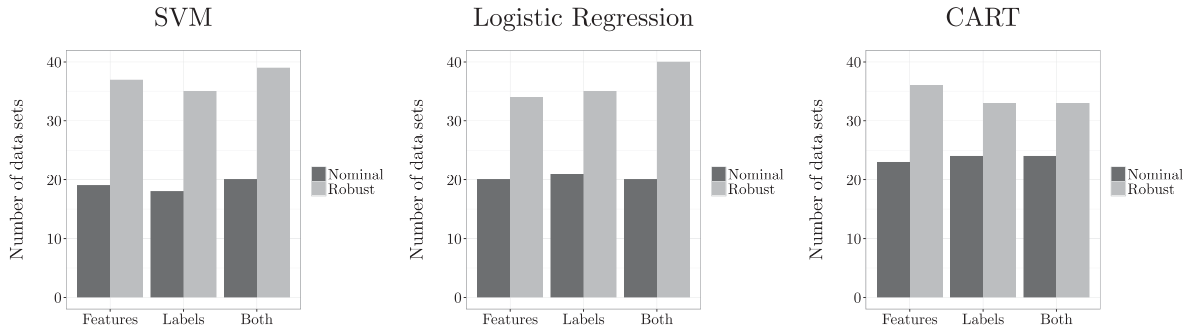
To obtain a binary classification problem for each data set, we considered the *one-versus-rest* problem of predicting the occurrence of the first class in the data set. Each data set was split into three parts: the training set (60%), the validation set (20%), and the testing set (20%). The training set was used to train each classifier for a variety of combinations of input parameters. For each combination of parameters, the misclassification error on the validation set was calculated, and this was used to select the best combination of parameters for each classifier. Finally, the classifier was trained by using these best parameters on the combined training and validation sets, before reporting the out-of-sample misclassification error on the testing set. All methods were trained, validated, and tested on the same random splits, and computational experiments were repeated five times for each data set with different splits. For each data set and classification method, we report the average out-of-sample accuracy across all five splits.

8.2. Classification Methods

In these real-world experiments, we considered all three classification methods: SVM, logistic regression, and decision trees. We set $q = \infty$ for all of the feature-robust and robust-in-both uncertainty sets, so that all the norms in the robust methods are ℓ_1. The implementations for SVM and logistic regression were identical to those used in the synthetic experiments, which are described in Section 7.1. We implemented optimal decision trees using the JuMP software package in Julia, and the commercial solver Gurobi (Gurobi Optimization Inc. 2015) was used to solve the mixed-integer optimization problems.

As in the other two methods, for optimal decision trees, we selected the values of ρ and Γ through validation when using the corresponding robust classifier. During validation, we also selected the complexity parameter (cp), the minimum number of points per node (minbucket), and the exploration depth around the warm start solution (explorationdepth). See Bertsimas and Dunn (2017) for a full description of these parameters. We compare the robust counterparts of the optimal decision tree problem to the CART heuristic rather than the nominal optimal decision tree problem. This allowed us to provide a benchmark of the robust methods against the state-of-the-art

Figure 3. Pairwise Comparisons Between Nominal and Individual Robust Methods



Notes. For each type of robustness, the plots compare that particular robust method and the nominal method and show the number of data sets for which each had the highest out-of-sample accuracy. CART, classification and regression trees; SVM, support vector machines.

Table 3. Out-of-Sample Accuracy Averaged Across Five Seeds for Each Classification Method and its Robust Counterparts on All Data Sets

Data set information			Support vector machines				Logistic regression				Classification and regression trees			
University of California Irvine data set name	n	p	Nominal	Features	Labels	Both	Nominal	Features	Labels	Both	Nominal	Features	Labels	Both
acute-inflammations-1	120	7	**1.0000**	**1.0000**	**1.0000**	0.9083	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9583	**1.0000**	**1.0000**	**1.0000**
acute-inflammations-2	120	7	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9750	**1.0000**	0.9833	**1.0000**
arrhythmia	68	280	0.5692	**0.7077**	0.6923	0.6308	**0.6923**	**0.6923**	**0.6923**	**0.6923**	0.6769	0.7077	0.6923	**0.7692**
balance-scale	625	5	**0.9200**	**0.9200**	**0.9200**	**0.9200**	**0.9200**	**0.9200**	**0.9200**	**0.9200**	**0.9200**	**0.9200**	**0.9200**	**0.9200**
banknote-authentication	1,372	5	0.9912	0.9869	**0.9927**	0.9912	0.9920	0.9905	**0.9927**	0.9905	0.9533	**0.9642**	0.9635	0.9635
blood-transfusion	748	5	**0.7638**	**0.7638**	**0.7638**	**0.7638**	**0.7826**	0.7812	0.7785	0.7812	0.7799	**0.7893**	0.7799	0.7799
breast-cancer	683	10	0.9500	**0.9574**	0.9559	0.9559	0.9559	**0.9574**	0.9544	**0.9574**	0.9338	**0.9456**	0.9426	0.9426
breast-cancer-diagnostic	569	31	0.9351	0.9596	0.9439	**0.9614**	0.9561	**0.9684**	0.9526	**0.9684**	0.9281	0.9053	**0.9333**	0.9018
breast-cancer-prognostic	194	33	0.7128	0.7128	0.7128	**0.7179**	0.7128	0.7231	**0.7692**	0.7590	0.7436	0.7385	0.7436	**0.7538**
car-evaluation	1,728	16	**0.8000**	0.7930	0.7832	0.7826	**0.8006**	0.7925	0.7994	0.7925	**0.8603**	0.8551	0.8597	0.8597
chess-king-rook-vs-king	28,056	35	**0.9004**	**0.9004**	**0.9004**	**0.9004**	**0.9004**	**0.9004**	**0.9004**	**0.9004**	**0.9004**	**0.9004**	**0.9004**	**0.9004**
chess-king-rook-vs-king-pawn	3,196	38	**0.9743**	0.9731	**0.9743**	0.9687	**0.9756**	0.9750	0.9734	0.9743	**0.9693**	**0.9693**	**0.9693**	**0.9693**
climate-model-crashes	540	19	0.9500	**0.9593**	0.9537	0.9574	0.9500	0.9537	**0.9556**	0.9537	0.9259	0.9241	**0.9296**	0.9241
cnae-9	1,080	857	0.9750	**0.9861**	0.9685	0.9481	0.9806	0.9796	0.9796	**0.9824**	0.9657	**0.9722**	0.9704	0.9694
congressional-voting-records	232	17	0.9565	**0.9870**	0.9783	0.9826	0.9739	**0.9826**	0.9739	**0.9826**	**0.9870**	0.9826	**0.9870**	**0.9870**
connectionist-bench	990	11	0.9758	0.9758	**0.9778**	0.9768	0.9747	**0.9778**	0.9737	0.9768	**0.9747**	0.9737	0.9727	0.9727
connectionist-bench-sonar	208	61	0.7073	**0.7707**	0.7561	0.7561	0.7463	0.7512	**0.7756**	0.7659	0.7268	**0.7317**	0.7122	0.7317
contraceptive-method-choice	1,473	12	0.6776	0.6769	**0.6789**	0.6755	0.6714	0.6769	0.6748	**0.6776**	0.6891	0.6980	**0.6986**	**0.6986**
credit-approval	653	38	0.8508	0.8569	0.8492	**0.8585**	0.8615	0.8615	0.8600	**0.8631**	**0.8569**	0.8415	0.8554	0.8415
cylinder-bands	277	485	0.5564	**0.7164**	0.5891	0.6691	**0.6727**	**0.6727**	**0.6727**	**0.6727**	0.6764	0.6800	0.6691	**0.7018**
dermatology	358	35	0.9662	0.9887	**0.9972**	0.9803	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**0.9887**	**0.9887**	0.9859	**0.9887**
echocardiogram	61	7	**0.7167**	0.7000	0.6833	0.6833	**0.7833**	0.7500	**0.7833**	0.7333	**0.7500**	0.7167	0.7333	**0.7500**
ecoli	336	8	0.9582	0.9522	0.9582	**0.9582**	**0.9612**	**0.9612**	**0.9612**	0.9582	**0.9493**	0.9343	0.9284	0.9284
fertility	100	13	0.8700	**0.9000**	0.8800	**0.9000**	0.8700	**0.9000**	0.8800	**0.9000**	**0.9000**	0.8400	0.8900	0.8400
flags	194	60	0.6923	**0.8769**	0.7949	0.8205	0.7641	**0.8564**	0.8462	**0.8564**	0.8821	0.8872	0.8923	**0.9026**
glass-identification	214	10	0.7163	0.7070	**0.7395**	0.7256	0.7116	0.7209	**0.7488**	0.7349	0.7674	0.7814	**0.7860**	**0.7860**
haberman-survival	306	4	0.7279	**0.7344**	**0.7344**	**0.7344**	**0.7410**	0.7311	0.7344	0.7311	**0.7049**	0.6623	0.6820	0.6787
hayes-roth	132	5	**0.6846**	0.6846	0.6769	0.6692	0.6615	**0.8000**	0.6769	0.7923	0.8154	**0.8154**	0.7385	0.7385
heart-disease-cleveland	297	19	**0.8407**	0.8339	0.8339	0.8203	0.8305	0.8271	**0.8339**	0.8305	0.7559	0.8000	0.7695	**0.8068**
hepatitis	80	20	**0.8500**	**0.8500**	0.8000	0.8125	0.8375	0.8250	**0.8625**	0.8250	0.8125	0.7875	**0.8250**	0.7875
hill-valley	606	101	0.5884	**0.9620**	0.5884	**0.9620**	**0.9934**	0.9636	0.9421	0.9636	**0.5504**	**0.5504**	**0.5504**	**0.5504**
hill-valley-noise	606	101	0.8612	0.8545	**0.8628**	0.8512	0.8463	**0.8876**	0.8083	**0.8876**	0.4744	**0.4992**	0.4942	0.4959
image-segmentation	210	20	0.9286	**0.9857**	0.9667	0.9476	0.9762	**0.9762**	**0.9762**	**0.9762**	0.9476	**0.9810**	0.9714	**0.9810**
indian-liver-patient	579	11	**0.7155**	**0.7155**	**0.7155**	**0.7155**	0.7172	0.7155	**0.7224**	**0.7224**	**0.6931**	0.6862	0.6914	0.6845
ionosphere	351	35	**0.8743**	**0.8743**	0.8514	**0.8743**	**0.8829**	0.8743	0.8571	0.8714	0.8971	**0.9086**	0.8914	**0.9086**
iris	150	5	**1.0000**	**1.0000**	**1.0000**	0.9800	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9867	**1.0000**
letter-recognition	20,000	17	0.9921	0.9922	**0.9923**	**0.9923**	0.9907	0.9907	0.9907	**0.9908**	**0.9896**	**0.9896**	**0.9896**	**0.9896**
libras-movement	360	91	0.9056	**0.9694**	0.9611	**0.9694**	0.9444	**0.9639**	0.9528	**0.9639**	0.9333	0.9361	**0.9528**	0.9389
magic-gamma-telescope	19,020	11	0.7922	0.7923	**0.7924**	**0.7924**	0.7916	0.7919	**0.7920**	0.7919	0.8364	0.8364	**0.8367**	**0.8367**
mammographic-mass	830	11	**0.8193**	0.8072	0.8000	0.8060	**0.8289**	**0.8289**	0.8217	0.8217	**0.8289**	0.8145	0.8217	0.8108
monks-problems-1	124	12	**0.8240**	0.7360	0.8000	0.8000	0.7440	0.7680	0.7760	**0.7920**	0.8080	**0.8400**	**0.8400**	**0.8400**
monks-problems-2	169	12	0.6118	**0.6176**	0.6118	**0.6176**	0.5647	**0.6235**	0.6176	**0.6235**	0.6118	0.6294	**0.6353**	**0.6353**

Table 3. (Continued)

Data set information			Support vector machines				Logistic regression				Classification and regression trees			
University of California Irvine data set name	*n*	*p*	Nominal	Features	Labels	Both	Nominal	Features	Labels	Both	Nominal	Features	Labels	Both
monks-problems-3	122	12	0.9167	**0.9333**	0.9167	**0.9333**	0.8500	0.9250	**0.9333**	0.9250	0.8917	**0.9333**	**0.9333**	**0.9333**
mushroom	5,644	77	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9894	**0.9901**	0.9894	0.9894
nursery	12,960	20	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.7910	0.7910
optical-recognition	3,823	65	0.9929	0.9961	**0.9974**	0.9966	0.9942	0.9969	**0.9976**	0.9971	**0.9830**	**0.9830**	**0.9830**	**0.9830**
ozone-level-detection-eight	1,847	73	**0.9301**	0.9301	0.9301	0.9295	0.9312	**0.9322**	0.9295	0.9317	0.9322	0.9322	**0.9328**	**0.9328**
ozone-level-detection-one	1,848	73	0.9545	**0.9702**	0.9561	**0.9702**	0.9561	**0.9702**	0.9686	**0.9702**	**0.9702**	**0.9702**	**0.9702**	**0.9702**
parkinsons	195	22	0.8564	0.8359	0.8513	**0.8615**	0.8410	0.8103	**0.8462**	0.8205	0.8615	0.8410	**0.8821**	0.8410
pen-based-recognition	7,494	17	**0.9904**	0.9889	0.9901	0.9891	**0.9903**	0.9899	0.9897	0.9897	0.9849	**0.9893**	0.9848	0.9848
pima-indians-diabetes	768	9	0.7765	0.7778	0.7765	**0.7791**	0.7778	0.7739	**0.7791**	0.7752	**0.7542**	0.7373	0.7477	0.7294
planning-relax	182	13	**0.7222**	**0.7222**	**0.7222**	**0.7222**	0.6778	0.6944	0.6944	**0.7000**	**0.6889**	0.6833	**0.6889**	0.6556
poker-hand	25,010	11	0.5010	0.5023	**0.5028**	0.5023	**0.5028**	0.5006	0.5018	0.5000	**0.5913**	**0.5913**	**0.5913**	**0.5913**
post-operative-patient	87	14	0.6235	0.6471	**0.7059**	**0.7059**	0.6118	0.6353	**0.6588**	**0.6588**	**0.6824**	0.6235	0.6471	0.6235
qsar-biodegradation	1,055	42	0.8749	0.8758	**0.8777**	0.8758	**0.8730**	**0.8730**	0.8701	0.8682	0.7943	**0.8142**	0.8104	0.8114
seeds	210	8	0.9524	0.9429	**0.9524**	0.9429	**0.9571**	0.9524	0.9524	0.9476	0.8429	0.8762	0.8667	**0.9000**
seismic-bumps	2,584	21	**0.9342**	**0.9342**	**0.9342**	**0.9342**	0.9319	**0.9327**	0.9327	0.9327	**0.9342**	**0.9342**	**0.9342**	**0.9342**
skin-segmentation	245,057	4	0.9281	0.9348	0.9328	**0.9366**	0.9184	0.9184	0.9345	**0.9345**	**0.9656**	**0.9656**	**0.9656**	**0.9656**
soybean-large	266	63	0.7736	**0.8830**	0.8642	0.8717	0.7962	0.8792	**0.9019**	0.8868	**0.8830**	0.8642	0.8491	0.8491
soybean-small	47	38	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
spambase	4,601	58	0.9257	**0.9265**	0.9252	**0.9265**	0.9230	**0.9248**	0.9246	0.9246	0.8896	0.8913	**0.8926**	**0.8926**
spect-heart	80	23	0.5875	**0.7125**	0.6500	**0.7125**	0.6750	**0.7875**	0.6625	0.7625	0.7625	**0.7750**	**0.7750**	**0.7750**
spectf-heart	80	45	0.5625	**0.6750**	0.6625	0.6500	0.5875	0.7500	0.5375	**0.7625**	0.7625	0.7250	**0.8000**	0.7375
statlog-project-german-credit	1,000	49	0.7300	**0.7420**	0.7380	0.7400	0.7380	**0.7470**	0.7400	0.7400	**0.7250**	0.7100	0.7030	0.7150
statlog-project-landsat-satellite	4,435	37	0.9811	0.9813	**0.9822**	0.9820	**0.9833**	0.9826	0.9833	0.9824	0.9477	0.9484	**0.9511**	**0.9511**
teaching-assistant-evaluation	151	53	**0.7000**	0.6733	0.6867	0.6733	**0.7133**	0.7067	**0.7133**	0.7067	0.6467	0.6733	0.6267	**0.7067**
thoracic-surgery	470	25	0.8426	**0.8511**	0.8426	**0.8511**	0.8213	**0.8489**	0.8362	0.8468	**0.8511**	0.8426	**0.8511**	0.8426
thyroid-disease-ann-thyroid	3,772	22	0.9920	**0.9926**	0.9918	0.9915	0.9934	0.9934	**0.9936**	0.9934	0.9915	**0.9971**	**0.9971**	**0.9971**
thyroid-disease-new-thyroid	215	6	0.8977	0.8837	**0.8977**	0.8884	**0.8977**	**0.8977**	**0.8977**	**0.8977**	0.8884	0.9023	**0.9302**	0.9116
tic-tac-toe-endgame	958	19	**0.9801**	**0.9801**	**0.9801**	**0.9801**	**0.9801**	**0.9801**	**0.9801**	**0.9801**	0.9005	**0.9026**	0.8995	0.8995
wall-following-robot-2	5,456	3	0.6220	**0.6544**	0.5415	0.5553	0.6081	0.6114	**0.6609**	**0.6609**	0.9879	**1.0000**	**1.0000**	**1.0000**
wall-following-robot-24	5,456	5	0.6235	0.6544	0.6420	**0.6561**	0.6301	0.6348	**0.6565**	0.6563	0.9879	**1.0000**	**1.0000**	**1.0000**
wine	178	14	0.9657	0.9657	**0.9714**	0.9657	0.9714	0.9714	**0.9943**	**0.9943**	0.9257	**0.9429**	0.9371	0.9371
yeast	1,484	9	**0.6902**	**0.6902**	**0.6902**	**0.6902**	0.6801	0.6828	**0.6929**	**0.6929**	**0.7286**	0.7219	0.7219	0.7219
zoo	101	17	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**

Note. For each data set, the best result (or multiple in the case of ties) for each method is indicated in bold, and the best method overall for the data set is underlined.

methods that are widely used today. For the CART method, we used the RPART package (Therneau et al. 2015) in the R programming language (R Core Team 2015).

Table 3 shows the out-of-sample accuracy performance of each classification method and its robust counterparts on all selected data sets. For each data set, the best result (or multiple in the case of ties) for each method is indicated in bold, and the best method overall for the data set is underlined.

8.3. Pairwise Comparisons

First, we present the results comparing individual robust classification methods against their nominal counterparts.

Results for the three nominal methods and all robust variations are summarized in Figure 3. Each pair of bars in the graph represents a pairwise comparison between a specific robust method and its nominal counterpart. Each bar represents the number of data sets for which the either the robust or nominal method produced the single strongest classifier, based on out-of-sample accuracy. We see that for each classification method, all types of robustness have a lead over the nominal ones. In the case of logistic regression and SVM, robust-in-both produces the most improvement in the number of correctly classified data sets. However, for CART, it is the feature robust method that is most effective in improving classification over the nominal counterpart. Because the robust-in-both method encompasses the individual feature and label robust methods, this result could be due to difficulties in validation where the selected combination of robustness parameters did not lead to better out-of-sample performance than the individual robust methods. The exact counts of wins, ties, and losses for each robust counterpart compared to the corresponding nominal method are shown in Table 4.

Next, we considered the best of the nominal and robust-in-both methods across SVM, logistic regression, and CART. For each data set, we recorded which of these six methods had the highest out-of-sample accuracy. Figure 4 shows the breakdown of counts for data sets in which there is a unique highest out-of-sample accuracy. All of the six methods yielded the unique highest out-of-sample accuracy for certain data sets, which indicates that each type of classifier is able to exploit different aspects of the data set in their own ways to potentially lead to higher-quality solutions. In all cases, the robust counterpart produced the highest number of uniquely optimal solutions.

8.4. Predicting the Effectiveness of Robust Classification

Thus far, we have demonstrated the strength of robust methods compared to their nominal counterparts over the set of 75 problems from the UCI Machine Learning Repository. For machine-learning practitioners, we would also like to provide guidance about when it is worthwhile to use robust classification methods in practical applications. In this section, we consider the problem of predicting whether or not a robust classifier is likely to improve out-of-sample accuracy relative to the nominal method, using only the dimension of the training data and the accuracy of the nominal method on these data. Note that we consider in-sample nominal accuracy because this is an attribute of the training problem and therefore is available at the validation stage when selecting the final classification method.

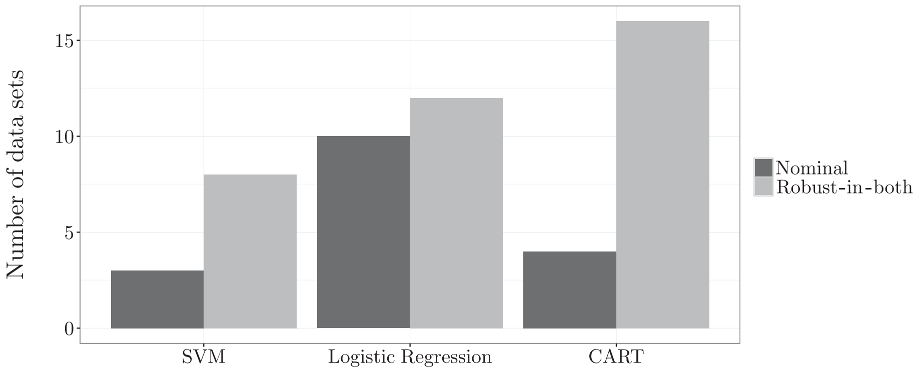
First, we considered the influence of nominal in-sample accuracy in isolation. Table 5 shows the improvement in out-of-sample accuracy of robust-in-both methods over their nominal counterparts for different ranges of nominal in-sample accuracy. We defined the *robust improvement* as the absolute difference in out-of-sample accuracy between the methods—that is, the accuracy of the robust-in-both method less the accuracy of the nominal method. For instance, if the robust-in-both and nominal methods had accuracies of 84.7% and 81.3%, respectively, the robust improvement would be +3.4%.

The most significant result was for data sets where nominal SVM had in-sample accuracy below 60%. For these six problems, robust-in-both SVM improved upon the out-of-sample accuracy in every instance and yielded an average

Table 4. Pairwise Comparisons of Robust Classification Methods Against Their Nominal Counterparts

Nominal method	Robustness type	Wins	Losses	Ties
Support vector machines	Features	37	19	19
	Labels	35	18	22
	Both	39	20	16
Logistic regression	Features	34	20	21
	Labels	35	21	19
	Both	40	20	15
Classification and regression trees	Features	36	23	16
	Labels	33	24	18
	Both	33	24	18

Figure 4. Comparison of the Number of Data Sets for Which the Nominal and Robust-in-Both Approaches for Each Method Give the Highest Out-of-Sample Accuracy



Note. CART, classification and regression trees; SVM, support vector machines.

robust improvement of 10.7%. For logistic regression and SVM, we saw that as the nominal accuracy increased, both the proportion of robust-in-both wins and the robust improvement in accuracy decreased. For CART, the robust improvement was largely independent of the nominal accuracy, although the win proportion was higher for problems with nominal accuracy in the range of 90%–100%. This suggests that nominal in-sample accuracy by itself is not a strong predictor of robust effectiveness for CART methods. However, note that there are only four data sets with a nominal CART accuracy below 70%, the region where the other robust methods are strongest.

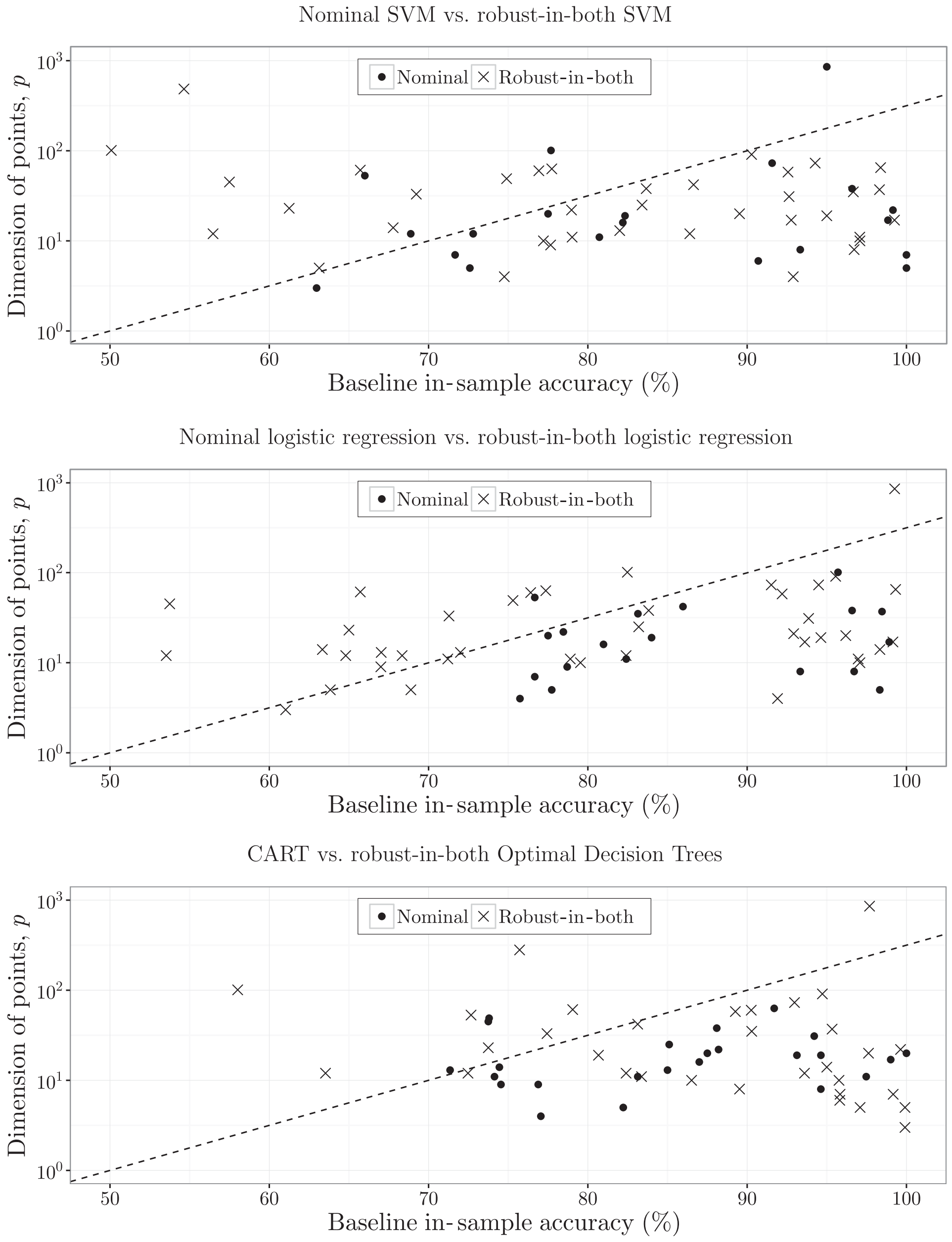
Next, we considered the combined influence of nominal in-sample accuracy and dimension of data points on the robust improvement. Figure 5 plots the winning method against these two attributes of the training problem. We have constructed a dividing line that is identical on all three plots that partitions the points into two regions. This line follows the equation $\log_{10}(p) = 0.05a - 2.5$, where a is the in-sample accuracy of the nominal method on the data set, p is the dimension of the data set, and the coefficients 0.05 and 2.5 were selected manually. In Table 6, we present a breakdown of the relative performance of the nominal and robust-in-both methods in the two regions. For all three classifiers, robust methods beat nominal methods for a majority of data sets in the region of lower nominal accuracy and high dimensionality (above the dividing line). In this region, we saw significant average improvements in out-of-sample accuracy of 5.3% for SVM, 4.0% for logistic regression, and 1.3% for CART. Below the dividing line, we observed that robust methods were still competitive with nominal methods, with neither offering a significant advantage.

We also include in Table 6 a comparison of robust-in-both optimal decision trees to nominal optimal decision trees. Previously, we have only considered the performance relative to CART in order to provide a strong benchmark against the state-of-the-art methods, but it is also insightful to directly compare the robust formulation

Table 5. Improvement Due to Robustness by Baseline In-Sample Accuracy, Comparing the Baseline Method to the Corresponding Robust-in-Both Classifier

Nominal method	Nominal accuracy	Wins	Losses	Ties	Robust improvement
Support vector machines	0%–60%	6	0	0	10.7 ± 5.6%
	60%–70%	5	3	1	2.2 ± 1.9%
	70%–80%	8	5	3	0.9 ± 1.1%
	80%–90%	6	3	1	0.3 ± 0.5%
	90%–100%	14	9	11	0.0 ± 0.4%
Logistic regression	0%–60%	2	1	0	7.7 ± 5.2%
	60%–70%	10	0	0	4.5 ± 1.2%
	70%–80%	8	7	0	1.2 ± 1.0%
	80%–90%	4	5	1	1.1 ± 0.9%
	90%–100%	16	7	14	0.2 ± 0.1%
Classification and regression trees	0%–60%	1	0	2	0.7 ± 0.7%
	60%–70%	1	0	0	2.4 ± —%
	70%–80%	7	8	2	0.1 ± 0.9%
	80%–90%	7	8	0	−0.3 ± 0.9%
	90%–100%	17	8	14	−0.1 ± 0.6%

Figure 5. Plots of Winning Method (Nominal vs. Robust-in-Both) by the Baseline In-Sample Accuracy and Dimension of Points in Each Data Set



Notes. The dashed line divides each plot into two regions with different levels of robustness gain. Nominal and robust-in-both wins are indicated by • and ×, respectively. CART, classification and regression trees; SVM, support vector machines.

to its nominal counterpart. Below the dividing line, the robust-in-both approach is not as strong compared to the optimal decision trees as it is compared to CART. This can be attributed to the fact that the optimal decision trees are a stronger classification method than CART and thus provide a stronger nominal baseline. However, we see that above the line, the relative improvement of robust-in-both optimal decision trees over optimal decision trees is very similar to their improvement over CART, with an average improvement in out-of-sample accuracy of 1.4%. This, therefore, shows that the dividing line is a strong predictor for when robust methods perform strongest relative to nominal methods, even in the presence of a significantly stronger nominal method.

It seems natural that the data dimension and nominal accuracy are likely indicative of the problem difficulty. This implies that robust methods are most beneficial for harder problems. We also expect robust methods to perform

Table 6. Improvement Due to Robustness by Baseline In-Sample Accuracy and Dimension of Points, Comparing the Nominal Method to the Corresponding Robust-in-Both Classifier

Baseline method	Region	Wins	Losses	Ties	Robust improvement
Nominal support vector machines	Above	14	4	3	5.3 ± 1.9%
	Below	25	16	13	−0.2 ± 0.3%
Nominal logistic regression	Above	17	2	1	4.0 ± 1.0%
	Below	23	18	14	0.4 ± 0.3%
Nominal optimal decision trees	Above	7	3	4	1.4 ± 0.8%
	Below	17	25	19	−0.7 ± 0.4%
Classification and regression trees	Above	9	3	2	1.3 ± 0.9%
	Below	24	21	16	−0.3 ± 0.5%

Note. Region "Above" refers to the top-left sections in Figure 5 (high data dimension, low baseline accuracy); region "Below" refers to the bottom-right sections in Figure 5 (low data dimension, high baseline accuracy).

strongest on noisy data. Together, this offers evidence that problem difficulty and data uncertainty are correlated, a result that is consistent with intuition. Based on the dividing line used earlier, we present the following decision rule to address the task of predicting the effectiveness of robust methods over nominal:

$$\log_{10}(p) \geq 0.05a - 2.5, \tag{31}$$

where p is the dimension of the data points, and a is the nominal in-sample accuracy. If this relationship is satisfied, the data set falls into the "Above" region of Table 6, and therefore the robust classification methods are highly likely to offer significant accuracy improvements over their nominal counterparts.

This demonstrates that we can predict with high accuracy a significant improvement in out-of-sample accuracy when using robust methods for classification problems with high-dimensional data and low nominal accuracy. This has large practical importance for machine learning; given a real-world classification problem, (31) gives a simple but strong recommendation for when to use robust classification in place of nominal SVM, logistic regression, or CART.

8.5. Comparison with Regularized Methods

To demonstrate the added value of our principled framework for modeling data uncertainty with robust optimization, we compare the robust classification methods to other popular methods that exhibit robust properties indirectly.

First, we compared our feature-robust SVM to ℓ_1-regularized SVM, which is equivalent to classical SVM except for the ℓ_1 norm regularizer term. This is a feature-robust method under a different uncertainty set (see Section 4.2). We implemented Problem (3) in JuMP and solved this problem with Gurobi. Experimentally, feature-robust SVM and ℓ_1-regularized SVM produced comparable classifiers; across all 75 data sets analyzed, the average difference in out-of-sample accuracy between these two methods was 0.2 ± 0.4%. This, therefore, gives evidence that our proposed uncertainty set for feature robustness is an equally strong model of the uncertainty in the features of the data.

Next, to benchmark robust-in-both methods against regularized methods, we compared robust-in-both SVM against ℓ_1-regularized SVM, and robust-in-both logistic regression against ℓ_1-regularized logistic regression (which uses an ad-hoc method for introducing robustness). For ℓ_1-regularized logistic regression, we implemented Problem (5) with $q = 1$ in JuMP and solved this problem with Ipopt. We present the accuracy results for this comparison in Table 7.

In Table 8, we present the relative performance of robust-in-both and regularized methods broken down into the same two regions as defined in Section 8.4. As before, the regions were determined by the in-sample accuracy of the nonrobust method and the data dimension. We saw that for both SVM and logistic regression, robust methods still offered improved accuracy over regularized methods for a majority of data sets in the region of lower nominal accuracy and high dimensionality (above the dividing line). In this region, we saw average improvements in out-of-sample accuracy of 0.5% over regularized SVM and 1.9% over regularized logistic regression. Below the dividing line, we observed that robust methods were still competitive with nominal methods, although regularized SVM outperformed robust SVM by 0.7% in this region. If we consider alternate norms and compare robust SVM and logistic regression against ℓ_2-regularized methods instead, we obtain similar results.

These results demonstrate that classifiers do benefit from a principled approach to robustness evidenced in real-world data, even when compared to regularized methods that are stronger than nominal ones. In all cases, we

Table 7. Out-of-Sample Accuracy Averaged Across Five Seeds for Each Method Using Both Regularized and Robust-in-Both Methods on All Data Sets

Data set information			Support vector machines		Logistic regression	
University of California Irvine data set name	*n*	*p*	Regularized	Robust	Regularized	Robust
acute-inflammations-1	120	7	**1.0000**	0.9083	**1.0000**	**1.0000**
acute-inflammations-2	120	7	**1.0000**	**1.0000**	**1.0000**	**1.0000**
arrhythmia	68	280	0.6154	**0.6308**	**0.7538**	0.6923
balance-scale	625	5	**0.9200**	**0.9200**	**0.9200**	**0.9200**
banknote-authentication	1,372	5	0.9869	**0.9912**	0.9855	**0.9905**
blood-transfusion	748	5	**0.7638**	0.7638	0.7664	**0.7812**
breast-cancer	683	10	**0.9679**	0.9559	**0.9664**	0.9574
breast-cancer-diagnostic	569	31	**0.9719**	0.9614	**0.9684**	0.9684
breast-cancer-prognostic	194	33	**0.7692**	0.7179	**0.7692**	0.7590
car-evaluation	1,728	16	**0.7977**	0.7826	**0.7936**	0.7925
chess-king-rook-vs-king	28,056	35	**0.9004**	**0.9004**	**0.9004**	**0.9004**
chess-king-rook-vs-king-pawn	3,196	38	**0.9743**	0.9687	**0.9756**	0.9743
climate-model-crashes	540	19	**0.9611**	0.9574	**0.9556**	0.9537
cnae-9	1,080	857	**0.9769**	0.9481	0.9713	**0.9824**
congressional-voting-records	232	17	0.9787	**0.9826**	0.9574	**0.9826**
connectionist-bench	990	11	0.9737	**0.9768**	0.9707	**0.9768**
connectionist-bench-sonar	208	61	0.7268	**0.7561**	0.7073	**0.7659**
contraceptive-method-choice	1,473	12	**0.6800**	0.6755	**0.6814**	0.6776
credit-approval	653	38	**0.8626**	0.8585	**0.8733**	0.8631
cylinder-bands	277	485	**0.7200**	0.6691	0.6182	**0.6727**
dermatology	358	35	**0.9915**	0.9803	**1.0000**	**1.0000**
echocardiogram	61	7	**0.7000**	0.6833	0.6667	**0.7333**
ecoli	336	8	**0.9791**	0.9582	**0.9731**	0.9582
fertility	100	13	0.8500	**0.9000**	0.8400	**0.9000**
flags	194	60	**0.8872**	0.8205	**0.8615**	0.8564
glass-identification	214	10	**0.7302**	0.7256	**0.7395**	0.7349
haberman-survival	306	4	0.7279	**0.7344**	0.7180	**0.7311**
hayes-roth	132	5	**0.8519**	0.6692	**0.8074**	0.7923
heart-disease-cleveland	297	19	**0.8305**	0.8203	**0.8441**	0.8305
hepatitis	80	20	**0.8375**	0.8125	0.8125	**0.8250**
hill-valley	606	101	0.8364	**0.9620**	**0.9884**	0.9636
hill-valley-noise	606	101	0.8132	**0.8512**	0.8678	**0.8876**
image-segmentation	210	20	**0.9905**	0.9476	**0.9810**	0.9762
indian-liver-patient	579	11	**0.7155**	**0.7155**	**0.7224**	0.7224
ionosphere	351	35	**0.8743**	**0.8743**	**0.8943**	0.8714
iris	150	5	**1.0000**	0.9800	**1.0000**	**1.0000**
letter-recognition	20,000	17	0.9916	**0.9923**	0.9904	**0.9908**
libras-movement	360	91	**0.9694**	**0.9694**	0.9583	**0.9639**
magic-gamma-telescope	19,020	11	0.7848	**0.7924**	0.7862	**0.7919**
mammographic-mass	830	11	**0.8120**	0.8060	**0.8301**	0.8217
monks-problems-1	124	12	0.6960	**0.8000**	0.6560	**0.7920**
monks-problems-2	169	12	0.5824	**0.6176**	0.5882	**0.6235**
monks-problems-3	122	12	**0.9360**	0.9333	**0.9360**	0.9250
mushroom	5,644	77	**1.0000**	**1.0000**	**1.0000**	**1.0000**
nursery	12,960	20	**1.0000**	**1.0000**	**1.0000**	**1.0000**
optical-recognition	3,823	65	0.9956	**0.9966**	0.9958	**0.9971**
ozone-level-detection-eight	1,847	73	**0.9355**	0.9295	**0.9366**	0.9317
ozone-level-detection-one	1,848	73	**0.9702**	**0.9702**	0.9675	**0.9702**
parkinsons	195	22	**0.8872**	0.8615	**0.8462**	0.8205
pen-based-recognition	7,494	17	**0.9893**	0.9891	0.9896	**0.9897**
pima-indians-diabetes	768	9	0.7647	**0.7791**	0.7660	**0.7752**
planning-relax	182	13	0.7027	**0.7222**	**0.7027**	0.7000
poker-hand	25,010	11	0.5018	**0.5023**	**0.5005**	0.5000
post-operative-patient	87	14	**0.7059**	**0.7059**	**0.7059**	0.6588
qsar-biodegradation	1,055	42	0.8692	**0.8758**	0.8578	**0.8682**
seeds	210	8	0.9333	**0.9429**	**0.9619**	0.9476
seismic-bumps	2,584	21	**0.9342**	**0.9342**	**0.9335**	0.9327
skin-segmentation	245,057	4	0.9326	**0.9366**	0.9187	**0.9345**

Table 7. (Continued)

Data set information						
			Support vector machines		Logistic regression	
University of California Irvine data set name	*n*	*p*	Regularized	Robust	Regularized	Robust
soybean-large	266	63	**0.9094**	0.8717	**0.9170**	0.8868
soybean-small	47	38	**1.0000**	**1.0000**	**1.0000**	**1.0000**
spambase	4,601	58	**0.9287**	0.9265	0.9241	**0.9246**
spect-heart	80	23	0.6375	**0.7125**	0.6750	**0.7625**
spectf-heart	80	45	0.6375	**0.6500**	0.6750	**0.7625**
statlog-project-german-credit	1,000	49	**0.7420**	0.7400	0.7350	**0.7400**
statlog-project-landsat-satellite	4,435	37	**0.9867**	0.9820	**0.9851**	0.9824
teaching-assistant-evaluation	151	53	**0.7200**	0.6733	**0.8067**	0.7067
thoracic-surgery	470	25	**0.8511**	**0.8511**	**0.8532**	0.8468
thyroid-disease-ann-thyroid	3,772	22	0.9905	**0.9915**	0.9910	**0.9934**
thyroid-disease-new-thyroid	215	6	0.8837	**0.8884**	0.8977	**0.8977**
tic-tac-toe-endgame	958	19	**0.9812**	0.9801	**0.9801**	0.9801
wall-following-robot-2	5,456	3	**0.6440**	0.5553	0.6537	**0.6609**
wall-following-robot-24	5,456	5	0.6436	**0.6561**	**0.6565**	0.6563
wine	178	14	**0.9829**	0.9657	0.9886	**0.9943**
yeast	1,484	9	0.6869	**0.6902**	0.6842	**0.6929**
zoo	101	17	**1.0000**	**1.0000**	**1.0000**	**1.0000**

Note. For each data set, the best result (or both in the case of a tie) for each method is indicated in bold.

observed that our robust methods performed best on classification problems that satisfy the decision rule given by Equation (31).

8.6. Computational Tractability and Speed

Table 9 shows the complexity of each nominal method and its robust counterparts. Under all three classifiers, the feature robustness does not change the nature of the optimization problem complexity. Logistic regression changes from unconstrained convex optimization to constrained when label robustness is introduced. Label-robust SVM introduces integer-valued variables and therefore becomes a mixed-integer optimization problem. For decision trees, because the nominal formulation is mixed-integer optimization formulation, label robustness did not change the nature of the problem. Robustness-in-both takes the maximum complexity between feature-robust and label-robust formulations; in this case, the complexity is equal to that of the label-robust in all three classifiers.

In order to provide empirical measures of the complexity of each method, we also compared the total time required to solve a problem instance for each method with or without robustness across a selection of UCI data sets. These sets were chosen to be representative of the various dimensions and separability among all data sets. For the robust methods, a typical choice of $\rho = 0.01$, $\Gamma = 10\%$ was used. The problems were solved on a machine with a 16-core, Intel Xeon E5-2687W (3.1 GHz) Processor and 128 GB RAM, and the total solver time taken to solve each problem instance to optimality was recorded. All tests were limited to a single thread for consistency. If the problem was not solved to optimality within an hour, the solve was terminated. In this case, we report the time taken to find the solution that was best under the hour time limit. In particular for robust counterparts of CART, strong heuristics gave very good solutions almost instantly, and sometimes these solutions were not further improved after an hour.

Table 8. Improvement due to Robustness by Baseline In-Sample Accuracy and Dimension of Points, Comparing the Regularized Method to the Corresponding Robust-in-Both Classifier

Baseline method	Region	Wins	Losses	Ties	Robust improvement
Regularized support vector machines	Above	8	6	1	0.5 ± 1.1%
	Below	18	29	13	−0.7 ± 0.5%
Regularized logistic regression	Above	8	5	0	1.9 ± 1.6%
	Below	24	28	10	0.1 ± 0.3%

Note. Region "Above" refers to the top-left sections in Figure 5 (high data dimension, low baseline accuracy); region "Below" refers to the bottom-right sections in Figure 5 (low data dimension, high baseline accuracy).

Table 9. Problem Complexity of Nominal and Robust Classification Methods

Method	Nominal	Feature-robust	Label-robust	Robust-in-both
Support vector machines	Linear optimization	Linear optimization	Mixed-integer optimization	Mixed-integer optimization
Logistic regression	Unconstrained convex optimization	Unconstrained convex optimization	Constrained convex optimization	Constrained convex optimization
Decision trees	Mixed-integer optimization	Mixed-integer optimization	Mixed-integer optimization	Mixed-integer optimization

In a real-world application of these methods, the time taken to find the solution is the more important measure than the time taken to prove the solution optimal; therefore, time to finding solution is used.

The results for selected data sets are presented in Table 10. In general, the nominal and feature-robust classifiers required solver time of around the same order of magnitude. Label robustness generally slowed down computation by 1–2 orders of magnitude; in particular, because label robustness for SVM changed the problem from a linear-optimization problem to a mixed-integer optimization problem, the computational time was considerably longer. The robust-in-both classifier tended to exhibit similar solution times to the label-robust method.

8.7. The Price of Robustness

Introducing robustness in classifiers generates solutions that may be suboptimal under the nominal data, but are likely to remain feasible or close to optimal when the data change (Bertsimas and Sim 2004). We can evaluate this trade-off for the robust classifiers by comparing the out-of-sample accuracies, as evaluating the model accuracy on the unobserved testing data can be thought of as a way of exposing the solution to perturbations in the training data.

The empirical findings show that robustness improves prediction accuracy in many real-world data sets across all three classifications methods. In each classifier family individually, feature-robust, label-robust, and robust-in-both generally had higher winning counts compared to their nominal counterpart. When comparing all three nominal methods and their robust versions together, robustness continued to perform well in the majority of data sets, particularly in subsets of data sets that were more difficult to classify. Overall, robust methods offered quality solutions that nominal ones could not achieve.

Another practical aspect on the price of robustness is the computational time requirement. In most cases, the computational time for robust methods is on the same order of magnitude as their respective nominal ones, suggesting that robustification does not incur a significant burden on speed. It should also be noted that, as mixed-integer optimization problems, label-robust SVM and CART can easily be limited by computational constraints. Several problems we considered were not solved to optimality and, rather, were stopped after a smaller time limit

Table 10. Solver Time for Selected University of California Irvine Data Sets in Seconds for $\rho = 0.01$ and $\Gamma = 10\%$

		University of California Irvine data set (number of points; dimension)					
		hayes-roth	banknote-authentication	nursery	skin-segmentation	flags	cnae-9
Method	Type of robustness	(132; 4)	(1,372; 4)	(12,960; 19)	(245,057; 3)	(194; 59)	(1,080; 856)
Support vector machines	Nominal	0.00	0.02	0.05	454.38	0.01	0.02
	Feature	0.00	0.02	0.36	553.94	0.01	0.32
	Label	0.23	4.50	58.58	695.06[a]	0.37	2.41
	Both	0.24	4.77	91.70	695.06[a]	0.60	15.81
Logistic regression	Nominal	0.00	0.05	0.02	0.03	0.03	0.41
	Feature	0.00	0.08	0.03	0.16	0.16	113.24
	Label	0.03	0.24	4.70	56.33	0.06	0.52
	Both	0.03	0.25	5.45	71.12	0.06	0.51
Decision trees	Nominal	0.02	0.02	0.18	1.44	0.02	0.65
	Feature	0.04	0.02[a]	1.06	1.46[a]	0.64	0.65[a]
	Label	3.39	45.00[a]	0.18[a]	1.47[a]	3.01	183.43
	Both	0.05	—[b]	0.18[a]	1.48[a]	2.39	146.01

[a] Not solved to optimality within the time limit. The time reported is instead the time taken to find the solution that is best at termination.
[b] The robust-in-both optimal decision tree problem is infeasible for this particular choice of ρ/Γ.

to get a strong, yet suboptimal, solution. Allowing for longer time limits in these cases has the potential for further improving the accuracy.

9. Conclusions

In this paper, we consider three major classification methods under a modern robust optimization perspective: SVM, logistic regression, and CART. For each classifier, we address uncertainties in features, labels, and both simultaneously in a principled manner by constructing appropriate uncertainty sets and deriving robust counterparts in the same way for all methods. We also discuss the implementation and practical solvability for each method with robustness.

Synthetic experiments demonstrate that our methods derived by taking a principled approach to robust classification may improve greatly upon existing classification methods. In the synthetic study, we show that robust-in-both SVM and logistic regression outperform both nominal and regularized methods and produce classifiers closer to the underlying truth, especially in the worst-case scenarios. In particular, the 90^{th} percentile out-of-sample errors for our methods are significantly lower than the 90^{th} percentile out-of-sample errors for the benchmark methods. Because regularized SVM can be cast as a feature-robust optimization problem for a particular uncertainty set, this shows that the choice of uncertainty set may be critical. For the simple synthetic problems considered here, the robust methods derived using label uncertainty sets performed best.

To evaluate the value of adding robustness in practice, we performed computational experiments on a large sample of data sets from the UCI Machine Learning Repository, comparing nominal, regularized, and robust methods for each of the three classifiers. We found that robust solutions provided higher out-of-sample accuracy for many data sets, and the large majority of classifiers that strictly outperformed all other methods were robust. In particular, we identified that high-dimensional and hard-to-separate problems benefited most from our principled approach to robustness. The findings suggest that we can predict how much value robustness will add to a data set, given only the accuracy of a classical method and dimension of the data set features. This allows us to offer guidance as to when robust classification methods can deliver significant improvements in practical settings.

Appendix A. Equivalence with Classical Support Vector Machines

The feature-robust counterpart presented in Theorem 1 is similar to the classical SVM problem (2). Making the substitution $\tilde{\xi}_i = \xi_i - \rho\|\mathbf{w}\|_{q^*}$ in Problem (11), we obtain

$$\begin{aligned} \min_{\mathbf{w},b} \quad & n\rho\|\mathbf{w}\|_{q^*} + \sum_{i=1}^n \tilde{\xi}_i \\ \text{s.t.} \quad & y_i(\mathbf{w}^T\mathbf{x}_i - b) \ge 1 - \tilde{\xi}_i \qquad i = 1,\ldots,n, \\ & \tilde{\xi}_i \ge -\rho\|\mathbf{w}\|_{q^*} \qquad i = 1,\ldots,n. \end{aligned} \tag{A.1}$$

Comparing Problem (A.1) to the classical SVM formulation (2), we observe that adding feature robustness or regularization to the hinge loss classifier leads to nearly identical optimization problems. Depending upon the choice of uncertainty set and the selection of the regularizing term, this equivalence may be exact. Under the assumption that the training data are nonseparable, Fertis (2009) has shown that the robust optimization problem

$$\begin{aligned} \min_{\mathbf{w},b} \max_{\Delta\mathbf{X}\in\tilde{\mathcal{U}}_x} \quad & \sum_{i=1}^n \xi_i \\ \text{s.t.} \quad & y_i(\mathbf{w}^T\mathbf{x}_i - b) \ge 1 - \xi_i \qquad i = 1,\ldots,n, \\ & \xi_i \ge 0 \qquad i = 1,\ldots,n, \end{aligned} \tag{A.2}$$

is exactly equivalent to the problem

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \rho\|\mathbf{w}\|_{q^*} + \sum_{i=1}^n \xi_i \\ \text{s.t.} \quad & y_i(\mathbf{w}^T\mathbf{x}_i - b) \ge 1 - \xi_i \qquad i = 1,\ldots,n, \\ & \xi_i \ge 0 \qquad i = 1,\ldots,n, \end{aligned} \tag{A.3}$$

where

$$\tilde{\mathcal{U}}_x = \left\{ \Delta\mathbf{X} \in \mathbb{R}^{n\times p} \,\middle|\, \sum_{i=1}^n \|\Delta x_i\|_q \le \rho \right\}.$$

It follows that (A.3) is equivalent to the classical SVM problem (2) for the choice of $q^* = 2$, or the ℓ_1-regularized SVM problem (3) for the choice of $q^* = \infty$. This implies that the classical and regularized SVM problems are indeed robust formulations of the nominal hinge loss classifier under specific choices of uncertainty set.

Appendix B. Robust-in-Both Proofs

B.1. Soft-Margin Support Vector Machines

Proof of Theorem 7. Using a similar process as in the proof of Theorem 1, we rearrange the first constraint and solve the minimization problem. Problem (27) can be reformulated as

$$\begin{aligned} \min_{\mathbf{w},b} \quad \max_{\Delta\mathbf{y}\in\mathcal{U}_y} \quad & \sum_{i=1}^{n} \xi_i \\ \text{s.t.} \quad & y_i(1-2\Delta y_i)(\mathbf{w}^T\mathbf{x}_i - b) \geq 1 - \xi_i + \rho\|\mathbf{w}\|_{q^*} && i = 1,\ldots,n, \\ & \xi_i \geq 0 && i = 1,\ldots,n. \end{aligned}$$

We can reformulate this as

$$\min_{\mathbf{w},b} \quad \max_{\Delta\mathbf{y}\in\mathcal{U}_y} \quad \sum_{i=1}^{n} \max\{1 - y_i(1-2\Delta y_i)(\mathbf{w}^T\mathbf{x}_i - b) + \rho\|\mathbf{w}\|_{q^*}, 0\}.$$

Now we follow the approach in the proof of Theorem 4. □

B.2. Logistic Regression

Proof of Theorem 8. Using a similar process as in the proof of Theorem 2, we first solve the innermost minimization problem and show that Problem (29) is equivalent to

$$\max_{\boldsymbol{\beta},\beta_0} \quad \min_{\Delta\mathbf{y}\in\mathcal{U}_y} - \sum_{i=1}^{n} \log\left(1 + e^{-y_i(1-2\Delta y_i)(\boldsymbol{\beta}^T\mathbf{x}_i + \beta_0) + \rho\|\boldsymbol{\beta}\|_{q^*}}\right). \tag{B.1}$$

Now we follow the approach in the proof of Theorem 5. Because the polyhedron $\{\Delta\mathbf{y}\in\mathbb{R}^n | \sum_{i=1}^{n}\Delta y_i \leq \Gamma, 0 \leq \Delta y_i \leq 1\}$ has integer extreme points, the inner minimization problem above has the same objective as when the integer constraints are relaxed:

$$\begin{aligned} \min_{\Delta\mathbf{y}} \quad & -\sum_{i=1}^{n} \log\left(1 + e^{-y_i(1-2\Delta y_i)(\boldsymbol{\beta}^T\mathbf{x}_i + \beta_0) + \rho\|\boldsymbol{\beta}\|_{q^*}}\right) \\ \text{s.t.} \quad & 0 \leq \Delta y_i \leq 1 \;\; i = 1,\ldots,n, \\ & \sum_{i=1}^{n} \Delta y_i \leq \Gamma. \end{aligned}$$

Define the function $f_i(\Delta y_i) = -\log\left(1 + e^{-y_i(1-2\Delta y_i)(\boldsymbol{\beta}^T\mathbf{x}_i + \beta_0) + \rho\|\boldsymbol{\beta}\|_{q^*}}\right)$ for $i = 1,\ldots,n$. Because $\Delta y_i \in \{0,1\}$, we can express $f_i(\Delta y_i)$ as

$$\begin{aligned} f_i(\Delta y_i) &= [f(1) - f(0)]\Delta y_i + f(0) \\ &= \log\left(\frac{1 + e^{-y_i(\boldsymbol{\beta}^T\mathbf{x}_i + \beta_0) + \rho\|\boldsymbol{\beta}\|_{q^*}}}{1 + e^{y_i(\boldsymbol{\beta}^T\mathbf{x}_i + \beta_0) + \rho\|\boldsymbol{\beta}\|_{q^*}}}\right)\Delta y_i - \log\left(1 + e^{-y_i(\boldsymbol{\beta}^T\mathbf{x}_i + \beta_0) + \rho\|\boldsymbol{\beta}\|_{q^*}}\right). \end{aligned}$$

The inner minimization problem can thus be expressed as

$$\begin{aligned} \min_{\Delta\mathbf{y}} \quad & \sum_{i=1}^{n} \left[\log\left(\frac{1 + e^{-y_i(\boldsymbol{\beta}^T\mathbf{x}_i + \beta_0) + \rho\|\boldsymbol{\beta}\|_{q^*}}}{1 + e^{y_i(\boldsymbol{\beta}^T\mathbf{x}_i + \beta_0) + \rho\|\boldsymbol{\beta}\|_{q^*}}}\right)\Delta y_i - \log\left(1 + e^{-y_i(\boldsymbol{\beta}^T\mathbf{x}_i + \beta_0) + \rho\|\boldsymbol{\beta}\|_{q^*}}\right)\right] \\ \text{s.t.} \quad & 0 \leq \Delta y_i \leq 1 \;\; i = 1,\ldots,n, \\ & \sum_{i=1}^{n} \Delta y_i \leq \Gamma. \end{aligned}$$

By strong duality, the inner minimization problem has the same objective value as its dual problem. Replacing the inner minimization in Problem (B.1) with the dual problem yields the desired result. □

B.3. Optimal Decision Trees

Proof of Theorem 9. Since the set of constraints affected by applying Theorem 3 and set of constraints affected by applying Theorem 6 are disjoint, we can simply apply them both simultaneously to yield the stated result. □

The full robust-in-both optimal tree formulation is therefore

$$\min \quad \sum_{k=1}^{K} f_k - \sum_{k=1}^{K} \lambda_k d_k \tag{B.2a}$$

$$\text{s.t.} \quad g_k = \sum_{i=1}^{n} \frac{1-y_i}{2} z_{ik} \qquad k = 1,\ldots,K, \tag{B.2b}$$

$$h_k = \sum_{i=1}^{n} \frac{1+y_i}{2} z_{ik} \qquad k = 1,\ldots,K, \tag{B.2c}$$

$$f_k \le g_k - \Gamma\mu_{1,k} - \sum_{i=1}^{n} \nu_{1,ik} + M[w_k + (1-c_k)] \qquad k = 1,\ldots,K, \tag{B.2d}$$

$$f_k \le h_k - \Gamma\mu_{2,k} - \sum_{i=1}^{n} \nu_{2,ik} + M[(1-w_k) + (1-c_k)] \qquad k = 1,\ldots,K, \tag{B.2e}$$

$$f_k \ge g_k + \Gamma\mu_{3,k} + \sum_{i=1}^{n} \nu_{3,ik} - M[(1-w_k) + (1-c_k)] \qquad k = 1,\ldots,K, \tag{B.2f}$$

$$f_k \ge h_k + \Gamma\mu_{4,k} + \sum_{i=1}^{n} \nu_{4,ik} - M[w_k + (1-c_k)] \qquad k = 1,\ldots,K, \tag{B.2g}$$

$$\mu_{m,k} + \nu_{m,ik} \ge -y_i z_{ik} \qquad i = 1,\ldots,n,\ k = 1,\ldots,K,\ m = 1,4, \tag{B.2h}$$

$$\mu_{m,k} + \nu_{m,ik} \ge y_i z_{ik} \qquad i = 1,\ldots,n,\ k = 1,\ldots,K,\ m = 2,3, \tag{B.2i}$$

$$d_k = 1 \qquad k = \lceil K/2 \rceil,\ldots,K, \tag{B.2j}$$

$$d_k \le d_j \qquad k = 1,\ldots,K, \forall j \in \mathcal{P}_k, \tag{B.2k}$$

$$d_k + \sum_{l=1}^{p} a_{kl} = 1 \qquad k = 1,\ldots,K, \tag{B.2l}$$

$$\sum_{k=1}^{K} z_{ik} = 1 \qquad i = 1,\ldots,n, \tag{B.2m}$$

$$z_{ik} \le d_k \qquad i = 1,\ldots,n, k = 1,\ldots,K, \tag{B.2n}$$

$$z_{ik} \le 1 - d_j \qquad i = 1,\ldots,n, k = 1,\ldots,K, \forall j \in \mathcal{P}_k, \tag{B.2o}$$

$$\sum_{i=1}^{n} z_{ik} \ge N c_k \qquad k = 1,\ldots,K, \tag{B.2p}$$

$$c_k \ge d_k - \sum_{j \in \mathcal{P}_k} d_j \qquad k = 1,\ldots,K, \tag{B.2q}$$

$$\mathbf{a}_j^T \mathbf{x}_i + \rho + \epsilon \le b_j + (1 - z_{ik}) \qquad i = 1,\ldots,n,\ k = 1,\ldots,K,\ \forall j \in \mathcal{P}_l^u, \tag{B.2r}$$

$$\mathbf{a}_j^T \mathbf{x}_i - \rho \ge b_j + (1 - z_{ik}) \qquad i = 1,\ldots,n,\ k = 1,\ldots,K,\ \forall j \in \mathcal{P}_k^l, \tag{B.2s}$$

$$\mathbf{a}_k \in \{0,1\}^p \qquad k = 1,\ldots,K, \tag{B.2t}$$

$$0 \le b_k \le 1 \qquad k = 1,\ldots,K, \tag{B.2u}$$

$$z_{ik}, w_k, c_k, d_k \in \{0,1\} \qquad i = 1,\ldots,n, k = 1,\ldots,K, \tag{B.2v}$$

$$\mu_{m,k}, \nu_{m,ik} \ge 0 \qquad i = 1,\ldots,n,\ k = 1,\ldots,K,\ m = 1,\ldots,4. \tag{B.2w}$$

References

Ben-Tal A, Nemirovski A (2000) Robust solutions of linear programming problems contaminated with uncertain data. *Math. Programming* 88(3): 411–424.

Ben-Tal A, El Ghaoui L, Nemirovski A (2009) *Robust Optimization* (Princeton University Press, Princeton, NJ).

Ben-Tal A, Bhadra S, Bhattacharyya C, Nemirovski A (2012) Efficient methods for robust classification under uncertainty in kernel matrices. *J. Machine Learn. Res.* 13(10):2923–2954.

Bertsekas DP (1999) *Nonlinear Programming* (Athena Scientific, Belmont, MA).

Bertsimas D, Copenhaver MS (2017) Characterization of the equivalence of robustification and regularization in linear and matrix regression. *Eur. J. Oper. Res.* 270(3):931–942.

Bertsimas D, Dunn J (2017) Optimal classification trees. *Machine Learn.* 106(7):1039–1082.

Bertsimas D, King A (2015) An algorithmic approach to linear regression. *Oper. Res.* 64(1):2–16.

Bertsimas D, King A (2017) Logistic regression: From art to science. *Statist. Sci.* 32(3):367–384.

Bertsimas D, Mazumder R (2014) Least quantile regression via modern optimization. *Ann. Statist.* 42(6):2494–2525.

Bertsimas D, Sim M (2004) The price of robustness. *Oper. Res.* 52(1):35–53.

Bertsimas D, Tsitsiklis JN (2008) *Introduction to Linear Optimization,* Vol. 6 (Athena Scientific and Dynamic Ideas, Belmont, MA).

Bertsimas D, Brown DB, Caramanis C (2011) Theory and applications of robust optimization. *SIAM Rev.* 53(3):464–501.

Bertsimas D, King A, Mazumder R (2016) Best subset selection via a modern optimization lens. *Ann. Statist.* 44(2):813–852.

Bezanson J, Edelman A, Karpinski S, Shah VB (2017) Julia: A fresh approach to numerical computing. *SIAM Rev.* 59(1):65–98.

Bhattacharyya C, Pannagadatta K, Smola AJ (2005) A second order cone programming formulation for classifying missing data. Saul LK, Weiss Y, Bottou L, eds. *Proc. Adv. Neural Inform. Processing Systems* (Neural Information Processing Systems Foundation, La Jolla, CA), 153–160.

Bi J, Zhang T (2005) Support vector classification with input data uncertainty. *Adv. Neural Inform. Processing Systems* 17(1):161–168.

Biggio B, Nelson B, Laskov P (2011) Support vector machines under adversarial label noise. *Proc. Machine Learn. Res.* 20(2011):97–112.

Breiman L, Friedman J, Olshen R, Stone C (1984) *Classification and Regression Trees* (Wadsworth and Brooks, Monterey, CA).

Cortes C, Vapnik V (1995) Support-vector networks. *Machine Learn.* 20(3):273–297.

El Ghaoui L, Lanckriet GRG, Natsoulis G (2003) Robust classification with interval data. Technical Report UCB/CSD-03-1279 (University of California, Berkeley).

Fertis AG (2009) A robust optimization approach to statistical estimation problems. PhD thesis, Massachusetts Institute of Technology, Cambridge.

Friedman J, Hastie T, Tibshirani R (2001) *The Elements of Statistical Learning*, Springer Series in Statistics, Vol. 1 (Springer, Berlin).

Friedman J, Hastie T, Tibshirani R (2010) Regularization paths for generalized linear models via coordinate descent. *J. Statist. Software* 33(1):1.

Gurobi Optimization Inc. (2015) Gurobi Optimizer Reference Manual (Gurobi Optimization Inc., Beaverton, OR). Retrieved October 1, 2015, http://www.gurobi.com.

Harrington PL Jr, Zaas A, Woods CW, Ginsburg GS, Carin L, Hero AO III (2010) Robust logistic regression with bounded data uncertainties. Technical report, University of Michigan, Ann Arbor.

Huber PJ (1981) *Robust Statistics* (Wiley, New York).

Lichman M (2013) UCI machine learning repository (University of California, Irvine). Retrieved May 1, 2015, http://archive.ics.uci.edu/ml.

Livni R, Crammer K, Globerson A (2012) A simple geometric interpretation of SVM using stochastic adversaries. *Proc. Machine Learn. Res.* 22(2012):722–730.

Lubin M, Dunning I (2015) Computing in operations research using Julia. *INFORMS J. Comput.* 27(2):238–248.

Natarajan N, Dhillon IS, Ravikumar PK, Tewari A (2013) Learning with noisy labels. Burges CJC, Bottou L, Welling M, Ghahramani Z, Weinberger KQ, eds. *Proc. Adv. Neural Inform. Processing Systems 26* (Neural Information Processing Systems Foundation, La Jolla, CA), 1196–1204.

Pant R, Trafalis TB, Barker K (2011) Support vector machine classification of uncertain and imbalanced data using robust optimization. Mastorakis N, Mladenov V, Bojkovic Z, Topalis F, Psarris K, eds. *Proc. 15th WSEAS Internat. Conf. Comput.* (World Scientific and Engineering Academy and Society, Stephens Point, WI), 369–374.

R Core Team (2015) *R: A Language and Environment for Statistical Computing* (R Foundation for Statistical Computing, Vienna). Retrieved October 1, 2015, http://www.R-project.org/.

Therneau T, Atkinson B, Ripley B (2015) Rpart: Recursive partitioning and regression trees. R package version 4.1-9. Retrieved October 1, 2015, http://CRAN.R-project.org/package=rpart.

Tibshirani R (1996) Regression shrinkage and selection via the Lasso. *J. Royal Statist. Soc. Series B* 58(1):267–288.

Trafalis TB, Gilbert RC (2007) Robust support vector machines for classification and computational issues. *Optim. Methods Software* 22(1):187–198.

Wächter A, Biegler LT (2006) On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Math. Programming* 106(1):25–57.

Xu H, Caramanis C, Mannor S (2009) Robustness and regularization of support vector machines. *J. Machine Learn. Res.* 10(7):1485–1510.

Zhu J, Rosset S, Hastie T, Tibshirani R (2004) 1-norm support vector machines. *Adv. Neural Inform. Processing Systems* 16(1):49–56.

Zou H, Hastie T (2005) Regularization and variable selection via the elastic net. *J. Royal Statist. Soc. Series B* 67(2):301–320.