Journal of Machine Learning Research 21 (2020) 1-25 Submitted 5/19; Revised 9/20; Published 11/20

Stable Regression: On the Power of Optimization over Randomization in Training Regression Problems

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

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

Editor: Benjamin Recht

Abstract

We investigate and ultimately suggest remediation to the widely held belief that the best way to train regression models is via random assignment of our data to training and validation sets. In particular, we show that taking a robust optimization approach, and optimally selecting such training and validation sets, leads to models that not only perform significantly better than their randomly constructed counterparts in terms of prediction error, but more importantly, are considerably more stable in the sense that the standard deviation of the resulting predictions, as well as of the model coefficients, is greatly reduced. Moreover, we show that this optimization approach to training is far more effective at recovering the true support of a given data set, i.e., correctly identifying important features while simultaneously excluding spurious ones. We further compare the robust optimization approach to cross validation and find that optimization continues to have a performance edge albeit smaller. Finally, we show that this optimization approach to training is equivalent to building models that are robust to all subpopulations in the data, and thus in particular are robust to the hardest subpopulation, which leads to interesting domain specific interpretations through the use of optimal classification trees. The proposed robust optimization algorithm is efficient and scales training to essentially any desired size.

Keywords: stability, randomization, optimization, regression, robustness, interpretability

1. Introduction

In the practice of Machine Learning and Statistics today, the following paradigm is employed to train regression models (Mosteller and Tukey (1968)): From the given data, a

random subset is taken and placed to the side, to be used as a testing set. On the remaining data, we now randomly split it into training and validation sets, whereby we train the model (i.e., estimate coefficients and tune regularization parameters) on the training data, and then assess its accuracy on the validation data. After potentially several iterations of this process, the final accuracy of the model is then reported on its performance on the held out test set. Conceptually and practically, this procedure is straightforward, and hence is widely adopted. A question that is often overlooked, however, is how sensitive the model is to our random assignment of the data to training and validation splits. Not only might the numerical values of the learned coefficients change from one choice of split to another, but in fact which choice of coefficients themselves are nonzero, in other words the support, might change. This problem becomes even more pronounced in settings where there exists a paucity of data, such as in bioinformatics or genetics, as the choice of split can lead to massive differences in the kinds of models that are ultimately constructed. Such variability greatly hurts the interpretability of the resulting model, and brings into question the utility and applicability of the model's predictions. In this paper, we explore whether optimization rather than randomization can be employed to improve both accuracy and model stability, i.e., low variability in coefficients, predictions, and support. The theme of optimization vs randomization was first explored in Bertsimas et al. (2015) in the context of assigning control groups to create new homogenous groups. Bertsimas et al. (2015) demonstrated that the improvement can be orders of magnitudes better than the corresponding randomized approach. In this paper, we demonstrate that there is significant benefit in applying optimization to training regression models.

1.1 Literature

The idea of using randomization in determining training and validation splits is so widely held that indeed to the best of our knowledge, no other work exists on the subject of using alternative methods to select such splits. The closest comes from (Garbade (1977)) in which they investigate the stability of regression coefficients over choices of different splits via a variety of statistical hypothesis tests. In contrast, rather than investigating the stability of existing methods, we develop a general procedure for taking existing regression methods, and "stabilizing" them. Viewed from another perspective, our optimization approach to training, which identifies the "hardest" training set as a by product, is equivalent to guaranteeing that the learned model is robust to all sub-populations in the data (we expand on this perspective in the third part of the paper). From this angle, Duchi and Namkoong (2019) approach the problem of learning models that are robust to all subpopulations in the data from a distributionally robust framework. The advantage of our approach is that it is nonparametric, and significantly more computationally efficient as their problem is posed as a distributionally robust stochastic optimization problem whereas ours reduces to a linear or quadratic optimization problem depending on the type of regularization used.

1.2 Contributions and Structure

In this paper, we propose a new methodology for training regression models that is based on using optimization rather than randomization to select training/validation sets. We show that this optimization approach to training leads to models that have lower prediction error, significantly lower prediction and coefficient variability, as well as are far more effective at recovering the true support of a given data set. We also show that the optimization approach to training is equivalent to building models that are robust to all subpopulations in the data, and thus in particular are robust to the hardest subpopulation, which leads to interesting domain specific interpretations. Finally, all of this is accompanied by an efficient algorithm that scales this optimization approach to training to essentially any desired size.

The structure of the paper is as follows. In Section 2, we rigorously define what we mean by training a regression model using randomization, and then contrast it by introducing a robust optimization approach to accomplish the same. In Section 3, we derive an efficient algorithm for solving the robust optimization problem posed in the previous section. In Section 4, we describe our testing methodology, and then present computational results for both unregularized and regularized regression across four metrics: prediction error (MSE), standard deviation of prediction error (i.e., variability of prediction error with respect to the choice of test split), coefficient standard deviation (i.e., how spread out the coefficients are around their mean, where distance is measured via the typical Euclidean formula for vectors, with respect to the choice of test split), and hyperparameter standard deviation (variability of the hyperparameter with respect to the choice of test split). Note this last metric only applies for regularized regression. In Section 5, we expand our analysis to recovering the true support of a given data set. In Section 6, we draw the connection to subpopulation robustness, and then focus on interpreting the identified "hardest" training set. In Section 7, we explore the scalability of our method and demonstrate that it scales to problems of essentially any size. In Section 9, we summarize our results and report our conclusions.

2. The Robust Optimization Approach

In this section, we formally introduce the randomized approach towards training general regression models, and then followup by proposing an optimization approach to accomplish the same.

2.1 Traditional Randomization Approach

Given labeled points $(x_i, y_i), i = 1, ..., n$ where $x_i \in \mathbb{R}^p$ and $y_i \in \mathbb{R}$, the linear regression problem is to find a $\beta \in \mathbb{R}^p$ that minimizes the sum $\sum_{i=1}^{n} |y_i - x_i^T \beta|$ (note that one can also use the squared l_2 loss function $\sum_{i=1}^{n}(y_i - x_i^T \beta)^2$, however in this paper we focus on the ℓ_1 loss given above). In other words, the linear regression problem is given by the following

optimization problem:

$$\min_{\beta} \sum_{i=1}^{n} |y_i - x_i^T \beta|$$

To train such a model, one would now typically randomly partition the points $1, \ldots, n$ into two subsets, A_{train} and A_{val} where $A_{train} \cup A_{val} = \{1, ..., n\}$, $A_{train} \cap A_{val} = \emptyset$, and where $\frac{|A_{train}|}{n} \approx 0.7$, $\frac{|A_{val}|}{n} \approx 0.3$, although other proportions such as 50/50, 60/40, 90/10, etc. are also used.

Training the model would then amount to using the training data, i.e., the subset of points $(x_i, y_i), i \in A_{train}$ to find an optimal β^* via solving the following optimization problem:

$$\beta^* = \arg\min_{\beta} \sum_{i \in A_{train}} |y_i - x_i^T \beta|,$$

and then using β^* to evaluate performance on the validation set, i.e., the subset of points $(x_i, y_i), i \in A_{val}$ typically via:

$$\frac{1}{|A_{val}|} \sum_{i \in A_{test}} d(y_i, x_i^T \beta),$$

where $d(y_i, x_i^T \beta) = |y_i - x_i^T \beta^*|$ or $d(y_i, x_i^T \beta) = (y_i - x_i^T \beta^*)^2$.

As the standard regression model can be prone to over-fitting on the training data (and hence potentially have low out of sample accuracy), typically what is done in practice is regularization or penalization is often applied to the coefficients of the linear regression model (Tibshirani (1994)). Three popular regularized regression models are given below:

1. Lasso Regression proposed in Tibshirani (1994):

$$\min_{\beta} \sum_{i=1}^{n} |y_i - x_i^T \beta| + \lambda \sum_{i=1}^{p} |\beta_i|.$$

2. Ridge Regression discussed in Hoerl and Kennard (1970)):

$$\min_{\beta} \sum_{i=1}^{n} |y_i - x_i^T \beta| + \lambda \sum_{i=1}^{p} \beta_i^2.$$

3. Elastic Net Regression proposed in Zou and Hastie (2005):

$$\min_{\beta} \sum_{i=1}^{n} |y_i - x_i^T \beta| + \lambda_1 \sum_{i=1}^{p} |\beta_i| + \lambda_2 \sum_{i=1}^{p} \beta_i^2.$$

The additional complexity that arises when training a regularized regression model is the need to also tune the regularization parameter λ. This is done via the same procedure

outlined above, except that it is now applied over a sequence of $\lambda's$. Finally, that pair of λ and β that yielded the smallest error on the validation set is applied to make predictions on an independent test set and the error is reported via mean squared error (Zou and Hastie (2005)). Note that typically the way such test sets are acquired, is by first selecting a random 10% of the data, putting it to the side as the designated test set, and then running the aforementioned testing/validation procedure on the remaining 90% of the data.

In practice, in order to get a more stable estimate of the testing error, the above procedure is applied k times, i.e., we partition our non-test data into k pieces, assign one of them to be our validation set, the remaining $k-1$ to be our training set, apply the aforementioned procedure, and then one by one swap out which piece we designate to be our validation set and repeat. In the end, we take that pair of λ and β that yielded the smallest error across all the validation sets, and use them to make predictions on an independent test set. This procedure is known as k-fold cross validation.

2.2 Robust Optimization Approach

We now propose a robust optimization based approach to train a general regularized regression model (all of the above models are special cases of the one we present below, including the unregularized regression model which can be recovered by setting $\lambda = 0$). Namely, rather than randomly assigning data to training and validation sets, we instead integrate this selection right into the optimization problem directly, now solving instead the following problem:

$$\min_{\beta} \max_{z \in \mathcal{Z}} \sum_{i=1}^{n} z_i |y_i - x_i^T \beta| + \lambda \sum_{i=1}^{p} \Gamma(\beta_i) \quad \text{with} \quad \mathcal{Z} = \left\{ z : \sum_{i=1}^{n} z_i = k, \quad z_i \in \{0,1\} \right\}, \quad (1)$$

where $\Gamma(\cdot)$ is the chosen regularization function. At an optimal solution of (1), each z_i will be equal to either 0 or 1, and thus is to be interpreted as an indicator variable, indicating which point (x_i, y_i) belongs to the training set and which to the validation set. More precisely, if $z_i = 1$, then point (x_i, y_i) is assigned to the training set, otherwise it it is assigned to the validation set. The number k indicates the desired proportion between the size of the training and validations sets. Namely, by setting $k = 0.7n$ we recover the previously discussed 70/30 training/validation split and by setting $k = 0.5n$ we recover the 50/50 training/validation split, etc. As the inner maximization problem is linear in z, the problem is equivalent to optimizing over the convex hull of $\mathcal{Z}$

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

Thus, Problem (1) is equivalent to

$$\min_{\beta} \max_{z \in \text{conv}(\mathcal{Z})} \sum_{i=1}^{n} z_i |y_i - x_i^T \beta| + \lambda \sum_{i=1}^{p} \Gamma(\beta_i) \quad \text{with} \quad \text{conv}(\mathcal{Z}) = \left\{ z : \sum_{i=1}^{n} z_i = k, 0 \le z_i \le 1 \right\}, \quad (2)$$

Problem (2) belongs to the class of robust optimization problems, see Bertsimas et al. (2011) for a review. By training across all possible allocations of these z_i's, this results in a model that is explicitly built to do well not just over one training set of size k, as is typical, but over all possible training sets of size k, and hence in particular over the hardest training set of size k.

3. An Efficient Algorithm

In this section, we apply techniques from robust optimization to solve Problem (2) efficiently. To alleviate the multiplication of variables (namely the product of z_i with $|y_i - x_i^T\beta|$) we take the linear optimization dual of the inner maximization problem, i.e.,

$$\max_{z_i} \sum_{i=1}^n z_i|y_i - x_i^T\beta| \quad \text{subject to} \quad \sum_{i=1}^n z_i = k, \quad 0 \le z_i \le 1$$

by introducing the dual variable θ for the first constraint and the dual variables u_i for the second set of constraints to arrive at:

$$\min_{\theta,u_i} k\theta + \sum_{i=1}^n u_i \quad \text{subject to} \quad \theta + u_i \ge |y_i - x_i^T\beta|, \quad u_i \ge 0.$$

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

$$\min_{\beta,\theta,u_i} k\theta + \sum_{i=1}^n u_i + \lambda \sum_{i=1}^p \Gamma(\beta_i) \quad \text{subject to} \quad \theta + u_i \ge y_i - x_i^T\beta, \quad \theta + u_i \ge -(y_i - x_i^T\beta), \quad u_i \ge 0. \tag{3}$$

This is a linear optimization problem for the case of the Lasso or no regularization, or a convex quadratic optimization problem for the case of Ridge or Elastic Net regularization. In both cases, Problem (3) can be solved by commercial optimization software in very high dimensions. Having an efficient way to solve Problem (1), we proceed to comparing its performance to the typical randomization approach using computational experiments.

4. Computational Experiments: Regression

In this section, we present computational results comparing the typical randomized approach to the proposed optimization based approach for training, for both regularized and unregularized regression across four metrics: prediction error (MSE), standard deviation of prediction error, coefficient standard deviation, and hyperparameter standard deviation. For unregularized regression the validation set serves no purpose as there are no hyperparameters to select. The reason we included it in the experiments is to compare optimization versus randomization when they are exposed to the same number of points. For regularized

regression we also include results for various regimes of cross validation. The rationale for including these results, besides the fact that cross validation is widely used in practice, is that the robust optimization approach implicitly has access to the entire training set, while the randomized approach uses only a single training split. Since cross validation ends up examining all of the training data, this is a fairer comparison. Finally, as explained previously, we report results on the ℓ_1 loss, but for the sake of completeness, all results were rerun using the ℓ_2 loss and they were very similar to the ℓ_1 loss both in direction and magnitude.

4.1 Testing Methodology

To compare the optimization approach (1) to training regression models to the typical randomization approach, we employ the following methodology:

1. First, we focus our results on optimally training unregularized regression and Lasso regression (all of the results presented below were also run for Ridge and Elastic Net Regression and results were similar).

2. We collected 10 data sets from the UCI Machine Learning Repository (Dua and Taniskidou (2017)): Abalone, Auto MPG, Computer Hardware, Concrete, Ecoli, Forest Fires, Glass, Housing, Space Shuttle, Breast Cancer Wisconsin (Diagnostic). We also generate two large-scale synthetic data sets in the following way: The input samples $X = (x_1, x_2, ..., x_n)$ are drawn iid with $x_i \sim N(0, \Sigma), \quad \forall i = 1, ..., n$ with $\sigma_{ij} = \delta_{ij}$, where δ_{ij} is the Kronecker delta function, i.e., $\delta_{ij} = 1$ if $i = j$ and 0 otherwise. The noise components for ϵ_i are similarly drawn from $\epsilon_i \sim N(0, 1), \quad \forall i = 1, ..., n$. The unobserved true regression β_{true} has exactly $0.2p$ nonzero components at indices selected uniformly without replacement. Likewise, the nonzero coefficients in β_{true} are drawn uniformly at random from the set $\{+1, -1\}$. Finally, the response data y are generated synthetically as $y = X\beta_{\text{true}} + \epsilon$.

3. We applied the following procedure 1000 times: For a given data set, we took a random 10% subset of the data and put it to the side as the testing set. We then divided the remaining 90% of the data into training and validation sets. For the randomized procedure, we did this in one of two ways: for the first, by randomly selecting 10%, 20%, 30%, 40%, or 50% of the data and designating it as validation data, and then, respectively, leaving the remaining 90%, 80%, 70%, 60%, or 50% as training data; for the second, applying the previously described k-fold cross validation procedure for $k = 5$ and $k = 10$ (only for the regularized case). For the optimization procedure, this splitting was done via solving (3),which is equivalent to (1). We then learned the optimal coefficients β from these training/validation splits over a sequence of λ values, and then that pair of λ and β that yielded the smallest error on the validation set was selected. Note for the case of unregularized regression, λ was simply taken to be equal to zero. Finally, these coefficients were applied to the held out testing set, and results were reported on this testing set. This procedure was then repeated 1000 times, each time selecting a random 10% testing set, and proceeding as just described.

Unregularized Regression: Prediction Error (Mean Squared Error)												
Data Sets			**Randomization**					**Optimization**				
	n	p	50/50	60/40	70/30	80/20	90/10	50/50	60/40	70/30	80/20	90/10
Abalone	4177	8	5.38	5.34	5.4	5.36	5.34	5.25	**5.24**	5.32	5.34	5.33
Auto MPG	392	7	12.62	12.51	12.52	12.60	12.69	**11.88**	11.92	12.29	12.45	12.66
Comp Hard	209	6	7403.3	7087	6724.66	6727.94	7347.67	6927.36	6763.99	**6526.6**	6687.18	7320.65
Concrete	103	7	77.66	76	71.33	69.59	67.53	**61.15**	65.43	66.35	67.02	65.51
Ecoli	336	7	1.93	1.7	1.65	1.63	1.59	1.62	1.58	1.6	1.59	**1.57**
Forest Fi.	517	12	4241.2	4288.48	3531.04	4095.05	3909.26	4196.96	4262.65	**3519.97**	4087.77	3907.50
Glass	214	9	1.53	1.44	1.38	1.33	1.31	1.36	1.34	1.33	1.32	**1.31**
Housing	506	13	27.26	27.73	27.66	27.13	27.28	**26.18**	26.93	27.04	26.89	27.16
Space Sh.	23	4	0.53	0.48	0.43	0.40	0.35	0.33	**0.32**	0.34	0.34	0.34
Synth 1	30000	1000	0.10	0.10	0.09	0.09	0.09	**0.09**	0.09	0.09	0.09	0.09
Synth 2	1000	30000	672.76	494.82	461.29	434.09	441.81	484.18	449.78	453.10	**427.78**	431.31
WPBC	683	10	0.63	0.59	0.63	0.66	0.64	0.56	**0.54**	0.59	0.65	0.64

Table 1: Comparison of Prediction Error (MSE) for Randomized and Optimization Approaches for unregularized regression across five regimes, 50/50, 60/40, 70/30, 80/20, 90/10 training/validation splits respectively. The results that in all cases, the optimization approach outperforms the randomized approach by an average of 15.12%, 10.11%, 5.51%, 3.23%, 1.00%, for the 50/50, 60/40, 70/30, 80/20, 90/10 training/validation splits, respectively. The best (i.e., smallest) entry in each row has been made bold.

4. We report: prediction error (MSE), standard deviation of prediction error, standard deviation of the coefficients, and standard deviation of the hyperparameter.

4.2 Prediction Error

We first report results on prediction error. For the sake of thoroughness, we ran these experiments using 90/10, 80/20, 70/30, 60/40, and 50/50 training/validation splits. The results are reported for unregularized and regularized regression below in Tables 1 and 2. The error bars (i.e. one sample standard deviation over the 1000 independent runs) associated with these tables are reported in the appendix.

For regularized regression we also include results for 5 and 10 fold cross validation. As we are examining prediction error, in this case lower numbers are desirable as they indicate greater predictive ability.

In the very left column we have listed in alphabetical order the names of the various data sets we used from the UCI Machine Learning Repository as well as the two large scale synthetic data sets. The next two columns to the right contain the number of rows and columns, respectively, of each data set. The remaining columns store the actual results. Under the heading "Randomization" are the MSE for each of the 90/10, 80/20, 70/30, 60/40, and 50/50 (as indicated by the corresponding column label). The same holds for the numbers under the heading "Optimization", except now for the optimization procedure.

In all cases, the results indicate that the optimization approach outperforms the randomized approach. More specifically: for unregularized regression, we see an average improvement

Regularized Regression: Prediction Error (Mean Squared Error)														
Data Sets			**Randomization**							**Optimization**				
	n	p	50/50	60/40	70/30	80/20	90/10	5-CV	10-CV	50/50	60/40	70/30	80/20	90/10
Abalone	4177	8	5.33	5.4	5.4	5.38	5.34	5.38	5.33	**5.17**	5.27	5.32	5.35	5.32
Auto MPG	392	7	12.72	12.65	12.62	12.49	12.36	12.35	12.29	**12.04**	12.15	12.42	12.26	12.28
Comp Hard	209	6	6889.83	6907.53	7194.27	7878.94	7581.67	7843.44	7555.49	**6433.21**	6571.57	7069.26	7686.26	7541.71
Concrete	103	7	77.62	74.43	70.9	68.09	71.55	63.13	71.1	**62.14**	64.78	65.2	62.86	70.88
Ecoli	336	7	1.66	1.62	1.63	1.60	1.56	1.59	1.56	1.60	1.58	1.59	1.58	**1.56**
Forest Fi.	517	12	3927.89	4124.78	3974.49	4820.6	3392.12	4816.53	**3389.44**	3886.07	4101.03	3962.84	4812.52	3390.43
Glass	214	9	1.35	1.36	1.28	1.3	1.26	1.3	1.25	1.32	1.35	1.28	1.3	**1.24**
Housing	506	13	28.24	27.23	28.05	29.26	26.77	29.02	26.76	27.2	**26.52**	27.58	28.95	26.72
Space Sh.	23	4	0.5	0.46	0.41	0.37	0.38	0.37	0.38	**0.34**	0.41	0.37	0.36	0.37
Synth 1	30000	1000	0.09	0.10	0.09	0.10	0.09	0.10	0.09	**0.08**	0.09	0.08	0.09	0.09
Synth 2	1000	30000	666.45	492.53	431.86	427.92	438.22	425.65	436.18	481.92	443.35	430.08	**422.22**	435.44
WPBC	683	10	0.6	0.58	0.61	0.49	0.62	0.48	0.62	0.54	0.54	0.58	**0.48**	0.62

Table 2: Comparison of Prediction Error (MSE) for Randomized and Optimization Approaches for regularized regression across five regimes, 50/50, 60/40, 70/30, 80/20, and 90/10 training/validation splits respectively, as well as for 5 and 10 fold cross validation. The results indicate that in all cases, the optimization approach outperforms the randomized approach by an average of 10.82%, 5.31%, 3.51%, 2.15%, and 0.31% for the 50/50, 60/40, 70/30, 80/20, 90/10 training/validation splits, respectively, and 0.75% and 0.43% for 5 and 10 fold cross validation, respectively. The best (i.e., smallest) entry in each row has been made bold.

of 15.12%, 10.11%, 5.51%, 3.23%, 1.00%, for the 50/50, 60/40, 70/30, 80/20, and 90/10 training/validation splits, respectively. For regularized regression, we see an average improvement of 10.82%, 5.31%, 3.51%, 2.15%, and 0.31% for the 50/50, 60/40, 70/30, 80/20, and 90/10 training/validation splits, respectively, and 0.75% and 0.43% for 5 and 10 fold cross validation, respectively.

4.3 Standard Deviation of Prediction Error

We next report results on the standard deviation of the prediction error. As before, we ran these experiments using 50/50, 60/40, 70/30, 80/20, and 90/10 training/testing splits. The results are reported for unregularized and regularized regression below in Tables 3 and 4. The error bars associated with these tables are reported in the appendix. For regularized regression we also include results for 5 and 10 fold cross validation. As we are examining standard deviation of the prediction error, in this case lower numbers are desirable as they indicate lower variability of error, i.e., a higher likelihood that the prediction for any given sample is actually near the mean.

The structure of the table is identical to the ones presented before, except where before MSE was reported, now standard deviation of the prediction error is reported.

In all cases the results indicate that the optimization approach outperforms the randomized approach. More specifically: for unregularized regression, we see an average improvement of 111.61%, 46.74%, 7.77%, 3.36%, and 1.14% for the 50/50, 60/40, 70/30, 80/20, 90/10

Unregularized Regression: Standard Deviation of Prediction Error												
Data Sets			**Randomization**					**Optimization**				
	n	p	50/50	60/40	70/30	80/20	90/10	50/50	60/40	70/30	80/20	90/10
Abalone	4177	8	0.74	0.75	0.78	0.74	0.75	**0.68**	0.7	0.71	0.71	0.74
Auto MPG	392	7	4.19	4.14	4.03	4.15	4.3	**3.83**	3.94	3.93	4.12	4.3
Comp Hard	209	6	9991.12	9881	**9203.01**	9683.71	10292.97	9878.8	9835.49	9230.04	9770.97	10331.14
Concrete	103	7	36.54	36.62	33.27	31.08	30.29	**21.75**	23.63	27.18	28.32	28.12
Ecoli	336	7	4.28	1.8	0.46	0.45	0.46	0.43	0.43	0.44	**0.43**	0.45
Forest Fi.	517	12	7807.59	7538.78	6851.84	7526.66	7238.95	7776.73	7523.75	**6846.77**	7522.2	7237.04
Glass	214	9	1.05	0.88	0.7	0.64	0.63	**0.61**	0.64	0.64	0.61	0.62
Housing	506	13	13.19	13.18	13.32	13.35	13.62	**12.87**	12.92	13	13.29	13.59
Space Sh.	23	4	0.63	0.54	0.52	0.5	0.45	0.44	**0.42**	0.44	0.45	0.45
Synth 1	30000	1000	0.009	0.006	0.006	0.010	0.009	0.007	0.004	**0.004**	0.009	0.005
Synth 2	1000	30000	302.39	219.43	252.80	411.81	334.09	224.18	**102.28**	137.67	221.78	231.31
WPBC	683	10	0.76	0.7	0.76	0.8	0.79	0.66	**0.62**	0.7	0.78	0.79

Table 3: Comparison of Standard Deviation of Prediction Error for Randomized and Optimization Approaches for unregularized regression across five regimes, 50/50, 60/40, 70/30, 80/20, 90/10 training/testing splits respectively. The results indicate that in all cases, the optimization approach outperforms the randomized approach by an average of 111.61%, 46.74%, 7.77%, 3.36%, and 1.14% for the 50/50, 60/40, 70/30, 80/20, 90/10 training/validation splits, respectively. The best (i.e., smallest) entry in each row has been made bold.

Regularized Regression: Standard Deviation of Prediction Error														
Data Sets			**Randomization**							**Optimization**				
	n	p	50/50	60/40	70/30	80/20	90/10	5-CV	10-CV	50/50	60/40	70/30	80/20	90/10
Abalone	4177	8	0.75	0.77	0.74	0.83	0.73	0.82	0.73	**0.67**	0.70	0.69	0.79	0.73
Auto MPG	392	7	4.11	4.07	4.24	3.98	3.73	3.92	**3.69**	3.77	3.86	4.15	3.90	3.70
Comp Hard	209	6	9464.71	9628.97	9689.87	9711.63	9983.69	9645.05	9980.15	**9401.39**	9643.33	9837.88	9770.23	9934.77
Concrete	103	7	37.65	35.41	32.60	28.99	33.57	27.55	33.47	**22.82**	23.96	26.65	24.90	32.45
Ecoli	336	7	0.48	0.46	0.45	0.45	0.45	0.44	0.45	0.45	0.44	**0.43**	0.43	0.45
Forest Fi.	517	12	7371.71	7591.80	7401.56	8126.87	7404.85	8125.08	7404.47	**7343.27**	7576.11	7393.83	8122.21	7404.44
Glass	214	9	0.68	0.68	0.63	0.67	0.61	0.67	0.61	**0.60**	0.63	0.62	0.66	0.61
Housing	506	13	13.57	13.00	13.66	14.63	13.57	14.38	13.56	13.35	**12.85**	13.42	14.50	13.52
Space Sh.	23	4	0.71	0.52	0.51	0.44	0.52	0.44	0.52	**0.44**	0.52	0.48	0.46	0.52
Synth 1	30000	1000	0.005	0.005	0.004	0.007	0.004	0.005	0.004	0.004	0.004	0.003	0.005	**0.003**
Synth 2	1000	30000	265.35	192.40	182.98	207.36	206.85	167.13	172.29	102.84	90.73	132.91	**81.54**	147.99
WPBC	683	10	0.71	0.68	0.74	0.58	0.75	0.56	0.75	0.60	0.63	0.69	**0.56**	0.75

Table 4: Comparison of Standard Deviation of Prediction Error for Randomized and Optimization Approaches across five regimes for regularized regression, 50/50, 60/40, 70/30, 80/20, and 90/10 training/testing splits respectively. The results indicate that in all cases, the optimization approach outperforms the randomized approach by an average of 18.84%, 8.49%, 5.19%, 2.81%, and 0.34% for the 50/50, 60/40, 70/30, 80/20, 90/10 training/validation splits, respectively, and 1.24% and 0.36% for 5 and 10 fold cross validation, respectively. The best (i.e., smallest) entry in each row has been made bold.

Unregularized Regression: Standard Deviation of Coefficients												
Data Sets			**Randomization**					**Optimization**				
	n	p	50/50	60/40	70/30	80/20	90/10	50/50	60/40	70/30	80/20	90/10
Abalone	4177	8	0.27	0.27	0.27	0.27	0.27	0.26	**0.26**	0.26	0.26	0.27
Auto MPG	392	7	0.07	0.06	0.07	0.06	0.06	0.06	0.06	**0.06**	0.06	0.06
Comp Hard	209	6	0.09	0.08	0.07	0.07	0.07	0.08	**0.06**	0.07	0.07	0.07
Concrete	103	7	0.02	**0.02**	0.04	0.03	0.02	0.03	0.03	0.04	0.03	0.03
Ecoli	336	7	0.19	0.17	0.17	0.18	0.18	0.18	**0.17**	0.18	0.17	0.17
Forest Fi.	517	12	0.07	0.06	0.06	0.06	0.06	**0.04**	0.05	0.05	0.05	0.06
Glass	214	9	0.2	0.29	0.3	0.23	0.29	0.22	**0.13**	0.24	0.16	0.27
Housing	506	13	**0.11**	0.14	0.11	0.12	0.13	0.14	0.12	0.12	0.13	0.13
Space Sh.	23	4	0.04	0.03	0.03	0.03	0.04	0.03	**0.03**	0.03	0.03	0.03
Synth 1	30000	1000	0.93	0.93	0.92	0.92	0.93	0.92	0.92	**0.91**	0.92	0.92
Synth 2	1000	30000	2.36	3.31	2.36	3.30	3.19	2.16	3.11	**1.89**	2.96	2.89
WPBC	683	10	0.02	0.02	0.02	0.02	0.02	**0.02**	0.02	0.02	0.02	0.02

Table 5: Comparison of Coefficients Average Standard Deviation for Randomized and Optimization Approaches across five regimes for unregularized regression, 50/50, 60/40, 70/30, 80/20, 90/10 training/testing splits respectively. The results indicate that in all cases, the optimization approach outperforms the randomized approach by an average of 8.30%, 16.36%, 5.16%, 6.58%, and 1.33% for the 50/50, 60/40, 70/30, 80/20, 90/10 training/testing splits, respectively. The best (i.e., smallest) entry in each row has been made bold.

training/validation splits, respectively. For regularized regression, we see an average improvement of 18.84%, 8.49%, 5.19%, 2.81%, and 0.34% for the 50/50, 60/40, 70/30, 80/20, 90/10 training/validation splits, respectively, and 1.24% and 0.36% for 5 and 10 fold cross validation, respectively.

4.4 Coefficients Standard Deviation

In this section, we report results on the average standard deviation of the coefficients of our models. As before, we ran these experiments using 50/50, 60/40, 70/30, 80/20, and 90/10 training/validation splits training/validation splits. The results are reported in the Tables 5 and 6. The error bars associated with these tables are reported in the appendix. For regularized regression we also include results for 5 and 10 fold cross validation. As we are examining the standard deviation of the learned coefficients, in this case lower numbers are desirable as they indicate lower variability with respect to which model is chosen, which confers the advantage of greater model interpretability and plausibility.

The structure of Tables 5 and 6 is identical to the ones presented before, except where before MSE was reported, now coefficient average standard deviation is reported.

The results indicate that the optimization approach outperforms the randomized approach. More specifically: for unregularized regression, we see an average improvement of 8.30%, 16.36%, 5.16%, 6.58%, and 1.33% for the 50/50, 60/40, 70/30, 80/20, 90/10 training/testing

Regularized Regression: Standard Deviation of Coefficients														
Data Sets			**Randomization**							**Optimization**				
	n	p	50/50	60/40	70/30	80/20	90/10	5-CV	10-CV	50/50	60/40	70/30	80/20	90/10
Abalone	4177	8	**0.066**	0.077	0.075	0.842	0.864	0.840	0.860	0.070	0.071	0.073	0.834	0.846
Auto MPG	392	7	0.004	0.003	0.003	0.193	0.193	0.193	0.190	0.003	**0.003**	0.004	0.189	0.198
Comp Hard	209	6	0.005	0.007	0.008	0.227	0.218	0.226	0.216	0.005	0.005	**0.004**	0.235	0.200
Concrete	103	7	0.002	0.001	0.001	0.063	0.074	0.062	0.074	**0.001**	0.001	0.001	0.071	0.072
Ecoli	336	7	0.030	0.029	0.028	0.540	0.522	0.540	0.520	0.028	**0.027**	0.028	0.530	0.538
Forest Fi.	517	12	0.003	0.005	0.004	0.180	0.183	0.180	0.183	**0.001**	0.002	0.003	0.173	0.184
Glass	214	9	0.003	0.003	0.003	0.170	0.172	0.170	0.172	0.003	0.025	**0.003**	0.170	0.179
Housing	506	13	0.014	0.014	**0.013**	0.409	0.391	0.403	0.391	0.015	0.014	0.015	0.380	0.368
Space Sh.	23	4	0.002	0.001	0.001	0.110	0.104	0.110	0.103	**0.001**	0.001	0.001	0.105	0.102
Synth 1	30000	1000	0.310	0.653	0.305	0.655	0.654	0.652	0.651	**0.301**	0.651	0.302	0.652	0.651
Synth 2	1000	30000	1.632	3.221	1.260	3.263	3.196	3.164	3.138	1.201	2.536	**1.193**	2.561	2.569
WPBC	683	10	0.000	0.000	0.000	0.062	0.062	0.062	0.062	0.000	**0.000**	0.000	0.060	0.061

Table 6: Comparison of Coefficients Average Standard Deviation for Randomized and Optimization Approaches across five regimes for regularized regression, 50/50, 60/40, 70/30, 80/20, and 90/10 training/testing splits respectively. The results indicate that in all cases, the optimization approach outperforms the randomized approach by an average of 47.57%, 13.10%, 10.86%, 1.02%, and 0.75% for the 50/50, 60/40, 70/30, 80/20, 90/10 training/testing splits, respectively, and 0.64% and 0.95% for 5 and 10 fold cross validation, respectively. The best (i.e., smallest) entry in each row has been made bold.

splits, respectively. For regularized regression, we see an average improvement of 47.57%, 13.10%, 10.86%, 1.02%, and 0.75% for the 50/50, 60/40, 70/30, 80/20, 90/10 training/testing splits, respectively, and 0.64% and 0.95% for 5 and 10 fold cross validation, respectively.

4.5 Hyperparamter Standard Deviation

In this section, we report results on the average standard deviation of the regularization hyperparameter in our models. As before, we ran these experiments using 50/50, 60/40, 70/30, 80/20, and 90/10 training/validation splits training/validation splits, as well as for 5 and 10 fold cross validation. The results are reported in Table 7. The error bars associated with Table 7 are reported in the appendix. As we are examining the standard deviation of the chosen hyperparameter, in this case lower numbers are desirable as they indicate lower variability with respect to which model is chosen, which confers the advantage of greater model interpretability and plausibility.

The structure of Table 7 is identical to the ones presented before, except where before MSE was reported, now hyperparameter average standard deviation is reported.

The results indicate that the optimization approach outperforms the randomized approach. More specifically: we see an average improvement of 5.83%, 14.39%, 12.49%, 37.71%, and 29.10% for the 50/50, 60/40, 70/30, 80/20, 90/10 training/testing splits, respectively, and 26.19% and 22.15% for 5 and 10 fold cross validation, respectively.

Regularized Regression: Hyperparameter Standard Deviation														
Data Sets			**Randomization**							**Optimization**				
	n	p	50/50	60/40	70/30	80/20	90/10	5-CV	10-CV	50/50	60/40	70/30	80/20	90/10
Abalone	4177	8	0.402	0.420	0.412	0.431	0.367	0.369	0.336	0.319	0.268	0.217	0.095	**0.095**
Auto MPG	392	7	0.366	0.395	0.375	0.373	0.383	0.367	0.374	0.378	0.390	**0.348**	0.377	0.387
Comp Hard	209	6	0.224	0.200	**0.176**	0.192	0.230	0.192	0.226	0.330	0.332	0.342	0.355	0.324
Concrete	103	7	0.333	0.326	0.345	0.369	0.349	0.369	**0.321**	0.371	0.401	0.374	0.369	0.392
Ecoli	336	7	0.361	0.378	0.381	0.401	0.414	0.341	0.369	0.281	**0.243**	0.257	0.279	0.336
Forest Fi.	517	12	0.420	0.400	0.389	0.417	0.412	0.336	0.407	0.336	**0.219**	0.305	0.330	0.397
Glass	214	9	**0.362**	0.381	0.401	0.405	0.412	0.404	0.399	0.387	0.399	0.363	0.370	0.374
Housing	506	13	0.429	0.409	0.407	0.399	0.435	0.397	0.410	**0.349**	0.382	0.402	0.422	0.370
Space Sh.	23	4	0.394	0.367	0.402	0.378	0.363	0.374	0.361	**0.352**	0.362	0.402	0.382	0.402
Synth 1	30000	1000	0.386	0.418	0.370	0.400	0.411	0.393	0.409	0.227	0.264	0.185	**0.171**	0.283
Synth 2	1000	30000	0.420	0.423	0.400	0.263	0.274	0.261	0.269	0.233	0.158	0.156	0.188	**0.113**
WPBC	683	10	0.391	0.399	**0.366**	0.385	0.405	0.385	0.405	0.404	0.389	0.382	0.399	0.405

Table 7: Comparison of Hyperparameter Average Standard Deviation for Randomized and Optimization Approaches across five regimes for regularized regression, 50/50, 60/40, 70/30, 80/20, and 90/10 training/testing splits respectively. The results indicate that in all cases, the optimization approach outperforms the randomized approach by an average of 5.83%, 14.39%, 12.49%, 37.71%, and 29.10% for the 50/50, 60/40, 70/30, 80/20, 90/10 training/testing splits, respectively, and 26.19% and 22.15% for 5 and 10 fold cross validation, respectively.

5. Recovering Support

In this next section, we explore which method of training, randomization or optimization, is more effective at recovering the true support of a given data set.

To do so, we employ the mixed integer methodology proposed by (Bertsimas et al. (2016c)). Namely, we append to each of our original problems an L_0 constraint, which enforces that exactly k coefficients are nonzero. The specific way we implement this in the case of using optimization to train is via:

$$\min_{\beta} \max_{z_i} \sum_{i=1}^{n} z_i |y_i - x_i^T \beta| + \lambda \sum_{i=1}^{p} \Gamma(\beta_i) \quad \text{subject to} \quad \sum_{i=1}^{n} z_i = k, \quad \sum_{i=1}^{p} \delta_i = s, \quad |\beta_i| \leq M\delta_i$$
$$\delta_i \in \{0,1\}, \quad 0 \leq z_i \leq 1. \tag{4}$$

and in the case of using randomization to train via:

$$\min_{\beta} \sum_{i \in A_{train}}^{n} |y_i - x_i^T \beta| + \lambda \sum_{i=1}^{p} \Gamma(\beta_i) \quad \text{s.t.} \quad \sum_{i=1}^{p} \delta_i = s, \quad |\beta_i| \leq M\delta_i, \qquad \delta_i \in \{0,1\}. \tag{5}$$

where in both cases above, M is chosen to be some very large number and the δ_i represent which coefficients are nonzero. These binary coefficient "switches" in conjunction with the constraint $\sum_{i=1}^{p} \delta_i = s$ then efficiently implement the L_0 constraint. Note that the resulting problems are now mixed integer optimization (MIO) problems, which are nowadays highly

solvable due to the remarkable progress in MIO solvers in recent years. The tractability of such problem is discussed further in (Bertsimas et al. (2016c)).

5.1 Testing Methodology

Using these formulations, we next employ the following procedure 1000 times:

1. We first generate a synthetic data set in the following manner. The input samples $X = (x_1, x_2, ..., x_n)$ are drawn iid with $x_i \sim N(0, \Sigma), \quad \forall i = 1, ..., n$ with $\sigma_{ij} = \delta_{ij}$, where δ_{ij} is the Kronecker delta function, i.e., $\delta_{ij} = 1$ if $i = j$ and 0 otherwise. The noise components for ϵ_i are similarly drawn from $\epsilon_i \sim N(0, 1), \quad \forall i = 1, ..., n$. The unobserved true regression β_{true} has exactly s nonzero components at indices selected uniformly without replacement. Likewise, the nonzero coefficients in β_{true} are drawn uniformly at random from the set $\{+1, -1\}$. Finally, the response data y are generated synthetically as $y = X\beta_{\text{true}} + \epsilon$.

2. The generation of these data sets is repeated for increasing values of n (i.e., generating data sets containing more sample points), with the aim being to elucidate the behavior of support recovery as a function of n.

3. We then, in the case of using optimization to train, solve Problem (4) and in the case of using randomization to train, solve Problem (5). Note that in this section when we say randomization, we mean cross-validation.

4. We then apply the same training/testing/validation methodology described several times before, using the 70/30 split, for two kinds of experiments: the first where we know the true value of s (we have access to this because we generated the data), and the second where we used cross-validation to estimate the value of s.

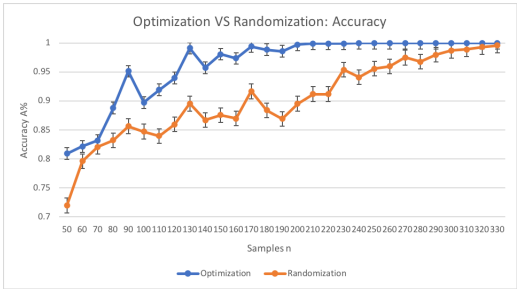
5. Finally, we compare performance using two support recovery metrics: the accuracy and false alarm rate of a certain solution β^* in recovering the correct support as:

$$A\% = 100 \times \frac{|\text{supp}(\beta^{\text{true}}) \cap \text{supp}(\beta^*)|}{s},$$

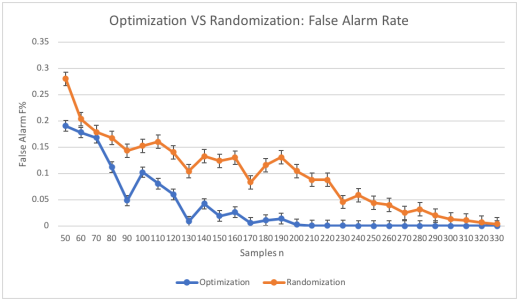
and

$$F\% = 100 \times \frac{|\text{supp}(\beta^*) - \text{supp}(\beta^{\text{true}})|}{|\text{supp}(\beta^*)|}.$$

Perfect support recovery occurs only when β^* tells the whole truth ($A\% = 100$) and nothing but the truth ($F\% = 0$).



(a) Comparison of support recovery accuracy as a function of n for randomized and optimization approaches for known s. Note that the optimization approach begins with a higher accuracy, and maintains this advantage as we increase n, eventually reaching a perfect accuracy score



(b) Comparison of support recovery false alarm rate as a function of n for randomized and optimization approaches for known s. Note that the optimization approach begins with a lower false alarm rate, and maintains this advantage as we increase n, eventually reaching a perfect false alarm rate score.

Figure 1

5.2 Recovering Support: Results

5.2.1 Knowing the True Sparsity

We first present results when the true s is known. Figure 1(a) below plots Accuracy A% as a function of the number of samples n while Figure 1(b) plots the False Alarm Rate F% as a function of the number of samples n.

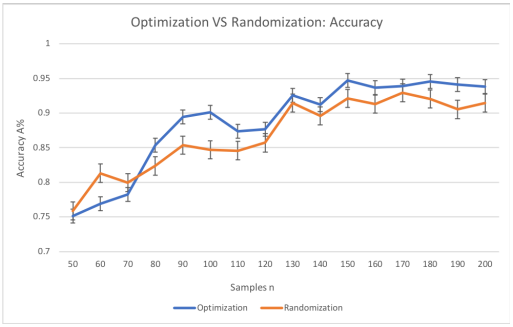
Note that for Accuracy A%, that training via optimization results in a significant initial advantage, which is maintained throughout as we increase n, before eventually reaching a perfect accuracy score. The same kind of advantage and behavior is also present for the False Alarm Rate F% plotted in Figure 1(b).

5.2.2 Unknown Sparsity

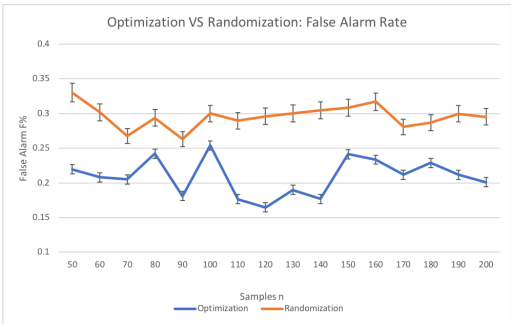
We next present results for when the true s is unknown, and is instead estimated via cross-validation. Figure 2(a) plots Accuracy A% as a function of the number of samples n while Figure 2(b) plots the False Alarm Rate F% as a function of the number of samples n.

Note that for Accuracy A%, that training via randomization results in a slight initial advantage, which is quickly made up by training via optimization, and then that lead is maintained throughout as we increase n, before eventually approaching a near perfect accuracy score.

For the False Alarm Rate F%, note that training via optimization results in a significant initial advantage, which is maintained throughout as we increase n, resulting in a gradual decrease in the false alarm rate.



(a) Comparison of support recovery accuracy as a function of n for randomized and optimization approaches for unknown s. Note that the randomization approach begins with a slight initial advantage, which is quickly made up by the optimization approach, and then that lead is maintained throughout as we increase n, before eventually approaching a near perfect accuracy score.



(b) Comparison of support recovery false alarm rate as a function of n for randomized and optimization approaches for unknown s. Note that the optimization approach begins with a lower false alarm rate, and maintains this advantage as we increase n, resulting in a gradual decrease in the false alarm rate score.

Figure 2

6. Subpopulation Robustness and Interpreting the "Hardest Training Set"

The results up to this point have focused on demonstrating that there are several advantages in training a regression model against the "hardest" training set: lower mean squared error, lower prediction standard deviation, lower coefficient standard deviation, and superior support recovery. In this next section, we focus on an equally important advantage: the "hardest" training set usually has a very interesting interpretation, as opposed to a randomly selected training set which, by definition, is uninterpretable.

For example, when applying a medication to some population, we are often interested in identifying that subpopulation for which the drug will be least effective, or for which subpopulation the drug's efficacy will be least certain. The same holds when structuring a portfolio: in a universe of potential assets under consideration, one is interested in identifying that subset of assets that will most underperform or expose the portfolio to the greatest level of risk. There are several other examples where we are interested in identifying the worst subpopulation in some larger population, of which the previous examples are just two.

Not only does training against the hardest training set often have interesting interpretations, as we discuss below, but it also guarantees that we are robust to all possible subpopulations of size equal to the training set. This follows from our earlier observation that by training across all possible allocations of the z_i's, this results in a model that is explicitly built to do well not just over one training set of size k, as is typical, but over all possible training sets of size k, and hence in particular over the hardest training set of size k. This perspective

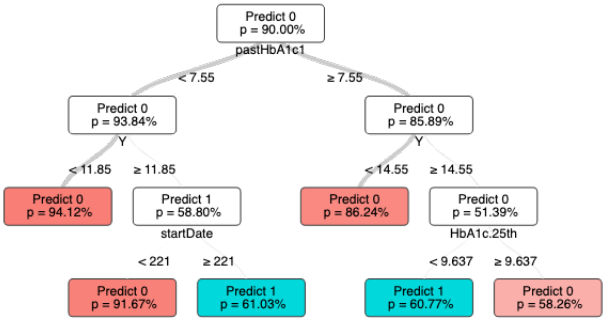


Figure 3: Classification Tree for Diabetes Data Set. A 1 corresponds to a point that belongs to the hardest training set, a 0 corresponds to a point that belongs to the validation set.

on what the optimization methodology is actually accomplishing gives further insight into its superior performance.

In what follows, we investigate the interpretation of the hardest training set in a large (more than 100,000 observations) diabetes data set, previously examined in (Bertsimas et al. (2016b)), as well as in a emergency department arrivals data set provided by the Beth Israel Deaconess Medical Center, previously examined in (Bertsimas et al. (2016a)).

6.1 Testing Methodology

1. For both the diabetes and emergency department data sets, we use all the data available, and run the optimization procedure to discover the "hardest" training set. Note that this exactly corresponds to those points x_i for which the corresponding z_i was equal to 1 at an optimal solution.

2. All the other points for which z_i was equal to 0 correspond to our validation set.

3. We now turn the problem into a classification problem, where we use the original X matrix paired with the previously computed z vector of 1's and 0's as labels. We then run optimal classification trees, which were introduced in (Bertsimas and Dunn (2017)), to attempt to gain a deeper understanding of which features contribute towards the designation of a given point to the "hardest" training set.

6.2 Diabetes Data Set Results

Running the aforementioned procedure on the diabetes data set, we arrive at the following classification tree, which had an excellent associated accuracy rate of 90%.

Looking at the tree in Figure 3, three critical points stand out. First, by examining the path from the root down to the left most leaf on the right side of the tree, we see that

fairly large gaps between the patient's past HbA1C score, the patient's current HbA1C score, and the 25th percentile of the patient's historical HbA1C score is highly predictive of it being difficult to predict that patient's next HbA1C score. This makes sense, because if the patient's previous HbA1C measurements have varied significantly across the board, this implies the patient's blood sugar levels are highly mercurial, and thus by definition difficult to predict. This might seem obvious to a person, but it is a good sanity check that the model was able to detect this common sense relationship on its own. Second, by examining the path from the root down to the right most leaf on the left side of the tree, we see that a high current HbA1C score for a patient that's been on medication for greater than 221 days (i.e., a long time) is also highly predictive of it being difficult to predict that patient's next HbA1C score. This also makes sense, because if a patient has been on medication for a long time, and his or her HbA1C score is still very high, by definition that means the treatment isn't working. This could either be due to the patient not following the instructions carefully enough or them having some genetic makeup that renders them irresponsive to it. It will of course be more difficult to predict the HbA1C score of a patient who is taking his or her medication erratically, as it will also be for a patient who has a severe form of the disease and who's body isn't responding to treatment. This is an interesting relationship identified by the model, and one that could be further explored by an endocrinologist. Third, when we extracted the list of top 10 features from the tree that were most predictive of a patient's presence in the hardest training set, we saw the feature corresponding to kidney complication was present. This makes sense, because if the diabetes has progressed to the point where the kidneys are starting to malfunction, that means things in the body are going haywire, and thus greater variability in pancreatic performance is to be expected. This again is an interesting relationship that could be further explored by an endocrinologist (in particular why the emphasis on a complication with the kidney as opposed to a complication with any other organ).

6.3 Emergency Department Data Set Results

Running the aforementioned procedure on the emergency department data set, we arrive at the following classification tree, which had an associated accuracy rate of 71%.

Looking at the tree in Figure 4, three critical points stand out. First, by examining the path from the root down to the second to left most leaf on the right side of the tree, we see that during the weekends of college spring break, the regression models are able to do a strong job in predicting the number of arrivals to the emergency department. This aligns with feedback that we have received from the Beth Israel Deaconess Medical Center (BIDMC). Namely, during spring break, students tend to get more rowdy, especially over the weekends, accidents happen, and consequently there is a very predictable trend of high number of arrivals to the emergency department. Second, by examining the second node on the left hand side of the tree, we see that the presence of a Red Sox home game, in conjunction with the number of people in the city (PastDay48_96), is very important in determining how well we will be able to predict arrivals to the emergency department. This again aligns with feedback we received from BIDMC. Namely, if it is a big game and there

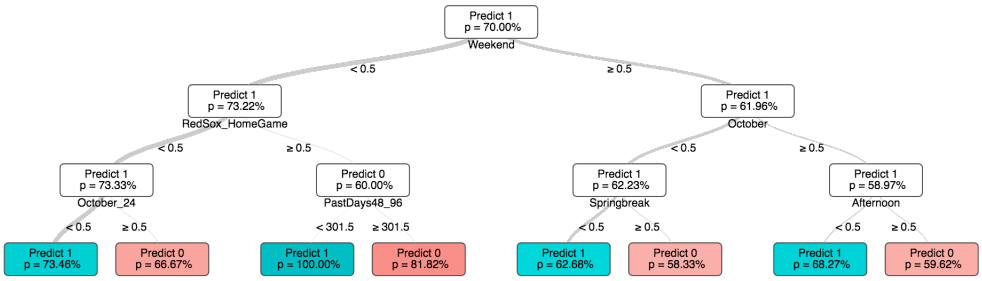


Figure 4: Classification Tree for Emergency Department Data Set. A 1 corresponds to a point that belongs to the hardest training set, a 0 corresponds to a point that belongs to the validation set.

are many people in the city (and especially if they are playing the New York Yankees), fights are almost certain to break out, and the emergency department knows to brace themselves for a high load of incoming patients. On the other hand, if it is a less important game and there are less people in town, its not as easy to say whether there will be a large influx to the hospital, as it might depend on, for example, the specifics of the game as well as the temperature on the day of the game. Finally, by examining the rightmost branch of the tree, we see that non-afternoon weekends in October also tend to be difficult to predict. Each of these factors individually are things BIDMC has warned us about. Namely, October is a very packed month for holidays (Halloween, various food festivals such as Octoberfest, etc.) and thus people tend to eat and drink excessively, which stresses their system. At the same time, visits to the emergency department in the evenings or early mornings, as well as over the weekend, are less common because it inconveniences people, and they don't want to spend their free time in a hospital. Thus, the people showing up at those odd hours are mainly the ones that are really having an emergency that cannot wait. It is therefore plausible that there is some connection between the three factors. To the best of our knowledge, BIDMC has not yet considered the effect of them together, and thus this could be an interesting direction to explore for them.

Overall, the results of the optimal tree experiments are, in our opinion, compelling. We deliberately picked examples for which we had some intuition or external experts we could consult with, so that we would have an independent manner in which we could validate the results. Of course, the real power of our method will come from applying it to domains for which we have no prior intuition or for which our understanding is incomplete, so that analyses such as the one above could help elucidate important unkown relationships and direct future research efforts.

7. Scalability

In this next section, we briefly comment on the scalability of the optimization approach. To do so, we compare the performance of training a regression using randomization to that of training a regression via optimization for each of the 10 data sets in the UCI Machine Learning Repository, as well as the two large scale synthetic data sets from earlier.

7.1 Running Times

To compare the performance of training a regression using randomization to that of training a regression via optimization we employ the following methodology:

1. For each data set, we time how long it takes to train a regression model using randomization for that data set using 50/50, 60/40, 70/30, 80/20, 90/10 training/validation sets.
2. We do the same for the optimization procedure of training.
3. This is done 1000 times for each regime, and then the resulting timings are averaged to get a stable estimate of the required compute time.
4. Note that for each regime listed above, one run corresponds to the total time to first split the data into training/validation sets, and then solve the resulting regression problem on the training data. For the randomization scheme, this happens in two parts (the first being trivial, just a random assignment of points to training and validation sets) and the second the actual solving of the model. For the optimization procedure, this all happens in one step.

We report the results in Table 7. We again employ the same organization scheme used before, except now in each cell, we have listed the amount of time in seconds it took to both split the data into training/validation sets and then solve the resulting model for the corresponding data set and training/validation proportion, averaged over 1000 runs. Overall, the results indicate that the optimization approach takes about 2 to 3 times longer than the randomization procedure. More specifically, for the 50/50 split it takes on average 2.0601 times longer, for the 60/40 split it takes on average 2.8044 times longer, for the 70/30 split it takes on average 2.6538 times longer, for the 80/20 split, on average 1.0276 times longer, and finally for the 90/10 split, on average 1.0192 times longer.

We expected that the optimization procedure would take longer than the randomization procedure, because the way it decides which parts of the data get assigned to training and which to validation is more sophisticated than the trivial procedure employed by the randomization approach. It is encouraging, however, that the slowdown is only 2 to 3 times as regression problems can be solved very quickly and hence we would expect this optimization procedure to be very usable in practice.

Randomization vs Optimization: Timings (in seconds)												
Data Sets			**Randomization**					**Optimization**				
	n	p	50/50	60/40	70/30	80/20	90/10	50/50	60/40	70/30	80/20	90/10
Abalone	4177	8	3.33E-03	3.86E-03	4.46E-03	2.42E+00	3.06E+00	6.87E-03	7.32E-03	1.10E-02	1.37E+00	1.56E+00
Auto MPG	392	7	1.01E-02	1.69E-02	1.73E-02	2.82E-02	3.83E-02	4.99E-02	6.86E-02	6.68E-02	4.22E-02	4.22E-02
Comp Hard	209	6	1.27E-03	2.48E-03	4.85E-03	1.03E-02	1.30E-02	1.35E-03	2.80E-03	5.67E-03	1.63E-02	1.71E-02
Concrete	103	7	2.77E-03	4.41E-03	5.41E-03	4.22E-03	4.36E-03	5.18E-03	8.15E-03	8.89E-03	5.93E-03	6.85E-03
Ecoli	336	7	2.08E-03	4.19E-03	5.18E-03	2.29E-02	2.66E-02	3.47E-03	6.86E-03	7.61E-03	1.89E-02	2.29E-02
Forest Fi.	517	12	5.48E-02	8.02E-02	1.19E-01	5.63E-02	7.17E-02	4.49E-01	5.42E-01	7.03E-01	8.04E-02	8.02E-02
Glass	214	9	2.33E-03	4.15E-03	7.30E-03	1.19E-02	1.48E-02	4.37E-03	7.12E-03	1.25E-02	1.66E-02	1.74E-02
Housing	506	13	2.41E-03	4.61E-03	7.30E-03	6.27E-02	7.60E-02	4.76E-03	8.44E-03	1.28E-02	7.74E-02	8.03E-02
Space Sh.	23	4	4.49E-03	8.81E-03	1.41E-02	5.11E-04	4.83E-04	1.17E-02	2.10E-02	3.20E-02	1.13E-03	1.18E-03
Synth 1	30000	1000	4.26E+01	6.08E+01	7.13E+01	7.80E+01	8.99E+01	8.32E+01	9.40E+01	8.63E+01	7.68E+01	8.72E+01
Synth 2	1000	30000	2.16E+01	3.46E+01	4.91E+01	2.81E+01	3.91E+01	4.88E+01	6.07E+01	7.34E+01	4.86E+01	4.52E+01
WPBC	683	10	1.61E-02	3.01E-02	4.19E-02	9.47E-02	1.25E-01	8.78E-02	1.44E-01	1.80E-01	1.12E-01	1.18E-01

Table 8: Comparison of running times (in seconds) for optimization procedure and randomized procedure, averaged over 1000 runs. We see that for the 50/50 split the optimization procedure takes on average 2.0601 times longer, for the 60/40 split on average 2.8044 times longer, for the 70/30 split, on average 2.6538 times longer, for the 80/20 split, on average 1.0276 times longer, and for the 90/10 split, on average 1.0192 times longer.

8. Discussion of Results

Overall, the optimization approach has a clear advantage over the randomization approach with a single validation set. This is true across the board: it yields lower prediction error, lower standard deviation of prediction error, lower coefficient standard deviation, and lower hyperparameter standard deviation. This advantage is highest when the 50/50 training/validation scheme is employed, and decreases as a higher percentage of the data is used for training (this pattern holds monotonically for all results except hyperparameter standard deviation). This makes sense, because when 50% of the data is used for training, the space of all possible training/validation splits is largest (the function $\binom{n}{k}$ is maximized when $k = n/2$) and hence the optimization approach has the greatest maneuverability in this regime. In contrast, as a greater proportion of the data is used for training, the number of options that the optimization procedure has decreases (in the limit when 100% of the data is used for training, there is only 1 possible training/validation split), and thus we would expect the corresponding edge to also decrease. When compared to cross-validation, the optimization approach still has an advantage, but it is markedly smaller. This also makes sense, as the cross validation approach ends up sifting through all of the non-test data, and so in some sense has access to more information than the regime where only one training set is taken at random. Despite this, there still are regimes where the advantage that the optimization approach brings over cross-validation is substantial, for example in hyperparameter standard deviation, where we saw percentage improvements between 20% and 30%. Finally, the optimization approach also results in superior support recovery, and most uniquely, when used in concert with Optimal Decision Trees, provides insight into which subpopulations are hardest to learn.

9. Conclusions

In this paper, we investigated the commonly held belief that the best way to train regression models is via random assignment of the data to training and validation sets. In particular, we proposed an optimization framework for optimally selecting such training and validation sets for both regularized and unregularized regression problems. We applied this algorithm to 10 data sets collected from the UCI Machine Learning Repository, as well as two large scale synthetic data sets, and reported very encouraging results. For both regularized and unregularized regression, we observed that the optimization approach led to predictions that had both lower error (MSE), lower standard deviation, and that the resulting constructed models were more stable (in the sense that their coefficients had lower standard deviation) as compared to the predictions yielded by and the models constructed by the randomization approach in all twelve data sets. The edge of optimization versus randomization continued to hold for recovering the true support of a data set in terms of both higher accuracy and lower false discovery rate, and that this advantage was present for when the true number of coefficients was known, as well as when this was estimated via cross validation. The increased stability is important as it enhances interpretability, since the constructed model varies less, both in terms of its support and in terms of the numerical value of its coefficients. Moreover, we showed that training via optimization was equivalent to constructing models that were robust to all subpopulations in the data, and thus in particular robust to the hardest subpopulation, which is identified as a by product of this process. Once identified, this hardest subset can then be further studied to understand what properties exactly made it "hard," and indeed we found such analysis to provide interesting domain specific insights. Finally, all of this is accompanied by an efficient algorithm that scales this optimization approach to training to essentially any desired size.

Acknowledgements

We would like to thank the two reviewers of the paper, whose insightful suggestions improved the paper substantially.

Appendix: Error Bars for Tables 1 - 7

Error Bars for Table 1												
Data Sets			**Randomization**					**Optimization**				
	n	p	50/50	60/40	70/30	80/20	90/10	50/50	60/40	70/30	80/20	90/10
Abalone	4177	8	2.94E-02	1.88E-02	1.64E-02	4.70E-03	3.14E-03	2.56E-02	1.78E-02	1.89E-02	4.28E-03	2.91E-03
Auto MPG	392	7	1.60E-01	1.03E-01	7.22E-02	3.00E-02	4.18E-03	1.31E-01	1.08E-01	7.18E-02	2.74E-02	3.63E-03
Comp Hard	209	6	1.23E+02	9.19E+01	5.94E+01	6.28E+00	4.97E+00	1.06E+02	9.82E+01	8.34E+01	5.60E+00	3.71E+00
Concrete	103	7	4.60E+00	3.34E+00	1.68E+00	5.72E-01	5.27E-01	3.70E+00	1.94E+00	1.63E+00	6.47E-01	4.80E-01
Ecoli	336	7	9.46E-02	3.57E-02	1.24E-02	1.12E-02	7.15E-03	7.15E-02	3.87E-02	1.12E-02	1.17E-02	7.15E-03
Forest Fi.	517	12	8.61E+00	8.10E+00	4.46E+00	1.74E+00	5.00E-01	8.54E+00	9.22E+00	4.24E+00	1.56E+00	6.06E-01
Glass	214	9	5.57E-02	2.17E-02	1.66E-02	4.16E-03	5.33E-04	4.90E-02	1.83E-02	1.63E-02	4.44E-03	5.28E-04
Housing	506	13	4.20E-01	3.20E-01	2.39E-01	8.51E-02	3.17E-02	3.70E-01	3.55E-01	2.71E-01	8.95E-02	2.72E-02
Space Sh.	23	4	7.14E-02	5.44E-02	3.02E-02	1.55E-02	2.07E-03	6.67E-02	4.85E-02	2.50E-02	1.35E-02	2.10E-03
Synth 1	30000	1000	3.60E-03	2.09E-03	2.71E-05	8.29E-06	5.03E-06	2.81E-03	1.89E-03	2.40E-05	7.56E-06	4.58E-06
Synth 2	1000	30000	3.60E+01	1.45E+01	2.76E+00	1.12E+00	2.57E+00	1.33E+01	7.39E+00	1.56E+00	6.91E-01	2.00E+00
WPBC	683	10	1.70E-02	2.04E-02	1.14E-02	2.98E-03	6.47E-04	1.49E-02	1.62E-02	9.48E-03	3.14E-03	6.46E-04

Table 9: Error Bars for Table 1.

Error Bars for Table 2														
Data Sets			**Randomization**							**Optimization**				
	n	p	50/50	60/40	70/30	80/20	90/10	5-CV	10-CV	50/50	60/40	70/30	80/20	90/10
Abalone	4177	8	3.36E-02	2.31E-02	1.22E-02	9.08E-03	3.12E-03	1.18E-02	2.44E-03	2.48E-02	2.26E-02	1.18E-02	1.01E-02	3.71E-03
Auto MPG	392	7	1.10E-01	1.03E-01	6.52E-02	7.79E-02	1.78E-02	8.39E-02	1.95E-02	9.68E-02	9.80E-02	6.37E-02	9.76E-02	2.01E-02
Comp Hard	209	6	1.02E+02	9.92E+01	4.33E+01	5.22E+01	7.92E+00	7.25E+01	5.12E+00	8.52E+01	1.12E+02	5.17E+01	5.72E+01	7.86E+00
Concrete	103	7	4.73E+00	2.50E+00	1.63E+00	1.66E+00	1.27E-01	1.25E+00	1.17E-01	2.84E+00	1.94E+00	1.23E+00	1.90E+00	1.10E-01
Ecoli	336	7	1.32E-02	9.70E-03	8.59E-03	6.00E-03	3.42E-04	6.97E-03	2.77E-04	1.22E-02	8.72E-03	8.61E-03	5.89E-03	3.61E-04
Forest Fi.	517	12	7.65E+00	6.92E+00	5.05E+00	3.01E+00	5.90E-01	6.98E+00	5.76E-01	6.97E+00	6.71E+00	4.63E+00	2.84E+00	5.76E-01
Glass	214	9	8.29E-03	2.95E-03	2.91E-03	7.93E-04	6.51E-03	7.56E-04	5.20E-03	8.88E-03	2.73E-03	2.59E-03	7.25E-04	5.52E-03
Housing	506	13	4.25E-01	3.00E-01	1.29E-01	5.82E-02	1.35E-02	7.14E-02	1.54E-02	4.74E-01	2.71E-01	1.38E-01	5.65E-02	1.16E-02
Space Sh.	23	4	6.64E-02	1.59E-02	1.37E-02	3.40E-03	2.94E-03	3.61E-03	1.97E-03	6.69E-02	1.47E-02	9.68E-03	3.36E-03	3.02E-03
Synth 1	30000	1000	1.89E-03	2.29E-03	2.54E-03	7.16E-06	4.34E-06	4.91E-06	2.56E-06	1.19E-03	1.66E-03	2.26E-03	6.82E-06	4.22E-06
Synth 2	1000	30000	5.09E+01	1.60E+01	5.92E-01	1.38E+00	6.56E-01	9.48E-01	3.86E-01	3.04E+01	1.38E+01	3.87E-01	1.66E+00	6.74E-01
WPBC	683	10	1.88E-02	1.60E-02	7.39E-03	3.07E-03	4.53E-04	1.19E-03	7.77E-04	1.41E-02	1.47E-02	1.10E-02	3.27E-03	4.83E-04

Table 10: Error Bars for Table 2.

Error Bars for Table 3												
Data Sets			**Randomization**					**Optimization**				
	n	p	50/50	60/40	70/30	80/20	90/10	50/50	60/40	70/30	80/20	90/10
Abalone	4177	8	1.877E-02	1.299E-02	1.139E-02	1.059E-02	1.994E-03	1.517E-02	1.228E-02	1.106E-02	1.128E-02	1.994E-03
Auto MPG	392	7	9.412E-02	3.609E-02	3.496E-02	1.001E-02	1.294E-03	9.024E-02	3.267E-02	3.445E-02	9.842E-03	1.285E-03
Comp Hard	209	6	3.611E+01	1.339E+01	8.560E+00	2.703E+01	1.270E+01	3.217E+01	1.258E+01	8.614E+00	2.754E+01	1.265E+01
Concrete	103	7	2.995E+00	2.817E+00	2.406E+00	5.883E-01	3.383E-01	2.090E+00	1.986E+00	2.228E+00	4.864E-01	3.629E-01
Ecoli	336	7	1.504E+00	5.906E-01	6.210E-03	8.884E-03	3.925E-03	1.269E+00	5.798E-01	5.429E-03	9.090E-03	3.864E-03
Forest Fi.	517	12	8.667E+00	6.387E+00	1.692E+00	1.188E+00	5.199E-01	9.276E+00	6.136E+00	1.930E+00	1.224E+00	5.517E-01
Glass	214	9	7.958E-02	6.796E-02	2.012E-02	1.351E-02	2.617E-03	6.570E-02	5.551E-02	2.050E-02	1.423E-02	2.442E-03
Housing	506	13	8.744E-02	7.577E-02	7.057E-02	1.898E-02	1.253E-02	7.058E-02	7.002E-02	6.859E-02	1.804E-02	1.306E-02
Space Sh.	23	4	5.220E-02	4.633E-02	3.264E-02	8.592E-03	1.274E-03	4.541E-02	4.102E-02	3.035E-02	9.143E-03	1.176E-03
Synth 1	30000	1000	6.571E-04	5.615E-04	4.461E-04	4.241E-04	1.154E-03	5.082E-04	4.790E-04	3.774E-04	3.958E-04	1.121E-03
Synth 2	1000	30000	2.948E+01	1.982E+01	3.997E+01	8.515E+01	3.250E+01	1.080E+01	9.551E+00	2.149E+01	5.373E+01	2.694E+01
WPBC	683	10	2.440E-02	1.840E-02	1.620E-02	7.475E-03	4.748E-04	2.035E-02	1.506E-02	1.432E-02	7.931E-03	4.344E-04

Table 11: Error Bars for Table 3.

Error Bars for Table 4														
Data Sets			**Randomization**							**Optimization**				
	n	p	50/50	60/40	70/30	80/20	90/10	5-CV	10-CV	50/50	60/40	70/30	80/20	90/10
Abalone	4177	8	1.223E-02	1.571E-02	8.945E-03	1.333E-02	5.087E-04	1.538E-02	3.973E-04	1.025E-02	1.416E-02	8.411E-03	1.374E-02	5.288E-04
Auto MPG	392	7	7.894E-02	3.697E-02	3.538E-02	2.704E-02	6.172E-03	2.188E-02	6.767E-03	6.930E-02	3.354E-02	3.594E-02	2.496E-02	5.781E-03
Comp Hard	209	6	1.614E+01	5.834E+00	4.159E+01	1.896E+01	1.324E+01	2.202E+01	8.562E+00	1.355E+01	5.732E+00	4.043E+01	1.726E+01	1.142E+01
Concrete	103	7	4.413E+00	2.996E+00	2.244E+00	1.473E+00	4.487E-01	1.190E+00	4.143E-01	2.362E+00	2.609E+00	1.995E+00	1.464E+00	4.223E-01
Ecoli	336	7	9.409E-03	5.899E-03	6.401E-03	5.437E-03	1.297E-03	5.577E-03	1.049E-03	8.820E-03	5.662E-03	5.830E-03	5.140E-03	1.302E-03
Forest Fi.	517	12	8.001E+00	5.318E+00	2.374E+00	1.523E+00	1.111E-01	2.882E+00	1.084E-01	8.101E+00	5.461E+00	2.534E+00	1.586E+00	1.094E-01
Glass	214	9	2.178E-02	1.901E-02	3.385E-03	3.324E-03	1.842E-03	2.883E-03	1.473E-03	2.123E-02	1.686E-02	3.285E-03	3.226E-03	1.992E-03
Housing	506	13	7.088E-02	4.981E-02	6.734E-02	4.236E-02	1.112E-02	4.575E-02	1.268E-02	6.797E-02	5.083E-02	6.898E-02	4.057E-02	1.051E-02
Space Sh.	23	4	5.243E-02	4.050E-03	8.039E-03	7.065E-03	3.733E-03	5.180E-03	2.501E-03	4.889E-02	3.782E-03	6.961E-03	7.244E-03	3.808E-03
Synth 1	30000	1000	2.804E-04	3.257E-04	4.081E-04	8.673E-04	2.131E-04	5.102E-04	1.253E-04	1.727E-04	2.385E-04	3.738E-04	7.751E-04	2.058E-04
Synth 2	1000	30000	3.154E+01	1.837E+01	1.972E+01	3.411E+01	1.920E+01	2.006E+01	1.129E+01	1.845E+01	1.594E+01	1.331E+01	3.840E+01	1.964E+01
WPBC	683	10	4.009E-02	1.229E-02	2.149E-02	4.335E-03	4.093E-04	1.521E-03	7.017E-04	3.244E-02	1.333E-02	2.901E-02	4.333E-03	3.879E-04

Table 12: Error Bars for Table 4.

Error Bars for Table 5												
Data Sets			**Randomization**					**Optimization**				
	n	p	50/50	60/40	70/30	80/20	90/10	50/50	60/40	70/30	80/20	90/10
Abalone	4177	8	3.405E-03	2.772E-03	2.196E-03	2.969E-03	5.960E-05	1.880E-03	1.531E-03	1.336E-03	2.256E-03	5.452E-05
Auto MPG	392	7	2.804E-03	7.979E-05	2.958E-03	1.502E-05	9.659E-06	1.594E-03	4.741E-05	2.112E-03	1.038E-05	7.106E-06
Comp Hard	209	6	3.278E-03	3.893E-03	9.269E-05	2.124E-05	1.191E-05	1.454E-03	1.993E-03	5.059E-05	1.319E-05	9.690E-06
Concrete	103	7	2.413E-03	2.392E-03	5.735E-04	1.243E-04	3.735E-03	1.152E-03	1.453E-03	4.055E-04	9.842E-05	3.151E-03
Ecoli	336	7	3.117E-03	2.064E-03	3.889E-03	1.882E-03	3.967E-03	1.660E-03	1.103E-03	2.124E-03	1.177E-03	3.207E-03
Forest Fi.	517	12	7.772E-03	3.921E-03	3.396E-03	1.637E-03	5.593E-06	3.752E-03	2.630E-03	1.822E-03	1.053E-03	4.012E-06
Glass	214	9	4.568E-03	3.182E-02	1.978E-02	2.469E-02	4.298E-03	2.992E-03	2.337E-02	1.221E-02	1.709E-02	3.381E-03
Housing	506	13	8.755E-03	6.478E-03	2.775E-03	5.321E-03	1.508E-05	4.470E-03	3.586E-03	1.577E-03	4.065E-03	1.293E-05
Space Sh.	23	4	3.757E-03	5.979E-04	4.427E-04	8.585E-05	2.166E-03	1.688E-03	2.709E-04	1.841E-04	5.726E-05	1.445E-03
Synth 1	30000	1000	4.264E-03	3.458E-03	2.840E-03	1.320E-04	2.981E-03	2.040E-03	1.844E-03	1.717E-03	8.774E-05	2.284E-03
Synth 2	1000	30000	5.672E-02	4.853E-02	1.356E-01	9.440E-02	7.142E-02	4.890E-02	4.247E-02	1.233E-01	8.746E-02	6.830E-02
WPBC	683	10	9.591E-05	7.719E-05	6.298E-05	1.478E-05	9.938E-06	5.297E-05	4.263E-05	3.699E-05	1.030E-05	7.742E-06

Table 13: Error Bars for Table 5.

Error Bars for Table 6														
Data Sets			**Randomization**							**Optimization**				
	n	p	50/50	60/40	70/30	80/20	90/10	5-CV	10-CV	50/50	60/40	70/30	80/20	90/10
Abalone	4177	8	9.436E-04	2.687E-03	5.715E-04	2.249E-03	5.166E-03	1.998E-03	4.922E-03	5.112E-04	1.608E-03	3.214E-04	1.713E-03	5.126E-03
Auto MPG	392	7	2.808E-04	7.646E-05	3.377E-04	8.349E-04	9.724E-06	7.692E-04	9.607E-06	1.455E-04	4.202E-05	2.422E-04	5.925E-04	7.631E-06
Comp Hard	209	6	1.601E-04	6.064E-04	8.952E-04	2.649E-03	2.901E-03	2.630E-03	2.482E-03	7.039E-05	3.385E-04	4.871E-04	1.699E-03	2.019E-03
Concrete	103	7	2.887E-04	6.794E-04	5.225E-04	1.577E-03	5.755E-04	1.779E-03	5.576E-04	1.304E-04	3.932E-04	3.946E-04	1.193E-03	4.797E-04
Ecoli	336	7	5.134E-04	5.146E-04	1.456E-03	1.666E-03	4.794E-03	1.687E-03	4.493E-03	3.458E-04	3.283E-04	9.951E-04	1.232E-03	3.943E-03
Forest Fi.	517	12	5.040E-04	1.134E-03	2.205E-04	7.813E-06	6.639E-06	8.756E-06	5.971E-06	2.882E-04	6.814E-04	1.221E-04	5.642E-06	5.477E-06
Glass	214	9	1.031E-03	6.530E-03	1.056E-03	4.803E-04	2.188E-03	3.621E-04	2.084E-03	2.507E-03	4.589E-03	3.014E-03	9.305E-04	2.704E-03
Housing	506	13	3.623E-04	1.217E-04	5.732E-04	1.044E-02	4.395E-03	1.036E-02	4.400E-03	2.405E-04	6.968E-05	3.974E-04	8.391E-03	4.235E-03
Space Sh.	23	4	2.303E-04	4.872E-04	4.479E-04	1.538E-03	6.091E-04	1.014E-03	5.152E-04	8.867E-05	2.416E-04	3.062E-04	1.283E-03	5.834E-04
Synth 1	30000	1000	2.545E-03	4.090E-04	7.989E-04	6.428E-04	6.859E-04	3.781E-04	4.035E-04	1.241E-03	2.231E-04	4.873E-04	4.348E-04	5.374E-04
Synth 2	1000	30000	1.295E-01	2.393E-01	1.879E-02	1.988E-01	1.915E-01	1.169E-01	1.127E-01	8.592E-02	1.662E-01	1.262E-02	1.429E-01	1.480E-01
WPBC	683	10	9.610E-05	7.616E-05	6.188E-05	5.445E-04	9.447E-06	5.215E-04	8.586E-06	5.289E-05	4.094E-05	4.415E-05	4.016E-04	7.501E-06

Table 14: Error Bars for Table 6.

Error Bars for Table 7														
Data Sets			**Randomization**							**Optimization**				
	n	p	50/50	60/40	70/30	80/20	90/10	5-CV	10-CV	50/50	60/40	70/30	80/20	90/10
Abalone	4177	8	1.894E-02	3.336E-02	7.270E-02	1.140E-01	7.930E-02	6.099E-02	9.536E-02	1.557E-02	1.064E-02	2.926E-02	3.455E-03	6.542E-03
Auto MPG	392	7	4.406E-03	1.363E-03	6.279E-03	9.011E-04	5.830E-03	8.848E-04	5.585E-03	4.415E-03	1.555E-03	6.423E-03	8.863E-04	6.237E-03
Comp Hard	209	6	2.999E-02	3.847E-02	4.113E-02	4.311E-02	1.825E-02	2.472E-02	1.873E-02	3.065E-02	4.529E-02	4.830E-02	5.123E-02	2.164E-02
Concrete	103	7	1.150E-02	2.083E-02	1.052E-02	2.736E-03	1.407E-02	2.710E-03	1.486E-02	1.759E-02	2.635E-02	1.130E-02	2.919E-03	1.833E-02
Ecoli	336	7	2.510E-02	4.858E-02	5.507E-02	4.338E-02	1.229E-02	3.750E-02	1.075E-02	1.661E-02	2.454E-02	3.436E-02	3.029E-02	9.534E-03
Forest Fi.	517	12	2.824E-02	6.404E-02	2.514E-02	2.432E-02	4.010E-03	2.248E-02	4.140E-03	2.601E-02	4.649E-02	1.727E-02	2.322E-02	3.963E-03
Glass	214	9	8.209E-03	5.698E-03	1.182E-02	1.096E-02	1.247E-02	1.179E-02	1.244E-02	8.403E-03	5.189E-03	1.609E-02	8.632E-03	9.590E-03
Housing	506	13	2.180E-02	5.238E-03	1.916E-03	6.289E-03	2.036E-02	5.698E-03	1.863E-02	2.000E-02	4.755E-03	2.049E-03	6.003E-03	1.743E-02
Space Sh.	23	4	1.212E-02	1.623E-03	1.279E-02	1.175E-03	8.669E-03	1.260E-03	7.272E-03	1.069E-02	1.765E-03	1.299E-02	1.141E-03	8.487E-03
Synth 1	30000	1000	4.265E-02	4.330E-02	6.097E-02	9.128E-02	5.061E-02	5.369E-02	2.977E-02	1.842E-01	9.218E-02	3.000E-01	3.507E-01	5.528E-02
Synth 2	1000	30000	4.858E-02	8.075E-02	4.639E-02	2.049E-02	6.201E-02	1.205E-02	3.648E-02	8.040E-02	2.334E-01	1.941E+01	3.661E-02	8.870E-02
WPBC	683	10	3.766E-03	3.483E-03	3.222E-03	3.764E-03	6.206E-04	4.055E-03	5.842E-04	3.651E-03	3.239E-03	3.269E-03	3.979E-03	6.385E-04

Table 15: Error Bars for Table 7.

References

Dimitris Bertsimas and Jack Dunn. Optimal classification trees. *Machine Learning*, 106: 1039–1082, 04 2017.

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

Dimitris Bertsimas, Mac Johnson, and Nathan Kallus. The power of optimization over randomization in designing experiments involving small samples. *Operations Research*, 63(4):868–876, 2015.

Dimitris Bertsimas, Allison K. O'Hair, and William R. Pulleyblank. *The Analytics Edge*, volume 1. 2016a. 934829119.

Dimitris Bertsimas, Nathan Kallus, Alexander M. Weinstein, and Ying Zhuo. Personalized diabetes management using electronic medical records. *Diabetes Care*, 40:210–217, 12 2016b.

Dimitris Bertsimas, Angela King, and Rahul Mazumder. Best subset selection via a modern optimization lens. *The Annals of Statistics*, 44:813–852, 07 2016c.

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

John Duchi and Hongseok Namkoong. Learning models with uniform performance via distributionally robust optimization. *arXiv:1810.08750v2 [stat.ML]*, 2019.

Kenneth Garbade. Two methods for examining the stability of regression coefficients. *Journal of the American Statistical Association*, 72(357):54–63, 1977.

Arthur Hoerl and Robert Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12:55–67, 1970.

Frederick Mosteller and John Wilder Tukey. Data analysis, including statistics. In G. Lindzey and E. Aronson, editors, *Handbook of Social Psychology, Vol. 2*. Addison-Wesley, 1968.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1994.

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, 67:301–320, 2005.