**Noname manuscript No.**
(will be inserted by the editor)

# Sparse Regression over Clusters: SparClur

**Dimitris Bertsimas · Jack Dunn · Lea Kapelevich · Rebecca Zhang**

December 28, 2020

**Abstract** Prediction tasks in personalized medicine require models that combine accuracy and interpretability. We propose an integer optimization approach for building sparse regression models with enforced coordination, using data partitioned among leaves in a prediction tree. We show that the method recovers the true underlying relationship between observations and target variables in large-scale synthetic data in seconds. We apply our method to several real-world medical prediction problems and observe that the additional structure imposed provides a substantial gain in interpretability, at a low cost to accuracy.

**Keywords** prediction trees · regression · integer optimization · personalized medicine

## 1 Introduction

Motivated by by applications of medicine, we aim to develop machine learning models that combine state of the art accuracy and interpretability. We focus on three medi-

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. 1122374.

Dimitris Bertsimas
Sloan School of Management, MIT
E-mail: dbertsim@mit.edu

Jack Dunn
Operations Research Center, MIT
E-mail: jackdunn@mit.edu

Lea Kapelevich
Operations Research Center, MIT
E-mail: lkap@mit.edu

Rebecca Zhang
Operations Research Center, MIT
E-mail: rzhang15@mit.edu

cal prediction tasks that require capturing highly nonlinear relationships between features and continuous target variables. Currently, the most popular methods for such problems include decision trees (classification and regression trees), random forests, and boosted trees.

Deep learning and ensemble models (including random forests and boosted trees) achieve state of the art accuracy, but are not interpretable. This limits their applicability in areas where understanding the rationale of a model's prediction is important. On the other hand, sparse regression models and decision trees aspire to be interpretable, but have weaker out of sample accuracy. In this paper, we combine ideas from new developments in sparse regression and classification and regression trees to propose a method that is interpretable, and also provides state of the art accuracy.

To motivate the method, assume we have data $(\mathbf{x}_i, y_i), i \in [\![n]\!]$ with $\mathbf{x}_i \in \mathbb{R}^p$, $y_i \in \mathbb{R}$, and $[\![n]\!]$ are the integers $1,\ldots,n$. Let $\mathbf{x}_i$ represent electronic medical records of patient $i$ and $y_i$ represent a medical outcome, for example, glucose levels of patient $i$. Applying *Optimal Regression Trees* (ORTs) from [10] gives rise to trees such as that depicted in Figure 1.1. In each of the leaves $L_1, L_2$, the outcome is predicted as a

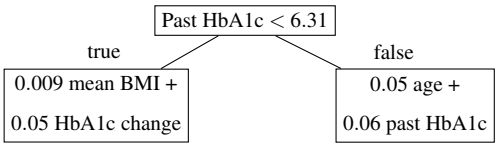



Fig. 1.1: Example of an Optimal Regression Tree of depth one for predicting blood glucose levels.

regression $\hat{y}_i = \mathbf{w}_j^\top \mathbf{x}_i$ for all $j \in [\![J]\!]$, $J = 2$. This allows different features in each leaf to affect the prediction.

Suppose we impose the additional constraint that the support of each vector $\mathbf{w}_j$ is the same for all leaves and in addition the cardinality of this support is limited. That is, $|\text{supp}(\mathbf{w}_j)| \leq q$ for some positive integer $q$, and $\text{supp}(\mathbf{w}_j) = \text{supp}(\mathbf{w}_k)$, for $j,k \in [\![J]\!]$. With this criterion, the regression in each leaf is sparse when $q$ is small, and coordinated among leaves to involve the same variables. This increases the interpretability of the model significantly. Specifically, in the uncoordinated case, it is possible that past HbA1c affects glucose level in some leaves while not in others, which is medically implausible.

This setting leads us to consider the following general problem. Given input data $\mathbf{X} = (\mathbf{x}_1,\ldots,\mathbf{x}_n)$, response data data $\mathbf{y} = (y_1,\ldots,y_n)$, and a partition within *clusters* $L_j$ such that $\exists j \in [\![J]\!] : \ (\mathbf{x}_i, y_i) \in L_j, \forall i \in [\![n]\!]$, we wish to solve:

$$\min_{\mathbf{w}} \quad \frac{1}{2\gamma}\sum_{j=1}^{J} ||\mathbf{w}_j||^2 + \sum_{j=1}^{J}\sum_{i\in L_j}(y_i - \mathbf{w}_j^\top \mathbf{x}_i)^2 \tag{1.1a}$$

$$\text{s.t.} \quad ||\mathbf{w}_j||_0 \leq q \quad \forall j \in [\![J]\!] \tag{1.1b}$$

$$\text{supp}(\mathbf{w}_1) = \ldots = \text{supp}(\mathbf{w}_J). \tag{1.1c}$$

The term $\frac{1}{2\gamma}\sum_{j=1}^{J}||\mathbf{w}_j||^2$ is a Tikhonov [17] regularization term, that makes the overall model more robust [2] and $\gamma$ is a parameter to be tuned. Problem (1.1) reduces to the sparse regression problem studied by Bertsimas and Van Parys [4] when $J = 1$.

### 1.1 Existing methods

Recently introduced in [10, 3], ORTs are a predictive tool similar to CART [9]. Given a loss function, ORTs are constructed so that this function is minimized with respect to local neighborhoods in the tree. ORTs have a higher degree of interpretability than black-box methods such as random forest and boosted trees, while attaining comparable performance in terms of predictive accuracy. In addition, ORTs have notably higher predictive accuracy than CART [10].

Linear regression models at the nodes of ORTs are currently computed heuristically [3]. These regression models are either point predictions, or linear models built using Lasso [16]. This has several shortcomings. As shown by Bertsimas et al. [5], while Lasso generally performs well at the task of discovering relevant features, it also selects a significant number of features that are not part of the true support. This hinders the interpretability of trees because often a large number of features are selected in the support of the linear models, and it is unclear which features are truly relevant. Second, the use of heuristics for sparse linear regression does not leverage the potential of optimal regression methods.

A powerful approach for building sparse regression models with an explicit constraint on the zero norm of the weights vector was proposed in [4]. The authors showed that the method outperforms Lasso in terms of accuracy, and particularly in false recovery rate on a test set of problems. The paper also introduced the phenomenon of *phase transitions* for the exact sparse regression problem. That is, at a critical number of observations, the performance of the algorithm peaks in terms of accuracy and false detection, and *improves* in terms of computational speed. This is a notable empirical result, as it puts in question the commonly held belief that exact algorithms are not comparable in practice with heuristics for solving large scale regression problems.

### 1.2 Contributions

There is currently no approach that leverages the interpretability and predictive power of integer optimization methods for regression together with tree based methods. To that end, we propose an integer optimization approach for building linear models that can be naturally applied to prediction trees. The technique we propose, called *SparClur* (*spar*se *clu*ster *r*egression), computes a number of regression models simultaneously at different leaves in a tree, and enforces coordination between nodes by requiring the support within all regression models to be the same. We follow the approach initiated by [4], and include the additional requirement of coordinated sparsity in order to improve performance and interpretability in real-world prediction problems. We demonstrate the validity of SparClur using real medical datasets. Specifically, we

show that imposing the coordination constraint (1.1c) is computationally inexpensive while the formulation results in similar accuracy to the uncoordinated problem. Using synthetic data, we demonstrate that SparClur recovers the true support while ignoring irrelevant features for large problems in seconds.

We present a formulation and describe an algorithm for solving Problem (1.1) in the next section. In Section 3, we test the performance of solutions generated by SparClur in synthetic datasets. In Section 4, we apply the same approach to three high dimensional medical problems. Since we cannot make the medical datasets publically available, we repeat our experiments on a publicly available dataset related to real estate pricing. All our code is available at: `https://github.com/SparClur/SparClur.jl` .

## 2 Problem formulation

Let $\mathbf{s} \in \{0,1\}^p$ denote the common support of all sparse weight vectors. Let $\mathbf{S}$ be a diagonal matrix with elements of $\mathbf{s}$. The problem that SparClur seeks to solve can be written as:

$$\min_{\mathbf{w},\mathbf{s}} \quad \frac{1}{2\gamma}\sum_{j=1}^{J}||\mathbf{w}_j||_2^2+\frac{1}{2}\sum_{j=1}^{J}\sum_{i\in L_j}\left(y_i-\mathbf{x}_i^\top \mathbf{S}\mathbf{w}_j\right)^2 \tag{2.1a}$$

$$\text{s.t.} \quad \mathbf{1}^\top\mathbf{s}\leq q \tag{2.1b}$$

$$\mathbf{w}\in\mathbb{R}^p, \mathbf{s}\in\{0,1\}^p \tag{2.1c}$$

where $\gamma > 0$.

**Lemma 1** *Let $(\mathbf{X}_j, \mathbf{y}_j)$ denote the slices of $(\mathbf{X},\mathbf{y})$ containing rows that belong to cluster $j$. Problem* (2.1) *is equivalent to solving:*

$$\min_{\mathbf{s}\in D}\frac{1}{2}\sum_{j=1}^{J}\left[\frac{1}{2}\mathbf{y}_j^\top\left(\mathbb{I}_n+\gamma\sum_{i=1}^{p}s^i\mathbf{K}_j^i\right)^{-1}\mathbf{y}_j\right] \tag{2.2}$$

*where $D=\{\mathbf{s}\in\{0,1\}^p : \mathbf{1}^\top\mathbf{s}\leq q\}$, $\mathbf{K}_j^i=\mathbf{X}_j^i\mathbf{X}_j^{i\top}$, and $\mathbf{X}_j^{i\top}$ is a column of $\mathbf{X}_j$ that pertains to feature $i$.*

*Proof* The proof follows by the same argument as in [4, Lemma 1]. For a fixed support vector $\mathbf{s}$, the objective is equal to:

$$c(\mathbf{s})=\frac{1}{2}\sum_{j=1}^{J}\left[\mathbf{y}_j^\top\left(\mathbb{I}_n+\gamma\mathbf{X}_j\mathbf{S}\mathbf{X}_j^\top\right)^{-1}\mathbf{y}_j\right],$$

where the optimal weights for cluster $j$ are used:

$$\mathbf{w}_j^*=\left(\frac{1}{\gamma}\mathbb{I}_p+(\mathbf{X}_j\mathbf{S})^\top\mathbf{X}_j\mathbf{S}\right)^{-1}(\mathbf{X}_j\mathbf{S})^\top\mathbf{y}_j. \quad \square \tag{2.3}$$

The convexity of the objective function in $\mathbf{s}$ enables us to apply an outer approximation algorithm [12] to solve (2.2). It is convenient to consider the dual of Problem (2.1) with $\mathbf{s}$ fixed in order to derive the form of the cuts. The dual has the following form [4]:

$$\max_{\alpha_j, j\in[\![J]\!]} \quad \frac{-\gamma}{2}\sum_{j=1}^{J}\alpha_j^\top \mathbf{K}_j(\mathbf{s})\alpha_j - \frac{1}{2}\alpha_j^\top\alpha_j + \mathbf{y}_j^\top\alpha_j \tag{2.4a}$$

$$\text{s.t.} \quad \alpha_j \in \mathbb{R}^{n_j} \quad \forall j \in [\![J]\!], \tag{2.4b}$$

where $\mathbf{K}_j(\mathbf{s}) = \mathbf{X}_j\mathbf{S}\mathbf{X}_j^\top$ and $n_j$ is the number of observations in cluster $j$. The variables $\alpha_j$ can be interpreted as the Lagrangian duals corresponding to constraints of the form $\mathbf{y}_j = \mathbf{X}_j\mathbf{S}\mathbf{w}_j$. As (2.4) is an unconstrained quadratic problem, we can derive a closed form solution for the optimal dual variables $\alpha_j^*$:

$$\alpha_j^* = (\mathbb{I}_n + \gamma\mathbf{K}_j)^{-1}\mathbf{y}_j \quad \forall j \in [\![J]\!]. \tag{2.5}$$

Now, at a given candidate solution $\mathbf{s}$, we have that our kernel matrices $\mathbf{K}_j(\mathbf{s})$ are differentiable and furthermore

$$\frac{d\mathbf{K}_j(\mathbf{s})}{ds_i} = \mathbf{X}_j^i\mathbf{X}_j^{i\top} \quad \forall i \in [\![p]\!], j \in [\![J]\!],$$

so we can always attain a subgradient [4, Lemma 2]:

$$\nabla c(\mathbf{s}) = -\frac{\gamma}{2}\sum_{j=1}^{J}\alpha_j^{*\top}\frac{d\mathbf{K}_j(\mathbf{s})}{d\mathbf{s}}\alpha_j^*. \tag{2.6}$$

In practice, to avoid computing the inverse of the $n \times n$ matrix in (2.5), we compute the *capacitance matrix* as in Bertsimas and Van Parys [4, Equation 20]. This gives rise to Algorithm 1.

Besides the exact cutting plane algorithm, we can use an algorithm for the convex relaxation of (2.1) such as the subgradient descent method suggested in [7]. The convex relaxation is useful for providing warm starts to a mixed integer solver, but can also provide high quality solutions on its own. The convex relaxation takes the form:

$$\min_{\mathbf{s}\in\text{conv}(D)} \max_{\alpha_j\in\mathbb{R}^{n_j}} \quad f(\alpha_1,\ldots,\alpha_J,\mathbf{s}), \tag{2.7}$$

with $f(\alpha_1,\ldots,\alpha_J,\mathbf{s}) = \frac{-\gamma}{2}\sum_{j=1}^{J}\alpha_j^\top\mathbf{K}_j(\mathbf{s})\alpha_j - \frac{1}{2}\alpha_j^\top\alpha_j + \mathbf{y}_j^\top\alpha_j$. We can exchange the order of the minimization and maximization operators, so (2.7) is equivalent to:

$$\max_{\alpha_j\in\mathbb{R}^{n_j}} \sum_{j=1}^{J}\mathbf{y}_k^\top\alpha_j - \frac{1}{2}\alpha_j^\top\alpha_j - \max_{\mathbf{s}\in\text{conv}(D)} g(\alpha_1,\ldots,\alpha_J) \tag{2.8}$$

with $g(\alpha_1,\ldots,\alpha_J) = \frac{\gamma}{2}\sum_{i=1}^{p}s_i\alpha_j^\top\mathbf{X}_j^i\mathbf{X}_j^{i\top}\alpha_j$. The inner maximization problem always has at least one analytic solution that can be constructed by finding $q$ indices $i$ where $\alpha_j^\top\mathbf{X}_j^i\mathbf{X}_j^{i\top}\alpha_j$ take on the largest values in order, and assigning $s_i = 1$ to those indices. The outer maximization problem can be solved via a nonsmooth optimization algorithm. That is, for a given candidate dual solution $\hat{\alpha}_1,\ldots,\hat{\alpha}_J$ we analytically compute the optimal support vector and a subgradient $\nabla f(\hat{\alpha}_1,\ldots,\hat{\alpha}_J,\mathbf{s})$ and apply a suitable global first order method to Equation (2.8).

**Algorithm 1** SparClur

**Input:** $\mathbf{X}_j \in \mathbb{R}^{n_j \times p}, \mathbf{y}_j \in \mathbb{R}^{n_j}, \forall j \in [\![J]\!], q \in [\![p]\!], \gamma \in \mathbb{R}_+$
**Output:** $\mathbf{s} \in \{0,1\}^p$

1: **procedure** CUTTING PLANE ALGORITHM
2: $\mathbf{s}_1 \leftarrow$ warm start
3: $\eta_1 \leftarrow 0$
4: $v \leftarrow 0$
5: $c(\mathbf{s}_1) \leftarrow \infty$
6: **while** $\eta_v < c(\mathbf{s}_v)$
7: $v \leftarrow v+1$
8: **for** $j \in [\![J]\!]$
9: $\mathbf{X}_{s,j} \leftarrow \mathbf{X}_j[:,\mathbf{s}]$
10: $\alpha_j^* \leftarrow \mathbf{y}_j - \mathbf{X}_{s,j}(\mathbb{I}_q/\gamma + \mathbf{X}_{s,j}^\top \mathbf{X}_{s,j})^{-1}\mathbf{X}_{s,j}^\top \mathbf{y}_j$
11: $c(\mathbf{s}_v) \leftarrow \frac{1}{2}\sum_{j=1}^J \mathbf{y}_j^\top \alpha_j^*$
12: **for** $i \in [\![p]\!]$
13: $\nabla c(\mathbf{s}_v)_i \leftarrow -\frac{\gamma}{2}\sum_{j=1}^J (\mathbf{X}_j^{i\top}\alpha_j^*)^2$
14: $\mathbf{s}_{v+1}, \eta_{v+1} \leftarrow \arg\min_{\mathbf{s},\eta}\{\eta : \eta \geq c(\mathbf{s}_t) + \nabla c(\mathbf{s}_t)^\top(\mathbf{s}-\mathbf{s}_t), \forall t \in [\![v]\!], \mathbf{s} \in \{0,1\}^p, \mathbf{1}^\top \mathbf{s} \leq q\}$

## 3 Experiments with synthetic data

We address three key questions:

1. Does our mixed integer formulation recover correct solutions to the sparse regression problem, particularly in the presence of noise?
2. Does SparClur enjoy practical solving times as the dimensionality of a problem grows?
3. What is the cost of imposing the assumption of common support among clusters, when there is no such phenomenon in the underlying data?

We define accuracy $A$ and false positive rate $F$ as follows. Let $\text{supp}(\mathbf{w}_{true})$ denote the known true support in a synthetic dataset. Then for solution $\mathbf{w}^*$ we have:

$$A = \frac{|\text{supp}(\mathbf{w}_{true}) \cap \text{supp}(\mathbf{w}^*)|}{q}$$
$$F = \frac{|\text{supp}(\mathbf{w}^*) \setminus \text{supp}(\mathbf{w}_{true})|}{|\text{supp}(\mathbf{w}^*)|}.$$

All time related experiments were performed on a Linux system with an AMD Ryzen 9 3950X 16-Core processor. Our algorithms were implemented in Julia [8], and all optimization problems were built using JuMP [11] and solved in Cplex 12.10. The problem in Line 14 of Algorithm 1 was modified in each iteration using lazy callbacks.

### 3.1 Support recovery

Each entry of a matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$ was independently generated from a standard Gaussian distribution for $n$ ranging between 60 and 800 observations and $p = 2000$. Our

observations were randomly and evenly divided among $J$ clusters. The value of $J$ was varied in the range $\{1,2,5,10,20\}$. The set of features in the support was fixed with 10 randomly selected features. For each feature $i$ in the true support, a corresponding coefficient $w^i \in \{-1,1\}$ was sampled. The target variables were computed as $\mathbf{y}_j = \mathbf{X}_j \mathbf{w}_j + \xi_j$, where $\xi_j \sim N(\mathbf{0}, \Sigma)$ was scaled so that we have a signal-to-noise ratio $||\mathbf{y}_j||/||\xi_j|| = 20$.

We generated five synthetic datasets as described above for varying values of $n$, and report the mean out of sample accuracy and false positive rate for each $n$. These are shown in the plots of Figure 3.1. For each datapoint shown, the value of $\gamma$ was taken to be some constant multiplied by $q/n$. This value was picked following a cross validation procedure, where we picked the value of $\gamma$ that gave the highest MSE on separated data. The plots demonstrate the occurrence of a *phase transition*, and demonstrate that the point of this phase transition depends on the number of observations in each cluster and the number of clusters.

### *3.1.1 Time of phase transition*

Our experiments confirm the phenomenon observed by Bertsimas and Van Parys [4], where the solving time of a problem begins to *decrease* once some critical number of observations is exceeded. This has an interesting implication for the SparClur formulation. As an example, consider the computational time for the synthetic problem described above, shown in Figures 3.2a for $J = 1$ and 3.2b for $J = 5$. For a modeler computing regression weights at several leaf nodes, it is desirable for the number of observations at each leaf to be greater than the critical value mentioned, since this results in significantly lower solving times. A key advantage of SparClur is that the coordination imposed reduces the number of observations necessary to attain the solving times observed beyond the *phase transition*. For the case illustrated in Figures 3.2a and 3.2b, if a model consisted of five leaf nodes, then around 300 observations would be sufficient to achieve phase transition with SparClur. On the other hand, the single cluster model experiences a phase transition beyond 120 observations, meaning that if a modeler was to use an uncoordinated regression model at each leaf, $5 \times 120 = 600$ observations would be necessary.

## 3.2 Scalability

Table 3.1 summarizes the solving times we observe as we increase the number of observations to the range of the hundreds of thousands, and the number of features to the tens of thousands. At this scale, we are able to recover the full support with no false detection for all instances in seconds.

## 3.3 Effects of clusters with varying support

We now explore the behavior of SparClur when it is applied to observations that do not truly share the same support. To do so, we generate data for observations $\mathbf{X}_j$ and

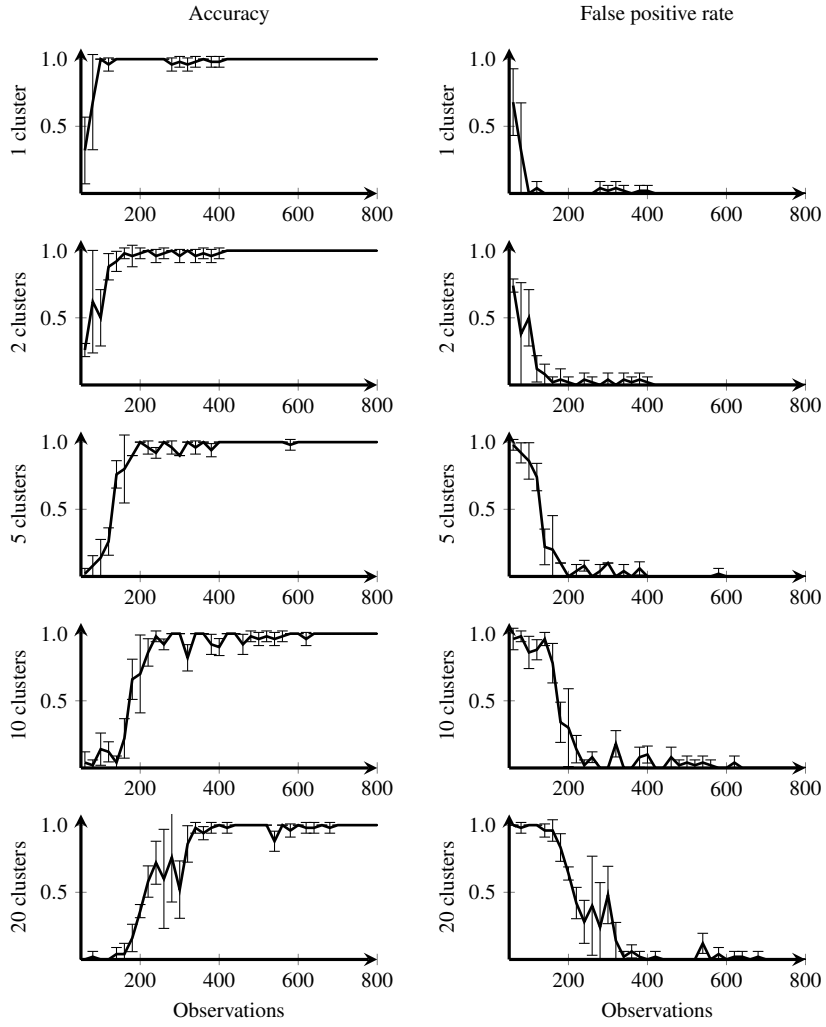
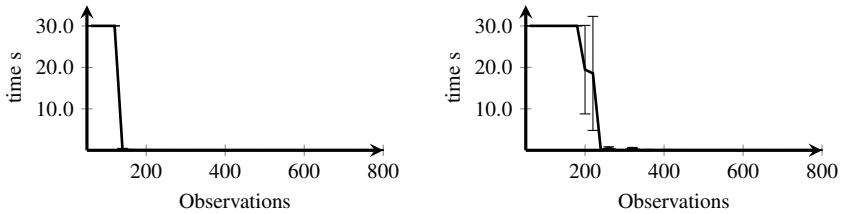



Fig. 3.1: Accuracy and false positive rate as a function of number of observations for synthetic data with SNR=20, $q$=10. Each curve passes through the average measurement made over five sets of synthetic data and error bars correspond to one standard deviation.

$\mathbf{y}_j$ as described in the previous section, with the number of clusters $J = 2$. The weights for each cluster $\mathbf{w}_1, \mathbf{w}_2$ were generated so that there are 10 features in the true support of each cluster, but not necessarily the same 10 features in both clusters. The number of features in $\{\text{supp}(\mathbf{w}_1) \cap \text{supp}(\mathbf{w}_2)\}$ was varied.

When we come to build a model for our synthetic data, we must assume some underlying sparsity $q$ which may be lower than, or greater than, the total number of




(a) One cluster (b) Five clusters

Fig. 3.2: Computational times in seconds as a function of number of observations.

| | | time | | |
|---|---|---|---|---|
| $\gamma$ | $p$ | $n = 20,000$ | $n = 50,000$ | $n = 100,000$ |
| 0.005 | 20,000 | 2.52 | 4.51 | 8.38 |
| 0.01 | 20,000 | 1.7 | 4.08 | 7.73 |
| 0.02 | 20,000 | 1.69 | 4.08 | 7.73 |
| 0.005 | 50,000 | 4.83 | 10.65 | 19.88 |
| 0.01 | 50,000 | 4.67 | 10.39 | 19.4 |
| 0.02 | 50,000 | 4.71 | 10.36 | 19.46 |

Table 3.1: Computational times in seconds (mean over five datasets) for different values of $n, p$, and $\gamma$ with $J = 10$. In each experiment accuracy was 100% and false positive rate was 0%.

features in both clusters. When $q < |\text{supp}(\mathbf{w}_1) \cup \text{supp}(\mathbf{w}_2)|$, it is not possible to attain an accuracy of 100%. Instead, the maximum attainable accuracy is $\frac{q}{|\text{supp}(\mathbf{w}_1) \cup \text{supp}(\mathbf{w}_2)|}$.

In Figures 3.3 and 3.4, the dashed curves correspond to the maximum attainable accuracy. The points correspond to the accuracy attained each time the problem is solved with SparClur. In every case, the accuracy matches closely with the maximum attainable accuracy, and we never detect any features not in the support of one of the two clusters, except when $q > |\text{supp}(\mathbf{w}_1) \cup \text{supp}(\mathbf{w}_2)|$. We do not claim, however, that when $q < |\text{supp}(\mathbf{w}_1) \cup \text{supp}(\mathbf{w}_2)|$, features in the support of both clusters, $|\text{supp}(\mathbf{w}_1) \cap \text{supp}(\mathbf{w}_2)|$, are always in the set of features discovered. Rather, any of the features in either support vector may be in the solution.

In practice, the choice of $q$ would be determined following a cross validation procedure. We wish to ensure that the correct value can be recovered when the ground truth is not known. Figure 3.5 shows the out of sample $R^2$ for different values of $q$. We see that the best choices of $q$ are 18 in the first experiment, and 15 in the second experiment from the figures. These values conform with $|\text{supp}(\mathbf{w}_1) \cup \text{supp}(\mathbf{w}_2)|$.

## 3.4 Findings from synthetic experiments

Our experiments confirm that:

1. The algorithm recovers the true support in a set of features when this support is known, and is capable of successfully ignoring irrelevant features.

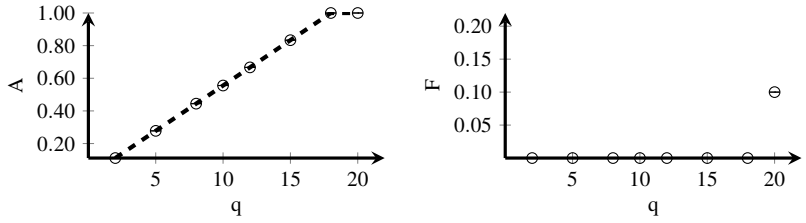



Fig. 3.3: Accuracy and false positive rate when $|\text{supp}(\mathbf{w}_1) \cap \text{supp}(\mathbf{w}_2)| = 2$.

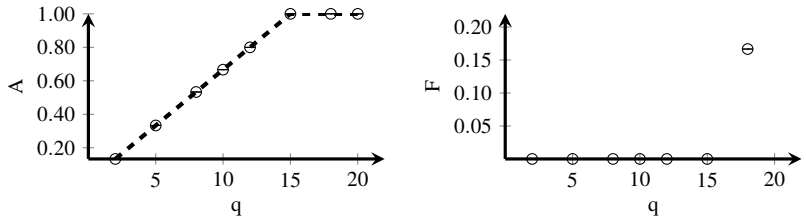



Fig. 3.4: Accuracy and false positive rate when $|\text{supp}(\mathbf{w}_1) \cap \text{supp}(\mathbf{w}_2)| = 5$.

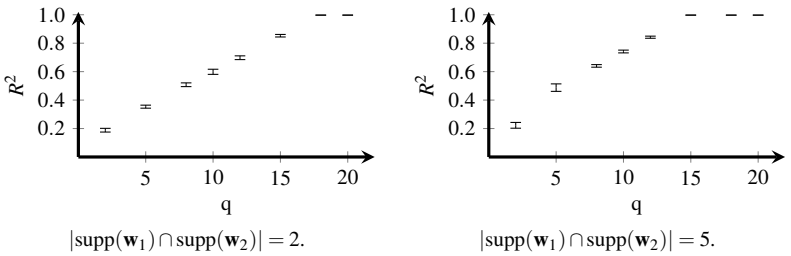



Fig. 3.5: Out of sample $R^2$ as a function of $q$ for two clusters without fully coordinated support.

2. The algorithm is practical for offline problems where the number of observations is in the hundreds of thousands, and the number of features is in the tens of thousands. That is, we can attain high quality solutions for problems of such scale in seconds. Furthermore, the quality of solutions for a fixed number of features has the potential to be higher with SparClur than with uncoordinated sparse regression, because we often require a *smaller* number of observations to be present before the phase transition phenomenon occurs.
3. When $q$ was chosen to be smaller than the total number of relevant features, no features which were not part of the true support were present in our models. When the underlying regression models in clusters do not truly share a common sparsity pattern, there is a tradeoff to be made between interpretability (which improves as $q$ decreases) and accuracy (which improves up to a limit as $q$ increases).

## 4 Experiments with real world data

The first medical prediction task we look at is the problem of predicting glucose levels in diabetes patients. Oftentimes, clinicians intuitively use a uniform series of questions to arrive at an estimate for patient outcomes. Key characteristics of the patient such as weight, age, and medical history could be taken into consideration. However with an algorithmic approach, different models can be applied to predict glucose levels in different patients. The second dataset we turn our attention to is derived from the Framingham Heart Study [1]. The same dataset has been used to study a number of medical conditions, due to the richness of the features collected and the longitudinal aspect of the study. Here, our goal is to predict two outcomes of interest: the change in blood pressure of patients at subsequent visits, and the time of stroke occurrence from the first observation of a subject. In addition, we test SparClur using a low dimensional, publicly available dataset from [13]. The dataset is used to predict sale prices of houses in King County, Washington State.

### 4.1 Description of data

For the glucose prediction problem, we obtained electronic medical records (EMR) of 10,806 Boston Medical Center (BMC) patients who met the inclusion criteria described in [6]. We had access to demographic data, including date of birth, sex, and ethnicity, and to all BMC EMR data, including a history of drug prescriptions and measurements of height, weight, BMI, and HbA1c (an indicator of past blood sugar levels) as well as creatinine levels. All together, the model considered 85 features.

The Framingham Heart Study contains the examination data of 41 clinical exams from 1948 to 2010. Our data comprises two cohorts. The *Original Cohort* is a random sample of 5,209 respondents between 30 and 62 years of age, tracked from 1948 until 2010. The *Offspring Study* Cohort was initiated in 1971 comprising 5,124 adult offspring of the Original Cohort. Unifying the two cohorts, we have patient characteristic data at each visitation for 10,092 unique patients. Recorded characteristics include age, gender, and BMI, as well as biological information derived from blood test results.

For the time of stroke prediction problem, we considered only the 10% of patients who experienced a stroke. We retained health information from patients' initial baseline visit and computed the number of days from when that data was collected to the date of their first occurrence of stroke. Overall, the model we built had 1,266 observations and 40 features. For the blood pressure prediction problem, we calculated the change in systolic blood pressure of all patients between consecutive visits, and treated each pair of consecutive visits as an observation. The final model had 91,955 observations and 41 features.

The dataset for house sale price predictions contained 21,613 observations with 18 features describing each property. The features used and output from ORTs are available with our code [14].

| | | | SparClur | | |
|---|---|---|---|---|---|
| depth | Lasso | point | exact | relaxation | sparse |
| 0 | 0.506 | 0.000 | 0.499 | 0.464 | 0.464 |
| 1 | 0.521 | 0.323 | 0.514 | 0.498 | 0.490 |
| 2 | 0.524 | 0.438 | 0.517 | 0.502 | 0.490 |
| 3 | 0.532 | 0.476 | 0.524 | 0.509 | 0.490 |
| 4 | 0.535 | 0.502 | 0.530 | 0.525 | 0.495 |
| 5 | 0.535 | 0.511 | 0.530 | 0.526 | 0.497 |

Table 4.1: Mean (over five folds) out of sample $R^2$ for prediction of glucose levels using different depths.

## 4.2 Comparison of methods

We examined the performance of ORTs with point predictions at leaf nodes, as well as linear prediction models at leaf nodes. When using linear prediction models, we compared the performance of SparClur with several other techniques. All linear prediction models were constructed after the optimal regression tree (with point predictions at the leaves) was found and the parameters for the tree were cross validated to minimize the MSE. The leaves of each tree were treated as clusters. We considered building linear models using Lasso (with the implementation in [15]), sparse regression (without coordination as in [7]), SparClur with Algorithm 1, and the convex relaxation of SparClur.

We set a time limit of two minutes in CPLEX and used the incumbent solution if the time limit was reached. When using uncoordinated sparse regression, we did not employ an exact cutting plane algorithm, but used the convex relaxation suggested by Bertsimas et al. [7]. This is because training and cross-validating for appropriate parameters using uncoordinated sparse regression, which involves building linear models within each leaf separately, would take over a week of computational time for our deepest trees, which we consider impractical.

We measured the average out of sample accuracy from five different training and testing splits of our data and report the averaged out of sample $R^2$ for trees of increasing depth. Our results are summarized in Tables 4.1 to 4.4. Each tree was created using the software package `OptimalTrees.jl` described in [10] and for each regression model, the hyperparameters $q$ (in the range 1—10) and $\gamma$ were chosen following a cross validation procedure. We chose to restrict the sparsity of our models to 10 features (including a bias term) or fewer, as we consider this to be an appropriate number of features for clinicians to interpret.

The models created for glucose and blood pressure prediction attained $R^2$ scores of around 0.5. The best models we produced for the time of stroke prediction problem (which had the fewest observations) attained $R^2$ scores of 0.3–0.4. The best models we produced for predicting house prices, attained $R^2$ scores of 0.7–0.8.

In each example, the trees we obtained with sparse regression methods had fewer features than the trees obtained with Lasso at the leaves. This difference was more significant for the medical datasets. In Table 4.5, we show the minimum, maximum, and mean number of features that appear per leaf using uncoordinated sparse re-

| | | | SparClur | | |
|---|---|---|---|---|---|
| depth | Lasso | point | exact | relaxation | sparse |
| 0 | 0.364 | -0.005 | 0.352 | 0.261 | 0.261 |
| 1 | 0.359 | 0.172 | 0.365 | 0.303 | 0.272 |
| 2 | 0.358 | 0.234 | 0.350 | 0.312 | 0.316 |
| 3 | 0.357 | 0.253 | 0.342 | 0.316 | 0.320 |
| 4 | 0.357 | 0.253 | 0.341 | 0.316 | 0.320 |
| 5 | 0.357 | 0.253 | 0.341 | 0.316 | 0.312 |

Table 4.2: Mean (over five folds) out of sample $R^2$ for prediction of days until stroke onset using different depths.

| | | | SparClur | | |
|---|---|---|---|---|---|
| depth | Lasso | point | exact | relaxation | sparse |
| 0 | 0.306 | 0.000 | 0.296 | 0.281 | 0.281 |
| 1 | 0.341 | 0.151 | 0.328 | 0.319 | 0.300 |
| 2 | 0.521 | 0.372 | 0.516 | 0.503 | 0.465 |
| 3 | 0.527 | 0.459 | 0.523 | 0.513 | 0.468 |
| 4 | 0.528 | 0.487 | 0.522 | 0.518 | 0.506 |
| 5 | 0.528 | 0.498 | 0.524 | 0.520 | 0.506 |

Table 4.3: Mean (over five folds) out of sample $R^2$ for prediction of change in blood pressure using different depths.

| | | | SparClur | | |
|---|---|---|---|---|---|
| depth | Lasso | point | exact | relaxation | sparse |
| 0 | 0.693 | 0.00 | 0.691 | 0.684 | 0.684 |
| 1 | 0.750 | 0.314 | 0.742 | 0.719 | 0.711 |
| 2 | 0.780 | 0.491 | 0.774 | 0.735 | 0.734 |
| 3 | 0.815 | 0.605 | 0.814 | 0.786 | 0.791 |
| 4 | 0.823 | 0.660 | 0.823 | 0.809 | 0.810 |
| 5 | 0.826 | 0.697 | 0.831 | 0.820 | 0.818 |

Table 4.4: Mean (over five folds) out of sample $R^2$ for prediction of house prices using different depths.

gression, Lasso and SparClur for a tree of depth five (this depth is typically where out-of-sample $R^2$ values begin to plateau). The values in the table are averaged over five folds of data. To get a sense of the interpretability of these models, we also report the number of features that appear in more than half of the leaves, and the number of features that appear in fewer than half of the leaves. Note that it is possible (but rare) for certain features to appear in only some leaves when using SparClur, since our parameter $q$ is only an upper bound on the number of nonzero coefficients.

| | | # features per leaf | | | # features repeating | |
|---|---|---|---|---|---|---|
| | | min | max | mean | $>$ half | $\leq$ half |
| gluc. | Lasso | 1.0 | 52.2 | 21.2 | 6.8 | 74.0 |
| | sparse | 7.0 | 9.4 | 9.3 | 4.4 | 44.8 |
| | SparClur | 7.2 | 9.0 | 8.83 | 9.0 | 0.0 |
| press. | Lasso | 0.2 | 34.4 | 14.3 | 6.4 | 31.2 |
| | sparse | 9.8 | 10.0 | 9.99 | 6.0 | 29.8 |
| | Sparclur | 5.6 | 9.6 | 8.6 | 8.6 | 0.0 |
| days. | Lasso | 0.2 | 16.6 | 8.5 | 2.8 | 16.2 |
| | sparse | 6.6 | 6.6 | 6.6 | 4.6 | 7.6 |
| | Sparclur | 5.2 | 6.8 | 6.0 | 5.8 | 0.4 |
| house | Lasso | 1.0 | 16.0 | 10.5 | 12.0 | 6.0 |
| | sparse | 3.2 | 10.0 | 8.9 | 7.8 | 10.2 |
| | SparClur | 9.8 | 9.8 | 9.8 | 9.8 | 0.0 |

Table 4.5: Minimum, maximum, and mean number of features per leaf in a tree of depth 5, as well as number of features that repeat in more than half the leaves and fewer than half the leaves. All values are averaged over five folds. The total number of features in the datasets were 85, 40, 41 and 18.

### 4.3 General trends

The results in Tables 4.1 to 4.3 and 4.5 display some interesting patterns. Within the medical datasets, each of the regression methods examined attained an $R^2$ score within 5% of the other methods, apart from the point-prediction model for time of stroke. Yet, the difference in interpretability between the models is substantial. In particular, the results in Table 4.5 show that Lasso gave rise to models where the number of features in each node was highly variable, and some nodes included more than half the features in the data. Furthermore, Table 4.5 shows that both Lasso and uncoordinated sparse regression gave rise to relatively few features that appear in more than half of the leaves, and many features that appear in fewer than half of the leaves, although it does not make medical sense for so many features to affect only subsets of the population. Given that the difference in interpretability between the models is significant, the similarity in $R^2$ scores demonstrates that the price to be paid for imposing additional structure that favors this interpretability is small.

Interestingly, the uncoordinated sparse regression model had inferior out of sample performance to optimal regression trees with point predictions in the glucose prediction problem, and inferior performance to SparClur in the other problems. This would typically be an indicator of overfitting. In this case, the behavior could be an artifact of a large number of clusters and an insufficiently large number of observations within each cluster to attain good performance with uncoordinated sparse regression. The performance of SparClur is always favorable compared to point predictions at high depths. We also make the observation that the optimization of the exact coordinated regression problem provides a relatively small improvement in the out of sample $R^2$ over the relaxation solution in each case study, which was used as a warm start.

For stroke prediction, the tree structure did not provide as much lift (depth 0/1 performed best). For the other cases, however, the $R^2$ increased significantly with depth. This could be because the features in the stroke prediction problem did not differ much in the leaves, indicating that the problem is more uniform across certain demographics than believed.

## 5 Conclusions

We offer SparClur as an approach for building regression models within tree based prediction methods, combining the goals of accuracy and interpretability. SparClur enforces additional structure within predictive models, but leads to models that are arguably more interpretable than other linear regression methods. Furthermore, we have shown with synthetic data that the method is correct, scalable, and capable of attaining more favourable results than sparse regression without coordination when the number of observations available is below a certain threshold. In the large scale datasets we studied, SparClur improves on the accuracy of ORTs with point predictions, and has very similar out of sample accuracy to models utilizing uncoordinated sparse regression, and Lasso. In other words, we see a substantial gain in interpretability at a very small cost to accuracy.

## Data Availability Statement

All synthetic datasets and all publicly available datasets are available to interested readers. Medical data are protected under privacy rules and are not available.

## References

1. Benjamin EJ, Levy D, Vaziri SM, D'agostino RB, Belanger AJ, Wolf PA (1994) Independent risk factors for atrial fibrillation in a population-based cohort: the Framingham Heart Study. Jama 271(11):840–844
2. Bertsimas D, Copenhaver MS (2018) Characterization of the equivalence of robustification and regularization in linear and matrix regression. European Journal of Operational Research 270:931–942
3. Bertsimas D, Dunn J (2019) Machine Learning Under a Modern Optimization Lens. Dynamic Ideas
4. Bertsimas D, Van Parys B (2020) Sparse high-dimensional regression: Exact scalable algorithms and phase transitions. Annals of Statistics 48(1):300–323
5. Bertsimas D, King A, Mazumder R, et al. (2016) Best subset selection via a modern optimization lens. Annals of Statistics 44(2):813–852
6. Bertsimas D, Kallus N, Weinstein AM, Zhuo YD (2017) Personalized diabetes management using electronic medical records. Diabetes Care 40(2):210–217
7. Bertsimas D, Pauphilet J, Van Parys B (2017) Sparse classification and phase transitions: A discrete optimization perspective. arXiv preprint arXiv:171001352

8. Bezanson J, Edelman A, Karpinski S, Shah VB (2017) Julia: A fresh approach to numerical computing. SIAM review 59(1):65–98
9. Breiman L, Friedman J, Stone CJ, Olshen RA (1984) Classification and regression trees. CRC press
10. Dunn J (2018) Optimal trees for prediction and prescription. PhD thesis, Massachusetts Institute of Technology
11. Dunning I, Huchette J, Lubin M (2017) Jump: A modeling language for mathematical optimization. SIAM Review 59(2):295–320
12. Duran MA, Grossmann IE (1986) An outer-approximation algorithm for a class of mixed-integer nonlinear programs. Mathematical programming 36(3):307–339
13. Kaggle (2016) House sales in King County, USA. `https://www.kaggle.com/harlfoxem/housesalesprediction`, accessed: 2020-12-05
14. Kapelevich L, Zhang R (2020) Sparclur/Sparclur.jl: v0.1. DOI 10.5281/zenodo.4308537, URL `https://doi.org/10.5281/zenodo.4308537`
15. Kornblith S, Contributors (2013) GLMNet.jl: Julia wrapper for fitting Lasso/ElasticNet GLM models using glmnet. `https://github.com/JuliaStats/GLMNet.jl`
16. Tibshirani R (1996) Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society Series B (Methodological) pp 267–288
17. Tikhonov AN (1943) On the stability of inverse problems. In: Dokl. Akad. Nauk SSSR, vol 39, pp 195–198