Journal of Machine Learning Research 17 (2016) 1-31 Submitted 7/15; Revised 1/16; Published 6/16

Convex Regression with Interpretable Sharp Partitions

Ashley Petersen AJPETE@UW.EDU

Noah Simon NRSIMON@UW.EDU
Department of Biostatistics
University of Washington
Seattle, WA 98195

Daniela Witten DWITTEN@UW.EDU
Departments of Biostatistics and Statistics
University of Washington
Seattle, WA 98195

Editor: Maya Gupta

Abstract

We consider the problem of predicting an outcome variable on the basis of a small number of covariates, using an interpretable yet non-additive model. We propose *convex regression with interpretable sharp partitions* (CRISP) for this task. CRISP partitions the covariate space into blocks in a data-adaptive way, and fits a mean model within each block. Unlike other partitioning methods, CRISP is fit using a non-greedy approach by solving a convex optimization problem, resulting in low-variance fits. We explore the properties of CRISP, and evaluate its performance in a simulation study and on a housing price data set.

Keywords: convex optimization, interpretability, non-additivity, non-parametric regression, prediction

1. Introduction

Classification and regression trees (CART) are immensely popular for flexible and non-additive predictive modeling, despite the fact that they date back more than thirty years (Breiman et al., 1984). The trees are fit using a two-stage process in which the tree is first greedily "grown" to some maximum size, and then "pruned" to avoid overfitting. The final tree with K terminal nodes can be visually displayed as a decision tree with $K-1$ splits, or equivalently as K disjoint boxes that completely partition the covariate space. CART has stood the test of time, because its output is highly interpretable and it can easily incorporate complex non-additive relationships between features. However, it is a greedy procedure, and a small perturbation of the data can produce a very different tree. The high variability of the fitted values can compromise the scientific utility of the tree, as well as the tree's prediction accuracy on test data. While an ensemble approach, like random forests, can reduce CART's variability, this comes at the expense of interpretability (Breiman, 2001).

Two other well-known methods for flexible and non-additive predictive modeling are multivariate adaptive regression splines (MARS) (Friedman, 1991) and thin-plate splines (TPS) (Duchon, 1977). The MARS fit is a weighted sum of basis functions, which are

greedily chosen and some of which involve pairs of features. TPS fits the observed data, regularized by smoothness penalties. In the case of two covariates x_1 and x_2 and a response y, the TPS fit is the solution to

$$\underset{f}{\text{minimize}} \quad \sum_{i=1}^{n}[y_i - f(x_{1i}, x_{2i})]^2 + \lambda \int\int_{\mathbb{R}^2} \|\nabla^2 f(x_1, x_2)\|_F^2 \, dx_1 \, dx_2.$$

The fits from MARS and TPS are incredibly flexible, but can be less interpretable than the fits from CART.

In recent years, the statistical community has been very interested in formulating predictive models as solutions to convex optimization problems. However, to the best of our knowledge, no proposals have been made for flexible, non-additive, and interpretable modeling via convex optimization. To close this gap, we propose a non-greedy procedure whose fits have a block structure reminiscent of CART. Our proposal, *convex regression with interpretable sharp partitions* (CRISP), is the solution to a convex optimization problem with predictions that are much less variable than those of CART. Also unlike CART, CRISP borrows information across the blocks, and is able to adequately model the data when the mean model is smooth. Thus our method provides a compromise between the interpretability of CART and the flexibility of MARS and TPS. In this paper, we consider the low-dimensional setting in which there are a small number of covariates of interest ($p \ll n$). We leave an extension to the $p > n$ setting to future work.

CRISP has a number of attractive properties:

- CRISP can accommodate interactions between pairs of covariates in a flexible way. This is useful when the impact of one covariate may depend on the value of another covariate, but there is not strong *a priori* knowledge about the form of the interaction.
- CRISP fits a piecewise constant model, which is easily interpreted by even those with limited statistical background.
- CRISP is formulated as a convex optimization problem. Thus we can solve for the global optimum, and can derive an expression for CRISP's degrees of freedom.

The remainder of this paper is organized as follows. In Section 2, we introduce our method and present an algorithm to implement it. We compare our method to existing approaches using simulated data in Section 3. In Section 4, we derive some properties of the method. In Section 5, we discuss connections between our method and other work. We illustrate our method on a housing price data set in Section 6. We consider a modification to our proposal in Section 7, and close with the discussion in Section 8. Proofs are in the Appendix.

2. Convex Regression with Interpretable Sharp Partitions

Throughout most of this paper, for ease of exposition, we focus on the case of $p = 2$ features. An extension to the case of $p > 2$ is given in Section 7.

We first present an overview of the CRISP approach. We wish to predict a random variable $y \in \mathbb{R}$ using $x_1, x_2 \in \mathbb{R}$. We assume that $y = f(x_1, x_2) + \epsilon$, where ϵ is a mean-zero

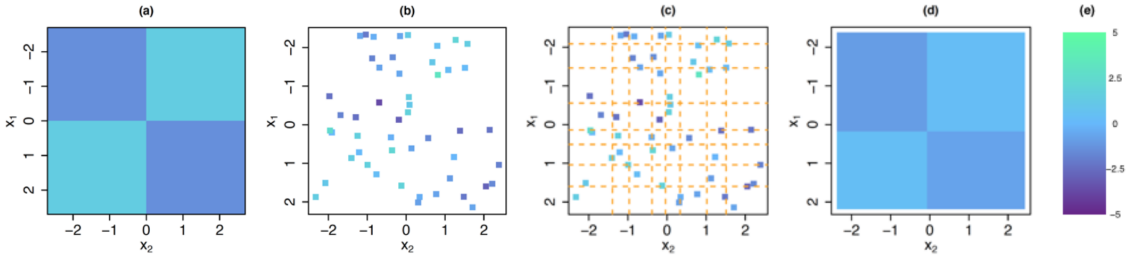

Figure 1: In *(a)*, the mean model $f(x_1, x_2)$ used to generate data. In *(b)*, each of the 50 squares represents an observation (x_1, x_2, y) with $y = f(x_1, x_2) + \epsilon$ with $\epsilon \sim N(0, 1)$. In *(c)*, there are $q^2 = 64$ bins of (x_1, x_2) values, whose boundaries coincide with the octiles (- - -) of $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. In *(d)*, CRISP estimates $f(x_1, x_2)$ to be constant within each bin, and furthermore encourages adjacent bins to take on the same value. When applied to the data in *(b)* with $q = 8$, this leads to an estimated $f(x_1, x_2)$ with four *blocks*. In *(e)*, we show the heat scale legend.

error term, and f is an unknown function that we wish to estimate. An example of $f(x_1, x_2)$ is displayed in Figure 1(a). Figure 1(b) displays a training set of n i.i.d. observations of (x_1, x_2, y). We first partition the feature space into q^2 bins, as shown in Figure 1(c) with $q = 8$. The CRISP approach estimates $f(x_1, x_2)$ to be constant within each bin, and further encourages f to take on the same value at adjacent bins; this leads to constant-valued *blocks*. The CRISP output is shown in Figure 1(d); there are four estimated blocks. More details about this simulation set-up are provided in Section 3.

2.1 Notation and Goal of CRISP

We now introduce some new notation, and provide further intuition for CRISP, before presenting the optimization problem for CRISP in Section 2.2.

As is shown in Figure 1(c), we wish to estimate the mean model $f(x_1, x_2)$ for a $q \times q$ grid of bins, where $f(x_1, x_2)$ is estimated to be constant within each bin. Let $\boldsymbol{M} \in \mathbb{R}^{q \times q}$ denote a mean matrix whose element $M_{(i)(j)}$ contains the mean for pairs of covariate values within a *quantile range* of the observed predictors $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{R}^n$. For example, $M_{(1)(2)}$ represents the mean of the observations with x_1 less than the $\frac{1}{q}$-quantile of $\boldsymbol{x}_1$, and x_2 between the $\frac{1}{q}$- and $\frac{2}{q}$-quantiles of $\boldsymbol{x}_2$. In Figure 1(c), the corner grid bins correspond to $M_{(1)(1)}$, $M_{(8)(1)}$, $M_{(8)(8)}$, and $M_{(1)(8)}$, starting at the top-left corner of the grid and moving counter-clockwise. In CRISP, our goal is to estimate the $q \times q$ matrix $\boldsymbol{M}$ on the basis of $\boldsymbol{y} \in \mathbb{R}^n$, which contains n noisy observations from various bins of $\boldsymbol{M}$.

In the example shown in Figure 1, we partition the feature space into an 8×8 grid (shown in Figure 1(c)), which translates to estimating an 8×8 matrix $\boldsymbol{M}$. Therefore, instead of estimating $f(x_1, x_2)$ over the entire joint range of $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, we need only estimate the 64 elements of $\boldsymbol{M}$. Furthermore, CRISP borrows information across bins of the grid by encouraging pairs of neighboring rows and columns of $\boldsymbol{M}^*$ to be equal, leading

to an estimated mean model with a *block structure*. For instance, in Figure 1(d), $\boldsymbol{M}$ is estimated to have four *blocks*, or regions of feature space over which $f(x_1, x_2)$ is constant. Consequently, the CRISP solution $\boldsymbol{M}^*$ shown in Figure 1(d) only has 4 unique elements, while $\boldsymbol{M}$ is an 8×8 matrix. If we examined the estimate $\boldsymbol{M}^*$, we would see that all pairs of neighboring rows and neighboring columns of $\boldsymbol{M}^*$ are identical, except for one pair of columns and one pair of rows.

While the true mean model in this example has a block structure (as seen in Figure 1(a)), we will show in Section 3 that CRISP can perform well even when the true mean model is smooth. The data in this example were uniformly distributed in the covariate space. CRISP is most suitable for data applications where observations are distributed throughout the covariate space. Highly correlated covariates will lead to an insufficient amount of data to estimate the mean model over the entire covariate space.

2.2 The Optimization Problem

The CRISP optimization problem balances the trade-off between fitting the data and encouraging a block structure. We estimate $\boldsymbol{M}$ by solving the convex optimization problem

$$\underset{\boldsymbol{M} \in \mathbb{R}^{q \times q}}{\text{minimize}} \quad \frac{1}{2} \sum_{i=1}^{n} \left(y_i - \Omega(\boldsymbol{M}, x_{1i}, x_{2i})\right)^2 + \lambda P(\boldsymbol{M}). \tag{1}$$

In (1), the function Ω extracts the element of $\boldsymbol{M}$ corresponding to the bin to which the observation (x_{1i}, x_{2i}) belongs. For instance, in Figure 1(c), $\Omega(\boldsymbol{M}, 0, -1) = M_{(4)(2)}$. Note that Ω is explicitly defined in Appendix A. Furthermore, $\lambda \geq 0$ is a tuning parameter, and the penalty P is defined as

$$P(\boldsymbol{M}) = \sum_{i=1}^{q-1} \left[\left\| \boldsymbol{M}_{i\cdot} - \boldsymbol{M}_{(i+1)\cdot} \right\|_2 + \left\| \boldsymbol{M}_{\cdot i} - \boldsymbol{M}_{\cdot(i+1)} \right\|_2 \right], \tag{2}$$

where $\boldsymbol{M}_{i\cdot}$ and $\boldsymbol{M}_{\cdot i}$ denote the ith row and column of $\boldsymbol{M}$, respectively. The sum of squared errors in (1) encourages the estimate of $\boldsymbol{M}$ to fit the data, while the group lasso penalty (Yuan and Lin, 2006) in (2) encourages pairs of neighboring rows (or columns) to be exactly identical. This leads to the formation of constant-valued blocks, which are comprised of multiple bins of the $q \times q$ grid. Appendix B discusses other possible penalties that could be used in (1).

We now rewrite (1) in a way that will be useful later. We introduce a vectorized form of $\boldsymbol{M}$, which is denoted by $\boldsymbol{m} = \text{vec}(\boldsymbol{M}) = \left((\boldsymbol{M}_{\cdot 1})^T, (\boldsymbol{M}_{\cdot 2})^T, \cdots, (\boldsymbol{M}_{\cdot q})^T\right)^T$ where $\boldsymbol{M}_{\cdot i}$ is the ith column of $\boldsymbol{M}$. The correspondence between $\boldsymbol{M}$ and $\boldsymbol{m}$ is shown in Figure 10 of Appendix A. In what follows, we will switch between using the matrix $\boldsymbol{M}$ and the vectorized $\boldsymbol{m}$. Then (1) can be rewritten as

$$\underset{\boldsymbol{m} \in \mathbb{R}^{q^2}}{\text{minimize}} \quad \frac{1}{2} \left\| \boldsymbol{y} - \boldsymbol{Q}\boldsymbol{m} \right\|_2^2 + \lambda \sum_{i=1}^{q-1} \left[\left\| \boldsymbol{R}_i \boldsymbol{m} \right\|_2 + \left\| \boldsymbol{C}_i \boldsymbol{m} \right\|_2 \right], \tag{3}$$

where each row of $\boldsymbol{Q} \in \mathbb{R}^{n \times q^2}$ contains $q^2 - 1$ elements that equal 0, and a single 1, such that $\boldsymbol{Q}_{i\cdot}\boldsymbol{m} = \Omega(\boldsymbol{M}, x_{1i}, x_{2i})$, where $\boldsymbol{Q}_{i\cdot}$ indicates the ith row of $\boldsymbol{Q}$. (Though $\boldsymbol{Q}$ is a function

of $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, we suppress this to simplify the notation.) In (3), $\boldsymbol{R}_i, \boldsymbol{C}_i \in \mathbb{R}^{q \times q^2}$ extract differences between neighboring rows and columns of $\boldsymbol{M}$ (i.e., $\boldsymbol{R}_i \boldsymbol{m} = \boldsymbol{M}_{i\cdot} - \boldsymbol{M}_{(i+1)\cdot}$ and $\boldsymbol{C}_i \boldsymbol{m} = \boldsymbol{M}_{\cdot i} - \boldsymbol{M}_{\cdot(i+1)}$). An example of $\boldsymbol{Q}$ and explicit definitions of $\boldsymbol{Q}$, $\boldsymbol{R}_i$, and $\boldsymbol{C}_i$ are in Appendix A. We let $\boldsymbol{A} = \left(\boldsymbol{R}_1^T, \cdots, \boldsymbol{R}_{q-1}^T, \boldsymbol{C}_1^T, \cdots, \boldsymbol{C}_{q-1}^T\right)^T \in \mathbb{R}^{2q(q-1) \times q^2}$, and then rewrite (3) as

$$\underset{\boldsymbol{m} \in \mathbb{R}^{q^2}, \boldsymbol{z} \in \mathbb{R}^{2q(q-1)}}{\text{minimize}} \quad \frac{1}{2} \|\boldsymbol{y} - \boldsymbol{Q}\boldsymbol{m}\|_2^2 + \lambda \sum_{i=1}^{q-1} [\|\boldsymbol{z}_{1i}\|_2 + \|\boldsymbol{z}_{2i}\|_2] \quad \text{subject to } \boldsymbol{A}\boldsymbol{m} = \boldsymbol{z}, \tag{4}$$

where $\boldsymbol{z} = \left((\boldsymbol{z}_{11})^T, \cdots, (\boldsymbol{z}_{1(q-1)})^T, (\boldsymbol{z}_{21})^T, \cdots, (\boldsymbol{z}_{2(q-1)})^T\right)^T$ with $\boldsymbol{z}_{1i}, \boldsymbol{z}_{2i} \in \mathbb{R}^q$.

While (1), (3), and (4) have the same solution, it is most convenient to derive an algorithm to solve CRISP using the parameterization in (4). Throughout this paper, we will alternate between using the notation $\boldsymbol{M}^*$ and $\boldsymbol{m}^*$, where $\boldsymbol{m}^* = \text{vec}(\boldsymbol{M}^*)$, to represent the CRISP solution to (4). The training set predictions for CRISP are given by $\hat{\boldsymbol{y}} = \boldsymbol{Q}\boldsymbol{m}^*$.

2.3 An Algorithm for CRISP

We solve for the global optimum of the convex optimization problem (4) using the *alternating directions method of multipliers* (ADMM) algorithm (Boyd et al., 2011). This is summarized in Algorithm 1. Additional details are in Appendix C.

Algorithm 1 — Alternating Directions Method of Multipliers for Equation (4)

1. Let $\boldsymbol{u} = \left((\boldsymbol{u}_{11})^T, \ldots, (\boldsymbol{u}_{1(q-1)})^T, (\boldsymbol{u}_{21})^T, \ldots, (\boldsymbol{u}_{2(q-1)})^T\right)^T$ denote the scaled dual variables. Initialize $\boldsymbol{m}^{(0)} := \boldsymbol{0}, \boldsymbol{z}^{(0)} := \boldsymbol{0}$, and $\boldsymbol{u}^{(0)} := \boldsymbol{0}$.

2. For $k = 1, 2, \ldots$, until the primal and dual residuals satisfy a stopping criterion:

   (a) $\boldsymbol{m}^{(k)} := \left[\boldsymbol{Q}^T\boldsymbol{Q} + \rho\boldsymbol{A}^T\boldsymbol{A}\right]^{-1}\left[\boldsymbol{Q}^T\boldsymbol{y} + \rho\boldsymbol{A}^T(\boldsymbol{z}^{(k-1)} - \boldsymbol{u}^{(k-1)})\right]$

   (b) $\boldsymbol{z}_{1i}^{(k)} := (\boldsymbol{R}_i\boldsymbol{m}^{(k)} + \boldsymbol{u}_{1i}^{(k-1)})(1 - \lambda/(\rho\|\boldsymbol{R}_i\boldsymbol{m}^{(k)} + \boldsymbol{u}_{1i}^{(k-1)}\|_2))_+$,
   $\boldsymbol{z}_{2i}^{(k)} := (\boldsymbol{C}_i\boldsymbol{m}^{(k)} + \boldsymbol{u}_{2i}^{(k-1)})(1 - \lambda/(\rho\|\boldsymbol{C}_i\boldsymbol{m}^{(k)} + \boldsymbol{u}_{2i}^{(k-1)}\|_2))_+$ for $i = 1, \ldots, q-1$

   (c) $\boldsymbol{u}^{(k)} := \boldsymbol{u}^{(k-1)} + \boldsymbol{A}\boldsymbol{m}^{(k)} - \boldsymbol{z}^{(k)}$

In Algorithm 1, the computational bottleneck occurs in Step 2(a). Evaluating the q-banded matrix $\boldsymbol{Q}^T\boldsymbol{Q} + \rho\boldsymbol{A}^T\boldsymbol{A}$ has a one-time cost of $\mathcal{O}(n+q^4)$ operations, and computing its LU factorization requires an additional $\mathcal{O}(q^4)$ operations. Then Step 2(a) can be performed in $\mathcal{O}(q^3)$ operations (Boyd and Vandenberghe, 2004). Therefore, Algorithm 1 requires an initial step of $\mathcal{O}(n + q^4)$ operations, followed by a per-iteration complexity of $\mathcal{O}(q^3)$.

On a Macbook Pro with a 2.0 GHz Intel Sandy Bridge Core i7 processor, our `Python` implementation of CRISP with $n = q = 50$ takes 20.1 seconds for a sequence of 20 λ values. For $n = q = 100$ and $n = q = 200$, the run times are 84.7 and 383.6 seconds, respectively. Increasing n while holding q constant has little effect on the run times; this is consistent with the discussion in the previous paragraph. Thus even for very large n, the computational time is reasonable.

We chose to solve CRISP using an ADMM algorithm, as ADMM works well in related problems. For example, in the context of trend filtering, Ramdas and Tibshirani (forthcoming) found that their ADMM implementation converged more reliably across a variety of tuning parameter values and sample sizes than the primal-dual interior point method of Kim et al. (2009). In our setting, an interior point algorithm for CRISP involves solving a dense system of equations at each iteration, which has a computational complexity of $\mathcal{O}(q^6)$. Additionally, an interior point method would not recover the exact block structure (any strictly feasible solution would have no zero row or column differences). In contrast, we directly recover the block structure of our estimated mean model from the $\boldsymbol{z}$ variables of our ADMM algorithm. Furthermore, ADMM algorithms typically converge to moderate accuracy within only tens of iterations (Boyd et al., 2011), which is acceptable in our setting.

The value of λ can be chosen using K-fold cross-validation. Alternatively, λ can be selected using approaches based on Akaike's information criterion (AIC; Akaike, 1973) or Bayesian information criterion (BIC; Schwarz, 1978) using the degrees of freedom estimator proposed in Section 4.1. The roles of λ and q in controlling the granularity of the model are further characterized in Sections 4.2 and 4.3.

3. Simulations

In this section, we compare the performance of CRISP to CART, TPS, and competing methods. We consider a variety of mean models, as well as smaller ($n = 100$) and larger ($n = 10,000$) training set sample sizes.

3.1 Methods

We generate data with either $n = 100$ or $n = 10,000$, and $p = 2$. We independently sample each element of $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ from a Unif$[-2.5, 2.5]$ distribution, and then take $\boldsymbol{y} = f(\boldsymbol{x}_1, \boldsymbol{x}_2) + \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \text{MVN}(\boldsymbol{0}, \sigma^2 \boldsymbol{I}_n)$ with $\sigma = 1$ for $n = 100$ and $\sigma = 10$ for $n = 10,000$. Note that we use the notation MVN to indicate a multivariate normal distribution.

We consider four mean models for $f(x_1, x_2)$; these are displayed in the top panel of Figure 2, and defined in detail in Appendix D. In Scenario 1, the mean model is additive in $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. Scenario 2 is similar to Scenario 1, but the mean model is non-additive. The mean model in Scenario 3 is piecewise constant, with the cut points for $\boldsymbol{x}_2$ depending on $\boldsymbol{x}_1$. Finally, Scenario 4 is a smooth mean model.

For each scenario, we generate 200 data sets and estimate $\boldsymbol{M}$ using CRISP (with $q = 100$) and several competitors: FLAM (implemented with the `R` package `flam` (Petersen, 2014)); CART (implemented with the `R` package `rpart` (Therneau et al., 2014)); TPS (implemented with the `R` package `fields` (Nychka et al., 2014)); a linear model with predictors $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, and their interaction; and an "oracle" linear model based on knowing *a priori* which regions of the mean model take on a constant value.

For each of the four scenarios, we plot mean squared prediction error[1] versus degrees of freedom (a notion that will be discussed extensively in Section 4.1). CRISP and FLAM are fit over a sequence of exponentially decreasing λ values, with the degrees of freedom estimated using (6) and a result from Petersen et al. (forthcoming), respectively. TPS is fit over a sequence of degrees of freedom. For CART, we vary the number of terminal nodes in the tree, and average the estimator (7) over the replicates in order to estimate the degrees of freedom for each number of terminal nodes. Note that the number of degrees of freedom of CART is non-monotonic for small numbers of terminal nodes (as seen in Figure 3).

3.2 Results for $n = 100$

Results are shown in Figure 3. We see that both CRISP and TPS perform reasonably well in terms of prediction error in all scenarios, regardless of the true mean model. FLAM outperforms the other methods in Scenario 1, which is unsurprising as the mean model is truly additive, and FLAM boils down to CRISP with an additivity constraint (Section 5.2). However, FLAM performs poorly for mean models with substantial non-additivity (Scenarios 2 and 4). Outside of Scenario 1, CART performs worse than TPS and CRISP. CRISP, TPS, and CART all perform better than a linear model with an interaction in Scenarios 1–3. However, in Scenario 4, the mean model is well-approximated using a linear model. We also fit MARS for all scenarios; however, performance was poor and the results are omitted.

While CRISP and TPS have comparable prediction error, their fits are quite different. In Figure 2, we show the estimated mean models for CRISP, TPS, and CART for a single replicate of data in each scenario. CRISP provides fits that reflect the true mean model well, even when the true mean model is smooth. While TPS has low prediction error, the smooth fits from TPS are not easily interpreted and are far from the true mean model in some scenarios. While the fits from CART reflect the mean model reasonably well in Scenarios 1 and 2, the fits from CART in all scenarios are highly variable. CART fits from different replicates of Scenario 4 are shown in Figure 4. The average variance of an element of $\boldsymbol{M}^*$ across the 200 replicates for Scenario 4 was 0.843 for CART, compared to 0.0935 for CRISP and 0.0653 for TPS. The variance of CART's fitted values is similarly inflated for the other scenarios. Small perturbations of the data can produce very different qualitative conclusions when examining CART's fits.

3.3 Results for $n = 10,000$

We compare CRISP to TPS and CART. Results are in Figures 2 and 5. Again, CRISP performs well in all scenarios, and the CART fits are much more variable than those of CRISP and TPS. The average variance of an element of $\boldsymbol{M}^*$ across the 200 replicates for Scenario 1 was 0.111 for CART, compared to 0.051 for CRISP and 0.083 for TPS. For Scenario 2, the average variance was 1.42 for CART, compared to 0.056 for CRISP and 0.083 for TPS. For Scenario 3, the average variance was 0.692 for CART, compared to 0.077 for CRISP and 0.129 for TPS. And finally, for Scenario 4, the average variance was 1.89 for

1. Mean squared prediction error is defined as $\frac{1}{q^2}\|\boldsymbol{M}-\boldsymbol{M}^*\|_F^2$, where $\boldsymbol{M} \in \mathbb{R}^{q\times q}$ is the true mean matrix and $\boldsymbol{M}^* \in \mathbb{R}^{q\times q}$ is the estimate from a given method. For methods other than CRISP, $\boldsymbol{M}^*$ was constructed using the mean model estimate at the midpoint of each bin of the $q \times q$ grid.

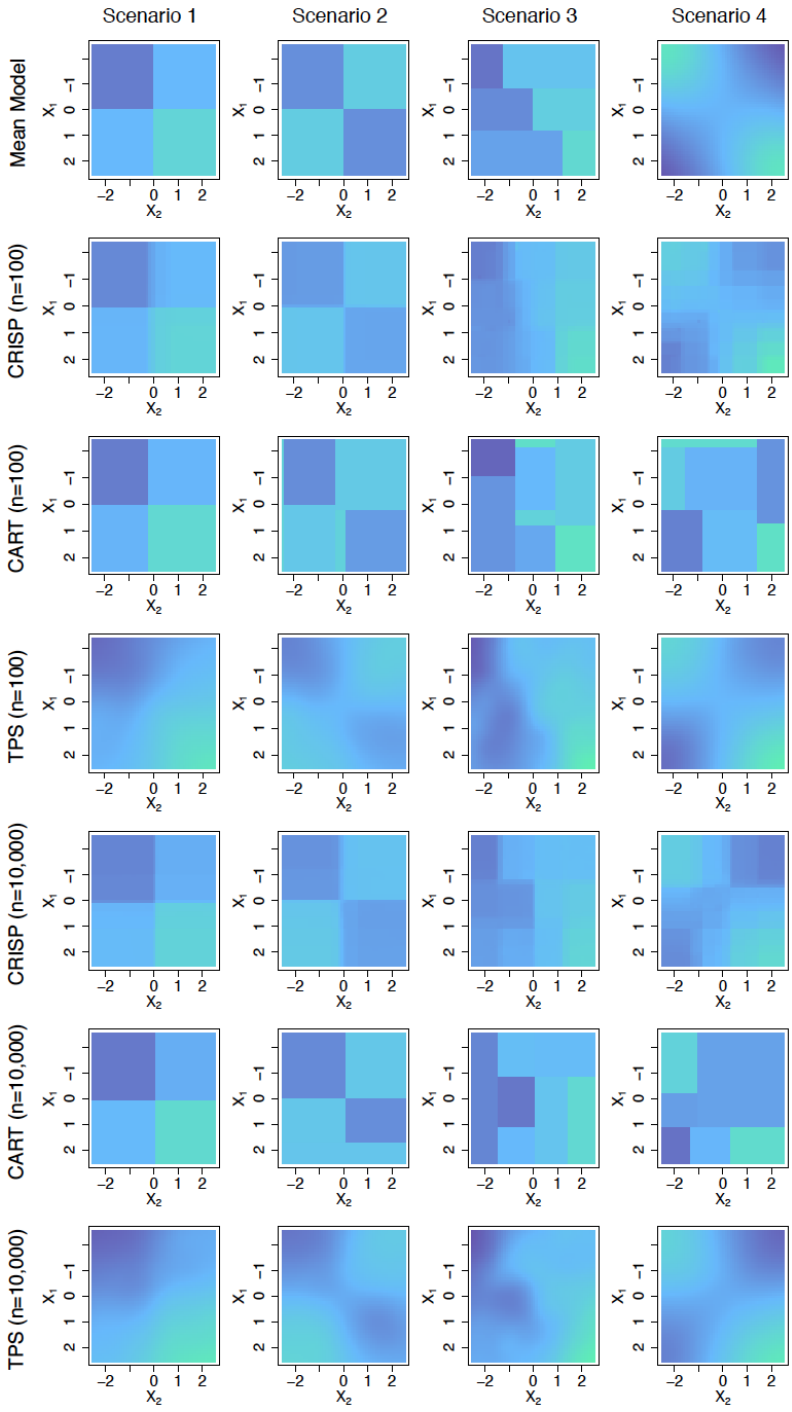

Figure 2: The mean models for Scenarios 1–4, as well as estimated mean models from CRISP, CART, and TPS for the simulations considered in Section 3. Each fit is from a single replicate of data, with the number of degrees of freedom indicated in Figures 3 and 5 for $n = 100$ and $n = 10,000$, respectively. The heat scale legend is in Figure 1(e).

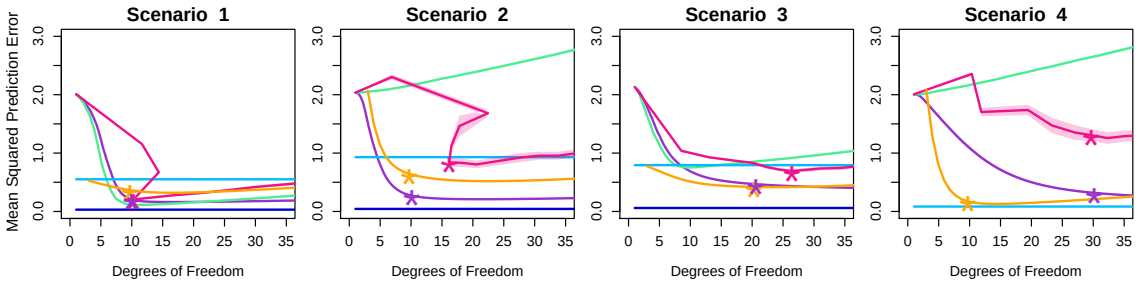

Figure 3: Mean squared prediction error, as a function of the degrees of freedom, for the four scenarios considered in the simulations of Section 3.2. The methods displayed are CRISP (▬), FLAM (▬), TPS (▬), CART (▬), linear model with an interaction (▬), and the oracle linear model (▬). The oracle linear model is only fit for Scenarios 1–3, for which the mean models have constant regions. Shaded bands (only visible for CART) indicate point-wise 95% confidence intervals over the 200 replicate data sets. The linear models have a fixed number of degrees of freedom, but are shown as horizontal lines. Asterisks indicate the degrees of freedom used for the fits shown in Figure 2.

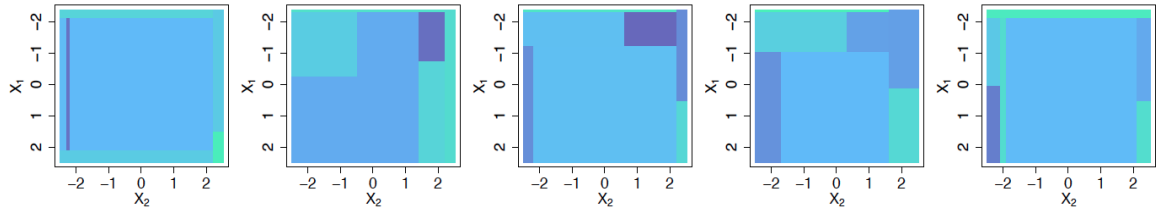

Figure 4: Fits for CART in Scenario 4 with $n = 100$ (as also shown in Figure 2) corresponding to five additional replicates of data. The heat scale legend is in Figure 1(e).

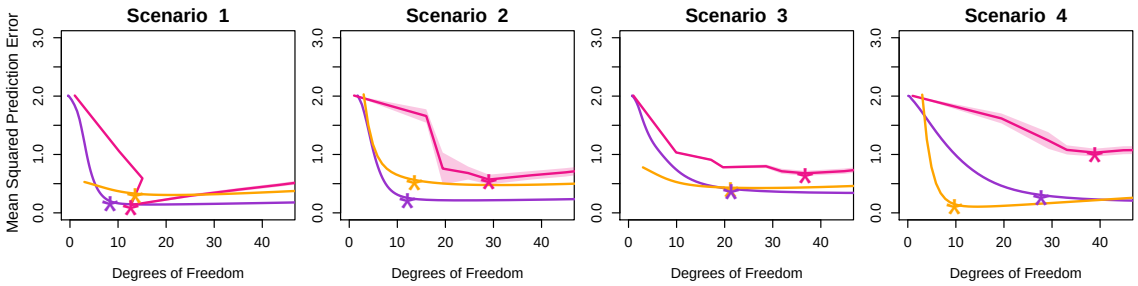

Figure 5: Results for $n = 10,000$ for CRISP (—), TPS (—), and CART (—) in the simulations of Section 3.3. Details are as given in Figure 3.

CART, compared to 0.096 for CRISP and 0.061 for TPS. Notably, a large sample size is not sufficient for producing stable CART fits, unless the signal-to-noise ratio is suitably large.

4. Properties of CRISP

In this section, we provide an unbiased estimator for CRISP's degrees of freedom. We also derive an analytical expression for the range of λ for which the solution to (4) takes a constant value, $\boldsymbol{m}^* = \left(\frac{1}{n}\mathbf{1}^T\boldsymbol{y}\right)\mathbf{1}$. Lastly, we discuss the role of q and λ in controlling the granularity of CRISP. Throughout this section, we use $\boldsymbol{A}^+$ to denote the Moore-Penrose pseudoinverse of a matrix $\boldsymbol{A}$.

4.1 Degrees of Freedom

Suppose that $\text{Var}(\boldsymbol{y}) = \sigma^2\boldsymbol{I}$, and let $g(\boldsymbol{y}) = \hat{\boldsymbol{y}}$ denote the fit corresponding to some model-fitting procedure g. Then the degrees of freedom of g is defined as $\frac{1}{\sigma^2}\sum_{i=1}^{n}\text{Cov}(y_i, \hat{y}_i)$ (Hastie and Tibshirani, 1990; Efron, 1986).

The concept of degrees of freedom provides a common framework for comparing the complexities of various models; this is particularly useful when the models under consideration are complex or unrelated. Ye (1998) proposed a computationally-burdensome Monte Carlo approach for estimating the degrees of freedom of a model-fitting procedure. In recent years, unbiased estimators for the degrees of freedom have been derived for the lasso and generalized lasso (Zou et al., 2007; Tibshirani and Taylor, 2012), among other methods. These estimators allow us to characterize a model's complexity, and also can be used in order to develop an approach for tuning parameter selection based on Akaike's information criterion (AIC; Akaike, 1973) or Bayesian information criterion (BIC; Schwarz, 1978).

Problem (3) is equivalent to the problem

$$\underset{\boldsymbol{m}}{\text{minimize}} \quad \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{Q}\boldsymbol{m}\|_2^2 + \lambda\sum_{i=1}^{q-1}[\|\boldsymbol{R}_i\boldsymbol{m}\|_2 + \|\boldsymbol{C}_i\boldsymbol{m}\|_2] + \frac{\gamma}{2}\|\boldsymbol{m}\|_2^2 \tag{5}$$

with $\gamma = 0$. In the rest of this section, we take γ to be a small positive constant, which ensures strong convexity and enforces uniqueness of the solution.

We now introduce some notation. First, we define $\mathcal{C}$, the set of difference matrices corresponding to equal neighboring rows or columns in the solution $\boldsymbol{m}^*$ to (5). That is, $\mathcal{C} = \{\boldsymbol{A}_i : \|\boldsymbol{A}_i\boldsymbol{m}^*\|_2 = 0\}$ where $\boldsymbol{A}_1 = \boldsymbol{R}_1, \boldsymbol{A}_2 = \boldsymbol{R}_2, \ldots, \boldsymbol{A}_{q-1} = \boldsymbol{R}_{q-1}, \boldsymbol{A}_q = \boldsymbol{C}_1, \boldsymbol{A}_{q+1} = \boldsymbol{C}_2, \ldots, \boldsymbol{A}_{2q-2} = \boldsymbol{C}_{q-1}$. Then we define $\boldsymbol{A}_*$ to be the submatrix of $\boldsymbol{A}$ obtained by retaining only the rows of $\boldsymbol{A}$ corresponding to matrices $\boldsymbol{A}_i \in \mathcal{C}$. Note that $\boldsymbol{A}_* \in \mathbb{R}^{q|\mathcal{C}|\times q^2}$. We propose to estimate the degrees of freedom of CRISP as

$$\hat{df}_{CRISP} = \text{Tr}\left[\boldsymbol{Q}\left(\boldsymbol{D} + \lambda\boldsymbol{P}\sum_{i:\boldsymbol{A}_i\notin\mathcal{C}} S_2(\boldsymbol{A}_i, \boldsymbol{m}^*)\boldsymbol{P} + \gamma\boldsymbol{I}\right)^{-1}\boldsymbol{P}\boldsymbol{Q}^T\right], \tag{6}$$

where $\boldsymbol{P} = \boldsymbol{I}_{q^2} - \boldsymbol{A}_*^+\boldsymbol{A}_*$, $S_2(\boldsymbol{A}_i, \boldsymbol{m}^*) = \frac{\boldsymbol{A}_i^T\boldsymbol{A}_i}{\|\boldsymbol{A}_i\boldsymbol{m}^*\|_2} - \frac{\boldsymbol{A}_i^T\boldsymbol{A}_i\boldsymbol{m}^*\boldsymbol{m}^{*T}\boldsymbol{A}_i^T\boldsymbol{A}_i}{\|\boldsymbol{A}_i\boldsymbol{m}^*\|_2^3}$, and $\boldsymbol{Q}$ was defined in (3). Recall that $\boldsymbol{M}^*$ will tend to contain row-column blocks of constant value, as shown in Figure 1(d). We define $\boldsymbol{D} = \text{diag}\left(h(m_1^*), \cdots, h(m_{q^2}^*)\right)$, where $h(m_i^*)$ is the ratio of the number of observations in the block of $\boldsymbol{M}^*$ that contains m_i^* to the number of elements of $\boldsymbol{M}^*$ in the block of $\boldsymbol{M}^*$ that contains m_i^*. We use the notation MVN to indicate a multivariate normal distribution.

Proposition 1 *Assume $\boldsymbol{y} \sim MVN(\boldsymbol{\mu}, \sigma^2\boldsymbol{I})$. Then $\hat{df}_{CRISP}$ is an unbiased estimator of the degrees of freedom of CRISP.*

The following corollary indicates that the estimator (6) simplifies substantially when the CRISP solution takes a particular form.

Corollary 2 *Assume $\boldsymbol{y} \sim MVN(\boldsymbol{\mu}, \sigma^2\boldsymbol{I})$. If either all rows or all columns of $\boldsymbol{M}^*$ are equal, then the total number of blocks of $\boldsymbol{M}^*$ is an unbiased estimator of the degrees of freedom.*

In 100 replicate data sets with $y_i \sim N(\mu_i, \sigma^2)$, we compare the mean of (6) to the mean of

$$\frac{1}{\sigma^2}\sum_{i=1}^{n}\left(\hat{y}_i - \mu_i\right)\left(y_i - \mu_i\right), \tag{7}$$

which provides a Monte Carlo estimate of $\frac{1}{\sigma^2}\sum_{i=1}^{n}\text{Cov}(y_i, \hat{y}_i)$, the true degrees of freedom of CRISP. The results in Figure 6(a) empirically validate Proposition 1, showing that (6) is an unbiased estimator of CRISP's degrees of freedom. Note that the proofs of Proposition 1 and Corollary 2 can be found in Appendices E and F, respectively.

4.2 Range of λ that Yields a Constant Solution

CRISP has a single tuning parameter λ, which we typically will select via cross-validation or a related approach. Here, we derive the minimum value of λ such that $\boldsymbol{m}^* = \left(\frac{1}{n}\boldsymbol{1}^T\boldsymbol{y}\right)\boldsymbol{1}$, corresponding to a fit in which all elements of $\boldsymbol{m}^*$ are equal.

Lemma 3 *The solution to (4) is constant (i.e., $\boldsymbol{m}^* = \left(\frac{1}{n}\boldsymbol{1}^T\boldsymbol{y}\right)\boldsymbol{1}$) if and only if*

$$\lambda \geq \max_{1\leq i\leq q-1}\left\{\|\boldsymbol{d}_{1i}^*\|_2, \|\boldsymbol{d}_{2i}^*\|_2\right\},$$

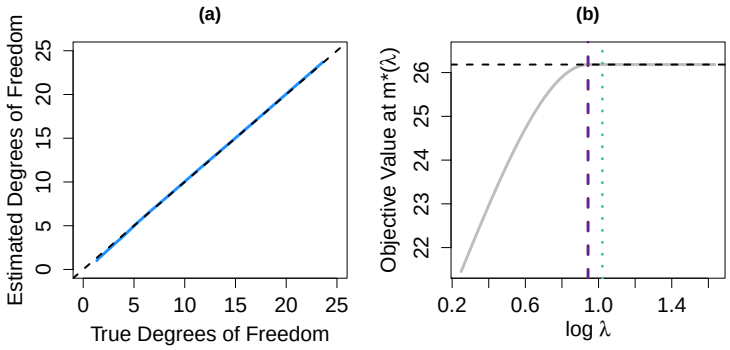

Figure 6: In *(a)*, we compare the degrees of freedom calculated using our estimator (7) (y-axis) from Section 4.1 to the unbiased, Monte Carlo estimator (6) (x-axis). Varying λ gives the solid line, and the dashed line indicates $y = x$. In *(b)*, we plot the value of the objective of (4) at $\boldsymbol{m}^*(\lambda)$, the minimizer of (4) at λ, for a replicate of data as λ varies. We compare two ways of finding a λ large enough such that $\boldsymbol{m}^*(\lambda) = \left(\frac{1}{n}\mathbf{1}^T\boldsymbol{y}\right)\mathbf{1}$, which results in the objective shown as - - -. We take $\lambda = \max_{1\le i\le q-1}\{\|\boldsymbol{d}_{1i}\|_2, \|\boldsymbol{d}_{2i}\|_2\}$ with either $\boldsymbol{d}$ being the solution to (8) (— —) or $\boldsymbol{d} = (\boldsymbol{A}^T)^+\boldsymbol{Q}^T\left(\boldsymbol{y} - \left(\frac{1}{n}\mathbf{1}^T\boldsymbol{y}\right)\mathbf{1}\right)$ (· · · ·). The former (— —) matches the result of Lemma 3 in Section 4.2.

where $\boldsymbol{d}^* = (\boldsymbol{d}_{11}^{*T} \cdots \boldsymbol{d}_{1(q-1)}^{*T}\ \boldsymbol{d}_{21}^{*T} \ldots \boldsymbol{d}_{2(q-1)}^{*T})^T$ *is the solution to*

$$\underset{\boldsymbol{d}}{\text{minimize}} \quad \max_{1\le i\le q-1}\{\|\boldsymbol{d}_{1i}\|_2, \|\boldsymbol{d}_{2i}\|_2\} \quad \textit{subject to} \quad \boldsymbol{Q}^T\left(\boldsymbol{y} - \left(\frac{1}{n}\mathbf{1}^T\boldsymbol{y}\right)\mathbf{1}\right) = \boldsymbol{A}^T\boldsymbol{d}. \tag{8}$$

Recall that the matrix $\boldsymbol{Q}$ was defined in (3). Taking $\lambda = \max_{1\le i\le q-1}\left\{\|\tilde{\boldsymbol{d}}_{1i}\|_2, \|\tilde{\boldsymbol{d}}_{2i}\|_2\right\}$ for any feasible vector $\tilde{\boldsymbol{d}}$ for (8) will give a value of λ sufficiently large so $\boldsymbol{m}^*$ is constant. For example, we can choose $\tilde{\boldsymbol{d}} = (\boldsymbol{A}^T)^+\boldsymbol{Q}^T\left(\boldsymbol{y} - \left(\frac{1}{n}\mathbf{1}^T\boldsymbol{y}\right)\mathbf{1}\right)$. However, choosing λ in accordance with Lemma 3 will give the minimum value of λ such that $\boldsymbol{m}^* = \left(\frac{1}{n}\mathbf{1}^T\boldsymbol{y}\right)\mathbf{1}$. The optimization problem (8) can be solved using a standard convex solver, such as SDPT3 via `CVX` in `MATLAB` (Grant and Boyd, 2008, 2014). An illustration of Lemma 3 is provided in Figure 6(b).

4.3 Controlling the Granularity of CRISP

Both q and λ control the granularity of the final CRISP model: q controls the size of the grid used to construct $\boldsymbol{M}$, and λ controls the number of blocks in the final fitted CRISP model. For a range of very small λ values, there will be q^2 blocks; for larger λ values, the CRISP solution will have a smaller number of blocks.

Given that q and λ both influence the number of blocks in the final fitted CRISP model, one might wonder whether it is necessary to have both q and λ. We illustrate the value of both q and λ through some simple examples.

4.3.1 Choice of q

In principle, q may be chosen to equal n. This means that each bin of the $q \times q$ grid would contain at most one observation. However, when n is large, choosing $q = n$ can lead to excessive computational time, memory burden, and variance in the fit. Instead, we aim to choose q to be large enough to allow for adequate granularity, but not excessively large. What constitutes adequate granularity will depend on the context of the problem.

In our analyses, we choose to treat q as a fixed parameter that is chosen prior to fitting CRISP. However, if desired, q could be chosen by K-fold cross validation.

4.3.2 Choice of λ

To illustrate the role of λ, consider taking $\lambda = 0$ in (3), and treating q as a tuning parameter rather than a fixed value. When $\lambda = 0$, (3) contains only a sum of squared errors term, so the estimate within each bin is the mean value of the observations in that bin. For bins without any observations, we estimate the corresponding element of $\boldsymbol{M}$ to be the overall mean of $\boldsymbol{y}$.

For the mean models shown in Figure 2, we compare CRISP to (3) with $\lambda = 0$ and q chosen adaptively. We focus on the general findings here, but detailed results are given in Appendix H. When the true mean model is piecewise constant with boundaries that are well-approximated by a grid of bins (as in Scenarios 1–3), CRISP and (3) with $\lambda = 0$ and variable q perform similarly. However, CRISP is clearly superior at estimating the smooth mean model of Scenario 4 (Figure 12), as it is able to borrow information across bins, instead of simply fitting the mean of observations within each bin. CRISP also allows the granularity of the fitted model to vary adaptively over the covariate space, as shown in Figure 13(a) of Appendix H. The blocks of this mean model perfectly align with a grid that has $q = 3$, but the mean model only has 4 blocks. While (3) with $\lambda = 0$ and $q = 3$ fits 9 blocks, CRISP correctly identifies 4 blocks (Figures 13(b) and 13(c) of Appendix H).

5. Connections to Other Methods

In this section, we establish connections between CRISP and two previous proposals.

5.1 Connection to One-Dimensional Fused Lasso

Suppose that for a given value of λ, the CRISP fit involves only one covariate: that is, $\boldsymbol{M}^* = \tilde{\boldsymbol{m}}\mathbf{1}_q^T$ or $\boldsymbol{M}^* = \mathbf{1}_q\tilde{\boldsymbol{m}}^T$ for some $\tilde{\boldsymbol{m}} \in \mathbb{R}^q$. We will now show that in this setting, the CRISP solution can be recovered by solving a one-dimensional fused lasso problem (Tibshirani et al., 2005).

Before presenting Lemma 4, we introduce some notation. Define $\boldsymbol{D} = [\boldsymbol{I}_{(q-1)\times(q-1)}\ \mathbf{0}_{(q-1)\times 1}] - [\mathbf{0}_{(q-1)\times 1}\ \boldsymbol{I}_{(q-1)\times(q-1)}]$ to be the first difference matrix. Define $\tilde{\boldsymbol{y}} \in \mathbb{R}^q$ such that $\tilde{y}_i$ is the mean outcome value of the observations in the ith row of the $q \times q$ grid used to construct $\boldsymbol{M}$. Let n_i denote the number of observations in the ith row of the $q \times q$ grid used to construct $\boldsymbol{M}$. Define $\boldsymbol{W} \in \mathbb{R}^{q\times q}$ to be the diagonal matrix with entries $\sqrt{n_1}, \sqrt{n_2}, \ldots, \sqrt{n_q}$.

Lemma 4 *Suppose that, for some value of λ, the CRISP solution is of the form $\boldsymbol{M}^* = \tilde{\boldsymbol{m}}\mathbf{1}_q^T$ for some $\tilde{\boldsymbol{m}} \in \mathbb{R}^q$. Then $\tilde{\boldsymbol{m}}$ is the solution to the problem*

$$\underset{\tilde{\boldsymbol{m}} \in \mathbb{R}^q}{\text{minimize}} \quad \frac{1}{2} \left\| \boldsymbol{W}(\tilde{\boldsymbol{y}} - \tilde{\boldsymbol{m}}) \right\|_2^2 + \lambda\sqrt{q} \left\| \boldsymbol{D}\tilde{\boldsymbol{m}} \right\|_1 . \tag{9}$$

If instead $\boldsymbol{M}^* = \mathbf{1}_q\tilde{\boldsymbol{m}}^T$, then a result similar to Lemma 4 holds, with modifications to the definitions of $\boldsymbol{W}$ and $\tilde{\boldsymbol{y}}$.

Equation 9 is a weighted fused lasso problem with response vector $\tilde{\boldsymbol{y}}$ and weights $\sqrt{n_1}, \sqrt{n_2}, \ldots, \sqrt{n_q}$. When $q = n$, (9) simplifies to a standard one-dimensional fused lasso problem.

Corollary 5 *If $q = n$ and $\boldsymbol{M}^* = \tilde{\boldsymbol{m}}\mathbf{1}_n^T$, then $\tilde{\boldsymbol{m}}$ is the solution to the one-dimensional fused lasso problem*

$$\underset{\tilde{\boldsymbol{m}} \in \mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2} \left\| \boldsymbol{P}\boldsymbol{y} - \tilde{\boldsymbol{m}} \right\|_2^2 + \lambda\sqrt{n} \left\| \boldsymbol{D}\tilde{\boldsymbol{m}} \right\|_1 , \tag{10}$$

where $\boldsymbol{P}$ is the permutation matrix that orders the elements of $\boldsymbol{x}_1$ from least to greatest.

If instead $\boldsymbol{M}^* = \mathbf{1}_n\tilde{\boldsymbol{m}}^T$, then Corollary 5 holds with $\boldsymbol{P}$ defined to be the permutation matrix that orders the elements of $\boldsymbol{x}_2$ from least to greatest.

5.2 Connection to Fused Lasso Additive Model

In this subsection, we will establish that CRISP is a generalization of the fused lasso additive model (FLAM) proposal of Petersen et al. (forthcoming). FLAM fits an additive model in which each covariate's fit is estimated to be piecewise constant with adaptively-chosen knots.

For simplicity, assume that $q = n$. Consider a modification of CRISP in which we impose additivity on the mean matrix $\boldsymbol{M}$. That is, we assume $f(x_1, x_2) = \theta_0 + f_1(x_1) + f_2(x_2)$, where θ_0 is an overall mean, and f_1 and f_2 are mean-zero over the training observations. We introduce the n-vectors $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$, where $f_1(x_{i1}) = \theta_{1i}$ and $f_2(x_{i2}) = \theta_{2i}$ for all $i = 1, \ldots, n$. Thus the additivity constraint for the (i, j) element of $\boldsymbol{M}$, $M_{(i)(j)}$, can be expressed as

$$M_{(i)(j)} = \theta_0 + \theta_{1i} + \theta_{2j} \text{ for } i = 1, \ldots, n;\ j = 1, \ldots, n \quad \text{with} \quad \mathbf{1}^T\boldsymbol{\theta}_1 = \mathbf{1}^T\boldsymbol{\theta}_2 = 0. \tag{11}$$

Lemma 6 *CRISP (1)–(2) with $q = n$ and with the additional additivity constraint (11) is equivalent to FLAM with $p = 2$, which is the solution to the optimization problem*

$$\begin{aligned} &\underset{\theta_0 \in \mathbb{R}, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in \mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2} \left\| \boldsymbol{y} - (\theta_0\mathbf{1} + \boldsymbol{\theta_1} + \boldsymbol{\theta_2}) \right\|_2^2 + \lambda \left(\left\| \boldsymbol{D}\boldsymbol{P}_1\boldsymbol{\theta}_1 \right\|_1 + \left\| \boldsymbol{D}\boldsymbol{P}_2\boldsymbol{\theta}_2 \right\|_1 \right) \\ &\textit{subject to } \mathbf{1}^T\boldsymbol{\theta}_1 = \mathbf{1}^T\boldsymbol{\theta}_2 = \mathbf{0}, \end{aligned} \tag{12}$$

where $\lambda \geq 0$ is a tuning parameter, $\boldsymbol{P}_j$ is the permutation matrix that orders the elements of $\boldsymbol{x}_j$ from least to greatest, and $\boldsymbol{D} = [\boldsymbol{I}_{(n-1)\times(n-1)}\ \mathbf{0}_{(n-1)\times 1}] - [\mathbf{0}_{(n-1)\times 1}\ \boldsymbol{I}_{(n-1)\times(n-1)}]$ is the first difference matrix.

The proof of Lemma 6 follows from algebraic manipulation.

CRISP (1)–(2) with the additivity constraint (11) is also equivalent to FLAM when the ℓ_2 norms in the penalty (2) are changed to ℓ_1 or ℓ_∞ norms. These alternative penalties are discussed further in Appendix B.

Lemma 6 can be generalized in order to establish that CRISP with $q < n$ is equivalent to a version of FLAM that re-weights the loss function in (12) appropriately.

6. Data Application

We consider predicting median house value on the basis of median income and average occupancy, measured for 20,640 neighborhoods in California. The data set was originally considered in Pace and Barry (1997) and is publicly available from the Carnegie Mellon StatLib data repository (`lib.stat.cmu.edu`).

For this analysis, we focus on predicting median house value for the central area of the covariate space. In particular, we filter the neighborhoods to select those with median incomes and average occupancies that both fall within the central 95% of the covariate distribution, which results in 18,662 neighborhoods to be analyzed. Further details are provided in Appendix I. To illustrate the impact that the size of the data set may have on the preferred analysis approach, we consider five different training set sizes: 100, 500, 1000, 5000, and 11,198 (which corresponds to 60% of the observations). We use the observations not selected for the training set as the test set. For each training set size, we consider 10 different data samples. We compare the performance of CRISP (with $q = 100$) to CART and TPS.

Figure 7 shows that income is positively associated with house value. Occupancy is not strongly associated with house value in low-income neighborhoods. However, among neighborhoods with median incomes exceeding around \$50,000, neighborhoods with mostly single or double occupancy tend to have more expensive homes than those with higher occupancies and the same income. This is perhaps because single people and couples without children have more disposable income to spend on housing than families at the same income level.

In Figure 7, we show estimated mean models from CRISP for two different values of λ. The larger value of λ has slightly worse prediction performance, but has a simple block structure reminiscent of CART. The smaller value of λ gives better prediction performance with a more complex fit structure that resembles the fits from TPS. This illustrates how CRISP's tuning parameter, λ, balances the trade-off between interpretability and prediction performance.

While the fit from CART in Figure 7 is quite interpretable, CART gives highly-variable fits across different splits of the data. This is illustrated in Figure 8. The average variance of predictions from CART across the 10 splits of data is more than three times that of CRISP and TPS. For larger training sets, the variance decreases, though the variability of the CART predictions remains much larger than that of CRISP and TPS. In Figure 8, we also see that CART's performance in terms of test set mean squared error (MSE) is worse than CRISP and TPS, but becomes increasingly similar with larger sample sizes. For example, in Figure 9, we show the results for the largest training set sample considered ($n = 11,198$). We see that all three methods perform very similarly in terms of test set MSE, and provide qualitatively similar estimated mean models. As the available sample size increases, the differences between CRISP, TPS, and CART in terms of prediction performance and interpretability of fits become less pronounced.

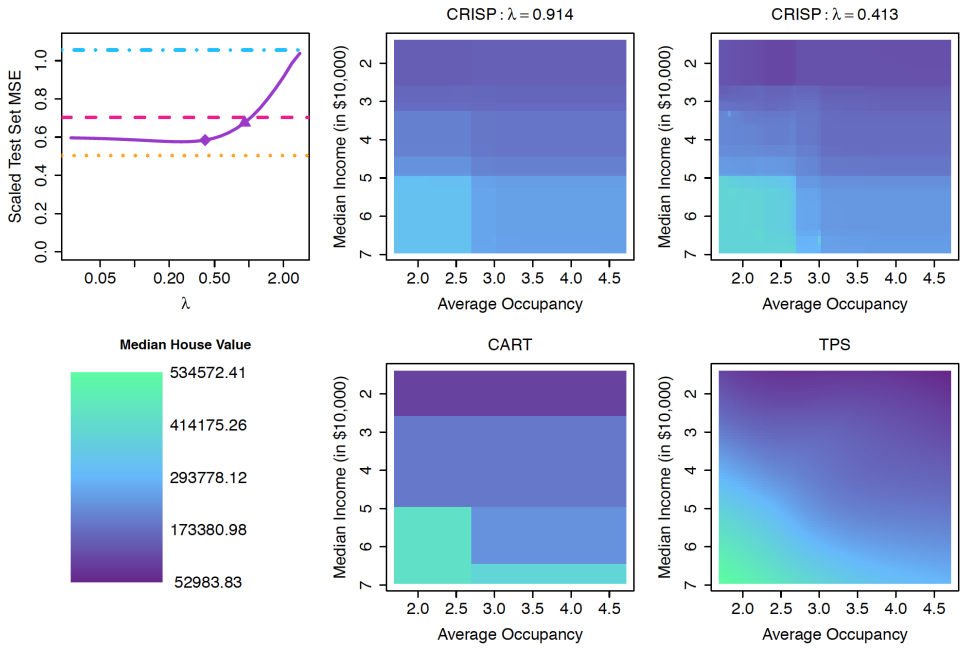

Figure 7: We consider predicting median house value on the basis of median income and average occupancy using a training set of size $n = 100$, as considered in Section 6. We plot the average value for 10 data samples of test set MSE divided by the variance of the training set outcome. We plot this scaled test set MSE versus λ for CRISP (▬), and show the minimum scaled test set MSE achieved by CART (- - -), TPS (- - -), and an intercept-only model (- · -). Estimated mean models for CRISP are shown for a larger value of λ (indicated by ▲) and a smaller value (indicated by ◆). The estimated mean models shown for CART and TPS correspond to the tuning parameter with the minimum test set MSE. The heat scale legend for the median house value is shown.

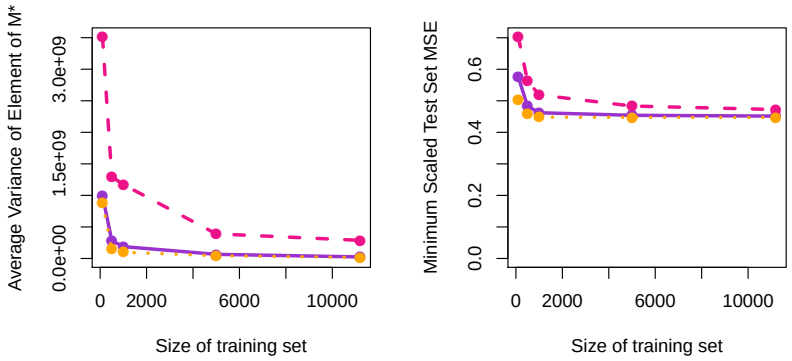

Figure 8: We plot the average variance of predictions and the minimum scaled test set MSE (as defined in Figure 7) as a function of training set sample size for CRISP (—), CART (– – –), and TPS (- - -) applied to the housing data considered in Section 6.

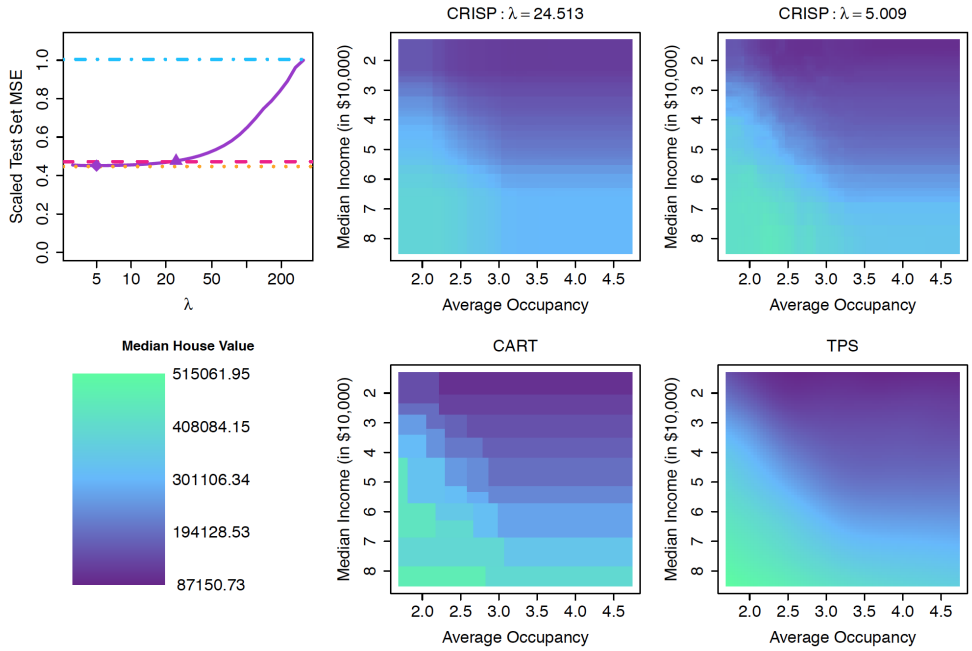

Figure 9: Results using median income and average occupancy as predictors of median house value using a training set of size $n = 11,198$, as considered in Section 6. Details are as in Figure 7.

7. Extension to $p > 2$

We have assumed thus far that $p = 2$. In this case, the estimated mean model for the entire covariate space can be summarized in a single plot, as in Figure 2.

We extend CRISP to the setting of $p > 2$ by constructing an *additive model of bivariate fits*. That is, we estimate the fit for each of the $\frac{p(p-1)}{2}$ *pairs* of features, giving a bivariate fit for each pair of covariates like those obtained in the setting of $p = 2$ and shown in Figure 2. We assume that the mean model is additive in these fits. We restrict the model to pairwise interactions between covariates for a couple of reasons. First, only considering pairwise interactions increases interpretability and reduces model complexity. Our model fit with pairwise interactions can be summarized using $\frac{p(p-1)}{2}$ plots, like those shown in Figure 2. There is no analogous way to easily summarize the model if we were to include higher-order interactions. Second, considering higher-order interactions would cause our model to suffer from the *curse of dimensionality*. That is, as the number of covariates increases, the data in any region of the p-dimensional space will become sparser and sparser: there would be an insufficient density of data throughout the covariate space to reasonably estimate a mean model with higher-order interactions.

We now present the details of our proposal for CRISP with $p > 2$. We consider interactions between each pair of features, $\{(j, j') : 1 \leq j < j' \leq p\}$. For ease of notation, we refer to the elements of this set using the index $k \in (1, \ldots, K)$ where $K = \frac{p(p-1)}{2}$. Recall that for $p = 2$, the mean model for CRISP is $\mathrm{E}[\boldsymbol{y} \mid \boldsymbol{x}_1, \boldsymbol{x}_2] = \boldsymbol{Q}\boldsymbol{m}$, where $\boldsymbol{m} \in \mathbb{R}^{q^2}$ is the vectorized mean matrix and $\boldsymbol{Q}$ selects the elements of $\boldsymbol{m}$ corresponding to the covariate bins of the elements of $\boldsymbol{y}$. Recall that $\boldsymbol{Q}$ is a function of $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, though we suppress this to simplify the notation. For $p > 2$, we consider the mean model

$$\mathrm{E}[\boldsymbol{y} \mid \boldsymbol{x}_1, \ldots, \boldsymbol{x}_p] = m_0\mathbf{1} + \sum_{k=1}^{K} \boldsymbol{Q}_k\boldsymbol{m}_k,$$

where $m_0 \in \mathbb{R}$ is an intercept, $\boldsymbol{m}_k \in \mathbb{R}^{q^2}$ is the vectorized mean matrix for the pair of features indexed by k, and $\boldsymbol{Q}_k \in \mathbb{R}^{n \times q^2}$ selects the elements of $\boldsymbol{m}_k$ corresponding to the covariate bins for the pair of covariates indexed by k. We include the intercept $m_0 \in \mathbb{R}$ in our model, and assume that $\boldsymbol{m}_1, \ldots, \boldsymbol{m}_K$ are mean-zero, to ensure identifiability.

When $p > 2$, we extend the CRISP optimization problem (4) as follows:

$$\begin{aligned} \underset{m_0, \boldsymbol{m}_k, \boldsymbol{z}_k : k=1,\ldots,K}{\text{minimize}} \quad & \frac{1}{2}\left\| \boldsymbol{y} - \left(m_0\mathbf{1} + \sum_{k=1}^{K} \boldsymbol{Q}_k\boldsymbol{m}_k \right) \right\|_2^2 + \lambda \sum_{k=1}^{K} \sum_{i=1}^{q-1} \left[\|\boldsymbol{z}_{k,1i}\|_2 + \|\boldsymbol{z}_{k,2i}\|_2 \right] \\ \text{subject to } & \boldsymbol{A}\boldsymbol{m}_k = \boldsymbol{z}_k, \mathbf{1}^T\boldsymbol{m}_k = 0, \end{aligned} \tag{13}$$

where $\boldsymbol{A}$ is as defined in Section 2.2. Thus $\hat{\boldsymbol{y}} = m_0^*\mathbf{1} + \sum_{k=1}^{K} \boldsymbol{Q}_k\boldsymbol{m}_k^*$, where $(m_0^*, \boldsymbol{m}_1^*, \ldots, \boldsymbol{m}_K^*)$ is the solution to (13).

Problem (13) can be solved using block coordinate descent (Tseng, 2001), which gives Algorithm 2. We iterate through the pairs of covariates, and perform a partial minimization (using Algorithm 1) for each $\boldsymbol{m}_k$, while keeping the others fixed. Using an argument similar to that in Section 2.3, the computational complexity of Algorithm 2 is $\mathcal{O}(K(n + q^4))$ for

an initial step and $\mathcal{O}(q^3)$ for each iteration of Step 2(b) of Algorithm 2. In practice, the number of iterations needed to achieve convergence in Step 2(b) of Algorithm 2 is relatively small.

We present a block coordinate descent algorithm, since it is a natural extension of Algorithm 1 to the $p > 2$ setting. However, CRISP with $p \gg 2$ can alternatively be fit using generalized gradient descent, which allows the updates for each bivariate fit to be run in parallel on a cluster.

Algorithm 2 — Block Coordinate Descent for CRISP with $p > 2$ (Equation (13))

1. Initialize $m_0^* = 0$ and $\boldsymbol{m}_k^* = \mathbf{0}$ for all $k = 1, \ldots, K$.
2. For $k = 1, \ldots, K, 1, \ldots, K, \ldots$, until convergence of the objective of (13):
   (a) Compute the residual $\boldsymbol{r}_k = \boldsymbol{y} - \left(m_0^* \mathbf{1} + \sum_{k' \neq k} \boldsymbol{Q}_{k'} \boldsymbol{m}_{k'}^*\right)$.
   (b) Using Algorithm 1, solve
   $$\underset{\boldsymbol{m}_k, \boldsymbol{z}_k}{\text{minimize}} \quad \frac{1}{2} \|\boldsymbol{r}_k - \boldsymbol{Q}_k \boldsymbol{m}_k\|_2^2 + \lambda \sum_{i=1}^{q-1} \left[\|\boldsymbol{z}_{k,1i}\|_2 + \|\boldsymbol{z}_{k,2i}\|_2\right] \quad \text{subject to } \boldsymbol{A}\boldsymbol{m}_k = \boldsymbol{z}_k.$$
   Let $\boldsymbol{m}_k^*$ denote the solution.
   (c) Compute the intercept, $m_0^* \leftarrow m_0^* + \text{mean}(\boldsymbol{m}_k^*)$, and center, $\boldsymbol{m}_k^* \leftarrow \boldsymbol{m}_k^* - \text{mean}(\boldsymbol{m}_k^*)$.

8. Discussion

We have presented CRISP, a method for fitting interpretable, flexible, and non-additive predictive models. CRISP fits have an easily-interpreted block structure, which is somewhat reminiscent of the fits from CART. But the fits from CRISP result from a non-greedy procedure, and are much less variable than those of CART. In our numerical studies, the prediction performance of CRISP is similar to TPS, and in many cases CRISP provides a simpler and more interpretable fit.

Future work could consider an alternative penalization scheme. Recall that CRISP first divides the covariate space into a $q \times q$ grid of bins. Our proposal only uses the information about the bin into which each of the n observations falls, which is used to construct $\boldsymbol{Q}$ in (4). Thus CRISP only makes use of the rankings of the observations for each covariate, rather than the actual values of the covariates. A modification to (4) could allow us to more heavily penalize the differences between pairs of neighboring rows or columns corresponding to observations with similar values in a given covariate. This modification is not very important when the covariate pairs are distributed uniformly over the covariate space, as in our simulation study in Section 3.

In this paper, we have only considered the setting of $p \ll n$. An extension of CRISP to larger p is left to future work.

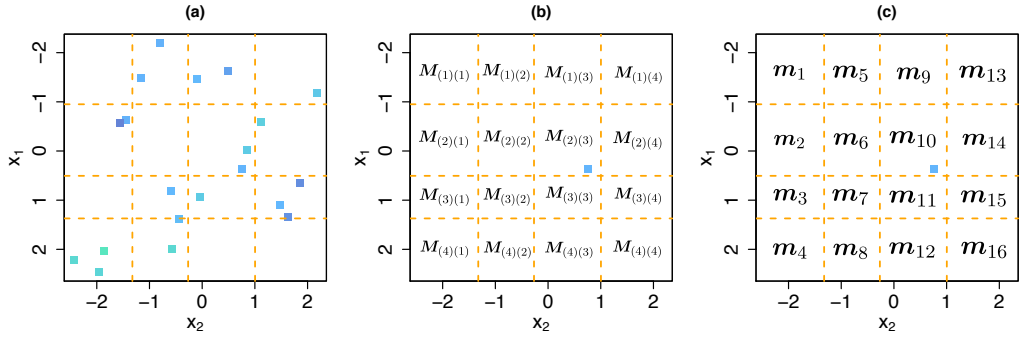

Figure 10: In *(a)*, each of the 20 squares represents an observation (x_1, x_2, y). There are $q^2 = 16$ bins of (x_1, x_2) values, whose boundaries coincide with the quartiles (- - -) of $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. In *(b)* and *(c)*, we label the elements of $\boldsymbol{M}$ and $\boldsymbol{m}$, respectively, corresponding to each bin of (x_1, x_2) values. Additionally, in *(b)* and *(c)*, we show $(x_{1i}, x_{2i}) = (0.4, 0.8)$, which is used in Appendix A to describe the construction of $\boldsymbol{Q}$.

Acknowledgments

We thank the associate editor and three referees for helpful comments. D.W. was supported by NIH Grant DP5OD009145, NSF CAREER Award DMS-1252624, and an Alfred P. Sloan Foundation Research Fellowship. N.S. was supported by NIH Grant DP5OD019820.

Appendix A. Notational Details

We first give an intuitive explanation of our vectorization scheme. Recall that each row of $\boldsymbol{Q} \in \mathbb{R}^{n \times q^2}$ contains $q^2 - 1$ elements that equal 0, and a single 1 that extracts an element of $\boldsymbol{m}$ according to the covariate values for that observation. For example, consider the ith row of $\boldsymbol{Q}$ for $(x_{1i}, x_{2i}) = (0.4, 0.8)$ in Figure 10(a). These covariate values fall within the 2nd row and 3rd column of the 4×4 grid, meaning that $\boldsymbol{M}_{(2)(3)}$ provides an estimate for y_i. After vectorizing $\boldsymbol{M}$, $\boldsymbol{M}_{(2)(3)}$ is $\boldsymbol{m}_{10}$, the 10th element of the mean vector. Note that we can convert between the matrix and vector notation by taking the column number minus one multiplied by q and adding the row number (e.g., $(3-1) \times 4 + 2$). The correspondence between $\boldsymbol{M}$ and $\boldsymbol{m}$ is illustrated in Figures 10(b) and 10(c). Thus the ith row of $\boldsymbol{Q}$ would contains all zeros, except a single 1 for the 10th element. Finally, $(\boldsymbol{Qm})_i = m_{10}$.

Before formally defining the function Ω and matrices $\boldsymbol{Q}$, $\boldsymbol{R}_i$ for $i = 1, \ldots, q-1$, and $\boldsymbol{C}_i$ for $i = 1, \ldots, q-1$ introduced in Section 2.2, we define a quantile function. We use quantile$(\cdot)$ to denote the quantile range into which an element falls: quantile$(x_{1i}) = k$ if x_{1i} is between the $\frac{k-1}{q}$- and $\frac{k}{q}$-quantiles of $\boldsymbol{x}_1$. For example, if $n = q = 4$ and $\boldsymbol{x}_1 = (9\ 3\ 5\ 2)^T$, then quantile$(x_{11}) = 4$. Similarly, if $n = 6$, $q = 3$, and $\boldsymbol{x}_1 = (7\ 2\ 3\ 8\ 1\ 5)^T$, then quantile$(x_{16}) = 2$.

We define the function Ω as $\Omega(\boldsymbol{M}, x_{1i}, x_{2i}) = M_{(a)(b)}$ where $a = \text{quantile}(x_{1i})$ and $b = \text{quantile}(x_{2i})$.

We construct $\boldsymbol{Q} \in \mathbb{R}^{n \times q^2}$ such that

$$[\boldsymbol{Q}]_{jk} = \begin{cases} 1 & \text{if } k = \text{quantile}(x_{1j}) + q \times (\text{quantile}(x_{2j}) - 1) \\ 0 & \text{otherwise} \end{cases},$$

$\boldsymbol{R}_i \in \mathbb{R}^{q \times q^2}$ for $i = 1, \ldots, q-1$ such that

$$[\boldsymbol{R}_i]_{jk} = \begin{cases} 1 & \text{if } k = i + q \times (j-1) \\ -1 & \text{if } k = i + 1 + q \times (j-1) \\ 0 & \text{otherwise} \end{cases},$$

and $\boldsymbol{C}_i \in \mathbb{R}^{q \times q^2}$ for $i = 1, \ldots, q-1$ such that

$$[\boldsymbol{C}_i]_{jk} = \begin{cases} 1 & \text{if } k = j + q \times (i-1) \\ -1 & \text{if } k = j + q \times i \\ 0 & \text{otherwise} \end{cases}.$$

Appendix B. Alternative Penalties

A more general formulation of our proposal in (1) is

$$\underset{\boldsymbol{M} \in \mathbb{R}^{q \times q}}{\text{minimize}} \quad \frac{1}{2} \sum_{i=1}^{n} \left(y_i - \Omega(\boldsymbol{M}, x_{1i}, x_{2i})\right)^2 + \lambda \sum_{i=1}^{q-1} \left[\left\| \boldsymbol{M}_{i\cdot} - \boldsymbol{M}_{(i+1)\cdot} \right\|_t + \left\| \boldsymbol{M}_{\cdot i} - \boldsymbol{M}_{\cdot(i+1)} \right\|_t \right], \quad (14)$$

which is equivalent to (1) for $t = 2$. One might consider solving (14) for $t = \infty$, which (like $t = 2$) encourages pairs of neighboring rows or columns of $\boldsymbol{M}$ to be identical. We compare the fit for $t = 2$ to that for $t = \infty$ in Figure 11(a)–(b). While $t = \infty$ gives desirable fits similar to $t = 2$, the computational time required is much higher than that for $t = 2$. This is because when adapted to $t = \infty$, Step 2(b) of Algorithm 1 no longer has a closed-form solution (Duchi and Singer, 2009).

We also consider the use of $t = 1$ in (14); this encourages each element of $\boldsymbol{M}$ to equal its four adjacent elements. However, using $t = 1$ gives very poor results: the bins of $\boldsymbol{M}$ containing observations are estimated to be shrunken versions of their observed values, while the bins of $\boldsymbol{M}$ without observations are estimated to be a common value (Figure 11(c)). In a sense, the penalization for $t = 1$ is too local given the data sparsity (e.g., only q of q^2 elements observed when $q = n$).

The results for $t = 1$ improve if an additional penalty is added to the objective function. First, note that (14) can also be written as

$$\underset{\boldsymbol{M} \in \mathbb{R}^{q \times q}}{\text{minimize}} \quad \frac{1}{2} \sum_{i=1}^{n} \left(y_i - \Omega(\boldsymbol{M}, x_{1i}, x_{2i})\right)^2 + \lambda \left(\|\boldsymbol{M}^T \boldsymbol{D}^T\|_{t,1} + \|\boldsymbol{M} \boldsymbol{D}^T\|_{t,1} \right), \quad (15)$$

where $\boldsymbol{D} = [\boldsymbol{I}_{(q-1)\times(q-1)}\ \boldsymbol{0}_{(q-1)\times 1}] - [\boldsymbol{0}_{(q-1)\times 1}\ \boldsymbol{I}_{(q-1)\times(q-1)}]$. Motivated by a proposal from van de Geer (2000), we add an additional penalty to (15) with $t = 1$,

$$\underset{\boldsymbol{M} \in \mathbb{R}^{q \times q}}{\text{minimize}} \quad \frac{1}{2} \sum_{i=1}^{n} \left(y_i - \Omega(\boldsymbol{M}, x_{1i}, x_{2i})\right)^2 + \lambda \left(\|\boldsymbol{M}^T \boldsymbol{D}^T\|_{1,1} + \|\boldsymbol{M} \boldsymbol{D}^T\|_{1,1} + \|\boldsymbol{D} \boldsymbol{M} \boldsymbol{D}^T\|_{1,1} \right). \quad (16)$$

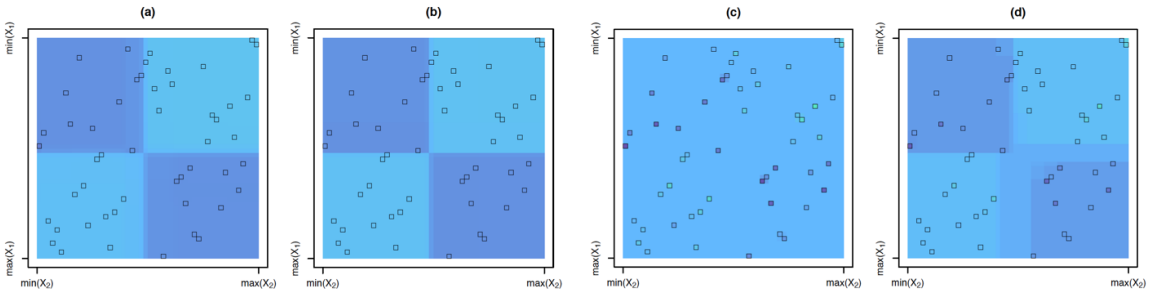

Figure 11: The estimated mean model from solving (14) for *(a)* $t = 2$ (CRISP), *(b)* $t = \infty$, and *(c)* $t = 1$, as well as *(d)* the estimated mean model from solving (16). The methods are described in detail in Appendix B. Note that $q = n$ was used for all methods. Data was generated for $n = 50$ from Scenario 2 (described in Section 3). The locations of the 50 observations are outlined in each plot. The heat scale legend is in Figure 1(e).

The penalty $\|\boldsymbol{DMD}^T\|_{1,1}$ encourages $|M_{(i)(j)} + M_{(i-1)(j-1)} - M_{(i-1)j} - M_{i(j-1)}|$ to equal zero, which results in a block structure as shown in Figure 11(d). While (16) outperforms (14) with $t = 1$, CRISP with $t = 2$ yields better results.

Appendix C. Details of Algorithm 1

C.1 Derivation of Algorithm 1

The scaled augmented Lagrangian of (4) is

$$L_\rho(\boldsymbol{m}, \boldsymbol{z}, \boldsymbol{u}) = \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{Q}\boldsymbol{m}\|_2^2 + \lambda \sum_{i=1}^{q-1} [\|\boldsymbol{z}_{1i}\|_2 + \|\boldsymbol{z}_{2i}\|_2] + \frac{\rho}{2} \sum_{i=1}^{q-1} \left[\|\boldsymbol{R}_i\boldsymbol{m} - \boldsymbol{z}_{1i} + \boldsymbol{u}_{1i}\|_2^2 + \|\boldsymbol{C}_i\boldsymbol{m} - \boldsymbol{z}_{2i} + \boldsymbol{u}_{2i}\|_2^2\right] \quad (17)$$

where $\boldsymbol{u} = \left((\boldsymbol{u}_{11})^T \dots (\boldsymbol{u}_{1(q-1)})^T (\boldsymbol{u}_{21})^T \dots (\boldsymbol{u}_{2(q-1)})^T\right)^T$ is the scaled dual variable. Solving (4) using ADMM relies on initializing estimates $\boldsymbol{m}^{(0)} := \boldsymbol{0}, \boldsymbol{z}^{(0)} := \boldsymbol{0}$, and $\boldsymbol{u}^{(0)} := \boldsymbol{0}$ and then iterating over three steps until convergence. At iteration k, the updates are

Step 1. $\boldsymbol{m}^{(k)} := \underset{\boldsymbol{m}}{\text{argmin}} \quad L_\rho\left(\boldsymbol{m}^{(k-1)}, \boldsymbol{z}^{(k-1)}, \boldsymbol{u}^{(k-1)}\right)$

Step 2. $\boldsymbol{z}^{(k)} := \underset{\boldsymbol{z}}{\text{argmin}} \quad L_\rho\left(\boldsymbol{m}^{(k)}, \boldsymbol{z}^{(k-1)}, \boldsymbol{u}^{(k-1)}\right)$

Step 3. $\boldsymbol{u}_{1i}^{(k)} := \boldsymbol{u}_{1i}^{(k-1)} + \boldsymbol{R}_i\boldsymbol{m}^{(k)} - \boldsymbol{z}_{1i}^{(k)}$ for $i = 1, \dots, q-1$
$\boldsymbol{u}_{2i}^{(k)} := \boldsymbol{u}_{2i}^{(k-1)} + \boldsymbol{C}_i\boldsymbol{m}^{(k)} - \boldsymbol{z}_{2i}^{(k)}$ for $i = 1, \dots, q-1$

Note that Step 3 can equivalently be written as $\boldsymbol{u}^{(k)} := \boldsymbol{u}^{(k-1)} + \boldsymbol{A}\boldsymbol{m}^{(k)} - \boldsymbol{z}^{(k)}$. We provide details regarding Steps 1 and 2 below.

Details of Step 1

The optimality condition of (17) for $\boldsymbol{m}$ is

$$\frac{\partial L_\rho}{\partial \boldsymbol{m}} = -\boldsymbol{Q}^T(\boldsymbol{y} - \boldsymbol{Q}\boldsymbol{m}) + \rho \sum_{i=1}^{q-1} \left[\boldsymbol{R}_i^T(\boldsymbol{R}_i\boldsymbol{m} - \boldsymbol{z}_{1i} + \boldsymbol{u}_{1i}) + \boldsymbol{C}_i^T(\boldsymbol{C}_i\boldsymbol{m} - \boldsymbol{z}_{2i} + \boldsymbol{u}_{2i})\right] = \boldsymbol{0}$$

or equivalently, $-\boldsymbol{Q}^T(\boldsymbol{y} - \boldsymbol{Q}\boldsymbol{m}) + \rho\boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{m} + \boldsymbol{u} - \boldsymbol{z}) = \boldsymbol{0}$. Therefore the update for Step 1 is $\boldsymbol{m}^{(k)} := \left[\boldsymbol{Q}^T\boldsymbol{Q} + \rho\boldsymbol{A}^T\boldsymbol{A}\right]^{-1}\left[\boldsymbol{Q}^T\boldsymbol{y} + \rho\boldsymbol{A}^T(\boldsymbol{z}^{(k-1)} - \boldsymbol{u}^{(k-1)})\right]$.

Details of Step 2

The *proximal operator* $\mathbf{prox}_{\lambda f}$ of λf is defined by $\mathbf{prox}_{\lambda f}(\boldsymbol{v}) = \underset{\boldsymbol{x}}{\operatorname{argmin}}\left(f(\boldsymbol{x}) + \frac{1}{2\lambda}\|\boldsymbol{x} - \boldsymbol{v}\|_2^2\right)$. The minimization for Step 2 is separable in the $\boldsymbol{z}_{1i}$ and $\boldsymbol{z}_{2i}$ for $i = 1, \ldots, q-1$. The minimization for $\boldsymbol{z}_{1i}$ is

$$\begin{aligned}
\boldsymbol{z}_{1i}^{(k)} &:= \underset{\boldsymbol{z}_{1i}}{\operatorname{argmin}} \left[\lambda \|\boldsymbol{z}_{1i}\|_2 + \frac{\rho}{2}\left\|\boldsymbol{R}_i\boldsymbol{m}^{(k)} - \boldsymbol{z}_{1i} + \boldsymbol{u}_{1i}^{(k-1)}\right\|_2^2\right] \\
&= \mathbf{prox}_{\frac{\lambda}{\rho}\|\cdot\|_2}\left(\boldsymbol{R}_i\boldsymbol{m}^{(k)} + \boldsymbol{u}_{1i}^{(k-1)}\right) \\
&= \left(\boldsymbol{R}_i\boldsymbol{m}^{(k)} + \boldsymbol{u}_{1i}^{(k-1)}\right)\left(1 - \frac{\lambda}{\rho\left\|\boldsymbol{R}_i\boldsymbol{m}^{(k)} + \boldsymbol{u}_{1i}^{(k-1)}\right\|_2}\right)_+ .
\end{aligned}$$

Similarly, the update for $\boldsymbol{z}_{2i}$ is $\boldsymbol{z}_{2i}^{(k)} := \left(\boldsymbol{C}_i\boldsymbol{m}^{(k)} + \boldsymbol{u}_{2i}^{(k-1)}\right)\left(1 - \frac{\lambda}{\rho\left\|\boldsymbol{C}_i\boldsymbol{m}^{(k)} + \boldsymbol{u}_{2i}^{(k-1)}\right\|_2}\right)_+$.

C.2 Stopping Criterion

We use the stopping criterion for Algorithm 1 suggested in Boyd et al. (2011), stopping when the primal residual $\boldsymbol{r}^{(k)} = \boldsymbol{A}\boldsymbol{m}^{(k)} - \boldsymbol{z}^{(k)}$ and dual residual $\boldsymbol{s}^{(k)} = \rho\boldsymbol{A}^T\left(\boldsymbol{z}^{(k-1)} - \boldsymbol{z}^{(k)}\right)$ are sufficiently small. Specifically, we check if

$$\|\boldsymbol{r}^{(k)}\|_2 \leq \sqrt{2q(q-1)}\epsilon^{abs} + \epsilon^{rel}\max\{\|\boldsymbol{A}\boldsymbol{m}^{(k)}\|_2, \|\boldsymbol{z}^{(k)}\|_2\} \quad \text{and} \quad \|\boldsymbol{s}^{(k)}\|_2 \leq q\epsilon^{abs} + \epsilon^{rel}\|\rho\boldsymbol{A}^T\boldsymbol{u}^{(k)}\|_2$$

with $\epsilon^{abs}, \epsilon^{rel} > 0$. We use $\epsilon^{abs} = 10^{-4}$ and $\epsilon^{rel} = 10^{-2}$ in order to obtain the results presented in Sections 3 and 6.

C.3 Varying Penalty Parameter

We can vary ρ from iteration to iteration in order to achieve better convergence and reduce the dependence of performance on the initially chosen ρ. We adopt the scheme for varying ρ that is reviewed in Boyd et al. (2011). Since we use the scaled dual variable, $\boldsymbol{u}$ must also

be updated in conjunction with the updating of ρ. At the end of each iteration, we apply the updates

$$(\rho^{(k+1)}, \boldsymbol{u}^{(k+1)}) := \begin{cases} (\tau^{incr}\rho^{(k)}, \boldsymbol{u}^{(k)}/\tau^{incr}) & \text{if } \|\boldsymbol{r}^{(k)}\|_2 > \delta\|\boldsymbol{s}^{(k)}\|_2 \\ (\rho^{(k)}/\tau^{decr}, \tau^{decr}\boldsymbol{u}^{(k)}) & \text{if } \|\boldsymbol{s}^{(k)}\|_2 > \delta\|\boldsymbol{r}^{(k)}\|_2 \\ (\rho^{(k)}, \boldsymbol{u}^{(k)}) & \text{otherwise} \end{cases}$$

where $\delta, \tau^{incr}, \tau^{decr} > 1$. We choose $\delta = 10$ and $\tau^{incr} = \tau^{decr} = 2$. Updating ρ keeps the norms of the residuals $\boldsymbol{r}^{(k)}$ and $\boldsymbol{s}^{(k)}$ within a factor of δ of one another. While convergence of ADMM has only been proven for fixed ρ, varying ρ has been shown to work well in practice (Boyd et al., 2011).

C.4 Modification to Provide Sparsity

Inspection of the updates for $\boldsymbol{z}_{1i}^*$ and $\boldsymbol{z}_{2i}^*$ in Algorithm 1 indicates that the ADMM algorithm yields sparsity in $\boldsymbol{z}_{1i}^*$ and $\boldsymbol{z}_{2i}^*$, but not necessarily exact equality of the rows and columns of $\boldsymbol{M}^*$. This is in effect a numerical issue: our algorithm might yield $\boldsymbol{z}_{1i} = \boldsymbol{0}$, but $\|\boldsymbol{M}_{i\cdot}^* - \boldsymbol{M}_{(i+1)\cdot}^*\|_2 = 1 \times 10^{-8}$. To resolve this issue, we first determine the "blocks" of $\boldsymbol{m}^*$ using an initial run of Algorithm 1, and then solve (4) once more with constraints on the rows and columns of $\boldsymbol{M}$ to enforce equality of the appropriate rows and columns. This second optimization is performed simply to yield an estimate of $\boldsymbol{M}$ for which elements are exactly equal within each block.

Appendix D. Details of Simulations in Section 3

The mean models $f(x_1, x_2)$ used to generate data for Scenarios 1–4 in Section 3 are defined as follows. Note that x_1 and x_2 are sampled uniformly from $[-2.5, 2.5]$. We define the indicator function $\mathbf{1}_{\mathcal{A}}(x) = \begin{cases} 1 & \text{if } x \in \mathcal{A} \\ 0 & \text{otherwise} \end{cases}$.

Scenario 1: $f(x_1, x_2) = \text{sign}(x_1) \times \mathbf{1}_{[0,\infty)}(x_1 \times x_2)$

Scenario 2: $f(x_1, x_2) = -\text{sign}(x_1 \times x_2)$

Scenario 3: $f(x_1, x_2) = -3 \times \mathbf{1}_{[-2.5,-0.83)}(x_1) \times \mathbf{1}_{[-2.5,-1.25)}(x_2) + \mathbf{1}_{[-2.5,-0.83)}(x_1) \times \mathbf{1}_{[-1.25,2.5]}(x_2) - 2 \times \mathbf{1}_{[-0.83,0.83]}(x_1) \times \mathbf{1}_{[-2.5,0)}(x_2) + 2 \times \mathbf{1}_{[-0.83,0.83]}(x_1) \times \mathbf{1}_{[0,2.5]}(x_2) - \mathbf{1}_{(0.83,2.5]}(x_1) \times \mathbf{1}_{[-2.5,1.25)}(x_2) + 3 \times \mathbf{1}_{(0.83,2.5]}(x_1) \times \mathbf{1}_{[1.25,2.5]}(x_2)$

Scenario 4: $f(x_1, x_2) = \frac{10}{\left(\frac{x_1-2.5}{3}\right)^2 + \left(\frac{x_2-2.5}{3}\right)^2 + 1} + \frac{10}{\left(\frac{x_1+2.5}{3}\right)^2 + \left(\frac{x_2+2.5}{3}\right)^2 + 1}$

Each of the mean models $f(x_1, x_2)$ defined above is centered and scaled such that $\int_{-2.5}^{2.5}\int_{-2.5}^{2.5} f(x_1, x_2)\, dx_1\, dx_2 = 0$ and $\frac{1}{25}\int_{-2.5}^{2.5}\int_{-2.5}^{2.5} f(x_1, x_2)^2\, dx_1\, dx_2 = 2$.

Appendix E. Proof Sketch of Proposition 1

Proof Using the dual problem of (5) and Lemma 1 of Tibshirani and Taylor (2012), it can be shown that $g : \mathbb{R}^n \to \mathbb{R}^n$ with $\hat{\boldsymbol{y}} = g(\boldsymbol{y}) = (g_1(\boldsymbol{y}), \ldots, g_n(\boldsymbol{y}))^T$ is continuous and almost differentiable. Thus, Stein's lemma implies that $\text{df}(\hat{\boldsymbol{y}}) = \text{E}\left[\text{Tr}\left(\frac{\partial g(\boldsymbol{y})}{\partial \boldsymbol{y}}\right)\right]$. At the optimum

of (5), we have

$$\boldsymbol{Q}^T(\boldsymbol{y}-\boldsymbol{Q}\boldsymbol{m}^*) = \lambda\sum_{i=1}^{q-1}[\boldsymbol{R}_i^T S_1(\boldsymbol{R}_i,\boldsymbol{m}^*) + \boldsymbol{C}_i^T S_1(\boldsymbol{C}_i,\boldsymbol{m}^*)] + \gamma\boldsymbol{m}^*, \tag{18}$$

where $S_1(\boldsymbol{A}_i,\boldsymbol{m}^*) = \begin{cases} \frac{\boldsymbol{A}_i\boldsymbol{m}^*}{\|\boldsymbol{A}_i\boldsymbol{m}^*\|_2} & \text{if } \|\boldsymbol{A}_i\boldsymbol{m}^*\|_2 \neq 0 \\ \in \{\boldsymbol{g} : \|\boldsymbol{g}\|_2 \leq 1\} & \text{if } \|\boldsymbol{A}_i\boldsymbol{m}^*\|_2 = 0 \end{cases}$.

We define $\mathcal{C} = \{\boldsymbol{A}_i : \|\boldsymbol{A}_i\boldsymbol{m}^*\|_2 = 0\}$ where $\boldsymbol{A}_1 = \boldsymbol{R}_1, \boldsymbol{A}_2 = \boldsymbol{R}_2, \ldots, \boldsymbol{A}_{q-1} = \boldsymbol{R}_{q-1}, \boldsymbol{A}_q = \boldsymbol{C}_1, \boldsymbol{A}_{q+1} = \boldsymbol{C}_2, \ldots, \boldsymbol{A}_{2q-2} = \boldsymbol{C}_{q-1}$. We define $\boldsymbol{A}_*$ to be the submatrix of $\boldsymbol{A}$ with the rows corresponding to $\boldsymbol{A}_i \notin \mathcal{C}$ removed, and let $\boldsymbol{P} = \boldsymbol{I}_{q^2} - \boldsymbol{A}_*^+\boldsymbol{A}_*$, the projection onto the space orthogonal to the row space of $\boldsymbol{A}_*$. We left-multiply (18) by $\boldsymbol{P}$ to give

$$\boldsymbol{P}\boldsymbol{Q}^T(\boldsymbol{y}-\boldsymbol{Q}\boldsymbol{m}^*) = \lambda\boldsymbol{P}\sum_{i:\boldsymbol{A}_i\notin\mathcal{C}}\frac{\boldsymbol{A}_i^T\boldsymbol{A}_i\boldsymbol{m}^*}{\|\boldsymbol{A}_i\boldsymbol{m}^*\|_2} + \gamma\boldsymbol{P}\boldsymbol{m}^*, \tag{19}$$

since $\boldsymbol{P}\boldsymbol{A}_i^T S_1(\boldsymbol{A}_i,\boldsymbol{m}^*) = \boldsymbol{0}$ if $\boldsymbol{A}_i \in \mathcal{C}$ (i.e., $\|\boldsymbol{A}_i\boldsymbol{m}^*\|_2 = 0$). Because $\boldsymbol{P}\boldsymbol{m}^* = \boldsymbol{m}^*$, (19) can be rewritten as

$$\boldsymbol{P}\boldsymbol{Q}^T(\boldsymbol{y}-\boldsymbol{Q}\boldsymbol{P}\boldsymbol{m}^*) = \lambda\boldsymbol{P}\sum_{i:\boldsymbol{A}_i\notin\mathcal{C}}\frac{\boldsymbol{A}_i^T\boldsymbol{A}_i\boldsymbol{P}\boldsymbol{m}^*}{\|\boldsymbol{A}_i\boldsymbol{m}^*\|_2} + \gamma\boldsymbol{m}^*. \tag{20}$$

We let $\boldsymbol{D} = \text{diag}\left(h(m_1^*),\cdots,h(m_{q^2}^*)\right)$, where $h(m_i^*)$ is defined to be the ratio of the number of observations in the block of $\boldsymbol{M}^*$ that contains m_i^* to the number of elements of $\boldsymbol{M}^*$ in the block of $\boldsymbol{M}^*$ that contains m_i^*. Note that $\boldsymbol{P}\boldsymbol{Q}^T\boldsymbol{Q}\boldsymbol{P} = \boldsymbol{D}\boldsymbol{P}$. Thus $\boldsymbol{P}\boldsymbol{Q}^T\boldsymbol{Q}\boldsymbol{P}\boldsymbol{m}^* = \boldsymbol{D}\boldsymbol{m}^*$, and (20) is equivalent to

$$\boldsymbol{P}\boldsymbol{Q}^T\boldsymbol{y} = \boldsymbol{D}\boldsymbol{m}^* + \lambda\boldsymbol{P}\sum_{i:\boldsymbol{A}_i\notin\mathcal{C}}\frac{\boldsymbol{A}_i^T\boldsymbol{A}_i\boldsymbol{P}\boldsymbol{m}^*}{\|\boldsymbol{A}_i\boldsymbol{m}^*\|_2} + \gamma\boldsymbol{m}^*. \tag{21}$$

We conjecture that there is a neighborhood around almost every $\boldsymbol{y}$ such that the blocks of $\boldsymbol{m}^*$ do not change. That is, $\mathcal{C}$ and $\boldsymbol{P}$ in (21) are constant with respect to $\boldsymbol{y}$, and the derivative of (21) with respect to $\boldsymbol{y}$ is

$$\boldsymbol{P}\boldsymbol{Q}^T = \left(\boldsymbol{D} + \lambda\boldsymbol{P}\sum_{i:\boldsymbol{A}_i\notin\mathcal{C}} S_2(\boldsymbol{A}_i,\boldsymbol{m}^*)\boldsymbol{P} + \gamma\boldsymbol{I}\right)\frac{\partial\boldsymbol{m}^*}{\partial\boldsymbol{y}}, \tag{22}$$

where $S_2(\boldsymbol{A}_i,\boldsymbol{m}^*) = \frac{\boldsymbol{A}_i^T\boldsymbol{A}_i}{\|\boldsymbol{A}_i\boldsymbol{m}^*\|_2} - \frac{\boldsymbol{A}_i^T\boldsymbol{A}_i\boldsymbol{m}^*\boldsymbol{m}^{*T}\boldsymbol{A}_i^T\boldsymbol{A}_i}{\|\boldsymbol{A}_i\boldsymbol{m}^*\|_2^3}$. Recall $\hat{\boldsymbol{y}} = \boldsymbol{Q}\boldsymbol{m}^*$, so solving (22) for $\frac{\partial\boldsymbol{m}^*}{\partial\boldsymbol{y}}$ and left-multiplying by $\boldsymbol{Q}$ gives

$$\frac{\partial\hat{\boldsymbol{y}}}{\partial\boldsymbol{y}} = \boldsymbol{Q}\left(\boldsymbol{D} + \lambda\boldsymbol{P}\sum_{i:\boldsymbol{A}_i\notin\mathcal{C}} S_2(\boldsymbol{A}_i,\boldsymbol{m}^*)\boldsymbol{P} + \gamma\boldsymbol{I}\right)^{-1}\boldsymbol{P}\boldsymbol{Q}^T,$$

where $\left(\boldsymbol{D} + \lambda \boldsymbol{P} \sum_{i:\boldsymbol{A}_i \notin \mathcal{C}} S_2(\boldsymbol{A}_i, \boldsymbol{m}^*)\boldsymbol{P} + \gamma \boldsymbol{I}\right)$ is invertible as both $\boldsymbol{D}$ and $\lambda \boldsymbol{P} \sum_{i:\boldsymbol{A}_i \notin \mathcal{C}} S_2(\boldsymbol{A}_i, \boldsymbol{m}^*)\boldsymbol{P}$ are positive semi-definite. Therefore, the degrees of freedom is

$$\mathrm{E}\left[\mathrm{Tr}\left(\boldsymbol{Q}\left(\boldsymbol{D} + \lambda \boldsymbol{P} \sum_{i:\boldsymbol{A}_i \notin \mathcal{C}} S_2(\boldsymbol{A}_i, \boldsymbol{m}^*)\boldsymbol{P} + \gamma \boldsymbol{I}\right)^{-1} \boldsymbol{P}\boldsymbol{Q}^T\right)\right].$$

This establishes the unbiasedness of the estimator (6). ■

Appendix F. Proof of Corollary 2

Proof This corollary pertains to the setting in which either all rows of $\boldsymbol{M}^*$ are equal (i.e., $\boldsymbol{R}_i \in \mathcal{C}$ for all i) or all columns of $\boldsymbol{M}^*$ are equal (i.e., $\boldsymbol{C}_i \in \mathcal{C}$ for all i). In this setting, we will show $\boldsymbol{P}S_2(\boldsymbol{A}_i, \boldsymbol{m}^*) = \boldsymbol{0}$ for any $\boldsymbol{A}_i \notin \mathcal{C}$ using two facts: (1) $\boldsymbol{A}_i\boldsymbol{m}^* = c_i\boldsymbol{1}_q$ for some $c_i \in \mathbb{R}$ and (2) $\boldsymbol{P}\boldsymbol{A}_i^T = \boldsymbol{v}_i\boldsymbol{1}_q^T$ for some $\boldsymbol{v}_i \in \mathbb{R}^{q^2}$. These facts follow from the assumption that either all rows or all columns of $\boldsymbol{M}^*$ are equal. Consider some $\boldsymbol{A}_i \notin \mathcal{C}$. We have

$$\begin{aligned}
\boldsymbol{P}S_2(\boldsymbol{A}_i, \boldsymbol{m}^*) &= \frac{\boldsymbol{P}\boldsymbol{A}_i^T\boldsymbol{A}_i}{\|\boldsymbol{A}_i\boldsymbol{m}^*\|_2} - \frac{\boldsymbol{P}\boldsymbol{A}_i^T\boldsymbol{A}_i\boldsymbol{m}^*\boldsymbol{m}^{*T}\boldsymbol{A}_i^T\boldsymbol{A}_i}{\|\boldsymbol{A}_i\boldsymbol{m}^*\|_2^3} \\
&= \frac{\boldsymbol{P}\boldsymbol{A}_i^T\boldsymbol{A}_i}{\|\boldsymbol{A}_i\boldsymbol{m}^*\|_2} - \frac{(\boldsymbol{v}_i\boldsymbol{1}_q^T)(c_i\boldsymbol{1}_q)(c_i\boldsymbol{1}_q^T)\boldsymbol{A}_i}{c_i^2 q\|\boldsymbol{A}_i\boldsymbol{m}^*\|_2} \\
&= \frac{\boldsymbol{P}\boldsymbol{A}_i^T\boldsymbol{A}_i}{\|\boldsymbol{A}_i\boldsymbol{m}^*\|_2} - \frac{\boldsymbol{v}_i\boldsymbol{1}_q^T\boldsymbol{A}_i}{\|\boldsymbol{A}_i\boldsymbol{m}^*\|_2} \\
&= \frac{\boldsymbol{P}\boldsymbol{A}_i^T\boldsymbol{A}_i}{\|\boldsymbol{A}_i\boldsymbol{m}^*\|_2} - \frac{\boldsymbol{P}\boldsymbol{A}_i^T\boldsymbol{A}_i}{\|\boldsymbol{A}_i\boldsymbol{m}^*\|_2} \\
&= \boldsymbol{0}.
\end{aligned}$$

Therefore, the estimator (6) with $\gamma = 0$ simplifies to $\mathrm{Tr}[\boldsymbol{Q}\boldsymbol{D}^{-1}\boldsymbol{P}\boldsymbol{Q}^T] = \mathrm{Tr}[\boldsymbol{D}^{-1}\boldsymbol{P}\boldsymbol{Q}^T\boldsymbol{Q}]$. Recall that $\boldsymbol{D}$ is a diagonal matrix with $D_{ii} = h(m_i^*) = N_i^0/N_i$, where N_i^0 and N_i are the number of observations and the number of elements, respectively, in the block of $\boldsymbol{M}^*$ containing m_i^*. Note that $(\boldsymbol{P}\boldsymbol{Q}^T\boldsymbol{Q})_{ii}$ equals n_i^0/N_i, where n_i^0 is the number of observations corresponding to m_i^*. Thus

$$\mathrm{Tr}[\boldsymbol{D}^{-1}\boldsymbol{P}\boldsymbol{Q}^T\boldsymbol{Q}] = \sum_{i=1}^{q^2} \frac{(\boldsymbol{P}\boldsymbol{Q}^T\boldsymbol{Q})_{ii}}{D_{ii}} = \sum_{i:m_i^* \text{ observed}} \frac{N_i}{N_i^0}\frac{n_i^0}{N_i} = \sum_{i:m_i^* \text{ observed}} \frac{n_i^0}{N_i^0},$$

which equals the total number of blocks of $\boldsymbol{M}^*$ since the n_i^0's for a block sum to N_i^0. ■

Appendix G. Proof of Lemma 3

Proof If $\boldsymbol{m}^* = \left(\frac{1}{n}\mathbf{1}_n^T\boldsymbol{y}\right)\mathbf{1}_{q^2}$ solves (3), then there exist q-vectors $\boldsymbol{d}_{1i}, \boldsymbol{d}_{2i}$ with $\|\boldsymbol{d}_{1i}\|_2 \leq \lambda$ and $\|\boldsymbol{d}_{2i}\|_2 \leq \lambda$ such that

$$\boldsymbol{Q}^T\left(\boldsymbol{y} - \left(\frac{1}{n}\mathbf{1}_n^T\boldsymbol{y}\right)\mathbf{1}_q\right) = \sum_{i=1}^{q-1}\left[\boldsymbol{R}_i^T\boldsymbol{d}_{1i} + \boldsymbol{C}_i^T\boldsymbol{d}_{2i}\right], \tag{23}$$

since $\boldsymbol{Q}\mathbf{1}_{q^2} = \mathbf{1}_q$. Let $\boldsymbol{d} = (\boldsymbol{d}_{11}^T \ \cdots \ \boldsymbol{d}_{1(q-1)}^T \ \boldsymbol{d}_{21}^T \ \ldots \ \boldsymbol{d}_{2(q-1)}^T)^T$. Then (23) can be rewritten as

$$\boldsymbol{Q}^T\left(\boldsymbol{y} - \left(\frac{1}{n}\mathbf{1}_n^T\boldsymbol{y}\right)\mathbf{1}_q\right) = \boldsymbol{A}^T\boldsymbol{d}. \tag{24}$$

Note that $\boldsymbol{m}^* = \left(\frac{1}{n}\mathbf{1}_n^T\boldsymbol{y}\right)\mathbf{1}_{q^2}$ for a certain λ if and only if (24) is satisfied for some $\boldsymbol{d}$ for which $\|\boldsymbol{d}_{1i}\|_2 \leq \lambda, \|\boldsymbol{d}_{2i}\|_2 \leq \lambda$ for $i = 1, \ldots, q-1$. We find the $\boldsymbol{d}^*$ corresponding to the minimum λ for which $\boldsymbol{m}^* = \left(\frac{1}{n}\mathbf{1}_n^T\boldsymbol{y}\right)\mathbf{1}_{q^2}$ by solving the convex optimization problem

$$\underset{\boldsymbol{d}}{\text{minimize}} \quad \max_{1\leq i\leq q-1}\left\{\|\boldsymbol{d}_{1i}\|_2, \|\boldsymbol{d}_{2i}\|_2\right\} \quad \text{subject to} \quad \boldsymbol{Q}^T\left(\boldsymbol{y} - \left(\frac{1}{n}\mathbf{1}_n^T\boldsymbol{y}\right)\mathbf{1}_q\right) = \boldsymbol{A}^T\boldsymbol{d}.$$

Thus $\boldsymbol{m}^* = \left(\frac{1}{n}\mathbf{1}_n^T\boldsymbol{y}\right)\mathbf{1}_{q^2}$ if and only if $\lambda \geq \max_{1\leq i\leq q-1}\left\{\|\boldsymbol{d}_{1i}^*\|_2, \|\boldsymbol{d}_{2i}^*\|_2\right\}$. ■

Appendix H. Simulations Illustrating Performance of (3) with $\lambda = 0$ and Variable q

We illustrate how (3) with $\lambda = 0$ over a range of q values performs compared to CRISP for a variety of scenarios. We generate data with $n = 100$ by independently sampling each element of $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ from a Unif$[-2.5, 2.5]$ distribution, and then taking $\boldsymbol{y} = f(\boldsymbol{x}_1, \boldsymbol{x}_2) + \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \text{MVN}(\mathbf{0}, \boldsymbol{I}_n)$. The four mean models $f(\boldsymbol{x}_1, \boldsymbol{x}_2)$ we consider are shown in Figure 12. Note that these are the same mean models we consider extensively in Section 3.

For each mean model, we generate 1000 replicates of data and estimate the mean model using (3) with $\lambda = 0$ and various q. We plot the MSE, squared bias, and variance of the mean model estimate as a function of q in Figure 12. In Scenarios 1 and 2, $q = 2$ has the best performance, which is unsurprising given the mean model structure. Using $q = 2$, there will be four bins whose boundaries roughly coincide with the true boundaries of the mean model. As q increases, the bias increases in an oscillating fashion where even values of q give better performance than odd ones. This is because odd values of q will not tend to have bins with boundaries that coincide with the true boundaries the mean model. As q increases, most of the q^2 bins will not have observations in them, and their estimates will be the mean of $\boldsymbol{y}$. Thus the variance decreases as many bins take on the same value, but the squared bias continues to increase. In Scenarios 3 and 4, the minimum MSE occurs at $q = 4$, not $q = 2$ as in Scenarios 1 and 2. This is because the mean models in Scenarios 3 and 4 are more complex and not well-estimated using only 2×2 grid of bins.

We also consider the performance for an additional mean model, shown in Figure 13(a). The same simulation set-up was used as for Scenarios 1–4 above. Though the blocks of the true mean model perfectly align with a grid that has $q = 3$, there are only 4 distinct blocks.

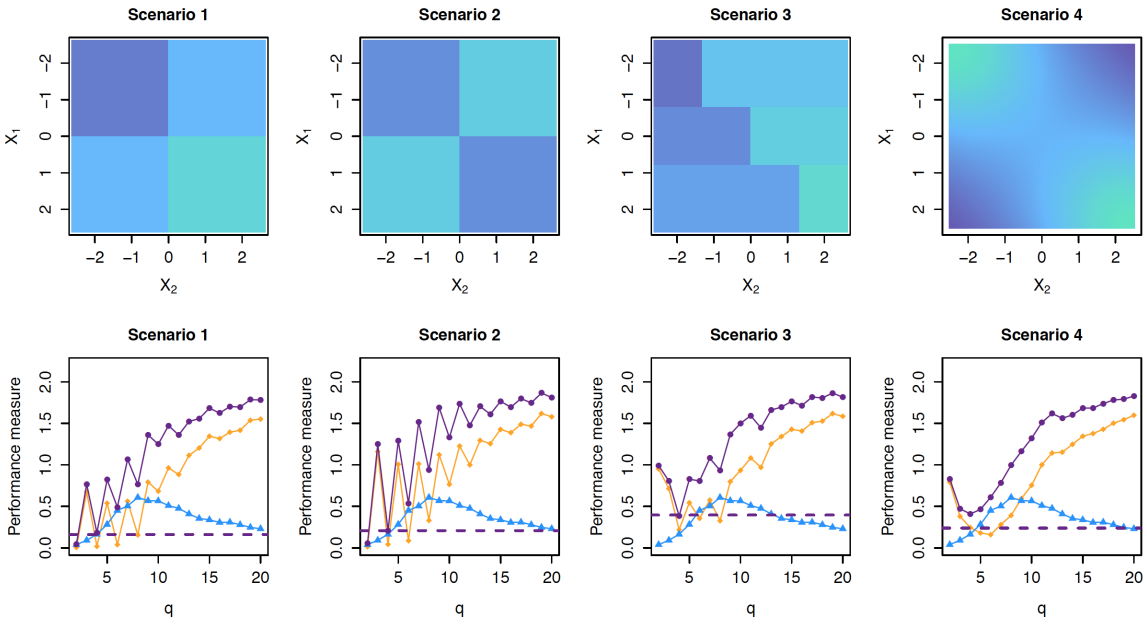

Figure 12: The top row of figures shows the mean models $f(x_1, x_2)$ used to generate data in each of the four scenarios in Appendix H. The bottom row of figures shows the performance of the method of (3) with $\lambda = 0$ as a function of q in terms of MSE (—●—), squared bias (—◆—), and variance (—▲—). The MSE for CRISP with $q = n$ and optimal λ is shown (- - -) for comparison.

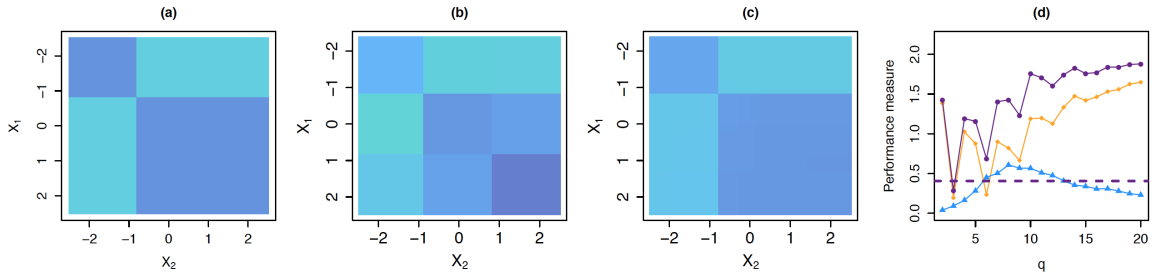

Figure 13: In *(a)*, we plot the mean model $f(x_1, x_2)$ used to generate data for the simulation described in Appendix H. In *(b)*, we show the estimated mean model from the method of (3) with $\lambda = 0$ and $q = 3$. In *(c)*, we show the estimated mean model from CRISP with $q = n$. In *(d)*, we show the performance of the method of (3) with $\lambda = 0$ as a function of q in terms of MSE (—●—), squared bias (—◆—), and variance (—▲—). The MSE for CRISP with $q = n$ and optimal λ is shown (- - -) for comparison.

The method of (3) with $\lambda = 0$ unsurprisingly has the best performance for $q = 3$, which is shown in Figure 13(d). The estimated mean model from using $q = 3$ and $\lambda = 0$ has $q^2 = 9$ blocks, as shown in Figure 13(b), since there is no adaptive shrinking together of blocks. However, CRISP is able to adaptively determine that only 4 blocks are needed, as shown in the estimated mean model in Figure 13(c). This example illustrates how CRISP is able to adaptively determine the amount of granularity over the covariate space. With $\lambda = 0$, the amount of granularity is constant across the covariate space.

Appendix I. Details of Data Application

In Section 6, we analyze housing data with the outcome of median house value and predictors of median income and average occupancy. We plot median income versus average occupancy in Figure 14. Note that 37 neighborhoods had an average occupancy larger than 10 and are omitted from the plot. The mean of average occupancy for these neighborhoods with an average occupancy greater than 10 was 88. In Figure 14, we outline the central 95% of the data in both covariates. That is, the 2.5% and 97.5% quantiles are shown for both covariates. We restrict our analysis to observations that fall in the central 95% of the data for both covariates. Of the original 20,640 neighborhoods, this excludes 1978 observations, leaving 18,662 observations for analysis.

References

Hirotugu Akaike. Information theory and an extension of the maximum likelihood principle. In *Second International Symposium on Information Theory*, pages 267–281. Akademinai Kiado, 1973.

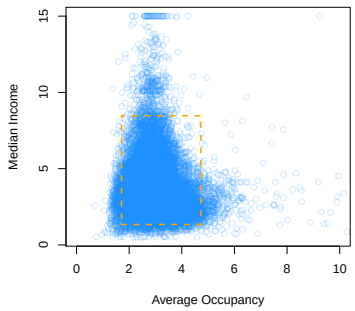

Figure 14: We plot median income versus average occupancy for the housing data considered in Section 6 and described in Appendix I. The rectangle (- - -) identifies observations falling within the central 95% of the data for both covariates.

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

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

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

Leo Breiman, Jerome Friedman, Charles J. Stone, and Richard A. Olshen. *Classification and Regression Trees*. CRC Press, 1984.

John Duchi and Yoram Singer. Efficient online and batch learning using forward backward splitting. *The Journal of Machine Learning Research*, 10:2899–2934, 2009.

Jean Duchon. Splines minimizing rotation-invariant semi-norms in Sobolev spaces. In *Constructive Theory of Functions of Several Variables*, pages 85–100. Springer, 1977.

Bradley Efron. How biased is the apparent error rate of a prediction rule? *Journal of the American Statistical Association*, 81(394):461–470, 1986.

Jerome H. Friedman. Multivariate adaptive regression splines. *The Annals of Statistics*, pages 1–67, 1991.

Michael Grant and Stephen Boyd. Graph implementations for nonsmooth convex programs. In V. Blondel, S. Boyd, and H. Kimura, editors, *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, pages 95–110. Springer-Verlag Limited, 2008. `http://stanford.edu/~boyd/graph_dcp.html`.

Michael Grant and Stephen Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. `http://cvxr.com/cvx`, March 2014.

Trevor J. Hastie and Robert J. Tibshirani. *Generalized Additive Models*, volume 43. CRC Press, 1990.

Seung-Jean Kim, Kwangmoo Koh, Stephen Boyd, and Dimitry Gorinevsky. ℓ_1 trend filtering. *SIAM Review*, 51(2):339–360, 2009.

Douglas Nychka, Reinhard Furrer, and Stephan Sain. *fields: Tools for spatial data*, 2014. URL `http://CRAN.R-project.org/package=fields`. R package version 7.1.

R. Kelley Pace and Ronald Barry. Sparse spatial autoregressions. *Statistics & Probability Letters*, 33(3):291–297, 1997.

Ashley Petersen. *flam: Fits Piecewise Constant Models with Data-Adaptive Knots*, 2014. URL `http://CRAN.R-project.org/package=flam`. R package version 1.0.

Ashley Petersen, Daniela Witten, and Noah Simon. Fused lasso additive model. *Journal of Computational and Graphical Statistics*, forthcoming.

Aaditya Ramdas and Ryan J. Tibshirani. Fast and flexible ADMM algorithms for trend filtering. *Journal of Computational and Graphical Statistics*, forthcoming.

Gideon Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2): 461–464, 1978.

Terry Therneau, Beth Atkinson, and Brian Ripley. *rpart: Recursive Partitioning and Regression Trees*, 2014. URL `http://CRAN.R-project.org/package=rpart`. R package version 4.1-8.

Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):91–108, 2005.

Ryan J. Tibshirani and Jonathan Taylor. Degrees of freedom in lasso problems. *The Annals of Statistics*, 40(2):1198–1232, 2012.

Paul Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of Optimization Theory and Applications*, 109(3):475–494, 2001.

Sara van de Geer. *Empirical Processes in M-estimation*, volume 6. Cambridge University Press, 2000.

Jianming Ye. On measuring and correcting the effects of data mining and model selection. *Journal of the American Statistical Association*, 93(441):120–131, 1998.

Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

Hui Zou, Trevor Hastie, and Robert Tibshirani. On the "degrees of freedom" of the lasso. *The Annals of Statistics*, 35(5):2173–2192, 2007.