Received: 22 September 2020 | Revised: 22 December 2021 | Accepted: 14 February 2022

DOI: 10.1111/sjos.12580

ORIGINAL ARTICLE

Scandinavian Journal of Statistics

Soft maximin estimation for heterogeneous data

Adam Lund[1] | **Søren Wengel Mogensen**[2] | **Niels Richard Hansen**[2]

[1]Speevr Labs, Speevr, Millbrae, California, USA

[2]Department of Mathematical Sciences, University of Copenhagen, Copenhagen, Denmark

Correspondence
Adam Lund, Speevr Labs, Speevr, PO Box 116, Millbrae, CA 94030, USA.
Email: adam.lund@speevr.com

Abstract

Extracting a common robust signal from data divided into heterogeneous groups is challenging when each group—in addition to the signal—contains large, unique variation components. Previously, maximin estimation was proposed as a robust method in the presence of heterogeneous noise. We propose soft maximin estimation as a computationally attractive alternative aimed at striking a balance between pooled estimation and (hard) maximin estimation. The soft maximin method provides a range of estimators, controlled by a parameter $\zeta > 0$, that interpolates pooled least squares estimation and maximin estimation. By establishing relevant theoretical properties we argue that the soft maximin method is statistically sensible and computationally attractive. We demonstrate, on real and simulated data, that soft maximin estimation can offer improvements over both pooled OLS and hard maximin in terms of predictive performance and computational complexity. A time and memory efficient implementation is provided in the R package `SMME` available on CRAN.

KEYWORDS

convex optimization, heterogeneous data, robust estimation, sparse estimation

1 | INTRODUCTION

We consider the problem of extracting a common signal from heterogeneous data. As heterogeneity is prevalent in large-scale settings our aim is a computationally efficient estimator (solution) with good statistical properties under varying degrees of data heterogeneity.

To make the concept of heterogeneity concrete, consider the linear *mixture* model with univariate response variables $Y_1, \ldots, Y_n$ generated as

$$Y_i = X_i^\top B_i + \varepsilon_i, \quad i = 1, \ldots, n. \tag{1}$$

Here, $B_1, \ldots, B_n$ and the feature vectors $X_1, \ldots, X_n$ are p-dimensional random variables and $\varepsilon_1, \ldots, \varepsilon_n$ are univariate noise variables. The feature vectors are observed and assumed i.i.d., and the noise variables are likewise i.i.d. The unobserved variables $B_1, \ldots, B_n$ are identically distributed with distribution F_B but not necessarily independent, see also Meinshausen and Bühlmann (2015).

Heterogeneity in the model given by (1) is due to the variation in B_i as governed by F_B. Because the B_is can be dependent, model (1) can capture heterogeneity caused by a group structure, that is, when data come with a natural grouping and B_i is constant within groups but vary across groups. Even if data are not grouped, or if the group structure is unknown, it is beneficial to study the setup with a known group structure. In the example below on bike sharing, a group structure is introduced to represent temporal heterogeneity, and Meinshausen and Bühlmann (2015) demonstrate how to construct group structures as part of the inference when no grouping is given.

Our focus is therefore on a setup with G groups and with B_i constant within groups. The objective is to learn a single $\beta \in \mathbb{R}^p$ that can sensibly be regarded as a *common signal* of the B_is. Pooling data across groups and computing the ordinary least squares (OLS) estimator may be nonrobust, depending on F_B, and Meinshausen and Bühlmann (2015) introduced maximin estimation as a robust alternative to OLS for heterogeneous data from the model (1). The common signal estimated by maximin estimation is the population quantity called the maximin effect.

While the maximin estimator is robust, it can also be conservative, and we propose *soft maximin estimation* to strike a good balance between maximin and pooled OLS estimation. The balance is controlled by a tuning parameter $\zeta > 0$, with $\zeta \to \infty$ corresponding to maximin estimation. Figure 1 shows the result of applying the soft maximin estimator, for three values of ζ, as well as the pooled OLS estimator to a real dataset. It illustrates how predictive performance of the two extreme estimators is interpolated by soft maximin estimation, quantified as cumulative root mean square error (RMSE) over time, see (21).

The specific application illustrated in Figure 1 is described in detail in Section 4.1. The data is on the hourly number of bike shares (see Fanaee-T & Gama, 2013) from 2 years (2011 and 2012), and the model predicts this number based on weekday, time of the day and the weather. In this application, 1 year is used for training and the other year is used for validation (prediction). To safeguard against temporal heterogeneity the data is grouped according to the months variable `mnth`, thus $G = 12$. Training on 2011 makes soft maximin with a high value of ζ too conservative, which leads to poor predictive performance. In this case, pooled OLS or soft maximin with a low ζ perform best. However, training on 2012 data makes the pooled OLS estimator overfit and soft maximin with a large ζ has better predictive performance.

The paper is organized as follows. The model and estimation framework is outlined in Section 2 and statistical properties of soft maximin estimation are discussed. This section includes

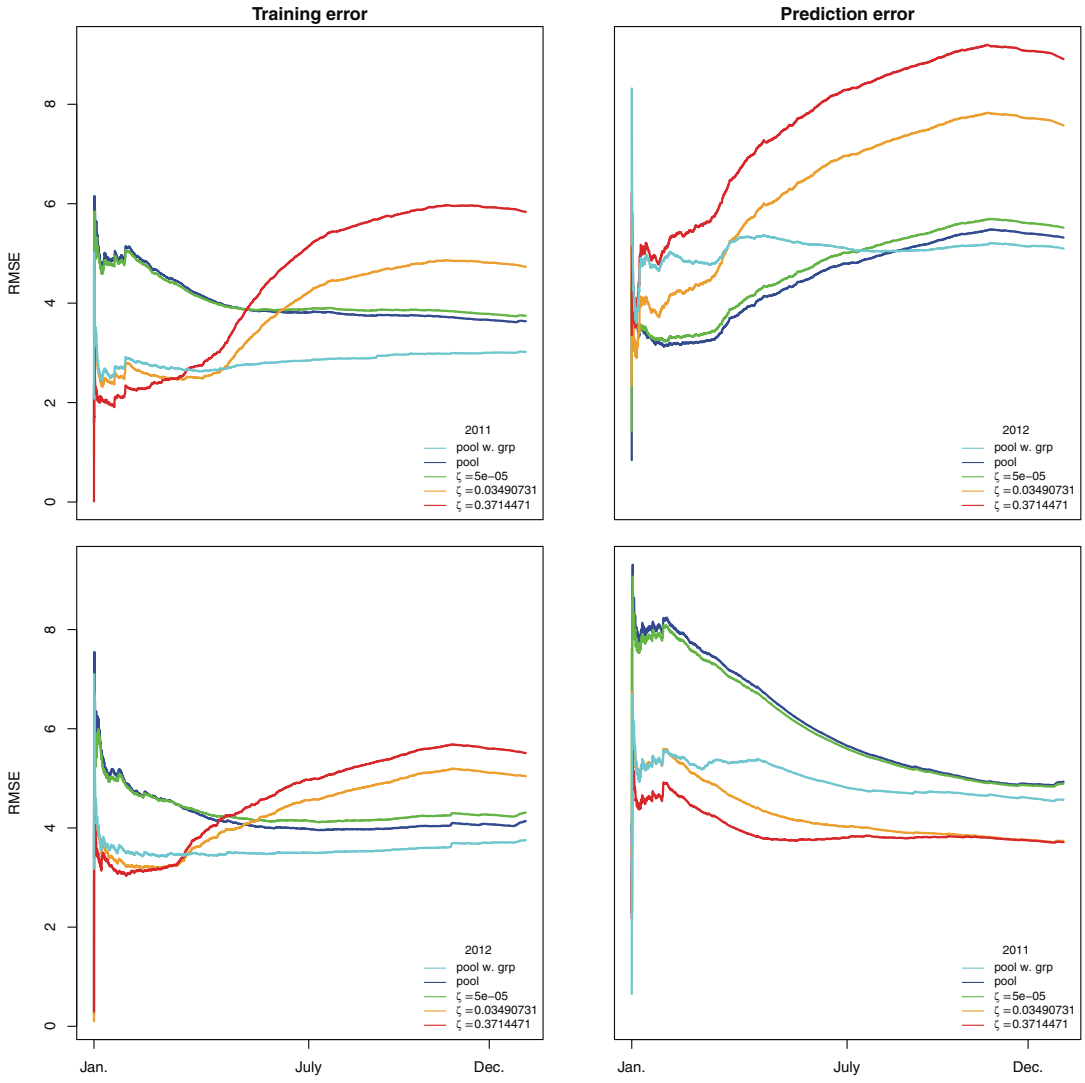

FIGURE 1 Top: RMSE computed for 2011 training data and 2012 validation data for the soft maximin estimators for 3 values of ζ (resp. green, orange and red), the pooled OLS estimator (blue) and the pooled OLS estimator using the group variable `mnth` (cyan). Bottom: RMSE computed for 2012 training data and 2011 validation data using the same methods and coloring [Colour figure can be viewed at wileyonlinelibrary.com]

theoretical results supporting that soft maximin interpolates maximin and pooled OLS estimation. In Section 3 we propose an algorithm for computing the general soft maximin estimator. The algorithm solves a nondifferentiable convex optimization problem, but as opposed to maximin estimation (Meinshausen & Bühlmann, 2015), the problem we solve is separable in the sense of Tseng and Yun (2009). This makes it notably easier to construct efficient algorithms with convergence guarantees. Section 3 also includes theoretical bounds on Lipschitz constants, which can be used to select an efficient step size in our solution algorithm, and a discussion of how our results can be applied efficiently to array tensor smoothing. In Section 4, we present the application to the bike share data and results from a simulation study on array tensor smoothing. The simulation study was inspired by the application to neuronal activity data analyzed in Appendix

B. Section 5 summarizes the proposed methodology and its relation to alternative methods. The algorithms are implemented in the R package `SMME` available from CRAN, see Lund (2021).

2 | SOFT MAXIMIN ESTIMATION

Here we present the methodology in a setup with a given group structure and effects B_i constant within each group. This in turn implies a finite support for F_B and this model is perhaps better understood as a type of linear *mixed* model as the grouping is available hence no longer part of the inference. In contrast to a traditional mixed model, however, we avoid explicitly modeling fixed and random effects since the aim is not to draw inference about these. Instead we seek to obtain only an estimate of the possible common effects present in the data.

To introduce $G \in \mathbb{N}$ groups for the model (1), suppose given a partition $I_1, \dots, I_G$ of the index set $\{1, \dots, n\}$ such that $|I_g| = n_g$, and $n = \sum_g n_g$ where for each g and for all $i \in I_g$, $B_i = B_g$ is the effect in group g. Thus F_B has finite unknown support with cardinality G, that is $\text{supp}(F_B) = \{b_1, \dots, b_G\} \subset \mathbb{R}^p$ with $b_g := B_g(\omega)$ the true unknown effect in group g.

Using this extra structure we can label the response, covariates and errors according to group. For the gth group let $\boldsymbol{Y}_g = (Y_{g,1}, \dots, Y_{g,n_g})^\top$ be the $n_g \times 1$ response vector, $\boldsymbol{X}_g = (X_{g,1} | \dots | X_{g,n_g})^\top$ the $n_g \times p$ design matrix, and $\boldsymbol{\varepsilon}_g = (\varepsilon_{g,1}, \dots, \varepsilon_{g,n_g})^\top$ the $n_g \times 1$ error vector. The linear model for the gth group is then

$$\boldsymbol{Y}_g = \boldsymbol{X}_g b_g + \boldsymbol{\varepsilon}_g, \quad g \in \{1, \dots, G\}. \tag{2}$$

A *common signal* in this framework is represented by a single $\beta \in \mathbb{R}^p$ such that $\boldsymbol{X}_g\beta$ is a good and robust approximation of $\boldsymbol{X}_g b_g$ across all G groups.

To gauge the quality of the approximation we adopt the optimality criterion from Meinshausen and Bühlmann (2015). There the explained variance in group g when using some $\beta \in \mathbb{R}^p$ in (2) is defined as

$$V_{b_g}(\beta) := 2\beta^\top \Sigma b_g - \beta^\top \Sigma \beta. \tag{3}$$

The optimal approximation across all groups is then the so-called maximin effect defined as a $b^* \in \mathbb{R}^p$ that maximizes the minimum of the explained variances across groups, that is, $b^* := \arg\max_\beta \min_g V_{b_g}(\beta)$.

Since b_g is unknown, to make this criteria operational, let $\hat{\Sigma}_g := \boldsymbol{X}_g^\top \boldsymbol{X}_g / n_g$ denote the empirical Gram matrix in group g. By (2) replacing Σ with $\hat{\Sigma}_g$ in (3) we obtain the empirical explained variance in group g

$$\hat{V}_g(\beta) := \frac{1}{n_g}(2\beta^\top \boldsymbol{X}_g^\top \boldsymbol{Y}_g - \beta^\top \boldsymbol{X}_g^\top \boldsymbol{X}_g \beta). \tag{4}$$

The maximin effects estimator is obtained by maximizing (the penalized) minimum of (4) or equivalently by minimizing (the penalized) maximin loss function $\beta \mapsto \max_g\{-\hat{V}_g(\beta)\}$, see (23) in Section 5.

As shown in Meinshausen and Bühlmann (2015) the use of this estimator can lead to more robust estimates for heterogeneous data compared to an estimator that does not take grouping into account, that is, a pooled estimator. The intuition is that the maximin estimator extracts only

features that are active with the same sign across groups while setting group specific features to zero. This makes it a more crude estimator compared to one obtained using the full mixed model methodology; however, it is in principle also more robust and potentially computationally more attractive. In large-scale data settings, where data heterogeneity is typically encountered, the computational aspect of the estimator is crucial. However, since the max-function is nondifferentiable and nonseparable the maximin problem (23) is not easy to solve.

We address this computational hurdle by replacing the max-function with the following smooth function. For $G \in \mathbb{N}$ and $\zeta \neq 0$ consider the scaled log-sum exponential function

$$\operatorname{lse}_\zeta(x) := \frac{\log(\sum_j e^{\zeta x_j})}{\zeta}, \quad x \in \mathbb{R}^G. \tag{5}$$

Clearly lse_ζ is differentiable and as we show in Section 3 it has additional properties that makes it well suited for optimization purposes. First, the basic properties stated next are easily verified (see the appendix) and highlight why (5) is a sensible choice as an approximation of the max-function.

Lemma 1. *Let $G \in \mathbb{N}$ and $x \in \mathbb{R}^G$.*

(i) For $\zeta > 0$ it holds that

$$\max\{x_1, \dots, x_G\} \leq \operatorname{lse}_\zeta(x) \leq \frac{\log(G)}{\zeta} + \max\{x_1, \dots, x_G\}, \tag{6}$$

and in particular $\operatorname{lse}_\zeta(x) \searrow \max_g\{x_g\}$ as $\zeta \to \infty$.

(ii) For $\zeta \to 0$ it holds that

$$\operatorname{lse}_\zeta(x) = \frac{1}{G}\sum_{j=1}^{G} x_j + \frac{\log(G)}{\zeta} + o(1).$$

We define the soft maximin loss function, by

$$s_\zeta(\beta) := \operatorname{lse}_\zeta(-\hat{V}(\beta)), \quad \beta \in \mathbb{R}^p, \quad \zeta > 0,$$

where $\hat{V}(\beta) := (\hat{V}_1(\beta), \dots, \hat{V}_G(\beta))^\top$. For $\kappa > 0$ and $\zeta > 0$, the soft maximin estimator can now be defined by

$$\hat{\beta}^\kappa_{smm} := \arg\min_\beta \operatorname{lse}_\zeta(-\hat{V}(\beta)) \quad \text{s.t.} \quad \|\beta\|_1 \leq \kappa. \tag{7}$$

Using Lemma 1, it is possible to quantify the impact of the parameter ζ on the performance of the soft maximin estimator (7). The following result gives a bound on the maximum negative explained variance of the soft maximin estimator, using that of the theoretical maximin effect b^*.

Proposition 1. *Let $D = \max_g \|\hat{\Sigma}_g - \Sigma\|_\infty$ and $\delta = \max_g \|\boldsymbol{X}_g^\top \boldsymbol{\epsilon}_g / n_g\|_\infty$. For fixed $\zeta > 0$ and $\kappa > 0$, if $\kappa \geq \max_g \|b_g\|_1$,*

$$\max_g\{-V_{b_g}(\hat{\beta}^\kappa_{smm})\} \leq \max_g\{-V_{b_g}(b^*)\} + 6D\kappa^2 + 4\kappa\delta + \frac{\log(G)}{\zeta},$$

where b^* is the maximin effect. In particular

$$\|\hat{\beta}^{\kappa}_{smm} - b^*\|_\Sigma \leq 6\kappa^2 D + 4\kappa\delta + \frac{\log(G)}{\zeta}.$$

Proposition 1 is shown by combining Lemma 1 and results in Meinshausen and Bühlmann (2015), see the appendix. In particular, the performance loss incurred when using the soft maximin estimator is bounded by the same quantity as that of the maximin estimator plus the soft maximum approximation bias $\log(G)/\zeta$ from Lemma 1. Thus the soft maximin estimator enjoys theoretical properties similar to those of the (hard) maximin estimator, when controlling for the parameter ζ. In particular, for $D = 0$ (e.g., for a fixed design) and a fixed number of groups, if $n_g \to \infty$ for all g, the soft maximin estimator only retains the approximation bias.

Proposition 1 establishes a connection between the soft maximin performance and the maximin effect and shows that for $\zeta \uparrow \infty$ we indeed obtain the maximin estimator performance. However, it also highlights that for $\zeta \downarrow 0$ the performance of the soft maximin estimator can stray arbitrarily far away from that of the maximin estimator. To shed light on this note that by Lemma 1, for small $\zeta > 0$,

$$s_\zeta(\beta) \approx \frac{1}{G}\sum_{j=1}^{G} - \hat{V}_j(\beta) + \frac{\log(G)}{\zeta} \propto \frac{1}{n}\sum_{j=1}^{G}\sum_{i=1}^{n_j} \frac{n}{G n_j}((X_j\beta)_i - Y_{j,i})^2,$$

and (7) effectively becomes a penalized weighted least squares (PWLS) problem over all n observations. Thus solving (7) for a small $\zeta \geq 0$ approximately yields the pooled PWLS estimator with weights amplifying observations from smaller than average groups. With the same number of observations in each group, the soft maximin estimator in turn interpolates the pooled PLS estimator and the maximin estimator.

In this sense ζ reflects the heterogeneity in the data. If there is little heterogeneity a low or even zero ζ might work well corresponding to grouping not being relevant. However, for heterogeneous data a low ζ might lead to predictions that are worse than the zero prediction, whereas a high ζ can still work well.

To illustrate the interpolation, consider a small data example with data generated according to (2) with $G = 20$ groups, $n_g = 400$ observations in each group, and a two-dimensional (2D) parameter space. For fixed effects $\{b_1, \ldots, b_{20}\} \subset \mathbb{R}^2$ we sample $\boldsymbol{X}_g$ and $\boldsymbol{\epsilon}_g$, $M = 10$ times for each g resulting in 10 different datasets. For each of these ten small data sets we can compute the unpenalized softmaximin estimate (i.e., $\kappa = \infty$ in (7)) for a sequence of ζ values as well as the corresponding maximin estimate (i.e., $\lambda = 0$ in (23)) using base functionality in R.

Figure 2 displays the interpolation paths of the soft maximin estimator connecting the population LS estimates and the maximin estimates. Note that all maximin type estimates are clustered around the theoretical maxmin effect indicated with a ▲ on the edge of the convex hull of $\{b_1, \ldots, b_{20}\}$ while the pooled estimates are well inside the convex hull.

We note that very recently another regression method, anchor regression, was proposed in Rothenhäusler et al. (2021) to handle data heterogeneity in situations where the response distribution can shift yielding a difference between the training data distribution and test data distribution. If this heterogeneity can be encoded or generated by a known anchor variable their method can lead to improved and particularly more stable prediction performance. Especially by controlling for an anchor parameter $\gamma > 0$, this method interpolates three different regression methods where $\gamma = 1$ yields OLS regression and $\gamma = \infty$ an instrumental variabel regression.

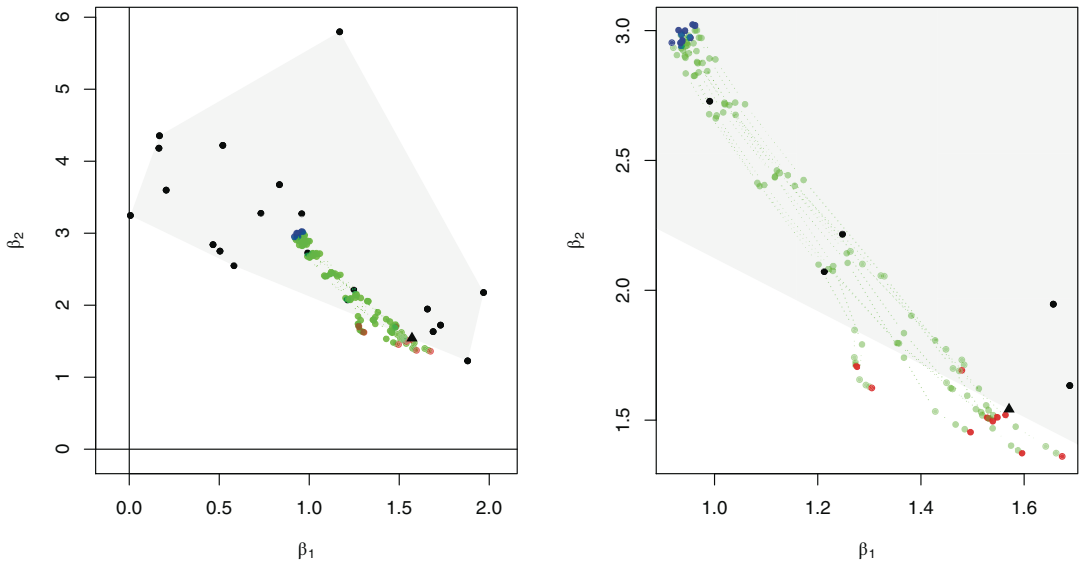

FIGURE 2 Left: Convex hull (grey shaded area) of supp(F_B) (black points). Theoretical maximin effect b^* (▲), maximin estimates (red points), soft maximin estimates (green points) for various ζ, and population OLS estimates (blue points). Right: Close up [Colour figure can be viewed at wileyonlinelibrary.com]

In some sense the anchor plays the role of the grouping structure $I_1, \ldots, I_G$ used in the definition of the soft maximin estimator. However we note that in a general setup with unknown groups, given certain structural assumptions, we may construct index sets $I_1, \ldots, I_G$, for example, as in the example above or by random sampling. Theoretical guarantees in this case are given in Meinshausen and Bühlmann (2015) for the maximin estimator and similarly to Proposition 1 should extend to the soft maximin estimator.

In addition to mathematical complexity, this adds a layer of substantial computational complexity to the inference procedure since the number of groups G is then a hyperparameter that needs to be inferred, for example, by cross-validation (CV). Thus this clustering layer only amplifies the importance of an efficiently computable base estimator.

3 | COMPUTATIONAL PROPERTIES

Here we shall consider a general estimation setup where the empirical explained variance $-\hat{V}_g$ from (4) is replaced by a general convex group divergence function, $h_g : \mathbb{R}^p \to \mathbb{R}$. Particularly, in parallel to the Bregman divergence let $\psi : \mathbb{R}^n \to \mathbb{R}$ be a convex function, and define

$$D_\psi(x, y) := \psi(x) - \nabla\psi(y)^\top x, \quad x, y \in \mathbb{R}^n.$$

The group divergence function can then be defined by

$$h_g(\beta) := D_\psi(\eta_g(\beta), Y_g), \quad \beta \in \mathbb{R}^p,$$

where $\eta_g(\beta) = \boldsymbol{X}_g\beta$ is the linear predictor in group g. Note that as ψ is convex it follows that D_ψ, like the Bregman divergence, is convex in its first argument and in particular h_g is convex. However, unlike the Bregman divergence D_ψ is not nonnegative.

A *general* soft maximin loss function $l_\zeta : \mathbb{R}^p \to \mathbb{R}$, is now given by

$$l_\zeta(\beta) := \mathrm{lse}_\zeta \circ h(\beta) = \frac{\log(\sum_{j=1}^G e^{\zeta h_j(\beta)})}{\zeta}, \quad \zeta > 0,$$

and our aim is to solve the general soft maximin problem formulated as

$$\min_{\beta\in\mathbb{R}^p} l_\zeta(\beta) + \lambda J(\beta), \quad \lambda \geq 0. \tag{8}$$

Here J is a proper convex penalty function and λ is the penalty parameter.

Choosing ψ as the square norm yields the negative empirical explained variance $-\hat{V}_g$ as group divergence, that is, $h = -\hat{V}$. If $J = \|\cdot\|_1$, both loss and penalty are convex in this case, and (8) is equivalent to the (constrained) soft maximin problem (7) by strong Lagrangian duality. Hence in this case the solution to (8) is exactly the soft maximin estimator.

We note that for a different choice of ψ we would obtain an entirely new estimator potentially with properties very different from those of the soft maximin estimator. An immediate Mahalanobis type generalization would arise if we let ψ be given as a weighted square norm.

Solving (8) in a large-scale setting requires an efficient optimization algorithm for nondifferentiable problems. In contrast to the hard maximin problem, (8) is, in addition to convex and nondifferentiable, a (partially) differentiable and separable problem (see Tseng & Yun, 2009). This means that a range of efficient algorithm will solve (8), for example, first-order operator splitting algorithms like ADMM or a second-order algorithm like coordinate descent. Here we are going to consider modified versions of the proximal gradient algorithm.

3.1 | Solution algorithm

The proximal gradient algorithm fundamentally works by iteratively applying the proximal operator

$$\mathrm{prox}_{\Delta J}(\beta) = \underset{\gamma\in\mathbb{R}^p}{\arg\min} \left\{ \frac{1}{2\Delta}\|\gamma - \beta\|_2^2 + J(\gamma) \right\}, \quad \Delta > 0, \tag{9}$$

to gradient-based proposal steps. For a loss function with a Lipschitz continuous gradient with constant L, such an algorithm is guaranteed to converge to the solution as long as $\Delta \in (0, 2/L)$, hence making it attractive to obtain the smallest possible Lipschitz constant L.

With known L and fixed $\Delta \in (0, 2/L)$ a proximal gradient algorithm essentially consists of the following steps:

(i) evaluate the gradient of the loss;
(ii) evaluate the proximal operator $\mathrm{prox}_{\Delta J}$;
(iii) evaluate the loss function and penalty function.

The computational complexity in steps i and iii is dominated by matrix-vector products, (see e.g., (4) for the soft maximin problem). The complexity in step ii is determined by J. As noted

in Beck and Teboulle (2009) when J is separable (e.g., the ℓ_1-norm) $\text{prox}_{\Delta J}$ can be computed analytically or at low cost.

If L is not known (or if $\Delta \geq 2/L$ for a known, but perhaps conservative, L) we cannot guarantee convergence with a fixed choice of Δ, but adding a backtracking step will ensure convergence of the iterates. This extra step will increase the per-step computational cost of the algorithm.

When the gradient is not globally Lipschitz, it is no longer guaranteed that iterating steps (i)–(iii) will yield a solution to (8) for any fixed Δ. However, we verify (Proposition 3) that the following nonmonotone proximal gradient (NPG) algorithm, see Wright et al. (2009) and Chen et al. (2016), will converge to a solution of (8) under some regularity conditions (Algorithm 1).

Algorithm 1. NPG minimizing $F = f + \lambda J$

Require: β^0, $L_{\max} \geq L_{\min} > 0$, $\tau > 1$, $c > 0$, $M \in \mathbb{N}$.

for $k = 0$ to $K \in \mathbb{N}$ **do**
  choose $L_k \in [L_{\min}, L_{\max}]$
  solve $\beta = \text{prox}_{\lambda J/L_k}(\beta^{(k)} - \frac{1}{L_k}\nabla f(\beta^{(k)}))$
  if $F(\beta) \leq \max_{[k-M]_+ \leq i \leq k} F(\beta^{(i)}) - c/2\|\beta - \beta^{(k)}\|^2$ **then**
    $\beta^{(k+1)} = \beta$
  else
    $L_k = \tau L_k$ and go to 3
  end if
end for

In particular, we show that while l_ζ does not have a Lipschitz continuous gradient in general, convergence of the NPG algorithm is still guaranteed under general conditions on the group functions $h_1, \ldots, h_G$. Furthermore, in the special case where $h_g = -\hat{V}_g$ with all groups sharing the same design we show that l_ζ has a globally Lipschitz continuous gradient, and we derive a Lipschitz constant.

The first result states that l_ζ inherits strong convexity from any individual group divergence function h_g given that all $h_1, \ldots, h_G$ are convex and twice continuously differentiable. The proof is given in the appendix.

Proposition 2. *For $g \in \{1, \ldots, G\}$ assume h_g is twice continuously differentiable and let $w_{g,\zeta}(\beta) := e^{\zeta h_g(\beta) - \zeta l_\zeta(\beta)}$, $\beta \in \mathbb{R}^p$. Then $(w_{j,\zeta}(\beta))_j$ are convex weights and*

$$\nabla l_\zeta(\beta) = \sum_{j=1}^{G} w_{j,\zeta}(\beta)\nabla h_j(\beta). \tag{10}$$

$$\nabla^2 l_\zeta(\beta) = \sum_{i=1}^{G}\sum_{j=i+1}^{G} w_{i,\zeta}(\beta)w_{j,\zeta}(\beta)(\nabla h_i(\beta) - \nabla h_j(\beta))(\nabla h_i(\beta) - \nabla h_j(\beta))^\top + \sum_{j=1}^{G} w_{j,\zeta}(\beta)\nabla^2 h_j(\beta). \tag{11}$$

Furthermore if $h_1, \ldots, h_G$ are convex with at least one h_g strongly convex, then l_ζ and $e^{\zeta l_\zeta}$ are strongly convex.

Proposition 2 applies to the soft maximin loss with $h_g = -\hat{V}_g$. In this case $\nabla^2 h_g = 2\boldsymbol{X}_g^\top \boldsymbol{X}_g / n_g$, and h_g is strongly convex if and only if $\boldsymbol{X}_g$ has rank p. Proposition 2 implies that if one of the matrices $\boldsymbol{X}_g$ has rank p, l_ζ is strongly convex. However, we also see from Proposition 2 that $\nabla^2 l_\zeta(\beta)$ is not globally bounded in general. Consider the soft maximin loss, for instance, with $G = 2$, $p = n_1 = n_2 = 2$ and

$$\boldsymbol{X}_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \text{and} \quad \boldsymbol{X}_2 = \begin{pmatrix} 0 & 0 \\ \sqrt{2} & 0 \end{pmatrix}. \tag{12}$$

Take also $y_1 = y_2 = 0$. When $\beta_1 = \beta_2 = r \in \mathbb{R}$ it holds that $h_1(\beta) = h_2(\beta) = r^2$ and thus $w_{1,\zeta} = w_{2,\zeta} = 1/2$ for any ζ, while

$$(\nabla h_1(\beta) - \nabla h_2(\beta))(\nabla h_1(\beta) - \nabla h_2(\beta))^\top = \begin{pmatrix} \beta_1^2 & -\beta_1\beta_2 \\ -\beta_1\beta_2 & \beta_2^2 \end{pmatrix} = \begin{pmatrix} r^2 & -r^2 \\ -r^2 & r^2 \end{pmatrix},$$

is unbounded. In turn the gradient is not Lipschitz.

The following result shows, on the other hand, that for soft maximin estimation with identical $\boldsymbol{X}_g$-matrices across the groups, ∇l_ζ is, in fact, Lipschitz continuous. The proof is in the appendix.

Corollary 1. *For each $g \in \{1, \dots, G\}$ let $\boldsymbol{X}_g = \boldsymbol{X}$ be a $m \times p$ matrix, $\boldsymbol{Y}_g$ a $m \times 1$ vector and the group divergence given by*

$$h_g(\beta) = \frac{1}{m}(2\beta^\top \boldsymbol{X}^\top \boldsymbol{Y}_g - \beta^\top \boldsymbol{X}^\top \boldsymbol{X}\beta).$$

Then ∇l_ζ has Lipschitz constant L bounded by

$$\frac{4}{m^2}\max_{i,j}\|\boldsymbol{X}^\top(\boldsymbol{Y}_i - \boldsymbol{Y}_j)\|^2 + \frac{2|||\boldsymbol{X}^\top\boldsymbol{X}|||}{m} \le \frac{4|||\boldsymbol{X}^\top\boldsymbol{X}|||}{m^2}\left(\max_{i,j}\|\boldsymbol{Y}_i - \boldsymbol{Y}_j\|^2 + \frac{m}{2}\right), \tag{13}$$

with $|||\cdot|||$ the matrix norm induced by the 2-norm $\|\cdot\|$.

By Corollary 1 if we have identical designs across groups we can obtain the soft maximin estimator by applying the fast proximal gradient algorithm from Beck and Teboulle (2009) to the optimization problem (8). Furthermore in this setting the corollary gives an explicit expression for the Lipschitz constant that will yield an efficient step size Δ for the solution algorithm.

Finally, in the general setup the following proposition shows that Algorithm 1, which does not rely on a global Lipschitz property (Chen et al., 2016), solves the problem (8) given the assumptions in Proposition 2. The proof of the proposition is given in the appendix.

Proposition 3. *Assume $h_1, \dots, h_G$ satisfy the assumptions in Proposition 2. Let $(\beta^{(k)})_k$ be a sequence of iterates obtained by applying the NPG algorithm to (8). Then $\beta^{(k)} \to \beta^*$ where β^* is a critical point of $l_\zeta + \lambda J$.*

In summary given a strongly convex group divergence function, for example, satisfied in the soft maximin setup when one $\boldsymbol{X}_g$ has full rank, we can always solve the general problem (8) using a proximal gradient based algorithm.

3.2 | Array tensor smoothing

We shall here briefly discuss an important special case, array tensor smoothing, where the design is fixed and identical across groups and the objective is to extract a common signal understood as a smooth function over a possibly multidimensional domain. Typically the scale of the data in this setting will prevent the (hard) maximin estimator from being applicable.

By array data we mean data with a geometry such that it is most naturally organized in a multidimensional array, as opposed to an unstructured vector. Canonical examples are images, movies, etc.. To formalize this data setting and model consider a regular grid in $\mathbb{R}^d$, that is, a d-dimensional lattice

$$\mathcal{Z}_1 \times \mathcal{Z}_2 \times \ldots \times \mathcal{Z}_d, \tag{14}$$

where $\mathcal{Z}_j = \{z_{j,1}, \ldots, z_{j,m_j}\} \subset \mathbb{R}$ with $m_j \in \mathbb{N}, j \in \{1, \ldots, d\}$. Let $m := \prod_{j=1}^d m_j$. If for each group g data $y_{g,1}, \ldots, y_{g,m}$ is sampled across all points in (14) it may be organized in a fully populated d-dimensional $m_1 \times \cdots \times m_d$-array

$$\mathbf{Y}_g = (y_{g,i_1,\ldots,i_d})_{i_1,\ldots,i_d}, \quad i_j = 1, \ldots, m_j,\ j = 1, \ldots, d. \tag{15}$$

For this reason we refer to this type of data as array data.

Preserving the array structure when formulating a multivariate smoothing model in this data setting leads to the array model equation

$$Y_{g,i_1,\ldots,i_d} = f_g(z_{1,i_1}, \ldots, z_{d,i_d}) + \varepsilon_{g,i_1,\ldots,i_d}, \quad z_{j,i_j} \in \mathcal{Z}_j, \tag{16}$$

for each $g \in \{1, \ldots, G\}$, where f_g is a smooth group signal and $\varepsilon_{g,i_1,\ldots,i_d}$ an appropriate error term.

To obtain a linear array model from (16) we parameterize f_g using a basis expansion. A particularly convenient way of representing a multivariate function, is to use the tensor product construction to specify multivariate basis functions in terms of (tensor) products of families of univariate basis functions $((\varphi_{j,k})_{k=1}^{\infty})_{j=1}^d$. That is with $\varphi_{j,k} : \mathbb{R} \to \mathbb{R}$ the kth basis function in the jth dimension (kth j-marginal basis function), we can represent the smooth signal as

$$f_g(z_1, \ldots, z_d) = \sum_{k_1,\ldots,k_d} \Theta_{g,k_1,\ldots,k_d} \prod_{j=1}^d \varphi_{j,k_j}(z_j), \quad z_j \in \mathbb{R}, \tag{17}$$

where $(\Theta_{g,k_1,\ldots,k_d})_{k_1,\ldots,k_d}$ are basis coefficients.

Now, in order to implement this representation for the model (16) we need to determine the number of basis functions to use in the jth dimension, $p_j \in \mathbb{N}$, to obtain an (finite) approximation of f_g. For tensor product basis functions it is customary to choose p_j as a function of the cardinality of $\mathcal{Z}_j$, for example, $p_j = [m_j/5]$ (see Currie et al., 2006).

With $p_j \in \mathbb{N}$ fixed for each $j \in \{1, \ldots, d\}$ we obtain the model (16) as a linear array model in the following way. For each $j \in \{1, \ldots, d\}$ define a $m_j \times p_j$ matrix $\Phi_j = (\varphi_{j,k}(z_{j,i}))_{i,k}$ containing the values of the p_j basis functions evaluated at the m_j points in $\mathcal{Z}_j$. We call Φ_j a marginal design matrix. Also define a $p_1 \times \cdots \times p_d$-array $\mathbf{\Theta}_g = (\Theta_{g,j_1,\ldots,j_d})_{j_1=1,\ldots,j_d=1}^{p_1,\ldots,p_d}$ containing the corresponding basis coefficients.

It then follows directly from the identity (17) that the tensor (Kronecker) product of these marginal design matrices,

$$\Phi = \Phi_d \otimes \cdots \otimes \Phi_2 \otimes \Phi_1, \tag{18}$$

is the design matrix for the linear model version of (16). This means that we can in principle implement (16) as a standard linear model using Φ. However from a computational complexity perspective that is suboptimal and potentially not feasible in large scale data settings as Φ grows as $\prod p_j \prod m_j$.

Instead, we can exploit the array tensor structure of the problem and only rely on the much smaller marginal matrices. As shown in Currie et al. (2006) any linear model with array structured data (15) and tensor structured design (18) can be formulated as a linear array model and fitted using so-called array arithmetic. The key computation is the rotated H-transform ρ, (see Currie et al., 2006 for details), that allows us to write the model (16) as a linear array model

$$\mathbf{Y}_g = \rho(\Phi_d, \rho(\Phi_{d-1}, \ \dots \ , \rho(\Phi_1, \mathbf{\Theta}_g))) + \boldsymbol{\varepsilon}_g, \tag{19}$$

where $\boldsymbol{\varepsilon}_g$ is a $m_1 \times \cdots \times m_d$ array containing the error terms.

As indicated by (19), using ρ the design matrix-parameter vector products, needed in steps i and iii above, are computed without having access to the (very large) matrix Φ. In addition the computation has lower complexity than the corresponding matrix-vector product (Buis & Dyksen, 1996; De Boor, 1979).

Finally, the tensor structure in (18) makes the constant L in Corollary 1 easy to compute, see (30) in Lund et al. (2017). The implication is that we can run the proximal gradient algorithm without performing any backtracking.

Following Lund et al. (2017) we have implemented both the fast proximal algorithm as well as the NPG algorithm 1 in a way that exploits the array-tensor structure described above. Implementations are provided for one-, two-, and three-dimensional array data in the R package `SMME`, Lund (2021) along with the implementation for general unstructured data.

4 | NUMERICAL EXPERIMENTS

To demonstrate the properties of the soft maximin estimator we present two data examples. The first example is a real dataset, also analyzed in Rothenhäusler et al. (2021), with seemingly a high signal to noise ratio and moderate data heterogeneity. We use this data set to highlight the interpolation property inherent in the soft maximin methodology.

In the second example we use a simulated large-scale data set with low signal to noise ratio and strong heterogeneity to benchmark our methodology. We compare the run time and prediction accuracy of the soft maximin estimator to that of the pooled OLS estimator and the maximin aggregation method, see Bühlmann and Meinshausen (2016). This simulation based example is inspired by a large scale neuronal dataset analyzed in Appendix B.

4.1 | Washington DC bike sharing data

The data used to produce the results in Figure 1 are described in Fanaee-T and Gama (2013). The dataset contains 2 years (2011 and 2012) of data (variable `cnt` see Figure 3) from a bike sharing

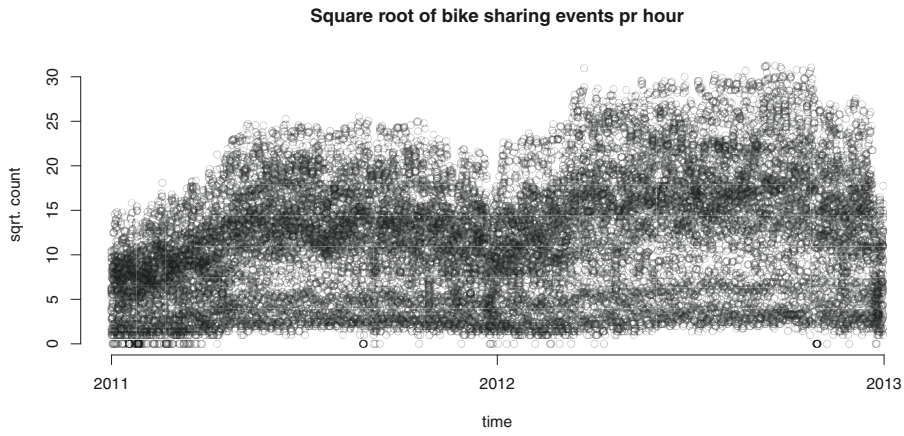

FIGURE 3 Square root of hourly number of bike shares in Washington DC 2011 and 2012

scheme in Washington DC along with auxiliary data presumably relevant for bike usage such as weather. We model the hourly number of bike shares from the dataset shown in Figure 3. Specifically the square root of the number of bike events (`cnt`) is modeled as a smooth function of hour of day (`hr` with values 0 to 23), smooth function of day of week (`weekday` with values 0 to 6) and the weather situation (`weathersit` with levels 1,2,3). The model equation for observation i can be written as

$$\sqrt{\texttt{cnt}_i} = \sum_{j=1}^{10} \alpha_j \varphi_j(\texttt{hr}_i) + \sum_{j=1}^{5} \beta_j \varphi_j(\texttt{weekday}_i) + \sum_{j=1}^{3} \gamma_j 1_j(\texttt{weathersit}_i) + \varepsilon_i, \tag{20}$$

where φ are cubic basis spline functions.

Note the raw data do not contain any hours with zero counts. However as there are 165 hours unaccounted for in the dataset, we suspect these are hours with zero counts (e.g., during hurricane Sandy in October 2012) and impute this data for simplicity. This imputation has no effect on the analysis. Also there are three observations that have `weathersit` $= 4$. To have all `weathersit` levels present for the relevant test and train split we change the level of these observations to 3. Again this change has no effect relative to leaving out the observations entirely.

The data do not a priori have a grouping structure that explains the heterogeneity. However, a safeguard against temporal heterogeneity may be obtained by using the variable `mnth` to group the data in months. Using this grouping, to show the effect of the temporal heterogeneity, we perform a simple experiment where we (i) fit the model on 2011 data and predict on 2012 data and (ii) fit the model on 2012 data and predict on 2011 data.

We fit the model (20) to data using the unpenalized soft maximin estimator ($\lambda = 0$ in (8)) and the pooled OLS estimator. We also fit an extension of (20) that includes a smooth function of the grouping variable `mnth` using OLS. This model potentially controls for the heterogeneity and might have superior performance. Note the aim of the analysis is not to obtain the model with best predictive performance but rather to highlight how maximin type estimators can safeguard against heterogeneity. In settings where the source of structured variation is more opaque and cannot be taken into account easily in the model this idea may still work.

Figure 1 shows the cumulative root mean squared error

$$\mathrm{RMSE}_\zeta(t) = \sqrt{\frac{1}{t}\sum_{i=1}^{t}(y_i - \hat{y}_{i,\zeta})^2}, \tag{21}$$

on both training data and test data for the model fit for, respectively, (i) and (ii). Here $y = \sqrt{\texttt{cnt}}$ and $\hat{y}_\zeta$, the fitted values for resp. pooling ($\zeta = 0$) and soft maximin ($\zeta > 0$), are chronologically ordered.

The findings in Figure 1 are in line with Figure 4 that shows the predictions (out of sample) for 5 consecutive days for experiments (i) and (ii). In the left panel we observe that the high ζ soft maximin (red) underfits the 2012 data while the low ζ soft maximin like pooling predicts well. Conversely in the right panel the low ζ and pooling overfits the high levels observed in some parts of the 2012 data and the result is worse predictions on the 2011 data.

The bottom panel in Figure 4 shows the deciles of RMSEs for the test set (2012 resp. 2011) averaged for each week as a function of ζ. Training on 2011 data and testing on 2012 data yields weekly averaged prediction errors for the pooling estimator with lower median and more narrow support compared to those of the soft maximin estimators. Conversely training on 2012 data and testing on 2011 a strictly positive ζ yields more stable predictions and also a lower median. Note Figure 4 resembles Figure 5 in Rothenhäusler et al. (2021).

In Appendix C we fit alternative models that include temperature and humidity. The results are similar to those presented above. We also fit a model where we use `weathersit` as grouping. For this grouping the soft maximin estimators still appear attractive but the picture is less clear.

Also in the appendix a CV scheme is used to tune ζ for experiments (i) and (ii). The results are in line with the findings in Figures 1 and 4 and suggest that hard maximin is not optimal in neither (i) nor (ii). For (i) pooling ($\zeta = 0$) works best and for (ii) a ζ around 0.03 results in the lowest prediction error. Fitting a spectrum of estimators can be advantageous compared to fitting either of the extreme estimators, pooling respectively maximin, in the given context.

4.2 | Benchmark on simulated array data

To benchmark the soft maximin method against existing alternatives we set up a prediction experiment for the model described in Section 3.2 on a simulated dataset. In Appendix B we carry out the same experiment on a real large-scale neuroscientific dataset. The experiment is structured like K-fold cross-validation aimed at discerning the optimal choice of hyperparameters ζ and λ in a grid search.

To evaluate the performance of the soft maximin estimator as a function of the parameter ζ we train the model for $\zeta \in \{2{,}100, 200\}$. We also compute the pooled estimator corresponding to $\zeta = 0$ and the maximin aggregation (magging) estimator from Bühlmann and Meinshausen (2016). In general magging is approximately (hard) maximin estimation and should therefore correspond to $\zeta = \infty$.

This in turn entails solving five different ℓ_1-penalized estimation problems:

- To obtain the three soft maximin estimators we need to solve the problem (8) with ℓ_1-norm penalty for each $\zeta \in \{2{,}100, 200\}$. To do this we use the R package `SMME`, Lund (2021).

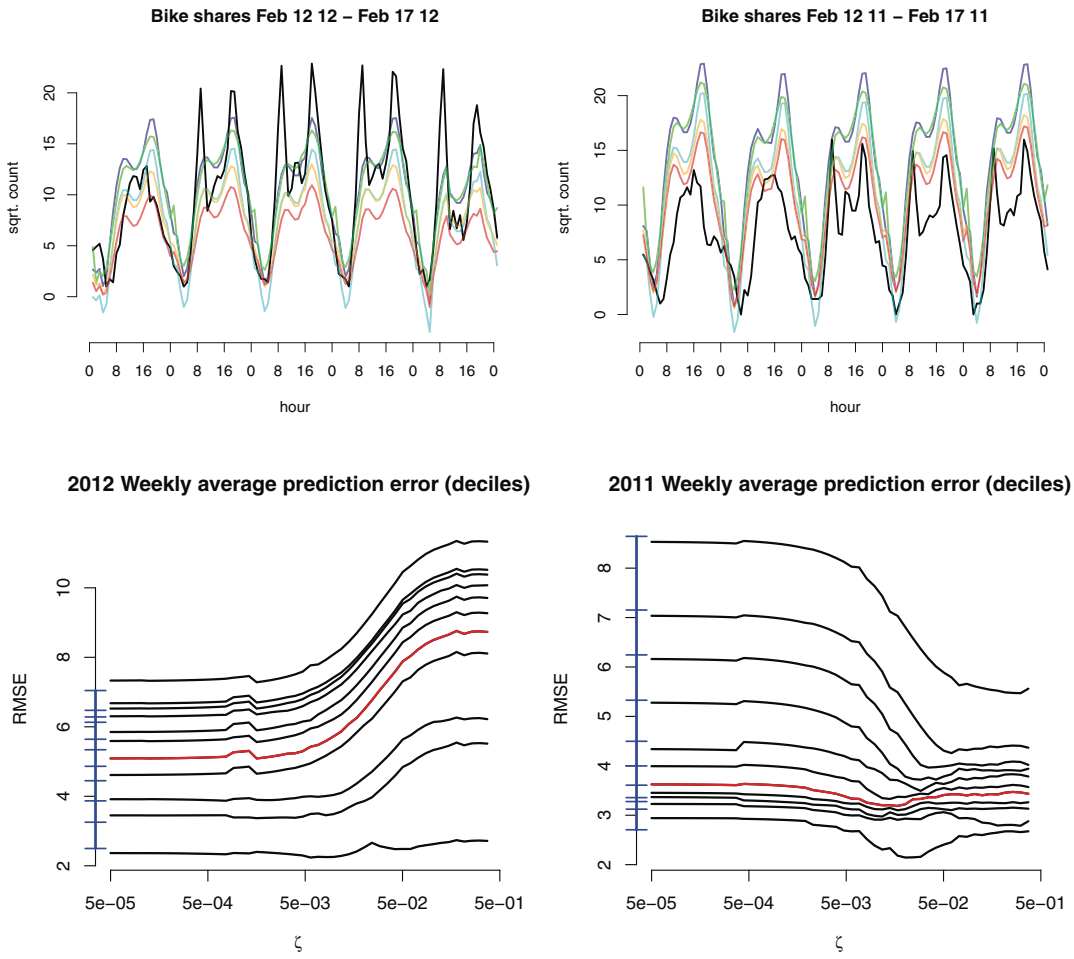

FIGURE 4 Top: Predictions for 2012 when training on 2011 (left) and for 2011 when training on 2012 (right), for the soft maximin estimator with $\zeta \in \{0.0001, 0.01, 1\}$ (resp. green, orange and red), the pooled OLS estimator (blue and cyan) and $\sqrt{\texttt{cnt}}$ (black). Bottom: Deciles of weekly averaged prediction error on 2012 data for 2011 training data (left) and on 2011 data for 2012 training data (right) for soft maximin estimation as a function of ζ. Blue part of the y-axis shows deciles of the pooled estimator. Red curve indicates the median [Colour figure can be viewed at wileyonlinelibrary.com]

- With identical (fixed) design across groups, we obtain the (penalized) pooled estimator as (penalized) regression of the empirical average across groups on the fixed design. We use the R package `glamlasso`, Lund (2018) to solve the resulting lasso problem.
- For the magging estimator we have to solve a lasso problem for each group, given λ and the design. We use the R package `glamlasso`, Lund (2018) to obtain the individual group fits. These fits are then maximin aggregated (magging) across groups by solving a quadratic optimization problem as proposed in Bühlmann and Meinshausen (2016).

All computations are carried out on a Macbook Pro with a 2.8 GHz Intel core i7 processor and 16 GB of 1600 MHz DDR3 memory.

4.2.1 | Simulated array data

We simulate data with three components: (i) a common Gaussian signal of interest $\varphi(x, y, t) = 200\varphi_{12.5,4}(x)\varphi_{12.5,4}(y)\varphi_{50,25}(t)$ (φ_{μ,σ^2} is the density for the $\mathcal{N}(\mu, \sigma^2)$ distribution) superimposed with (ii) periodic group specific signals with randomly varying frequency and phase, and (iii) additive white noise. Specifically for each $g \in \{1, \dots, G\}$ the 3D data array was simulated according to

$$Y_{g,i,j,k} = \varphi(x_i, y_j, t_k) + 5\sum_{j \in J_g} \varphi_j(x_i + p_g)\varphi_j(y_i + p_g)\varphi_j(t_k + p_g) + \varepsilon_{g,i,j,k}, \tag{22}$$

with $x_i = 1, 2, \dots, 25$, $y_i = 1, 2, \dots, 25$, and $t_i = 1, 2, \dots, 101$. Here J_g is a set of seven integers sampled uniformly from $\{1, \dots, 101\}$, φ_j is the jth Fourier basis function, $p_g \sim \text{unif}(-\pi, \pi)$, and $\varepsilon_{g,i,j,k} \sim \mathcal{N}(0, 10)$.

We note that the common 3D Gaussian signal φ, due to its light tails, is spatially as well as temporally localized.

Figure 5 shows the simulated signals for three different groups plotted across time for $(i, j) = (12, 12)$. The common signal is dominated by group-specific fluctuations in each group and not visually apparent.

4.2.2 | Experiment setup

We use the array-tensor model from Section 3.2 with B-splines as basis functions in each dimension. The number of basis functions used in each dimension is, respectively, $p_1 = p_2 = 10$ (spatial) and $p_3 = 23$ (temporal). This gives us an array model with marginal design matrices Φ_1, Φ_2, and Φ_3, of size 25×10, 25×10, and 101×23, respectively, given by B-spline function evaluations over the marginal domains. The model has $p = 2300$ parameters.

To set up the experiment we simulate $G = 100$ groups of 3D signals according to (22). We then randomly sample $K = 7$ folds (14 groups in each fold). This gives us a total of $m = 883{,}750$ observations in each fold.

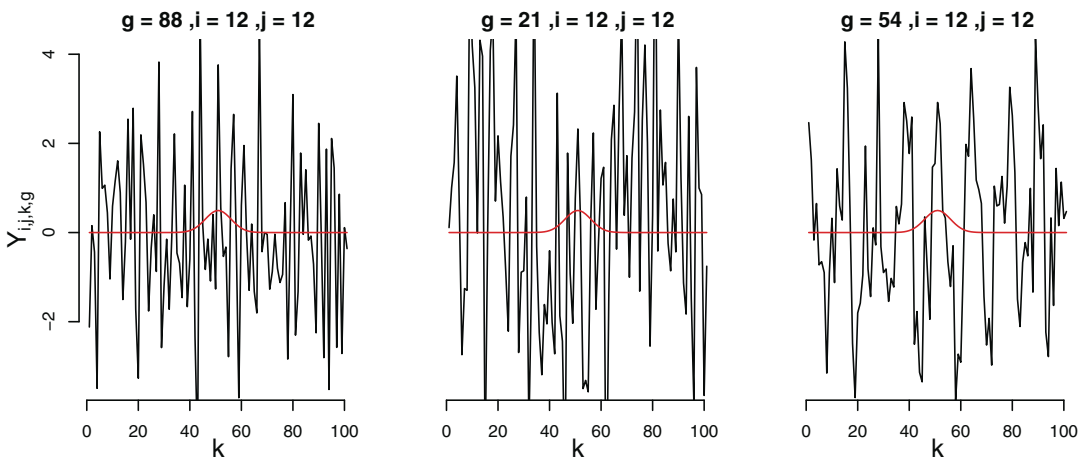

FIGURE 5 Temporal plot of the simulated data (black) for three different groups and the underlying common Gaussian signal (red) [Colour figure can be viewed at wileyonlinelibrary.com]

For each method we train the model on each fold and test on the remaining six folds. By repeating this procedure $N = 10$ times we obtain 70 fits and corresponding test metrics for each method and each setting of λ.

4.2.3 | Experimental results

Predictions for the experiment for four different validation sets are shown in Figure 6. For one dataset the pooled estimator is successful in extracting a clear Gaussian signal but it fails on the other three. In contrast the soft maximin estimators ($\zeta \in \{100,200\}$) perform well on all sets and display less variation.

Figure 7 shows the average of the root mean squared prediction error (RMSPE) as a function of model complexity (λ) for each method. RMSPE is defined as $\|\hat{Y}_{x,s} - Y_{-s}\|_2/\sqrt{m}$ where $\hat{Y}_{x,s}$ is the prediction from training on set s using method x and Y_{-s} are the observations in the complement to s.

The dashed line represents the average error made when using the zero signal, the most conservative estimate. The black line on the other hand is the average error when using the true signal as predicted values and is the optimal prediction for this data. We see the best performing method is the soft maximin with $\zeta = 200$ (red) while the worst is the pooled estimator (blue). In particular, the pooled estimator performs no better than the zero prediction. Surprisingly the magging estimator (yellow) does not perform much better than the zero prediction on average.

The right display in Figure 7 illustrates the variability in prediction accuracy for the different methods using their relative deviation in RMSE from that of the zero prediction. The low ζ estimators (pooled and $\zeta = 2$) display high variability, reflecting a tendency to overfit group-specific signals in the data. The high ζ soft maximin estimators and magging show much less variability. This underlines the robustness of the estimation methodology.

We note that only the high ζ methods succeed in extracting a common signal that is significantly more accurate than the zero prediction.

We note that while the gain in prediction performance, relative to the zero prediction is small, due to the low signal to noise ratio, it is not insignificant in terms of the quality of the extracted signal as illustrated in Figure 6. Here, the fit on four different training sets are displayed, and we observe directly how the low ζ methods tend to fit group fluctuation compared to the high ζ methods.

We quantify this by computing the average of the root mean squared signal error $\|\hat{Y}_{x,s} - \varphi\|_2/\sqrt{m}$ made by the prediction obtained by training on set s using method x. The left display in Figure 8 shows the result for each method x and set s as a function of model complexity as well as the average over s (folds) and confirms the impression from Figures 6 and 7.

Finally, the right display in Figure 8 shows how each method performs in terms of run time. In this setting given the average response across the 14 groups in a fold, computing the pooled estimator has the same complexity as computing one group fit in the magging procedure. Not surprisingly the pooled estimator (0.8 s) is roughly 12 times faster than magging (9.7 s), and also faster than the high ζ methods (8.1 s resp. 16.6 s). However, the soft maximin estimator with $\zeta = 2$ (0.5 s) is faster than the pooled least square estimator.

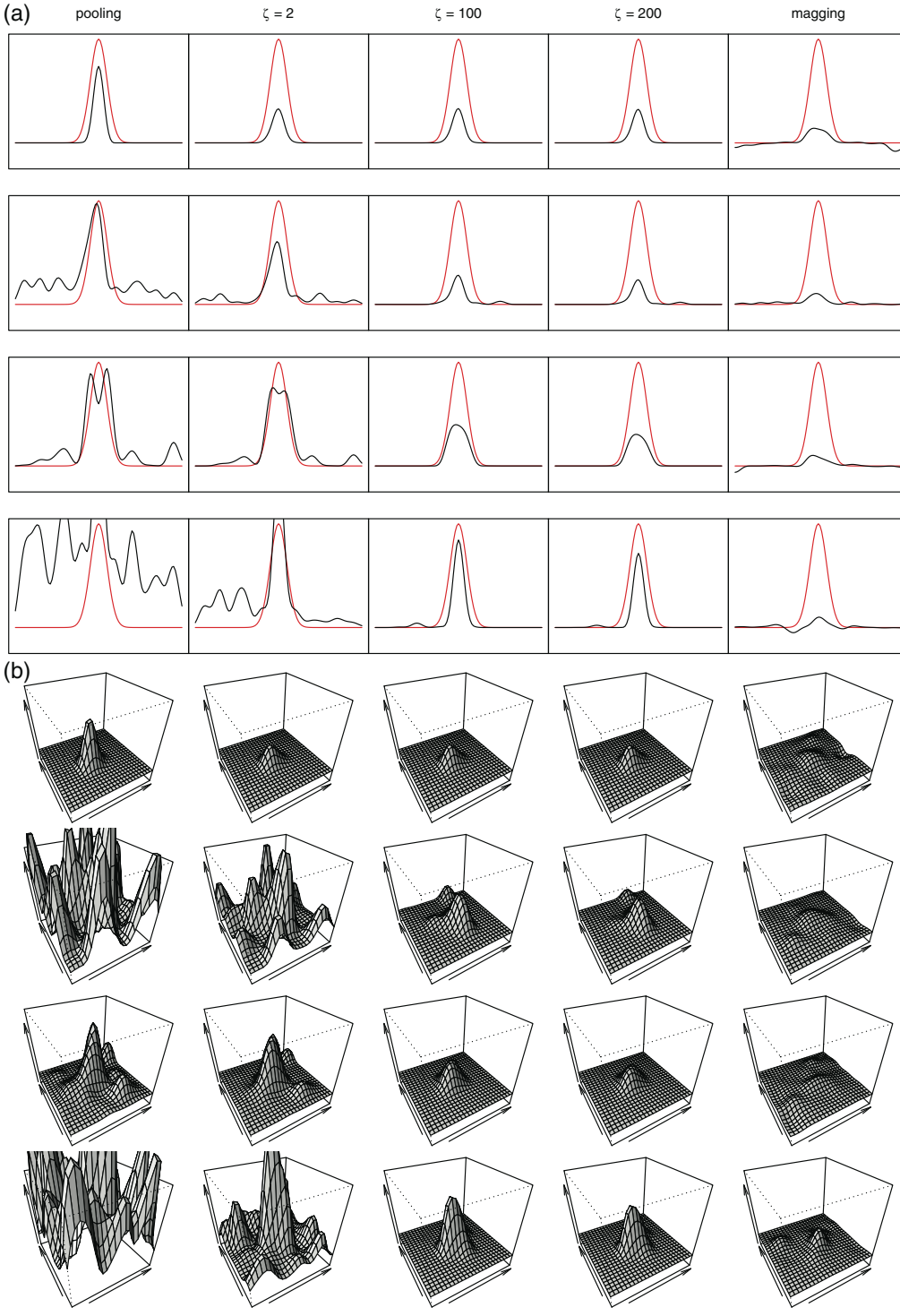

FIGURE 6 (a) Temporal plots for $(x, y) = (12, 12)$. True signal $\varphi(x, y, t)$ (red) and estimate $\hat{Y}_{x,y,t}$ (black) for model no. 10. Columns left to right; pooling, soft maximin $\zeta \in \{2{,}100, 200\}$ and magging. Each row corresponds to a training set with 14 groups. (b) Spatial plots for $t = 50$ of the estimates from panel a [Colour figure can be viewed at wileyonlinelibrary.com]

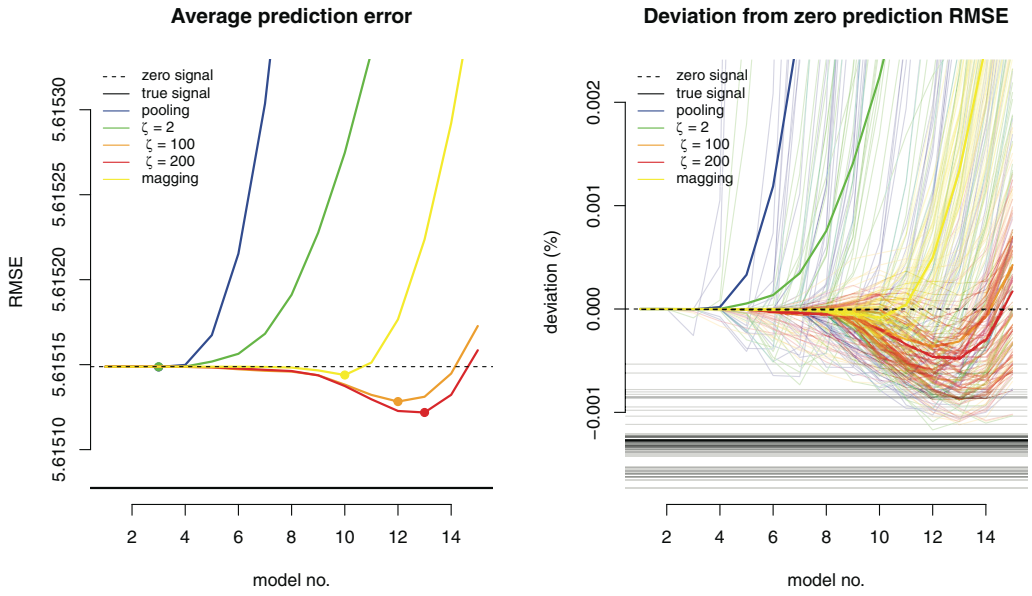

FIGURE 7 Left: Average RMSE across 70 test sets as a function of model complexity (model no.). Zero signal (dashed), true signal (black), pooled estimator (blue), soft maximin $\zeta = 2$ (green), soft maximin $\zeta = 100$ (orange), soft maximin $\zeta = 200$ (red) and magging (yellow). The minimum is indicated with a bullet. Right: Deviation in RMSE relative to the zero prediction RMSE (percentage difference) for each method and on each test set (thin lines), as a function of model complexity. The corresponding averages are indicated with a thick line of the same color [Colour figure can be viewed at wileyonlinelibrary.com]

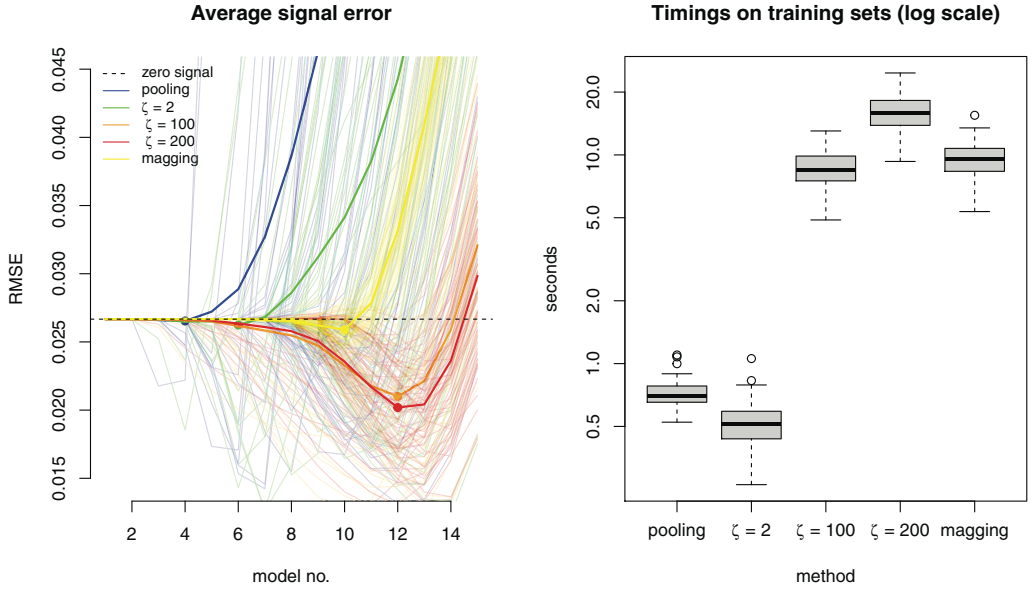

FIGURE 8 Left: Mean squared signal error based on each of the 70 training set. Pooled estimator (blue), soft maximin $\zeta = 2$ (green), soft maximin $\zeta = 100$ (orange), soft maximin $\zeta = 200$ (red) and magging (yellow). Right: Summary of run times (log scale) for the 70 training sets for each method [Colour figure can be viewed at wileyonlinelibrary.com]

5 | DISCUSSION

The maximin estimator with the ℓ_1-penalty, as defined in Meinshausen and Bühlmann (2015), solves the minimization problem

$$\hat{\beta}_{mm} := \arg\min_{\beta} \max_{g} \{-\hat{V}_g(\beta)\} + \lambda \|\beta\|_1. \tag{23}$$

Though the objective function is convex, it is nondifferentiable as well as nonseparable, and contrary to the claim in section 4 of Meinshausen and Bühlmann (2015), coordinate descent will not always solve (23), see Tseng and Yun (2009).

Two approximate approaches for solving (23) were suggested in Meinshausen and Bühlmann (2015). The first, also a smooth approximation of the term $\max_g\{-\hat{V}_g(\beta)\}$, however, appears theoretically invalid and we did not find it to work in practice either. The second approximation, the maximal penalty solution, obtains a solution to (23) for the maximum λ that yields a nonzero solution (at least one active feature) to (23). This solution is appropriate as an efficient initial estimator and clearly much cruder than a finely tuned maximin (type) estimator.

We note in passing that the solution path of (23) is piecewise linear in λ, and it may thus be computed using a method like LARS, see Roll (2008). A LARS-type algorithm or a coordinate descent algorithm of a smooth majorant, such as the soft maximin loss, has subsequently also been proposed to us by Meinshausen (personal communication) as better alternatives to those suggested in Meinshausen and Bühlmann (2015). In our experience, the LARS-type algorithm scales poorly with the size of the problem, and neither LARS nor coordinate descent can exploit the array-tensor structure.

We have developed soft maximin estimation as an alternative to maximin estimation that retains desirable statistical properties and is computationally more efficient. Furthermore the soft maximin parameter ζ controls the tradeoff between groups with large explained variance and groups with small explained variance leading to an interpolation of pooled estimation and maximin estimation. The gradient representation (10) shows explicitly how this tradeoff works: the gradient of the soft maximin loss is a convex combination of the gradients of the group-wise loss functions with weights controlled by ζ. The largest weights are on those groups with the smallest explained variances and as $\zeta \to \infty$ the weights concentrate on the groups with minimal explained variance.

On the bike sharing data we have illustrated this interpolation property and the benefit of having a spectrum of estimators available instead of only the extremes (pooling resp. maximin). Notably, the optimal value of ζ in terms of prediction accuracy depends on the data context. Specifically we conclude that for heterogeneous data (hard) maximin estimator is not necessarily the best choice. Instead by tuning ζ, for example, by cross-validation it might be possible to obtain an estimator in the spectrum between hard maximin and pooling that works better for the specific context. We note that a similar interpolation idea for heterogenous data has recently been proposed in the so called anchor regression framework, see Rothenhäusler et al. (2021).

On the simulated data (see also the neuronal VSDI data in Appendix B) it was demonstrated how the soft maximin estimator was able to extract a signal, and how the choice of the tuning parameter ζ affects the extracted signal and the prediction performance. In particular the simulations showed that soft maximin estimation ($\zeta \in \{100,200\}$) was able to extract a signal even in the presence of large heterogeneous noise components where the other methods (pooling and magging) failed.

In addition, we note that magging, proposed in Bühlmann and Meinshausen (2016) as a computationally attractive and generic alternative to (23) for estimation of maximin effects, in our numerical experiment, is not faster than using the soft maximin estimator.

In summary our proposed algorithm provides a means for approximately minimizing (23) and is as such an alternative to magging as an estimator of the maximin effect. More importantly, by the introduction of the tuning parameter ζ in the soft maximin loss we not only achieved an approximate solution of (23) but an interpolation between the (hard) maximin estimator and the pooled WLS estimator.

We expect that soft maximin estimation will be practically useful in a number of different contexts, as a way of aggregating explained variances across groups. In particular because it down-weights groups with a large explained variance that might simply be outliers, while it does not go to the extreme of the maximin effect, that can kill the signal completely.

ORCID

Adam Lund https://orcid.org/0000-0001-9033-5047
Søren Wengel Mogensen https://orcid.org/0000-0002-6652-155X
Niels Richard Hansen https://orcid.org/0000-0003-3883-365X

REFERENCES

Beck, A., & Teboulle, M. (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, *2*(1), 183–202.

Bühlmann, P., & Meinshausen, N. (2016). Magging: Maximin aggregation for inhomogeneous large-scale data. *Proceedings of the IEEE*, *104*(1), 126–135.

Buis, P. E., & Dyksen, W. R. (1996). Efficient vector and parallel manipulation of tensor products. *ACM Transactions on Mathematical Software (TOMS)*, *22*(1), 18–23.

Chen, X., Lu, Z., & Pong, T. K. (2016). Penalty methods for a class of non-lipschitz optimization problems. *SIAM Journal on Optimization*, *26*(3), 1465–1492.

Currie, I. D., Durban, M., & Eilers, P. H. (2006). Generalized linear array models with applications to multidimensional smoothing. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *68*(2), 259–280.

De Boor, C. (1979). Efficient computer manipulation of tensor products. *ACM Transactions on Mathematical Software (TOMS)*, *5*(2), 173–182.

Fanaee-T, H., & Gama, J. (2013). Event labeling combining ensemble detectors and background knowledge. *Progress in Artificial Intelligence*, *2*(2), 113–127.

Grinvald, A., & Bonhoeffer, T. (2002). Optical imaging of electrical activity based on intrinsic signals and on voltage sensitive dyes: The methodology.

Lund, A. (2018). *glamlasso: Penalization in large scale generalized linear array models*. R package version 3.0.

Lund, A. (2021). *SMME: Soft maximin estimation for large scale heterogeneous data*. R package version 1.0.1.

Lund, A., Vincent, M., & Hansen, N. R. (2017). Penalized estimation in large-scale generalized linear array models. *Journal of Computational and Graphical Statistics*, *26*(3), 709–724.

Meinshausen, N., & Bühlmann, P. (2015). Maximin effects in inhomogeneous large-scale data. *The Annals of Statistics*, *43*(4), 1801–1830.

Roland, P. E., Hanazawa, A., Undeman, C., Eriksson, D., Tompa, T., Nakamura, H., Valentiniene, S., & Ahmed, B. (2006). Cortical feedback depolarization waves: A mechanism of top-down influence on early visual areas. *Proceedings of the National Academy of Sciences*, *103*(33), 12586–12591.

Roll, J. (2008). Piecewise linear solution paths with application to direct weight optimization. *Automatica*, *44*(11), 2732–2737.

Rothenhäusler, D., Meinshausen, N., Bühlmann, P., & Peters, J. (2021). Anchor regression: Heterogeneous data meet causality. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *83*(2), 215–246.

Tseng, P., & Yun, S. (2009). A coordinate gradient descent method for nonsmooth separable minimization. *Mathematical Programming*, *117*(1-2), 387–423.

Wright, S. J., Nowak, R. D., & Figueiredo, M. A. (2009). Sparse reconstruction by separable approximation. *IEEE Transactions on Signal Processing*, *57*(7), 2479–2493.

How to cite this article: Lund, A., Wengel Mogensen, S., & Richard Hansen, N. (2022). Soft maximin estimation for heterogeneous data. *Scandinavian Journal of Statistics*, 1–30. https://doi.org/10.1111/sjos.12580

APPENDIX A. PROOFS

Proof of Lemma 1. (i) Since $x_g = \max\{x_1, \dots, x_G\}$ for some $g \in \{1, \dots, G\}$,

$$\max\{x_1, \dots, x_G\} = \frac{\log(e^{\zeta x_g})}{\zeta} \leq \frac{\log(e^{\zeta x_g} + \sum_{j \neq g} e^{\zeta x_j})}{\zeta} = \mathrm{lse}(x),$$

and also

$$\mathrm{lse}(x) \leq \frac{\log(\sum_j e^{\zeta x_g})}{\zeta} = \frac{\log(G e^{\zeta x_g})}{\zeta} = \frac{\log(G)}{\zeta} + \max\{x_1, \dots, x_G\},$$

and the statement follows.

(ii) From l'Hopitals rule we get

$$\lim_{\zeta \downarrow 0} \frac{\log\left(\frac{1}{G}\sum_j e^{\zeta x_j}\right)}{\zeta} = \lim_{\zeta \downarrow 0} \left(\frac{1}{G}\sum_j x_j e^{\zeta x_j}\right)\left(\frac{1}{G}\sum_j e^{\zeta x_j}\right)^{-1} = \frac{1}{G}\sum_j x_j,$$

implying

$$\mathrm{lse}(x) = \frac{\log(G)}{\zeta} + \frac{\log(\frac{1}{G}\sum_j e^{\zeta x_j})}{\zeta} = \frac{\log(G)}{\zeta} + \frac{1}{G}\sum_j x_j + o(1),$$

for $\zeta \downarrow 0$. ∎

Proof of Proposition 1. First note that for any $\beta \in \mathbb{R}^p$ and any $g \in \{1, \dots, G\}$

$$\begin{aligned} -\hat{V}_g(\beta) &= \beta^\top \hat{\Sigma}_g \beta - 2\beta^\top \hat{\Sigma}_g b_g - \frac{2\beta^\top \boldsymbol{X}_g^\top \boldsymbol{\varepsilon}_g}{n_g} \\ &= -V_{b_g}(\beta) + \beta^\top(\hat{\Sigma}_g - \Sigma)\beta - 2\beta^\top(\hat{\Sigma}_g - \Sigma)b_g - \frac{2\beta^\top \boldsymbol{X}_g^\top \boldsymbol{\varepsilon}_g}{n_g} \\ &\geq -V_{b_g}(\beta) - \|\beta\|_1^2 D - 2\|\beta\|_1 \max_g \|b_g\|_1 D - 2\|\beta\|_1 \delta, \end{aligned} \tag{A1}$$

and correspondingly

$$-\hat{V}_g(\beta) \leq -V_{b_g}(\beta) + \|\beta\|_1^2 D + 2\|\beta\|_1 \max_g \|b_g\|_1 D + 2\|\beta\|_1 \delta. \tag{A2}$$

By assumption $\max_g \|b_g\|_1 \leq \kappa$, and if also $\|\beta\|_1 \leq \kappa$ we can write

$$\text{lse}_\zeta(-V(\beta)) - 3\kappa^2 D - 2\kappa\delta \leq \text{lse}_\zeta(-\hat{V}(\beta)) \tag{A3}$$

$$\leq \text{lse}_\zeta(-V(\beta)) + 3\kappa^2 D + 2\kappa\delta, \tag{A4}$$

by (A1) and (A2) and the properties of lse_ζ.

Let H denote the convex hull of $\{b_1, \dots, b_g\}$. By theorem 1 in Meinshausen and Bühlmann (2015) we know that $b^* \in H$ and since $\max_g \|b_g\|_1 \leq \kappa$ we also have $\|b^*\|_1 \leq \kappa$. By (7) it follows that

$$\text{lse}_\zeta(-\hat{V}(\hat{\beta}^\kappa_{smm})) \leq \text{lse}_\zeta(-\hat{V}(b^*)). \tag{A5}$$

Using (A3) and (A4) on respectively the left and right hand side of (A5), yields

$$\text{lse}_\zeta(-V(\hat{\beta}^\kappa_{smm})) \leq \text{lse}_\zeta(-V(b^*)) + 6\kappa^2 D + 4\kappa\delta. \tag{A6}$$

Applying Lemma 1 on both left and right hand side of (A6) finally yields

$$\max_g\{-V_{b_g}(\hat{\beta}^\kappa_{smm})\} \leq \max_g\{-V_{b_g}(b^*)\} + 6\kappa^2 D + 4\kappa\delta + \frac{\log(G)}{\zeta}. \tag{A7}$$

For the last statement, note that for fixed $\beta \in \mathbb{R}^p$, since $b \mapsto -V_b(\beta)$ is affine

$$\max_g\{-V_{b_g}(\beta)\} = \max_{b\in H}\{-V_b(\beta)\}. \tag{A8}$$

An implication of Theorem 1 in Meinshausen and Bühlmann (2015) is $(b^*)^\top \Sigma b \geq (b^*)^\top \Sigma b^*, \forall b \in H$, which combined with (A8) shows

$$\max_g\{-V_{b_g}(b^*)\} = \max_{b\in H}\{(b^*)^\top \Sigma b^* - 2(b^*)^\top \Sigma b\} = -(b^*)^\top \Sigma b^*. \tag{A9}$$

Using $b^* \in H$ and (A8), on the left-hand side of (A7), and using (A9) on the right-hand side of (A7), gives us

$$(\hat{\beta}^\kappa_{smm})^\top \Sigma \hat{\beta}^\kappa_{smm} - 2(\hat{\beta}^\kappa_{smm})^\top \Sigma b^* \leq -(b^*)^\top \Sigma b^* + 6\kappa^2 D + 4\kappa\delta + \frac{\log(G)}{\zeta},$$

which can be rearranged to yield the statement. ∎

To prove Proposition 2 we need the following technical lemma.

Lemma 2. *Assume* $\sum_i w_i = 1$ *and* $h_i \in \mathbb{R}^p$, $i \in \{1, \dots, G\}$, $G \in \mathbb{N}$. *Then*

$$\sum_i w_i h_i \left(h_i^\top - \sum_j w_j h_j^\top \right) = \sum_i \sum_{j>i} w_i w_j (h_i - h_j)(h_i - h_j)^\top.$$

Proof. First note that since $1 - w_i = \sum_{j \neq i} w_j$

$$\sum_i w_i h_i \left(h_i^\top - \sum_j w_j h_j^\top \right) = \sum_i w_i h_i \left((1 - w_i) h_i^\top - \sum_{j \neq i} w_j h_j^\top \right)$$
$$= \sum_i \sum_{j \neq i} w_i w_j h_i (h_i - h_j)^\top.$$

Letting $a_{i,j} = w_i w_j h_i (h_i - h_j)^\top$ we find that

$$\begin{aligned}
\sum_i \sum_{j \neq i} a_{i,j} &= \sum_i \sum_{j>i} a_{i,j} + \sum_i \sum_{i>j} a_{i,j} \\
&= \sum_i \sum_{j>i} a_{i,j} + \sum_j \sum_{i>j} a_{i,j} \quad \text{(interchange summation order)} \\
&= \sum_i \sum_{j>i} a_{i,j} + \sum_i \sum_{j>i} a_{j,i} \quad \text{(relabel summation indices)} \\
&= \sum_i \sum_{j>i} w_i w_j (h_i - h_j)(h_i - h_j)^\top,
\end{aligned}$$

where we used that $a_{j,i} = -w_i w_j h_j (h_i - h_j)^\top$. ∎

Proof of Proposition 2. First it is straightforward to compute the gradient of the loss

$$\begin{aligned}
\nabla l_\zeta(\beta) &= \frac{\sum_{j=1}^G e^{\zeta h_j(\beta)} \nabla \zeta h_j(\beta)}{\zeta \sum_{j=1}^G e^{\zeta h_j(\beta)}} \\
&= e^{-\log(\sum_{j=1}^G e^{\zeta h_j(\beta)})} \sum_{j=1}^G e^{\zeta h_j(\beta)} \nabla h_j(\beta) \\
&= \sum_{j=1}^G w_{j,\zeta}(\beta) \nabla h_j(\beta). \qquad \text{(A10)}
\end{aligned}$$

Then since

$$w_{j,\zeta}(\beta) = \frac{e^{\zeta h_j(\beta)}}{\sum_{i=1}^G e^{\zeta h_i(\beta)}} \geq 0$$

we see that $\sum_j w_{j,\zeta}(\beta) = 1$ and conclude that the weights $w_\zeta(\beta)$ are convex for any β and ζ.

Differentiating (A10) gives us

$$\nabla^2 l_\zeta(\beta) = \sum_{j=1}^G \nabla h_j(\beta) \nabla w_{j,\zeta}(\beta)^\top + \sum_{j=1}^G w_{j,\zeta}(\beta) \nabla^2 h_j(\beta).$$

Using the definition of $w_{j,\zeta}$ and (A10) the first term is equal to

$$\sum_{j=1}^G w_{j,\zeta}(\beta) \nabla h_j(\beta) \left(\nabla h_j(\beta)^\top - \sum_{i=1}^G w_{i,\zeta}(\beta) \nabla h_i(\beta)^\top \right)$$

$$= \sum_{j=1}^{G} \sum_{i>j} w_{j,\zeta}(\beta) w_{i,\zeta}(\beta) (\nabla h_j(\beta) - \nabla h_i(\beta))(\nabla h_j(\beta) - \nabla h_i(\beta))^\top,$$

where the equality follows from Lemma 2 since $(w_{j,\zeta}(\beta))_j$ are convex weights.

For a twice continuously differentiable function f it holds that f is strongly convex with parameter $\nu > 0$ if and only if $\nabla^2 f - \nu I$ is positive semi-definite. Assuming all h_i are convex and at least one is ν-strongly convex it follows directly from (11) that l_ζ is also ν-strongly convex.

Finally, the Hessian of $e^{l_\zeta(\beta)}$ is

$$\nabla^2 e^{l_\zeta(\beta)} = \nabla^2 l_\zeta(\beta) e^{l_\zeta(\beta)} + \nabla_\beta l_\zeta(\beta) \nabla_\beta l_\zeta(\beta)^\top e^{l_\zeta(\beta)},$$

where $\nabla_\beta l_\zeta(\beta) \nabla_\beta l_\zeta(\beta)^\top e^{l_\zeta(\beta)}$ is positive semi-definite for all β. Letting $m = \min_\beta l_\zeta(\beta) \in \mathbb{R}$, we have $e^{l_\zeta(\beta)} \geq e^m > 0$ for all β and must have $\nabla^2 e^{l_\zeta(\beta)} - \tilde{\nu} I$ is positive semi-definite for some $\tilde{\nu} > 0$, showing that e^{l_ζ} is strongly convex. ∎

Proof of Corollary 1. Let $\|\cdot\|_d$ denote the 2-norm on $\mathbb{R}^d$ and A be a $d_1 \times d_2$ matrix. Then $|||A|||_{d_1,d_2} := \sup_{v:\|v\|_{d_2}=1} \|Av\|_{d_1}$ is the sub-multiplicative matrix (operator) norm induced by the 2-norms on $\mathbb{R}^{d_1}$ and $\mathbb{R}^{d_2}$. For $a \in \mathbb{R}^{d_1}$ and $b \in \mathbb{R}^{d_2}$ note that $|||a|||_{d_1,1} = \|a\|_{d_1}$ (Cauchy–Schwarz) and we get

$$|||ab^\top||| = \sup_{v:\|v\|_{d_2}=1} \|ab^\top v\|_{d_1} = \sup_{v:\|v\|_{d_2}=1} \|a\|_{d_1} |b^\top v| = \|a\|_{d_1} \|b\|_{d_2}.$$

Now suppressing subscripts observe that $|||\nabla^2 h_g(\beta)||| = 2|||\boldsymbol{X}^\top \boldsymbol{X}|||/m$ and $\nabla h_i(\beta) - \nabla h_j(\beta) = 2\boldsymbol{X}^\top(\boldsymbol{Y}_i - \boldsymbol{Y}_j)/m$. Then by Proposition 2 it follows that

$$\begin{aligned}
|||\nabla^2 l_\zeta(\beta)||| &\leq \sum_i \sum_{j>i} w_{i,\zeta}(\beta) w_{j,\zeta}(\beta) |||(\nabla h_i(\beta) - \nabla h_j(\beta))(\nabla h_i(\beta) - \nabla h_j(\beta))^\top||| \\
&\quad + \sum_j w_{j,\zeta}(\beta) |||\nabla^2 h_j(\beta)||| \\
&= \frac{4}{m^2} \sum_i \sum_{j>i} w_{i,\zeta}(\beta) w_{j,\zeta}(\beta) \|\boldsymbol{X}^\top(\boldsymbol{Y}_i - \boldsymbol{Y}_j)\|^2 + \frac{2|||\boldsymbol{X}^\top \boldsymbol{X}|||}{m} \\
&\leq \frac{4}{m^2} \max_{i,j} \|\boldsymbol{X}^\top(\boldsymbol{Y}_i - \boldsymbol{Y}_j)\|^2 + \frac{2|||\boldsymbol{X}^\top \boldsymbol{X}|||}{m} \\
&\leq \frac{4|||\boldsymbol{X}^\top \boldsymbol{X}|||}{m^2} \left(\max_{i,j} \|\boldsymbol{Y}_i - \boldsymbol{Y}_j\|^2 + \frac{m}{2} \right),
\end{aligned}$$

using the properties of the matrix norm. By the mean value theorem it follows that ∇l_ζ is Lipschitz continuous with the claimed bound. ∎

Proof of Proposition 3. If we can show that assumption A.1 from Chen et al. (2016) holds for the soft maximin problem (8) we can use theorem A.1 in Chen et al. (2016) (or lemma 4 in Wright et al., 2009) to show that the sequence has an accumulation point. Theorem 1 in Wright et al. (2009) then establishes this accumulation point as a critical point for $F_\zeta = l_\zeta + \lambda J$.

Let $\Delta > 0$, $\beta_0 \in \mathbb{R}^p$, and define the set

$$A_0 = \{\beta : F_\zeta(\beta) \leq F_\zeta(\beta_0)\}$$
$$A_{0,\Delta} = \{\beta : \|\beta - \beta'\| \leq \Delta, \beta' \in A_0\}.$$

A.1(i): l_ζ is ν-strongly convex by Proposition 2 and since J is assumed convex it follows that F_ζ is strongly convex. So A_0 is compact hence $A_{0,\Delta}$ is compact as a closed neighbourhood of A_0. As l_ζ is C^∞ everywhere, ∇l_ζ is Lipschitz on $A_{0,\Delta}$.

A.1(ii): Is satisfied by assumptions on J.

A.1(iii): Clearly $F_\zeta \geq 0$. Furthermore F_ζ is continuous hence uniformly continuous on the compact set A_0.

A.1(iv) $\sup_{\beta \in A_0} \|\nabla l_\zeta\| < \infty$ as A_0 is compact and ∇l_ζ is continuous. Moreover, $\sup_{\beta \in A_0} \|J\| < \infty$ as A_0 is compact and J is continuous. Finally, also $\inf J = 0$. ∎

APPENDIX B. BRAIN IMAGING DATA

The neuronal activity recordings were obtained using voltage-sensitive dye imaging (VSDI) in an experiment previously described in Roland et al. (2006). In short part of the visual cortex of a live ferret was exposed and stained with voltage-sensitive dye. Changes in membrane potential affects the dye and alters its fluorescence. The neuronal activity is recorded indirectly in terms of changes in the fluorescence using 464 photodiode channels organized in a 2D (hexagonal) array. By padding with zeros the 464 channels were mapped to a 25×25 matrix. We note the padding is chosen as the data is centered around zero implying the analysis is not altered by this manipulation. Alternatively observation weights can be used at a computational cost. During the trial (625 ms) an image was recorded every 0.6136 ms. For 250 ms of the trial a visual stimulus, a white square on a grey screen, was presented to the ferret. A total of $G = 275$ trials were recorded across 13 different ferrets.

Several sources of heterogeneity are potentially present in the raw data:

(i) The heart beat affects the light emission by expanding the blood vessels in the brain, creating a cyclic heart rate dependent artefact. A changing heart rate over trials for one animal (fatigue) as well as differences in heart rate between animals will cause heterogeneity in the data.
(ii) Spatial inhomogeneities can arise due to differences in the cytoarchitectural borders between the animals causing misalignment problems.
(iii) The VSDI technique is very sensitive, see Grinvald and Bonhoeffer (2002). Even small changes in the experimental surroundings could affect the recordings and create heterogeneity.
(iv) Differences between animals in how they respond to the visual stimulus.

A trial with no visual stimulus (baseline), was recorded right before recording the stimulus trial. By aligning the baseline and stimulus trial, using an electrocardiography recording, the two recordings were subtracted to remove the heart rate artefact. We use this preprocessed data in the experiment.

Figure B1 shows recordings for five trials in the temporal dimension (panel (a)) and spatial dimension (panel (b)). Note that following the onset of the visual stimulus after 200 ms (first

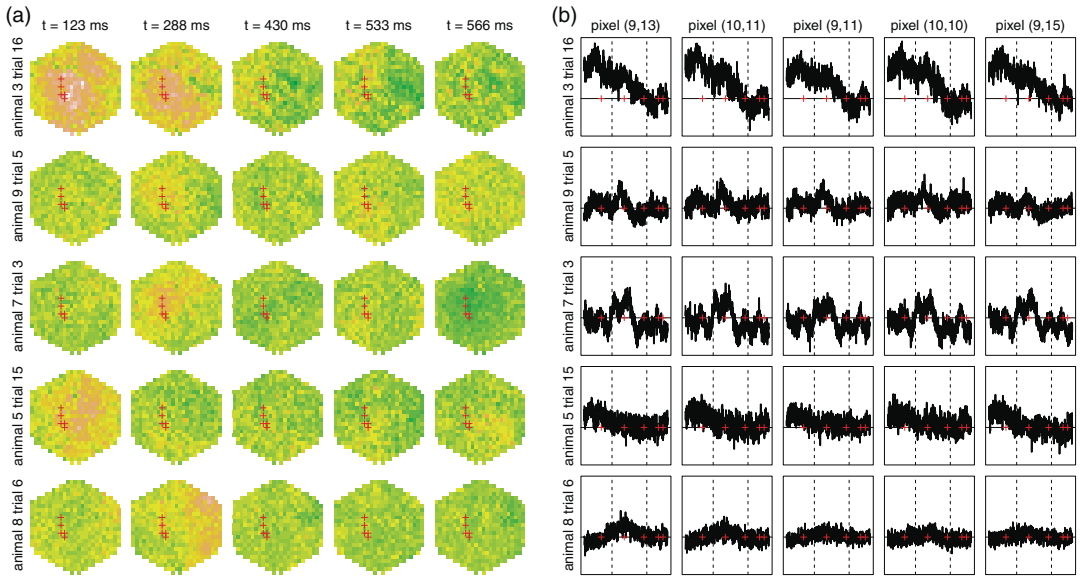

FIGURE B1 (a) Spatial plots for five trials and five time points. The red crosses indicate the pixel time series shown in b. (b) Temporal plots for the five same trials and five pixels indicated with red crosses in part a. The dashed lines indicate stimulus start and stop and red crosses indicate the time points plotted in part a [Colour figure can be viewed at wileyonlinelibrary.com]

dashed line), the recordings are expected to show the result of a depolarization of the neuronal cells. Visual inspection of Figure B1 however does not reveal a clear stimulus response in every trial. We note the presence of variation that seems to be specific to the trial and could reflect the heterogeneity listed above.

B.1 Experiment setup

We use the array-tensor model from Section 3.2 with $p_1 = p_2 = 9$ B-spline functions in each spatial dimension and $p_3 = 80$ B-splines in the temporal dimension. This gives us a model with marginal design matrices Φ_1, Φ_2, and Φ_3, of sizes 25×9, 25×9 and 977×80, respectively, given by the B-splines evaluations over the marginal domains. The model has $p = 6480$ parameters.

We let one fold consist of all data from 2 out of the 13 animals. The model is trained on the fold and tested on data from the remaining 11 animals, for each method and each value of λ. We repeat this procedure $N := \binom{13}{2} = 78$ times, yielding 78 fitted models and corresponding test metrics for each method and each value of λ. Since the number of trials is not constant across animals the number of groups in each fold ranges from 23 to 80 (average is 42) giving us 14,044,375 to 48,850,000 (average 25,646,250) observations in each fold.

B.2 Experimental results

From the left display in Figure B2 we see that on average soft maximin with $\zeta = 200$ (model no. 6) achieves the lowest over all out of sample RMSE. The low ζ estimators, pooled (model no. 5) and $\zeta = 2$ (model no. 5) perform somewhat worse on average but still achieves significant reduction compared to the zero prediction. The approximate maximin estimator, magging estimator (model no. 13), performs the worst on this data in terms of RMSE.

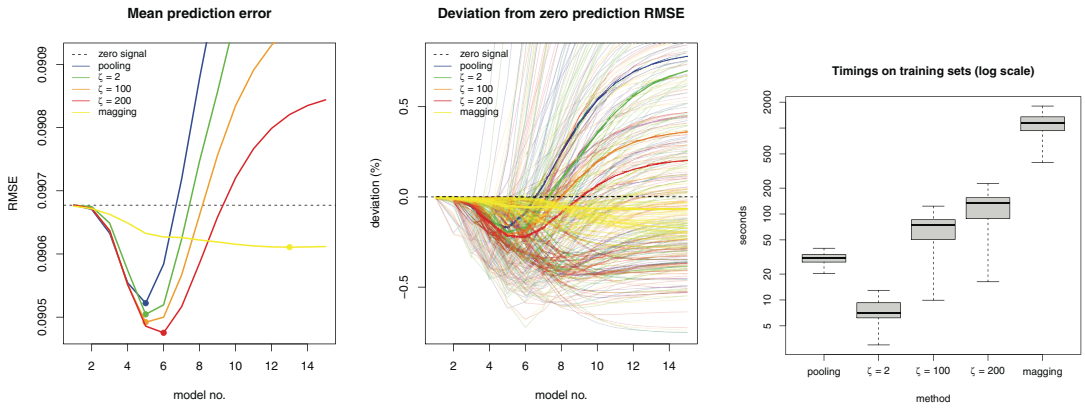

FIGURE B2 Left: Average RMSE across 78 test sets as a function of model complexity (model no.). Zero signal (dashed), pooling (blue), soft maximin $\zeta \in \{2{,}100, 200\}$ (resp. green, orange and red) and magging (yellow). Middle: Deviation in RMSE relative to the zero prediction for each method and on each test set (thin lines), as a function of model complexity. Averages are indicated with a thick line of the same color. Right: Run times (log scale) for the 78 training sets for each method [Colour figure can be viewed at wileyonlinelibrary.com]

Looking at Figure B2 the low ζ methods show more variability than the high ζ methods and in particular are more prone to make predictions that are worse than the zero prediction. However the picture is not as clear as on the simulated data. We note that the magging estimator is quite consistently better than the zero prediction but not much, possibly reflecting the conservative nature of the hard maximin method.

Figure B2 also summarizes the timings for each method. Notably all the soft maximin estimators (8.9, 15.8, and 18.7 s) outperform the pooled estimator (26.4 s) while also yielding better prediction accuracy (Figure B2). The magging estimator (931.4 s) suffers from having to compute individual fits for each group in the training set, making the the method orders of magnitudes slower in this case, without obtaining better accuracy. Note that the task alone of maximin aggregating the individual group estimates, by solving the associated quadratic programming problem, took on average 45 s. So even if fully parallelized the magging estimator is still computationally more demanding than the softmaximin estimator on this data set.

APPENDIX C. WASHINGTON DC BIKE DATA

Here we show how to systematically determine the soft maximin parameter ζ.

Owing to the temporal dependence in the data we will use a rolling cross-validation scheme to systematically tune ζ. We do this by training the model (20) on each set of six consecutive months and testing on the six following months. Following the experiment in Section 4.1 we perform this rolling window CV in two ways; (i) rolling forward from January 2011 to December 2012 and (ii) rolling backwards from December 2012 to January 2011.

For (i) and (ii), respectively, we then have 13 training and test pairs. We compute the soft maximin estimator on each of these training set for 50 values of ζ that range exponentially between 0.0001 and 0.3. On the corresponding test set we compute the mean squared prediction error.

Figure C1 shows the average prediction error (average RMSE) as a function of ζ for the forward rolling scheme (i) and the backward rolling scheme (ii), respectively. In line with Section 4.1 in (i) we see that low ζ values, that is, pooled OLS gives better predictions in terms of RMSE than higher

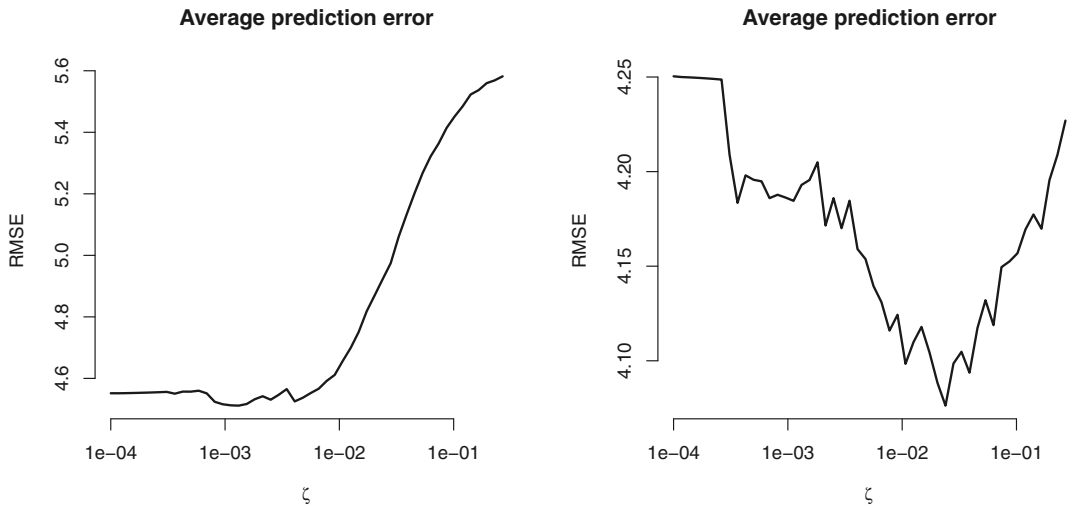

FIGURE C1 Left: Average prediction error as a function of ζ obtained by the forward rolling cross validation procedure in experiment (i). Right: Average prediction error as a function of ζ obtained by the backward rolling cross validation procedure in experiment (ii)

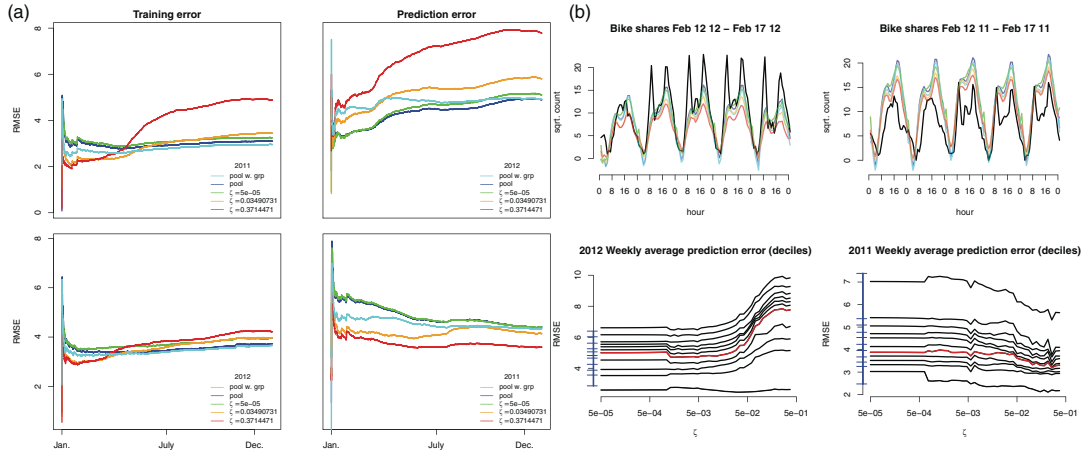

FIGURE C2 Results from fitting a model that includes temperature and humidity as covariates and using month as grouping [Colour figure can be viewed at wileyonlinelibrary.com]

values. For the experiment in (ii), however, ζ values around 0.03 yields the minimum prediction error.

We note that any conclusion will depend on the nature of the heterogeneity in the data as well as on how the cross-validation is carried out, that is, how the model is trained and tested. For the specific bike dataset heterogeneity is not very pronounced and is easily explained in terms of increasing utilization of the bike sharing scheme. This causes an optimistic method like pooling to perform better over time than a conservative method like maximin. However we observe that the hard maximin (high ζ) seems suboptimal in both experiments (i) and (ii) highlighting the benefit of computing a range of soft maximin estimators for a given dataset.

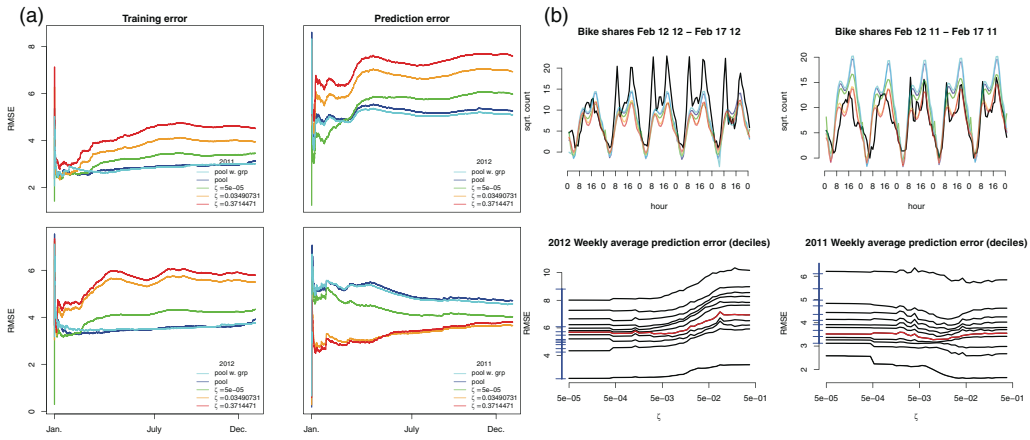

FIGURE C3 Results from fitting a model that use `weathersit` as grouping [Colour figure can be viewed at wileyonlinelibrary.com]

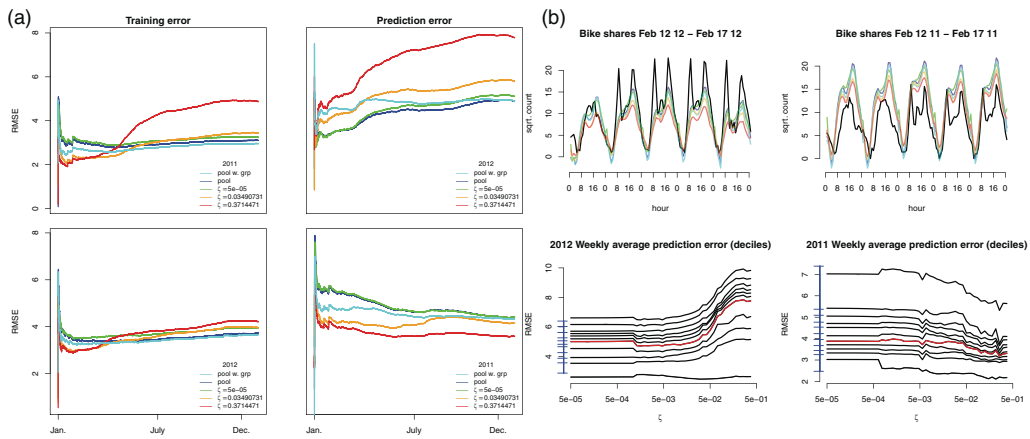

FIGURE C4 Results from fitting a model that includes temperature and humidity and use `weathersit` as grouping [Colour figure can be viewed at wileyonlinelibrary.com]

C.1 Alternative models and grouping

Figure C2 shows the results with temperature `temp` and humidity `hum` added to (20). Results for this model are similar to the those in Section 4.1 though with less pronounced gain in prediction robustness (Figure C2b Bottom right).

We also group the data according to `weathersit` and fit the model

$$\sqrt{\texttt{cnt}_i} = \alpha_0 + \sum_{j=1}^{10} \alpha_j \varphi_j(\texttt{hr}_i) + \sum_{j=1}^{5} \beta_j \varphi_j(\texttt{weekday}_i) + \sum_{j=1}^{5} \gamma_j \varphi_j(\texttt{mnth}_i) + \varepsilon_i. \tag{C1}$$

With this setup we obtain slightly higher 2011 prediction accuracy but less robust predictions (Figure C3a,b bottom right).

Finally we also add `temp` and `hum` to (C1). For this extended model the soft maximin estimator gives seemingly better 2011 predictions than the pooled estimator, see Figure C4.