Received: 28 February 2020 | Accepted: 27 April 2021

DOI: 10.1111/rssb.12432

ROYAL STATISTICAL SOCIETY
DATA | EVIDENCE | DECISIONS
Journal of the Royal Statistical Society
Series B
Statistical
Methodology
B

ORIGINAL ARTICLE

Modelling high-dimensional categorical data using nonconvex fusion penalties

Benjamin G. Stokell[1,*] | **Rajen D. Shah**[1,†] | **Ryan J. Tibshirani**[2]

[1] University of Cambridge, Cambridge, UK

[2] Carnegie Mellon University, Pittsburgh, PA, USA

Correspondence
Rajen D. Shah, University of Cambridge, Cambridge, UK.
Email: r.shah@statslab.cam.ac.uk

Abstract

We propose a method for estimation in high-dimensional linear models with nominal categorical data. Our estimator, called SCOPE, fuses levels together by making their corresponding coefficients exactly equal. This is achieved using the minimax concave penalty on differences between the order statistics of the coefficients for a categorical variable, thereby clustering the coefficients. We provide an algorithm for exact and efficient computation of the global minimum of the resulting nonconvex objective in the case with a single variable with potentially many levels, and use this within a block coordinate descent procedure in the multivariate case. We show that an oracle least squares solution that exploits the unknown level fusions is a limit point of the coordinate descent with high probability, provided the true levels have a certain minimum separation; these conditions are known to be minimal in the univariate case. We demonstrate the favourable performance of SCOPE across a range of real and simulated datasets. An R package `CatReg` implementing SCOPE for linear models and also a version for logistic regression is available on CRAN.

[*] Supported by the Cantab Capital Institute for the Mathematics of Information.

[†] Supported by an EPSRC Programme Grant.

KEYWORDS
categorical data, dynamic programming, high-dimensional regression, nonconvex penalty

1 | INTRODUCTION

Categorical data arise in a number of application areas. For example, electronic health data typically contain records of diagnoses received by patients coded within controlled vocabularies and also prescriptions, both of which give rise to categorical variables with large numbers of levels (Jensen et al., 2012). Vehicle insurance claim data also contain a large number of categorical variables detailing properties of the vehicles and parties involved (Hu et al., 2018). When performing regression with such data as covariates, it is often helpful, both for improved predictive performance and interpretation of the fit, to fuse the levels of several categories together in the sense that the estimated coefficients corresponding to these levels have exactly the same value.

To fix ideas, consider the following ANOVA model relating response vector $Y = (Y_1, \ldots, Y_n)^T \in \mathbb{R}^n$ to categorical predictors $X_{ij} \in \{1, \ldots, K_j\}, j = 1, \ldots, p$:

$$Y_i = \mu^0 + \sum_{j=1}^{p} \sum_{k=1}^{K_j} \theta^0_{jk} \mathbb{1}_{\{X_{ij}=k\}} + \varepsilon_i. \tag{1}$$

Here the ε_i are independent zero mean random errors, μ^0 is a global intercept and θ^0_{jk} is the contribution to the response of the kth level of the jth predictor; we will later place restrictions on the parameters to ensure they are identifiable. We are interested in the setting where the coefficients corresponding to any given predictor are clustered, so defining

$$s_j := |\{\theta^0_{j1}, \ldots, \theta^0_{jK_j}\}|, \tag{2}$$

we have $s_j \ll K_j$, at least when K_j is large. Note that our setup can include high-dimensional settings where p is large and many of the predictors do not contribute at all to the response: when $s_j = 1$, the contribution of the jth predictor is effectively null as it may be absorbed by the intercept term.

1.1 | Background and motivation

Early work on collapsing levels together in low-dimensional models of the form (1) focused on performing a variety of significance tests for whether certain sets of parameters were equal (Calinski & Corsten, 1985; Scott & Knott, 1974; Tukey, 1949). A more modern and algorithmic method based on these ideas is delete or merge regressors (DMR) (Maj-Kańska et al., 2015), which involves agglomerative clustering based on t-statistics for differences between levels.

The CART algorithm (Breiman et al., 1984) for building decision trees effectively starts with all levels of the variables fused together and greedily selects which levels to split. One potential drawback of these greedy approaches is that in high-dimensional settings where the search space is very large, they may fail to find good groupings of the levels. The popular random forest procedure (Breiman, 2001) uses randomisation to alleviate the issues with the greedy nature of the splits, but sacrifices interpretability of the fitted model.

An alternative to greedy approaches in high-dimensional settings is using penalty-based methods such as the Lasso (Tibshirani, 1996). This can be applied to continuous or binary data and involves optimising an objective for which global minimisation is computationally tractable, thereby avoiding some of the pitfalls of greedy optimisation. In contrast to random forest, the fitted models are sparse and interpretable. Inspired by the success of the Lasso and related methods for high-dimensional regression, a variety of approaches have proposed estimating $\boldsymbol{\theta}^0 = (\theta^0_{jk})_{j=1,\ldots,p,k=1,\ldots,K_j}$ and μ_0 via optimising over $(\mu, \boldsymbol{\theta})$ a sum of a least squares criterion

$$\ell(\mu, \boldsymbol{\theta}) := \frac{1}{2n}\sum_{i=1}^{n}\left(Y_i - \mu - \sum_{j=1}^{p}\sum_{k=1}^{K_j}\theta_{jk}\mathbb{1}_{\{X_{ij}=k\}}\right)^2 \tag{3}$$

and a penalty of the form

$$\sum_{j=1}^{p}\sum_{k=2}^{K_j}\sum_{l=1}^{k-1} w_{j,kl}\,|\theta_{jk} - \theta_{jl}|. \tag{4}$$

This is the CAS-ANOVA penalty of Bondell and Reich (2009). The weights $w_{j,kl}$ can be chosen to balance the effects of having certain levels of categories more prevalent than others in the data. The penalty is an ‘all-pairs’ version of the fused Lasso and closely related to so-called convex clustering (Chiquet et al., 2017; Hocking et al., 2011). We note that there are several other approaches besides using penalty functions. For instance, Pauger and Wagner (2019) proposes a Bayesian modelling procedure using sparsity-inducing prior distributions to encourage fusion of levels. See also Tutz and Gertheiss (2016) and references therein for a review of other methods including those based on mixture models and kernels.

The fact that the optimisation problem resulting from (4) is convex makes the procedure attractive. However, a drawback is that it may not give a desirable form of shrinkage. Indeed, consider the case where $p = 1$, and dropping subscripts for simplicity, all $w_{kl} = 1$. This would typically be the case if all levels were equally prevalent. Further suppose for simplicity that the number of levels K is even. Then if the coefficients are clustered into two groups where one contains only a single isolated coefficient, the number of non-zero summands in Equation (4) is only $K - 1$. This almost doubles to $2(K - 2)$ when one of the two groups is of size 2. The extreme case where the two groups are of equal size yields $(K/2)^2$ non-zero summands. This particular property of all-pairs penalties, which results in them favouring groups of unequal sizes, is illustrated schematically in Figure 1. We can see the impact of this in the following concrete example.

Suppose $K = 20$ levels are clustered into four groups with

$$\begin{aligned}\theta^0_1 = \cdots = \theta^0_4 = -6, &\quad \theta^0_5 = \cdots = \theta^0_{10} = -2.5\\ \theta^0_{11} = \cdots = \theta^0_{16} = 2.5, &\quad \theta^0_{17} = \cdots = \theta^0_{20} = 6.\end{aligned}$$

If the coefficient estimates satisfy $\hat{\theta}_1 = \cdots = \hat{\theta}_4 < \hat{\theta}_5 = \cdots = \hat{\theta}_{10} \le \hat{\theta}_k$ for all $k \ge 11$, so the first two groups have distinct coefficients, then moving any coefficient from the first group towards the second, and so increasing the number of estimated groups, actually *decreases* the penalty contribution in Equation (4). Specifically, if the kth coefficient for some $k \in \{1, \ldots, 4\}$ moves to $\hat{\theta}_k + t$ for $t \in [0, \hat{\theta}_5 - \hat{\theta}_4]$ with all other coefficients kept fixed, the penalty contribution decreases by $13t$. In this case, then CAS-ANOVA will struggle to keep the groups intact, especially smaller ones. We see this in Figure 2, which shows the result of applying CAS-ANOVA to data generated according to (1) with $p = 1$, $\boldsymbol{\theta}^0$ as above, $n = 20$ (so we

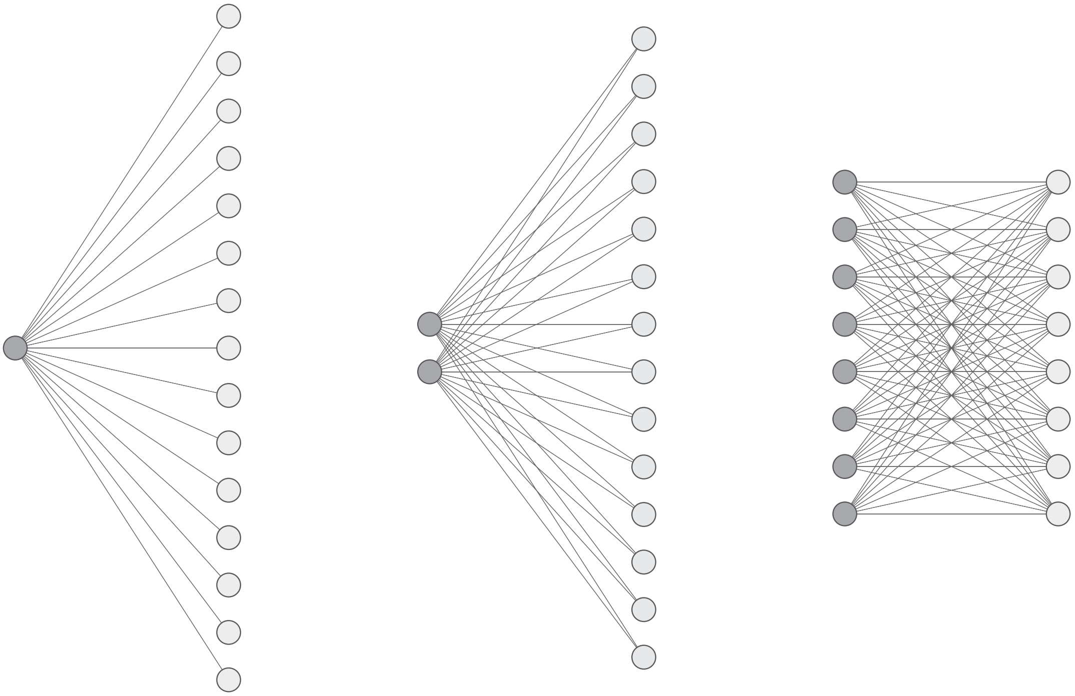


FIGURE 1 Illustration of the number of non-zero summands in (4) when $p = 1$, $K = 16$ and coefficients are clustered into two groups of equal size (right), and where one contains a single coefficient (left) and two coefficients (middle)

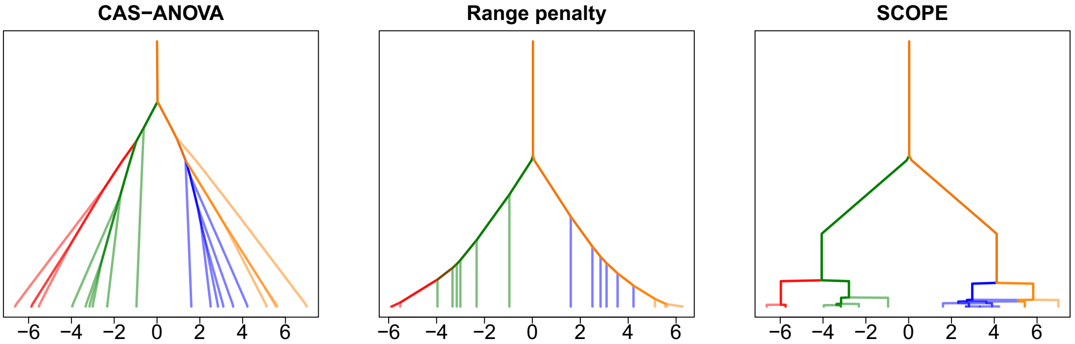


FIGURE 2 Solution paths as the tuning parameter varies in a univariate example where there are four true groups. From left to right: CAS-ANOVA, the range penalty and SCOPE with $\gamma = 8$. The setup is as described in the main text of Section 1.1, with the different colours corresponding to the different true groups. The tuning parameter varies along the y axis. In this example, only SCOPE identifies the four correct groups at any point along its solution path

have a single observation corresponding to each level), and $\varepsilon_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,\ 1)$. There is no value of the tuning parameter λ where the true groups are recovered.

As in the standard regression setting, the bias introduced by all-pairs ℓ_1-type penalties may be reduced by choosing data-adaptive weights analogously to the adaptive Lasso (Zou, 2006), or replacing the absolute value $|\theta_{jk} - \theta_{jl}|$ by $\rho(|\theta_{jk} - \theta_{jl}|)$ where ρ is a concave and non-decreasing penalty function (Ma & Huang, 2017; Oelker et al., 2015). However, this does not address the basic issue of a preference for groups of unequal sizes. Additionally, optimising an objective involving a penalty with $O\left(\sum_{j=1}^{p} K_j^2\right)$ summands can be computationally challenging, particularly in the case where ρ is not convex, both in terms of runtime and memory.

To help motivate the new approach we are proposing in this paper, let us consider the setting where the predictors are ordinal rather than nominal, so there is an obvious ordering among the levels. In these settings, it is natural to consider a fused Lasso (Tibshirani et al., 2005) penalty of the form

$$\sum_{j=1}^{p} \sum_{k=1}^{K_j - 1} |\theta_{j\pi_j(k+1)} - \theta_{j\pi_j(k)}|, \tag{5}$$

where π_j is a permutation of $\{1, \ldots, K_j\}$ specifying the given order; this is done in Gertheiss and Tutz (2010) who advocate using it conjunction with the all-pairs-type CAS-ANOVA penalty for nominal categories.

If, however, we treat the nominal variable setting as analogous to having ordinal variables with unknown orderings π_j, one might initially think of choosing π_j corresponding to the order of the estimates $\boldsymbol{\theta}_j := (\theta_{jk})_{k=1}^{K_j}$, such that $\theta_{j\pi_j(k)} = \theta_{j(k)}$, where $\theta_{j(k)}$ is the kth smallest entry in $\boldsymbol{\theta}_j$. This, however, leads to what we refer to as the 'range' penalty:

$$\sum_{k=1}^{K_j - 1} |\theta_{j(k+1)} - \theta_{j(k)}| = \max_k \theta_{jk} - \min_k \theta_{jk}. \tag{6}$$

While this shrinks the largest and smallest of the estimated coefficients together, the remaining coefficients lying in the open interval between these are unpenalised and so no grouping of the estimates is encouraged, as we observe in Figure 2; see also Oelker et al. (2015) for a discussion of this issue in the context of ordinal variables.

1.2 | Our contributions and organisation of the paper

Given how all-pairs penalties have an intrinsic and undesirable preference for unequal group sizes, and how the fused Lasso applied to ordered coefficients (6) does not result in grouping of the coefficients, we propose the following solution. Our approach is to use the penalty

$$\sum_{j=1}^{p} \sum_{k=1}^{K_j - 1} \rho_j(\theta_{j(k+1)} - \theta_{j(k)}),$$

for concave (and nonconvex) non-decreasing penalty functions ρ_j, which, for computational reasons we discuss in Section 3, we base on the minimax concave penalty (MCP) (Zhang, 2010). In Section 2 we formally introduce our method, which we call SCOPE, standing for Sparse Concave Ordering & Penalisation Estimator.

Note that whereas in conventional high-dimensional regression, the use of nonconvex penalties has been primarily motivated by a need to reduce bias in the estimation of large coefficients (Fan &

Li, 2001), here the purpose is very different: in our setting a nonconvex penalty is in fact even necessary for shrinkage to sparse solutions to occur (see Proposition 1). Because of these fundamental differences, the rich algorithmic and statistical theory concerning high-dimensional regression with nonconvex penalties (see, e.g. Loh and Wainwright (2012, 2015), Wang et al. (2014), Fan et al. (2018), Zhao et al. (2018) and references therein) is not directly applicable to our setting.

In Section 3, we therefore introduce a new dynamic programming approach that recovers the global minimum of the resulting objective function exactly in the univariate case, that is, when $p = 1$. We then build this into a blockwise coordinate descent approach to tackle the multivariate setting.

In Section 4 we study the theoretical properties of SCOPE and give sufficient conditions for the estimator to coincide with the least squares solution with oracular knowledge of the level fusions in the univariate case. These conditions involve a minimal separation between unequal coefficients that is, up to constant factors, minimax optimal. Our result contrasts sharply with Theorem 2 of Ma and Huang (2017) for an all-pairs nonconvex penalty. The latter instead shows the existence of a local optimum that coincides with the oracle least squares solution. While in conventional high-dimensional regression settings, it is known that under certain conditions, all local optima have favourable properties (Loh & Wainwright, 2015), we note that the separation requirements in Ma and Huang (2017) are substantially weaker than those indicated by the minimax lower bound, and so cannot be extended to a particular local optimum determined by the data; see the discussion following Theorem 5.

We use our univariate result to show that the oracle least squares solution is a fixed point of our blockwise coordinate descent algorithm in the multivariate case. In Section 5 we outline some extensions of our methodology including a scheme for handling settings when there is a hierarchy among the categorical variables. Section 6 contains numerical experiments that demonstrate the favourable performance of our method compared to a range of competitors on both simulated and real data. We conclude with a discussion in Section 7. Further details of our algorithm can be found in the Appendix. The supplementary material contains additional information on the runtime of our algorithm, and an approximate version suitable for very large-scale settings, all the proofs, and additional information on the experiments in Section 6.

2 | SCOPE METHODOLOGY

Recall that our goal is to estimate parameters $(\mu^0, \boldsymbol{\theta}^0)$ in model (1). Let us first consolidate some notation. For any $\boldsymbol{\theta} \in \mathbb{R}^{K_1} \times \cdots \times \mathbb{R}^{K_p}$, we define $\boldsymbol{\theta}_j := (\theta_{jk})_{k=1}^{K_j} \in \mathbb{R}^{K_j}$. We will study the univariate setting where $p = 1$ separately, and so it will be helpful to introduce some simplified notation for this case, dropping any extraneous subscripts. We thus write $K \equiv K_1$, $X_i \equiv X_{i1}$ and $\rho \equiv \rho_1$. Additionally, we let $\overline{Y}_k$ denote the average of the Y_i with $X_i = k$:

$$\overline{Y}_k = \frac{1}{n_k} \sum_{i=1}^{n} Y_i \mathbb{1}_{\{X_i=k\}}, \tag{7}$$

where $n_k = \sum_{i=1}^{n} \mathbb{1}_{\{X_i=k\}}$.

In order to avoid an arbitrary choice of corner point constraint, we instead impose the following to ensure that $\boldsymbol{\theta}^0$ is identifiable: for all $j = 1, \ldots, p$ we have

$$g_j(\boldsymbol{\theta}_j^0) = 0, \text{ where } g_j(\boldsymbol{\theta}_j) = \sum_{k=1}^{K_j} n_{jk}\theta_{jk} \text{ and } n_{jk} = \sum_{i=1}^{n} \mathbb{1}_{\{X_{ij}=k\}}. \tag{8}$$

Let $\Theta_j = \{\boldsymbol{\theta}_j \in \mathbb{R}^{K_j}: g_j(\boldsymbol{\theta}_j) = 0\}$, and let $\Theta = \Theta_1 \times \cdots \times \Theta_p$. We will construct estimators by minimising over $\mu \in \mathbb{R}$ and $\boldsymbol{\theta} \in \Theta$ an objective function of the form

$$\tilde{Q}(\mu, \boldsymbol{\theta}) = \ell(\mu, \boldsymbol{\theta}) + \sum_{j=1}^{p} \sum_{k=1}^{K_j - 1} \rho_j(\theta_{j(k+1)} - \theta_{j(k)}),$$

where ℓ is the least squares loss function (3) and $\theta_{j(1)} \leq \cdots \leq \theta_{j(K_j)}$ are the order statistics of $\boldsymbol{\theta}_j$. We allow for different penalty functions ρ_j for each predictor in order to help balance the effects of varying numbers of levels K_j. The identifiability constraint that $\boldsymbol{\theta} \in \Theta$ ensures that the estimated intercept $\hat{\mu}: = \arg\min_\mu \tilde{Q}(\mu, \boldsymbol{\theta})$ satisfies $\hat{\mu} = \sum_{i=1}^{n} Y_i / n$.

We note that while the form of the identifiability constraint would not have a bearing on the fitted values of unregularised least squares regression, this is not necessarily the case when regularisation is imposed. For example, consider the simple univariate setting with $p = 1$ and the corner point constraint $\theta_1 = 0$. Then the fitted value for an observation with level 1 would simply be the average $\overline{Y}_1$, coinciding with that of unpenalised least squares. However, the fitted values with observations with other level $k \geq 2$ would be subject to regularisation and in general be different to $\overline{Y}_k$. This inequitable treatment of the levels is clearly undesirable as they may have been labelled in an arbitrary way. Our identifiability constraint treats the levels more symmetrically, but also takes into account the prevalence of levels, so the fitted values corresponding to more prevalent levels effectively undergo less regularisation.

As the estimated intercept $\hat{\mu}$ does not depend on the tuning parameters, we define

$$Q(\boldsymbol{\theta}) = \frac{1}{2n} \sum_{i=1}^{n} \left(Y_i - \hat{\mu} - \sum_{j=1}^{p} \sum_{k=1}^{K_j} \theta_{jk} \mathbb{1}_{\{X_{ij}=k\}} \right)^2 + \sum_{j=1}^{p} \sum_{k=1}^{K_j - 1} \rho_j(\theta_{j(k+1)} - \theta_{j(k)}). \tag{9}$$

We will take the regularisers ρ_j: $[0, \infty) \to [0, \infty)$ in Equation (9) to be concave (and nonconvex); as discussed in the introduction and formalised in Proposition 1 below, a nonconvex penalty is necessary for fusion to occur.

Proposition 1 *Consider the univariate case with p = 1. Suppose the subaverages $(\overline{Y}_k)_{k=1}^{K}$ (7) are all distinct, and that $\rho_1 \equiv \rho$ is convex. Then any minimiser $\hat{\boldsymbol{\theta}}$ of Q has $\hat{\theta}_k \neq \hat{\theta}_l$ for all* k ≠ l *such that $\hat{\theta}_{(1)} < \overline{Y}_k - \hat{\mu} < \hat{\theta}_{(K)}$ or $\hat{\theta}_{(1)} < \overline{Y}_l - \hat{\mu} < \hat{\theta}_{(K)}$.*

We base the penalties ρ_j :$[0, \infty) \to [0, \infty)$ on the minimax concave penalty (MCP) (Zhang, 2010):

$$\rho(x) = \rho_{\gamma,\lambda}(x) = \int_0^x \lambda \left(1 - \frac{t}{\gamma\lambda}\right)_+ dt,$$

where $(u)_+ = u\mathbb{1}_{\{u \geq 0\}}$. This is a piecewise quadratic function with gradient λ at 0 and flat beyond $\gamma\lambda$. For computational reasons which we discuss in Section 3, the simple piecewise quadratic form of this is particularly helpful. In the multivariate case we take $\rho_j = \rho_{\gamma, \lambda_j}$ with $\lambda_j = \lambda\sqrt{K_j}$. This choice of scaling is motivated by requiring that when $\boldsymbol{\theta}^0 = 0$ we also have $\hat{\boldsymbol{\theta}} = 0$ with high probability; see Lemma 10 in the Supplementary material. We discuss the choice of the tuning parameters λ and γ in Section 3.3, but first turn to the problem of optimising (9).

3 | COMPUTATION

In this section we include details of how SCOPE is computed. Section 3.1 motivates and describes the dynamic programming algorithm we use to compute global minimiser of the SCOPE objective, which is highly nonconvex. Section 3.2 contains details of how this is used to solve the multivariate objective by embedding it within a blockwise coordinate descent routine. Discussion of practical considerations is contained in Section 3.3.

3.1 | Univariate model

3.1.1 | Preliminaries

We now consider the univariate case ($p = 1$) and explain how the solutions are computed. In this case, we may rewrite the least squares loss contribution to the objective function in the following way:

$$\begin{aligned}\frac{1}{2n}\sum_{i=1}^{n}\left(Y_i-\hat{\mu}-\sum_{k=1}^{K}\theta_k \mathbb{1}_{\{X_i=k\}}\right)^2 &= \frac{1}{2n}\sum_{k=1}^{K}\sum_{i=1}^{n}\mathbb{1}_{\{X_i=k\}}(Y_i-\hat{\mu}-\theta_k)^2 \\ &= \frac{1}{2}\sum_{k=1}^{K} w_k(\overline{Y}_k-\hat{\mu}-\theta_k)^2+\frac{1}{2n}\sum_{i=1}^{n}\sum_{k=1}^{K}\mathbb{1}_{\{X_j=k\}}(Y_i-\overline{Y}_k)^2,\end{aligned} \tag{10}$$

where $w_k = n_k/n$. Thus the optimisation problem (9) can be written equivalently as

$$\hat{\boldsymbol{\theta}} \in \underset{\theta\in\Theta}{\arg\min}\frac{1}{2}\sum_{k=1}^{K} w_k\left(\overline{Y}_k-\hat{\mu}-\theta_k\right)^2+\sum_{k=1}^{K-1}\rho\left(\theta_{(k+1)}-\theta_{(k)}\right), \tag{11}$$

suppressing the dependence of the MCP ρ on tuning parameters γ and λ. In fact, it is straightforward to see that the constraint that the solution lies in Θ will be automatically satisfied, so we may replace Θ with $\mathbb{R}^K$. Two challenging aspects of the optimisation problem above are the presence of the nonconvex ρ and the order statistics. The latter, however, are easily dealt with using the result below, which holds more generally whenever ρ is a concave function.

Proposition 2 *Consider the univariate optimisation (11) with* ρ *any concave function such that a minimiser* $\hat{\boldsymbol{\theta}}$ *exists. If for k, l we have* $\overline{Y}_k > \overline{Y}_l$*, then* $\hat{\theta}_k \geq \hat{\theta}_l$*.*

This observation substantially simplifies the optimisation: after re-indexing such that $\overline{Y}_1 \leq \overline{Y}_2 \leq \cdots \leq \overline{Y}_K$, we may re-express (11) as,

$$\hat{\boldsymbol{\theta}} \in \underset{\theta:\theta_1\leq\cdots\leq\theta_K}{\arg\min}\left\{\frac{1}{2}\sum_{k=1}^{K} w_k\left(\overline{Y}_k-\hat{\mu}-\theta_k\right)^2+\sum_{k=1}^{K-1}\rho\left(\theta_{k+1}-\theta_k\right)\right\}. \tag{12}$$

We use the following intermediate functions to structure the algorithm:

$$
\begin{aligned}
f_1(\theta_1) &= \frac{1}{2} w_1(\overline{Y}_1 - \widehat{\mu} - \theta_1)^2, \\
f_k(\theta_k) &= \min_{\theta_{k-1}:\theta_{k-1}\le\theta_k} \{f_{k-1}(\theta_{k-1}) + \rho(\theta_k - \theta_{k-1})\} + \frac{1}{2} w_k(\overline{Y}_k - \widehat{\mu} - \theta_k)^2, \\
b_k(\theta_k) &= \operatorname{sarg\,min}_{\theta_{k-1}:\theta_{k-1}\le\theta_k} \{f_{k-1}(\theta_{k-1}) + \rho(\theta_k - \theta_{k-1})\},
\end{aligned}
\tag{13}
$$

for $k = 2, \ldots, K$; here sarg min refers to the smallest minimiser in the case that it is not unique. Invariably, however, this will be unique, as the following result indicates.

Proposition 3 *The set of* $(\overline{Y}_k)_{k=1}^{K}$ *that yields distinct solutions to (11) has Lebesgue measure zero as a subset of* $\mathbb{R}^K$.

We will thus tacitly assume uniqueness in some of the discussion that follows, although this is not required for our algorithm to return a global minimiser. Observe now that $\widehat{\theta}_K$ is the minimiser of the univariate objective function f_K: indeed for $k \geq 2$,

$$
f_k(\theta_k) = \min_{(\theta_1,\ldots,\theta_{k-1}):\theta_1\le\cdots\le\theta_{k-1}\le\theta_k} \left\{ \frac{1}{2}\sum_{l=1}^{k} w_l(\overline{Y}_l - \widehat{\mu} - \theta_l)^2 + \sum_{l=1}^{k-1} \rho(\theta_{l+1} - \theta_l) \right\}. \tag{14}
$$

Furthermore, we have $\widehat{\theta}_{K-1} = b_K(\widehat{\theta}_K)$, and more generally $\widehat{\theta}_k = b_{k+1}(\widehat{\theta}_{k+1})$ for $k = K-1, \ldots, 1$. Thus provided f_K can be minimised efficiently (which we shall see is indeed the case), given this and the functions $b_2, \ldots, b_K$ we can iteratively compute $\widehat{\theta}_K, \widehat{\theta}_{K-1}, \ldots, \widehat{\theta}_1$. In order to make use of these properties, we must be able to compute f_K and the b_k efficiently; we explain how to do this in the following subsection.

3.1.2 | Computation of f_K and $b_2, \ldots, b_K$

The simple piecewise quadratic form of the MCP-based penalty is crucial to our approach for computing the f_K and the b_k. Some important consequences of this piecewise quadratic property are summarised in the following lemma.

Lemma 4 For each k,

(i) *f_k is continuous, coercive and piecewise quadratic with finitely many pieces;*
(ii) *b_k is piecewise linear with finitely many pieces;*
(iii) *for each $\theta_{k+1} \in \mathbb{R}$, if a minimiser $\tilde{\theta}_k = \tilde{\theta}_k(\theta_{k+1})$ of $\theta_k \mapsto f_k(\theta_k) + \rho(\theta_{k+1} - \theta_k)$ over $(-\infty, \theta_{k+1}]$ satisfies $\tilde{\theta}_k < \theta_{k+1}$, then f_k must be differentiable at $\tilde{\theta}_k$.*

Properties (i) and (ii) above permit exact representation of f_k and b_k with finitely many quantities. The key task then is to form the collection of intervals and corresponding coefficients of quadratic functions for

$$
g_k(\theta_{k+1}) := \min_{\theta_k:\theta_k\le\theta_{k+1}} \{f_k(\theta_k) + \rho(\theta_{k+1} - \theta_k)\} \tag{18}
$$

given a similar piecewise quadratic representation of f_k; and also the same for the linear functions composing b_k. A piecewise quadratic representation of f_{k+1} would then be straightforward to compute, and we can iterate this process. To take advantage of property (iii) above, in computing $g_k(\theta_{k+1})$ we can separately search for minimisers at stationary points in $(-\infty, \theta_{k+1})$ and compare the corresponding function values with $f_k(\theta_{k+1})$; the fact that we need only consider potential minimisers at points of differentiability will simplify things as we shall see below.

Suppose $I_{k,1}, \ldots, I_{k,m(k)}$ are intervals that partition $\mathbb{R}$ (closed on the left) and $q_{k,1}, \ldots, q_{k,m(k)}$ are corresponding quadratic functions such that $f_k(\theta_k) = q_{k,r}(\theta_k)$ for $\theta_k \in I_{k,r}$. Let us write

$$\tilde{q}_{k,r}(\theta_k) = \begin{cases} q_{k,r}(\theta_k) & \text{if } \theta_k \in I_{k,r} \\ \infty & \text{otherwise.} \end{cases}$$

We may then express f_k as $f_k(\theta_k) = \min_r \tilde{q}_{k,r}(\theta_k)$. We can also express the penalty $\rho = \rho_{\gamma,\lambda}$ in a similar fashion. Let

$$\begin{aligned} &\tilde{\rho}_1(x) := -\gamma\lambda^2\{1 - x/(\gamma\lambda)\}^2/2 + \gamma\lambda^2/2 \;\text{ if }\; 0 \le x < \gamma\lambda \;\text{ and }\; \infty \;\text{ otherwise,} \\ &\tilde{\rho}_2(x) := \gamma\lambda^2/2 \;\text{ if }\; x \ge \gamma\lambda \;\text{ and }\; \infty \;\text{ otherwise.} \end{aligned}$$

Then $\rho(x) = \min_t \tilde{\rho}_t(x)$ for $x \ge 0$. Let D_k be the set of points at which f_k is differentiable. We then have, using Lemma 4 (iii) that

$$\begin{aligned} g_k(\theta_{k+1}) &= \min_{\theta_k:\theta_k \le \theta_{k+1}} \{\min_r \tilde{q}_{k,r}(\theta_k) + \min_t \tilde{\rho}_t(\theta_{k+1} - \theta_k)\} \\ &= \min[\widetilde{\min_{\theta_k \in D_k:\theta_k < \theta_{k+1}}} \min_{r,t} \{\tilde{q}_{k,r}(\theta_k) + \tilde{\rho}_t(\theta_{k+1} - \theta_k)\}, f_k(\theta_{k+1})] \\ &= \min[\min_{r,t} \widetilde{\min_{\theta_k \in D_k:\theta_k < \theta_{k+1}}} \{\tilde{q}_{k,r}(\theta_k) + \tilde{\rho}_t(\theta_{k+1} - \theta_k)\}, f_k(\theta_{k+1})], \end{aligned} \tag{16}$$

where $\widetilde{\min}$ denotes the minimum if it exists and ∞ otherwise. The fact that in the inner minimisation we are permitted to consider only points in D_k simplifies the form of

$$u_{k,r,t}(\theta_{k+1}) := \widetilde{\min_{\theta_k \in D_k:\theta_k < \theta_{k+1}}} \{\tilde{q}_{k,r}(\theta_k) + \tilde{\rho}_t(\theta_{k+1} - \theta_k)\}. \tag{17}$$

We show in Section A.1 of the Appendix that this is finite only on an interval and there takes the value of a quadratic function; coefficients for this function and the interval endpoints have closed form expressions that are elementary functions of the coefficients and intervals corresponding to $\tilde{q}_{k,r}$. With this, we have an explicit representation of g_k as the minimum of a collection of functions that are quadratic on intervals and ∞ everywhere else. Let us refer to these intervals (closed on the left) and corresponding quadratic functions as $J_{k,1}, \ldots, J_{k,n(k)}$ and $p_{k,1}, \ldots, p_{k,n(k)}$ respectively.

In order to produce a representation of f_{k+1} for use in future iterations, we must express g_k as a collection of quadratics defined on *disjoint* intervals. To this end, define for each $x \in \mathbb{R}$ the *active set at* x, $A(x) = \{r: x \in J_{k,r}\}$. Note that the endpoints of the intervals $J_{k,r}$ are the points where the active set changes and it is thus straightforward to determine $A(x)$ at each x. Let $r(x)$ be the index such that $g_k(x) = p_{k,r(x)}(x)$. For large negative values of x, $A(x)$ will contain a single index and for such x this must be $r(x)$. Consider also for each $r \in A(x) \setminus \{r(x)\}$, the horizontal coordinate x' of the first intersection beyond x (if it exists) between $p_{k,r}$ and $p_{k,r(x)}$. We refer to the collection of all such tuples (x', r) as the *intersection set at* x and denote it by $N(x)$. Given $r(x)$, $N(x)$ can be computed easily. The intersection set $N(x)$ then in turn helps to determine the smallest $x' > x$ where $r(x') \neq r(x)$ changes, that is the next knot of g_k beyond x, as we now explain. Suppose at a point x_{old}, we have computed $r_{\text{old}} = r(x_{\text{old}})$. We set $x_{\text{cur}} = x_{\text{old}}$ and perform the following.

1. Given $r(x_{\text{cur}})$, compute $N(x_{\text{cur}})$ and set $(x_{\text{int}}, r_{\text{int}}) = \arg\min_{(x,r)\in N(x_{\text{cur}})} x$.
2. If there are no changes in the active set between x_{cur} and x_{int}, we have found the next knot point at x_{int} and $r_{\text{int}} = r(x_{\text{int}})$.
3. If instead the active set changes, move x_{cur} to the leftmost change point. We have that $r(x) = r_{\text{old}}$ for $x \in [x_{\text{old}}, x_{\text{cur}})$. To determine if $r(x)$ changes at x_{cur}, we check if
   (i). r_{old} leaves the active set at x_{cur}, so $r_{\text{old}} \notin A(x_{\text{cur}})$, or
   (ii). some r_{new} enters the active set at x_{cur} and 'beats' r_{old}, so $r_{\text{new}} \in A(x_{\text{cur}}) \setminus A(x_{\text{old}})$ and $p_{k,r_{\text{new}}}(x_{\text{cur}} + \epsilon) < p_{k,r_{\text{old}}}(x_{\text{cur}} + \epsilon)$ for $\epsilon > 0$ sufficiently small.

   If either hold x_{cur} is a knot and $r(x_{\text{cur}})$ may be computed via $r(x_{\text{cur}}) = \arg\min_{r\in A(x_{\text{cur}})} p_{k,r}(x_{\text{cur}})$. If neither hold, we conclude that $r(x_{\text{cur}}) = r_{\text{old}}$ and go to step 1 once more.

Hence we can proceed from one knot of g_k to the next by comparing the values and intersections of a small collection of quadratic functions, and thereby form a piecewise quadratic representation of g_k in a finite number of steps. Figure 3 illustrates the steps outlined above. The pieces of b_k may be computed in a similar fashion.

We note there are several modifications that can speed up the algorithm: for example, for each r, $u_{k,r,2}$ (17) is a constant function where it is finite (see $p_{k,3}$ in the figure), and these can

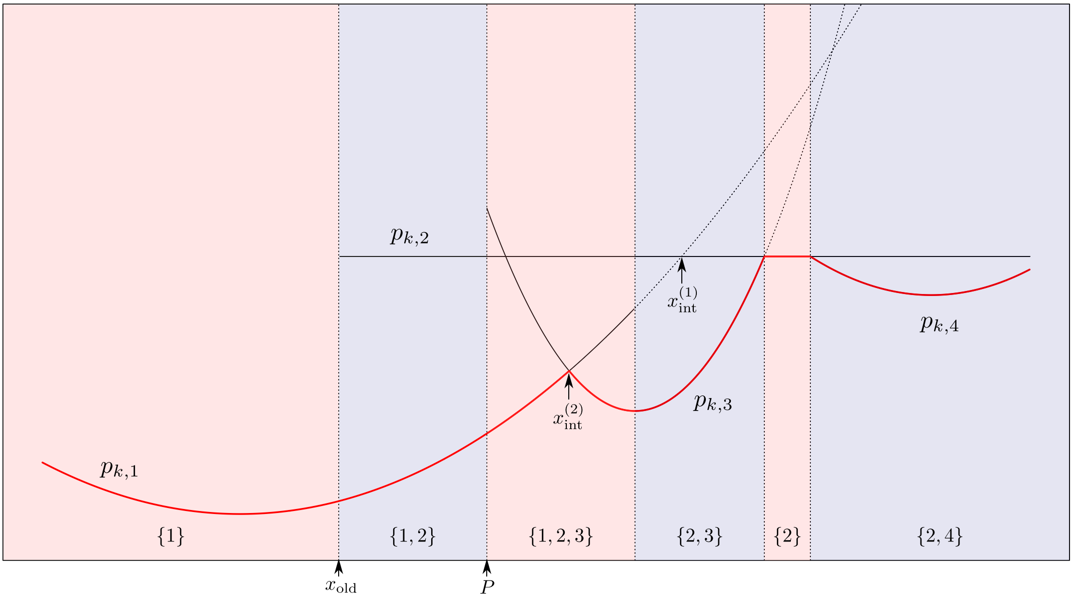


FIGURE 3 Illustration of the optimisation problem and our algorithm, to be interpreted with reference to steps 1, 2, 3 in the main text. Shading indicates regions where the active set, displayed at the bottom of the plot, is invariant, and vertical dotted lines signify changes. Dotted curves correspond to parts of quadratic functions $p_{k,l}$ lying outside their associated intervals $J_{k,l}$. At x_{old}, we have $r(x_{\text{old}}) = 1$, $A(x_{\text{old}}) = \{1, 2\}$ and $N(x_{\text{old}}) = \{(x_{\text{int}}^{(1)}, 2)\}$. Since the active set changes between x_{old} and $x_{\text{int}}^{(1)}$, we move x_{cur} to the first change point P and see neither (i) nor (ii) occur. We therefore return to step 1 and compute $N(x_{\text{cur}})$ which additionally contains $(x_{\text{int}}^{(2)}, 2)$. As the active set is unchanged between x_{cur} and $x_{\text{int}}^{(2)}$ we have determined the next knot point $x_{\text{int}}^{(2)}$ and minimising quadratic $p_{k,3}$

be dealt with more efficiently. For further details including pseudocode see Section A.2 of the Appendix.

In summary, our algorithm produces a piecewise quadratic representation of f_K, which we can minimise efficiently to obtain $\hat{\theta}_K$. We also have piecewise linear representations of functions $b_2, \ldots, b_K$ through which we may iteratively obtain $\hat{\theta}_k = b_{k+1}(\hat{\theta}_{k+1})$ for $k = K-1, \ldots, 1$.

It seems challenging to obtain meaningful bounds on the number of computations that must be performed at each stage of this process in terms of parameters of the data. However, to give an indication of the scalability of this algorithm, we ran a simple example with 3 true levels and found that with 50 categories the runtime was under 10^{-3} seconds; with 2000 categories it was still well under half a second. More details on computation time can be found in Sections 1.3 and 3.2 of the Supplementary material. In Section 1.4 of the Supplementary material, we describe an approximate version of the algorithm that can be used for fast computation in very large-scale settings.

3.2 | Multivariate model

Using our dynamic programming algorithm for the univariate problem, we can attempt to minimise the objective (9) for the multivariate problem using block coordinate descent. This has been shown empirically to be a successful strategy for minimising objectives for high-dimensional regression with nonconvex penalties such as the MCP (Breheny & Huang, 2011; Breheny & Huang, 2015; Mazumder et al., 2011), and we take this approach here. Considering the multivariate case, we iteratively minimise the objective Q over $\boldsymbol{\theta}_j := (\theta_{jk})_{k=1}^{K_j} \in \Theta_j$ keeping all other parameters fixed. Then for a given (γ,λ) and initial estimate $\hat{\boldsymbol{\theta}}^{(0)} \in \Theta$, we repeat the following until a suitable convergence criterion is met:

1. Initialise $m = 1$, and set for $i = 1, \ldots, n$

$$R_i = Y_i - \hat{\mu} - \sum_{l=1}^{p} \sum_{k=1}^{K_l} \hat{\theta}_{lk}^{(m-1)} \mathbb{1}_{\{X_{il}=k\}}.$$

2. For $j = 1, \ldots, p$, compute

$$R_i^{(j)} = R_i + \sum_{k=1}^{K_j} \hat{\theta}_{jk}^{(m-1)} \mathbb{1}_{\{X_{ij}=k\}} \quad \text{for each } i, \tag{18}$$

$$\begin{aligned} \hat{\boldsymbol{\theta}}_j^{(m)} &= \underset{\boldsymbol{\theta}_j \in \Theta_j}{\arg\min} \left\{ \frac{1}{2n} \sum_{i=1}^{n} \left(R_i^{(j)} - \sum_{k=1}^{K_j} \theta_{jk} \mathbb{1}_{\{X_{ij}=k\}} \right)^2 + \left(\sum_{k=1}^{K_j-1} \rho_j(\theta_{j(k+1)} - \theta_{j(k)}) \right) \right\} \\ R_i &= R_i^{(j)} - \sum_{k=1}^{K_j} \hat{\theta}_{jk}^{(m)} \mathbb{1}_{\{X_{ij}=k\}} \quad \text{for each } i. \end{aligned} \tag{19}$$

3. Increment $m \to m + 1$.

We define a blockwise optimum of Q to be any $\hat{\boldsymbol{\theta}} \in \Theta$, such that for each $j = 1, \ldots, p$,

$$\hat{\boldsymbol{\theta}}_j \in \underset{\boldsymbol{\theta}_j \in \Theta_j}{\arg\min} Q(\hat{\boldsymbol{\theta}}_1, \ldots, \hat{\boldsymbol{\theta}}_{j-1}, \boldsymbol{\theta}_j, \hat{\boldsymbol{\theta}}_{j+1}, \ldots, \hat{\boldsymbol{\theta}}_p). \tag{20}$$

This is equivalent to $\widehat{\boldsymbol{\theta}}$ being a fixed point of the block coordinate descent algorithm above. Provided $\gamma > 0$, Q is continuous in $\boldsymbol{\theta}$. As a consequence of Tseng (2001), Theorem 4.1 (c), provided the minimisers $\widehat{\boldsymbol{\theta}}_j^{(m)}$ in (19) are unique for all j and m (which will invariably be the case when the responses are realisations of continuous random variables; see Proposition 3), then all limit points of the sequence $(\widehat{\boldsymbol{\theta}}^{(m)})_{m=0}^{\infty}$ are blockwise optima.

3.3 | Practicalities

In practice the block coordinate descent procedure described above must be performed over a grid of (γ,λ) values to facilitate tuning parameter selection by cross-validation. In line with analogous recommendations for other penalised regression optimisation procedures (Breheny & Huang, 2011; Friedman et al., 2010), we propose, for each fixed γ, to iteratively obtain solutions for an exponentially decreasing sequence of λ values, warm starting each application of block coordinate descent at the solution for the previous λ. It is our experience that this scheme speeds up convergence and helps to guide the resulting estimates to statistically favourable local optima, as has been shown theoretically for certain nonconvex settings (Wang et al., 2014).

The grid of γ values can be chosen to be fairly coarse as the solutions appear to be less sensitive to this tuning parameter; in fact fixing $\gamma \in \{8, 32\}$ yields competitive performance across a range of settings (see Section 6). The choice $\gamma \downarrow 0$, which mimics the ℓ_0 penalty, has good statistical properties (see Theorem 5 and following discussion). However, the global optimum typically has a smaller basin of attraction and can be prohibitively hard to locate, particularly in low signal to noise ratio settings where larger γ tends to dominate.

4 | THEORY

In this section, we study the theoretical properties of SCOPE. Recall our model

$$Y_i = \mu^0 + \sum_{j=1}^{p} \sum_{k=1}^{K_j} \theta_{jk}^0 \mathbb{1}_{\{X_{ij}=k\}} + \varepsilon_i \tag{21}$$

for $i = 1, \ldots, n$, where $\boldsymbol{\theta}^0 \in \Theta$. We will assume the errors $(\varepsilon_i)_{i=1}^n$ have mean zero, are independent and sub-Gaussian with parameter σ. Let

$$\Theta_0 = \left\{ \boldsymbol{\theta} \in \Theta \colon \theta_{jk} = \theta_{jl} \text{ whenever } \theta_{jk}^0 = \theta_{jl}^0 \text{ for all } j \right\}$$

and define the *oracle least squares estimate*

$$\widehat{\boldsymbol{\theta}}^0 := \underset{\theta \in \Theta_0}{\arg\min} \frac{1}{2n} \sum_{i=1}^{n} \left(Y_i - \widehat{\mu} - \sum_{j=1}^{p} \sum_{k=1}^{K_j} \theta_{jk} \mathbb{1}_{\{X_{ij}=k\}} \right)^2 . \tag{22}$$

This is the least squares estimate of $\boldsymbol{\theta}^0$ with oracular knowledge of which categorical levels are fused in $\boldsymbol{\theta}^0$.

Note that in the case where the errors have equal variance v^2, the expected mean squared prediction error of $\hat{\boldsymbol{\theta}}^0$ satisfies

$$\mathbb{E}\left\{\frac{1}{n}\sum_{i=1}^{n}\left(\hat{\mu}-\mu^0+\sum_{j=1}^{p}\sum_{k=1}^{K_j}(\hat{\theta}^0_{jk}-\theta^0_{jk})\mathbb{1}_{\{X_{ij}=k\}}\right)^2\right\}\leq\frac{v^2}{n}\left(1+\sum_{j=1}^{p}(s_j-1)\right),$$

with equality when $\hat{\boldsymbol{\theta}}^0$ is unique.

Our results below establish conditions under which $\hat{\boldsymbol{\theta}}^0$ is a blockwise optimum (20) of the SCOPE objective function Q (9), or in the univariate case when this in fact coincides with SCOPE. The minimum differences between the signals defined for each j by

$$\Delta(\boldsymbol{\theta}^0_j):=\min_{k,l}\left\{|\theta^0_{jk}-\theta^0_{jl}|:\theta^0_{jk}\neq\theta^0_{jl}\right\}, \tag{23}$$

will play a key role. If all components of $\boldsymbol{\theta}^0_j$ are equal we take $\Delta(\boldsymbol{\theta}^0_j)$ to be ∞. We also introduce $n_{j,\min}=\min_k n_{jk}$,

$$n^0_{j,\min}=\min_k\sum_{l:\theta^0_{jl}=\theta^0_{jk}}n_{jl}\quad\text{and}\quad n^0_{j,\max}=\max_k\sum_{l:\theta^0_{jl}=\theta^0_{jk}}n_{jl};$$

these latter two quantities are the minimum and maximum number of observations corresponding to a set of fused levels in the jth predictor respectively.

4.1 | Univariate model

We first consider the univariate case, where as usual we will drop the subscript j for simplicity. The following result establishes conditions for recovery of the oracle least squares estimate (22).

Theorem 5 *Consider the model (21) in the univariate case with $p = 1$. Suppose there exists $\eta \in (0,1]$ such that $\eta/s\leq n^0_{j,\min}/n\leq n^0_{j,\max}/n\leq 1/\eta s$. Let $\gamma_*=\min\{\gamma,\eta s\}$ and $\gamma^*=\max\{\gamma,\eta s\}$. Suppose further that*

$$\Delta(\boldsymbol{\theta}^0)\geq 3\left(1+\sqrt{2}/\eta\right)\sqrt{\gamma\gamma^*}\,\lambda. \tag{24}$$

Then with probability at least

$$1-2\exp\left(-\frac{n_{\min}\eta s\gamma_*\lambda^2}{8\sigma^2}+\log(K)\right), \tag{25}$$

the oracle least squares estimate $\hat{\boldsymbol{\theta}}^0$ (22) is the global optimum of (9), so $\hat{\boldsymbol{\theta}}=\hat{\boldsymbol{\theta}}^0$.

For a choice of the tuning parameters (γ,λ) with $\gamma\leq\eta s$ and λ such that equality holds in (24), we have, writing $\Delta\equiv\Delta(\boldsymbol{\theta}^0)$, that $\hat{\boldsymbol{\theta}}=\hat{\boldsymbol{\theta}}^0$ with probability at least

$$1 - 2\exp\left(-c\eta^2 n_{\min}\Delta^2/\sigma^2 + \log(K)\right),$$

where c is an absolute constant. The quantity η reflects how equal the number of observations in the true fused levels are: in settings where the prevalences of the underlying true levels are roughly equal, we would expect this to be closer to 1.

Consider now an asymptotic regime where K, s and $1/\Delta$ are allowed to diverge with n, $n_{\min} \asymp n/K$, so all levels have roughly the same prevalence, and η is bounded away from zero, so all true underlying levels also have roughly the same prevalence. Then in order for $\hat{\boldsymbol{\theta}} = \hat{\boldsymbol{\theta}}^0$ with high probability, we require $\Delta \gtrsim \sigma\sqrt{K\log(K)/n}$. This requirement cannot be weakened for any estimator; this fact comes as a consequence of minimax lower bounds on mis-clustering errors in Gaussian mixture models (Lu & Zhou, 2016, Theorem 3.3).

We remark that our result here concerning properties of the global minimiser of our objective is very different from existing results on local minimisers of objectives involving all-pairs-type penalties. For example, in the setting above where $K = n$, Theorem 2 of Ma and Huang (2017) gives that provided $s = o(n^{1/3}(\log n)^{-1/3})$ and $\Delta \gg \sigma s^{3/2} n^{-1/2}\sqrt{\log(n)}$, there exists a sequence of local minimisers converging to the oracle least squares estimate with high probability. This is significantly weaker than the condition $\Delta \gtrsim \sigma\sqrt{\log(n)}$ required for any estimator to recover oracle least squares in this setting, illustrating the substantial difference between results on local and global optima here.

4.2 | Multivariate model

When the number of variables is $p > 1$, models can become high dimensional, with ordinary least squares estimation failing to provide a unique solution. We will, however, assume that the solution for $\boldsymbol{\theta} \in \Theta_0$ to

$$\sum_{j=1}^{p}\sum_{k=1}^{K_j}\theta^0_{jk}\mathbb{1}_{\{X_{ij}=k\}} = \sum_{j=1}^{p}\sum_{k=1}^{K_j}\theta_{jk}\mathbb{1}_{\{X_{ij}=k\}}$$

is unique, which occurs if and only if the oracle least squares estimate (22) is unique. In this case, we note that $\hat{\boldsymbol{\theta}}^0 = AY$ for a fixed matrix A. A necessary condition for this is that $\sum_j (s_j - 1) < n$.

Our result below provides a bound on the probability that the oracle least squares estimate is a blockwise optimum of the SCOPE objective (9) with $\rho_j = \rho_{\gamma_j,\lambda_j}$. This is much more meaningful than an equivalent bound for $\hat{\boldsymbol{\theta}}^0$ to be a local optimum as the number of local optima will be enormous. In general though there may be several blockwise optima, and it seems challenging to obtain a result giving conditions under which our blockwise coordinate descent procedure is guaranteed to converge to $\hat{\boldsymbol{\theta}}^0$. Our empirical results (Section 6), however, show that the fixed points computed in practice tend to give good performance.

Theorem 6 *Consider the model (21) and assume* $\hat{\boldsymbol{\theta}}^0 = AY$. *Suppose that there exists* $\eta \in (0, 1]$ *such that* $\eta/s_j \le n^0_{j,\min}/n \le n^0_{j,\max}/n \le 1/\eta s_j$ *for all* $j = 1, \ldots, p$. *Let* $\gamma_{*j} = \min\{\gamma_j, \eta s_j\}$ *and* $\gamma_j^* = \max\{\gamma_j, \eta s_j\}$. *Further suppose that*

$$\Delta(\boldsymbol{\theta}_j^0) \ge 3\left(\frac{4}{3} + \frac{\sqrt{2}}{\eta}\right)\sqrt{\gamma_j\gamma_j^*}\,\lambda_j. \tag{26}$$

Then letting $c_{\min}\colon = (\max_l(AA^T)_{ll})^{-1}$, with probability at least

$$1 - 4\sum_{j=1}^{p}\exp\left(-\frac{(n_{j,\min}\wedge c_{\min})\eta\gamma_{*j}s_j\lambda_j^2}{8\sigma^2} + \log(K_j)\right), \tag{27}$$

the oracle least squares estimate $\hat{\boldsymbol{\theta}}^0$ is a blockwise optimum of (9).

Now suppose $\gamma_j \leq \eta s_j$ and λ_j are such that equality holds in Equation (26) for all j. Then writing $K_{\max} = \max_j K_j$, $n_{\min} = \min_j n_{j,\min}$ and $\Delta_{\min} = \min_j \Delta(\boldsymbol{\theta}_j^0)$, we have that $\hat{\boldsymbol{\theta}}^0$ is a blockwise optimum of (9) with probability at least

$$1 - 4\exp\left(-c\eta^2(n_{\min}\wedge c_{\min})\Delta_{\min}^2/\sigma^2 + \log(K_{\max}p)\right),$$

where c is an absolute constant. Consider now an analogous asymptotic regime to that described in the previous section for the univariate case. Specifically assume $n_{\min} \asymp n/K_{\max}$ and $c_{\min} \gtrsim n_{\min}$ for simplicity. We then see that in order for $\hat{\boldsymbol{\theta}}^0$ to be a blockwise optimum with high probability, it is sufficient that $\Delta_{\min} \gtrsim \sigma\sqrt{K_{\max}\log(K_{\max}p)/n}$.

5 | EXTENSIONS

In this section, we describe some extensions of our SCOPE methodology.

Continuous covariates. If some of the covariates are continuous rather than categorical, we can apply any penalty function of choice to these, and perform a regression by optimising the sum of a least squares objective, our SCOPE penalty and these additional penalty functions, using (block) coordinate descent.

For example, consider the model (1) with the addition of d continuous covariates. Let $Z \in \mathbb{R}^{n\times d}$ be the centred design matrix for these covariates with ith row $Z_i \in \mathbb{R}^d$. One can fit a model with SCOPE penalising the categorical covariates, and the Lasso with tuning parameter $\alpha > 0$ penalising the continuous covariates, resulting in the following objective over $\boldsymbol{\beta} \in \mathbb{R}^d$ and $\boldsymbol{\theta} \in \Theta$:

$$\frac{1}{2n}\sum_{i=1}^{n}\left(Y_i - \hat{\mu} - Z_i^T\boldsymbol{\beta} - \sum_{j=1}^{p}\sum_{k=1}^{K_j}\theta_{jk}\mathbb{1}_{\{X_{ij}=k\}}\right)^2 + \alpha\|\boldsymbol{\beta}\|_1 + \sum_{j=1}^{p}\sum_{k=1}^{K_j-1}\rho_j(\theta_{j(k+1)} - \theta_{j(k)}).$$

This sort of integration of continuous covariates is less straightforward when attempting to use tree-based methods to handle categorical covariates, for example.

Generalised linear models. Sometimes a generalised linear model may be appropriate. Although a quadratic loss function is critical for our exact optimisation algorithm described in Section 3.1, we can iterate local quadratic approximations to the loss term in the objective and minimise this. This results in a proximal Newton algorithm and is analogous to the standard approach for solving ℓ_1-penalised generalised linear models (Friedman et al., 2010, Section 3). An implementation of this scheme in the case of logistic regression for binary responses is available in the accompanying R package `CatReg`. We remark that when computing logistic regression models with a SCOPE penalty it is advisable to use a larger value of γ than with a continuous response to aid convergence of the proximal Newton step; we recommend a default setting of $\gamma = 100$. In

Section 6.2 we use the approach described above to perform a logistic regression using SCOPE on US census data.

Hierarchical categories. Often certain predictors may have levels that are effectively subdivisions of the levels of other predictors. Examples include category of item in e-commerce or geographical data with predictors for continent, countries and district. For simplicity, we will illustrate how such settings may be dealt with by considering a case with two predictors, but this may easily be generalised to more complex hierarchical structures. Suppose there is a partition $G_1 \cup \cdots \cup G_{K_1}$ of $\{1, \ldots, K_2\}$ such that for all $k = 1, \ldots, K_1$,

$$X_{i2} \in G_k \Longrightarrow X_{i1} = k,$$

so the levels of the second predictor in G_k represent subdivisions of kth level of the first predictor. Let $K_{2k} := |G_k|$ and let $\boldsymbol{\theta}_{2k}$ refer to the subvector $(\theta_{2l})_{l \in G_k}$ for each $k = 1, \ldots, K_1$, so components of $\boldsymbol{\theta}_{2k}$ are the coefficients corresponding to the levels in G_k. Also let $\theta_{2k(r)}$ denote the rth order statistic within $\boldsymbol{\theta}_{2k}$. It is natural to encourage fusion among levels within G_k more strongly than for levels in different elements of the partition. To do this we can modify our objective function so the penalty takes the form

$$\sum_{k=1}^{K_1 - 1} \rho_1(\theta_{1(k+1)} - \theta_{1(k)}) + \sum_{k=1}^{K_1} \sum_{l=1}^{K_{2k} - 1} \rho_{2k}(\theta_{2k(l+1)} - \theta_{2k(l)}).$$

We furthermore enforce the identifiability constraints that

$$\sum_{l=1}^{K_1} n_{1l}\theta_{1l} = 0 \text{ and } \sum_{l \in G_k} n_{2l}\theta_{2l} = 0 \text{ for all } k = 1, \ldots, K.$$

As well as yielding the desired shrinkage properties, an additional advantage of this approach is that the least squares criterion is separable in $\boldsymbol{\theta}_{21}, \ldots, \boldsymbol{\theta}_{2K_1}$ so the blockwise update of $\boldsymbol{\theta}_2$ can be performed in parallel. This can lead to a substantial reduction in computation time if K_2 is large.

6 | NUMERICAL EXPERIMENTS

In this section we explore the empirical properties of SCOPE. We first present results on the performance on simulated data, and then in Sections 6.2–6.5 present analyses and experiments on US census data, insurance data and COVID-19 modelling data.

We denote SCOPE with a specific choice of γ as SCOPE-γ, and write SCOPE-CV to denote SCOPE with a cross-validated choice of γ. SCOPE solutions are computed using our R (R Core Team, 2020) package `CatReg` (Stokell, 2021), using fivefold cross-validation to select λ for all examples except those in Section 6.5. We compare SCOPE to linear or logistic regression where appropriate and a range of existing methods, including CAS-ANOVA (Bondell & Reich, 2009) (4), and an adaptive version where the weights $w_{j,kl}$ are multiplied by a factor proportional to the $|\hat{\theta}_{jk}^{\text{init}} - \hat{\theta}_{jl}^{\text{init}}|^{-1}$, where $\hat{\boldsymbol{\theta}}^{\text{init}}$ is an initial CAS-ANOVA estimate. For these methods the tuning parameter λ was also selected by fivefold cross-validation. As well as this, we include DMR (Maj-Kańska et al., 2015) and Bayesian effect fusion (BEF) (Pauger & Wagner, 2019) in some experiments. With the former, models were fitted using `DMRnet` (Prochenka-Sotys & Pokarowski, 2018) and selected by fivefold cross-validation where possible; otherwise an information criterion was used. With BEF, coefficients were modelled with a Gaussian mixture model with posterior mean estimated using 1000 samples using `effectFusion`

(Pauger et al., 2019). We also include comparison to the tree-based approaches CART (Breiman et al., 1984) and random forests (RF) (Breiman, 2001). Lastly, in some experiments, models were also fitted using the Lasso (Tibshirani, 1996). CART was implemented using `rpart` (Therneau & Atkinson, 2019) with pruning according to the one standard error rule. Random forests and Lasso were implemented using the default settings in `randomForest` (Liaw & Wiener, 2002) and `glmnet` (Friedman et al., 2010) packages respectively. For full details of the specific versions of these methods and software used in the numerical experiments, see Section 3.1 of the Supplementary material.

6.1 | Simulations

We simulated data according to the model (1) with the covariates X_{ij} generated randomly in the following way. We first drew $(W_{ij})_{j=1}^{p}$ from a multivariate $\mathcal{N}_p(0, \Sigma)$ distribution where the covariance matrix Σ had ones on the diagonal. The off-diagonal elements of Σ were chosen such that $U_{ij} := \Phi^{-1}(W_{ij})$ had $\mathrm{corr}(U_{ij}, U_{ik}) = \rho$ for $j \neq k$. The marginally uniform U_{ij} were then quantised this to give $X_{ij} = \lceil 24U_{ij} \rceil$, so the number of levels $K_j = 24$.

The errors ε_i were independently distributed as $\mathcal{N}(0, \sigma^2)$. The performance of SCOPE and competitor methods was measured using mean squared prediction error on 10^5 new (noiseless) observations generated in the same way as the training data, and final results are averages over 500 draws of training and test data. We considered various settings of $(n, p, \rho, \boldsymbol{\theta}^0, \sigma^2)$ below with low-dimensional and high-dimensional scenarios considered in Sections 6.1.1 and 6.1.2 respectively. The coefficient vectors for each experiment are specified up to an additive constant, which is required to satisfy the identifiability condition (8).

We measured predictive performance by the mean squared prediction error (MSPE) given by

$$\text{MSPE} := \mathbb{E}_x\{g(x) - \hat{g}(x)\}^2, \tag{28}$$

where g is the true regression function, $\hat{g}$ an estimate, and the expectation is taken over the covariate vector x.

6.1.1 | Low-dimensional experiments

Results are presented for three settings with $n = 500$, $p = 10$ given below.

1. $\boldsymbol{\theta}_j^0 = (\overbrace{-3, \ldots, -3}^{10 \text{ times}}, \overbrace{0, \ldots, 0}^{4 \text{ times}}, \overbrace{3, \ldots, 3}^{10 \text{ times}})$ for $j = 1, 2, 3$, and $\boldsymbol{\theta}_j^0 = 0$ otherwise; $\rho = 0$.
2. $\boldsymbol{\theta}_j^0 = (\overbrace{-3, \ldots, -3}^{8 \text{ times}}, \overbrace{0, \ldots, 0}^{8 \text{ times}}, \overbrace{3, \ldots, 3}^{8 \text{ times}})$ for $j = 1, 2, 3$, and $\boldsymbol{\theta}_j^0 = 0$ otherwise; $\rho = 0$.
3. As Setting 1, but with $\rho = 0.8$.

Each of these experiments were performed with noise variance $\sigma^2 = 1, 6.25, 25$ and 100. Note that the variance of the signal varies across each setting, and signal-to-noise ratio (SNR) for each experiment is displayed in Table 1. Methods included for comparison were SCOPE-8, SCOPE-32, SCOPE-CV, linear regression, vanilla and adaptive CAS-ANOVA, DMR, Bayesian effect fusion, CART and random forests. Also included are the results from the oracle least squares estimator (22).

TABLE 1 Mean squared prediction errors (and standard deviations thereof) of various methods on the settings described.

	Setting 1			
σ^2:	**1**	**6.25**	**25**	**100**
SNR:	**4.7**	**1.9**	**0.95**	**0.47**
SCOPE-8	**0.014** (0.0)	0.450 (0.5)	4.571 (1.0)	12.936 (2.8)
SCOPE-32	0.018 (0.0)	0.878 (0.6)	4.151 (0.9)	**12.356** (2.1)
SCOPE-CV	0.015 (0.0)	**0.407** (0.4)	**4.120** (0.9)	12.513 (2.5)
Linear regression	0.851 (0.1)	5.317 (0.7)	21.503 (2.7)	86.745 (10.7)
Oracle least squares	0.014 (0.0)	0.091 (0.1)	0.333 (0.2)	1.405 (0.8)
CAS-ANOVA	0.617 (0.3)	1.602 (0.3)	5.448 (1.0)	14.814 (2.2)
Adaptive CAS-ANOVA	0.135 (0.1)	0.880 (0.4)	5.076 (1.2)	22.896 (4.7)
DMR	**0.014** (0.0)	0.448 (0.4)	4.884 (1.4)	18.394 (3.6)
BEF	0.020 (0.0)	2.209 (1.1)	6.297 (1.8)	21.927 (2.3)
CART	3.844 (0.4)	5.099 (0.9)	13.219 (2.1)	22.431 (1.2)
RF	9.621 (0.5)	10.944 (0.5)	13.217 (0.7)	16.344 (0.9)
	Setting 2			
σ^2:	**1**	**6.25**	**25**	**100**
SNR:	**4.2**	**1.7**	**0.85**	**0.42**
SCOPE-8	**0.015** (0.0)	**0.285** (0.3)	6.775 (0.9)	12.697 (2.3)
SCOPE-32	0.019 (0.0)	0.655 (0.4)	5.026 (1.0)	**12.037** (2.0)
SCOPE-CV	0.016 (0.0)	0.292 (0.3)	**5.005** (1.1)	12.444 (2.5)
Linear regression	0.869 (0.1)	5.406 (0.7)	21.216 (2.5)	85.439 (10.9)
Oracle least squares	0.014 (0.0)	0.088 (0.0)	0.336 (0.2)	1.532 (0.8)
CAS-ANOVA	1.483 (0.4)	1.626 (0.3)	5.466 (1.0)	13.421 (2.2)
Adaptive CAS-ANOVA	0.134 (0.1)	0.912 (0.3)	5.535 (1.2)	22.213 (4.9)
DMR	0.016 (0.0)	0.409 (0.4)	6.430 (1.4)	17.457 (2.1)
BEF	0.019 (0.0)	1.055 (0.9)	8.183 (2.0)	18.236 (1.5)
CART	5.530 (0.6)	7.457 (0.9)	13.280 (1.8)	18.198 (0.7)
RF	8.947 (0.3)	9.747 (0.4)	11.249 (0.6)	13.646 (0.8)
	Setting 3			
σ^2:	**1**	**6.25**	**25**	**100**
SNR:	**7.3**	**2.9**	**1.5**	**0.73**
SCOPE-8	**0.015** (0.0)	0.967 (0.7)	5.060 (1.3)	14.555 (2.9)
SCOPE-32	0.018 (0.0)	0.713 (0.4)	3.580 (0.8)	**9.721** (1.9)
SCOPE-CV	0.022 (0.1)	0.582 (0.3)	**3.368** (0.9)	10.168 (2.6)
Linear regression	0.879 (0.1)	5.485 (0.7)	21.987 (2.7)	87.820 (11.9)
Oracle least squares	0.014 (0.0)	0.092 (0.0)	0.362 (0.2)	1.488 (1.0)

(Continues)

TABLE 1 (Continued)

	Setting 3			
$\boldsymbol{\sigma^2}$**:**	**1**	**6.25**	**25**	**100**
SNR:	**7.3**	**2.9**	**1.5**	**0.73**
CAS-ANOVA	0.710 (0.2)	1.601 (0.3)	4.732 (0.9)	12.708 (2.1)
Adaptive CAS-ANOVA	0.189 (0.2)	0.701 (0.3)	3.705 (1.0)	16.186 (3.6)
DMR	**0.015** (0.0)	**0.553** (0.5)	5.730 (1.9)	18.594 (4.5)
BEF	0.019 (0.0)	1.716 (0.9)	8.143 (2.6)	26.923 (7.0)
CART	4.336 (0.6)	5.685 (1.0)	9.910 (1.7)	18.543 (2.2)
RF	4.039 (0.3)	5.673 (0.5)	9.157 (0.9)	13.766 (1.7)

The best results for each setting are in bold.

Results are shown in Table 1 and further details are given in Section 3.2.1 of the Supplementary material. Across all experiments, SCOPE with a cross-validated choice of γ exhibits prediction performance at least as good as the optimal approaches, and in all but the lowest noise settings performs better than the other methods that were included. In these exceptions, we see that fixing γ to be a small value (corresponding to high concavity) provides leading performance.

In these low noise settings, we see that the methods based on first estimating the clusterings of the levels and then estimating the coefficients without introducing further shrinkage, such as DMR or Bayesian effect Fusion, perform well. However, they tend to struggle when the noise is larger. In contrast the tree-based methods perform poorly in low noise settings but exhibit competitive performance in high noise settings.

6.1.2 | High-dimensional experiments

We considered eight settings as detailed below, each with $n = 500$, $p = 100$ and simulated 500 times.

1. $\boldsymbol{\theta}_j^0 = (\overbrace{-2,\ \dots,\ -2}^{8\text{ times}}, \overbrace{0,\ \dots,\ 0}^{8\text{ times}}, \overbrace{2,\ \dots,\ 2}^{8\text{ times}})$ for $j = 1, 2, 3$, $\boldsymbol{\theta}_j^0 = (\overbrace{-2,\ \dots,\ -2}^{10\text{ times}}, \overbrace{0,\ \dots,\ 0}^{4\text{ times}}, \overbrace{2,\ \dots,\ 2}^{10\text{ times}})$ for $j = 4, 5, 6$, and $\boldsymbol{\theta}_j^0 = 0$ otherwise; $\rho = 0$ and $\sigma^2 = 50$.

2. As Setting 1, but with $\rho = 0.5$.

3. $\boldsymbol{\theta}_j^0 = (\overbrace{-2,\ \dots,\ -2}^{8\text{ times}}, \overbrace{0,\ \dots,\ 0}^{8\text{ times}}, \overbrace{2,\ \dots,\ 2}^{8\text{ times}})$ for $j = 1, 2, 3$, $\boldsymbol{\theta}_j^0 = (\overbrace{-2,\ \dots,\ -2}^{16\text{ times}}, \overbrace{3,\ \dots,\ 3}^{8\text{ times}})$ for $j = 4, 5, 6$, and $\boldsymbol{\theta}_j^0 = 0$ otherwise; $\rho = 0.5$ and $\sigma^2 = 100$.

4. $\boldsymbol{\theta}_j^0 = (\overbrace{-2,\ \dots,\ -2}^{5\text{ times}}, \overbrace{-1,\ \dots,\ -1}^{5\text{ times}}, \overbrace{0,\ \dots,\ 0}^{4\text{ times}}, \overbrace{1,\ \dots,\ 1}^{5\text{ times}}, \overbrace{2,\ \dots,\ 2}^{5\text{ times}})$ for $j = 1, \dots, 5$, and $\boldsymbol{\theta}_j^0 = 0$ otherwise; $\rho = 0$ and $\sigma^2 = 25$.

5. $\boldsymbol{\theta}_j^0 = (\overbrace{-2,\ \dots,\ -2}^{16\text{ times}}, \overbrace{3,\ \dots,\ 3}^{8\text{ times}})$ for $j = 1, \dots, 25$, and $\boldsymbol{\theta}_j^0 = 0$ otherwise; $\rho = 0$ and $\sigma^2 = 1$.

6. As Setting 5, but with $\rho = 0.5$.

7. $\boldsymbol{\theta}_j^0 = (\overbrace{-2, \ldots, -2}^{4 \text{ times}}, \overbrace{0, \ldots, 0}^{12 \text{ times}}, \overbrace{2, \ldots, 2}^{8 \text{ times}})$ for $j = 1, \ldots, 10$, and $\boldsymbol{\theta}_j^0 = 0$ otherwise; $\rho = 0$ and $\sigma^2 = 25$.

8. $\boldsymbol{\theta}_j^0 = (\overbrace{-3, \ldots, -3}^{6 \text{ times}}, \overbrace{-1, \ldots, -1}^{6 \text{ times}}, \overbrace{1, \ldots, 1}^{6 \text{ times}}, \overbrace{3, \ldots, 3}^{6 \text{ times}})$ for $j = 1, \ldots, 5$, and $\boldsymbol{\theta}_j^0 = 0$ otherwise; $\rho = 0$ and $\sigma^2 = 25$.

Models were fitted using SCOPE-8, SCOPE-32, SCOPE-CV, DMR, CART, Random forests and the Lasso. Table 2 gives the mean squared prediction errors across each of the settings.

As well as prediction performance, it is interesting to see how the methods perform in terms of variable selection performance. With categorical covariates, there are two potential ways of evaluating this. The first is to consider the number of false positives and false negatives across the $p = 100$ categorical variables, defining a variable j to have been selected if $\widehat{\boldsymbol{\theta}}_j \neq 0$. These results are shown in Table 3. This definition of a false positive can be considered quite conservative; typically one can find that often the false signal variables have only two levels, each with quite small coefficients. This means that the false positive rate can increase substantially with only a small increase in the dimension of the estimated linear model.

The second is to see within the signal variables (i.e. the j for which $\boldsymbol{\theta}_j^0 \neq 0$), how closely the estimated clustering resembles the true structure. To quantify this, we use the *adjusted Rand index* (Hubert & Arabie, 1985). This is the proportion of all pairs of observations that are either (i) in different true clusters and different estimated clusters, or (ii) in the same true cluster and estimated cluster;

TABLE 2 Mean squared prediction errors (and standard deviations thereof) of each of the methods in the 8 high-dimensional settings considered.

Setting:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
SNR:	**0.6**	**1.0**	**1.0**	**0.64**	**12**	**36**	**0.87**	**1.0**
SCOPE-8	14.319 (2.0)	15.445 (2.9)	30.597 (5.6)	7.254 (1.2)	96.538 (25.0)	7.960 (23.2)	15.867 (1.4)	11.028 (1.6)
SCOPE-32	**14.009** (1.6)	**10.780** (1.6)	**21.841** (3.4)	7.256 (0.9)	65.344 (13.4)	0.107 (0.0)	14.867 (1.2)	11.218 (1.4)
SCOPE-CV	14.026 (1.7)	10.843 (1.8)	22.004 (3.9)	**7.191** (1.0)	**54.030** (19.2)	**0.084** (0.0)	**14.865** (1.3)	**10.941** (1.5)
Oracle LSE	5.044 (0.6)	5.130 (0.6)	2.664 (1.0)	1.09 (0.3)	0.054 (0.0)	0.055 (0.0)	1.087 (0.3)	0.799 (0.3)
DMR	18.199 (1.4)	22.627 (4.4)	42.979 (9.2)	9.645 (1.2)	139.095 (4.3)	213.691 (35.7)	19.298 (0.8)	11.737 (2.4)
CART	18.146 (0.5)	31.235 (3.6)	58.73 (6.6)	10.466 (0.3)	139.35 (2.1)	614.739 (42.8)	19.021 (0.4)	23.775 (1.5)
RF	16.181 (0.6)	16.345 (1.4)	31.561 (2.6)	9.053 (0.4)	128.618 (2.2)	264.374 (14.4)	17.224 (0.4)	19.783 (0.7)
Lasso	18.136 (0.5)	24.839 (1.3)	48.162 (2.5)	10.473 (0.4)	135.375 (5.0)	154.656 (7.8)	18.886 (0.6)	23.813 (1.6)

The best results for each setting are in bold.

TABLE 3 (False positive rate)/(False negative rate) of linear modelling methods considered in the high-dimensional settings

Setting:	1	2	3	4	5	6	7	8
SCOPE-8	0.02/0.35	0.04/0.23	0.04/0.25	0.02/0.15	0.02/0.23	0.02/0.01	0.02/0.35	0.01/0.00
SCOPE-32	0.14/0.15	0.30/0.02	0.30/0.02	0.15/0.04	0.52/0.00	0.00/0.00	0.21/0.08	0.21/0.00
SCOPE-CV	0.12/0.20	0.30/0.02	0.29/0.03	0.12/0.07	0.59/0.00	0.00/0.00	0.21/0.11	0.09/0.00
DMR	0.00/0.86	0.00/0.44	0.00/0.47	0.00/0.62	0.00/0.91	0.03/0.60	0.00/0.88	0.00/0.02
Lasso	0.01/0.88	0.00/1.00	0.00/1.00	0.01/0.83	0.00/0.98	0.00/1.00	0.00/0.91	0.00/0.90

TABLE 4 Average adjusted Rand index among true signal variables for the high-dimensional settings

Setting:	1	2	3	4	5	6	7	8
SCOPE-8	0.23	0.36	0.38	0.15	0.39	0.96	0.13	0.29
SCOPE-32	0.29	0.46	0.48	0.19	0.56	1.00	0.17	0.34
SCOPE-CV	0.27	0.45	0.46	0.18	0.56	1.00	0.17	0.31
DMR	0.04	0.20	0.23	0.06	0.04	0.19	0.03	0.28
Lasso	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

this is then corrected to ensure that its value is zero when exactly one of the clusterings is 'all-in-one'. In Table 4 we report the average adjusted Rand index over the true signal variables in each setting.

Further details can be found in Section 3.2.2 of the Supplementary material. In particular we include a table with the distribution of cross-validated choices of γ (from a grid $\{4, 8, 16, 32, 64\}$) for each experimental setting. Note that a choice of $\gamma = 4$ is close to the setting of $\gamma = 3$ recommended in Zhang (2010), although the problem of categorical covariates is very different in nature than the vanilla variable selection problem considered there. Our results there suggest that for SCOPE, a larger value of γ is preferable across a range of settings, which is also visible in the comparison between $\gamma = 8$ and $\gamma = 32$ in Table 2.

Across all the settings in this study, SCOPE performs better than any of the other methods included. This is regardless of which of the three γ regimes is chosen, although cross-validating γ gives the strongest performance overall. Comparing the results for $\gamma = 8$ and $\gamma = 32$ suggests that a larger (low-concavity) choice of γ is preferable for higher-dimensional settings. In setting 6, we see from Tables 3 and 4 that SCOPE obtains the true underlying groupings of the coefficients and obtains the oracle least squares estimate in every case, giving these striking results. This is also achieved for some of the experiments in setting 5. In contrast, DMR, which initially applies a group Lasso (Yuan & Lin, 2006) to screen the categorical variables and give a low-dimensional model, necessarily misses some signal variables in this first stage and hence struggles here.

6.2 | Adult dataset analysis

The *Adult dataset*, available from the UCI Machine Learning Repository (Dua & Graff, 2019), contains a sample of 45,222 observations based on information from the 1994 US census. The binary response variable is 0 if the individual earns at most $50,000 a year, and 1 otherwise. There are 2 continuous and eight categorical variables; some such as 'native country' have large numbers of levels, bringing the total dimension to 93. An advantage of using SCOPE here over black-box predictive tools such as Random forests is the interpretability of the fitted model.

In Table 5, we show the 25-dimensional fitted model. Within the Education category, we see that six distinct levels have been identified. These agree almost exactly with the stratification one would expect, with all school dropouts before 12th grade being grouped together at the lowest level Figure 4.

Here we assess performance in the challenging setting when the training set is quite small by randomly selecting 1% (452) of the total observations for training, and using the remainder as a test set. Any observations containing levels not in the training set were removed. Models were fitted with SCOPE-100, SCOPE-250, logistic regression, vanilla and adaptive CAS-ANOVA, DMR, Bayesian effect fusion, CART and random forests.

We see in Figure 4 that both SCOPE-100 and SCOPE-250 are competitive, with CART and Random forests also performing well, although the latter two include interactions in their fits. CAS-ANOVA also performs fairly well, the misclassification error is larger that for both versions of SCOPE, and the average fitted model size is larger (see Table 6).

6.3 | Adult dataset with artificially split levels

To create a more challenging example, we artificially created additional levels in the *Adult dataset* as follows. For each categorical variable we recursively selected a level with probability proportional to its prevalence in the data and then split it into two by appending "−0" or "−1" to the level for each observation independently and with equal probabilities. We repeated this until the total number of levels reached m times the original number of levels for that variable for $m = 2, 3, 4$. This process simulates for example responses to a survey, where different respondents might answer 'US', 'U.S.', 'USA', 'U.S.A.', 'United States' or 'United States of America' to a question, which would naively all be treated as different answers.

We used 2.5% (1130) of the observations for training and the remainder for testing and applied SCOPE with $\gamma = 100$ and logistic regression. Results were averaged over 250 training and test splits. Figure 5 shows that as the number of levels increases, the misclassification error of SCOPE increases only slightly and the fitted model dimension remains almost unchanged, whereas both increase with m for logistic regression.

6.4 | Insurance data example

The Prudential Life Insurance Assessment challenge was a prediction competition run on Kaggle (2015). By more accurately predicting risk, the burden of extensive tests and check-ups for life insurance policyholders could potentially be reduced. For this experiment, we use the training set that was provided for entrants of the competition.

We removed a small number of variables due to excessive missingness, leaving five continuous variables and 108 categorical variables, most with two or three levels but with some in the hundreds (and the largest with 579 levels). Rather than using the response from the original dataset, which is ordinal, to better suit the regression setting we are primarily concerned with in this work, we artificially generated a continuous response. To construct this signal, firstly 10 of the categorical variables were selected at random, with probability proportional to the number of levels. For the jth of these, writing K_j for the number of levels, we set $s_j := \lfloor 2 + \frac{1}{2}\log K_j \rfloor$ and assigned each level a coefficient in $1, \ldots, s_j$ uniformly at random,

TABLE 5 Coefficients of SCOPE model trained on the full dataset. Here, $\gamma = 100$ and λ was selected by fivefold cross-validation (with cross-validation error of 16.82%). Countries, aside from those in the United Kingdom, are referred to by their (possibly historical) internet top-level domains

Variable	Coefficient	Levels
Intercept	−3.048	—
Age	0.027	—
Hours per week	0.029	—
Work class	0.378	Federal government, Self-employed (incorporated)
	0.058	Private
	−0.143	Local government
	−0.434	Self-employed (not incorporated), State government, Without pay
Education level	1.691	Doctorate, Professional school
	1.023	Master's
	0.646	Bachelor's
	−0.132	Associate's (academic), Associate's (vocational), Some college (non-graduate)
	−0.546	12th, High school grad
	−1.539	Preschool, 1st–4th, 5th–6th, 7th–8th, 9th, 10th, 11th
Marital status	0.059	Divorced, Married (armed forces spouse), Married (civilian spouse), Married (absent spouse), Separated, Widowed
	−0.476	Never married
Occupation	0.560	Executive/Managerial
	0.311	Professional/Specialty, Protective service, Tech support
	−0.003	Armed forces, Sales
	−0.168	Admin/Clerical, Craft/Repair
	−0.443	Machine operative/inspector, Transport
	−1.107	Farming/Fishing, Handler/Cleaner, Other service, Private house servant
Relationship*	1.498	Wife
	0.332	Husband
	−1.220	Not in family
	−1.482	Unmarried, Other relative
	−2.144	Own child
Race	0.013	White
	0.008	Asian/Pacific islander, Other
	−0.182	Native-American/Inuit, Black
Sex	0.139	Male
	−0.619	Female
Native country	0.018	KH, CA, CU, ENG, FR, DE, GR, HT, HN, HK, HU, IN, IR, IE, IT, JM, JP, PH, PL, PT, PR, TW, US, YU
	−0.882	CN, CO, DO, EC, SV, GT, NL, LA, MX, NI, GU-VI-etc, PE, SCT, ZA, TH, TT, VN

*Relation with which the subject lives.

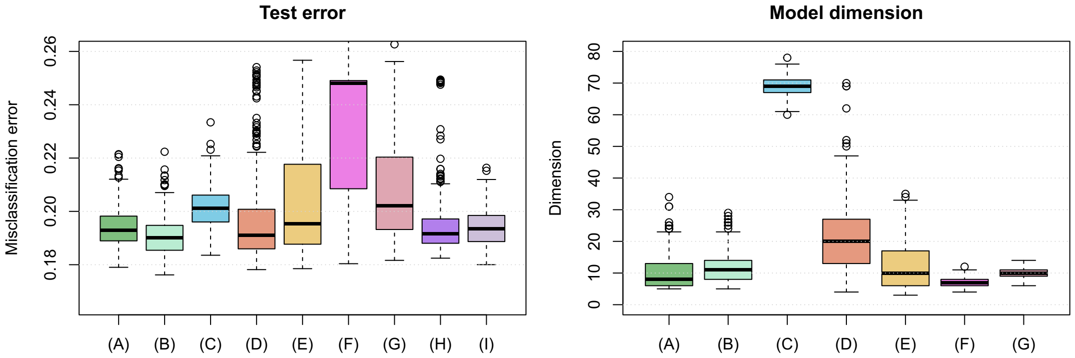


FIGURE 4 Prediction performance and fitted model dimension (respectively) of various methods on the Adult dataset: (A) SCOPE-100; (B) SCOPE-250; (C) Logistic regression; (D) CAS-ANOVA; (E) Adaptive CAS-ANOVA; (F) DMR; (G) BEF; (H) CART; (I) RF

TABLE 6 Results of experiments on the Adult dataset.

Method	Misclassification error	Model dimension	Computation time (s)
SCOPE-100	0.194	10.5	467
SCOPE-250	**0.191**	11.8	450
Logistic regression	0.202	68.9	0.04
CAS-ANOVA	0.198	21.5	429
Adaptive CAS-ANOVA	0.205	11.7	8757
DMR	0.235	6.9	11
BEF	0.207	9.8	1713
CART	0.196		0.01
RF	0.194		0.14

The best result is in bold.

thus yielding s_j true levels. The coefficients for the 5 continuous covariates were generated as draws from $\mathcal{N}_5(0, I)$. The response was then scaled to have unit variance, after which standard normal noise was added.

We used 10% ($n = 5938$) of the 59,381 total number of observations for training, and the remainder to compute an estimated MSPE (28) by taking an average over these observations. We repeated this 1000 times, sampling 10% of the observations and generating the coefficients as above anew in each repetition. The average mean squared prediction errors achieved by the various methods under comparison are given in Figure 6. We see that SCOPE with a cross-validated choice of γ performs best, followed by the Lasso and SCOPE-32.

6.5 | COVID-19 forecast Hub example

As well as the prediction performance experiments in the rest of this section, we include an exploratory data analysis example based on data relating to the ongoing (at time of writing) global COVID-19 pandemic. The COVID-19 Forecast Hub (2020) '*... serves as a central repository of forecasts and predictions from over 50 international research groups.*' A collection of different research groups publish forecasts every week of case incidence in each US state for some number of weeks into the future.

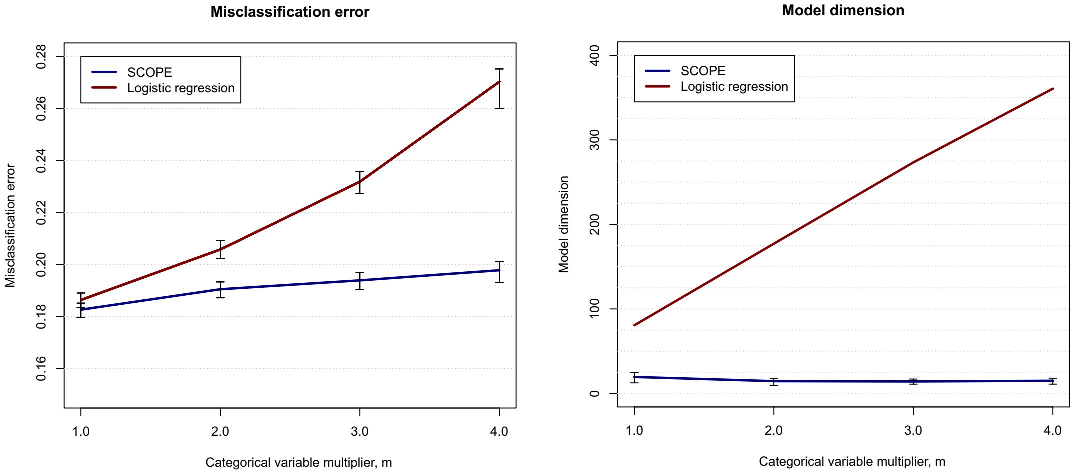
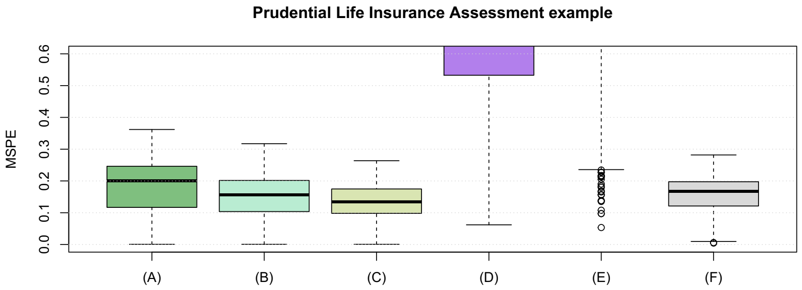


FIGURE 5 Misclassification error and dimensions of models fitted on a sample of the *Adult dataset* when levels have been artificially split m times



Method	MSPE
SCOPE-8	0.265
SCOPE-32	0.244
SCOPE-CV	**0.211**
CART	0.851
RF	1.120
Lasso	0.244

FIGURE 6 Mean squared prediction error on the example based on the Prudential Life Insurance Assessment dataset. Methods used are: (a) SCOPE-8; (b) SCOPE-32; (c) SCOPE-CV; (d) CART; (e) RF; (f) Lasso. The best result is in bold

In order to understand some of the difficulties of this challenging forecasting problem, we fitted an error decomposition model of the form

$$\log\left(\frac{1+\text{cases}_{w,l}}{1+\text{est.cases}_{m,t,w,l}}\right) = \alpha_0 + \alpha_{m,t} + \beta_{w,l} + \eta_{m,t,w,l}, \tag{29}$$

where w is the week that the forecast is for, l is the state, m indexes the forecasting model, t is the ‘target’ number of weeks in the future the forecast is for, $\eta_{m,t,w,l}$ is an error term, and $\text{cases}_{w,l}$ and $\text{est.cases}_{m,t,w,l}$ are the observed and estimated cases respectively. This decomposition allows an interaction term between time and location, which is important given that the pandemic was known to be more severe at different times for different areas. An interaction between model and forecasting distance was also included in order to capture the effect of some models potentially being more ‘short-sighted’ than others. The inclusion of the +1 on the left-hand side is to avoid numerators or denominators of zero.

We used data from 6 April 2020 to 19 October 2020, giving a total of 100,264 (m, t, w, l)-tuples. We applied a SCOPE penalty with $\gamma = 8$ to $\beta_{w,l}$, which had 1428 levels. The $\alpha_{m,t}$ coefficients, which amounted to 170 levels, were left unpenalised. The additional tuning parameter λ was selected using

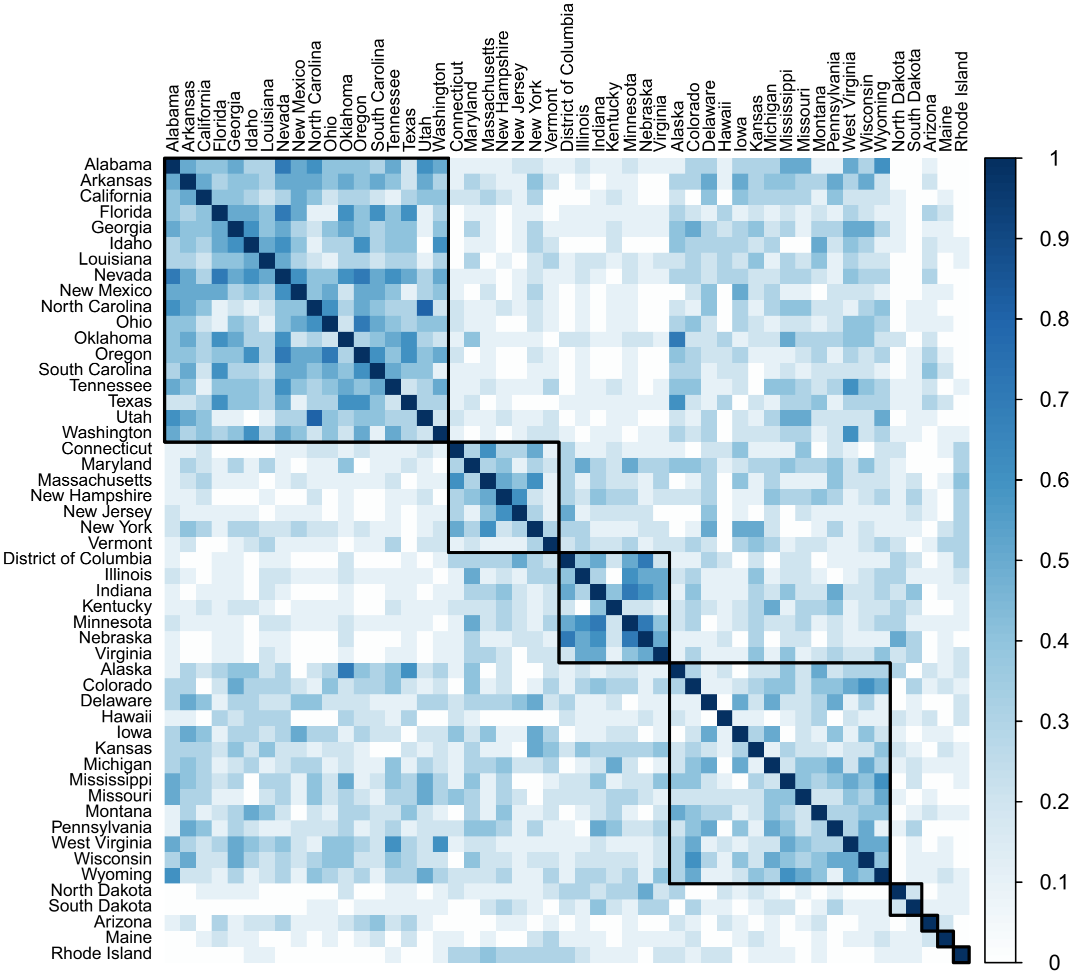


FIGURE 7 Similarity matrix for US states computed based on data relating to the second 'wave' of the COVID-19 pandemic in the United States, taken to be from 26 June 2020 to 29 August 2020

the extended Bayesian information criterion (Chen & Chen, 2008) rather than cross-validation, as it was more suited to this sort of exploratory analysis on data with a chronological structure.

The resulting estimates $\widehat{\beta}_{w,l}$ had 8 levels. We measured the 'similarity' of two US states l_a and l_b over a period of time by computing the proportion of weeks at which their estimates $\widehat{\beta}_{w,l_a} = \widehat{\beta}_{w,l_b}$ coincided. The similarity matrix presented in Figure 7 was constructed based on the second 'wave' of the epidemic which occurred in Summer 2020, with clusters identified by applying spectral clustering on the similarity matrix and plotted in order of decreasing within-cluster median pairwise similarity.

The resulting clusters are at once interpretable and interesting. Roughly speaking, the top 3 clusters ('top' when ordered according to median pairwise within-cluster agreement) correspond to states that experienced notable pandemic activity in the second, first and third 'waves' of the US coronavirus pandemic respectively. The first cluster features several southern States (e.g. Georgia, Florida, Texas) which experienced a surge of COVID cases in June–July. The second cluster features east coast states (e.g. New Jersey and New York) which experienced an enormous pandemic toll in April–May. And the third features midwestern states (e.g. Kentucky, Indiana, Nebraska) which had upticks most recently in September–October.

7 | DISCUSSION

In this work we have introduced a new penalty-based method for performing regression on categorical data. An attractive feature of a penalty-based approach is that it can be integrated easily with existing methods for regression with continuous data, such as the Lasso. Our penalty function is nonconvex, but in contrast to the use of nonconvex penalties in standard high-dimensional regression problems, the nonconvexity here is necessary in order to obtain sparse solutions, that is fusions of levels. While computing the global optimum of nonconvex problems is typically very challenging, for the case with a single categorical variable with several hundred levels, our dynamic programming algorithm can typically solve the resulting optimisation problem in less than a second on a standard laptop computer. The algorithm is thus fast enough to be embedded within a block coordinate descent procedure for handling multiple categorical variables.

We give sufficient conditions for SCOPE to recover the oracle least squares solution when $p = 1$ involving a minimal separation between unequal coefficients that is optimal up to constant factors. For the multivariate case where $p > 1$, we show that oracle least squares is a fixed point of our block coordinate descent algorithm, with high probability.

Our work offers several avenues for further work. On the theoretical front, it would be interesting to obtain guarantees for block coordinate descent to converge to a local optimum with good statistical properties, a phenomenon that we observe empirically. On the methodology side, it would be useful to generalise the penalty to allow for clustering multivariate coefficient vectors; such clustering could be helpful in the context of mixtures of regressions models, for example.

ORCID

Benjamin G. Stokell https://orcid.org/0000-0002-8365-715X
Rajen D. Shah https://orcid.org/0000-0001-9073-3782

REFERENCES

Bondell, H.D. & Reich, B.J. (2009) Simultaneous factor selection and collapsing levels in ANOVA. *Biometrics*, 65(1), 169–177.

Breheny, P. & Huang, J. (2011) Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. *The Annals of Applied Statistics*, 5(1), 232.

Breheny, P. & Huang, J. (2015) Group descent algorithms for nonconvex penalized linear and logistic regression models with grouped predictors. *Statistics and Computing*, 25(2), 173–187.

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

Breiman, L., Friedman, J., Stone, C. & Olshen, R. (1984) *Classification and regression trees*. The Wadsworth and Brooks-Cole statistics-probability series. Milton: Taylor & Francis.

Calinski, T. & Corsten, L. (1985) Clustering means in ANOVA by simultaneous testing. *Biometrics*, 41, 39–48.

Chen, J. & Chen, Z. (2008) Extended Bayesian information criteria for model selection with large model spaces. *Biometrika*, 95(3), 759–71.

Chiquet, J., Gutierrez, P. & Rigaill, G. (2017) Fast tree inference with weighted fusion penalties. *Journal of Computational and Graphical Statistics*, 26(1), 205–216.

COVID-19 Forecast Hub. (2020) Forecast Hub. Available from: https://covid19forecasthub.org.

Dua, D. & Graff, C. (2019) UCI machine learning repository.

Fan, J. & Li, R. (2001) Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456), 1348–1360.

Fan, J., Liu, H., Sun, Q. & Zhang, T. (2018) I-LAMM for sparse learning: Simultaneous control of algorithmic complexity and statistical error. *The Annals of Statistics*, 46(2), 814–841.

Friedman, J., Hastie, T. & Tibshirani, R. (2010) Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1), 1–22.

Gertheiss, J. & Tutz, G. (2010) Sparse modeling of categorial explanatory variables. *The Annals of Applied Statistics*, 4(4), 2150–2180.

Hocking, T.D., Joulin, A., Bach, F. & Vert, J.-P. (2011) Clusterpath an algorithm for clustering using convex fusion penalties. In: *28th International Conference on Machine Learning*, pp. 1.

Hu, S., O'Hagan, A. & Murphy, T.B. (2018) Motor insurance claim modelling with factor collapsing and bayesian model averaging. *Stat*, 7 (1), e180.

Hubert, L. & Arabie, P. (1985) Comparing partitions. *Journal of Classification*, 2(1), 193–218.

Jensen, P.B., Jensen, L.J. & Brunak, S. (2012) Mining electronic health records: towards better research applications and clinical care. *Nature Reviews Genetics*, 13(6), 395.

Kaggle. (2015) Prudential life insurance assessment. Available from: https://www.kaggle.com/c/prudential-life-insurance-assessment/dat.

Liaw, A. & Wiener, M. (2002) Classification and regression by randomforest. *R News*, 2(3), 18–22.

Loh, P.-L. & Wainwright, M.J. (2012) High-dimensional regression with noisy and missing data: provable guarantees with nonconvexity. *The Annals of Statistics*, 40(3), 1637–1664.

Loh, P.-L. & Wainwright, M.J. (2015) Regularized M-estimators with nonconvexity: Statistical and algorithmic theory for local optima. *Journal of Machine Learning Research*, 16(19), 559–616.

Lu, Y. & Zhou, H.H. (2016) Statistical and computational guarantees of lloyd's algorithm and its variants. *arXiv preprint arXiv:1612.02099*.

Ma, S. & Huang, J. (2017) A concave pairwise fusion approach to subgroup analysis. *Journal of the American Statistical Association*, 112(517), 410–423.

Maj-Kańska, A., Pokarowski, P., Prochenka, A. (2015) Delete or merge regressors for linear model selection. *Electronic Journal of Statistics*, 9(2), 1749–1778.

Mazumder, R., Friedman, J.H. & Hastie, T. (2011) Sparsenet: coordinate descent with nonconvex penalties. *Journal of the American Statistical Association*, 106(495), 1125–1138.

Oelker, M.-R., Pößnecker, W. & Tutz, G. (2015) Selection and fusion of categorical predictors with l 0-type penalties. *Statistical Modelling*, 15(5), 389–410.

Pauger, D. & Wagner, H. (2019) Bayesian effect fusion for categorical predictors. *Bayesian Analysis*, 14(2), 341–369.

Pauger, D., Leitner, M., Wagner, H. & Malsiner-Walli, G. (2019) EffectFusion: Bayesian effect fusion for categorical predictors. R package version 1.1.1.

Prochenka-Sotys, A. & Pokarowski, P. (2018) DMRnet: delete or merge regressors algorithms for linear and logistic model selection and high-dimensional data. R package version 0.2.0.

R Core Team. (2020) *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing.

Scott, A.J. & Knott, M. (1974) A cluster analysis method for grouping means in the analysis of variance. *Biometrics*, 30, 507–512.

Stokell, B. (2021) CatReg: solution paths for linear and logistic regression models with categorical predictors, with SCOPE penalty. `https://CRAN.Rproject.org/package=CatReg`.

Therneau, T. & Atkinson, B. (2019) rpart: recursive partitioning and regression trees. R package version 4.1-15.

Tibshirani, R. (1996) Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1), 267–288.

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J. & Knight, K. (2005) Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1), 91–108.

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

Tukey, J.W. (1949) Comparing individual means in the analysis of variance. *Biometrics*, 5, 99–114.

Tutz, G. & Gertheiss, J. (2016) Regularized regression for categorical data. *Statistical Modelling*, 16(3), 161–200.

Wang, Z., Liu, H. & Zhang, T. (2014) Optimal computational and statistical rates of convergence for sparse nonconvex learning problems. *The Annals of Statistics*, 42(6), 2164.

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

Zhang, C.-H. (2010) Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics*, 38(2), 894–942.

Zhao, T., Liu, H. & Zhang T. (2018) Pathwise coordinate optimization for sparse learning: algorithm and theory. *The Annals of Statistics*, 46(1), 180–218.

Zou, H. (2006) The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476), 1418–1429.

SUPPORTING INFORMATION

Additional supporting information may be found online in the Supporting Information section.

How to cite this article: Stokell BG, Shah RD, Tibshirani RJ. Modelling high-dimensional categorical data using nonconvex fusion penalties. *J R Stat Soc Series B*. 2021;00:1–33. https://doi.org/10.1111/rssb.12432

APPENDIX A

A.1 Candidate minimiser functions

In this section we give explicit forms of the functions $p_{k,r}$ as defined in Section 3.1. We write $q_{k,r}(x) = a_r x^2 + b_r x + c_r$ for simplicity, suppressing the subscript k. For $S \subseteq \mathbb{R}$ and $a,\ b \in \mathbb{R}$, we write $aS + b$ for the set $\{ax + b\colon x \in S\}$.

Recall from Section 3.1 that

$$u_{k,r,t}(\theta_{k+1})\colon = \tilde{\min_{\theta_k \in D_k : \theta_k < \theta_{k+1}}} \{\tilde{q}_{k,r}(\theta_k) + \tilde{\rho}_t(\theta_{k+1} - \theta_k)\}.$$

For a function $f\colon \mathbb{R} \to \mathbb{R} \cup \{\infty\}$, we denote the *effective domain* of f by

$$\operatorname{dom} f\colon = \{x \in \mathbb{R}\colon f(x) < \infty\}.$$

For each $r = 1, \ldots, m(k)$, there are cases corresponding to $t = 1$ and $t = 2$. The formulas are as follows:

$$u_{k,r,1}(x) = \frac{2a_r x^2 + 2(b_r - 2a_r\gamma\lambda)x + (b_r - 2a_r\gamma\lambda)^2}{2(1 - 2a_r\gamma)} + c_r,$$

$$\text{with} \quad \operatorname{dom} u_{k,r,1} = \begin{cases} \big((1 - 2a_r\gamma)I_{k,r} + \gamma(\lambda - b_r)\big) \cap \left[\dfrac{4a_r\gamma\lambda - b_r}{2a_r}, \dfrac{\lambda - b_r}{2a_r}\right] & \text{if} \quad 2a_r - 1/\gamma > 0 \\ \varnothing & \text{otherwise.} \end{cases}$$

If $g_k(\theta_{k+1}) = u_{k,r,1}(\theta_{k+1})$, then

$$b_k(\theta_{k+1}) = \frac{\theta_{k+1} + \gamma(b_r - \lambda)}{1 - 2a_r\gamma}.$$

The second case is

$$u_{k,r,2}(x) = -\frac{b_r^2}{4a_r} + c + \frac{1}{2}\gamma\lambda^2,$$

$$\text{with} \quad \operatorname{dom} u_{k,r,2} = \begin{cases} \left[-\dfrac{b_r}{2a_r} + \gamma\lambda, \infty\right] & \text{if} \quad a_r > 0 \text{ and} -b_r/2a_r \in I_{k,r} \\ \varnothing & \text{otherwise.} \end{cases}$$

Here, if $g_k(\theta_{k+1}) = u_{k,r,2}(\theta_{k+1})$, then

$$b_k(\theta_{k+1}) = -b_r/2a_r.$$

Considering (16), we see that we can also have the case where $g_k(\theta_{k+1}) = f_k(\theta_{k+1})$. Thus we can form the set of quadratics $p_{k,r}$ and associated intervals as the set of $u_{k,r,t}$ as above for $t = 1,2$ and the $q_{k,r}$ themselves. Note that when $g_k(\theta_{k+1}) = q_{k,r}(\theta_{k+1})$, we have $b_k(\theta_{k+1}) = \theta_{k+1}$.

A.2 Algorithm details

Algorithm 1 Outline of procedure for computing f_k

```
while $E, N(x) \neq \emptyset$ do
    if $\min\{y\colon (y,r) \in N(x)\} < \min E$ then
        $(y^*, r^*) = \arg\min\{y\colon (y,r) \in N(x)\}$
        $U = U \cup \{([\tilde{x}, y^*), r(x))\}$, $x = \tilde{x} = y^*$, $r(x) = r^*$
        $N(x) = \emptyset$, for any intersection between $p_{k-1,r(x)}$ and any $p_{k-1,r}$ with $r \in A(x)\backslash\{r(x)\}$
        at location $y > x$, set $N(x) = N(x) \cup \{(y,r)\}$.
    else
        $y^* = \min E$, $E = E\backslash\{y^*\}$,
        Update active set $A(y^*)$
        if $r(x) \notin A(y^*)$ then
            Set $r^*$ such that $p_{k-1,r^*} = \mathsf{ChooseFunction}(A(y^*), y^*)$
            $U = U \cup \{([\tilde{x}, y^*), r(x))\}$, $x = \tilde{x} = y^*$, $r(x) = r^*$
            $N(x) = \emptyset$, for any intersection between $p_{k-1,r(x)}$ and any $p_{k-1,r}$ with $r \in A(x)\backslash\{r(x)\}$
            at location $y \geq x$, set $N(x) = N(x) \cup \{(y,r)\}$.
        else
            if $p_{k-1,r(x)} \neq p_{k-1,r^*} = \mathsf{ChooseFunction}(A(y^*), y^*)$ then
                $U = U \cup \{([\tilde{x}, y^*), r(x))\}$, $x = \tilde{x} = y^*$, $r(x) = r^*$
                $N(x) = \emptyset$, for any intersection between $p_{k-1,r(x)}$ and any $p_{k-1,r}$ with $r \in$
                $A(x)\backslash\{r(x)\}$ at location $y > x$, set $N(x) = N(x) \cup \{(y,r)\}$.
            else
                if $A(y^*) \neq A(x)$ then
                    For any intersection between $r(x)$ and any $r \in A(y^*)\backslash A(x)$ at location $y > x$,
                    set $N(y^*) = N(y^*) \cup \{(y,r)\}$.
                    For any $(y,r) \in N(x)$ with $r \notin A(y^*)$, set $N(y^*) = N(y^*)\backslash\{(y,r)\}$
                    $x = y^*$
                end if
            end if
        end if
    end if
end while
```

```
Algorithm 2 ChooseFunction($(H, x)$)
Input: $H = \{h_1, \dots, h_n\}$ a set of functions, $x$ a real number
Set $H_1 = \arg\min\{h(x) : h \in H\}$
if $|H_1| = 1$ then
    Select $h^* \in H_1$
else
    Set $H_2 = \arg\min\{h'(x) : h \in H_1\}$
    if $|H_2| = 1$ then
        Select $h^* \in H_2$
    else
        Set $H_3 = \arg\min\{h''(x) : h \in H_2\}$
        Select $h^* \in H_3$ (choosing $h_i \in H_3$ with $i$ minimal if $|H_3| > 1$)
    end if
end if
Output: $h^*$
```

Algorithm 1 describes in detail how the optimisation routine works. In the algorithm we make use of the following objects:

1. For $x \in \mathbb{R}$, $A(x)$ is the active set at x;
2. E is the set of points at which the active set changes;
3. $N(x)$ is the intersection set at x;
4. U is a set of tuples (I, r) where $I \subseteq \mathbb{R}$ is an interval and r is an integer, which is dynamically updated as the algorithm progresses.

See Section 3.1.2 for definitions of the sets above. We also use the convention that if $x = -\infty$ then $[x,y) = (-\infty,y)$.

All of the $p_{k,1}, \dots, p_{k,m(k)}$ and $J_{k,m}$ are computed at the start of each iterate k. We then initialise

$$E = \bigcup_{r=1}^{n(k)} \partial J_{k-1,r},$$

the set of all of the end-points of the intervals $J_{k-1,1}, \dots, J_{k-1,n(k)}$.

Here x can be thought of as the 'current position' of the algorithm; $\tilde{x}$ is used to store when the minimising function $p_{k-1,r(x)}$ last changed. We initialise $\tilde{x} = -\infty$ and $x = -1 + \max\{y \in I_{k-1,1} : f'_{k-1}(y_-) \leq 0\}$. This choice of x ensures that the active set $A(x)$ contains only one element (as mentioned in Section 3.1); this will always be the index corresponding to the function $\tilde{q}_{k-1,1}$.

We initialise the output set $U = \emptyset$, which by the end of this algorithm will be populated with the functions $\tilde{q}_{k,1}, \dots, \tilde{q}_{k,m(k)}$ and their corresponding intervals $I_{k,1}, \dots, I_{k,m(k)}$ that partition $\mathbb{R}$. Finally, we initialise the set $N(x)$ which will contain the intersections between $p_{k-1,r(x)}$ and other functions in the active set. As the active set begins with only one function, we set $N(x) = \emptyset$.

As mentioned in Section 3.1, there are several modifications that can speed up the algorithm. One such modification follows from the fact that for each r, $u_{k,r,2}$ is a constant function over its effective domain, and their effective domain is a semi-infinite interval (see Section A.1 of the Appendix for their expressions). Therefore, for a given point $x \in \mathbb{R}$, we can remove all such functions from $A(x)$ except for the one taking the minimal value.

We also note that in Algorithm 1, the set $N(x)$ is not recomputed in its entirety at every point x at which $A(x)$ is updated, as is described in Section 3.1. Line 13 shows how sometimes $N(x)$ can instead be updated by adding or removing elements from it. Often, points 3 (i) and 3 (ii) from the description in the Section 3.1 will coincide, and in such instances some calls to ChooseFunction (Algorithm 2) can be skipped.