Penalized angular regression for personalized predictions

Kristoffer H. Hellton
Norwegian Computing Center, Norway

March 7, 2022

Abstract

Personalization is becoming an important aspect of many predictive applications. We introduce a penalized regression method which inherently implements personalization. Personalized angle (PAN) regression constructs regression coefficients that are specific to the covariate vector for which one is producing a prediction, thus personalizing the regression model itself. This is achieved by penalizing the normalized prediction for a given covariate vector. The method therefore penalizes the normalized regression coefficients, or the angles of the regression coefficients in a hyperspherical parametrization, introducing a new angle-based class of penalties. PAN hence combines two novel concepts: penalizing the normalized coefficients and personalization. For an orthogonal design matrix, we show that the PAN estimator is the solution to a low-dimensional eigenvector equation. Based on the hyperspherical parametrization, we construct an efficient algorithm to calculate the PAN estimator. We propose a parametric bootstrap procedure for selecting the tuning parameter, and simulations show that PAN regression can outperform ordinary least squares, ridge regression and other penalized regression methods in terms of prediction error. Finally, we demonstrate the method in a medical application.

Keywords: Angular estimation; Cosine similarity; Hyperspherical coordinates; Penalized regression; Personalization; Personalized predictions; Shrinkage.

1 Introduction

The ambition to perform personalization when predicting is becoming an important aspect of many applications: medicine (Cheng et al., 2012; Carrión et al., 2016), fraud detection (Alowais and Soon, 2012), marketing (Tang et al., 2013), item recommendation (Rafailidis et al., 2014), nutrition (Zeevi et al., 2015) and education (Reber et al., 2018). Personalized medicine or precision medicine, for instance, utilizes the genomic information, proteins, or the environment of a patient to predict individualized treatment decisions (Zhang and Nebert, 2017; Kosorok and Laber, 2019). Other examples include personalized marketing (e.g. delivering individualized product prices or messages to specific costumers) and personalized education (e.g. tailoring learning materials and questions to each individual student to increase interest). We believe that these applications also call for statistical prediction methods targeting the individual on a *methodological level*, meaning that the estimated model itself may vary with each prediction one wishes to make. The aim is to minimize the prediction error for each individual covariate vector, instead of minimizing the average prediction error. We propose a form of penalized regression which inherently features this personalized approach to prediction.

Penalized regression is a class of methods useful for prediction, particularly for high-dimensional or multicollinear data. The standard methods, e.g. ridge regression, lasso and elastic net (Hoerl and Kennard, 1970; Tibshirani, 1996; Zou and Hastie, 2005) penalize some norm, or norm combination, of the regression coefficients. The norms measure some length of the regression coefficient vector, where the simplest example is the L_2 norm which equals the Euclidean length. A p-dimensional vector can, however, always be parametrized (using hyperspherical coordinates) in terms of the Euclidean length and a normalized direction vector, corresponding to $p-1$ angles. In this paper, we introduce a new type of penalty based on penalizing the normalized coefficient vector, or the *angles* of the regression coefficients, instead of the length. Note that this is not related to least angle regression, despite the name similarity. The hyperspherical coordinates generalize polar coordinates to p dimensions and are commonly used in physics, e.g. to solve three- and four-particle problems and Laplace's equation (Öhrn and Linderberg, 1983; Cohl, 2011). There has been an increased interest in the statistical distribution of angles in high dimension (Cho, 2009; Cai et al., 2013) and the use of the hyperspherical parametrization in statistics and machine learning (Pourahmadi and Wang, 2015; Liu et al., 2017). Related fields include directional statistics (regression models for circular and spherical outcomes, see e.g. Mardia, 1972) and compositional data

(Scealy and Welsh, 2011).

In model selection, Claeskens and Hjort (2003) introduced the concept of addressing the *aim* of the statistical analysis with the focused information criterion. Focused model selection defines an *a priori* quantity-of-interest that guides the selection of a statistical model, instead of considering overall goodness-of-fit measures (Claeskens and Hjort, 2008). For different aims, different models will be optimal. Frameworks such as targeted learning also introduce a notion of a pre-defined target parameter representing the scientific question (Van der Laan and Rose, 2011). Within the focused approach, Hellton and Hjort (2018) used a specific prediction as the aim, framing the resulting model as personalized. They proposed to vary the tuning parameter in ridge regression with the covariate vector, x_0, for which one wishes to make a prediction. This personalized tuning parameter could be estimated via a two-stage plug-in procedure, or by adaptive validation (Huang et al., 2019).

Currently, the term personalization is often understood in applied fields as standard regression models, where covariates account for the differences between individuals. But personalization of the regression model can be implemented in different ways. Hastie and Tibshirani (1993) proposed to estimate sample-specific linear regression coefficients that can change smoothly with the value of other variables, which they referred to as "effect modifiers", and Visweswaran et al. (2010) proposed a Bayesian algorithm for instance-specific Markov blanket models. More recently, Lengerich et al. (2019) proposed to estimate sample-specific models by regularizing a low-rank latent representation of the model parameters. Further, Jabbari et al. (2018) proposed to estimate instance-specific Bayesian Networks.

In this paper, we combine the two concepts – penalizing the normalized coefficients and personalization – to achieve a guided penalization of the regression coefficients. We incorporate the personalization in the penalty structure itself implementing it inherently in the method. Hence the method requires no additional covariates, describing the heterogeneity, to produce personalized regression coefficients and predictions. With this we introduce a new class of regression penalties based on the angle parameters, which can be exploited in other methodological contexts.

The remainder of the paper is organized as follows: In Section 2 we present the personalized angle penalty and show how it penalizes the angle parameter in a hyperspherical parametrization of linear regression. In Section 3, an algorithm for calculating the resulting estimator is given. Section 4 presents a simulation study comparing the proposed method to OLS, ridge and lasso regression, and in Section 5, we illustrate it in a medical application. Concluding remarks are discussed in Section 6, and all proofs are given in the Appendix.

2 Personalized angle regression

2.1 Definition

Suppose we have observed data $\{y_i, x_i\}, i = 1, \ldots, n$, consisting of p-dimensional covariate vectors, $x_i \in \mathbb{R}^p$, and univariate continuous outcomes, $y_i \in \mathbb{R}$, and consider the linear regression model

$$y_i = x_i^T \beta + \varepsilon_i \quad i = 1, \ldots, n,$$

where $\beta \in \mathbb{R}^p$ is a p-dimensional vector of regression coefficients and $\varepsilon_i \in \mathbb{R}^n$ is an identically and independently distributed noise term with zero mean, $E(\varepsilon_i) = 0$, and variance, $\text{Var}(\varepsilon_i) = \sigma^2$. The prediction of a new outcome y_0 given the covariate vector x_0 is then given by

$$\mu_0 = E(y_0 \mid x_0) = x_0^T \beta.$$

We further denote the vector of outcomes by $Y = [y_1, \ldots, y_n]^T$ and the $n \times p$ design matrix by X with x_i^T as each row. The outcome vector and the design matrix are assumed to be centered.

In a personalized prediction context, the primary aim is to achieve optimal predictive ability. We propose to penalize the prediction given a specific covariate vector, x_0. This personalizes the regression model and improves the prediction error (ignoring the estimation error) by leveraging the heterogeneity in the covariates. The covariate vector x_0 represents an instance for which we wish to produce a prediction, e.g. a patient in the personalized medicine context. Importantly, personalizing the regression model requires the regression coefficients to be recalculated for each new prediction. The penalty we introduce is based on the normalized inner product between x_0 and β, or the *normalized prediction* and has an angle-based interpretation in hyperspherical coordinates. The resulting regression estimates will also be optimal for the specific x_0. We therefore term the method Personalized Angle (PAN) regression.

Definition 1 (Cartesian coordinates)**.** *The Personalized Angle (PAN) estimator for a specific covariate vector $x_0 \neq \mathbf{0}$, $\hat{\beta}_{x_0}(\lambda) = [\hat{\beta}_{x_0,1}(\lambda), \cdots \hat{\beta}_{x_0,p}(\lambda)]^T$ is defined as*

$$\hat{\beta}_{x_0}(\lambda) = \arg\min_{\beta} \left\{ \sum_{i=1}^{n} \left(y_i - x_i^T \beta\right)^2 + \frac{\lambda}{x_0^T x_0} \frac{\beta^T x_0 x_0^T \beta}{\beta^T \beta} \right\}, \tag{1}$$

where $\lambda \in \mathbb{R}$ is a tuning parameter.

Let the normalized unit-vectors of β and x_0 be denoted by $\gamma_\beta = \beta/\|\beta\|$ and $\gamma_{x_0} = x_0/\|x_0\|$, respectively. We refer to γ_β as the normalized regression coefficients and γ_{x_0} as the normalized covariate vector. The PAN penalty, $J_{PAN}(\beta)$, is then equal to the squared L_2 norm of the normalized prediction, $\gamma_{x_0}^T\gamma_\beta$, for a given x_0

$$J_{PAN}(\beta) = \frac{1}{x_0^T x_0}\frac{\beta^T x_0 x_0^T \beta}{\beta^T\beta} = \frac{(x_0^T\beta)^2}{\|x_0\|^2\|\beta\|^2} = \|\gamma_{x_0}^T\gamma_\beta\|^2,$$

where $\|\cdot\|$ denotes the L_2 norm. The normalized prediction is also related to the cosine similarity, a concept in the machine learning literature (Salton and McGill, 1983; Romesburg, 1984). The normalized prediction equals the cosine similarity between β and x_0

$$\text{CosSim}(\beta, x_0) = \frac{x_0^T\beta}{\|x_0\|\|\beta\|}.$$

Thus the PAN penalty is also given by the squared cosine similarity: $J_{PAN}(\beta) = \text{CosSim}^2(\beta, x_0)$.

Since the penalty $J_{PAN}(\beta)$ shrinks the prediction of the outcome given x_0 towards zero, it will introduce a bias while also lowering the variance. There will then be an optimal trade-off which will improve the mean squared prediction error for x_0 only. In the parameter space, the prediction of zero given x_0 corresponds to a hyperplane with x_0 as its normal vector, denoted by H_0:

$$H_0 = \{\beta \in \mathbb{R}^p : x_0^T\beta = 0\},$$

with dimension, $\dim(H_0) = p - 1$. The penalty in Equation (1) therefore shrinks the regression estimator towards the hyperplane H_0. As the tuning parameter λ increases, the part of the estimator orthogonal to H_0 decreases. As $\lambda \to \infty$, the prediction becomes zero and the estimator converges to the projection of the OLS estimator $\tilde{\beta}$ onto the hyperplane H_0.

Remark 1 (Negative tuning parameter). An important but counter-intuitive aspect of the PAN penalty is that the tuning parameter may be *negative*. This stands in stark contrast to other penalized regression methods where the tuning parameter is required to be positive. PAN regression allows for a negative tuning parameter because the penalty in Equation (1) is bounded between 0 and 1. The boundedness ensures that the objective function does not explode if the tuning parameter is negative. A negative tuning parameter corresponds to shifting the prediction away from zero, and essentially "expanding" rather than shrinking the prediction. Changing to a hyperspherical coordinate system gives further intuition regarding this novel feature and will be explored in Section 2.2.

Remark 2 (High-dimensionality). The shrinkage induced by penalizing the normalized prediction, $\gamma_{x_0}^T\gamma_\beta$, will be particularly effective in higher dimensions. Cai et al. (2013) established

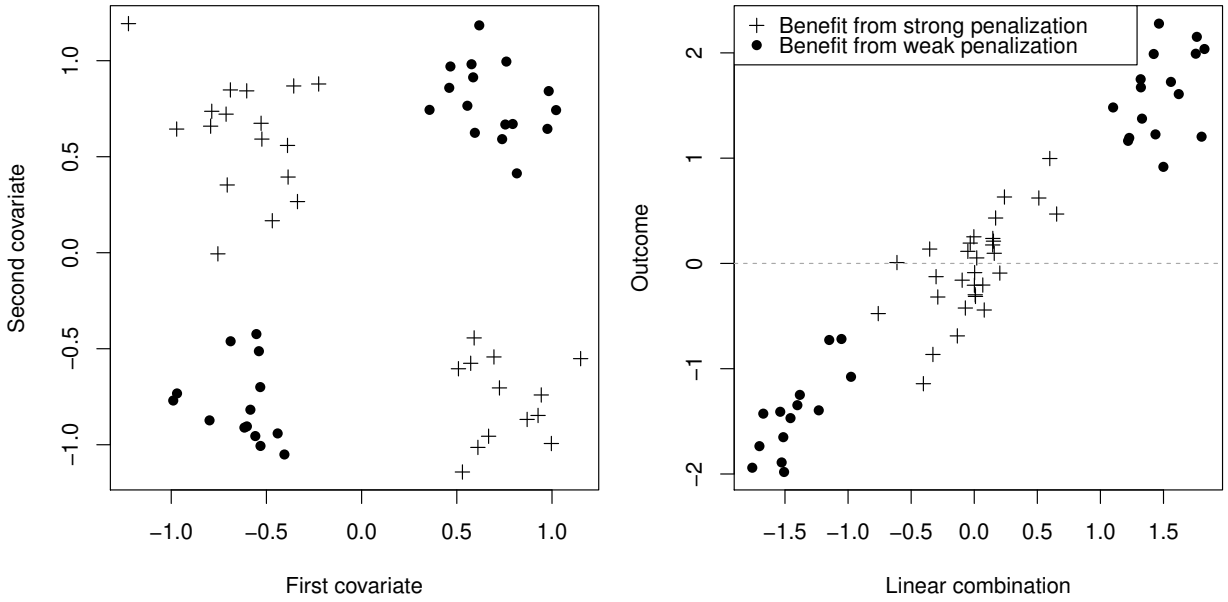

Figure 1: An example of heterogeneous covariates and regression coefficients ($\beta = [1, 1]^T$) where a differentiated shrinkage improves the overall prediction error. Observations corresponding to outcomes around zero will benefit in terms of a lower mean squared error from a stronger penalization, while observations corresponding to outcomes far from zero will benefit from a weaker penalization. The left panel shows the two covariates, while the right panel plots the outcome against the linear combination $x_i^T\beta$.

the folklore that "all high-dimensional random vectors are almost always nearly orthogonal to each other". They proved that the angle between two random vectors will converge to 90° as the dimension increases, $p \to \infty$, demonstrating that the random vectors will be asymptotically orthogonal. For increasing dimensions, the normalized prediction will therefore be more closely distributed around 0 and hence have a greater benefit from a shrinkage towards zero.

Example 1. *Figure 1 shows a situation where the personalization of PAN regression is beneficial for improving the prediction error. The left panel shows the two dimensional covariates ($p = 2$) with four distinct clusters centered at (1,1), (1,-1), (-1,1) and (-1,-1). The right panel shows the outcome Y plotted against the linear combination $x_i^T\beta$ for the regression coefficients $\beta = [1, 1]^T$.*

When shrinking towards zero, the predictions close to zero achieve the lowest mean squared error for a larger shrinkage, while predictions far from zero achieve the lowest mean

squared error for a smaller (or zero) shrinkage (Gruber, 1998). In the situation shown in Figure 1, the lower right and upper left clusters will have outcomes close to zero (as the regression coefficients and covariate vectors are orthogonal). These two clusters will therefore benefit, in terms of a lower prediction error, from more penalization and a stronger shrinkage towards zero. The upper right and the lower left clusters, on the other hand, will have outcomes further away from zero, and will benefit from a weaker penalization. A personalized prediction method will tailor the penalization according to this covariate vector information. It will be able to exploit the heterogeneity and shrink some clusters more and others less.

2.2 Angular interpretation

Geometrically, any p-dimensional vector $x = [x_1, \ldots, x_p]^T$ can be described by a length r and $p-1$ angles, $\alpha_1, \ldots, \alpha_{p-1}$, defined relative to the unit vectors. The standard hyperspherical parametrization, generalizing polar coordinates to $\mathbb{R}^p$, is given by

$$\begin{aligned}
x_1 &= r\cos\alpha_1, \\
x_2 &= r\sin\alpha_1\cos\alpha_2, \\
&\vdots \\
x_{p-1} &= r\sin\alpha_1\sin\alpha_2\cdots\sin\alpha_{p-2}\cos\alpha_{p-1}, \\
x_p &= r\sin\alpha_1\sin\alpha_2\cdots\sin\alpha_{p-2}\sin\alpha_{p-1},
\end{aligned}$$

where $r \geq 0$ and the angles fulfill $0 \leq \alpha_j \leq \pi$ for $j = 1, 2, \ldots, p-2$ and $-\pi < \alpha_{p-1} \leq \pi$. Using hyperspherical coordinates, we can reparametrize the regression coefficient vector as

$$\beta = r_\beta \gamma_\beta, \tag{2}$$

by its length $r_\beta = \|\beta\|$ and a direction vector, the normalized coefficient vector

$$\gamma_\beta = \beta/\|\beta\| = [\cos(\alpha_{\beta,1}), \ldots, \sin(\alpha_{\beta,1})\cdots\sin(\alpha_{\beta,p-2})\sin(\alpha_{\beta,p-1})]^T.$$

For simplicity, we first consider only two dimensions, $p = 2$. We can then transform the standard linear regression model (using the definition of the dot product) into a nonlinear regression problem with the parameters r_β and α_β

$$y_i = x_i^T\beta + \varepsilon_i = r_\beta r_{x_i}\cos(\alpha_{x_i} - \alpha_\beta) + \varepsilon_i, \qquad i = 1, \ldots, n. \tag{3}$$

Here r_{x_i} and α_{x_i} are the length and the angle of the ith covariate vector, respectively. As there is only one angle parameter in two dimensions, we omit the index $j = 1$. The regression

parameters could then be found by estimating r_β as an amplitude and α_β as a phase shift in Equation (3). This reparametrization supplies an alternative estimation approach for the linear regression problem. A related setting was, for instance, explored by Welsh (1985).

If $p > 2$, the estimator of the transformed model can be found by minimizing the following residual sum-of-squares

$$(\tilde{r}_\beta, \tilde{\alpha}_{\beta,1}, \ldots, \tilde{\alpha}_{\beta,p-1}) = \arg\min_{r_\beta, \alpha_\beta} \left\{ \sum_{i=1}^{n} \Big[y_i - r_\beta r_{x_i} \big(\cos(\alpha_{\beta,p-1} - \alpha_{x_i,p-1}) \prod_{j=1}^{p-2} \sin\alpha_{\beta,j} \sin\alpha_{x_i,j} + \sum_{j=2}^{p-2} \cos\alpha_{\beta,j} \cos\alpha_{x_i,j} \prod_{k=1}^{p-2} \sin\alpha_{\beta,k} \sin\alpha_{x_i,k} \big) \Big]^2 \right\}, \quad (4)$$

which naturally yields the ordinary least squares (OLS) estimator $\tilde{\beta} = (X^T X)^{-1} X^T Y$ transformed to hyperspherical coordinates

$$\begin{aligned}
\tilde{r}_\beta &= \sqrt{\tilde{\beta}_p^2 + \tilde{\beta}_{p-1}^2 + \cdots + \tilde{\beta}_2^2 + \tilde{\beta}_1^2}, \\
\tilde{\alpha}_{\beta,j} &= \arccos \frac{\tilde{\beta}_j}{\sqrt{\tilde{\beta}_p^2 + \tilde{\beta}_{p-1}^2 + \cdots + \tilde{\beta}_j^2}}, \quad j = 1, \ldots, p-2, \\
\tilde{\alpha}_{\beta,p-1} &= \begin{cases} \arccos \frac{\tilde{\beta}_{p-1}}{\sqrt{\tilde{\beta}_p^2 + \tilde{\beta}_{p-1}^2}} & \tilde{\beta}_p \geq 0, \\ 2\pi - \arccos \frac{\tilde{\beta}_{p-1}}{\sqrt{\tilde{\beta}_p^2 + \tilde{\beta}_{p-1}^2}} & \tilde{\beta}_p < 0. \end{cases}
\end{aligned}$$

2.2.1 Penalizing the length

Ridge regression (Hoerl and Kennard, 1970) adds a squared L_2 penalty to the residual sum-of-squares in Equation (4), which corresponds to the squared length of the regression coefficient vector in hyperspherical coordinates

$$J_{ridge}(\beta) = \sum_{j=1}^{p} \beta_j^2 = \|\beta\|_2^2 = r_\beta^2.$$

Ridge regression thus tries to shrink the length of the OLS regression coefficients towards the origin. The ridge estimator has the explicit solution $\tilde{\beta}(\lambda) = (X^T X + \lambda I_p)^{-1} X^T Y$, where I_p is the p-dimensional identity matrix and the tuning parameter λ controls the penalization. Zero penalization corresponds to the OLS estimator, $\tilde{\beta}(0) = \tilde{\beta}$.

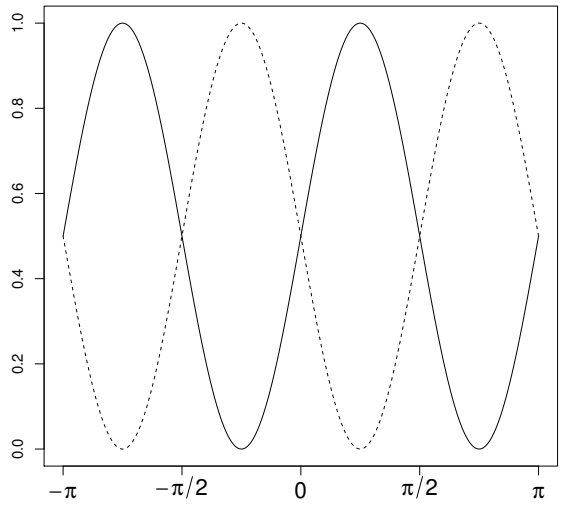

Figure 2: The penalty function of PAN in Equation (5) as a function of the angle parameter, $\alpha_\beta \in (-\pi, \pi]$. The angle of the covariate vector x_0 equals $\alpha_{x_0} = \pi/4$. The penalty associated with a positive tuning parameter is shown by the solid line, while the flipped penalty function induced by a negative tuning parameter is shown by the dashed line.

2.2.2 Penalizing the angle

In two dimensions, $p = 2$, the PAN penalty reduces to a squared cosine penalty for the angle parameter α_β

$$J_{PAN}(\beta) = \frac{1}{x_0^T x_0} \frac{\beta^T x_0 x_0^T \beta}{\beta^T \beta} = \cos^2(\alpha_\beta - \alpha_{x_0}) = 1 - \cos^2\left(\alpha_\beta - \left(\alpha_{x_0} \pm \frac{\pi}{2}\right)\right), \tag{5}$$

where we omit the index $j = 1$ for simplicity. The penalty hence corresponds to a ridge-type penalty for the *angle parameter*, and it therefore enforced the shrinkage by rotating the OLS estimator. The angle α_{x_0} is determined by x_0, the covariate vector for which we wish to make a prediction. Figure 2 shows the penalty as a function of the angle parameter α_β on the interval $(-\pi, \pi]$. The penalty given a positive tuning parameter is shown by the solid line, while the penalty for a negative tuning parameter is shown by the dashed line. This form of the penalty demonstrates how the tuning parameter may be negative, as the cosine function is bounded between 0 and 1 regardless of the sign of the penalty parameter. Changing the sign of the tuning parameter from positive to negative simply flips the cosine function. This moves the minimum of the penalty function to a different parameter value, $\frac{\pi}{2}$ away from the

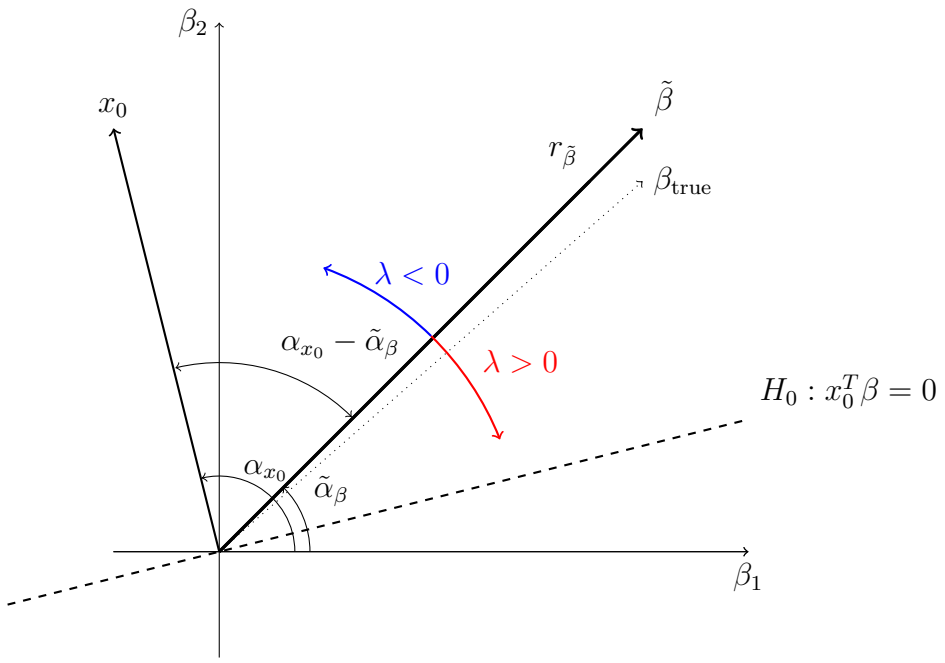

Figure 3: Illustration of the shrinkage induced by the PAN penalty for $p = 2$. The penalty rotates the OLS estimator $\tilde{\beta}$ towards H_0 (dashed line) when the tuning parameter λ is positive, and away from H_0 when it is negative.

old minimum.

Figure 3 illustrates the OLS estimator, $\tilde{\beta}$, parametrized by its length $\tilde{r}_\beta$ and the angle $\tilde{\alpha}_\beta$ in the parameter space. The covariate vector, x_0, given by the angle α_{x_0}, is visualized by laying the covariate space on top of the parameter space. The zero prediction for x_0 is then used as an angular origin to shrink towards. The prediction equals zero when the regression coefficients, β, fulfills the equation $x_0^T \beta = 0$, i.e. the vectors β and x_0 are orthogonal. In two dimensions, this corresponds to the angle of β being equal to $\alpha_\beta = \alpha_{x_0} \pm \frac{\pi}{2}$, visualized by the dashed line in Figure 3. Hence, when λ increases, the estimated angle rotates away from $\tilde{\alpha}_\beta$ and towards H_0, the line orthogonal to x_0, as illustrated in Figure 3. The estimated angle, $\hat{\alpha}_\beta$, is rotated towards the closest of the two angles $\alpha_{x_0} \pm \frac{\pi}{2}$, shrinking the prediction towards zero. For a negative tuning parameter value, on the other hand, the estimated angle is rotated away from H_0 and towards x_0, as illustrated in Figure 3.

With the PAN penalty in Equation (5), the penalized residual sum-of-squares regularizing

the angle parameter is for $p = 2$ given by

$$(\hat{r}_{\beta,x_0}(\lambda), \hat{\alpha}_{\beta,x_0}(\lambda)) = \arg\min_{r_\beta,\alpha_\beta} \left\{ \sum_{i=1}^{n} \left[y_i - r_\beta r_{x_i} \cos(\alpha_\beta - \alpha_{x_i}) \right]^2 + \lambda \cos^2 (\alpha_\beta - \alpha_{x_0}) \right\},$$

where $\alpha_\beta \in (-\pi, \pi], r_\beta \geq 0$ and $\lambda \in \mathbb{R}$. For an orthonormal design matrix, $X^T X = I_2$, the normal equations give explicit solutions for the parameter estimators

$$\tan 2\hat{\alpha}_{\beta,x_0}(\lambda) = \frac{\tilde{r}_\beta^2 \sin 2\tilde{\alpha}_\beta + \lambda \sin 2\left(\alpha_{x_0} \pm \frac{\pi}{2}\right)}{\tilde{r}_\beta^2 \cos 2\tilde{\alpha}_\beta + \lambda \cos 2\left(\alpha_{x_0} \pm \frac{\pi}{2}\right)}, \quad \hat{r}_{\beta,x_0}(\lambda) = \tilde{r}_\beta \cos(\tilde{\alpha}_\beta - \hat{\alpha}_{\beta,x_0}(\lambda)). \tag{6}$$

The derivation of the result is found in the Appendix. Equation (6) demonstrates that for $\lambda = 0$ the estimated angle and length are equal to the angle and length of the OLS estimator. As $\lambda \to \infty$, the estimated angle converges to either $\hat{\alpha}_{\beta,x_0}(\lambda) \to \alpha_{x_0} + \frac{\pi}{2}$, if $\tilde{\alpha}_\beta \in [\alpha_{x_0}, \alpha_{x_0} + \pi]$, or to $\hat{\alpha}_{\beta,x_0}(\lambda) \to \alpha_{x_0} - \frac{\pi}{2}$, if $\tilde{\alpha}_\beta \in [\alpha_{x_0} - \pi, \alpha_{x_0}]$, becoming exactly orthogonal to x_0. The estimated angle and length will hence shrink the prediction given x_0 towards zero.

In the orthonormal design case, the prediction $\hat{y}_0$ given x_0 is determined in hyperspherical coordinates by the estimated length and the double tangent expression in Equation (6) as

$$\hat{y}_0 = x_0^T \hat{\beta}_{x_0}(\lambda) = \underbrace{r_0 \tilde{r}_\beta \cos(\alpha_{x_0} - \tilde{\alpha}_\beta)}_{\text{OLS prediction}} \underbrace{\left(\frac{1}{2} + \frac{1}{2} \frac{\tilde{r}_\beta^2 - \lambda}{\sqrt{(\tilde{r}_\beta^2 + \lambda)^2 - 4\lambda \tilde{r}_\beta^2 \cos^2(\alpha_{x_0} - \tilde{\alpha}_\beta)}} \right)}_{\text{Shrinkage factor}},$$

where $r_0 \tilde{r}_\beta \cos(\alpha_{x_0} - \tilde{\alpha}_\beta) = x_0^T \tilde{\beta}$ is the OLS prediction. The PAN prediction hence equals the OLS prediction multiplied by a shrinkage factor. When λ increases, the shrinkage factor decreases and as $\lambda \to \infty$, the factor converges to zero. Importantly, the shrinkage factor depends on the angle of the specific covariate vector x_0, such that the shrinkage will *vary* for different x_0 when λ is fixed. The shrinkage term thus explicitly expresses the feature of personalization inherent in the PAN penalty.

Definition 2 (Hyperspherical coordinates)**.** *The Personalized Angle (PAN) estimator in hyperspherical coordinates* $\hat{r}_{\beta,x_0}, \hat{\alpha}_{\beta,x_0,1}, \ldots, \hat{\alpha}_{\beta,x_0,p-1}$ *for a specific covariate vector* x_0 *parametrized*

by r_0 *and* $\alpha_{x_0,1}, \ldots, \alpha_{x_0,p-1}$ *is defined as*

$$(\hat{r}_{\beta,x_0}, \hat{\alpha}_{\beta,x_0,1}, \ldots, \hat{\alpha}_{\beta,x_0,p-1}) = \arg\min_{r_\beta, \alpha_{\beta,1}, \ldots} \Bigg\{ \sum_{i=1}^{n} \Big(y_i - r_\beta r_{x_i} \big(\cos(\alpha_{\beta,p-1} - \alpha_{x_i,p-1}) \prod_{j=1}^{p-2} \sin\alpha_{\beta,j} \sin\alpha_{x_i,j} + \sum_{j=2}^{p-2} \cos\alpha_{\beta,j} \cos\alpha_{x_i,j} \prod_{k=1}^{p-2} \sin\alpha_{\beta,k} \sin\alpha_{x_i,k} \big) \Big)^2$$
$$+\lambda \Big(\cos(\alpha_{\beta,p-1} - \alpha_{x_0,p-1}) \prod_{j=1}^{p-2} \sin\alpha_{\beta,j} \sin\alpha_{x_0,j} + \sum_{j=2}^{p-2} \cos\alpha_{\beta,j} \cos\alpha_{x_0,j} \prod_{k=1}^{p-2} \sin\alpha_{\beta,k} \sin\alpha_{x_0,k} \Big)^2 \Bigg\},$$

where r_{x_i} *and* $\alpha_{x_i,j}$ *for* $j = 1, \ldots, p-1$ *are the length and angles of the covariate vectors* x_i *for* $i = 1, \ldots, n$ *and* $\lambda \in \mathbb{R}$ *is a tuning parameter.*

The hyperspherical parametrization has a computational advantage when optimizing numerically as it easily avoids dividing by zero. From Definitions 1 and 2, the PAN estimator can be described by its length and direction vector, using the hyperspherical parametrization

$$\hat{\beta}_{x_0}(\lambda) = \hat{r}(\lambda)\hat{\gamma}(\lambda),$$

which are summarized in the following lemma. We suppress in following the subscripts of x_0 and β for notational convenience.

Lemma 1. *The direction vector of the PAN estimator,* $\hat{\gamma}(\lambda)$*, fulfills the equation*

$$\frac{\hat{\gamma}(\lambda)^T A \hat{\gamma}(\lambda)}{(\hat{\gamma}(\lambda)^T X^T X \hat{\gamma}(\lambda))^2} X^T X \hat{\gamma}(\lambda) - \lambda(\hat{\gamma}(\lambda)^T B \hat{\gamma}(\lambda))\hat{\gamma}(\lambda) = \frac{1}{\hat{\gamma}(\lambda)^T X^T X \hat{\gamma}(\lambda)} A\hat{\gamma}(\lambda) - \lambda B \hat{\gamma}(\lambda),$$

where $A = X^T X \tilde{\beta}\tilde{\beta}^T X^T X$ *and* $B = x_0 x_0^T / \|x_0\|^2$*, while the length of the PAN estimator,* $\hat{r}(\lambda)$*, is given by the direction vector as*

$$\hat{r}(\lambda) = \frac{\tilde{\beta}^T X^T X \hat{\gamma}(\lambda)}{\hat{\gamma}(\lambda)^T X^T X \hat{\gamma}(\lambda)}.$$

The proof of Lemma 1 is given in the Appendix.

2.3 Orthonormal design case

Insight regarding the behavior of the PAN penalty in both Cartesian and hyperspherical coordinates is gained by considering the case of the orthonormal design matrix, $X^T X = I_p$. The PAN estimator and prediction are then given explicitly.

Lemma 2. *Assuming an orthonormal design matrix, $X^TX = I_p$, the length of the PAN estimator is given by*

$$\hat{r}(\lambda) = \tilde{\beta}^T\hat{\gamma}(\lambda) = \left(\frac{1}{2} + \frac{1}{2}\, c(\lambda)\right)^{\frac{1}{2}} \|\tilde{\beta}\|,$$

while the direction vector of the PAN estimator equals the first normalized eigenvector of the following $p \times p$ matrix of rank 2

$$M_0 := \tilde{\beta}\tilde{\beta}^T - \frac{\lambda}{\|x_0\|^2} x_0 x_0^T.$$

The direction vector is given by

$$\hat{\gamma}(\lambda) = \left(\frac{1}{2} + \frac{1}{2}\, c(\lambda)\right)^{\frac{1}{2}} \frac{\tilde{\beta}}{\|\tilde{\beta}\|} - \left(\frac{1}{2} - \frac{1}{2}\, c(\lambda)\right)^{\frac{1}{2}} \frac{\|\tilde{\beta}\|^2 x_0 - (x_0^T\tilde{\beta})\tilde{\beta}}{\|\tilde{\beta}\|\sqrt{\|\tilde{\beta}\|^2\|x_0\|^2 - (x_0^T\tilde{\beta})^2}},$$

and depends on the tuning parameter, λ, through

$$c(\lambda) = \frac{\|\tilde{\beta}\|^2(\|\tilde{\beta}\|^2 + \lambda) - 2\lambda(x_0^T\tilde{\beta})^2/\|x_0\|^2}{\|\tilde{\beta}\|^2\sqrt{(\|\tilde{\beta}\|^2 + \lambda)^2 - 4\lambda(x_0^T\tilde{\beta})^2/\|x_0\|^2}}. \tag{7}$$

The PAN estimator is then given by

$$\hat{\beta}_{x_0}(\lambda) = \hat{r}(\lambda)\hat{\gamma}(\lambda) = \frac{1}{2}\left(1 + c(\lambda)\right)\tilde{\beta} - \frac{1}{2}\left(1 - c^2(\lambda)\right)^{\frac{1}{2}} \frac{\|\tilde{\beta}\|^2 x_0 - (x_0^T\tilde{\beta})\tilde{\beta}}{\sqrt{\|\tilde{\beta}\|^2\|x_0\|^2 - (x_0^T\tilde{\beta})^2}}.$$

The proof of Lemma 2 is found in the Appendix. For $\lambda = 0$, the constant in Equation (7) is $c(0) = 1$, such that the PAN estimator equals the OLS estimator. In the limit, $\lim_{\lambda\to\infty} c(\lambda) = 1 - \frac{2(x_0^T\tilde{\beta})^2}{\|\tilde{\beta}\|^2\|x_0\|^2}$, such that the length and direction vector converge to

$$\lim_{\lambda\to\infty} \hat{r}(\lambda) = \left(1 - \frac{(x_0^T\tilde{\beta})^2}{\|\tilde{\beta}\|^2\|x_0\|^2}\right)^{\frac{1}{2}} \|\tilde{\beta}\|,$$

$$\lim_{\lambda\to\infty} \hat{\gamma}(\lambda) = \frac{\|x_0\|}{\sqrt{\|\tilde{\beta}\|^2\|x_0\|^2 - (x_0^T\tilde{\beta})^2}} \left(\tilde{\beta} - \frac{x_0^T\tilde{\beta}}{x_0^T x_0} x_0\right).$$

The direction vector is then equal to the normalized projection of $\tilde{\beta}$ onto H_0. From Lemma (2), it is seen that the PAN estimator depends on the tuning parameter, λ, through the direction vector.

Corollary 1. Assuming an orthonormal design matrix, the PAN prediction of the outcome $\hat{y}_0$ given the covariate vector x_0 is given by

$$\hat{y}_0 = x_0^T \hat{\beta}_{x_0}(\lambda) = x_0^T \tilde{\beta}\, \eta(\lambda; x_0),$$

where $x_0^T \tilde{\beta}$ is the OLS prediction and $\eta(\lambda; x_0)$ is a shrinkage factor varying with x_0

$$\eta(\lambda; x_0) = \frac{1}{2} + \frac{1}{2} \frac{1 - \lambda'}{\sqrt{(1+\lambda')^2 - 4\lambda' \,\mathrm{CosSim}^2(x_0, \tilde{\beta})}}, \quad \lambda' = \lambda / \|\tilde{\beta}\|^2, \tag{8}$$

through $\mathrm{CosSim}(x_0, \tilde{\beta})$, the cosine similarity between x_0 and the OLS estimator $\tilde{\beta}$.

The proof of Corollary 1 is found in the Appendix. In the limit, $\lambda \to \infty$, the prediction converges to $x_0^T \hat{\beta}(\lambda) \to x_0^T \tilde{\beta}\,[1/2 - 1/2] = 0$, while for $\lambda \to -\infty$, the prediction converges to $x_0^T \hat{\beta}(\lambda) \to x_0^T \tilde{\beta}\,[1/2 + 1/2] = x_0^T \tilde{\beta}$, the OLS prediction. In the latter case where λ decreases from 0, the prediction will first increase or expand. At a certain value of λ, however, the length of the regression vector will cancel out the effect of the rotation in the direction vector, such that the prediction will start to decrease and in the end converge to the OLS prediction.

Figure 4 shows how the shrinkage factor $\eta(\lambda; x_0)$ of the PAN prediction in Equation (8) varies with the cosine similarity between x_0 and $\tilde{\beta}$ when the lengths of x_0 and $\tilde{\beta}$ are fixed to unit length, $\|x_0\| = \|\tilde{\beta}\| = 1$. The shrinkage factors are shown for different values of the PAN tuning parameter. The factor for a positive tuning parameter is the smallest, giving the strongest shrinkage, for the cosine similarity values closest to zero and increases to 1 when the cosine similarity approaches 1 and -1. The shrinkage becomes stronger with an increasing tuning parameter, but it inverts if the parameter becomes negative. Then the "expansion" factor is strongest for the smallest cosine similarities in absolute value. The ridge shrinkage, in comparison, does not vary with x_0 and is constant across the cosine similarity.

2.4 Prediction error

The main aim of personalizing a prediction is to lower the prediction error for each given covariate vector, x_0, instead of minimizing the average prediction error (Hellton and Hjort, 2018; Huang et al., 2019). The predictive performance of the regression methods can be evaluated by the mean squared error (MSE) of the prediction for covariate vector x_0 under the linear model

$$\mathrm{MSE}(x_0, \beta, \lambda) = E\left[(x_0^T \hat{\beta}(\lambda) - x_0^T \beta)^2 \mid X\right],$$

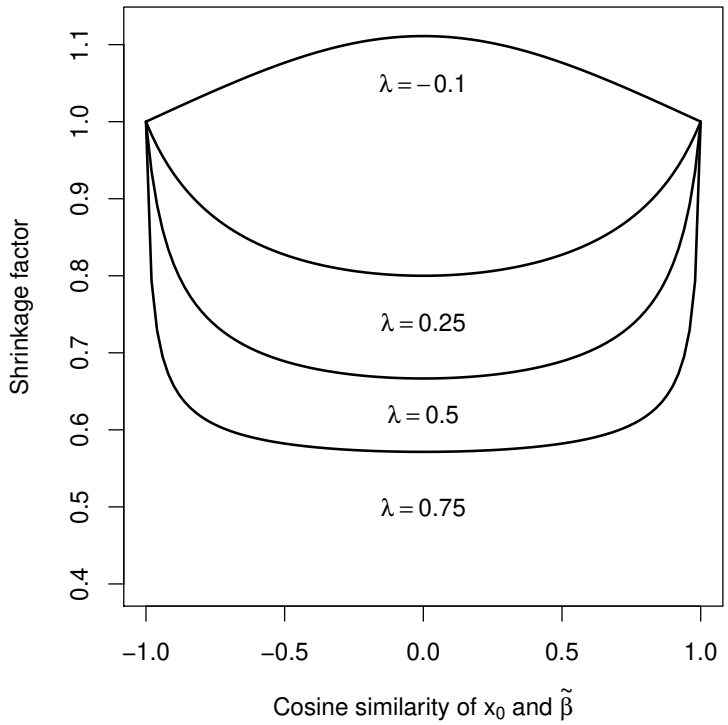

Figure 4: The shrinkage factor of the PAN prediction, $\eta(\lambda; x_0)$, as a function of the cosine similarity between x_0 and $\tilde{\beta}$ for different tuning parameter values. The length of $\tilde{\beta}$ and x_0 are fixed to $\|\tilde{\beta}\| = \|x_0\| = 1$. When x_0 changes, the shrinkage factor would also change. The shrinkage factor of ridge regression, in comparison, would be constant across the cosine similarity.

related to the prediction error as $E\left[(x_0^T\hat{\beta}(\lambda) - y_0)^2 \mid X\right] = \mathrm{MSE}(x_0, \beta, \lambda) + \sigma^2$. For a given x_0, we will compare the predictions in terms of the MSE to omit the intrinsic error σ^2. Later, the average MSE on a test set is used to evaluate the overall prediction performance. We first present a lemma demonstrating the behavior of the optimal λ in terms of minimum MSE. We scale the design matrix by n to ensure the asymptotic convergence of the OLS estimator.

Lemma 3. *Under a scaled orthogonal design matrix, $X^TX = nI_p$, the derivative of the mean squared error with respect to λ evaluated at 0 is given by*

$$\left.\frac{\partial\,\mathrm{MSE}(x_0, \beta, \lambda)}{\partial\lambda}\right|_{\lambda=0} = -\frac{1}{n^2}C_1\Big(1 - 4\frac{(x_0^T\beta)^2}{\|x_0\|^2\|\beta\|^2}\Big) + O\left(\frac{1}{n^3}\right),$$

with the positive constant $C_1 = 2\sigma^2\frac{\|x_0\|^2}{\|\beta\|^2}\big(1 - \frac{(x_0^T\beta)^2}{\|x_0\|^2\|\beta\|^2}\big) > 0$ for $\sigma > 0$, $x_0, \beta \neq \mathbf{0}$ and

$|x_0^T\beta| \neq \|x_0\|\|\beta\|$.

The proof of Lemma 3 can be found in the Appendix. As the value $\lambda = 0$ corresponds to OLS regression, Lemma 3 reveals when the PAN penalty improves the mean squared error of the prediction, compared to OLS.

Theorem 1. *Assume a scaled orthogonal design matrix, $X^TX = nI_p$, and $\sigma > 0$, $x_0, \beta \neq \mathbf{0}$ and $|x_0^T\beta| \neq \|x_0\|\|\beta\|$. Then if $|x_0^T\beta| < \frac{1}{2}\|x_0\|\|\beta\|$, there exists a $\lambda > 0$, and if $|x_0^T\beta| > \frac{1}{2}\|x_0\|\|\beta\|$, there exists a $\lambda < 0$, for which the mean squared error asymptotically as $n \to \infty$ satisfies the inequality*

$$\mathrm{MSE}(x_0, \beta, \lambda) < \mathrm{MSE}(x_0, \beta, 0) = \mathrm{MSE}_{OLS}(x_0, \beta).$$

When $|x_0^T\beta| = \frac{1}{2}\|x_0\|\|\beta\|$, the minimum of the mean squared error is asymptotically obtained for $\lambda = 0$.

The proof of Lemma 3 and Theorem 1 can be found in the Appendix. Theorem 1 demonstrates that unless $|\,\mathrm{CosSim}(x_0, \beta)| = 0.5$, there always exists a $\lambda \neq 0$ for which the MSE of PAN is smaller than the MSE of OLS. The challenge in practice is to estimate this optimal value from data. Theorem 1 also shows that the sign of the optimal value for λ is dependent on whether the absolute value of $x_0^T\beta$ is smaller or larger than $\frac{1}{2}\|x_0\|\|\beta\|$. This again corresponds to the absolute value of the cosine similarity between x_0 and β being smaller or larger than 0.5. For small cosine similarities, the optimal PAN tuning parameter is hence positive, while for large cosine similarities the optimal value will be negative.

Remark 3. Lemma 3 further reveals that the benefit of estimating a common PAN tuning parameter for all observations will depend on the dimension. For instance, if the covariates are assumed to be standard normally distributed in p dimensions, $x_0 \sim N(0, I_p)$, for a fixed, arbitrary, β, the cosine similarity between x_0 and β, $z = x_0^T\beta/(\|x_0\|\|\beta\|)$, follows the distribution

$$f_p(z) = \frac{1}{\sqrt{\pi}} \frac{\Gamma(\frac{p}{2})}{\Gamma(\frac{p-1}{2})} \left(1 - z^2\right)^{\frac{p-3}{2}}, \quad \text{for } -1 < z < 1,$$

where $\Gamma(\cdot)$ is the gamma function (see Cho, 2009). The proportion of covariate vectors with a cosine similarity between -1/2 and 1/2, i.e. the covariate vectors that will benefit from a positive PAN parameter, will greatly increase with the dimension p. For $p = 2$ and 3, this proportion is 1/3 and 1/2, respectively. Hence, in dimensions two and three, a third or a half of the observations will benefit from a positive tuning parameter value, while the rest will in fact benefit from a negative value. Therefore if we select one common tuning

parameter, either positive or negative, it will be unsuitable for a substantial proportion of the possible covariate vectors. But importantly, the proportion benefiting from a positive tuning parameter will increase rapidly with p, e.g. to 74.7% for $p = 6$, 95.1% for $p = 15$ and 99.6% for $p = 30$. The selection of a common (positive) tuning parameter value will therefore be more beneficial for the prediction error in higher dimensions.

3 Algorithm

In this section, we propose a naïve algorithm to efficiently calculate an approximation of the PAN estimator for a general design matrix. The algorithm is given as an alternative to numerical optimization of the objective function in Definition 1, and it is based on iteratively solving an eigen-equation and updating the PAN solution, along a path of tuning parameters.

Let the three quadratic forms in Lemma 1 be denoted by $c_1(\gamma) = \gamma^T X^T X \gamma$, $c_2(\gamma) = \gamma^T A \gamma$ and $c_3(\gamma) = \gamma^T B \gamma$, where $A = X^T X \tilde{\beta}\tilde{\beta}^T X^T X$ and $B = x_0 x_0^T / \|x_0\|^2$. By rearranging the terms in Lemma 1, the PAN direction vector $\hat{\gamma}$ has to fulfill the equation

$$\underbrace{\left[c_1(\hat{\gamma})A - c_2(\hat{\gamma})X^T X - \lambda B c_2^2(\hat{\gamma})\right]}_{M(\hat{\gamma})} \hat{\gamma} = -\lambda c_2^2(\hat{\gamma}) c_3^2(\hat{\gamma})\, \hat{\gamma}, \tag{9}$$

where $M(\gamma)$ is a matrix. If the constants $c_1(\gamma)$ and $c_2(\gamma)$ are fixed to an initial value of the direction vector, γ_0, we can fix the matrix $M(\gamma)$ to be $M(\gamma_0) = M^*$. Then Equation (9) becomes an eigen-equation, $M^*\gamma^* = \theta\gamma^*$, which can be explicitly solved. The PAN direction vector can then be approximated by the first normalized eigenvector, γ_1^* of the matrix M^*. For $\lambda = 0$, the direction vector $\hat{\gamma}$ equals the normalized OLS estimator. If we slowly increase the tuning parameter from $\lambda = 0$, we can initialize the approximations of the constants by the OLS estimator, and then iteratively update the direction vector and the quadratic forms along a path of tuning parameters. This way we can construct an algorithm to efficiently calculate an approximate PAN estimator for a sequence of tuning parameters. The sequence starts from $\lambda = 0$ and stops at a positive or negative value λ_{stop} using a step size $\Delta\lambda > 0$. As the normalized eigenvector is not uniquely defined up to a constant ± 1, the sign is determined by ensuring that the PAN length is positive, before updating the direction vector. The algorithm for the PAN estimator along the path of tuning parameters is given as follows

Algorithm 1: Naïve algorithm for approximate Personalized Angle Regression

Input : Data matrix X, OLS estimate $\tilde{\beta}$ and covariate vector x_0.
Step size $\Delta\lambda > 0$ and stop value $\lambda_{\text{stop}} \in \mathbb{R}$.

Output: Approximation of PAN estimate $\hat{\beta}^*_{x_0}(\lambda_k)$ for a path of tuning parameters.

Initialize $\gamma_0 = \tilde{\beta}$, $\lambda_0 = 0$ and $k = 0$.
Define the matrices $A = X^T X \tilde{\beta}\tilde{\beta}^T X^T X$ and $B = x_0 x_0^T / \|x_0\|^2$, and the constants $c_1(\gamma) = \gamma^T X^T X \gamma$ and $c_2(\gamma) = \gamma^T A \gamma$.
while $|\lambda_k| < |\lambda_{\text{stop}}|$ **do**
  Increment: $k = k + 1$.
  Increment: $\lambda_k = \lambda_{k-1} + \text{sgn}(\lambda_{\text{stop}})\Delta\lambda$.
  Calculate the first normalized eigenvector, v_1, of the matrix

$$M(\gamma_{k-1}) = c_1(\gamma_{k-1})A - c_2(\gamma_{k-1})X^T X - \lambda_k c_1^2(\gamma_{k-1})B.$$

  if $\tilde{\beta}^T X^T X v_1 \geq 0$, **then**
    update the direction vector: $\gamma_k = v_1$,
  else
    change the sign of the eigenvector and update: $\gamma_k = -v_1$.
  end
  Calculate and update the length: $r_k = \frac{\tilde{\beta}^T X^T X \gamma_k}{\gamma_k^T X^T X \hat{\gamma}_k}$.
  Output $r_k \gamma_k$
end

The stop value of the tuning parameter may be either positive or negative. In the following simulations and data example, the analyses will first use a positive stop value and then repeat the full algorithm with a negative stop value to cover the full range of tuning parameters. An implementation of the algorithm is found in the R package `panreg`, available at `http://github.com/khellton/panreg`.

4 Simulations

In this section, we present a simulation study comparing PAN regression to OLS, ridge, lasso and elastic net regression. For all settings, we simulated 200 data sets consisting of $n = 50$

observations from a linear model with $p = 6, 15$ and 30:

$$y_i = x_i^T \beta + \varepsilon_i, \quad i = 1, \ldots, 50,$$

where the noise is normally distributed $\varepsilon_i \sim N(0, \sigma^2)$ with $\sigma = 1$. An independent test set with $n_{test} = 1000$ observations was predicted for each simulation. To select the tuning parameter of the PAN penalty, we used the parametric bootstrap procedure described in Section 4.1. For all instances, the number of bootstrap samples was set to $B = 2000$. For lasso, elastic net and ridge regression, the tuning parameters were chosen by cross-validation. There were two simulation setups, where the first used an orthonormal design matrix and the second setup used a correlated design matrix with three levels of correlation. In the first setup with an orthonormal design, the explicit solution of the PAN estimator given in Lemma 2 was used, while in the second simulation setup, Algorithm 1 was used to calculate the PAN estimator.

4.1 Selecting the tuning parameter

We propose to select the tuning parameter in PAN regression by the following procedure based on parametric bootstrap (Efron and Tibshirani, 1994):

1. Use the OLS estimates $\tilde{\beta}$ and $\tilde{\sigma}^2$ as plug-in estimates to simulate $r = 1, \ldots, B$ bootstrap samples of n observations $Y^{(r)} = [y_1^{(r)}, \ldots, y_n^{(r)}]^T$ from

   $$y_i^{(r)} = x_i^T \tilde{\beta} + \varepsilon_i, \quad \varepsilon_i \sim N(0, \tilde{\sigma}^2), \qquad i = 1, \ldots, n,$$

   to produce the rth bootstrap sample OLS estimate $\tilde{\beta}^{(r)}$ for $r = 1, \ldots, B$.

2. Over a suitable grid of λ, hold the tuning parameter value fixed:

   - find the prediction $x_i^T \hat{\beta}_{x_i}^{(r)}(\lambda)$ for each bootstrap sample r and vector x_i,
   - calculate the mean squared prediction error given by the squared bias, corrected for the variance of the bias, and the variance, over all i and r:

     $$\text{MSE} = \Big(\text{Bias}(x_i^T \hat{\beta}_{x_i}^{(r)}(\lambda), x_i^T \tilde{\beta})^2 - \text{Var Bias}\Big)_+ + \text{Var}(x_i^T \hat{\beta}_{x_i}^{(r)}(\lambda)),$$

     where $\text{Var Bias} = \text{Var}\left(x_i^T \hat{\beta}_{x_i}^{(r)}(\lambda) - x_i^T \tilde{\beta}^{(r)}\right)$ and $(\cdot)_+ = \max\{\cdot, 0\}$.

3. Select the tuning parameter value, $\hat{\lambda}$, with the smallest mean squared error over the grid of λ.

The same procedure was used by Hellton and Hjort (2018) to estimate the personalized tuning parameter in ridge regression. We subtract the variance of the bias to correct for the overestimation when squaring the bias directly (see Claeskens and Hjort, 2008, p. 150, for further details).

4.2 Orthogonal design

In the first setup, the data matrix was simulated from a standard normal distribution and transformed to be orthonormal, such that $X^TX = I_p$. Table 1 shows the mean squared error over the test set sample, $\frac{1}{1000}\sum_{i=1}^{1000}(x_i^T\hat{\beta} - x_i^T\beta)^2$ averaged over 200 simulations, for the OLS, lasso, elastic net, ridge and PAN regression estimator. The standard deviation of the mean squared error over the 200 simulations is shown in parentheses. The simulations were performed for four scenarios of increasing signal strengths with equal regression coefficients: 1) $\beta_j = 0.05, \forall j$, 2) $\beta_j = 0.1, \forall j$, 3) $\beta_j = 0.15, \forall j$ and 4) $\beta_j = 0.2, \forall j$. The four values of β_j were chosen such that ridge regression would yield an improvement compared to OLS. The PAN tuning parameter was found using the parametric bootstrap procedure from Section 4.1 with $B = 2000$.

The results of Table 1 show that for $p = 6$, PAN regression has a lower prediction error than ridge regression for $\beta_j = 0.05$ and 0.10, the smallest signal strengths, while ridge regression is better for $\beta_j = 0.15$ and 0.20. As the dimension increases to $p = 15$ and 30, PAN regression performs increasingly better compared to ridge regression. For $p = 30$, PAN gives a lower or equal prediction error than ridge regression for all signal strengths. This suggests that the PAN tuning parameter may be difficult to set correctly if the dimension is small. We see that when the covariates are uncorrelated PAN and ridge give a similar performance. For all settings, both lasso and elastic net perform worse than ridge and PAN regression due to the non-sparse regression coefficients. The standard deviations of the MSE for PAN are similar to (or lower than) the standard deviations for ridge regression. This demonstrates that the improvement in MSE for the PAN procedure is not achieved at the expense of the stability. The lower standard deviations of the MSE for PAN, seen especially for the higher dimensions, are due to the fixed tuning parameter grid.

4.3 Correlated design

In the second setup, the data matrix was simulated from a normal distribution with correlated covariates where $\Sigma_{ij} = \rho^{|i-j|}, i, j = 1, \ldots, p$, and $\rho = 0.2, 0.5$ and 0.8. The regression

				$p = 6$				
Method	$\lvert\beta_j\rvert = 0.05$		$\lvert\beta_j\rvert = 0.10$		$\lvert\beta_j\rvert = 0.15$		$\lvert\beta_j\rvert = 0.20$	
OLS	0.133	(0.076)	0.125	(0.066)	0.124	(0.065)	0.123	(0.070)
Lasso (CV)	0.062	(0.061)	0.094	(0.044)	0.136	(0.055)	0.159	(0.066)
Elastic net (CV)	0.058	(0.055)	0.091	(0.045)	0.133	(0.051)	0.154	(0.067)
Ridge (CV)	0.031	(0.035)	0.057	(0.031)	**0.088**	(0.041)	**0.103**	(0.059)
PAN	**0.029**	(0.037)	**0.055**	(0.031)	0.090	(0.040)	0.109	(0.059)
				$p = 15$				
Method	$\lvert\beta_j\rvert = 0.05$		$\lvert\beta_j\rvert = 0.10$		$\lvert\beta_j\rvert = 0.15$		$\lvert\beta_j\rvert = 0.20$	
OLS	0.317	(0.109)	0.308	(0.114)	0.290	(0.117)	0.308	(0.108)
Lasso (CV)	0.095	(0.072)	0.189	(0.065)	0.302	(0.074)	0.380	(0.148)
Elastic net (CV)	0.087	(0.065)	0.183	(0.064)	0.292	(0.074)	0.363	(0.146)
Ridge (CV)	0.058	(0.05)	0.128	(0.048)	0.194	(0.071)	0.241	(0.104)
PAN	**0.054**	(0.045)	**0.118**	(0.043)	**0.187**	(0.061)	**0.239**	(0.093)
				$p = 30$				
Method	$\lvert\beta_j\rvert = 0.05$		$\lvert\beta_j\rvert = 0.10$		$\lvert\beta_j\rvert = 0.15$		$\lvert\beta_j\rvert = 0.20$	
OLS	0.606	(0.154)	0.610	(0.146)	0.614	(0.149)	0.605	(0.153)
Lasso (CV)	0.136	(0.085)	0.328	(0.064)	0.609	(0.093)	0.860	(0.309)
Elastic net (CV)	0.122	(0.066)	0.318	(0.055)	0.587	(0.103)	0.814	(0.320)
Ridge (CV)	0.101	(0.064)	0.248	(0.069)	0.370	(0.106)	**0.436**	(0.114)
PAN	**0.095**	(0.059)	**0.231**	(0.064)	**0.351**	(0.082)	0.439	(0.113)

Table 1: The average and the standard deviation (in parentheses) of the mean squared error on the test set ($n_{test} = 1000$) over 200 simulations with $n = 50$ and $\sigma = 1$ in the training set. The simulations were carried out for an orthogonal design with $p = 6, 15, 30$ variables and four values of equal regression coefficients. The lowest prediction error across the methods is shown in bold.

	$p = 6$							
Method	$\lvert\beta_j\rvert = 0.05$		$\lvert\beta_j\rvert = 0.10$		$\lvert\beta_j\rvert = 0.15$		$\lvert\beta_j\rvert = 0.20$	
OLS	0.128	(0.077)	0.132	(0.077)	0.147	(0.089)	0.133	(0.085)
Lasso (CV)	0.053	(0.062)	0.082	(0.059)	0.128	(0.061)	0.170	(0.060)
Elastic net (CV)	0.051	(0.058)	0.082	(0.054)	0.124	(0.056)	0.168	(0.062)
Ridge (CV)	0.023	(0.030)	0.051	(0.03)	0.091	(0.041)	0.122	(0.058)
PAN	**0.018**	(0.023)	**0.046**	(0.031)	**0.085**	(0.044)	**0.112**	(0.052)
	$p = 15$							
Method	$\lvert\beta_j\rvert = 0.05$		$\lvert\beta_j\rvert = 0.10$		$\lvert\beta_j\rvert = 0.15$		$\lvert\beta_j\rvert = 0.20$	
OLS	0.426	(0.173)	0.445	(0.191)	0.430	(0.224)	0.420	(0.185)
Lasso (CV)	0.071	(0.075)	0.145	(0.086)	0.267	(0.126)	0.400	(0.096)
Elastic net (CV)	0.066	(0.074)	0.141	(0.089)	0.256	(0.100)	0.390	(0.081)
Ridge (CV)	0.042	(0.053)	0.115	(0.071)	0.217	(0.076)	0.325	(0.088)
PAN	**0.036**	(0.030)	**0.096**	(0.053)	**0.170**	(0.060)	**0.247**	(0.079)
	$p = 30$							
Method	$\lvert\beta_j\rvert = 0.05$		$\lvert\beta_j\rvert = 0.10$		$\lvert\beta_j\rvert = 0.15$		$\lvert\beta_j\rvert = 0.20$	
OLS	1.470	(0.546)	1.435	(0.475)	1.424	(0.595)	1.439	(0.505)
Lasso (CV)	0.140	(0.176)	0.282	(0.118)	0.597	(0.237)	0.925	(0.268)
Elastic net (CV)	0.118	(0.127)	0.265	(0.094)	0.552	(0.175)	0.893	(0.210)
Ridge (CV)	0.085	(0.157)	0.229	(0.113)	0.441	(0.166)	0.638	(0.177)
PAN	**0.081**	(0.083)	**0.198**	(0.055)	**0.395**	(0.108)	**0.595**	(0.130)

Table 2: The average and the standard deviation (in parentheses) of the mean squared error on the test set ($n_{test} = 1000$) over 200 simulations with $n = 50$ and $\sigma = 1$ in the training set. The simulations were carried out for correlated covariates with covariance matrix $\Sigma_{ij} = 0.2^{|i-j|}$, for $i, j = 1, \ldots, p$, with $p = 6, 15, 30$ variables and four values of equal regression coefficients with alternating signs. The lowest prediction error across the methods is shown in bold.

	$p = 6$							
Method	$\lvert\beta_j\rvert = 0.05$		$\lvert\beta_j\rvert = 0.10$		$\lvert\beta_j\rvert = 0.15$		$\lvert\beta_j\rvert = 0.20$	
OLS	0.140	(0.098)	0.141	(0.077)	0.136	(0.072)	0.132	(0.078)
Lasso (CV)	0.061	(0.083)	0.075	(0.068)	0.099	(0.058)	0.137	(0.070)
Elastic net (CV)	0.058	(0.086)	0.068	(0.059)	0.092	(0.054)	0.131	(0.066)
Ridge (CV)	0.030	(0.058)	0.043	(0.039)	0.065	(0.033)	0.096	(0.038)
PAN	**0.024**	(0.055)	**0.035**	(0.034)	**0.054**	(0.024)	**0.082**	(0.032)
	$p = 15$							
Method	$\lvert\beta_j\rvert = 0.05$		$\lvert\beta_j\rvert = 0.10$		$\lvert\beta_j\rvert = 0.15$		$\lvert\beta_j\rvert = 0.20$	
OLS	0.423	(0.214)	0.428	(0.182)	0.431	(0.169)	0.394	(0.161)
Lasso (CV)	0.077	(0.115)	0.106	(0.088)	0.165	(0.066)	0.253	(0.083)
Elastic net (CV)	0.067	(0.097)	0.094	(0.066)	0.160	(0.065)	0.250	(0.079)
Ridge (CV)	0.045	(0.106)	0.075	(0.065)	0.133	(0.043)	0.220	(0.054)
PAN	**0.036**	(0.070)	**0.061**	(0.031)	**0.107**	(0.029)	**0.163**	(0.044)
	$p = 30$							
Method	$\lvert\beta_j\rvert = 0.05$		$\lvert\beta_j\rvert = 0.10$		$\lvert\beta_j\rvert = 0.15$		$\lvert\beta_j\rvert = 0.20$	
OLS	1.410	(0.595)	1.440	(0.545)	1.458	(0.536)	1.382	(0.491)
Lasso (CV)	0.122	(0.150)	0.208	(0.215)	0.342	(0.257)	0.556	(0.284)
Elastic net (CV)	0.096	(0.111)	0.170	(0.133)	0.324	(0.207)	0.511	(0.228)
Ridge (CV)	0.067	(0.100)	0.156	(0.193)	0.299	(0.256)	0.469	(0.205)
PAN	**0.064**	(0.070)	**0.127**	(0.070)	**0.236**	(0.122)	**0.355**	(0.104)

Table 3: The average and the standard deviation (in parentheses) of the mean squared error on the test set ($n_{test} = 1000$) over 200 simulations with $n = 50$ and $\sigma = 1$ in the training set. The simulations were carried out for correlated covariates with covariance matrix $\Sigma_{ij} = 0.5^{|i-j|}$, for $i, j = 1, \ldots, p$, with $p = 6, 15, 30$ variables and four values of equal regression coefficients with alternating signs. The lowest prediction error across the methods is shown in bold.

				$p = 6$				
Method	$\lvert\beta_j\rvert = 0.05$		$\lvert\beta_j\rvert = 0.10$		$\lvert\beta_j\rvert = 0.15$		$\lvert\beta_j\rvert = 0.20$	
OLS	0.136	(0.085)	0.144	(0.089)	0.142	(0.079)	0.128	(0.080)
Lasso (CV)	0.053	(0.075)	0.067	(0.080)	0.083	(0.077)	0.097	(0.068)
Elastic net (CV)	0.050	(0.072)	0.061	(0.073)	0.075	(0.068)	0.093	(0.063)
Ridge (CV)	0.028	(0.049)	0.036	(0.053)	0.047	(0.047)	0.060	(0.036)
PAN	**0.021**	(0.044)	**0.031**	(0.050)	**0.040**	(0.042)	**0.055**	(0.034)
				$p = 15$				
Method	$\lvert\beta_j\rvert = 0.05$		$\lvert\beta_j\rvert = 0.10$		$\lvert\beta_j\rvert = 0.15$		$\lvert\beta_j\rvert = 0.20$	
OLS	0.440	(0.194)	0.415	(0.188)	0.432	(0.181)	0.407	(0.164)
Lasso (CV)	0.086	(0.131)	0.083	(0.100)	0.111	(0.126)	0.142	(0.098)
Elastic net (CV)	0.074	(0.119)	0.077	(0.090)	0.113	(0.128)	0.133	(0.084)
Ridge (CV)	0.047	(0.088)	0.054	(0.082)	0.090	(0.098)	0.113	(0.067)
PAN	**0.031**	(0.057)	**0.042**	(0.054)	**0.066**	(0.056)	**0.086**	(0.032)
				$p = 30$				
Method	$\lvert\beta_j\rvert = 0.05$		$\lvert\beta_j\rvert = 0.10$		$\lvert\beta_j\rvert = 0.15$		$\lvert\beta_j\rvert = 0.20$	
OLS	1.423	(0.461)	1.408	(0.569)	1.441	(0.516)	1.390	(0.479)
Lasso (CV)	0.091	(0.144)	0.121	(0.166)	0.168	(0.131)	0.228	(0.120)
Elastic net (CV)	0.066	(0.086)	0.100	(0.113)	0.165	(0.162)	0.212	(0.107)
Ridge (CV)	0.055	(0.135)	0.082	(0.133)	0.134	(0.152)	0.209	(0.139)
PAN	**0.039**	(0.051)	**0.068**	(0.071)	**0.112**	(0.071)	**0.164**	(0.073)

Table 4: The average and the standard deviation (in parentheses) of the mean squared error on the test set ($n_{test} = 1000$) over 200 simulations with $n = 50$ and $\sigma = 1$ in the training set. The simulations were carried out for correlated covariates with covariance matrix $\Sigma_{ij} = 0.8^{|i-j|}$, for $i, j = 1, \ldots, p$, with $p = 6, 15, 30$ variables and four values of equal regression coefficients with alternating signs. The lowest prediction error across the methods is shown in bold.

coefficients have the same absolute values as in the previous setup, but with alternating signs: 1) $\beta_j = 0.05 \cdot (-1)^j, \forall j$, 2) $\beta_j = 0.1 \cdot (-1)^j, \forall j$, 3) $\beta_j = 0.15 \cdot (-1)^j, \forall j$ and 4) $\beta_j = 0.2 \cdot (-1)^j$, e.g. $\beta = [-0.05, 0.05, -0.05, \dots]$, for $p = 6, 15$ and 30. The PAN tuning parameter was found using the parametric bootstrap procedure with $B = 2000$ on the same tuning parameter grid used by Algorithm 1 to calculate the estimate. For $\rho = 0.2$ and 0.5, the algorithm was first run with step size $\Delta\lambda = 1$ and stopping values $\lambda_{\text{stop}} = 100, 150, 400$, for $p = 6, 15, 40$ and then re-run with $\Delta\lambda = 1$ and stopping value $\lambda_{\text{stop}} = -20$ for all p. For $\rho = 0.8$, PAN required a higher tuning parameter value and the algorithm was, for all p, first run with $\Delta\lambda = 15$ and $\lambda_{\text{stop}} = 1500$ and then re-run with $\Delta\lambda = 1$ and $\lambda_{\text{stop}} = -20$.

Table 2 shows the mean squared error for the test set averaged over 200 simulations for the OLS, lasso, elastic net, ridge and PAN estimator, in the case of $\rho = 0.2$. The standard deviations of the mean squared error over the 200 simulations are shown in parentheses. It is seen that improvement in the prediction error of PAN increases as the dimension increases, as was observed in the orthogonal case. Secondly, PAN performs better than ridge regression for all the different signal strengths, in particular for higher dimensions. This is due to the correlation between the covariates and that the regression coefficients are such that the distribution of the inner products between the covariate vectors and the β vector is more concentrated around zero than in the uncorrelated setup. Finally, lasso and elastic net again perform worse than ridge and PAN regression due to the non-sparse regression coefficients.

Tables 3 and 4 display the mean squared error when $\rho = 0.5$ and $\rho = 0.8$, respectively. The standard deviations of the MSE over the 200 simulations are shown in parentheses. Tables 3 and 4 show that PAN performs better than the other methods for the correlated setup and that the improvement increases with the correlation. The standard deviations of the MSE for PAN are lower than the standard deviations for ridge regression and demonstrate again that the improvement in prediction error is not associated with a loss of stability. The lower standard deviation seen for PAN is due to the fixed tuning parameter grid.

5 Example: Prostate cancer data

We demonstrate PAN regression on a classical dataset previously used to illustrate penalized regression methods (Tibshirani, 1996). The dataset examines the relation between prostate specific antigen (PSA) and clinical measurements in 97 prostate cancer patients (Stamey et al., 1989). We predicted the log PSA values based on the eight covariates; log tumor volume (`lcavol`), log tumor weight (`lweight`), age (`age`), log of benign prostatic hyperplasia

Table 5: The prediction and regression coefficients of OLS and PAN for the observations with the four largest and smallest cosine similarities between x_0 and $\tilde{\beta}$ in absolute value.

Observation		18	44	24	51	35	4	3	92
Cosine similarity		-0.009	-0.012	0.031	-0.036	-0.704	-0.709	-0.724	0.742
OLS prediction		-0.014	-0.016	0.040	-0.053	-1.208	-1.610	-1.764	1.364
PAN prediction		-0.012	-0.013	0.036	-0.047	-1.203	-1.601	-1.711	1.232
	OLS	PAN coefficients							
`lcavol`	0.576	0.576	0.576	0.576	0.576	0.594	0.583	0.594	0.584
`lweight`	0.231	0.231	0.231	0.231	0.232	0.226	0.235	0.213	0.242
`age`	-0.137	-0.137	-0.137	-0.137	-0.137	-0.138	-0.150	-0.126	-0.135
`lbph`	0.122	0.121	0.121	0.121	0.120	0.116	0.119	0.120	0.105
`svi`	0.273	0.273	0.273	0.275	0.273	0.282	0.291	0.291	0.229
`lcp`	-0.128	-0.128	-0.128	-0.130	-0.130	-0.152	-0.161	-0.169	-0.103
`gleason`	0.031	0.031	0.031	0.032	0.030	0.014	0.023	0.034	0.003
`pgg45`	0.109	0.109	0.108	0.107	0.111	0.119	0.123	0.117	0.138

amount (`lbph`), seminal vesicle invasion (`svi`), log of capsular penetration (`lcp`), Gleason score (`gleason`) and percent of Gleason score 4 or 5 (`ppg45`). The PAN tuning parameter was determined following the procedure described in Section 4.1 with $B = 2000$, while the ridge tuning parameter was chosen by leave-one-out cross-validation. We assessed the out-of-sample prediction error by dividing the data randomly in a training and test set with 1/3 and 2/3 of the observations, respectively.

To illustrate the personalized regression coefficients of PAN, Table 5 displays the OLS and selected PAN regression coefficients for the full data set ($n = 97$) for the optimal value found by the grid search in the bootstrap procedure. The PAN coefficients are calculated for the patients with the four smallest and the four largest cosine similarities between x_0 and $\tilde{\beta}$, in absolute value. Table 5 shows that even though it is the observations with the smallest cosine similarity that experience the largest shrinkage factor for the prediction (as seen in Figure 4), the observations with the highest cosine similarity will experience the largest change in the regression coefficients, compared to the OLS coefficients. The personalized regression coefficients of observation 51 barely change, while for observation 92, the parameter of `svi` decreases from 0.273 to 0.229 and the parameter of `pgg45` increases from 0.109 to 0.138.

Table 6: The mean squared prediction error on a random test set ($n_{test} = 65$) for OLS, PAN and ridge regression with tuning parameters.

Method	OLS	Ridge ($\lambda = 7.53$)	PAN ($\lambda = 2.95$)
Test error	0.378	0.345	0.339

The prediction error of the independent test set ($n_{test} = 65$) is shown in Table 6. The prediction performance of all three regression methods is very similar, but PAN regression has a slightly better test error than ridge and OLS. The tuning parameters were set to 2.95 for PAN and 7.53 for ridge.

Table 7 shows the computation time and the accuracy of Algorithm 1 on the training and test set when the step size $\Delta\lambda$ decreases (with the stop value fixed to $\lambda_{\text{stop}} = 2.95$). We see that when the step size is halved, the number of values in the tuning parameter grid and the computation time approximately doubles. To assess the accuracy, we calculate the average squared difference between the predictions given by the smallest step size ($\Delta\lambda = 0.0025$) and all other step sizes. We see that this average squared difference in the predictions is very small and negligible compared to the test error in Table 6.

Table 7: Computation time and accuracy of Algorithm 1 for decreasing step size ($\Delta\lambda$) and a fixed $\lambda_{\text{stop}} = 2.95$ for the training and test set. The length of the grid gives the number of λ values considered. The squared difference between the predictions from the smallest step size ($\Delta\lambda = 0.0025$) and all other step sizes is averaged over the datasets and the simulations.

			Training set	Test set
Step size ($\Delta\lambda$)	Length of grid	Time (s)	Mean difference	Mean difference
0.1000	30	0.52	$3.3 \cdot 10^{-7}$	$4.4 \cdot 10^{-7}$
0.0500	60	1.05	$1.5 \cdot 10^{-9}$	$2.0 \cdot 10^{-9}$
0.0250	119	2.02	$3.3 \cdot 10^{-10}$	$4.6 \cdot 10^{-10}$
0.0100	296	5.02	$3.7 \cdot 10^{-11}$	$5.1 \cdot 10^{-11}$
0.0050	591	10.12	$4.1 \cdot 10^{-12}$	$5.7 \cdot 10^{-12}$
0.0025	1181	19.78	–	–

6 Discussion

We have introduced a new regression penalty based on the normalized values, or angles, of the regression parameters. The proposed penalty is inherently personalized and is constructed to produce individualized regression coefficients and predictions. The PAN penalty has the advantage over other personalized prediction approaches (Hellton and Hjort, 2018; Huang et al., 2019) that a single, common tuning parameter can be chosen overall based on the training set. The PAN penalty can be defined in both Cartesian and hyperspherical coordinates. The Cartesian formulation (Definition 1) enables simple exact expressions in the orthonormal case, while the hyperspherical formulation (Definition 2) yields a more computationally efficient objective function.

The PAN penalty combines two novel aspects: personalization and shrinking normalized or angular coefficients. Both these aspects should be explored further for their own merit. One may include the personalization without normalizing the predictions, for instance with the penalty: $J(\beta) = \beta^T K \beta = \|\beta\|_K^2$ with $K = x_0 x_0^T / \|x_0\|^2$. This penalty is the same as the square of the group lasso penalty (Yuan and Lin, 2006). In hyperspherical coordinates, it is seen that this penalty is in fact a product of the non-personalized ridge penalty and the PAN penalty:

$$J(\beta) = \beta^T K \beta = r_\beta^2 \cdot \gamma_\beta^T K \gamma_\beta = J_{ridge}(\beta) J_{PAN}(\beta),$$

suggesting that the penalty may be less suitable to exploit the personalization. Further, one could also replace x_0 or $x_0 x_0^T$ in the PAN penalty with a population mean or covariance matrix to omit the personalization and construct a "population" version of the PAN penalty.

Due to the structure of the PAN penalty, the tuning parameter may be both positive and negative, in stark contrast to other penalization methods. This introduces challenges when selecting the tuning parameter value. Initial investigation revealed that (leave-one-out) cross-validation was too unstable for PAN to work well. A more stable procedure based on the parametric bootstrap approach (Section 4.1) was proposed as an alternative, yielding good results in simulations. However, as this procedure depends on a plug-in estimate, extensions to higher dimension require further work. Future work therefore includes to explore improved cross-validation procedures for PAN or develop alternative procedures, possibly marginal maximum likelihood or a Bayesian framework.

In a personalized framework, one aims to make inference regarding a single, specific case which has been and may only be observed once. The advantage of personalization therefore relies on the structure and, in particular, the heterogeneity of the data. Liu and

Meng (2016) commented: "The costs of individualization often outweighed its benefits", but that highly heterogeneous data will benefit more from personalization than homogeneous data. This highlights the opportunity of the Big Data era where data are becoming more heterogeneous. Big Data are typically characterized by a large sample size aggregated from multiple data sources and at different times, creating an intrinsic heterogeneity (Fan et al., 2014). This heterogeneity can be exploited by personalized prediction methods.

Finally, PAN regression may also have a Bayesian formulation which may be beneficial, for instance, for selecting the tuning parameter. Here the PAN penalty corresponds to a Bayesian prior following the generalized von Mises distribution (Gatto and Jammalamadaka, 2007). Future work includes developing improved algorithms for calculating the PAN estimator and to explore the wider class of angle-based penalties, i.e. the penalty corresponding to the lasso or L_1 norm in the angle space. Further, the PAN penalty or other personalized penalties should be extended to logistic regression and generalized linear models, and to more complex methods requiring regularization, such as smoothing spline regression or graphical lasso.

Acknowledgments

The author would like to thank two anonymous reviewers for their insightful and valuable remarks improving the paper, in addition to Claudio Heinrich-Mertsching, Martin Jullum and Nils Lid Hjort for fruitful and challenging discussions, and Mette Langaas, Jens Christian Wahl and Annabelle Redelmeier for helpful comments with the manuscript.

A Appendix

Derivation of Equation (6)

Assuming an orthonormal design matrix, $X^T X = I_2$, the normal equations following from the residual sum-of-squares are given as follows (omitting the index $j = 1$ from the notation of the angle parameters)

$$\begin{aligned} r_\beta \sin(\alpha_\beta) \sum_{i=1}^{n} y_i x_{1i} - r_\beta \cos(\alpha_\beta) \sum_{i=1}^{n} y_i x_{2i} + \lambda \cos\left(\alpha_\beta - \alpha_{x_0,1} + \frac{\pi}{2}\right) \sin(\alpha_\beta - \alpha_{x_0} + \frac{\pi}{2}) \quad &= 0, \\ - \cos(\alpha_\beta) \sum_{i=1}^{n} y_i x_{1i} - \sin(\alpha_\beta) \sum_{i=1}^{n} y_i x_{2i} + r_\beta &= 0, \end{aligned}$$

which yields when solving for α_β, the estimated angle

$$\tan 2\hat{\alpha}_{\beta,x_0}(\lambda) = \frac{2\sum_{i=1}^n y_i x_{1i} \sum_{i=1}^n y_i x_{2i} - \lambda \sin 2\alpha_{x_0}}{\left(\sum_{i=1}^n y_i x_{1i}\right)^2 - \left(\sum_{i=1}^n y_i x_{2i}\right)^2 - \lambda \cos 2\alpha_{x_0}}.$$

As

$$\sin 2\tilde{\alpha}_\beta = \frac{2\sum y_i x_{1i} \sum y_i x_{2i}}{\left(\sum y_i x_{1i}\right)^2 + \left(\sum y_i x_{2i}\right)^2}, \quad \cos 2\tilde{\alpha}_\beta = \frac{\left(\sum y_i x_{1i}\right)^2 - \left(\sum y_i x_{2i}\right)^2}{\left(\sum y_i x_{1i}\right)^2 + \left(\sum y_i x_{2i}\right)^2},$$

the regression coefficient angle can be expressed as

$$\tan 2\hat{\alpha}_{\beta,x_0}(\lambda) = \frac{\tilde{r}_\beta^2 \sin 2\tilde{\alpha}_\beta - \lambda \sin 2\alpha_{x_0}}{r_{\tilde{\beta}}^2 \cos 2\tilde{\alpha}_\beta - \lambda \cos 2\alpha_{x_0}} = \frac{\tilde{r}_\beta^2 \sin 2\tilde{\alpha}_\beta + \lambda \sin 2\left(\alpha_{x_0} \pm \frac{\pi}{2}\right)}{\tilde{r}_\beta^2 \cos 2\tilde{\alpha}_\beta + \lambda \cos 2\left(\alpha_{x_0} \pm \frac{\pi}{2}\right)},$$

whereas the regression coefficient length is given by

$$\hat{r}_{\beta,x_0}(\lambda) = \cos\hat{\alpha}_{\beta,x_0}(\lambda)\sum_{i=1}^n y_i x_{1i} + \sin\hat{\alpha}_{\beta,x_0}(\lambda)\sum_{i=1}^n y_i x_{2i} = \tilde{r}_\beta(\cos\tilde{\alpha}_\beta\cos\hat{\alpha}_{\beta,x_0}(\lambda) + \sin\tilde{\alpha}_\beta\sin\hat{\alpha}_{\beta,x_0}(\lambda)).$$

A.1 Proof of Lemma 1 and Lemma 2

Proof of Lemma 1. Suppose X is an $n \times p$ matrix of full rank. The gradient of the penalized residual sum-of-squares (penRSS) in Equation (1) is given by

$$\frac{\partial \text{penRSS}}{\partial \beta} = -2X^TY + 2X^TX\beta + 2\frac{\lambda}{\|x_0\|^2}\frac{x_0x_0^T}{\beta^T\beta}\beta - 2\frac{\lambda}{\|x_0\|^2}\frac{(x_0^T\beta)^2}{(\beta^T\beta)^2}\beta. \tag{10}$$

We suppress in the following the subscripts of x_0 and β and the dependence on λ for notational convenience. By setting the gradient to 0 and multiplying by β^T from the left

$$0 = -\hat{\beta}^TX^TY + \hat{\beta}^TX^TX\hat{\beta} + \frac{\lambda}{\|x_0\|^2}\frac{(x_0^T\hat{\beta})^2}{\hat{\beta}^T\hat{\beta}} - \frac{\lambda}{\|x_0\|^2}\frac{(x_0^T\hat{\beta})^2}{(\hat{\beta}^T\hat{\beta})^2}\hat{\beta}^T\hat{\beta}.$$

the last terms cancel. Using the hyperspherical parametrization of Equation (2), $\hat{\beta} = \hat{r}\hat{\gamma}$, the length of the PAN estimator is given by

$$\begin{aligned} \hat{\beta}^TX^TX\hat{\beta} &= \hat{\beta}^TX^TY, \\ \hat{r}^2\hat{\gamma}^TX^TX\hat{\gamma} &= \hat{r}\hat{\gamma}^TX^TX\tilde{\beta}, \\ \hat{r} &= \frac{\hat{\gamma}^TX^TX\tilde{\beta}}{\hat{\gamma}^TX^TX\hat{\gamma}}, \end{aligned} \tag{11}$$

where $\tilde{\beta} = (X^TX)^{-1}X^TY$ is the OLS estimator.

Again, we set the gradient in (10) to 0, rearrange terms and multiply by r to get

$$X^T X\hat{\beta} - \frac{\lambda}{\|x_0\|^2}\frac{(x_0^T\hat{\beta})^2}{(\hat{\beta}^T\hat{\beta})^2}\hat{\beta} = X^T Y - \frac{\lambda}{\|x_0\|^2}\frac{x_0 x_0^T}{\hat{\beta}^T\hat{\beta}}\hat{\beta},$$
$$\hat{r}^2 X^T X\hat{\gamma} - \frac{\lambda}{\|x_0\|^2}\hat{r}^3\frac{\hat{\gamma}^T x_0 x_0^T\hat{\gamma}}{\hat{r}^2}\hat{\gamma} = \hat{r}X^T Y - \frac{\lambda}{\|x_0\|^2}\hat{r}^2\frac{x_0 x_0^T}{\hat{r}^2}\hat{\gamma},$$
$$\hat{r}^2 X^T X\hat{\gamma} - \lambda\hat{r}(\hat{\gamma}^T B\hat{\gamma})\hat{\gamma} = \hat{r}X^T Y - \lambda B\hat{\gamma},$$

defining the matrix $B = x_0 x_0^T/\|x_0\|^2$. By inserting the expression of $\hat{r}$ in (11) and defining the matrix $A = X^T X\tilde{\beta}\tilde{\beta}^T X^T X$, the direction vector of the PAN estimator has to fulfill the equation

$$\frac{\hat{\gamma}^T A\hat{\gamma}}{(\hat{\gamma}^T X^T X\hat{\gamma})^2}X^T X\hat{\gamma} - \lambda(\hat{\gamma}^T B\hat{\gamma})\hat{\gamma} = \frac{1}{\hat{\gamma}^T X^T X\hat{\gamma}}A\hat{\gamma} - \lambda B\hat{\gamma},$$

where $\hat{\gamma}^T A\hat{\gamma}$, $\hat{\gamma}^T B\hat{\gamma}$ and $\hat{\gamma}^T X^T X\hat{\gamma}$ are scalar quadratic forms. □

Proof of Lemma 2. Assume an orthonormal design matrix $X^T X = I_p$. Then the denominator of Equation (11) equals one, $\hat{\gamma}^T\hat{\gamma} = 1$, as the direction vector is normalized, and the PAN length is therefore given by

$$\hat{r} = \tilde{\beta}^T\hat{\gamma}.$$

Thus the estimated PAN direction vector has to fulfill the simplified equation

$$(\hat{\gamma}^T M_0\hat{\gamma})\,\hat{\gamma} = M_0\hat{\gamma},$$

where the matrix M_0 is given by

$$M_0 := \tilde{\beta}\tilde{\beta}^T - \lambda x_0 x_0^T/\|x_0\|^2.$$

As the quadratic form $\hat{\gamma}^T M_0\hat{\gamma}$ is a scalar, the direction vector $\hat{\gamma}$ must be equal to a normalized eigenvector of M_0.

For linearly independent $\tilde{\beta}$ and x_0, and $\lambda \neq 0$, the rank of M_0 is 2. The range of M_0 is spanned by the orthonormal vectors

$$u_1 = \frac{\tilde{\beta}}{\|\tilde{\beta}\|}, \quad u_2 = \frac{\|\tilde{\beta}\|^2 x_0 - (x_0^T\tilde{\beta})\tilde{\beta}}{\|\tilde{\beta}\|\sqrt{\|\tilde{\beta}\|^2\|x_0\|^2 - (x_0^T\tilde{\beta})^2}}. \tag{12}$$

Hence the normalized eigenvectors of M_0 are equal to $(u_1, u_2)\eta$ where η are the normalized eigenvectors of the 2×2 matrix, $\tilde{M}_0$, for any p:

$$\tilde{M}_0 = \begin{bmatrix} \|\tilde{\beta}\|^2 - \lambda\frac{(x_0^T\tilde{\beta})^2}{\|\tilde{\beta}\|^2\|x_0\|^2} & -\lambda\frac{x_0^T\tilde{\beta}}{\|x_0\|^2\|\tilde{\beta}\|}\sqrt{\|x_0\|^2 - \frac{(x_0^T\tilde{\beta})^2}{\|\tilde{\beta}\|^2}} \\ -\lambda\frac{x_0^T\tilde{\beta}}{\|\tilde{\beta}\|\|x_0\|^2}\sqrt{\|x_0\|^2 - \frac{(x_0^T\tilde{\beta})^2}{\|\tilde{\beta}\|^2}} & -\frac{\lambda}{\|x_0\|^2}\left(\|x_0\|^2 - \frac{(x_0^T\tilde{\beta})^2}{\|\tilde{\beta}\|^2}\right) \end{bmatrix}.$$

The two eigenvectors with positive and negative sign give four stationary points for the penalized RSS in Equation (1). For the choice of basis in Equation (12), the global minimum is given by the first eigenvector of $\tilde{M}_0$ with a positive first entry. For a matrix, $\begin{bmatrix} a & -c \\ -c & b \end{bmatrix}, c > 0$, this eigenvector is given as

$$\eta_1 = \left[\left(\frac{1}{2} + \frac{a-b}{2\sqrt{(a-b)^2+4c^2}} \right)^{\frac{1}{2}}, - \left(\frac{1}{2} - \frac{a-b}{2\sqrt{(a-b)^2+4c^2}} \right)^{\frac{1}{2}} \right]^T,$$

such that the direction vector is

$$\hat{\gamma}(\lambda) = \left(\frac{1}{2} + \frac{1}{2}\, c(\lambda) \right)^{\frac{1}{2}} u_1 - \left(\frac{1}{2} - \frac{1}{2}\, c(\lambda) \right)^{\frac{1}{2}} u_2, \quad c(\lambda) = \frac{\|\tilde{\beta}\|^2(\|\tilde{\beta}\|^2+\lambda) - 2\lambda(x_0^T\tilde{\beta})^2/\|x_0\|^2}{\|\tilde{\beta}\|^2\sqrt{(\|\tilde{\beta}\|^2+\lambda)^2 - 4\lambda(x_0^T\tilde{\beta})^2/\|x_0\|^2}}.$$

As the vector u_2 is orthogonal to $\tilde{\beta}$, the length of the PAN estimator is

$$\hat{r}(\lambda) = \tilde{\beta}^T\hat{\gamma}(\lambda) = \left(\frac{1}{2} + \frac{1}{2}\, c(\lambda) \right)^{\frac{1}{2}} \|\tilde{\beta}\|.$$

□

A.2 Proof of Corollary 1

Proof of Corollary 1. The prediction for x_0 is given by

$$\hat{y}_0 = x_0^T\hat{\beta}_{x_0}(\lambda) = \hat{r}(\lambda)x_0^T\hat{\gamma}(\lambda) = \frac{1}{2}\left[1 + c(\lambda)\right] x_0^T\tilde{\beta} - \frac{1}{2}\left[1 - c^2(\lambda)\right]^{\frac{1}{2}} \sqrt{\|\tilde{\beta}\|^2\|x_0\|^2 - (x_0^T\tilde{\beta})^2},$$

where the last term simplifies to

$$\frac{1}{2}\left[1 - c^2(\lambda)\right]^{\frac{1}{2}} \sqrt{\|\tilde{\beta}\|^2\|x_0\|^2 - (x_0^T\tilde{\beta})^2} = x_0^T\tilde{\beta}\frac{\lambda(\|\tilde{\beta}\|^2 - (x_0^T\tilde{\beta})^2/\|x_0\|^2)}{\|\tilde{\beta}\|^2\sqrt{(\|\tilde{\beta}\|^2+\lambda)^2 - 4\lambda(x_0^T\tilde{\beta})^2/\|x_0\|^2}}.$$

Hence

$$\begin{aligned} x_0^T\hat{\beta}_{x_0}(\lambda) =& x_0^T\tilde{\beta}\left[\frac{1}{2} + \frac{1}{2}\frac{\|\tilde{\beta}\|^2(\|\tilde{\beta}\|^2+\lambda) - 2\lambda(x_0^T\tilde{\beta})^2/\|x_0\|^2}{\|\tilde{\beta}\|^2\sqrt{(\|\tilde{\beta}\|^2+\lambda)^2 - 4\lambda(x_0^T\tilde{\beta})^2/\|x_0\|^2}} \right] \\ & - \frac{x_0^T\tilde{\beta}\lambda(\|\tilde{\beta}\|^2 - (x_0^T\tilde{\beta})^2/\|x_0\|^2)}{\|\tilde{\beta}\|^2\sqrt{(\|\tilde{\beta}\|^2+\lambda)^2 - 4\lambda(x_0^T\tilde{\beta})^2/\|x_0\|^2}}, \\ =& x_0^T\tilde{\beta}\left[\frac{1}{2} + \frac{1}{2}\frac{\|\tilde{\beta}\|^2 - \lambda}{\sqrt{(\|\tilde{\beta}\|^2+\lambda)^2 - 4\lambda(x_0^T\tilde{\beta})^2/\|x_0\|^2}} \right]. \end{aligned}$$

□

A.3 Proof of Lemma 3 and Theorem 1

Proof of Lemma 3. Assume a scaled orthogonal design, $X^TX = nI_p$, such that the PAN prediction, denoted by $\hat{\mu}_0(\lambda) = x_0^T\hat{\beta}(\lambda)$, is given by

$$\hat{\mu}_0(\lambda) = x_0^T\tilde{\beta}\left[\frac{1}{2} + \frac{1}{2}\frac{\|\tilde{\beta}\|^2 - (\lambda/n)}{\sqrt{(\|\tilde{\beta}\|^2 + (\lambda/n))^2 - 4(\lambda/n)(x_0^T\tilde{\beta})^2/\|x_0\|^2}}\right],$$

with a scaling of the tuning parameter. The derivative of the mean squared error (MSE) of the prediction $\hat{\mu}_0(\lambda)$ is bounded in a neighborhood of 0, such that

$$\frac{\partial\,\mathrm{MSE}(x_0,\beta,\lambda)}{\partial\lambda}\bigg|_{\lambda=0} = E\left[2(\hat{\mu}_0(\lambda) - \mu_0)\frac{\partial\hat{\mu}_0(\lambda)}{\partial\lambda}\bigg|_{\lambda=0} \mid X\right],$$

where the derivative of the prediction is given by

$$\frac{\partial\hat{\mu}_0(\lambda)}{\partial\lambda} = -\frac{x_0^T\tilde{\beta}\left(\|x_0\|^2\|\tilde{\beta}\|^2 - (x_0^T\tilde{\beta})^2\right)\left((\lambda/n) + \|\tilde{\beta}\|^2\right)}{n\|x_0\|^2\left[(\|\tilde{\beta}\|^2 + (\lambda/n))^2 - 4(\lambda/n)(x_0^T\tilde{\beta})^2/\|x_0\|^2\right]^{3/2}}.$$

The derivative of the MSE evaluated at $\lambda = 0$ is given by

$$\frac{\partial\,\mathrm{MSE}(x_0,\beta,\lambda)}{\partial\lambda}\bigg|_{\lambda=0} = -\frac{2}{n}E\left[f(\tilde{\beta})\right],$$

where

$$f(\tilde{\beta}) = \left(x_0^T\tilde{\beta} - x_0^T\beta\right)\frac{x_0^T\tilde{\beta}(\|x_0\|^2\|\tilde{\beta}\|^2 - (x_0^T\tilde{\beta})^2)}{\|x_0\|^2\|\tilde{\beta}\|^4}. \tag{13}$$

Under the scaled orthogonal design, $X^TX = nI_p$, the variance of the OLS estimator is $\mathrm{Var}(\tilde{\beta}) = \frac{\sigma^2}{n}I_p$. Then the OLS estimator converges in distribution as $\sqrt{n}(\tilde{\beta}-\beta) \xrightarrow{d} \mathcal{N}(0,\sigma^2 I_p)$, (see e.g. Sen et al., 2010, Theorem 10.2.2 for further details). The expectation of the second-order Taylor expansion of a function of the estimator will then be given by

$$E\left[f(\tilde{\beta})\right] = f(\beta) + \frac{1}{2}\frac{\sigma^2}{n}\,\mathrm{tr}\left(\mathbf{H}(f(\beta))\right) + O\left(\frac{1}{n^2}\right),$$

where $\mathbf{H}$ is the Hessian. The trace of the Hessian, $\mathrm{tr}\left(\mathbf{H}(f(\beta))\right)$, will further equal the Laplacian of the function evaluated at β. As the Laplacian of (13) evaluated at β is

$$\nabla^2 f(\beta) = -\frac{2\left((\|x_0\|^2\|\beta\|^2 - 4(x_0^T\beta)^2)(\|x_0\|^2\|\beta\|^2 - (x_0^T\beta)^2)\right)}{\|x_0\|^2\|\beta\|^6},$$

the expectation is given by

$$E\left[f(\tilde{\beta})\right] = \frac{\sigma^2}{n}\frac{(\|x_0\|^2\|\beta\|^2 - 4(x_0^T\beta)^2)(\|x_0\|^2\|\beta\|^2 - (x_0^T\beta)^2)}{\|x_0\|^2\|\beta\|^6} + O\left(\frac{1}{n^2}\right).$$

The derivative of the MSE at the value $\lambda = 0$ is hence given by

$$\left.\frac{\partial \,\mathrm{MSE}(x_0,\beta,\lambda)}{\partial\lambda}\right|_{\lambda=0} = -\frac{1}{n^2}C_1\Big(1 - 4\frac{(x_0^T\beta)^2}{\|x_0\|^2\|\beta\|^2}\Big) + O\left(\frac{1}{n^3}\right), \tag{14}$$

where $C_1 = 2\sigma^2\frac{\|x_0\|^2}{\|\beta\|^2}\Big(1 - \frac{(x_0^T\beta)^2}{\|x_0\|^2\|\beta\|^2}\Big) > 0$ is a positive constant for $\sigma > 0$, $x_0, \beta \neq \mathbf{0}$ and $|x_0^T\beta| \neq \|x_0\|\|\beta\|$. □

Proof of Theorem 1. For $\sigma > 0$, $x_0, \beta \neq \mathbf{0}$ and $|x_0^T\beta| \neq \|x_0\|\|\beta\|$, as the constant C_1 in (14) is always positive, the limit of the derivative will satisfy asymptotically

$$\lim_{n\to\infty} n^2 \left.\frac{\partial \,\mathrm{MSE}(x_0,\beta,\lambda)}{\partial\lambda}\right|_{\lambda=0} \begin{cases} < 0, & \text{if } |x_0^T\beta| < \frac{1}{2}\|x_0\|\|\beta\|, \\ > 0, & \text{if } |x_0^T\beta| > \frac{1}{2}\|x_0\|\|\beta\|, \end{cases}$$

□

References

Alowais, M. I. and L.-K. Soon (2012). Credit card fraud detection: Personalized or aggregated model. In *2012 Third FTRA International Conference on Mobile, Ubiquitous, and Intelligent Computing*, pp. 114–119. IEEE.

Cai, T., J. Fan, and T. Jiang (2013). Distributions of angles in random packing on spheres. *Journal of Machine Learning Research 14*(1), 1837–1864.

Carrión, R. E., B. A. Cornblatt, C. Z. Burton, I. F. Tso, A. M. Auther, S. Adelsheim, R. Calkins, C. S. Carter, T. Niendam, and T. G. Sale (2016). Personalized prediction of psychosis: external validation of the NAPLS-2 psychosis risk calculator with the EDIPPP project. *American Journal of Psychiatry 173*(10), 989–996.

Cheng, L., R. E. Alexander, G. T. MacLennan, O. W. Cummings, R. Montironi, A. Lopez-Beltran, H. M. Cramer, D. D. Davidson, and S. Zhang (2012). Molecular pathology of lung cancer: key to personalized medicine. *Modern Pathology 25*(3), 347—369.

Cho, E. (2009). Inner product of random vectors. *International Journal of Pure and Applied Mathematics 56*(2), 217–221.

Claeskens, G. and N. L. Hjort (2003). The focused information criterion. *Journal of the American Statistical Association 98*(464), 900–916.

Claeskens, G. and N. L. Hjort (2008). *Model selection and model averaging*. Cambridge, UK: Cambridge University Press.

Cohl, H. S. (2011). Opposite antipodal fundamental solution of Laplace's equation in Hyperspherical geometry. *Symmetry Integrability and Geometry Methods and Applications 7*, 108–122.

Efron, B. and R. J. Tibshirani (1994). *An introduction to the bootstrap*. Boca Raton, FL: CRC Press.

Fan, J., F. Han, and H. Liufa (2014). Challenges of Big data analysis. *National Science Review 1*(2), 293–314.

Gatto, R. and S. R. Jammalamadaka (2007). The generalized von Mises distribution. *Statistical Methodology 4*(3), 341–353.

Gruber, M. (1998). *Improving efficiency by shrinkage: The James–Stein and ridge regression estimators*. Boca Raton, FL: CRC Press.

Hastie, T. and R. Tibshirani (1993). Varying-coefficient models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 55*(4), 757–779.

Hellton, K. H. and N. L. Hjort (2018). Fridge: Focused fine-tuning of ridge regression for personalized predictions. *Statistics in Medicine 37*(8), 1290–1303.

Hoerl, A. E. and R. W. Kennard (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics 12*(1), 55–67.

Huang, S.-T., Y. Düren, K. H. Hellton, and J. Lederer (2019). Tuning parameter calibration for prediction in personalized medicine. *arXiv:1909.10635*.

Jabbari, F., S. Visweswaran, and G. F. Cooper (2018). Instance-specific Bayesian network structure learning. In *International Conference on Probabilistic Graphical Models*, pp. 169–180. PMLR.

Kosorok, M. R. and E. B. Laber (2019). Precision medicine. *Annual Review of Statistics and Its Application 6*, 263–286.

Lengerich, B., B. Aragam, and E. P. Xing (2019). Learning sample-specific models with low-rank personalized regression. *arXiv:1910.06939*.

Liu, K. and X.-L. Meng (2016). There is individualized treatment. Why not individualized inference? *Annual Review of Statistics and Its Application 3*, 79–111.

Liu, W., Y.-M. Zhang, X. Li, Z. Yu, B. Dai, T. Zhao, and L. Song (2017). Deep hyperspherical learning. In *Advances in Neural Information Processing Systems*, pp. 3950–3960.

Mardia, K. V. (1972). *Statistics of directional data*. New York City, NY: Academic press.

Öhrn, Y. and J. Linderberg (1983). Hyperspherical coordinates in four particle systems. *Molecular Physics 49*(1), 53–64.

Pourahmadi, M. and X. Wang (2015). Distribution of random correlation matrices: Hyperspherical parameterization of the Cholesky factor. *Statistics & Probability Letters 106*, 5–12.

Rafailidis, D., A. Axenopoulos, J. Etzold, S. Manolopoulou, and P. Daras (2014). Content-based tag propagation and tensor factorization for personalized item recommendation based on social tagging. *ACM Transactions on Interactive Intelligent Systems 3*(4), 1–26.

Reber, R., E. A. Canning, and J. M. Harackiewicz (2018). Personalized education to increase interest. *Current directions in psychological science 27*(6), 449–454.

Romesburg, C. (1984). *Cluster analysis for researchers*. Belmont, CA: Lifetime Learning Publications.

Salton, G. and M. J. McGill (1983). *Introduction to modern information retrieval*. New York City, NY: McGraw Hill.

Scealy, J. L. and A. H. Welsh (2011). Regression for compositional data by using distributions defined on the hypersphere. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 73*(3), 351–375.

Sen, P. K., J. M. Singer, and A. C. P. de Lima (2010). *From finite sample to asymptotic methods in statistics*. Cambridge, UK: Cambridge University Press.

Stamey, T. A., J. N. Kabalin, J. E. McNeal, I. M. Johnstone, F. Freiha, E. A. Redwine, and N. Yang (1989). Prostate specific antigen in the diagnosis and treatment of adenocarcinoma

of the prostate. II. Radical prostatectomy treated patients. *Journal of Urology 141*(5), 1076–1083.

Tang, H., S. S. Liao, and S. X. Sun (2013). A prediction framework based on contextual data to support mobile personalized marketing. *Decision Support Systems 56*, 234–246.

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

Van der Laan, M. J. and S. Rose (2011). *Targeted learning: causal inference for observational and experimental data*. Berlin, Germany: Springer.

Visweswaran, S., G. F. Cooper, and M. Chickering (2010). Learning instance-specific predictive models. *Journal of Machine Learning Research 11*(12).

Welsh, A. (1985). An angular approach for linear data. *Biometrika 72*(2), 441–450.

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

Zeevi, D., T. Korem, N. Zmora, D. Israeli, D. Rothschild, A. Weinberger, O. Ben-Yacov, D. Lador, T. Avnit-Sagi, M. Lotan-Pompan, et al. (2015). Personalized nutrition by prediction of glycemic responses. *Cell 163*(5), 1079–1094.

Zhang, G. and D. W. Nebert (2017). Personalized medicine: Genetic risk prediction of drug response. *Pharmacology & Therapeutics 175*, 75–90.

Zou, H. and T. Hastie (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 67*(2), 301–320.