Electronic Journal of Statistics
Vol. 14 (2020) 2155–2197
ISSN: 1935-7524
https://doi.org/10.1214/20-EJS1711

Projective inference in high-dimensional problems: Prediction and feature selection

Juho Piironen, Markus Paasiniemi and Aki Vehtari

Helsinki Institute for Information Technology (HIIT)
Department of Computer Science, Aalto University
e-mail: juho.t.piironen@gmail.com, markus.paasiniemi@helsinki.fi, aki.vehtari@aalto.fi

Abstract: This paper reviews predictive inference and feature selection for generalized linear models with scarce but high-dimensional data. We demonstrate that in many cases one can benefit from a decision theoretically justified two-stage approach: first, construct a possibly non-sparse model that predicts well, and then find a minimal subset of features that characterize the predictions. The model built in the first step is referred to as the *reference model* and the operation during the latter step as predictive *projection*. The key characteristic of this approach is that it finds an excellent tradeoff between sparsity and predictive accuracy, and the gain comes from utilizing all available information including prior and that coming from the left out features. We review several methods that follow this principle and provide novel methodological contributions. We present a new projection technique that unifies two existing techniques and is both accurate and fast to compute. We also propose a way of evaluating the feature selection process using fast leave-one-out cross-validation that allows for easy and intuitive model size selection. Furthermore, we prove a theorem that helps to understand the conditions under which the projective approach could be beneficial. The key ideas are illustrated via several experiments using simulated and real world data.

MSC 2010 subject classifications: Primary 62F15, 62F07, 62J12.
Keywords and phrases: Projection, prediction, feature selection, sparsity, post-selection inference.

Received February 2019.

1. Introduction

Predictive inference and feature selection for generalized linear models (GLMs) in problems with scarce data but high-dimensional feature space—regime known as "small n, large p" [1]—remains a topic of active research. Often, albeit not always, the goals are twofold: the desire is to find a model that predicts unseen data well but utilizes only a small subset of features. This facilitates the interpretation and makes the model more convenient to use at prediction time.

A vast variety of different approaches have been proposed. Frequentist approaches typically formulate an estimator with a penalty that enforces sparsity

[1] Due to this historical naming we stick with these symbols but also use p to denote density functions. We hope this does not confuse the reader.

in the solution (e.g., Breiman, 1995; Tibshirani, 1996; Fan and Li, 2001; Zou and Hastie, 2005; Candes and Tao, 2007). A useful overview has been written by Hastie, Tibshirani and Wainwright (2015). Among Bayesians, the most common approach is to use a sparsifying prior that favors solutions with a small number of active predictors (e.g., George and McCulloch, 1993; Raftery, Madigan and Hoeting, 1997; Ishwaran and Rao, 2005; Johnson and Rossell, 2012; Carvalho, Polson and Scott, 2010). These approaches do not automatically produce truly sparse solutions since there is always nonzero probability for each feature being included in the model, but sparse models can be obtained for instance by removing features with estimated posterior effect below certain threshold (Barbieri and Berger, 2004; Ishwaran and Rao, 2005; Narisetty and He, 2014).

All these approaches attempt to solve the two problems—prediction and feature selection—simultaneously. In this paper we argue that in many situations one can gain if these problems are solved in two stages:

1. Construct the best predictive model you can (which potentially uses a lot of features). Call this model the *reference model*.
2. If the model is too complex, find a simpler model (with acceptable complexity) that gives as similar predictions as the reference model. For a given complexity (number of features), the model with the smallest predictive discrepancy to the reference model should be selected.

This strategy not only solves many issues that one might encounter in traditionally used Bayesian approaches (as we will discuss in Sec. 2) but has also shown empirically very good performance in comparison to many other methods with good tradeoff between sparsity and predictive accuracy (Piironen and Vehtari, 2017a). Our discussion will be mainly from the Bayesian viewpoint but is aimed to provide insights also for a non-Bayesian oriented reader since the idea of a reference model is not intrinsically limited only to the Bayesian paradigm (see, e.g, Paul et al., 2008; Harrell, 2015) or even to feature selection in generalized linear models (e.g., Bucilă, Caruana and Niculescu-Mizil, 2006; Hinton, Vinyals and Dean, 2015).

A piece of pioneering work in this line was carried out by Lindley (1968), who considered prediction in linear regression model when some of the features are unavailable at prediction time. A related but slightly different approach was proposed by Goutis and Robert (1998) and Dupuis and Robert (2003) who introduced the concept of *projecting* the posterior information in the reference model to smaller submodels, although they were mainly interested in feature selection and less so about predicting with the submodels. Since then, several papers have extended this literature by introducing new variants and computational heuristics Nott and Leng (2010); Tran, Nott and Leng (2012); Hahn and Carvalho (2015). We discuss these contributions in detail later on.

In principle, using a reference model means adopting $\mathcal{M}$-completed view (Bernardo and Smith (1994) and Vehtari and Ojanen (2012)). However, we assume the phenomena we are modeling are so complex that the true model is not included in the list of models under consideration. Thus when constructing the reference model, we adopt the $\mathcal{M}$-open view, but if we are able to find a

model that passes model assessment and checking (see, e.g., Gelman et al., 2013; Gabry et al., 2018), we can use it as a reference model in $\mathcal{M}$-completed setting. The benefit of a reference model is that it reduces the variance in the model selection in a same way as use of model assumptions reduce the uncertainty in the usual data modeling.

1.1. Our contributions

This paper makes the following contributions:

- We give a detailed review about the aforementioned projection techniques under unified notation, illustrate their differences and provide recommendations about the preferred approaches.
- We develop a new type of projection—called clustered projection—that can be considered as a unification of the approach of Goutis, Dupuis and Robert and that of Tran et al., and show that it gives a good balance between speed and accuracy.
- We propose a new efficient method for validating the selection process using approximate leave-one-out (LOO) cross-validation. This technique can be used to assess the predictive accuracy of the submodels which allows for intuitive model size selection.
- We discuss the typical difficulties encountered with the traditional Bayesian feature selection approaches via small examples and show how the projective approach yields more satisfactory results. Since an extensive comparison showing the superiority of the projection (in terms of sparsity-accuracy tradeoff) to many other Bayesian model selection strategies over a variety of data sets has already been carried out earlier (Piironen and Vehtari, 2017a), here we focus only on some of the most commonly used techniques and illustrate via small examples *why* they are problematic.
- We discuss the connection of the projection to the popular Lasso estimator (Tibshirani, 1996) together with several empirical results that demonstrate the benefit of the proposed approach in the "small n, large p" -setting.
- We prove a theorem that—at least in our knowledge—for the first time gives a theoretical argument of why and under which conditions the use of reference model could be beneficial for parameter learning in linear models.
- We provide an R software package `projpred` that implements all the discussed methods. The package is freely available and makes the method easily accessible to a wide audience. [2]

We hope this work will serve as a useful overview of the projective inference and spark further research on an important methodology we feel has largely been overlooked.

[2]The codes with installation instructions and examples are available at https://github.com/stan-dev/projpred.

1.2. Why does a reference model improve feature selection?

We begin with a simple example that motivates why use of a reference model can be useful for feature selection. Although the details are different, this example is greatly inspired by the one presented by Paul et al. (2008).

Assume we have collected n measurements of p features $x_j,\ j = 1, \ldots, p$ along with measurements of some target variable y. Assume also that the data are generated according to the following mechanism:

$$\begin{aligned} f &\sim \mathrm{N}(0, 1), \\ y \,|\, f &\sim \mathrm{N}(f, 1) \\ x_j \,|\, f &\sim \mathrm{N}(\sqrt{\rho} f,\, 1 - \rho), \qquad j = 1, \ldots, p_{\mathrm{rel}}\,, \\ x_j \,|\, f &\sim \mathrm{N}(0, 1), \qquad\quad\ \ j = p_{\mathrm{rel}} + 1, \ldots, p\,. \end{aligned} \tag{1}$$

The target variable values y are noisy observations from the latent function values f which are drawn randomly from a standard Gaussian distribution. The first p_{rel} features x_j are also noisy observations from the latent function f, which makes them correlated and on average equally predictive about y. The multiplier $\sqrt{\rho}$ and the noise variance $1 - \rho$ are chosen so that the marginal variance of each x_j is 1 and the pairwise correlations between the first p_{rel} features are all equal to ρ. The rest of the features are drawn randomly from a standard normal distribution and are thus uncorrelated and irrelevant for predicting y.

Suppose our goal is to assess how predictive each of the features is about the target variable. A simple strategy would be to compute the sample correlation $R(x_j, y)$ between each feature and the target variable and then rank the features based on the absolute values $|R(x_j, y)|$. Since the features are related to the target variable via the latent f, clearly our task would be easier if we had access to the noiseless values f instead of the noisy ones y, since the additional noise weakens the correlations, that is, $|\mathrm{Cor}(x_j, y)| < |\mathrm{Cor}(x_j, f)|$ for $j = 1, \ldots, p_{\mathrm{rel}}$. In practice we do not observe f directly, but intuitively if we could build up a model whose output f_* is fairly close to the true f, we might expect to benefit by making the assessment based on the sample correlations $R(x_j, f_*)$ instead of $R(x_j, y)$.

Figure 1 illustrates this idea. The left graph shows the absolute sample correlations $|R(x_j, y)|$ versus $|R(x_j, f)|$ for one data realization from (1) with $p = 500$, $p_{\mathrm{rel}} = 150$, $n = 30$ and $\rho = 0.5$. The relevant features (red dots) are much better separated from the irrelevant ones (gray dots) when we consider their correlation with f instead of y. The right graph demonstrates that this holds also when we replace the unknown f with predictions f_* of a reference model we can actually compute. Here the reference fit is obtained by Bayesian linear regression of y on the first three supervised principal components of all the features (the procedure is discussed in detail in Sec. 6).

Figure 2 shows that this pattern holds for a wide range of values for ρ and p_{rel}. Parameter ρ describes how strongly the relevant features are predictive about y, so when ρ is close to 1, they all are almost perfect copies of f and therefore easy to distinguish from the noise features. On the other hand when ρ gets smaller,

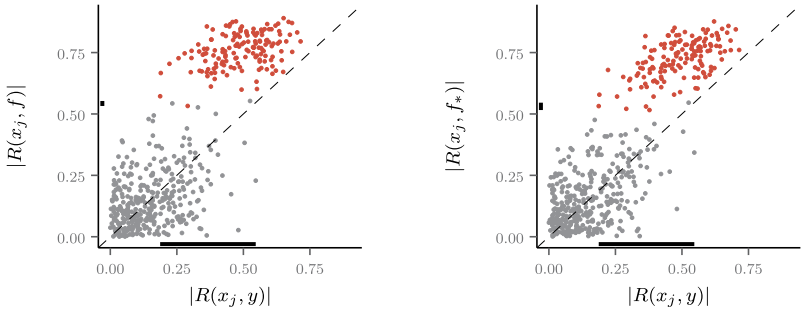

FIG 1. *Introductory example: Left: Absolute sample correlations of each feature x_j with the observed target variable y (horizontal axis) and with the noiseless latent value f (vertical axis) for $n = 30$ observations generated according to (1), with $p = 500$, $p_{rel} = 150$ and $\rho = 0.5$. Red dots denote the truly relevant features and gray dots irrelevant noise features. Right: The same but the true latent f replaced by the predictions f_* of a reference model we can actually compute (see the text for details). The relevant features are much better separated from the irrelevant ones when we consider their correlations with either the true f or the reference model predictions f_* instead of the observed y (the amount of overlap between the two groups is depicted by the black lines).*

the predictive power of the relevant features decreases and hence they are more difficult to identify. It is quite remarkable that above $\rho = 0.4$ the reference model approach gives nearly oracle results.

1.3. Remark on the terminology

To avoid confusion, it is useful to distinguish between two different problems both of which could be considered as "feature selection":

1. Find a *minimal* subset of features that yield a good predictive model for y, so that adding more features does not considerably improve predictive accuracy.
2. Identify *all* features (or as many as possible) that are statistically related to the target variable y.

In the remainder of this paper, we shall focus solely on the first problem, meaning that the central interest is the tradeoff between predictive accuracy and number of features. The latter problem—which is often referred to as *multiple (hypothesis) testing*—is more concerned with controlling metrics such as false discovery rate (FDR), and different means are more suitable for solving this problem. Still, as the previous example illustrates (Sec. 1.2), we expect the reference model approach to be beneficial also there.

2. Traditional Bayesian approaches

This section briefly reviews some of the most common Bayesian approaches for inference with large number of features and highlights their main difficulties.

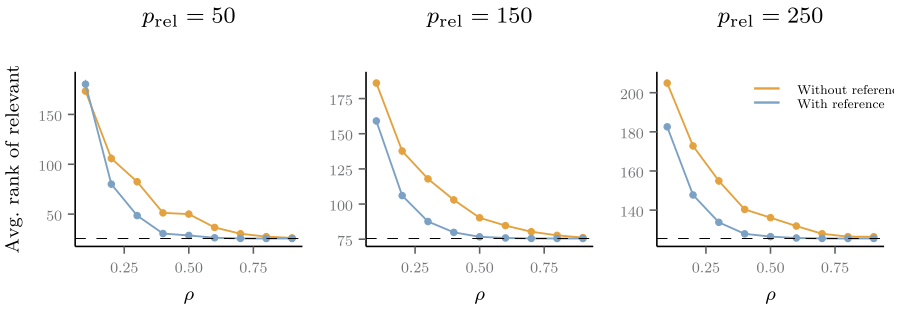

FIG 2. *Introductory example: Average rank of the truly relevant features when the features are sorted based on their absolute sample correlations with y (orange) or with the reference model predictions f_* (blue). The results are averages over 100 data realizations from mechanism* (1)*, with $n = 30$ and $p = 500$, and the results are shown for three different values of p_{rel} with varying ρ. Lower values are better, and the dashed lines denote the oracle results (that is, if all truly relevant features are ranked before the irrelevant ones). The standard errors (vertical lines) are in most cases smaller than the dot sizes.*

2.1. Sparsifying priors

Consider the standard Gaussian linear regression model

$$y_i = \boldsymbol{\beta}^{\mathsf{T}} \mathbf{x}_i + \varepsilon_i, \quad \varepsilon_i \sim \mathrm{N}\big(0, \sigma^2\big), \quad i = 1, \ldots, n\,, \tag{2}$$

where $\mathbf{x}$ is the p-dimensional vector of features, $\boldsymbol{\beta}$ contains the corresponding regression coefficients and σ^2 is the noise variance. A very popular Bayesian approach for assessing the relevances of the different features is to assign a sparsifying prior on each β_j, and then perform the relevance assessment based on the marginal distributions for each β_j (see Sec. 2.2).

A popular prior choice is the *spike-and-slab*, which is often written as a mixture of two Gaussians

$$\begin{gathered} \beta_j \,|\, \lambda_j, c, \varepsilon \sim \lambda_j \,\mathrm{N}\big(0, c^2\big) + (1 - \lambda_j)\,\mathrm{N}\big(0, \varepsilon^2\big), \\ \lambda_j \,|\, \pi \sim \mathrm{Ber}(\pi), \qquad j = 1, \ldots, p, \end{gathered} \tag{3}$$

where $\varepsilon \ll c$ and the indicator variable $\lambda_j \in \{0, 1\}$ denotes whether the coefficient β_j is close to zero (comes from the "spike", $\lambda_j = 0$) or nonzero (comes from the "slab", $\lambda_j = 1$). The width of the spike ε is either taken to be exactly zero or set to a small positive value (George and McCulloch, 1993; Ishwaran and Rao, 2005). The prior inclusion probability π is either fixed (typically to $\pi = 0.5$) or given a hyperprior such as $\pi \sim \mathrm{U}(0, 1)$ (Ishwaran and Rao, 2005).

A popular alternative to the spike-and-slab is to formulate the prior for β_js as a continuous mixture of Gaussians. Several such priors have been proposed (e.g. Carvalho, Polson and Scott, 2010; Armagan, Clyde and Dunson, 2011; Bhattacharya et al., 2015; Bhadra et al., 2017), but the most popular one is probably

the *horseshoe*

$$\begin{aligned} \beta_j \,|\, \lambda_j, \tau &\sim \mathrm{N}\big(0, \tau^2 \lambda_j^2\big), \\ \lambda_j &\sim \mathrm{C}^+(0,1)\,, \quad j = 1, \ldots, p, \end{aligned} \tag{4}$$

which has been shown to possess several attractive properties and has enjoyed a great empirical success (Carvalho, Polson and Scott, 2009; Polson and Scott, 2011; van der Pas, Kleijn and van der Vaart, 2014). The intuition is that the global scale τ drives all the coefficients toward zero, while the thick Cauchy-tails for the local scales λ_j allow some of the coefficients to escape the shrinkage. Piironen and Vehtari (2017b) proposed an extension to the formulation (4), called the *regularized horseshoe*

$$\begin{aligned} \beta_j \,|\, \lambda_j, \tau, c &\sim \mathrm{N}\big(0, \tau^2 \xi_j^2\big), \quad \xi_j^2 = \frac{c^2 \lambda_j^2}{c^2 + \tau^2 \lambda_j^2}, \\ \lambda_j &\sim \mathrm{C}^+(0,1)\,, \quad j = 1, \ldots, p, \\ c^2 &\sim \text{-Inv-}\chi^2\big(\nu, s^2\big), \end{aligned} \tag{5}$$

which introduces an additional regularization parameter c that brings the characteristics of the horseshoe even closer to those of the spike-and-slab (3). The idea is that unlike in the original horseshoe where the largest coefficients are only very weakly penalized (horseshoe has Cauchy-tails), here they face a regularization equivalent to a Student-t slab with scale s and ν degrees of freedom. For a fixed but finite slab width $c = s$ (obtained by letting $\nu \to \infty$), the prior is operationally similar to the spike-and-slab (3) with the same c, whereas the original horseshoe (4) (obtained by letting also $s \to \infty$) resembles the spike-and-slab with infinite slab width $c \to \infty$ (see Piironen and Vehtari, 2017b, for the derivations, more detailed discussion and illustrations). This additional regularization is useful if the parameters are weakly identified (e.g. coefficients in separable logistic regression) and often robustifies and speeds up the Markov chain Monte Carlo (MCMC) posterior inference.

It is possible to place a prior for the global parameter τ based on the sparsity assumptions analogous to the prior for π in spike-and-slab (3). Under certain assumptions, Piironen and Vehtari (2017c,b) showed that to concentrate prior mass onto solutions where p_0 coefficients are far from zero, most of the prior mass for τ should be concentrated near the reference value

$$\tau_0 = \frac{p_0}{p - p_0} \frac{\sigma}{\sqrt{n}}. \tag{6}$$

A recommended weakly informative prior is then $\tau \,|\, \sigma \sim \mathrm{C}^+\big(0, \tau_0^2\big)$, which we shall also use throughout this paper unless otherwise stated.

2.2. Bayes factors and marginal posterior relevance assessment

It should be made explicit that neither the spike-and-slab (3) nor the (regularized) horseshoe (5) performs actual feature *selection* in the sense that some of

the variables would have exactly zero coefficient with probability one, which is true for many of the non-Bayesian penalized estimators (see Sec. 4). Although often overlooked, the actual selection problem can remain highly non-trivial even after successfully fitting the model with a sparsifying prior.

In the spike-and-slab literature, the actual selection is most often carried out either by selecting the most probable feature combination (that is, using Bayes factors) or by selecting those features with posterior inclusion probability above some threshold, typically 0.5, although several thresholding rules have been proposed (Barbieri and Berger, 2004; Ishwaran and Rao, 2005; Narisetty and He, 2014). Analogous decision rules based on the posterior estimates for the so called shrinkage factors could also be devised for the continuous shrinkage priors (Carvalho, Polson and Scott, 2010).

Unfortunately both the Bayes factors and the marginal relevance assessment have difficulties that make them unsatisfactory in our opinion. Firstly, the posterior inference via MCMC for multimodal posterior resulting from one of the sparsifying priors can be a challenge for high-dimensional feature spaces, albeit sophisticated sampling techniques can alleviate this problem (see, e.g., Zanella and Roberts, 2019). Secondly, for large number of features p the Bayes factors typically have high Monte Carlo errors due to the fact that only a vanishingly small proportion of the 2^p models is visited during MCMC, and almost all models are not visited at all. The relevance assessment based on the marginal posteriors on the other hand can produce unintuitive results in the case of correlating features, since it can be that the marginals of two or more coefficients overlap with zero but the joint distribution is clearly distinguished from zero (see Sec. 2.3). Another major issue is that neither of these approaches provides a satisfactory answer to how to perform *post-selection inference* for the selected model, in particular, how to make inference and predictions after the selection, conditional on all the information available. For an example of how the projective approach can improve predictions using the selected model even when marginal posterior probabilities are used for selecting the features, see Figure 6 in Piironen and Vehtari (2017a).

2.3. An illustrative example

We illustrate the difficulties with the marginal relevance assessment discussed in Section 2.2 with similar data as in the introductory example, see Equation (1). We generated one data realization with $n = 50$ observations for three different number of features, $p = 4$, $p = 10$ and $p = 50$, each using $\rho = 0.8$ and $p_{\text{rel}} = \frac{p}{2}$, so in each case the first half of the features were truly relevant. For illustration purposes, we did this by first generating the data for $p = 4$ and then adding the right number of relevant and irrelevant features for cases $p = 10$ and $p = 50$. This way, the realized values for the first two relevant features x_1 and x_2 and the target variable y did not vary between the three data sets, which lets us illustrate how the total number of features p affects the relevance assessment of the two features.

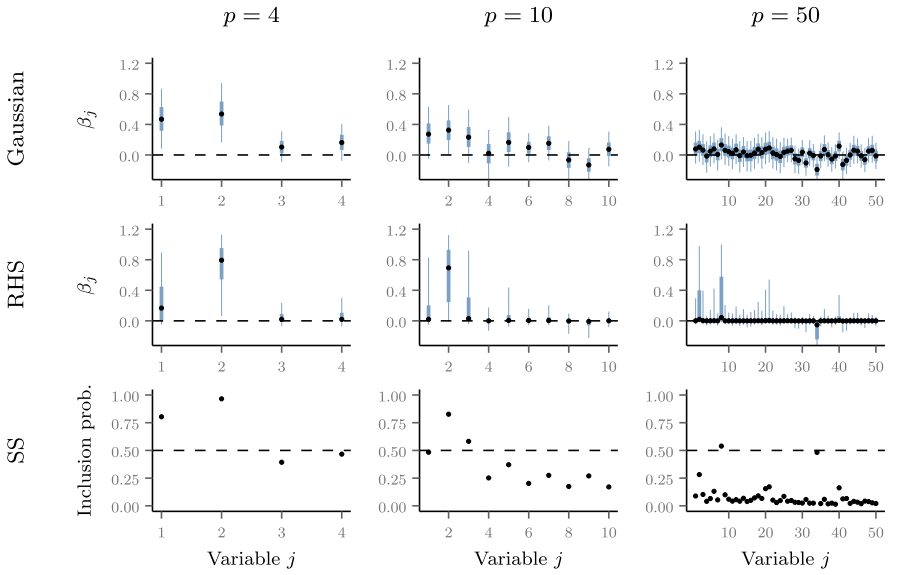

FIG 3. *Simulated example: The rows denote the results for the three different priors, Gaussian, regularized horseshoe (RHS) and spike-and-slab (SS), and the columns show the results for the three different number of features p. For the Gaussian and RHS priors the graphs show the posterior median (dots) with 50% and 90% credible intervals (thick and slim lines, respectively) for the regression coefficients β_j. For SS prior, the graphs show the posterior inclusion probabilities for each variable. As the dimensionality p increases, all the marginals start to overlap with zero, and the SS posterior inclusion probabilities get smaller.*

A Bayesian linear regression model was fitted to these data with three different priors on the regression coefficients:

- Gaussian $\beta_j \,|\, \tau \sim \mathrm{N}\big(0, \tau^2\big)$ with $\tau \sim \mathrm{C}^+(0,1)$
- Regularized horseshoe (RHS) with $p_0 = 1$, $\nu = 4$, $s^2 = 1$ (See Eq. (5) and (6))
- Spike-and-slab (SS)[3] with $\pi \sim \mathrm{U}(0,1)$

Figure 3 visualizes the posterior median and credible intervals for the regression coefficients under Gaussian and RHS priors, along with the marginal posterior inclusion probabilities for the different features obtained from the SS-posterior. With only $p = 4$ features and Gaussian prior, both x_1 and x_2 are detected to be relevant as the marginal posteriors of β_1 and β_2 are distinguished from zero. As the number of features grows, the marginals become more concentrated around zero and with $p = 50$ the marginals of all the relevant features are substantially overlapping with zero. The same applies also for the RHS prior, in fact it appears that the marginals start to concentrate around zero faster than for Gaussian prior. Also for the SS prior, the marginal inclusion probabilities generally decrease for all the relevant features as the dimensionality grows, and

[3]For inference, we used the R-package `spikeslab` (Ishwaran, Kogalur and Rao, 2010).

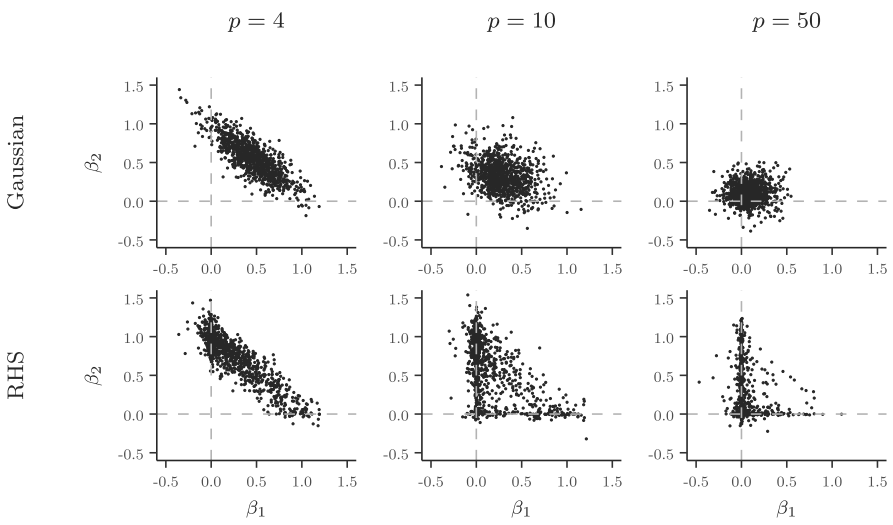

FIG 4. *Simulated example: Posterior draws for β_1 and β_2 with Gaussian and regularized horseshoe (RHS) priors (top and bottom row, respectively) when the total number of features p varies. In each graph, the observed data for x_1, x_2 and y are exactly the same, only the prior and the total number of features p varies. Notice how the marginal posteriors are always more overlapping with zero than the joint posterior. As the dimensionality increases (in particular, when the number of features correlating with x_1 and x_2 increases), the joint posterior becomes more closer to the product of the two marginals and more overlapping with zero.*

for $p = 50$ only one of them just barely has probability over 0.5. Notice how the marginals of the coefficients for the relevant variables are not substantially different from those of the irrelevant ones when $p = 50$ regardless of the prior.

The reason for this behaviour is quite simple: as the number of features carrying similar information grows, the coefficients of most of the relevant features could be set to zero as long as one (or a few) of them obtain nonzero coefficient. In other words, none of the features is so precious that it could not be removed, and therefore the marginals of all the features become more overlapping with zero.

Figure 4 further illustrates what happens to the posterior of β_1 and β_2 when the dimensionality changes. For $p = 4$ where x_1 and x_2 are the only relevant features, the posterior dependency between their coefficients is very strong; if one of the coefficients is set to zero, then the other one must be large. As the number of features p grows, the posterior dependency between β_1 and β_2 becomes weaker; when there are many features that carry similar information as x_1 and x_2, both coefficients could be set to zero because there are many substitutes. The results for $p = 50$ really summarize why the marginals and the pairwise posterior plots can be very challenging to interpret and even misleading: x_1 and x_2 have correlation of $\rho = 0.8$ and their correlation with y both exceed 0.6,[4] yet there is no apparent posterior dependency and both marginals clearly

[4]The correlation between each relevant x_j and y is $\sqrt{\frac{\rho}{2}} \approx 0.63$

overlap zero! Figure 9 in Appendix A.1 confirms that these observations are not due to cherry-picking a specific data set, but hold over multiple data realizations.

2.4. Why not to use cross-validation for selecting the feature combination?

Cross-validation (CV) and information criteria (IC) are widely used generic methods for estimating predictive performance of essentially any learning algorithm. One might be wondering why not to use them also for feature selection? While it is certainly true that for example cross-validation can be a robust and convenient method for comparing a few competing models, in feature selection the number of model comparisons becomes quickly impractically large even for a relatively small number of candidate features. The computational burden of fitting a large number of models becomes an obvious problem especially if Bayesian approach with MCMC is used for inference.

Another problem that is not always so well understood is that when many models are compared using cross-validation, the selection process is liable to overfitting which can lead to selection of non-optimal model due to relatively high variance in the cross-validation estimates. We have discussed this in detail in our earlier work (Piironen and Vehtari, 2017a) where we also show that the projective approach is considerably more resilient to this phenomenon. The selection induced bias has also been discussed by other authors, see for example Ambroise and McLachlan (2002), Reunanen (2003) and Cawley and Talbot (2010).

3. Predictive projection

This section discusses the projective approach in detail. We start by describing the projective idea in general, and then discuss the exponential family models and GLMs as special cases.

3.1. Remarks on notation

We shall denote the training data by $\mathcal{D}$. The 'tilde' notation is used to denote future measurements, for example symbol $\tilde{y}$ denotes unseen measurement for y. To simplify notation, we use $\tilde{y}_i$ to denote a new observation at the ith observed feature values $\mathbf{x}_i$, which allows us to drop the conditioning on $\mathbf{x}_i$ from the conditional distributions. Notice though that $\tilde{y}_i$ is in general different from the observed y_i.

3.2. General idea

In generic terms, *posterior projection* refers to a procedure of replacing the posterior distribution $p(\boldsymbol{\theta}_* \mid \mathcal{D})$ of the reference model with a simpler distribution

$q_\perp(\boldsymbol{\theta})$ that is restricted in some way. For example, in feature selection context for GLMs, this would mean constraining some of the regression coefficients to be exactly zero. In general, the domain of the projected parameters $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ can and typically will be different from the domain of the reference model parameters $\boldsymbol{\theta}_* \in \boldsymbol{\Theta}_*$. For this reason, it is not meaningful to define the projection directly via the discrepancy between $p(\boldsymbol{\theta}_* \,|\, \mathcal{D})$ and $q_\perp(\boldsymbol{\theta})$. Instead, a natural approach would be to define it via the discrepancy between the induced *predictive* distributions

$$\begin{aligned}
\mathrm{KL}(p(\tilde{y} \,|\, \mathcal{D}) \,\|\, q(\tilde{y})) = \;& \mathrm{E}_{\tilde{y}}(\, \log p(\tilde{y} \,|\, \mathcal{D}) - \log q(\tilde{y}) \,) \\
= \;& -\mathrm{E}_{\tilde{y}}(\, \log q(\tilde{y}) \,) + \text{const.} \\
= \;& -\mathrm{E}_{\tilde{y}}(\, \log \mathrm{E}_{\boldsymbol{\theta}}(\, p(\tilde{y} \,|\, \boldsymbol{\theta}) \,) \,) + \text{const.} \\
= \;& -\mathrm{E}_{\boldsymbol{\theta}_*}\big(\, \mathrm{E}_{\tilde{y} \,|\, \boldsymbol{\theta}_*}(\, \log \mathrm{E}_{\boldsymbol{\theta}}(\, p(\tilde{y} \,|\, \boldsymbol{\theta}) \,) \,) \,\big) + \text{const.} \qquad (7)
\end{aligned}$$

Here $\mathrm{E}_{\boldsymbol{\theta}_*}(\,\cdot\,)$, $\mathrm{E}_{\tilde{y} \,|\, \boldsymbol{\theta}_*}(\,\cdot\,)$ and $\mathrm{E}_{\boldsymbol{\theta}}(\,\cdot\,)$ denote expectations over $p(\boldsymbol{\theta}_* \,|\, \mathcal{D})$, $p(\tilde{y} \,|\, \boldsymbol{\theta}_*)$ and $q_\perp(\boldsymbol{\theta})$, respectively. Optimal projection of posterior $p(\boldsymbol{\theta}_* \,|\, \mathcal{D})$ from parameter space $\boldsymbol{\Theta}_*$ to $\boldsymbol{\Theta}$ in terms of minimal predictive loss would then be the distribution $q_\perp(\boldsymbol{\theta})$ that minimizes functional (7). In practice minimizing this is difficult even for relatively simple models and projected posterior $q_\perp(\boldsymbol{\theta})$ due to the many expectations, but expression (7) serves as the ideal when re-formulating the projection in a more tractable way. Below we define three different projections.

3.3. Practical projection techniques

Draw-by-draw Instead of trying to minimize the functional (7) assuming some parametric form for $q_\perp(\boldsymbol{\theta})$, we can obtain an easier optimization problem by formulating the projection as a pointwise mapping from a given $\boldsymbol{\theta}_* \in \boldsymbol{\Theta}_*$ to $\boldsymbol{\theta}_\perp \in \boldsymbol{\Theta}$ as

$$\begin{aligned}
\boldsymbol{\theta}_\perp &= \arg\min_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \mathrm{KL}(p(\tilde{y} \,|\, \boldsymbol{\theta}_*) \,\|\, p(\tilde{y} \,|\, \boldsymbol{\theta})) \\
&= \arg\max_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \mathrm{E}_{\tilde{y} \,|\, \boldsymbol{\theta}_*}(\, \log p(\tilde{y} \,|\, \boldsymbol{\theta}) \,). \qquad (8)
\end{aligned}$$

For models where the predictions are conditioned on some set of observed predictors $\tilde{\mathbf{x}}$, one takes the average of (8) over the distribution of the predictors. As the distribution of the future predictors $p(\tilde{\mathbf{x}})$ is typically not available, the expectations over this are most conveniently approximated by a sample mean over the observed $\{\mathbf{x}_i\}_{i=1}^n$. This results in a projection equation

$$\boldsymbol{\theta}_\perp = \arg\max_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \frac{1}{n} \sum_{i=1}^{n} \mathrm{E}_{\tilde{y}_i \,|\, \boldsymbol{\theta}_*}(\, \log p(\tilde{y}_i \,|\, \boldsymbol{\theta}) \,), \qquad (9)$$

which is the original formulation of (Goutis and Robert, 1998; Dupuis and Robert, 2003) (they used minimization of KL-divergence in their formulation, but this is equivalent to maximizing the expected likelihood in Eq. (9)). Given S draws $\{\boldsymbol{\theta}_*^s\}_{s=1}^S$ from the posterior $p(\boldsymbol{\theta}_* \,|\, \mathcal{D})$ we can project each of these separately via (9) to obtain the corresponding draws $\{\boldsymbol{\theta}_\perp^s\}_{s=1}^S$ in the projection

space $\boldsymbol{\Theta}$. These can be thought of as draws from a projected posterior distribution $q_\perp(\boldsymbol{\theta})$ (although this may not be available analytically), and hence they are used exactly as we would use posterior draws for that particular submodel. The appealing property of the draw-by-draw projection is that it is computationally feasible for many commonly used models such as the GLMs because the optimization problem will have the same form as the problem of finding the maximum likelihood parameter values (see Sec. 3.5). The introduced projection error or loss is then defined as the average loss over the draws

$$\delta_{\boldsymbol{\Theta}} = \frac{1}{S}\sum_{s=1}^{S} \mathrm{KL}(p(\tilde{y}\,|\,\boldsymbol{\theta}_*^s)\,\|\,p(\tilde{y}\,|\,\boldsymbol{\theta}_\perp^s)). \tag{10}$$

Single point (one cluster) Draw-by-draw projection (above) maps each parameter value $\boldsymbol{\theta}_*$ into a corresponding value $\boldsymbol{\theta}_\perp$ in the projection space. The single point projection (which is a special case of the clustered projection that we will introduce in a moment) instead maps the whole posterior $p(\boldsymbol{\theta}_*\,|\,\mathcal{D})$ into a single value $\boldsymbol{\theta}_\perp$. This can be obtained from (7) by assuming $q_\perp(\boldsymbol{\theta})$ is a point mass at $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, taking expectation over the predictors $\tilde{\mathbf{x}}$ and then optimizing the expression with respect to $\boldsymbol{\theta}$

$$\boldsymbol{\theta}_\perp = \arg\max_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \frac{1}{n}\sum_{i=1}^{n} \mathrm{E}_{\tilde{y}_i}(\,\log p(\tilde{y}_i\,|\,\boldsymbol{\theta})\,). \tag{11}$$

This is the formulation of Tran, Nott and Leng (2012). Notice that (11) is otherwise same as (9) except that here the expectation is computed over the posterior predictive distribution of the reference model, that is, $\mathrm{E}_{\tilde{y}_i}(\,\cdot\,) = \mathrm{E}_{\boldsymbol{\theta}_*}\big(\mathrm{E}_{\tilde{y}_i\,|\,\boldsymbol{\theta}_*}(\,\cdot\,)\big)$, where $\mathrm{E}_{\boldsymbol{\theta}_*}(\,\cdot\,)$ denotes expectation over $p(\boldsymbol{\theta}_*\,|\,\mathcal{D})$. In practice the expectation $\mathrm{E}_{\tilde{y}_i}(\,\cdot\,)$ is approximated using the posterior draws. Equation (11) can be used to compute optimal point estimates in the projection space. Also, when $\boldsymbol{\Theta} = \boldsymbol{\Theta}_*$ this computes the optimal predictive point estimates in the original parameter space (for a related approach, see Bernardo and Juárez, 2003). It is worth noticing that in general the result is often different from the usual point estimates, such as the posterior mean or median.

The benefit of the single point projection over the draw-by-draw is that it is much lighter computationally. For instance, for GLMs (Sec. 3.5), solving (11) has the same computational complexity as solving (9), and since the latter must be solved separately for each of the S posterior draws, single point projection essentially reduces the computations by a factor of S. Another benefit of formulation (11) is that it allows convenient search techniques, such as the Lasso type L_1-penalty, to be used for finding good submodels (see Section 4). The drawback is that it can be somewhat less accurate than the one-to-one projection, meaning that the predictive accuracy of the submodel can be compromised. To address this point, we shall introduce the clustered projection below.

Clustered The clustered projection is our novel approach that can be thought of as a unification of the draw-by-draw and single point projections. In

this approach one clusters the posterior draws $\{\boldsymbol{\theta}_*^s\}_{s=1}^S$ of the reference model into C clusters $\{\boldsymbol{\theta}_*^s : s \in I_c\}$, $c = 1, \ldots, C$, and then performs a single point projection within each cluster. Here $I_1, \ldots, I_C$ denote the index sets that indicate which draw belongs to which cluster (we discuss in a moment how to come up with such a division). The projection for the cth cluster then becomes

$$\boldsymbol{\theta}_\perp = \arg\max_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \frac{1}{n} \sum_{i=1}^{n} \mathrm{E}_{\tilde{y}_i \mid I_c}(\log p(\tilde{y}_i \mid \boldsymbol{\theta})), \tag{12}$$

where $\mathrm{E}_{\tilde{y}_i \mid I_c}(\cdot)$ denotes the predictive distribution of the reference model computed over the posterior draws in that cluster I_c. In other words, $\mathrm{E}_{\tilde{y}_i \mid I_c}(h(\tilde{y}_i)) = \frac{1}{|I_c|} \sum_{s \in I_c} \mathrm{E}_{\tilde{y}_i \mid \boldsymbol{\theta}_*^s}(h(\tilde{y}_i))$ for any function $h(\tilde{y}_i)$.[5] Solving (12) for each or the C clusters yields a set of projected parameters $\{\boldsymbol{\theta}_\perp^c\}_{c=1}^C$. Each of these is given a weight ω_c proportional to the number of draws in that cluster, $\omega_c = \frac{|I_c|}{S}$, and these weights are taken into account when computing expectations over the projected posterior. For example, the projected predictive density at future $\tilde{y}$ is then given by

$$q(\tilde{y}) = \sum_{c=1}^{C} \omega_c \, p(\tilde{y} \mid \boldsymbol{\theta}_\perp^c). \tag{13}$$

More generally, the expectation of an arbitrary function $h(\boldsymbol{\theta}_\perp)$ over the projected posterior is calculated as $\sum_{c=1}^C \omega_c h(\boldsymbol{\theta}_\perp^c)$.

A simple but generic and effective approach is to cluster the draws $\{\boldsymbol{\theta}_*^s\}_{s=1}^S$ based on the expected values they impose for y in the unconstrained (latent) space. That is, if $\mathbf{f}_s = g(\mathrm{E}(\tilde{\mathbf{y}} \mid \boldsymbol{\theta}_*^s))$, where $\tilde{\mathbf{y}} = (\tilde{y}_1, \ldots, \tilde{y}_n)$ and $g(\cdot)$ denotes the link function, we would cluster the vectors $\{\mathbf{f}_s\}_{s=1}^S$. This approach is convenient since it makes the clustering independent of the dimensionality of the parameter space of the reference model, and since in practice for projection we need only the vectors $\mathbf{f}_s$ (see Sec. 3.4 and 3.5), we can perform the clustering with access only to the predictions of the reference model (without access to the actual parameter values). As a clustering algorithm, we use k-means. Although k-means is known to have some limitations, in our experience it usually performs reasonably well. An alternative approach would be to minimize the locations of the projected parameters $\{\boldsymbol{\theta}_\perp^c\}_{c=1}^C$ jointly using for example the method of Snelson and Ghahramani (2005), but this is computationally much more expensive.

Both the draw-by-draw (9) and the single point projection (11) are obtained as special cases of the clustered projection (12). The draw-by-draw approach is obtained by setting the number of clusters C equal to the number of posterior draws $C = S$ and assigning each posterior draw into its own cluster. The single point projection is obtained by setting $C = 1$ and assigning all draws into the same cluster. The benefit of the clustered projection is that it improves the accuracy compared to the single point (one cluster) projection already with a

[5] Here we are slightly abusing the notation by using the symbol $\mathrm{E}(\cdot)$ to denote sample mean computed over a finite number of posterior draws, but we do this to simplify the notation.

small number of clusters, and thereby gives a good tradeoff between speed and accuracy. We will illustrate this with an example in Sec. 7.1.

3.4. Exponential family models

Assuming the observation model for y_i belongs to the exponential family with canonical parameter η_i and dispersion ϕ, the log-likelihood has the form (McCullagh and Nelder, 1989, ch. 2)

$$\mathcal{L}_i = \log p(y_i \,|\, \eta_i) = \frac{y_i \eta_i - B(\eta_i)}{A(\phi)} + H(y_i, \phi), \tag{14}$$

for some specific functions $A(\cdot)$, $B(\cdot)$ and $H(\cdot)$. Here the natural parameter is a function of the model parameters, $\eta_i = \eta_i(\boldsymbol{\theta})$. The maximum likelihood solution for the parameters $\boldsymbol{\theta}$ reduces to

$$\boldsymbol{\theta}_{\mathrm{ML}} = \arg\max_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \sum_{i=1}^{n} \big(y_i \eta_i(\boldsymbol{\theta}) - B(\eta_i(\boldsymbol{\theta})) \big), \tag{15}$$

which does not depend on the value for the dispersion ϕ (function $A(\phi)$ is assumed to be strictly positive). Let $\tilde{y}_i$ denote a new measurement at the ith observed feature values $\mathbf{x}_i$. Now, if we denote the expected value of $\tilde{y}_i$ over some reference distribution as $\mu_i^* = \mathrm{E}(\tilde{y}_i)$, we can write the draw-by-draw, single point and clustered projections (Eq. (9), (11) and (12)) all as

$$\boldsymbol{\theta}_{\perp} = \arg\max_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \sum_{i=1}^{n} \big(\mu_i^* \eta_i(\boldsymbol{\theta}) - B(\eta_i(\boldsymbol{\theta})) \big). \tag{16}$$

Thus when the observation model of the submodel is in the exponential family, the projection of the model parameters $\boldsymbol{\theta}$ is equivalent to finding the maximum likelihood solution with the observed targets $\mathbf{y} = (y_1, \ldots, y_n)$ replaced by their expected values $\boldsymbol{\mu}_* = (\mu_1^*, \ldots, \mu_n^*)$ as predicted by the reference model. Thus the projection can be considered as "fitting to the fit" of the reference model. As discussed in Section 3.3, in draw-by-draw projection these fitted values μ_i^* are computed separately for each posterior draw in the reference model, in clustered projection separately for each cluster, and ultimately in the one cluster (single point) projection over the whole posterior with the parameters $\boldsymbol{\theta}_*$ integrated out. Notice also that the projection of the parameters $\boldsymbol{\theta}$ does not depend on the value for the dispersion parameter ϕ.

It is worth emphasizing that this result assumes only that the observation model of the reduced model belongs to exponential family. In particular, we are not making any assumptions about the observation model of the reference model (which need not belong to the exponential family) or about the functional form of $\eta(\boldsymbol{\theta})$ or about how the reference fit $\boldsymbol{\mu}_*$ is formed. In principle this means that the projection could be applied to a wide class of learning algorithms simply by plugging in the fit of the reference model in place of the observed targets y_i

in maximum likelihood estimation. In practice, though, this does not work for nonparametric models such as Gaussian processes where the parameters are the values η_i themselves without further assumptions.

After computing the projected values for the model parameters $\boldsymbol{\theta}$ (Eq. (16)), the dispersion ϕ is computed from

$$\phi_\perp = \arg\max_{\phi} \sum_{i=1}^{n} \left(\frac{r_i(\boldsymbol{\theta}_\perp)}{A(\phi)} + \mathrm{E}_{\tilde{y}_i}(H(\tilde{y}_i, \phi)) \right), \tag{17}$$

where $r_i(\boldsymbol{\theta}_\perp) = \mu_i^* \eta_i(\boldsymbol{\theta}_\perp) - B(\eta_i(\boldsymbol{\theta}_\perp))$ does not depend on ϕ. Again, in draw-by-draw and clustered projection, the expectation in Equation (17) is computed separately for each draw or cluster, and in single point projection by integrating over the whole posterior.

3.5. Generalized linear models

GLMs have their observation model in the exponential family and thus the discussion of Section 3.4 applies. Let us first consider the projection onto a linear Gaussian model with feature matrix $\mathbf{X}$, where the parameters are the regression coefficients $\boldsymbol{\beta}$ and dispersion is the noise variance σ^2. For simplicity, let us now assume also that the reference model is a linear Gaussian model with feature matrix $\mathbf{Z}$ and parameters $(\boldsymbol{\beta}_*, \sigma_*^2)$ and that we have drawn a posterior sample $\{\boldsymbol{\beta}_*^s, \sigma_{*,s}^2\}_{s=1}^S$. Consider now the clustered projection with C clusters. As discussed in Section 3.4, the projection solution for $\boldsymbol{\beta}$ within each cluster is obtained by plugging in the fit of the reference model in place of $\mathbf{y}$ into the familiar maximum likelihood solution

$$\boldsymbol{\beta}_c = (\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}\boldsymbol{\mu}_*^c, \tag{18}$$

where $\boldsymbol{\mu}_*^c = \frac{1}{|I_c|}\sum_{s\in I_c} \mathbf{Z}\boldsymbol{\beta}_*^s$ denotes the prediction within the cth cluster. In the single point projection ($C = 1$) this reduces to $\boldsymbol{\mu}_* = \frac{1}{S}\sum_{s=1}^S \mathbf{Z}\boldsymbol{\beta}_*^s$, whereas in the draw-by-draw ($C = S$) we have $\boldsymbol{\mu}_*^s = \mathbf{Z}\boldsymbol{\beta}_*^s$.

After plugging (18) into (17), it is straightforward to show that the projection of the noise variance becomes

$$\sigma_c^2 = \frac{1}{n}\sum_{i=1}^{n} V_i^c + \frac{1}{n}||\mathbf{X}\boldsymbol{\beta}_c - \boldsymbol{\mu}_*^c||^2, \tag{19}$$

where V_i^c denotes the predictive variance of $\tilde{y}_i$ in the reference model within the cth cluster. This is given by

$$\begin{aligned} V_i^c = \mathrm{Var}(\tilde{y}_i \,|\, I_c) &= \mathrm{E}\big(\mathrm{Var}\big(\tilde{y}_i \,|\, \boldsymbol{\beta}_*, \sigma_*^2\big) \,|\, I_c \big) + \mathrm{Var}\big(\mathrm{E}\big(\tilde{y}_i \,|\, \boldsymbol{\beta}_*, \sigma_*^2 \big) \,|\, I_c\big) \\ &= \mathrm{E}\big(\sigma_*^2 \,|\, I_c \big) + \mathrm{Var}\big(\mathbf{z}_i^\mathsf{T}\boldsymbol{\beta}_*^s \,|\, I_c\big) \\ &= \frac{1}{|I_c|}\sum_{s\in I_c} \sigma_{*,s}^2 + \mathrm{V}_{s\in I_c}\big[\mathbf{z}_i^\mathsf{T}\boldsymbol{\beta}_*^s\big], \end{aligned} \tag{20}$$

where $\mathrm{V}_{s \in I_c}[\cdot]$ denotes sample variance over indices $s \in I_c$. Result (19) has a natural interpretation; the projected noise variance is the average predictive variance of the reference model plus the mismatch between the projected and the reference model. Therefore any systematic variation in the data captured by the reference model but not by the reduced model will be added to the unstructured noise term in the reduced model. Notice also that the predictive uncertainty of the projected model can never be smaller than in the reference model which shows why the projection provides guard against overfitting in the submodels.

Above we assumed that also the reference model is linear with Gaussian noise. As already pointed out in Section 3.4, we emphasize that Equations (18) and (19) hold even without these assumptions. For instance, $\boldsymbol{\mu}_*^c$ could come from an arbitrary model, such as Gaussian process (GP), neural network or some complex simulation model, and in the projection we investigate how much accuracy is sacrificed by replacing it with a linear model. Even when the reference model does not account for uncertainty in $\boldsymbol{\mu}_*$, that is, when no clustering can be made, the single point projection is always available for the reference fit $\boldsymbol{\mu}_*$. Also, the reference model noise could be non-Gaussian—Student-t, for instance—but we could still project this model onto a Gaussian noise.

When the observation model of the projected model is non-Gaussian or when the link is non-identity, the maximum likelihood solution is not available analytically, and therefore no closed form solutions for the projected regression coefficients or dispersion parameters exist. For solving the regression coefficients, the standard approach then is to use iteratively reweighted least squares algorithm (IRLS), where each of the log-likelihood terms $\mathcal{L}_i$ is replaced by a pseudo Gaussian observation whose mean and variance are determined either by second order Taylor series expansion to $\mathcal{L}_i$ (e.g. Gelman et al., 2013, ch. 16.2) or by linear approximation to the link function (McCullagh and Nelder, 1989, ch. 2.5) at the current iterate (with canonical link functions the two approaches are equivalent). The process is then iterated until convergence. Given the solution to the regression coefficients, one can then plug that into Equation (17) and solve the corresponding value for the dispersion (which might also require an iterative procedure).

4. Search strategies

Due to the combinatorial explosion, even for relatively small number of features it is infeasible to go through all the combinations when finding the optimal reduced model for a given number of features. Therefore one has to rely on approximate search heuristics for exploring promising submodels. Probably the simplest alternative is to use a forward stepwise excursion. This procedure starts from the model with only the intercept term and sequentially adds the feature that decreases the projection error the most. Forward search can be used together with any of the three projection techniques presented in Section 3.3 and often works well, but it can be computationally expensive for large number of features.

In the case of single point projection (11), a viable alternative is to use either a Lasso-type L_1-penalization (Tibshirani, 1996) or the more general elastic net penalty (Zou and Hastie, 2005) which contains L_1-penalty as a special case. The single point projection for GLMs with elastic net penalty can be written as

$$\min_{\boldsymbol{\beta}} \left\{ -\frac{1}{n} \sum_{i=1}^{n} \mathrm{E}_{\tilde{y}_i} \left(\mathcal{L}_i(\boldsymbol{\beta}, \tilde{y}_i) \right) + \lambda \left(\frac{1}{2}(1-\alpha)||\boldsymbol{\beta}||_2^2 + \alpha||\boldsymbol{\beta}||_1 \right) \right\}. \tag{21}$$

Here the first term is the expectation of the negative of the expected log-likelihood of the submodel with coefficient vector $\boldsymbol{\beta}$ over the predictive distribution of the reference model, and α is the elastic net mixing parameter that bridges the gap between Lasso ($\alpha = 1$) and ridge ($\alpha = 0$). Solving this for $\alpha > 0$ over a grid of values for λ yields a sequence of models with varying number of regression coefficients different from zero, which can then be used to order the features, for instance by recording the order in which their coefficients break nonzero as λ is decreased[6] (for more detailed discussion, see e.g. Hastie, Tibshirani and Wainwright, 2015).

One of the key advantages of elastic net over the forward stepwise search is that it is computationally very efficient. In particular, the coordinate descent algorithm of Friedman, Hastie and Tibshirani (2010) that exploits warm starts can often compute the solution path over the entire λ grid in comparable time to a single IRLS fit for a fixed variable combination. However, we do emphasize that unlike in the penalized GLM literature, we use the penalization *only* to find promising submodels, not to regularize their fit after selection. In other words, after we have solved problem (21) for a grid of values λ, we order the features from the most relevant to the least relevant, and find the projected parameter values (or projected posteriors) of the submodels *without* any penalization, or using only a small L_2-regularization to improve numerical stability. This is because the projection conditions on the information in the reference model and is therefore much more resilient to overfitting than maximum likelihood estimation for the parameters after selection. See Section 7.1 for an illustration of this point, and Section 7.3 for a demonstration of how the predictive accuracy can greatly benefit from not using the penalization for the submodels after selection.

In addition to Lasso and elastic net, there is a wide literature on different penalties for the (generalized) linear models, that are used to induce sparsity in the solution, and therefore could be used as search heuristics to find promising submodels for the projection also. One such method is the adaptive Lasso (Zou, 2006) which is obtained from (21) by introducing penalty factors γ_j that result in different penalization for different variables $\lambda_j = \gamma_j \lambda,\ j = 1, \ldots, p$. Plugging the local penalties into the regularization term in (21), the regularizer becomes

$$J(\boldsymbol{\beta}) = \lambda \sum_{j=1}^{p} \gamma_j \left(\frac{1}{2}(1-\alpha)\beta_j^2 + \alpha|\beta_j| \right).$$

[6]Notice that this is not necessarily the same order in which the coefficients go to zero as the penalty term λ is increased. This is because a coefficient that is nonzero can go back to zero as λ is reduced, but most of the time the two orderings are the same.

Using pilot estimates $\boldsymbol{\beta}'$ for the coefficients (that can be the univariate regression coefficients, for example) and setting $\gamma_j = 1/|\beta'_j|^\nu$ for some $\nu > 0$, adaptive Lasso reduces the excessive shrinkage of the relevant coefficients and recovers the true model under more general conditions than does the Lasso. Adaptive Lasso can also be used to encode preferences for different variables, for instance, due to varying measurement costs. In the projection context, Tran, Nott and Leng (2012) proposed to set $\boldsymbol{\beta}'$ to the posterior mode of the reference model (assuming it is also a GLM) whereas Hahn and Carvalho (2015) proposed to use the posterior mean (the two choices are in general different for GLMs with non-Gaussian priors for the reference model). Our approach differs from these in that we set $\gamma_j = 1$ for each feature in the selection phase but then relax completely $\gamma_j = 0$ after the feature selection is done. We also utilize clustered or draw-by-draw projection after selection when appropriate (see Sec. 7.1). Another difference to the approach of Hahn and Carvalho is that they used squared error instead of the KL-divergence to measure the discrepancy to the reference model. Nott and Leng (2010) also used L_1-penalization but for the draw-by-draw projection. In this method the different draws can generally project onto different feature combinations even for fixed λ, and thus this approach does not perform feature selection in the sense we are interested.

5. Validation and decision rules for model size selection

Although we can find the optimal reduced model for a given model complexity by selecting the model with minimal projection loss, making the decision about the appropriate model complexity using the KL-divergences is often difficult. A natural way of deciding the model complexity is to validate the predictive utility of both the reference model and the candidate reduced models on a validation set using a metric that is easy to interpret, and then make the decision based on these validation results. A generic and useful utility function is the mean log predictive density (MLPD) over the validation points (see, e.g. Vehtari and Ojanen, 2012), which has the advantage that it measures not only the point predictions but also how well the predictive uncertainties are calibrated. Various other utility and loss functions could also be used, such as mean squared error (MSE) or classification accuracy in classification problems, which are often easier to interpret.

If plenty of data are available and computation time is an issue, this assessment can be done on hold-out data. However, when data are scarce, more accurate assessment can be obtained using either leave-one-out (LOO) or K-fold cross-validation, which we shall discuss next.

5.1. K-fold cross-validation

In K-fold cross-validation both the reference model fitting and the selection is performed K times each time computing the utilities on the corresponding validation set (Peltola et al., 2014). This gives us the cross-validated pointwise

utilities $u_k^{(i)}$ for a given model complexity k (number of features) at each datapoint i. For instance, with log predictive density as the utility function, $u_k^{(i)}$ is the log predictive density of the submodel with k features evaluated at the left out y_i. These can then be used to make the final decision about the appropriate level of complexity. Our approach is to estimate the utility of each model size k relative to the reference model, that is, $\Delta U_k = U_k - U_*$, where U_k and U_* denote the true (unknown) utilities for the reduced and the reference model, respectively. The point estimate and the standard error for the relative utility ΔU_k in such pairwise comparison are given by

$$\Delta \bar{U}_k = \frac{1}{n} \sum_{i=1}^{n} \left(u_k^{(i)} - u_*^{(i)} \right), \tag{22}$$

$$s_k = \sqrt{\frac{1}{n} \mathrm{V}_{i=1}^{n} \left[u_k^{(i)} - u_*^{(i)} \right]}, \tag{23}$$

where $\mathrm{V}_{i=1}^{n}[\cdot]$ denotes the sample variance. Given the point estimate and its standard error it is easy to construct desired confidence intervals for ΔU_k. A natural choice is then to choose the simplest model that has acceptable difference relative to the reference model with some confidence (Piironen and Vehtari, 2017a).

A simple choice is to select the smallest model for which the utility estimate is no more than one standard error away from that of the reference model, that is, the smallest k that satisfies $\Delta \bar{U}_k + s_k \geq 0$, which means that the submodel is no worse than the reference model with probability approximately $\alpha = 0.16$. This approach has the drawback that such a model is not guaranteed to be found if the submodels all introduce a considerable loss in utility. Instead one could compare the utilities relative to the best submodel found, that is, in Equation (22) replace $u_*^{(i)}$ by $u_{k_{\mathrm{best}}}^{(i)}$ where $k_{\mathrm{best}} = \arg\max_k \Delta \bar{U}_k$. Based on the experiments in Section 7.4 the two choices perform quite similarly, the latter tending to select less parsimonious models but also with slightly better predictive accuracy. Depending on the application, one might be willing to sacrifice more utility in order to simplify the model ever further, and the decision about the appropriate model size could naturally be made on more subjective grounds also.

5.2. Leave-one-out cross-validation

The drawback in the K-fold cross-validation is that it requires fitting the reference model K times. Here we propose a new alternative approach using approximate leave-one-out (LOO) validation using the Pareto smoothed importance sampling (PSIS) (Vehtari, Gelman and Gabry, 2017), which avoids the repeated fitting of the reference model. In (PS)IS-LOO, the posterior draws can be treated as draws from the LOO posteriors given the importance weights. The weight for draw $\boldsymbol{\theta}_*^s$ after leaving ith observation out, $w_s^{(i)}$, is given by $w_s^{(i)} \propto \frac{1}{p(y_i \mid \boldsymbol{\theta}_*^s)}$. These raw weights are then regularized using Pareto smoothing to stabilize the

LOO estimates in case the importance weight distribution has a thick tail (see Vehtari, Gelman and Gabry, 2017, for the procedure). It is then easy to approximate the desired quantities for the LOO folds using these weights. For instance, in the clustered projection for the Gaussian linear model we need the predictive means $\boldsymbol{\mu}_*^c$ and variances $(V_1^c, \ldots, V_n^c)$ from the reference model for each cluster c. If the reference model is also linear with Gaussian noise, using the notation from Section 3.5, the predictive means at the observed inputs for the ith LOO are given by

$$\boldsymbol{\mu}_*^c = \sum_{s \in I_c} w_s^{(i)} \, \mathbf{Z} \boldsymbol{\beta}_*^s, \tag{24}$$

where the weights are assumed to be normalized $\sum_{s \in I_c} w_s^{(i)} = 1$. Correspondingly, the predictive variance at point j for the ith LOO is given by

$$V_j^c = \sum_{s \in I_c} w_s^{(i)} \, \sigma_{*,s}^2 + \mathrm{V}_{s \in I_c} \left[\mathbf{z}_j^\mathsf{T} \boldsymbol{\beta}_*^s, w_s^{(i)} \right], \tag{25}$$

where $\mathrm{V}_{s \in I_c}[\cdot, v_s]$ denotes the weighted sample variance over indices $s \in I_c$ with weights v_s. Equation (25) is merely the weighted version of formula (20). The feature selection and the projection onto the submodels at the search path are then carried out for each LOO exactly as in the K-fold case. Exactly the same decision rules as with the K-fold validation can be used to decide the appropriate model size, the LOO method simply gives an alternative procedure for computing the pointwise utilities, $u_k^{(i)}$ and $u_*^{(i)}$, for the reduced and reference models, respectively.

PSIS has the benefit that it gives us the Pareto $\hat{k}$-diagnostics for each LOO describing the accuracy of the importance sampling approximation, with $k \leq 0.7$ indicating reliable approximation (see Vehtari, Gelman and Gabry, 2017; Vehtari et al., 2019, for more precise discussion). Larger values indicate that the calculated utilities $u_k^{(i)}$ and $u_*^{(i)}$ for such observation i have high variance and can be biased (optimistic). In such cases, better estimates can be obtained by iterative moment matching LOO (Paananen et al., 2020) or K-fold validation. In Section 7.2 we demonstrate empirically that even when a few $\hat{k}$-values exceed this threshold, the relative utility estimate (22) can be nearly unbiased since the bias in both $u_k^{(i)}$ and $u_*^{(i)}$ tends to cancel out in the subtraction.

5.3. Importance of validating the search

In order to reduce computations, it might be tempting to perform the reference model fitting and feature selection only once using all the available data, and then simply use LOO or K-fold CV to estimate the performance of the found submodels. We strongly advice *not* to employ this strategy, as this is known to produce biased performance estimates, and the bias can be substantial especially for small n and large number of features (see Piironen and Vehtari, 2017a, for

illustrations). To avoid the selection induced bias, it is important that the same data are never used simultaneously for selection and assessment, meaning that the selection must be performed separately for each of the cross-validation folds regardless of the feature selection method. Section 7.2 shows an example of the resulting bias when the selection process is not taken into account in the model assessment.

6. On the construction of the reference model

How to construct a good reference model is naturally a central issue in the whole projective approach. It should be clear that this is essentially an open-ended question with no definite answer; for each problem there are endless choices. For simple linear and logistic regressions with moderate number (say less than a hundred) features we recommend using all the features with a sparsifying prior, which can work better than a non-sparse prior like Gaussian. If one is uncertain about the prior, the recommended strategy is to try different choices and compare the resulting fits with cross-validation.

In high-dimensional problems, say with hundreds of features or more, fully Bayesian approach can still provide a good fit but can also prove computationally expensive (Piironen and Vehtari, 2017b). Using either feature screening, dimension reduction or the combination of the two can be very successful for alleviating the computational burden without sacrificing the predictive accuracy (Neal and Zhang, 2006; Fan and Lv, 2008; Piironen and Vehtari, 2018). In our experience this is true especially for data sets that have plenty of features many of which are correlated with each other and predictive about the target variable. Microarray data sets (Sec. 7.4) are typical examples that fall into this category.

For these problems a simple but useful recipe combining feature screening and dimension reduction is known as supervised principal components (SPCs) (Bair et al., 2006), which works as follows. First, univariate correlations $R(x_j, y)$ between each feature x_j and the class label y are computed, and only features with $|R(x_j, y)|$ above some threshold γ are retained. This yields a reduced feature matrix $\mathbf{X}_\gamma$, from which one then computes the first n_c principal components $(z_1, \ldots, z_{n_c})$ and uses these as the predictors for the reference model. The advantage over the unsupervised principal components is that the screening step anticipates variation in the original features unrelated to the variance in y, and therefore the predictive power is typically more heavily loaded on the first few components. In the experiments of this paper, the screening threshold γ is selected using fivefold cross-validation from a coarse grid of $n_\gamma = 7$ values evenly spaced between $\gamma_{\min}$ and $\gamma_{\max}$, where $\gamma_{\min}$ is the largest γ such that none of the features are discarded and $\gamma_{\max}$ the smallest γ such that only one feature survives the screening. Furthermore, we use $n_c = 3$ SPCs with a Gaussian prior $\mathrm{N}\big(0, \tau^2\big)$ for the regression coefficients and hyperprior $\tau \sim t_4^+\big(0, s_{\max}^{-2}\big)$ where $s_{\max}$ denotes the standard deviation of the largest principal component (this is done only to make the prior roughly the same regardless of the scale of the SPCs).

We emphasize that we do *not* argue that this gives a foolproof method for constructing a good reference model. Rather the purpose is to demonstrate that even with such a simple, easy-to-implement and computationally light method it is possible to come up with a reference model that gives good results and improves feature selection in many cases. Indeed, in our earlier work we found that the optimal method is in general data set dependent, and in some cases better results can be obtained by other choices such as the iterative version of the above algorithm (Piironen and Vehtari, 2018). Again, cross-validation and posterior predictive checks should be used to guide the selection of the reference model (Gelman et al., 2013; Vehtari, Gelman and Gabry, 2017; Gabry et al., 2018). A generic strategy for improving the prediction accuracy is also to average over several models, using either stacking or (pseudo) Bayesian model averaging (Yao et al., 2018), or boosting or bagging in non-Bayesian context (see, e.g., Hastie, Tibshirani and Friedman, 2009).

7. Experiments

This section presents several examples of the projective method. We shall first demonstrate the basic usage of the different projection techniques and the new LOO validation for the model size selection, and then compare the projective approach to the elastic net family estimators. For fitting the Bayesian reference models we use Stan (Stan Development Team, 2018), with the convenient interfaced to GLMs provided by R-packages `rstanarm` (Goodrich et al., 2018) and `brms` (Bürkner, 2017). All the projections are computed using our R-package `projpred`. The results for the elastic net family methods are computed using R-package `glmnet` (Friedman, Hastie and Tibshirani, 2010).

7.1. Illustration of different projections

This section illustrates the differences between the three projection techniques introduced in Section 3.3. Consider the following synthetic binary classification data. For instances belonging to the first class, the first three features are drawn from independent Gaussians with mean 1 and scale 0.5, whereas for the observations from the second class the mean and scale of these features are -1 and 0.5, respectively. In addition, the data has 27 additional noise features that are drawn from independent standard Gaussians, so the data has 30 features altogether (out of which the only the first three are predictive about the class label). We generated one data realization with $n = 50$ observations and fitted Bayesian logistic regression model to those data using the RHS prior (5) with hyperparameter choices $p_0 = 1$, $s^2 = 1$ and $\nu = 4$. This serves as our reference model.

Figure 5 illustrates the posterior projection onto the first two features for the three different projections: draw-by-draw (left column), single point (middle column) and 10 clusters (right column). We observe that even with the single point projection, the predictive probabilities are very close to those of

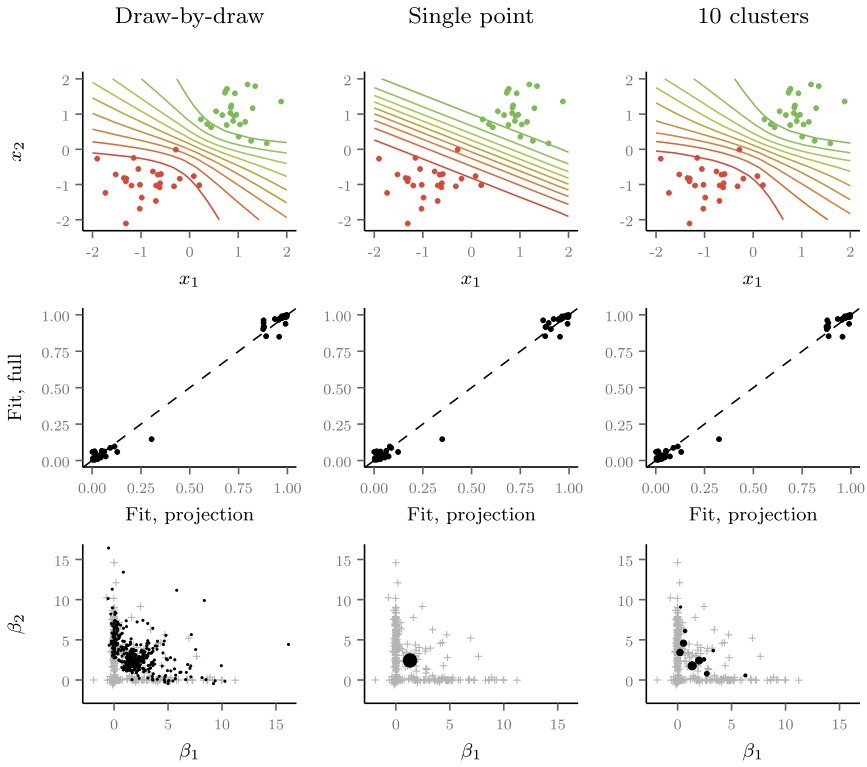

FIG 5. *Demonstration of different projections: The full posterior with $p = 30$ features projected onto the first two features using the draw-by-draw approach (left column), single point projection (middle column) or 10 clusters projection (right column). Top row shows the observed data and the contours (from 0.1 to 0.9) of the predictive probability for $\tilde{y} = 1$, whereas middle row shows the predictive probabilities at the observed input locations (vertical axis denoting the result for the full model with all features, and horizontal axis for the projection of the corresponding column). Bottom row shows the projected regression coefficients (black dots) as well as the draws from the full posterior (gray crosses). In bottom row plots, the dot sizes denote the relative weights (the dot sizes between different columns are not comparable).*

the draw-by-draw projection (see top and middle row), and projecting 10 clusters gives predictions indistinguishable from the draw-by-draw projection for all practical purposes. This result is insightful, as one might think that the single point projection would be substantially inferior because it computes only point estimates for the projected model. The key insight is that these point estimates are computed so as to take into account the uncertainty in the parameters of the full model. Therefore the resulting predictive distribution is much closer to that of the full model than what would be obtained by projecting only the posterior mean of the full model or by computing the maximum likelihood estimates for the submodel (which in this case do not even exist because the classes are separable).

Another important point is that even for the draw-by-draw method the projected posterior is in general different from the marginal posterior for those parameters in the full model (see the bottom left plot in Fig. 5). In particular, the projected posterior has vanishingly little mass near the origin $\beta_1 = \beta_2 = 0$, although the full posterior has substantial mass there. This makes sense: after removing feature x_3 which is predictive and highly correlated with x_1 and x_2 the coefficients of x_1 and x_2 can not *both* be set to zero, otherwise the predictions would seriously be affected.

As discussed earlier, the benefit of the clustered projection compared to the draw-by-draw projection is its speed; projecting only C clusters cuts down the computations by a factor of C/S, where S is the number of draws that would be projected in the draw-by-draw projection. The computational savings can be huge when projections need to be computed onto many models, such as with the LOO validation. For instance, for this data set computing the projections of each of the $n = 50$ LOO posteriors for all model sizes up to 30 features in a naive fashion would require a total of 1500 projections, each of which takes around a second or two depending on the hardware. Thereby with the clustered projection we can reduce the computation time from the order of 25–50 minutes to about 4–8 seconds[7]. The additional benefit of the single point projection is that it can be combined with the sparsity enforcing penalty functions (Sec. 4) which allows for fast searching for promising submodels.

For these reasons, our preferred choice is to use single point projection in the selection phase, and a small number of clusters (1–10) when making predictions with the submodels. Even though the difference in the predictions with 1 or 10 clusters is small in this example, adding more than one cluster can sometimes give slightly more accurate predictions with very little computational overhead. Still, we find the draw-by-draw projection most convenient for visualizing the projected posterior distributions for instance when credible intervals or regions are of interest. It also serves as a useful yardstick for checking and confirming the accuracy of the clustered projection.

7.2. Simulated example revisited with projection and LOO

We shall now revisit the simulated example discussed in Section 2.3 and illustrate the steps of projective selection as well as our new LOO validation technique. The first step is to decide the reference model, which we would in practice do by assessing the fits of each of the candidate models using cross-validation and posterior predictive checks. The sums of LOO log predictive densities for the Gaussian and RHS priors are -76.8 and -77.6 with standard errors 6.8 and 6.2, respectively, so there is no significant difference between the predictive fits between these models (this holds also if we make the comparison in pairwise

[7]In a careful implementation the difference would not be quite as dramatic since some of the submodels would be visited in many of the $n = 50$ folds, so their projections would not need to be computed again every time, but this example still gives a good idea of the computational gain.

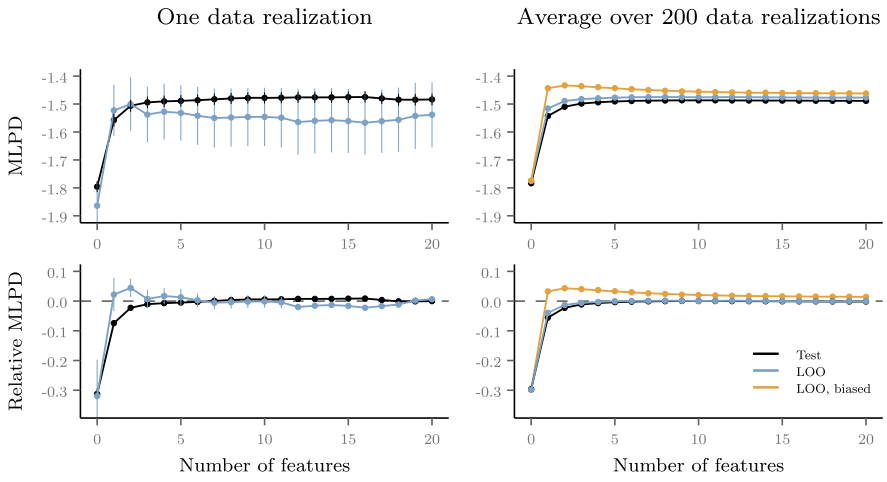

FIG 6. *Simulated example, projective selection: Left column: MLPD and relative MLPD with one standard error intervals on independent test data of 1000 points (black) and using LOO (blue) for the selected and projected submodels. The data has $p = 50$ features and the reference fit is the linear model with RHS prior (the same as in the right middle subplot of Figure 3). Right column: The same but results are averaged over 200 data realizations. The orange curves show the LOO for the submodels if the feature selection is done only once, and not separately for each of the n folds. The difference to the blue curve comes from the selection induced bias.*

fashion, like in Eq. (22) and (23)). The R package `spikeslab` does not provide the posterior draws for the regression coefficients and thus we cannot compute LOO for the SS prior, so we ignore it for now.

Suppose we select the model with RHS prior as our reference model (the results for Gaussian prior are shown in Appendix A). We then run the projective feature selection with the L_1-search and assess the accuracy of the submodels using the LOO approach (Sec. 5.2). The MLPD for the submodels relative to the reference model are shown in the bottom left subplot of Figure 6 (blue curve). The one standard error -rule (Sec. 5.1) would suggest selecting one feature, which results in a small loss in accuracy on test data (black curve) compared to the reference model. The top left subplot shows the MLPD on the actual scale, which demonstrates how much larger the uncertainty is about the actual MLPD than about the relative MLPD.

The right column of Figure 6 shows the average LOO curves for both MLPD and relative MLPD over 200 data replications. These graphs demonstrate that the actual LOO values for the submodels are slightly biased (optimistic). This is due to a small bias in the PSIS-LOO for the reference model, which is also diagnosed by a few $\hat{k}$-values that exceed 0.7 in most data realizations. Notice though, that the results for the relative MLPD are still essentially unbiased for submodels with performance close to the reference model, because the bias cancels out in subtraction (22) (see Sec. 5.2). In other words, even if we have

only a biased estimate of the reference model utility, we can still get a good indication of whether our submodel performance is close to that of the reference model.

Right column subplots of Figure 6 also show the expected LOO results if we do *not* take into account the selection induced bias but perform the selection only once (not separately for the n folds) and then compute LOO for the submodels (see Sec. 5.3). The selection induced bias is clear although only moderate in this particular example.

For assessing the submodel accuracies, LOO validation is very useful in this particular example because of a few reasons. Firstly, PSIS-LOO works pretty well for the full model (only a few k-values above 0.7 in most data realizations). Secondly, the number of features is only moderate and hence the feature selection is very fast. Thirdly, the number of observations is small, so the number of selection paths we need to compute is also small. Consequently, the whole validation process takes only a few seconds, which is much less than a single model fit with the horseshoe prior (around half a minute with a standard laptop), so the computational savings compared to K-fold cross-validation are clear.

7.3. The benefit of using a reference model

This section demonstrates the benefits of a reference model for feature selection and parameter estimation in the submodels. We again utilize simulated data generated by mechanism (1), and consider regression of the original y on $(x_1, \ldots, x_p)$.

We used a setup with $n = 50$ training observations with $p = 500$ features, out of which first $p_{\text{rel}} = 50$ were relevant, and report average results over 50 data realizations for ρ-values of 0.3, 0.5 and 0.8.[8] The reference model is fitted using SPCs as discussed in Section 6. We tested four different strategies for selecting features and making predictions with the selected subsets of features:

1. *Lasso*: Sort the variables from the most relevant to least relevant according to the order in which they enter the model as the regularization coefficient λ is decreased. For a given number of features, the submodel coefficients are computed using the smallest λ for which other variables do not enter the model. In the regression problems, the noise variance σ^2 is estimated as proposed by Reid, Tibshirani and Friedman (2016), that is, by dividing the sum of the squared residuals by $n - p_{\text{act}}$ where p_{act} denotes the number of active features in the submodel.
2. *Lasso, relaxed*: Same as Lasso, but after sorting the variables, the submodel coefficients and predictions are computed without any regularization (which affects also the estimated noise variance in regression).[9]

[8] We also considered varying values for p_{rel} but the conclusions are not sensitive to the selected value $p_{\text{rel}} = 50$.

[9] We are aware that the term 'relaxed Lasso' has been used to denote a more general method where after feature selection the coefficients are computed with a small but nonzero L_1-penalty (Meinshausen, 2007). The complete relaxation (i.e., zero penalty after selection) was referred to as 'Lasso-OLS hybrid' by Efron et al. (2004)

3. L_1-*projection*: L_1-penalized projection (21) varying λ similarly as in Lasso. In regression, the projected noise variance is computed according to Equation (19) (where $C = 1$).
4. L_1-*projection, relaxed*: Same as L_1-projection, but after sorting the variables, the submodel coefficients are projected without any regularization (which affects also the projected noise variance in regression).

Notice that all these methods utilize only point estimates for the model parameters in the submodels, the difference is only how they are computed.

Figure 7 shows the regression MLPD and MSE on independent test data as well as the projected noise standard deviation for different submodel sizes. The blue curves demonstrate the benefit of relaxation for L_1-projection: both eventually achieve the performance of the reference model but without relaxation this requires many more features. The reason is the inherent tradeoff between shrinkage and selection: in order to force most of the regression coefficients to zero, the regularization coefficient λ must be made large, but this will also overshrink the nonzero coefficients. Therefore projecting without any penalization after selection achieves greatly improved tradeoff between accuracy and model complexity. Notice in particular that here no regularization is needed to avoid overfitting in projection; when more features are added the projected submodels simply get closer to the reference model.

However, the picture is quite different when the parameter estimates are computed based on the observed data without utilizing the reference model (Fig. 7, orange curves). The relaxation improves the fit in terms of MSE for submodels with only a few features but results in overfitting for larger models. In terms of MLPD the relaxed Lasso performs worst overall indicating badly calibrated uncertainties in the predictive distributions, which is mostly due to underestimation of the noise variance (bottom row) for most model sizes. Projection methods on the other hand show very good calibration of predictive uncertainties which is evident from superior MLPD and noise variance estimation for most model sizes. Overall the projection approach shows a bigger edge for $\rho = 0.3$ and $\rho = 0.5$ where the individual features are less predictive. Results in Appendix A.3 show that these conclusions are very similar also for a binary classification (logistic regression) setup.

7.4. Real world benchmarks

This section shows how the projection compares in high-dimensional real world problems. We use microarray data sets[10] some of which have been used as benchmarks by several authors (Li, Campbell and Tipping, 2002; Lee et al., 2003; Hernández-Lobato, Hernández-Lobato and Suárez, 2010). All data sets deal with binary classification, and the number of features and data set sizes can be found in Table 1.

[10]All except the Ovarian data are available at http://featureselection.asu.edu/datasets.php.

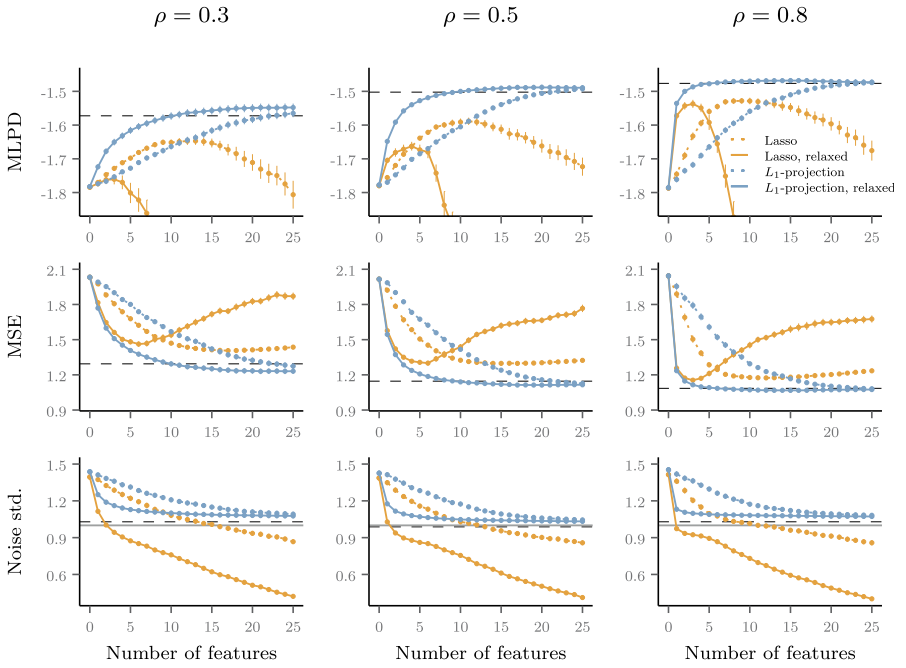

FIG 7. *Benefit of reference model, regression: MLPD and MSE on test data, along with the estimated noise standard deviation as a function of number of features selected after L_1-penalized search, before and after relaxation (dashed and solid, respectively), with and without utilizing the reference model (blue and orange, respectively). Different columns show results for different values of ρ (see Eq.* (1)*). Errorbars indicate one standard error intervals and black dashed lines the reference model result. In the bottom row plots the gray line denotes the true noise standard deviation.*

Again, as a reference model we use the one described in Section 6 and call it here 'Bayes SPC'. For the projection method, we used L_1-search and made the submodel predictions using five clusters projection. The number of features was decided based on fivefold cross-validation. To investigate the effect of the model size selection heuristic discussed in Section 5.1, we report results for the smallest number of features that had its cross-validated MLPD within one standard error away from the reference model ('Proj-ref-1se') or from the best submodel ('Proj-best-1se'). We also report results ('Proj-ref-1se-reg' and 'Proj-best-1se-reg') that are otherwise exactly the same as the two above but utilize a little bit of ridge regularization (with $\lambda = 0.1$) in the submodel projections which was observed to improve the numerical stability in cases where some of the reference model class probabilities are close to 0 and 1.

For comparison, we computed results for Lasso, elastic net (with $\alpha = 0.7$ and $\alpha = 0.3$) and ridge. To investigate the sensitivity of these to the selection of the regularization parameter λ, we report results for two choices: λ_{opt} denotes the value that minimizes the tenfold CV-error whereas λ_{1se} (default in `glmnet`)

TABLE 1

Microarray benchmark data sets: Average computation time (in seconds) over five repeated runs. In all cases the time contains the cross-validation of the tuning parameters and/or the model size. The first result for Lasso is computed using our software (`projpred`*) whereas the second result (and that of ridge) is computed using the R-package* `glmnet` *which is more highly optimized.*

Data set	n	p	Computation time				
			Bayes SPC	Projection	Lasso (`projpred`)	Lasso	Ridge
Ovarian	54	1536	30.4	3.6	1.3	0.2	1.5
Colon	62	2000	31.0	4.0	1.6	0.3	2.2
Prostate	102	5966	49.4	7.6	5.0	0.8	7.5
Leukemia	72	7129	47.0	6.3	5.6	0.7	9.4
Glioma	85	22283	95.8	14.2	15.6	2.6	52.2

denotes the largest λ which has its CV-error within one standard error of that of λ_{opt}. To avoid any possible biases in the comparisons, the out-of-sample predictive accuracies for all the methods were assessed using an outer tenfold cross-validation. That is, the reference model, projected submodels as well as the baseline methods were computed ten times, each time leaving one tenth of the data out and then validating the found models on this left-out data.

The MLPD and classification accuracies from the outer cross-validation are shown in the first two rows of Figure 8. Overall the differences in accuracy between the methods are fairly small compared to the standard errors in the estimates. In terms of MLPD, the reference model Bayes SPC gives somewhat better results than Lasso, elastic net and ridge with λ_{1se}, but all these give similar results when λ_{opt} is used, and in fact ridge gives a bit better results for Leukemia data. All projections have statistically indistinguishable MLPD compared to Bayes SPC, but the model size selection with 'best-1se' performs slightly better in terms of classification accuracy. Adding a little bit of regularization does not hurt predictive accuracy but we noticed that it makes the projection numerically more stable in cases where the reference class probabilities are close to 0 and 1.

The bottom row of Figure 8 shows the number of selected features for each method. The projection methods produce by far the most parsimonious models (notice the log scale). The only data set where Lasso (with λ_{1se}) selects fewer variables is Leukemia, but there it also yields inferior results in terms of MLPD. This is perfectly in accordance with the results shown in Figures 7 and 11: the projection finds very good tradeoff between sparsity and accuracy. Although not shown in Figure 8, the accuracy of the baseline methods would severely be affected were they allowed to use as few features as the projection methods. To fully respect the differences in the number of features used, we have also reported them using hard numbers in Table 2 (appendix A) since an accurate comparison on the log scale is somewhat cumbersome.

The computation times are shown in Table 1. After forming the reference model, the projection is computationally only slightly more expensive than Lasso and the increase comes from the relaxed projections (the predictions are com-

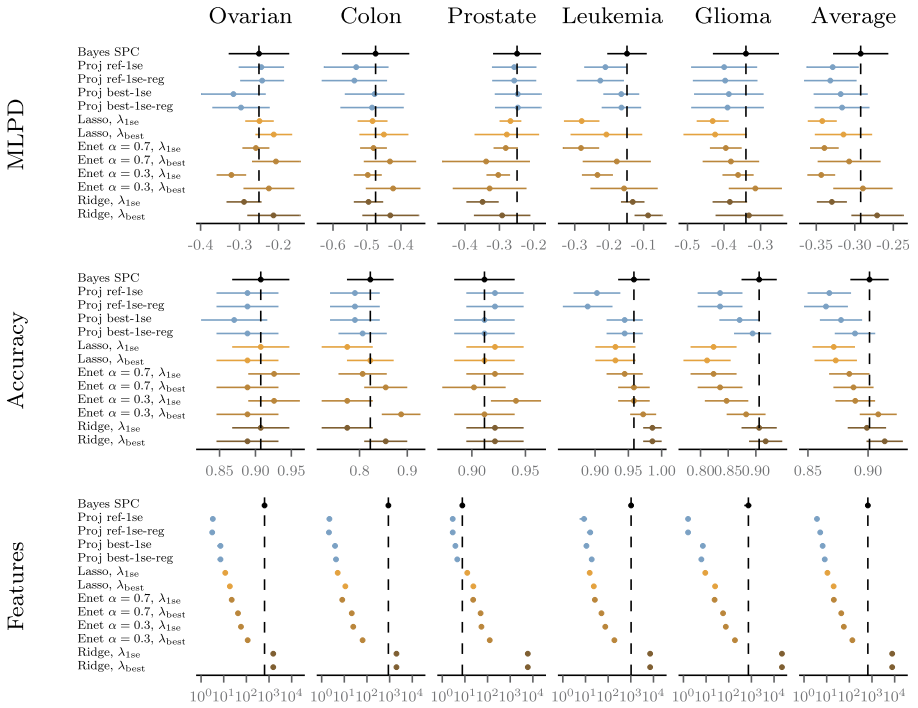

FIG 8. *Microarray benchmark data sets: MLPD (top row), classification accuracy (middle row) and the number of features used (bottom row) with one standard error intervals for the different data sets. The last column denotes the average. In all plots the dashed vertical line denotes the results for the Bayes SPC that is used as the reference for the projections. Many methods produce comparable predictive accuracy but the projection methods achieve the same accuracy with far fewer features (notice log scale in the bottom row plots).*

puted without the L_1-penalty). Although not as highly optimized as `glmnet`, our software is reasonably fast even for the largest problems. Forming the reference model (Bayes SPC) is computationally the most expensive part, though still very affordable considering that all the computations (reference model construction and projection) for the largest number of features can be done in about two minutes. Indeed, this demonstrates that the projection can be very feasible computationally and it can yield improved results to the standard approaches, as were shown in Figure 8.

8. Theoretical results

In this section we present a theorem that helps us to understand when the reference model could be helpful for parameter learning in linear submodels. Here we only state the results, the proofs can be found in appendix B.

Let $\mathbf{X} = (\mathbf{x}_1^\mathsf{T}, \ldots, \mathbf{x}_n^\mathsf{T}) \in \mathbb{R}^{n \times p}$ be the design matrix and $\mathbf{y} = (y_1, \ldots, y_n) \in \mathbb{R}^n$ the target measurements. Assume the target measurements decompose as

$y_i = \mu(\mathbf{x}_i) + \varepsilon_i$, where $\mu(\mathbf{x})$ is the true expected value of y given $\mathbf{x}$, $\mu(\mathbf{x}) = \mathrm{E}(\, y \,|\, \mathbf{x}\,)$, and ε_i are i.i.d. random numbers independent of $\mathbf{x}$ with zero mean and finite variance σ^2 denoting the variation in y that cannot be explained by $\mathbf{x}$. It should be emphasized that although we assume ε denotes random error independent of $\mathbf{x}$, it may contain systematic variation related to some other (unobserved) features not included in $\mathbf{x}$, and hence the magnitude of ε should be interpreted as the irremovable error for this particular set of features $\mathbf{x}$. In vector notation, $\mathbf{y}$ decomposes as $\mathbf{y} = \boldsymbol{\mu} + \boldsymbol{\varepsilon}$, where $\boldsymbol{\mu} = (\mu(\mathbf{x}_1), \ldots, \mu(\mathbf{x}_n))$ and $\boldsymbol{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_n)$. Furthermore, in what follows we shall use the shorthand notation $||\mathbf{v}||^2_{\mathbf{M}} = \mathbf{v}^\mathsf{T}\mathbf{M}\mathbf{v}$, where $\mathbf{v}$ is a vector and $\mathbf{M}$ a positive definite matrix.

Consider two methods of estimating the regression coefficients when regressing $\mathbf{y}$ on $\mathbf{X}$, namely

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}\mathbf{y} \quad \text{and} \quad \boldsymbol{\beta}_\perp = (\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}\boldsymbol{\mu}_*. \tag{26}$$

Here $\hat{\boldsymbol{\beta}}$ is the familiar least squares estimate, and $\boldsymbol{\beta}_\perp$ a projection of an arbitrary reference fit $\boldsymbol{\mu}_* \in \mathbb{R}^n$. Let us then define the expected prediction error for any vector of coefficients $\boldsymbol{\beta}$ as

$$\Delta(\boldsymbol{\beta}) = \mathrm{E}_{\tilde{\mathbf{y}}}\left(\frac{1}{n}||\mathbf{X}\boldsymbol{\beta} - \tilde{\mathbf{y}}||^2 \right).$$

Notice that although we consider here the predictions at the observed input locations $\mathbf{X}$, the expectation is with respect to a set of *new* measurements $\tilde{\mathbf{y}} = \boldsymbol{\mu} + \tilde{\mathbf{e}}$, where $\tilde{\mathbf{e}}$ is a vector of new noise terms $\tilde{\varepsilon}_1, \ldots, \tilde{\varepsilon}_n$. The gain from using $\boldsymbol{\beta}_\perp$ instead of $\hat{\boldsymbol{\beta}}$ is defined as the reduction in the expected prediction error

$$G = \Delta(\hat{\boldsymbol{\beta}}) - \Delta(\boldsymbol{\beta}_\perp). \tag{27}$$

We have the following lemma.

Lemma 1 *Assume regression coefficient estimators $\hat{\boldsymbol{\beta}}$ and $\boldsymbol{\beta}_\perp$ as defined by Equation* (26)*. The gain G (Eq.* (27)*) of using $\boldsymbol{\beta}_\perp$ instead of $\hat{\boldsymbol{\beta}}$ satisfies*

$$G = \frac{1}{n}\left(||\mathbf{y} - \boldsymbol{\mu}||^2_{\mathbf{P}} - ||\boldsymbol{\mu}_* - \boldsymbol{\mu}||^2_{\mathbf{P}} \right),$$

where $\mathbf{P} = \mathbf{X}(\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}$.

Since both $||\mathbf{y} - \boldsymbol{\mu}||^2_{\mathbf{P}}$ and $||\boldsymbol{\mu}_* - \boldsymbol{\mu}||^2_{\mathbf{P}}$ are non-negative, the interpretation of Lemma 1 is that for linear submodels, one can expect to gain (that is, $G \geq 0$) from using a reference model when the reference fit $\boldsymbol{\mu}_*$ is closer to the best possible prediction $\boldsymbol{\mu}$ (with features $\mathbf{x}$) than the observed noisy target values $\mathbf{y}$ (with the norms taken with respect to the projection matrix $\mathbf{P}$). This makes perfect sense: if we fit our model to pseudo-data $\boldsymbol{\mu}_*$ instead of the actual data $\mathbf{y}$, we expect to do better if the pseudo-data are closer to the true underlying conditional mean $\mu(\mathbf{x}) = \mathrm{E}(\, y \,|\, \mathbf{x}\,)$, that is, less noisy than the actual data. Notice

that the lemma makes no assumptions about the form of the true underlying mean $\mu(\mathbf{x})$ that captures the relationship between y and $\mathbf{x}$. In particular, $\mu(\mathbf{x})$ need not be linear in $\mathbf{x}$, not even smooth or continuous. Neither does the lemma assume anything about how the reference fit $\boldsymbol{\mu}_*$ is constructed.

Let us now assume the differences $\mathbf{e}_* = \boldsymbol{\mu}_* - \boldsymbol{\mu}$ are random numbers with mean $\mathbf{b}$ and covariance $\mathbf{K}$. These describe the bias and variance in the reference fit.[11] We have the following theorem

Theorem 2 *Assume the terms $\mathbf{e}_* = \boldsymbol{\mu}_* - \boldsymbol{\mu}$ have mean $\mathbf{b} \in \mathbb{R}^n$ and covariance $\mathbf{K} \in \mathbb{R}^{n \times n}$. Then the expected gain can be written as*

$$\mathrm{E}(G) = \frac{1}{n}\left(\sigma^2 p - \mathrm{Tr}(\mathbf{PK}) - ||\mathbf{b}||^2_{\mathbf{P}}\right).$$

Theorem 2 further decomposes the reference model error into bias and variance. The term $\mathrm{Tr}(\mathbf{PK})$ is difficult to grasp without further assumptions, but the theorem can be understood more easily by the following immediate corollary.

Corollary 2.1 *Assume the reference model errors are uncorrelated with a common variance, that is, $\mathbf{K} = \sigma^2_{\mu_*}\mathbf{I}$. Then the expected gain $\mathrm{E}(G)$ simplifies to*

$$\mathrm{E}(G) = \frac{p}{n}\left(\sigma^2 - \sigma^2_{\mu_*} - \frac{1}{p}||\mathbf{b}||^2_{\mathbf{P}}\right).$$

This corollary states that with an unbiased reference model ($\mathbf{b} = 0$) we can expect to gain when the variance of the residuals $\boldsymbol{\mu}_* - \boldsymbol{\mu}$ is smaller than the variance of $\mathbf{y} - \boldsymbol{\mu}$. Furthermore, the gain increases with the dimensionality of the projection space p, but on the other hand goes to zero when $n \to \infty$. This is also in perfect accordance with the empirical results, for instance those shown in the middle row of Figure 7. There the difference in predictive MSE between relaxed Lasso and projection is small up to about $p = 2$, but then starts to increase gradually. On the other hand we know the least squares fit gives us the optimal coefficients at the limit $n \to \infty$ and hence we do not expect to gain anything then with a reference model.

The above analysis assumes the future predictions are made at the observed input locations. Usually this is not quite a realistic assumption, but it still gives us some idea when the reference model could be useful. Extending the result to different $\tilde{\mathbf{x}}$ would require assumptions about the functional form of $\mu(\mathbf{x})$. Furthermore, here we used squared error as the loss function as it allows for tractable analysis, but we expect one of the major advantages of the projection to be that it conserves the predictive uncertainties well which are not measured by the squared error. Finally, this analysis considers only parameter learning

[11] Here we mean bias and variance both due to fitting the reference model to a finite data set and having unobserved features. For example, even if our reference model was a completely deterministic function of $\mathbf{x}$ and some other features $\mathbf{z}$, then its value in a particular location $\mathbf{x}_i$ is still random as it depends on the realized value for $\mathbf{z}$.

in the submodels with a *fixed* set of features, but it says little about when the reference model can improve the *selection* of a better feature combination. In principle it is possible to improve selection even when the reference model is not unbiased, as long as the bias is "in the right direction", for instance so that it favors certain features over the others. We have discussed in a bit more detail in our earlier work, see Section 3 in Piironen and Vehtari (2017a). The empirical evidence about the improved selection is convincing (Sec. 1.2) but currently we are not aware of any theoretical analysis on this topic.

9. Discussion

Sparsity enforcing (non-Bayesian) penalized estimators—in particular Lasso and the whole elastic net family—are both fast to compute and provide good results with minimal hand tuning. This makes them excellent baselines for almost any problem. One can, however, do even better in terms of tradeoff between sparsity and predictive accuracy, by forming a reasonable reference model and then finding its projections onto reduced set of features. The projective framework provides a systematic way of handling uncertainties in Bayesian fashion and also a principled way of estimating other parameters than the regression coefficients (such as the noise variance).

This paper has focused on selecting a minimal subset of features that are sufficient for achieving accurate predictions. As pointed out in Section 1.3, this is a different problem from what is known as multiple (hypothesis) testing, where the goal is to identify as many features as possible that are statistically related to the target variable (Johnstone and Silverman, 2004; Efron, 2010; Castillo and van der Vaart, 2012). The empirical evidence indicates that the reference model approach could be highly useful also in this problem setting since it tends to help rank the truly relevant features before the irrelevant ones (Sec. 1.2), but the topic requires more research.

Ultimately it would be desirable to extend the projective approach to nonlinear and nonparametric models such as Gaussian processes (GP) (for tentative work, see Piironen and Vehtari, 2016). The approach could also find more applications in improving interpretability and transparency of complex black box models such as deep neural networks, an idea that have been explored by Ribeiro, Singh and Guestrin (2016); Peltola (2018) and Afrabandpey et al. (2019). Although the "fitting-to-fit" approach could in principle be applied to almost any kind of model, plenty of work remains in formulating the projection in a computationally tractable way for all these different cases.

Acknowledgments

We thank anonymous reviewers for useful comments, Michael Riis Andersen for helpful discussions and Academy of Finland (grants 298742 and 313122) for partial funding. We also acknowledge the computational resources provided by the Aalto Science-IT project and support by the Academy of Finland Flagship programme: Finnish Center for Artificial Intelligence, FCAI.

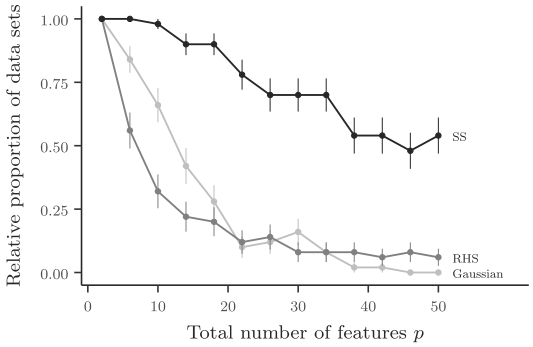

FIG 9. *Simulated example (Sec. 2.3): Relative proportion of data sets where at least one feature is found to be significant using marginal relevance assessment. With SS prior feature is considered significant if its posterior probability exceeds 0.5 and with Gaussian and RHS priors if its coefficient is either positive or negative with posterior probability 0.95 or more. The results are computed from 50 randomly generated data sets generated according to* (1) *with* $n = 50$, $\rho = 0.8$ *and* $p_{rel} = \frac{p}{2}$. *Vertical bars denote one standard error intervals.*

A. Extra experimental results

A.1. Marginal relevance assessment

Section 2.3 demonstrated how the posterior marginals tend to overlap with zero when the data contains many relevant correlated features. Figure 9 simply confirms that these observations are not due to cherry-picking a specific data set. For each of the three priors the relative proportion of data sets where at least one feature is found to be significant goes down when p increases. With Gaussian and RHS priors this probability is already fairly close to zero with $p = 50$, and even with SS we fail to find any relevant features in about half of the data sets. The exact proportions are naturally dependent on the selected thresholding rules (posterior probability of 0.5 in SS and credible level 0.95 for Gaussian and RHS) but these do not affect the main conclusions.

A.2. Simulated example with projection and LOO

Figure 10 shows the analogous results to Figure 6 but using the full model with Gaussian prior as the reference model. The results are otherwise similar to those in Figure 6, but here the submodels with 3 to 14 features achieve a slightly better generalization performance than the reference model. This is simply due to the fact that the Gaussian prior is not the optimal choice in this particular case since it does not help to shrink the coefficients of the truly irrelevant features, and hence we can gain by removing those features. This does not mean that the Gaussian prior would always be inappropriate even for very high-dimensional problems, see for instance the results for the microarray data

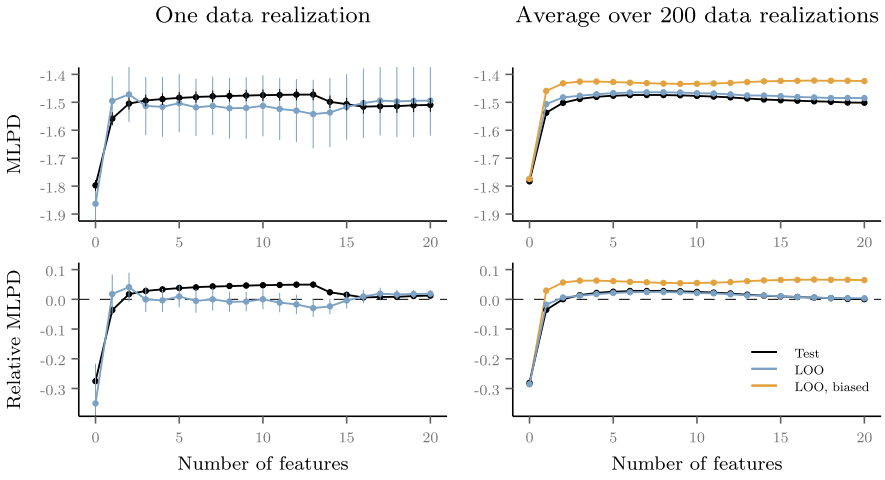

FIG 10. *Simulated example, projective selection: The same as in Figure 6 but using the full model with Gaussian prior as the reference model.*

sets in Section 7.4 where the ridge regression performs very well (corresponds to maximum a posteriori solution with the Gaussian prior).

A.3. The benefit of using a reference model

Section 7.3 demonstrated the benefit of a reference model with a regression example. Here we show the results for similar data, but converting the task to a binary classification (logistic regression) setup by defining the target variable as an indicator $y_{\text{class}} = \mathbb{1}(y > 0)$, where y is the continuous target used in the regression case.

Figure 11 shows the results for the classification data. The conclusions are very similar to those drawn from Figure 7. Here the relaxed Lasso overfits even more severely; although the classification accuracy is similar to the Lasso, the MLPD is very low indicating bad calibration in the predicted class probabilities. Again, the edge for projection is more pronounced for $\rho = 0.3$ and $\rho = 0.5$, but we observe that for these cases also the relaxed projection struggles to achieve the same MLPD as the reference model, and for larger number of features (15–25) the penalized projection achieves slightly better results. This is due to a small instability of the projection in data sets where some of the reference class probabilities are close to 0 and 1 and shows that even the projection, although very resilient, is not always entirely immune to overfitting.

A.4. Real world benchmarks

Table 2 shows the average number of features selected by the different methods in the microarray examples (Sec. 7.4). The projection methods clearly select the

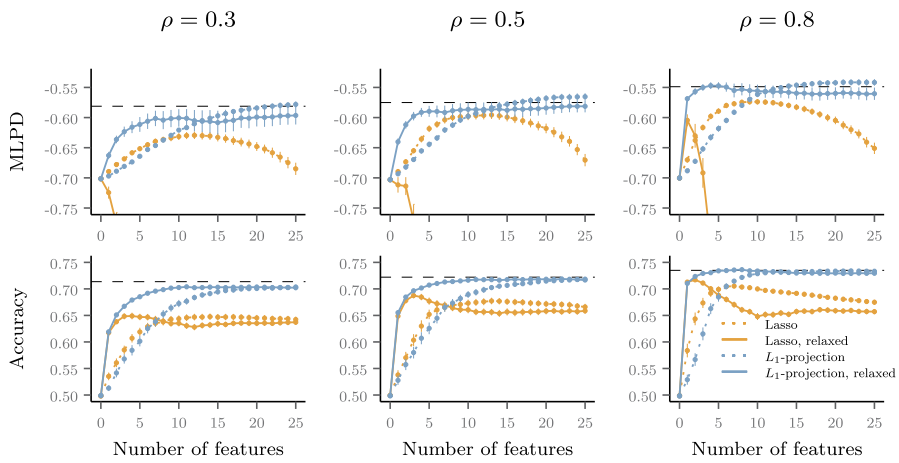

FIG 11. *Benefit of reference model, classification: Results analogous to those in Figure 7 but for the classification data. Here shown are MLPD and classification accuracy on test data.*

TABLE 2

Average number of features selected for the different methods in the microarray examples over the ten outer cross-validation folds. The last column denotes average over all data sets. The projection methods select by the most parsimonious models. The sparsest other method with similar predictive accuracy (Lasso with λ_{best}, see Fig. 8) selects on average over twice as many features as the most dense projection (Proj best-1se-reg).

Method	Ovarian	Colon	Prostate	Leukemia	Glioma	Average
Bayes SPC	633.0	881.0	7.7	1030.4	736.2	657.7
Proj ref-1se	3.3	2.2	2.9	8.6	1.6	3.7
Proj ref-1se-reg	7.2	3.9	3.8	11.0	7.2	6.6
Proj best-1se	3.1	2.1	2.9	16.2	1.6	5.2
Proj best-1se-reg	7.2	4.3	4.6	18.8	6.2	8.2
Lasso, λ_{1se}	11.7	5.1	12.7	15.3	9.2	10.8
Lasso, λ_{best}	18.6	11.1	23.6	23.2	25.0	20.3
Enet $\alpha = 0.7$, λ_{1se}	22.5	8.1	23.2	25.9	23.7	20.7
Enet $\alpha = 0.7$, λ_{best}	42.6	21.3	49.3	50.6	56.1	44.0
Enet $\alpha = 0.3$, λ_{1se}	57.7	24.8	53.6	74.5	74.1	56.9
Enet $\alpha = 0.3$, λ_{best}	114.3	64.2	124.8	188.0	187.7	135.8
Ridge, λ_{1se}	1536	2000	5966	7129	22283	7782.8
Ridge, λ_{best}	1536	2000	5966	7129	22283	7782.8

most parsimonious models.

B. Proofs of the theoretical results

B.1. Proof of Lemma 1

First plug in the definitions (26) into the formula of the expected gain (27) and expand

$$
\begin{aligned}
nG &= n\left(\Delta(\hat{\boldsymbol{\beta}}) - \Delta(\boldsymbol{\beta}_*)\right) \\
&= \mathrm{E}_{\tilde{\mathbf{y}}}\Big(\,||\mathbf{X}\hat{\boldsymbol{\beta}} - \tilde{\mathbf{y}}||^2\,\Big) - \mathrm{E}_{\tilde{\mathbf{y}}}\big(\,||\mathbf{X}\boldsymbol{\beta}_* - \tilde{\mathbf{y}}||^2\,\big) \\
&= \hat{\boldsymbol{\beta}}^\mathsf{T}\mathbf{X}^\mathsf{T}\mathbf{X}\hat{\boldsymbol{\beta}} - 2\mathrm{E}(\,\tilde{\mathbf{y}}\,)^\mathsf{T}\mathbf{X}\hat{\boldsymbol{\beta}} + \mathrm{E}\big(\,\tilde{\mathbf{y}}^\mathsf{T}\tilde{\mathbf{y}}\,\big) - \boldsymbol{\beta}_*^\mathsf{T}\mathbf{X}^\mathsf{T}\mathbf{X}\boldsymbol{\beta}_* + 2\mathrm{E}(\,\tilde{\mathbf{y}}\,)^\mathsf{T}\mathbf{X}\boldsymbol{\beta}_* \\
&\quad - \mathrm{E}\big(\,\tilde{\mathbf{y}}^\mathsf{T}\tilde{\mathbf{y}}\,\big) \\
&= \mathbf{y}^\mathsf{T}\mathbf{X}(\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}\mathbf{y} - 2\mathrm{E}(\,\tilde{\mathbf{y}}\,)^\mathsf{T}\mathbf{X}(\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}\mathbf{y} \\
&\quad - \boldsymbol{\mu}_*^\mathsf{T}\mathbf{X}(\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}\boldsymbol{\mu}_* + 2\mathrm{E}(\,\tilde{\mathbf{y}}\,)^\mathsf{T}\mathbf{X}(\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}\boldsymbol{\mu}_*.
\end{aligned}
$$

By plugging in $\mathrm{E}(\,\tilde{\mathbf{y}}\,) = \boldsymbol{\mu}$ and the definition $\mathbf{P} = \mathbf{X}(\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}$, we get

$$
\begin{aligned}
nG &= \mathbf{y}^\mathsf{T}\mathbf{P}\mathbf{y} - 2\boldsymbol{\mu}^\mathsf{T}\mathbf{P}\mathbf{y} - \boldsymbol{\mu}_*^\mathsf{T}\mathbf{P}\boldsymbol{\mu}_* + 2\boldsymbol{\mu}^\mathsf{T}\mathbf{P}\boldsymbol{\mu}_* \\
&= \mathbf{y}^\mathsf{T}\mathbf{P}\mathbf{y} - 2\boldsymbol{\mu}^\mathsf{T}\mathbf{P}\mathbf{y} - \boldsymbol{\mu}_*^\mathsf{T}\mathbf{P}\boldsymbol{\mu}_* + 2\boldsymbol{\mu}^\mathsf{T}\mathbf{P}\boldsymbol{\mu}_* - \boldsymbol{\mu}^\mathsf{T}\mathbf{P}\boldsymbol{\mu} + \boldsymbol{\mu}^\mathsf{T}\mathbf{P}\boldsymbol{\mu} \\
&= (\mathbf{y} - \boldsymbol{\mu})^\mathsf{T}\mathbf{P}(\mathbf{y} - \boldsymbol{\mu}) - (\boldsymbol{\mu}_* - \boldsymbol{\mu})^\mathsf{T}\mathbf{P}(\boldsymbol{\mu}_* - \boldsymbol{\mu}) \\
&= ||\mathbf{y} - \boldsymbol{\mu}||_\mathbf{P}^2 - ||\boldsymbol{\mu}_* - \boldsymbol{\mu}||_\mathbf{P}^2.
\end{aligned}
$$

Hence $G = \frac{1}{n}\left(||\mathbf{y} - \boldsymbol{\mu}||_\mathbf{P}^2 - ||\boldsymbol{\mu}_* - \boldsymbol{\mu}||_\mathbf{P}^2\right)$.

B.2. Proof of Theorem 2

By Lemma 1, $G = \frac{1}{n}\left(||\boldsymbol{\varepsilon}||_\mathbf{P}^2 - ||\boldsymbol{\mu}_* - \boldsymbol{\mu}||_\mathbf{P}^2\right)$, where $\boldsymbol{\varepsilon} = \mathbf{y} - \boldsymbol{\mu}$. Taking the expectation of G with respect to the $\boldsymbol{\varepsilon}$ as well as the randomness in the reference fit $\boldsymbol{\mu}_*$ yields

$$
\begin{aligned}
\mathrm{E}(\,G\,) &= \frac{1}{n}\left(\mathrm{E}\big(\,\boldsymbol{\varepsilon}^\mathsf{T}\mathbf{P}\boldsymbol{\varepsilon}\,\big) - \mathrm{E}\big(\,(\boldsymbol{\mu}_* - \boldsymbol{\mu})^\mathsf{T}\mathbf{P}(\boldsymbol{\mu}_* - \boldsymbol{\mu})\,\big)\right) \\
&= \frac{1}{n}\Big(\mathrm{Tr}(\mathbf{P}\mathrm{Cov}(\boldsymbol{\varepsilon})) - \mathrm{Tr}(\mathbf{P}\mathrm{Cov}(\boldsymbol{\mu}_* - \boldsymbol{\mu})) - \mathrm{E}(\,\boldsymbol{\mu}_* - \boldsymbol{\mu}\,)^\mathsf{T}\mathbf{P}\mathrm{E}(\,\boldsymbol{\mu}_* - \boldsymbol{\mu}\,)\Big) \\
&= \frac{1}{n}\left(\sigma^2\mathrm{Tr}(\mathbf{P}) - \mathrm{Tr}(\mathbf{P}\mathbf{K}) - \mathbf{b}^\mathsf{T}\mathbf{P}\mathbf{b}\right).
\end{aligned}
$$

Now we have

$$
\mathrm{Tr}(\mathbf{P}) = \mathrm{Tr}\big(\mathbf{X}(\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}\big) = \mathrm{Tr}\big(\mathbf{X}^\mathsf{T}\mathbf{X}(\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\big) = \mathrm{Tr}(\mathbf{I}_p) = p,
$$

so the expected gain simplifies to

$$
\mathrm{E}(\,G\,) = \frac{1}{n}\left(\sigma^2 p - \mathrm{Tr}(\mathbf{P}\mathbf{K}) - ||\mathbf{b}||_\mathbf{P}^2\right). \tag{28}
$$

B.3. Proof of Corollary 2.1

If the errors in the reference model are uncorrelated, that is, $\mathbf{K} = \sigma_{\mu_*}^2\mathbf{I}$, the expected gain (28) reduces to

$$
\begin{aligned}
\mathrm{E}(\,G\,) &= \frac{1}{n}\left(\sigma^2 p - \sigma_{\mu_*}^2 p - ||\mathbf{b}||_\mathbf{P}^2\right) \\
&= \frac{p}{n}\left(\sigma^2 - \sigma_{\mu_*}^2 - \frac{1}{p}||\mathbf{b}||_\mathbf{P}^2\right).
\end{aligned}
$$

References

Afrabandpey, H., Peltola, T., Piironen, J., Vehtari, A. and Kaski, S. (2019). Making Bayesian predictive models interpretable: a decision theoretic approach. *arXiv:1910.09358*.

Ambroise, C. and McLachlan, G. J. (2002). Selection bias in gene extraction on the basis of microarray gene-expression data. *Proceedings of the National Academy of Sciences* **99** 6562-6566.

Armagan, A., Clyde, M. and Dunson, D. B. (2011). Generalized beta mixtures of Gaussians. In *Advances in Neural Information Processing Systems 24* (J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira and K. Q. Weinberger, eds.) 523–531.

Bair, E., Hastie, T., Paul, D. and Tibshirani, R. (2006). Prediction by supervised principal components. *Journal of the American Statistical Association* **101** 119–137. MR2252436

Barbieri, M. M. and Berger, J. O. (2004). Optimal predictive model selection. *The Annals of Statistics* **32** 870–897. MR2065192

Bernardo, J. M. and Juárez, M. A. (2003). Intrinsic Estimation. In *Bayesian Statistics 7* (J. M. Bernardo, M. J. Bayarri, J. O. Berger, A. P. Dawid, D. Heckerman, A. F. M. Smith and M. West, eds.) 465–476. Oxford University Press. MR2003190

Bernardo, J. M. and Smith, A. F. M. (1994). *Bayesian Theory*. John Wiley & Sons. MR1274699

Bhadra, A., Datta, J., Polson, N. G. and Willard, B. (2017). The horseshoe+ estimator of ultra-sparse signals. *Bayesian Analysis* **12** 1105–1131. MR3724980

Bhattacharya, A., Pati, D., Pillai, N. S. and Dunson, D. B. (2015). Dirichlet-Laplace priors for optimal shrinkage. *Journal of the American Statistical Association* **110** 1479–1490. MR3449048

Breiman, L. (1995). Better subset regression using the nonnegative garrote. *Technometrics* **37** 373–384. MR1365720

Bucilă, C., Caruana, R. and Niculescu-Mizil, A. (2006). Model compression. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '06* 535–541. ACM.

Bürkner, P.-C. (2017). brms: An R Package for Bayesian Multilevel Models Using Stan. *Journal of Statistical Software* **80** 1–28.

Candes, E. and Tao, T. (2007). The Dantzig selector: statistical estimation when p is much larger than n. *The Annals of Statistics* **35** 2313–2351. MR2382644

Carvalho, C. M., Polson, N. G. and Scott, J. G. (2009). Handling sparsity via the horseshoe. In *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics* (D. van Dyk and M. Welling, eds.). *Proceedings of Machine Learning Research* **5** 73–80.

Carvalho, C. M., Polson, N. G. and Scott, J. G. (2010). The horseshoe estimator for sparse signals. *Biometrika* **97** 465–480. MR2650751

Castillo, I. and van der Vaart, A. (2012). Needles and straws in a haystack:

posterior concentration for possibly sparse sequences. *The Annals of Statistics* **40** 2069–2101. MR3059077

Cawley, G. C. and Talbot, N. L. C. (2010). On over-fitting in model selection and subsequent selection bias in performance evaluation. *Journal of Machine Learning Research* **11** 2079–2107. MR2678023

Dupuis, J. A. and Robert, C. P. (2003). Variable selection in qualitative models via an entropic explanatory power. *Journal of Statistical Planning and Inference* **111** 77–94. MR1955873

Efron, B. (2010). *Large-scale inference: empirical Bayes methods for estimation, testing, and prediction. Institute of Mathematical Statistics (IMS) Monographs* **1**. Cambridge University Press. MR2724758

Efron, B., Hastie, T., Johnstone, I. and Tibshirani, R. (2004). Least angle regression. *The Annals of Statistics* **32** 407–499. MR2060166

Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association* **96** 1348–1360. MR1946581

Fan, J. and Lv, J. (2008). Sure independence screening for ultrahigh dimensional feature space. *Journal of the Royal Statistical Society. Series B (Methodological)* **70** 849–911. MR2530322

Friedman, J., Hastie, T. and Tibshirani, R. (2010). Regularization Paths for Generalized Linear Models via Coordinate Descent. *Journal of Statistical Software* **33**.

Gabry, J., Simpson, D., Vehtari, A., Betancourt, M. and Gelman, A. (2018). Visualization in Bayesian workflow. *Journal of the Royal Statistical Society. Series A* **182** 389–402. MR3902665

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A. and Rubin, D. B. (2013). *Bayesian Data Analysis*, third ed. Chapman & Hall. MR3235677

George, E. I. and McCulloch, R. E. (1993). Variable selection via Gibbs sampling. *Journal of the American Statistical Association* **88** 881–889.

Goodrich, B., Gabry, J., Ali, I. and Brilleman, S. (2018). rstanarm: Bayesian applied regression modeling via Stan. R package version 2.17.4.

Goutis, C. and Robert, C. P. (1998). Model choice in generalised linear models: A Bayesian approach via Kullback–Leibler projections. *Biometrika* **85** 29–37. MR1627250

Hahn, P. R. and Carvalho, C. M. (2015). Decoupling shrinkage and selection in Bayesian linear models: a posterior summary perspective. *Journal of the American Statistical Association* **110** 435–448. MR3338514

Harrell, F. E. (2015). *Regression modeling strategies: with applications to linear models, logistic and ordinal regression, and survival analysis*, second ed. Springer.

Hastie, T., Tibshirani, R. and Friedman, J. (2009). *The Elements of Statistical Learning*, second ed. Springer-Verlag. MR2722294

Hastie, T., Tibshirani, R. and Wainwright, M. (2015). *Statistical learning with sparsity: the Lasso and generalizations*. Chapman & Hall. MR3616141

Hernández-Lobato, D., Hernández-Lobato, J. M. and Suárez, A.

(2010). Expectation propagation for microarray data classification. *Pattern Recognition Letters* **31** 1618–1626.

Hinton, G., Vinyals, O. and Dean, J. (2015). Distilling the knowledge in a neural network. *arXiv:1503.02531*.

Ishwaran, H., Kogalur, U. B. and Rao, J. S. (2010). spikeslab: Prediction and variable selection using spike and slab regression. *The R Journal* **2** 68–73.

Ishwaran, H. and Rao, J. S. (2005). Spike and slab variable selection: frequentist and Bayesian strategies. *The Annals of Statistics* **33** 730–773. MR2163158

Johnson, V. E. and Rossell, D. (2012). Bayesian model selection in high-dimensional settings. *Journal of the American Statistical Association* **107** 649–660. MR2980074

Johnstone, I. M. and Silverman, B. W. (2004). Needles and straw in haystacks: empirical Bayes estimates of possibly sparse sequences. *The Annals of Statistics* **32** 1594–1649. MR2089135

Lee, K. E., Sha, N., Dougherty, E. R., Vannucci, M. and Mallick, B. K. (2003). Gene selection: a Bayesian variable selection approach. *Bioinformatics* **19** 90–97.

Li, Y., Campbell, C. and Tipping, M. (2002). Bayesian automatic relevance determination algorithms for classifying gene expression data. *Bioinformatics* **18** 1332–1339. MR3577380

Lindley, D. V. (1968). The choice of variables in multiple regression. *Journal of the Royal Statistical Society. Series B (Methodological)* **30** 31–66. MR0231492

McCullagh, P. and Nelder, J. A. (1989). *Generalized linear models*, second ed. *Monographs on Statistics and Applied Probability*. Chapman & Hall. MR3223057

Meinshausen, N. (2007). Relaxed Lasso. *Computational Statistics & Data Analysis* **52** 374–393. MR2409990

Narisetty, N. N. and He, X. (2014). Bayesian variable selection with shrinking and diffusing priors. *The Annals of Statistics* **42** 789–817. MR3210987

Neal, R. and Zhang, J. (2006). High dimensional classification with Bayesian neural networks and Dirichlet diffusion trees. In *Feature Extraction, Foundations and Applications* (I. Guyon, S. Gunn, M. Nikravesh and L. A. Zadeh, eds.) 265–296. Springer.

Nott, D. J. and Leng, C. (2010). Bayesian projection approaches to variable selection in generalized linear models. *Computational Statistics and Data Analysis* **54** 3227–3241. MR2727748

Paananen, T., Piironen, J., Bürkner, P.-C. and Vehtari, A. (2020). Implicitly adaptive importance sampling. *arXiv:1906.08850*.

Paul, D., Bair, E., Hastie, T. and Tibshirani, R. (2008). "Preconditioning" for feature selection and regression in high-dimensional problems. *The Annals of Statistics* **36** 1595–1618. MR2435449

Peltola, T. (2018). Local interpretable model-agnostic explanations of Bayesian predictive models via Kullback-Leibler projections. In *Proceedings of the 2nd Workshop on Explainable Artificial Intelligence* (D. W. Aha, T. Darrell, P. Doherty and D. Magazzeni, eds.) 114–118.

Peltola, T., Havulinna, A. S., Salomaa, V. and Vehtari, A. (2014).

Hierarchical Bayesian survival analysis and projective covariate selection in cardiovascular event risk prediction. In *Proceedings of the 11th UAI Bayesian Modeling Applications Workshop. CEUR Workshop Proceedings* **1218** 79–88.

Piironen, J. and Vehtari, A. (2016). Projection predictive model selection for Gaussian processes. In *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)* 1–6. IEEE.

Piironen, J. and Vehtari, A. (2017a). Comparison of Bayesian predictive methods for model selection. *Statistics and Computing* **27** 711–735. MR3613594

Piironen, J. and Vehtari, A. (2017b). Sparsity information and regularization in the horseshoe and other shrinkage priors. *Electronic Journal of Statistics* **11** 5018–5051. MR3738204

Piironen, J. and Vehtari, A. (2017c). On the hyperprior choice for the global shrinkage parameter in the horseshoe prior. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics* (A. Singh and J. Zhu, eds.). *Proceedings of Machine Learning Research* **54** 905–913.

Piironen, J. and Vehtari, A. (2018). Iterative supervised principal components. In *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics* (A. Storkey and F. Perez-Cruz, eds.). *Proceedings of Machine Learning Research* **84** 106–114.

Polson, N. G. and Scott, J. G. (2011). Shrink globally, act locally: sparse Bayesian regularization and prediction. In *Bayesian statistics 9* (J. M. Bernardo, M. J. Bayarri, J. O. Berger, A. P. Dawid, D. Heckerman, A. F. M. Smith and M. West, eds.) 501–538. Oxford University Press, Oxford. MR3204017

Raftery, A. E., Madigan, D. and Hoeting, J. A. (1997). Bayesian model averaging for linear regression models. *Journal of the American Statistical Association* **92** 179–191. MR1436107

Reid, S., Tibshirani, R. and Friedman, J. (2016). A study of error variance estimation in Lasso regression. *Statistica Sinica* **26** 35–67. MR3468344

Reunanen, J. (2003). Overfitting in making comparisons between variable selection methods. *Journal of Machine Learning Research* **3** 1371–1382.

Ribeiro, M. T., Singh, S. and Guestrin, C. (2016). "Why should I trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '16* 1135–1144. ACM.

Snelson, E. and Ghahramani, Z. (2005). Compact approximations to Bayesian predictive distributions. In *Proceedings of the 22nd International Conference on Machine Learning. ICML '05* 840–847. ACM.

Stan Development Team (2018). Stan modeling language users guide and reference manual, Version 2.18.0.

Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)* **58** 267–288. MR1379242

Tran, M.-N., Nott, D. J. and Leng, C. (2012). The predictive Lasso. *Statistics and Computing* **22** 1069–1084. MR2950086

van der Pas, S. L., Kleijn, B. J. K. and van der Vaart, A. W. (2014). The horseshoe estimator: posterior concentration around nearly black vectors. *Electronic Journal of Statistics* **8** 2585–2618. MR3285877

Vehtari, A., Gelman, A. and Gabry, J. (2017). Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. *Statistics and Computing* **27** 1413–1432. MR3647105

Vehtari, A. and Ojanen, J. (2012). A survey of Bayesian predictive methods for model assessment, selection and comparison. *Statistics Surveys* **6** 142–228. MR3011074

Vehtari, A., Simpson, D., Gelman, A., Yao, Y. and Gabry, J. (2019). Pareto smoothed importance sampling. *arXiv:1507.02646*.

Yao, Y., Vehtari, A., Simpson, D. and Gelman, A. (2018). Using stacking to average Bayesian predictive distributions (with discussion). *Bayesian Analysis* **13** 917–1003. MR3853125

Zanella, G. and Roberts, G. (2019). Scalable importance tempering and Bayesian variable selection. *Journal of the Royal Statistical Society. Series B (Methodological)* **81** 489–517. MR3961496

Zou, H. (2006). The adaptive Lasso and its oracle properties. *Journal of the American Statistical Association* **101** 1418–1429. MR2279469

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society. Series B (Methodological)* **67** 301–320. MR2137327