Predictive State Smoothing (PRESS): Scalable non-parametric regression for high-dimensional data with variable selection

Georg M. Goerg
gmg@google.com

Google Inc.

June 3, 2017

Abstract

We introduce *predictive state smoothing (PRESS)*, a novel semi-parametric regression technique for high-dimensional data using predictive state representations. PRESS is a fully probabilistic model for the optimal kernel smoothing matrix. We present efficient algorithms for the joint estimation of the state space as well as the non-linear mapping of observations to predictive states and as an alternative algorithms to minimize leave-one-out cross validation error. The proposed estimator is straightforward to implement using (stochastic) gradient descent and scales well for large N and large p. LASSO penalty parameters as well the optimal smoothness can be estimated as part of the optimization. Finally we show that out-of-sample predictions are on par with or better than alternative state-of-the-art regression methods on the abalone and MNIST benchmark datasets. Yet unlike alternative methods PRESS gives meaningful domain-specific insights and can be used for statistical inference via regression coefficients.

Keywords: kernel regression, predictive states, LOOCV optimization, non-parametric smoothing, variable selection, high-dimensional data, minimal sufficient statistics, non-parametric dimension reduction, distribution clustering.

1 Introduction

Consider the high-dimensional regression problem

$$y = f(\mathbf{x}) + u, \quad u \overset{iid}{\sim} G(0, \sigma^2), \tag{1}$$

where p-dimensional features $\mathbf{x} \in \mathbb{R}^p$ are mapped to a continuous noisy $y \in \mathbb{R}$ through an unknown function $f : \mathbb{R}^p \to \mathbb{R}$ and G is some well-behaved error distribution (not necessarily Normal). The statistical and machine learning literature covers parametric, semi-parametric, and non-parametric models for f, algorithms to estimate f, or to directly optimize probabilistic or point predictions, $P(y \mid \mathbf{x})$ or $\mathbb{E}(y \mid \mathbf{x})$, respectively. References on specific models are too many to list, but any book on statistical learning contains most common techniques (e.g., Murphy, 2012; Hastie et al., 2001; Bishop, 2006). Statistical regression models can easily be used for further inference and as building blocks in generative models, but fall short in predictive performance compared to machine learning approaches. The latter, however, might be difficult or even impossible to use as part of a proper probabilistic inference and calculus as they are often solely defined in terms of an optimization algorithm. The method we propose achieves predictive performance en par with state-of-the-art machine learning approaches such as Random Forests, SVMs, or neural nets; yet it is a generative model with probabilistic predictions and hence estimates are easy to interpret using parametric inference familiar from logistic regression.

To achieve this we put regression in a predictive state framework, introduced in Lauritzen (1974b) and further developed in the causal state (Shalizi and Crutchfield, 2001; Shalizi, 2003) as well as the sufficient dimension reduction literature (Wang and Xia, 2008; Cook and Li, 2002). This approach reverses the usual procedure to define a statistical model first, and then analyze it for estimation and inference. Instead, Lauritzen (1974b) suggest to rather let the model be informed by the planned statistical analysis. The main objective for the regression problem in (1) often is not to estimate f directly, but most applications rather call for *optimal* predictions $P(y \mid \mathbf{x})$ or $\mathbb{E}(y \mid \mathbf{x})$. In order to obtain such optimal predictions we borrow from the causal state literature on non-linear time series and dynamic systems (Shalizi and Crutchfield, 2001; Shalizi and Shalizi, 2004; Boots et al., 2013) and define a function ϵ that maps features $\mathbf{x} \in \mathcal{X}$ to a predictive state space $\mathcal{S}$. This state space is constructed in such a way that it is *minimal sufficient* to predict y (see also *adequate* statistics in Lauritzen, 1974a; Dawid, 1979). It makes y independent of $\mathbf{x}$ given its state, $P(y \mid \mathbf{x}, \epsilon(\mathbf{x})) = P(y \mid \epsilon(\mathbf{x}))$, and it does so in way that achieves maximal compression of $\mathbf{x}$ while not losing any information to predict y. Following this approach naturally leads to a smoothing method that estimates a generative model for the optimal kernel matrix from the data, which *by construction* is minimal sufficient for prediction. It is related to sliced inverse regression (SIR) (Li, 1991; Wang and Xia, 2008) and mean subspace dimension reduction (Cook and Li, 2002); however, our proposed method is non-linear, works for

univariate $\mathbf{x}$ as well, and we propose a joint maximum likelihood estimator for the subspace and the mapping rather than iterative EM-like algorithms.

This work is organized as follows. Section 2 presents the predictive state framework and adapts it to the general high-dimensional, non-parametric regression setting. We also establish the connection to linear smoothers, metric learning methods, sliced inverse regression and sufficient dimension reduction, and reproducing kernel Hilbert spaces (RKHS). Section 3 explains how to obtain probabilistic and point predictions for new unseen data. In Section 4 we present efficient algorithms for maximum likelihood estimation (MLE) of the state space $\mathcal{S}$ and the mapping ϵ. As a useful alternative the closed-form leave-one-out and generalized cross-validation (LOOCV and GCV) MSE can be computed efficiently from the training data alone. LASSO penalty parameters as well as the optimal smoothness of the function f can also be estimated automatically from the data. In Section 5 we apply PRESS to the motorcycle, abalone, and MNIST dataset and show that it does not only match or even outperform state-of-the-art prediction methods, but can also be used statistical inference via an interpretable state space. Moreover PRESS models have a much lower-dimensional parameter space compared to deep neural nets or random forests. Finally, Section 6 summarizes the methodology and highlights its main advantages over existing regression and smoothing methods. Proofs and derivations are given in Appendix A.

2 Predictive States for Regression

Predictive state representations are statistically and computationally efficient for obtaining optimal forecasts of non-linear dynamical systems (Shalizi and Crutchfield, 2001). Examples include time series forecasting via ϵ-machines (Shalizi and Shalizi, 2004), learning non-linear dynamics of spatial fields (Jänicke and Scheuermann, 2009), and signal processing for artificial intelligence, e.g., moving robot arms or modeling car trajectories (Boots et al., 2013). In a nutshell, any observation X_t at time t has a corresponding latent predictive state S_t, which is minimal sufficient to predict the future X_{t+1}, i.e., $P(X_{t+1} \mid X_t, S_t) = P(X_{t+1} \mid S_t)$. Consequently all X_t with the same state $S_t = s$ share the same conditional predictive distribution $P(X_{t+1} \mid S_t = s)$. Goerg and Shalizi (2013) and Boots et al. (2013) propose algorithms for the non-parametric estimation of the continuous state space as well as the mapping from X_t to S_t.

In a regression problem one is usually concerned with obtaining estimates for $P(y \mid \mathbf{x})$ or $\mathbb{E}(y \mid \mathbf{x})$ if only a point prediction is required. Following causal state literature (Shalizi and Crutchfield, 2001) and borrowing notation and terminology from Goerg and Shalizi (2013)

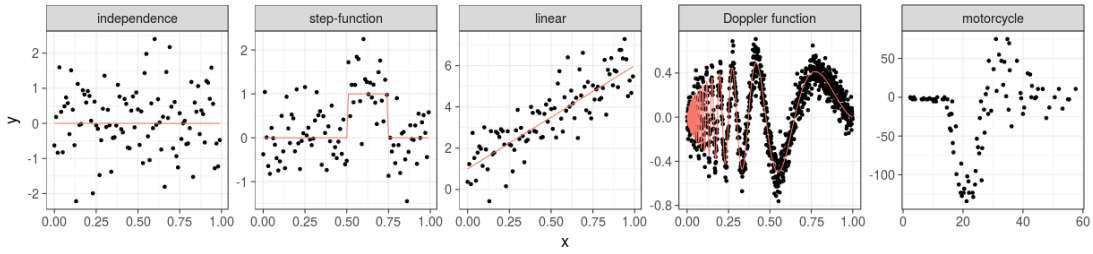


Figure 1: Univariate regression examples discussed in Section 2.1.

we introduce a latent predictive state random variable $S \in \mathcal{S}$, which is a function of $\mathbf{x}$,

$$\epsilon : \mathcal{X} \mapsto \mathcal{S}, \quad S = \epsilon(\mathbf{x}). \tag{2}$$

The function ϵ is such that it maps $\mathbf{x}$ to an equivalence class $[\mathbf{x}]$, which consists of all points in $\mathcal{X}$ that have the same predictive distribution $P(y \mid \mathbf{x})$ as $\mathbf{x}$. Formally,

$$\epsilon : \mathbf{x} \mapsto [\mathbf{x}] = \{\tilde{\mathbf{x}} \mid P(y \mid \mathbf{x}) \equiv P(y \mid \tilde{\mathbf{x}})\}. \tag{3}$$

The set $[\mathbf{x}]$ is non-empty as it contains at least $\mathbf{x}$ itself. It is important to point out that two observations $\mathbf{x}_{i_1} \neq \mathbf{x}_{i_2}$ can lie in the same state, s, even if $\mathbf{x}_{i_1}$ and $\mathbf{x}_{i_2}$ are very distinct from each other, i.e., they do not have to lie close in $\mathcal{X}$ as long as they predict the same y. Put in other words, it is advantageous to find similar observations in the conditional distribution space $p(y \mid \mathbf{x})$, rather than just locally in the Euclidean space of $\mathbf{x}$.

Lemma 2.1 (Sufficiency)**.** *$\epsilon(\mathbf{x})$ is sufficient to predict y from $\mathbf{x}$,*

$$P(y \mid \epsilon(\mathbf{x}), \mathbf{x}) = P(y \mid \epsilon(\mathbf{x})). \tag{4}$$

Corollary 2.2 (Minimal sufficiency)**.** *$\epsilon(\mathbf{x})$ is* minimal sufficient *to predict y from $\mathbf{x}$.*

See also Lauritzen (1974a) and Dawid (1979) on an *adequate* statistic.

Lemma 2.1 and Corollary 2.2 are key for prediction and estimation as

$$P(y \mid \mathbf{x}) = \int_{s \in \mathcal{S}} P(y \mid s, \mathbf{x}) P(s \mid \mathbf{x}) \mathrm{d}s = \int_{s \in \mathcal{S}} P(y \mid s) P(s \mid \mathbf{x}) \mathrm{d}s, \tag{5}$$

where the second equality follows from conditional independence of y and $\mathbf{x}$ given the (minimal) sufficient $\epsilon(\mathbf{x})$.

2.1 Predictive state examples

For sake of illustration we present several examples of a true f with their corresponding predictive state space. Figure 1 shows random draws for several examples ($u \overset{iid}{\sim} N(0, \sigma^2)$ and $x_i \sim U(0,1)$) plus the motorcycle dataset we analyze in Section 5.1.

independent y and x: The true model is

$$y = u, \quad \sigma = 1, \tag{6}$$

which implies that y is independent of $\mathbf{x}$. Hence $\mathcal{S} = \{s\}$ is just one state that contains the entire observation space $s = \{\mathbf{x} \mid \mathbf{x} \in \mathbb{R}^p\}$. ϵ is a constant, mapping every $\mathbf{x}$ to s. The best prediction in mean squared error (MSE) sense is $\mathbb{E}(y \mid s)$, which can be estimated using the sample average $\frac{1}{N}\sum_{i=1}^{N} y_i$.

step functions: The step function

$$y = \mathbf{1}\left(0.5 < x < 0.75\right) + u, \quad \sigma = 0.5 \tag{7}$$

has two predictive states partitioning the observation space in

$$s_1 = \{x \mid x \leq 0.5 \text{ or } x \geq 0.75\} \text{ and } s_2 = \{x \mid 0.5 < x < 0.75\}. \tag{8}$$

The ϵ mapping needs to learn the inequality restrictions defining each set in (8). The MSE-minimizing predictions for any $\tilde{x} \in \mathbb{R}$

$$\mathbb{E}(y \mid \tilde{x}) = \mathbb{E}(y \mid \epsilon(\tilde{x}) = s_j), \tag{9}$$

can be estimated using a sample average in each partition $\frac{1}{N_j}\sum_{i|\epsilon(x_i)=s_j} y_i$, where $N_j = \mid s_j \mid$ is the number of observations in state s_j or the *size* of state j.

linear regression: For a simple linear regression

$$y = \alpha + \beta \cdot x + u, \quad \sigma = 1, \tag{10}$$

ϵ is the identity function (assuming $\beta \neq 0$) as $[\mathbf{x}] = \{x\}$ consists only of itself with $P(y \mid x) = P(y \mid [\mathbf{x}]) = G(\alpha + \beta \cdot x, \sigma^2)$ (assuming G is a location family).

continuous, univariate: The Doppler function (see p. 77-78 of Wasserman, 2006)

$$f(x) = \sqrt{x(1-x)}\sin\left(\frac{2.1\pi}{x + 0.05}\right), \quad 0 \leq x \leq 1, \tag{11}$$

has an infinite state space $\mathcal{S} = \cup_y s_y$ with

$$s_y = \{x \mid f(x) = y\}. \tag{12}$$

As an application we consider the *motorcycle* dataset (Section 5.1), where the state space is a combination of discrete and continuous components.

continuous, multivariate f: The abalone dataset (see Section 5.2) contains observations of several sea shells features. The goal is to predict the number of rings $y > 0$ – which serves as a proxy for their age – from several covariates (e.g., diameter, weight, or height) collected in $\mathbf{X}$:

$$y_i = f(x_{i,1}, \ldots, x_{i,7}) + u_i. \tag{13}$$

Section 5.2 shows that in this context the (estimated) predictive states are even interpretable as $s_1 = \textit{infant}$, $s_2 = \textit{adult}$, and $s_3 = \textit{senescent}$.

We want to highlight that our proposed methodology with 3 states and 16 parameters total (2 logistic regressions with $1 + 7$ parameters each) achieves the same out-of-sample predictive performance as a 2 layer neural net with 10 nodes each ($\sim$ 200 parameters). A 4 state model estimate (24 parameters) outperforms the neural net.

The step-function example illustrates why the predictive state approach improves upon typical kernel methods. A standard kernel method does not pool observations from $x < 0.5$ and $x > 0.75$ to estimate the true $y = 0$. The predictive state approach does that and achieves smaller variance, while keeping bias the same, hence reducing MSE.

Put in other words, the crucial difference between typical kernel regression and PRESS is that the latter does not rely on geometry of the Euclidean $\mathcal{X}$ alone, but on the conditional distribution space $\mathcal{Y} \mid \mathcal{X}$. Hence points can be close in the PRESS framework, even if they are not close in $\mathcal{X}$. This is especially useful to reduce variance, while keeping bias the same since points are close in $\mathcal{Y}$ by construction.

2.2 Probabilistic predictive states

In real-world applications the unknown f is neither constant nor piecewise constant. We thus generalize discrete, deterministic states to a continuous probabilistic state space by convex combinations of a finite basis space $\mathcal{S} = \{s_1, \ldots, s_J\}$. Doing this naturally leads to our novel semi-parametric kernel regression method that we term *predictive state smoothing (PRESS)*.

For the remainder of this work we consider a finite $\mathcal{S} = \{s_1, \ldots, s_J\}$ with $1 \leq J < \infty$. As we will show $\mathcal{S}$ forms the basis of a continuous, probabilistic state space, where each s_j is

a deterministic or *extremal* state (see below). For finite J, (5) becomes

$$P(y \mid \mathbf{x}) = \sum_{j=1}^{J} P(y \mid s_j) P(s_j \mid \mathbf{x}) = \sum_{j=1}^{J} w_j(\mathbf{x}) \cdot P(y \mid s_j), \tag{14}$$

which is a finite mixture over state-conditional distributions with weights $w_j(\mathbf{x}) := P(s_j \mid \mathbf{x})$. Each weightvector $\mathbf{w}(\mathbf{x}) = (w_1(\mathbf{x}), \ldots, w_J(\mathbf{x}))$ lies in the J-dimensional probability simplex $\Delta^{(J)} := \{\mathbf{p} \in \mathbb{R}^J \mid p_j \geq 0 \text{ and } \sum_{j=1}^{J} p_j = 1\}$.

Probabilistic predictive state representation: Let $\mathbf{w} \in \Delta^{(J)}$, with $w_j = \mathbb{P}(S = s_j \mid \mathbf{x})$, $j = 1, \ldots, J$, be the probabilistic predictive state space representation of $\mathbf{x}$.

Each s_j can be represented as a deterministic mapping $\mathbf{w}^{(s_j)} = (0, \ldots, 0, 1, 0, \ldots, 0)$ with 1 in the j-th position and lies in the jth corner of the probability simplex $\Delta^{(J)}$. Any non-deterministic $\mathbf{w}$ is a convex combination of $\mathbf{w}^{(s_j)}$, $j = 1, \ldots, J$. A deterministic s_j cannot be represented as a convex combination of any other states – deterministic or probabilistic – since $s_j \neq s_k$ by definition. Following *extremal point models* (Lauritzen, 1974b; Lauritzen et al., 1984) we thus also refer to the deterministic s_j as an *extremal* state.

Let $\mathbf{y} = (y_1, \ldots, y_N)^\intercal \in \mathcal{Y} = \mathbb{R}^{N \times 1}$ be N real-valued observations that we want to predict using p features collected in the design matrix $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)^\intercal \in \mathcal{X} = \mathbb{R}^{N \times p}$, where $\mathbf{x}_i = (x_{i,1}, \ldots, x_{i,p})$ (for simplicity assume that $x_{\cdot,1}$ is the intercept). The probabilistic predictive states for $\mathbf{X} \in \mathbb{R}^{N \times p}$ can be represented as an $N \times J$ matrix $\mathbf{W}$ with

$$[0, 1] \ni \mathbf{W}_{i,j} = P(s_j \mid \mathbf{x}_i), \quad i = 1, \ldots, N; j = 1, \ldots, J, \tag{15}$$

with all elements being non-negative and rows adding up to 1. In particular, $\|\mathbf{W}\|_1 = \sum_{i=1}^{N} \sum_{j=1}^{J} \mathbf{W}_{i,j} = N$.

Notation: To avoid confusion, $\mathbf{w}_i$ refers to the i-th row of $\mathbf{W}$; w_j to the j-th element of $\mathbf{w}_i$; $\mathbf{W}_j$ to the j-th column of $\mathbf{W}$; and $\mathbf{W}_{i,j}$ to the (i, j) element of $\mathbf{W}$.

2.3 Interpreting the probabilistic predictive state space

For deterministic states the *size of state* s_j is the number of observations in each state

$$N_j = \sum_{i=1}^{N} \mathbb{1}\left(\epsilon(\mathbf{x}_i) = s_j\right), \tag{16}$$

where clearly $\sum_{j=1}^{J} N_j = N$. For probabilistic predictive states (16) can be generalized to the column sums of $\mathbf{W}$,

$$\sigma_j = \sum_{i=1}^{N} \mathbf{W}_{i,j} = \|\mathbf{W}_j\|_1, \tag{17}$$

which reduces to (16) if all $\mathbf{x}_i$ have deterministic mappings. Since $\|\mathbf{W}\|_1 = N$, it also holds that $\sum_{j=1}^{J} \sigma_j = N$ and hence σ_j can be interpreted as total number of (partial) observations credited to state j.

The J-dimensional $\mathbf{w}_i$ represents how far $\mathbf{x}_i$ lies from each of the J corners of the probability simplex $\Delta^{(J)}$ – the *extremal* states. For example, if $\mathbf{w}_i = (0.5, 0, \ldots, 0, 0.5)$, then observation i lies halfway between state s_1 and s_J with $P(y \mid \mathbf{x}_i) = \frac{1}{2}P(y \mid s_1) + \frac{1}{2}P(y \mid s_J)$. The rows of $\mathbf{W}$ induce a similarity between features $\mathbf{x}_i$ and $\mathbf{x}_j$, which can be used for dimension reduction, clustering, and visualization (see Fig. 5 in Section 5).

The function ϵ can vary between two extremes: either it is deterministic or it assigns states uniformly at random. Put differently, the mapping from observation to state can either be entirely certain or completely uncertain. As in Goerg and Shalizi (2013) we use Shannon entropy (Shannon, 1948)

$$\eta_i = \mathcal{H}(\mathbf{w}_i), \quad \text{where } \mathcal{H}(\mathbf{p}) = -\sum_{j=1}^{J} p_j \log_2 p_j, \tag{18}$$

to measure the uncertainty about the predictive state of observation $\mathbf{x}_i$. For a deterministic mapping $\eta_i = 0$; for a completely uninformative observation to state mapping, $w_j = \frac{1}{J}$ for all j, entropy achieves its maximum $\log_2(J)$. A low η_i does not imply though that the prediction per se, $P(y \mid \mathbf{x}_i)$, is uncertain; it just means one is sure about which of the J predictive distributions to pick from.

The states themselves are characterized by the equivalence classes in (3). With the probabilistic mapping $\mathbf{w}_i$ for each observation it is not anymore possible to consider exact equivalence classes. Instead state j is characterized by all $\mathbf{x}_i$ with probability $w_j = P(s_j = \epsilon(\mathbf{x}_i))$ falling into state s_j; i.e., states s_j, $j = 1, \ldots, J$ are characterized by the column space of $\mathbf{W}$. As $\mathcal{S}$ partitions the observation space $(\mathcal{Y}, \mathcal{X})$ we suggest to examine $p(y \mid \mathbf{W}_j)$ and $\mathbb{E}(\mathbf{X} \mid \mathbf{W}_j)$.[1] In real world applications, we propose to estimate summary statistics conditional on the j-th state, i.e., column j of $\mathbf{W}$, to gain a better understanding of what each $s_j \in \mathcal{S}$ represents. These will give *typical* predictions and features for the j-th predictive state – see for example Fig. 6a for the average handwritten digit of state $s_0, \ldots, s_9$ for the

[1] Since $\mathbf{X}$ is usually high-dimensional, it is less intuitive to consider $p(\mathbf{X} \mid \mathbf{W}_j)$ – especially for visualization when $p > 2$.

MNIST dataset. As an alternative we suggest to find that observation $\mathbf{x}_i^*$ with largest value in the j-th column. This serves as a representative *and* realistic example of state j. It has the advantage that such a $\mathbf{x}_i^*$ has been observed, whereas $\mathbb{E}(\mathbf{X} \mid \mathbf{W}_j)$ might not make sense for the domain-specific use case (akin to median vs. mean).

3 Prediction

The causal state literature has mostly focused on the characterization and estimation of the full predictive distribution $p(y \mid \epsilon(\mathbf{x}))$. For regression though, we are primarily concerned with estimating $\mathbb{E}(y \mid \mathbf{X})$ – which is a much simpler problem.

Just as the predictive distribution is a mixture over state-conditional distributions, so is the conditional expectation as – due to sufficiency of s_j –

$$\mathbb{E}(y \mid \mathbf{x}) = \sum_{j=1}^{J} P(S = s_j \mid \mathbf{x}) \mathbb{E}(y \mid s_j), \tag{19}$$

is a weighted average of state-conditional expectation.

Assume that both $\mathcal{S}$ and ϵ are known (see Section 4 for how to estimate them), i.e., we can map any new $\tilde{\mathbf{x}}$ to its state space representation $\tilde{\mathbf{w}} = \epsilon(\tilde{\mathbf{x}})$. The prediction in (19) can be estimated as

$$\hat{y}(\tilde{\mathbf{x}}) = \sum_{j=1}^{J} \tilde{w}_j(\tilde{\mathbf{x}}) \cdot \bar{y}^{(j)}, \quad \bar{y}^{(j)} = \frac{1}{\sigma_j} \sum_{i=1}^{N} \mathbf{W}_{i,j} \cdot y_i, \tag{20}$$

where the state-conditional point prediction $\bar{y}^{(j)}$ is a weighted average of observations in state j from observed (training) data $\{\mathbf{x}_i \mid i = 1, \ldots, N\}$. Eq. (20) holds for in-sample and out-of-sample predictions.

The in-sample fit can be written in matrix notation as

$$\mathbb{R}^N \ni \hat{\mathbf{y}} = \mathbf{S} \times \mathbf{y}, \tag{21}$$

$$\mathbb{R}^{N \times N} \ni \mathbf{S} := \mathbf{S}(\mathbf{W}) = \mathbf{W} \times \mathbf{D}(\mathbf{W}) \times \mathbf{W}^\intercal, \tag{22}$$

where $\mathbf{D}(\mathbf{W})$ is a diagonal matrix with $\mathbf{D}_{j,j} = \sigma_j^{-1}$. This shows that PRESS is just a linear smoother with kernel matrix $\mathbf{S}$ in (22).

Unlike traditional kernel smoothers, PRESS can rely on the kernel trick (Hofmann et al., 2008) and does not ever need to compute the $N \times N$ matrix explicitly, but predictions can be obtained in two steps with linear scaling in N. First, estimate J state-conditional point-predictions using a weighted average of $\mathbf{y}$ according to the re-normalized columns of

$\mathbf{W}$,

$$\mathbb{R}^{J\times 1} \ni \bar{\mathbf{y}}^{(1:J)} = \mathbf{D}(\mathbf{W}) \times \mathbf{W}^{\intercal} \times \mathbf{y}. \tag{23}$$

The prediction $\hat{y}_i$ can be obtained as a weighted average of the J point predictions, where the weight of state-prediction j equals the probability of observation i being in state s_j – which is just $\mathbf{w}_i$ –,

$$\mathbb{R}^{N\times 1} \ni \hat{\mathbf{y}} = \mathbf{W} \times \bar{\mathbf{y}}^{(1:J)}. \tag{24}$$

This property is essential for highly scalable and efficient implementations for prediction and estimation and distinguishes PRESS from traditional kernel smoothers, as they have to evaluate the full $N \times N$ matrix.

Another computational advantage is that once a model has been learned, only the J state point-predictions in (23) need to be stored for future (test) prediction. This is highly advantageous for large N datasets as usually $J \ll N$. For example, the MNIST handwritten digit dataset has $N_{train} = 60,000$, $N_{test} = 10,000$, but $J = 10$ (see Section 5.3).

3.1 Predictive distributions

While often a point-prediction and prediction intervals suffice, PRESS provides fully probabilistic predictions as

$$P(y \mid \tilde{\mathbf{x}}) = \sum_{j=1}^{J} \tilde{w}_j \cdot P(y \mid s_j). \tag{25}$$

The mixture components $P(y \mid s_j)$, $j = 1, \ldots, J$ can be estimated in parallel using a weighted non-parametric density estimator with weight of y_i proportional to $\mathbf{W}_{i,j}$.

4 Estimation

Section 3 showed how to obtain predictions $\hat{\mathbf{y}}$ from a weight matrix $\mathbf{W}$ and weigtvector $\tilde{\mathbf{w}}$, i.e., under the assumption that both $\mathcal{S}$ and ϵ are known. In this section we present estimators for $\mathcal{S}$ and ϵ given observations $(y_i, \mathbf{x}_i)$. In case the error distribution is Normal they are maximum likelihood estimators (MLEs).

Recall that our principal goal is to obtain optimal predictions of y given $\mathbf{x}$; estimating f in (1) is only secondary. As common we find the best model by minimizing the mean squared error (MSE). Following (21) & (22) the in-sample residuals are

$$\hat{\mathbf{u}} = \mathbf{y} - \hat{\mathbf{y}} = \mathbf{R}(\mathbf{W})\mathbf{y}, \tag{26}$$

where for better readability we define a residual operator

$$\mathbb{R}^{N\times N} \ni \mathbf{R}(\mathbf{W}) := \left(\mathbf{I}_N - \mathbf{W} \times \mathbf{D}(\mathbf{W}) \times \mathbf{W}^\intercal\right). \tag{27}$$

The MSE then becomes a quadratic form

$$\begin{aligned} \text{MSE}\left(\mathbf{W};\mathbf{y}\right) &= \frac{1}{N}\left(\mathbf{R}(\mathbf{W})\mathbf{y}\right)^\intercal\left(\mathbf{R}(\mathbf{W}))\mathbf{y}\right), \\ \text{subject to } \mathbf{w}_i &= \epsilon(\mathbf{x}_i), \quad i = 1,\ldots,N. \end{aligned} \tag{28}$$

Without any further specification of ϵ this is over-parametrized as $\mathbf{W}$ contains $(J-1)\cdot N$ free parameters; moreover optimizing (28) directly would not allows us to get weights for a new $\tilde{\mathbf{x}}$, but only for training data.

Since $\mathbf{w} \in \Delta^{(J)}$, we specify ϵ as the probability predictions of a multi-class classifier C_θ parametrized by θ, i.e., a softmax function,

$$\begin{aligned} C_\theta(\mathbf{x}) &\mapsto \{s_1,\ldots,s_J\}, \\ \epsilon(\mathbf{x}) &= (P(C_\theta(\mathbf{x}) = s_1),\ldots,P(C_\theta(\mathbf{x}) = s_J)). \end{aligned} \tag{29}$$

For example, for a logistic classifier $\theta = \{\boldsymbol{\beta}_j\}_{j=1}^J$ are p-dimensional coefficients for each class prediction $\mathbf{X}\cdot\boldsymbol{\beta}_j$; for a neural net, θ are node weights of all layers combined. As the softmax parametrization is unidentifiable for all J parameters, we add the restriction that the J coefficients related to the output layer must add up to $\mathbf{0} \in \mathbb{R}^p$. For logistic PRESS this means that a J state model, only requires estimating $\boldsymbol{\beta}_1,\ldots,\boldsymbol{\beta}_{J-1}$ as $\hat{\boldsymbol{\beta}}_J = -\sum_{j=1}^{J-1}\hat{\boldsymbol{\beta}}_j$. As state labels are invariant under permutation, we order the columns of $\mathbf{W}$ in increasing order of the state-conditional expectation $\mathbb{E}(y \mid \mathbf{W}_j)$ – which makes interpretation easier and keeps estimates consistent across re-runs with different initialization.[2]

The (i,j) element of $\mathbf{W}(\theta;\mathbf{X})$ is the predicted probability of C_θ for observation i being in class j. The resulting optimization problem (dropping the constant $\frac{1}{N}$)

$$\theta^* = \arg\min_\theta \left(\mathbf{R}(\mathbf{W}(\theta;\mathbf{X}))\mathbf{y}\right)^\intercal\left(\mathbf{R}(\mathbf{W}(\theta;\mathbf{X}))\mathbf{y}\right). \tag{30}$$

can be solved using (stochastic) gradient descent.[3] To avoid over-fitting θ can be tuned using a training vs. test split to optimize out of sample MSE.

We want to highlight that for the abalone dataset a simple logistic PRESS already provides excellent out-of-sample predictions. Adding hidden layers to obtain a deep PRESS variant did not further improve performance. This suggests that a wide net with just one softmax layer is enough for regression prediction, rather than a deep net. We will explore this in

[2] For some applications it might be more useful to order states by their size σ_j.
[3] We implement it in TensorFlow (Abadi et al., 2015).

future work with extensive simulation studies.

Maximum likelihood estimator (MLE): If $u \sim N(0, \sigma^2)$, then $\theta^* = \hat{\theta}_{MLE}$ is the maximum likelihood estimator (MLE) for θ.[4]

This joint optimization formulation and an ML estimate is one of our main statistical contributions compared to previous work on estimation in Goerg and Shalizi (2013), who propose an EM algorithm to estimate the state space and the mapping iteratively.

Sufficiency vs. minimal sufficiency: PRESS is not simply clustering y and then building a classifier to map to the clusters. Such an approach is not *minimal* sufficient to predict y. As an example consider a Bernoulli $y_i \in \{-1, 1\}$ with independent $\mathbf{x}_i$. Clustering in y-space leads to $J = 2$ clusters around $y = -1$ and $y = 1$. As $\mathbf{x}$ is independent of y, with enough training data a classifier will yield $\hat{\mathbf{w}}_i \approx \left(\frac{1}{2}, \frac{1}{2}\right)$ for each i and predictions are equal weight mixtures of $P(y \mid s_{-1}) = -1$ and $P(y \mid s_1) = 1$. However, there is a more compact representation of this data-generating process with $J = 1$ state, s, with $P(y \mid \mathbf{x}, s) = P(y \mid s) = \frac{1}{2}\mathbb{1}(y = -1) + \frac{1}{2}\mathbb{1}(y = 1)$. The ability to estimate such a minimal sufficient representation is a main advantage of PRESS compared to an iterative procedure that clusters first, and learns a classifier later.

4.1 Closed-form leave one out cross-validation (LOOCV)

Since PRESS is a linear smoother leave one out cross-validation (LOOCV) residuals and MSE can be computed in closed form based on training fit alone (see e.g., Wasserman, 2006), which greatly reduces computational complexity. LOOCV residuals are

$$\tilde{u}_i = \frac{u_i}{1 - s_{i,i}}, \quad u_i = y_i - \hat{y}_i, \tag{31}$$

where $s_{i,i}$ is the i-th diagonal entry of $\mathbf{S}$, and u_i is the i-th residual. The LOOCV MSE is

$$\text{MSE}^{(LOOCV)} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{u_i}{1 - s_{i,i}}\right)^2. \tag{32}$$

For any linear smoother the diagonal element $s_{i,i} \in [0, 1]$ measures the contribution of y_i to its own prediction $\hat{y}_i$: the closer $s_{i,i}$ to 1 the more the prediction relies on its own observed value, hence leaving it out would lead to a larger error during cross validation. Hence, minimizing (32) faces a trade-off between lower magnitude residuals u_i, yet not letting $s_{i,i}$ to get too close to 1 as then the $\text{MSE}^{(LOOCV)}$ diverges to infinity.

[4] Theoretical properties of the MLE in the PRESS setting are beyond the scope of this work.

Corollary 4.1 (LOOCV for PRESS)**.** *The i-th diagonal element of* $\mathbf{S}$ *equals*

$$s_{i,i} = \sum_{j=1}^{J} \frac{1}{\sigma_j} \cdot \mathbf{W}_{i,j}^2, \tag{33}$$

and $s_{i,i} \leq \sum_{j=1}^{J} \frac{1}{\sigma_j}$.

Proof. This follows directly from (22) and $\mathbf{D}$ being diagonal with $\mathbf{D}_{j,j} = \sigma_j^{-1}$. □

4.2 Smoothness, predictive manifold dimension, and generalized cross validation (GCV)

Computing $s_{i,i}$ for all $i = 1, \ldots, N$ might be prohibitive for large N – especially if this is used in every gradient descent step. In this case an alternative is to minimize *generalized* cross-validation (GCV) MSE

$$\text{MSE}^{(GCV)} = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{u_i}{1 - \frac{\nu}{N}} \right)^2 \tag{34}$$

where $\nu = \text{tr}(\mathbf{S}) = \sum_{i=1}^{N} s_{i,i}$ is the effective degrees of freedom of the smoother (see e.g., Wasserman, 2006). Instead of summing all $s_{i,i}$ from (33), ν can be computed more efficiently using the cyclic property of the trace operator.

Corollary 4.2. *The effective degrees of freedom* ν *of the PRESS smoother in* (22) *satisfies*

$$1 \leq \nu = \text{tr}\left(\mathbf{W} \times \mathbf{D}(\mathbf{W}) \times \mathbf{W}^\intercal\right) = \sum_{j=1}^{J} \frac{\|\mathbf{W}_j\|_2^2}{\|\mathbf{W}_j\|_1} \leq J. \tag{35}$$

$\nu = J$ if and only if all N state mappings are deterministic. $\nu = 1$ if and only if all N state assignments occur uniformly at random, i.e., $\mathbf{W}_{i,j} = \frac{1}{J}$ for all i and j.

The equality conditions are intuitive: a) $\nu = J$ re-iterates that there are J (deterministic) states; b) $\nu = 1$ means that there is *effectively* only one state and $\mathcal{S}$ is over-parametrized: if all states occur uniformly at random, then there is no point of having J states to begin with but $\mathcal{S}$ could be reduced to just 1 state, implying independence between $\mathbf{x}$ and y.

Corollary 4.2 suggests to use ν as a measure of smoothness: $\nu = 1$ gives a constant prediction, $\nu = J$ a step function with J levels, and for $1 < \nu < J$ the prediction function varies in between the two extremes. It is important to point out that ν is an inherent property of the predictive manifold describing $y \mid \mathbf{X}$. Hence in practice, as long as J is set large enough the estimated $\hat{\nu}$ should stay relatively stable for J greater than the (true, but unknown) ν

– similar to the "elbow" rule for PCA.

We can also view ν as a tuning parameter and let the user specify a target smoothness, ν_{smooth}, which can be incorporated in the optimization algorithm by adding a penalty of the form

$$\theta^* = \arg\min_{\theta} \text{LOOCV}(\boldsymbol{\theta}; \mathbf{X}) + \mu \cdot (\nu_{smooth} - \nu(\mathbf{W}(\theta; \mathbf{X}))^2, \tag{36}$$

where $\mu \geq 0$. We recommend to keep ν_{smooth} fairly close – but not equal – to J. If $\nu_{smooth} \ll J$ this suggests that the model is overparametrized and a smaller J should be used.

4.3 Model selection: choosing the number of states

In principle one could choose any of $J = 1, \ldots, N$ states. One state corresponds to independence and a simple sample average prediction for each i, $\hat{y}_i = \bar{\mathbf{y}}$. On the other extreme, N states mean that each observation is its own state – which gives perfect in-sample predictions, a trivial ϵ function as the identity, but infinite LOOCV MSE since $s_{i,i} = 1$ for all i. In practice the best J lies somewhere in between $1 \leq J^* < N$. Viewing it as a tuning parameter, we estimate J^* from the data.

While out-of-sample MSE is useful to avoid overfitting and parameter tuning, we observe that it is quite noisy for model selection. We notice that out-of-sample MSE stabilizes after a large enough J – supporting the proposition that there is a true ν for any predictive dependency $y \mid \mathbf{x}$ and one just has to choose J large enough. For model selection this means that choosing the best J according to minimum out-of-sample MSE is largely influenced by small variations due to noise in the out-of-sample estimate. Since PRESS is based on the principle of minimal sufficiency we should clearly favor smaller models over larger ones. We thus choose the best J^* according to minimum AIC, where

$$AIC(k) = N \cdot \log(MSE(k)) + 2 \cdot k, \tag{37}$$

where k is the number of free parameters. For logistic PRESS $k = (J-1) \cdot p$. We follow the suggestion of Sober (2002) and choose AIC over BIC as we are mainly interested in choosing the best model for prediction, rather than finding a *true* model.

Rather than trying every single $J \in \{1, \ldots, N\}$ we suggest to start with $J \approx \log N$ and monitor how far $\hat{\nu}$ is away from J. If $\hat{\nu} \ll J$ this suggest that J was too large to begin with and smaller J should be used; if $\hat{\nu} \approx J$ this suggest that J is too small and PRESS estimates the best step function (for a fixed, but too small J).

4.4 Variable selection

Even though PRESS is a kernel smoothing method, it is straightforward to include variable selection. For the MNIST dataset (Section 5.3) we use a LASSO (Tibshirani, 1994) penalty

$$PRESS_{LASSO}(\boldsymbol{\theta}) = LOOCV(\boldsymbol{\theta}) + \lambda \cdot \|\boldsymbol{\theta}_{\mathbf{X}}\|_1, \tag{38}$$

where $\boldsymbol{\theta}_{\mathbf{X}} \subseteq \boldsymbol{\theta}$ is the set of parameters that act directly on $\mathbf{x}$. For example, for logistic regression these are the non-intercept coefficients; for a neural net with several hidden layers $\boldsymbol{\theta}_{\mathbf{X}}$ are the weights from observations to the first hidden layer.

4.5 Predictive state estimation as a metric learning problem

The predictive state mapping ϵ can be viewed as a metric learner (Kulis, 2013), by reformulating (3) as

$$\epsilon : \mathbf{x} \mapsto \{\tilde{\mathbf{x}} \mid d_{\mathcal{X}}(\mathbf{x}, \tilde{\mathbf{x}}) = 0\}, \tag{39}$$

where

$$d_{\mathcal{X}}(\mathbf{x}, \tilde{\mathbf{x}}) = \begin{cases} 0, & \text{if } P(y \mid \tilde{\mathbf{x}}) = P(y \mid \mathbf{x}), \\ \tau > 0 & \text{otherwise.} \end{cases} \tag{40}$$

Metric learning approaches do not rely on an a-priori specified distance (or similarity) $d_{\mathcal{X}}(\mathbf{x}_i, \mathbf{x}_j)$, e.g., Euclidean, but learn the best metric for the task at hand. In that sense PRESS is closely related to Weinberger and Tesauro (2007), who estimate the optimal distance function $d_{\mathcal{X}}(\mathbf{x}_i, \mathbf{x}_j)$ in a Gaussian kernel regression $K\left(\frac{d_{\mathcal{X}}(\mathbf{x}_i, \mathbf{x}_j)}{h}\right)$ to minimize leave on out cross validation (LOOCV) error. However, unlike Weinberger and Tesauro (2007) we estimate ϵ – and hence the factorization of the kernel matrix – directly from $(\mathbf{y}, \mathbf{X})$ (see Section 4). We avoid the $N \times N$ evaluation of the kernel matrix on an estimated distance function, as the latter is merely a useful by-result using any metric or divergence $d_{\Delta^{(J)}}(\cdot, \cdot)$ on the probabilistic predictive state space $\mathbf{w} \in \Delta^{(J)}$. For example, information theoretic measures such as Kullback-Leibler (KL) or Jensen-Shannon (JS) divergence; or simply using cosine distance

$$d^{(cos)}_{\Delta^{(J)}}(\mathbf{w}_i, \mathbf{w}_j) = 1 - \frac{\langle \mathbf{w}_i, \mathbf{w}_j \rangle}{\|\mathbf{w}_i\|_2 \|\mathbf{w}_j\|_2}. \tag{41}$$

A zero distance implies that $\mathbf{x}_i$ and $\mathbf{x}_j$ have the same predictive distribution since their mixture weights are equal.

4.6 Step-functions, level sets, and eigen-approximations

By L2-normalizing columns, $\mathbf{v}_j := \mathbf{w}_j / \|\mathbf{w}_j\|_2$, and adjusting the diagonal matrix accordingly, $\lambda_{j,j} = \mathbf{D}_{j,j} \cdot \|\mathbf{w}_j\|_2^2$, (21) can be rewritten as

$$\hat{\mathbf{y}} = \mathbf{V} \times \mathbf{\Lambda}(\mathbf{V}) \times \mathbf{V}^\intercal \times \mathbf{y}, \quad \lambda_{k,k} = \frac{\|\mathbf{w}_j\|_2^2}{\sigma_j}, \tag{42}$$

which resembles an eigen-decomposition of $\mathbf{S}$. However, (42) is *not* a true eigen-decomposition since the columns of $\mathbf{V}$ are in general not orthogonal.

Lemma 4.3 (Eigen-state representation)**.** *The column vectors of* $\mathbf{V}$ *in* (42) *are orthogonal if and only if the state space is deterministic for each* $\mathbf{x}_i$*,* $i = 1, \ldots, N$*. In this case* $\mathbf{V}$ *only contains zeros and ones.*

Lemma 4.3 highlights another difference of PRESS to other eigen- and singular-value decomposition based methods, e.g., Laplacian clustering, diffusion maps, PCA, factors models. These approaches often take as input an estimated kernel matrix, $\mathbf{K} = K(\mathbf{X}; h) = K\left(\frac{d(\mathbf{x}_j, \mathbf{x}_i)}{h}\right)_{ij}$ evaluated on some pre-defined or estimated metric $d(\cdot, \cdot)$, and then compute exact eigenvectors of the observed matrix for clustering, dimension reduction, or regression. PRESS instead estimates the ϵ mapping from data $(\mathbf{y}, \mathbf{X})$, which approximates eigenvectors of the – optimally predictive, but unobserved – smoothing operator $\mathbf{S}$ in (22).

5 Applications

We apply PRESS to three benchmark datasets and demonstrate its usefulnessfor visualization (adding smooth lines to a scatterplot, dimension reduction), for prediction (en par or outperforming state-of-the-art regression methods), and last but not least for statistical inference and interpretation (MLE via logistic regression coefficients).

We implement the PRESS estimator from (30) in TensorFlow (Abadi et al., 2015).[5] Interestingly we found that adding hidden layers to ϵ id not improve predictions, suggesting that wide nets are sufficient for regression. We plan to investigate this in future work in more detail. For parameter tuning we optimize GCV (see Section 4.2) and use a true 80/20 hold-out split on the training data to determine an early stopping rule for the optimization algorithm: if the (20%) hold-out MSE has not decreased for more than 500 iterations, we stop the optimizer and pick the best model found so far. For model selection we use minimum AIC from (37).

For reproducibility and comparison to deep neural net predictions we use the training/test datasets provided on the online TensorFlow documentation. See each section for links to respective datasets.

[5] All other methods we show for comparison are based on the default options in their respective implementations in R (R Core Team, 2015).

5.1 Motorcycle dataset

The motorcycle dataset contains $N = 94$ measurements of head acceleration after impact ($t = 0$ milliseconds) in a simulated motorcycle accident for crash helmet testing. See Figure 2a for a scatterplot and several smooth fits. Here the true function f is continuous and the state space is unknown. However, given the domain specific context we can characterize at least one state as the "resting" state, $s_{resting} = \{t \mid f(t) = 0\}$, which can be approximated by examining again Figure 2a as

$$s_{resting} \approx \{t \mid 0 \leq t \leq 12, t \geq 45\}. \tag{43}$$

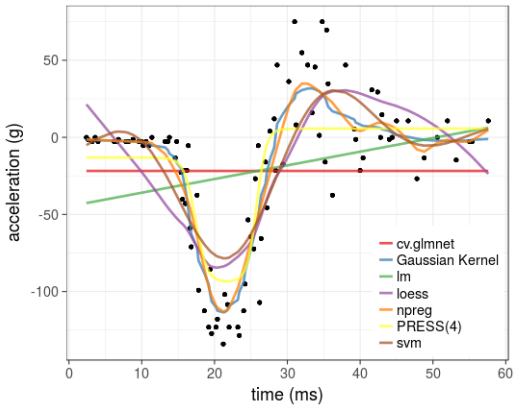
Figure 2b shows the Gaussian Kernel matrix with optimal bandwidth $h > 0$ according to LOOCV. While it fits the data very well (Fig. 2a), it could be improved in the beginning and end, i.e., for those t with $\epsilon(t) = s_{reting}$. The reason that usual kernel methods cannot use this information is that they rely on geometry of $\mathcal{X}$ alone, hence the smoothing matrix is always monotonically decreasing off the diagonal.

We estimate a series of PRESS models with varying number of states, $(J = 1, \ldots, 15)$ and pick the one with smallest AIC ($J^* = 4$, $\hat{\nu} = 2.705$). Figure 2c shows the minimal sufficient weights $\mathbf{w}_i$ and Fig. 2d the corresponding Kernel matrix $\hat{\mathbf{S}}$, which does show a distinct pattern illustrating that PRESS a) is prone to fitting step functions, and b) has an automatically adapting bandwidth (unlike a typical kernel smoother with a fixed h^* "optimal" bandwidth), and c) can pool observations far apart in $\mathcal{X}$ which keeps bias approximately the same but can greatly reduce variance (especially for $t < 12$).

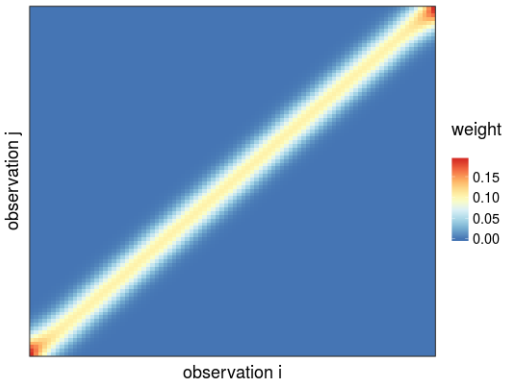
For this example competing methods such as a Gaussian kernel or loess smoothing provide better fits to the data than PRESS . We identified three potential reasons for this: a) the number of data points is fairly small; b) θ is currently initialized randomly, which could be improved by first clustering the data and use classifier estimates as starting values for the regression; and c) the conditional variance of y given $\mathbf{x}$ (and y given $\epsilon(\mathbf{x})$) is not constant; estimating the full conditional distribution might turn out helpful.

5.2 Abalone dataset

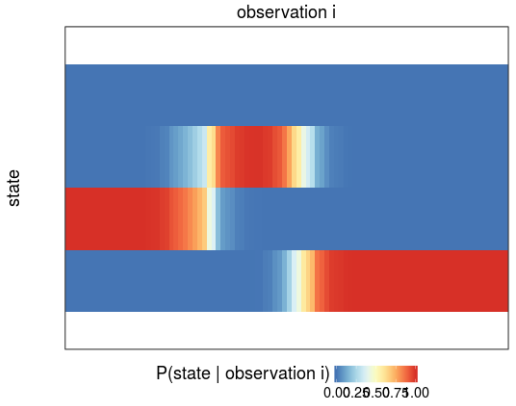
The abalone datset is a standard benchmark for regression and (ordered) classification methods and consists of $p = 7$ features (not including the categorical "sex" variable – we will revisit this later for interpretation of results) and $N_{train} = 3,320$ obervations ($N_{test} = 850$). The TensorFlow documentation contains an example of a 2-layer deep net with 10 nodes each. After $5,000$ iterations of fitting approximately 200 parameters it achieves an out-of-



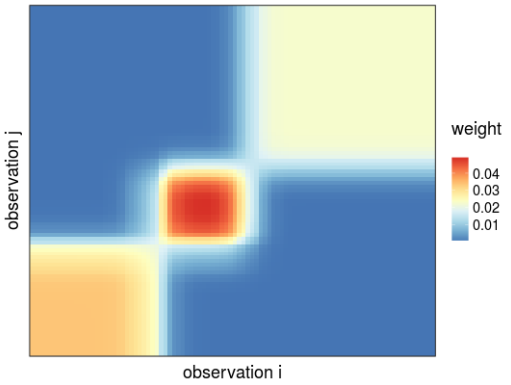
(a) Smoothing fits for acceleration as a function of time: $acceleration = f(t)$



(b) Gaussian Kernel with LOOCV bandwidth selection and degrees of freedom $df = 10.33$



(c) Probabilistic predictive state estimates $\hat{\mathbf{W}}$ (transposed with states in the rows and observations in columns so it matches up with coordinates above)



(d) PRESS Kernel estimate with $J = 4$ states and $\hat{\nu} = 2.71$ degrees of freedom.

Figure 2: Motorcycle dataset: comparison of several smoothing methods

sample test MSE of 5.581 (in-sample training MSE: 4.858).[6]

Again we fit PRESS for all $J = 1, \ldots, 15$ and pick the one with minimum AIC, $J^* = 10$. Additionally we also present $J = 3$ and $J = 4$ estimates: $J = 3$ estimate lends itself for visiualization as weightvectors $\mathbf{w}_i$ in the 3-dimensional probability simplex can be plotted in 2 dimensions; we want to relate the $J = 4$ state approximation to recent work by Golay et al. (2016) who suggests that the intrinsic fractal dimension of the abalone dataset lies between 3 and 4 (their estimate is $\hat{M}_2 = 3.66$).

Training PRESS in Tensorflow reaches a stable optimum after around $\sim$ 500 steps; the early stopping rule becomes active after $1,523$ iterations. Figure 3 summarizes the model

[6]We obtained data and MSE metrics from `https://www.tensorflow.org/extend/estimators` on May 12, 2017.

Table 1: Out-of-sample MSE for Abalone dataset. 'mse.rel.lm' is a normalized MSE relative to the linear model baseline.

method	mse	mad	mse.rel.lm	rmse
PRESS(10)	5.265	1.566	0.953	2.295
earth	5.266	1.592	0.953	2.295
svm	5.279	1.517	0.956	2.298
randomForest	5.335	1.567	0.966	2.310
PRESS(4)	5.413	1.615	0.980	2.326
lm	5.523	1.654	1	2.350
PRESS(3)	5.556	1.664	1.006	2.357
DNN(10x10)	5.581		1.010	2.362
cv.glmnet	5.642	1.665	1.021	2.375
mean	10.850	2.410	1.965	3.294

estimates and the resulting decomposition of the marginal distribution $p(y)$ into its mixture of predictive state distributions, $p(y \mid s_j)$. For $J = 3$ Table 2 presents coefficient estimates and state-conditional feature averages for easier interpretation for what state s_j represents. It is worthwhile to point out that PRESS only needs to fit J logistic regressions with p parameters each; hence it has a total of only $p \cdot (J - 1)$ parameters (minus 1 as we impose the $\sum_{j=1}^{J} \boldsymbol{\beta}_j \equiv \mathbf{0}_p$ identifiability condition for J logistic regressions). This means just 16 or 72 parameters, respectively, (much) less than a 10×10 deep neural net.

Apart from the deep neural net results, we also compare PRESS predictions to several state of the art regression methods such as SVM, RandomForest, or MARS. We use their respective R implementations without any particular tuning or feature engineering. Based on out-of-sample MSE a PRESS (3) model is as good as the deep neural net implementation. PRESS (10) has the best out-of-sample predictions amongst all considered methods with an MSE of 5.265.

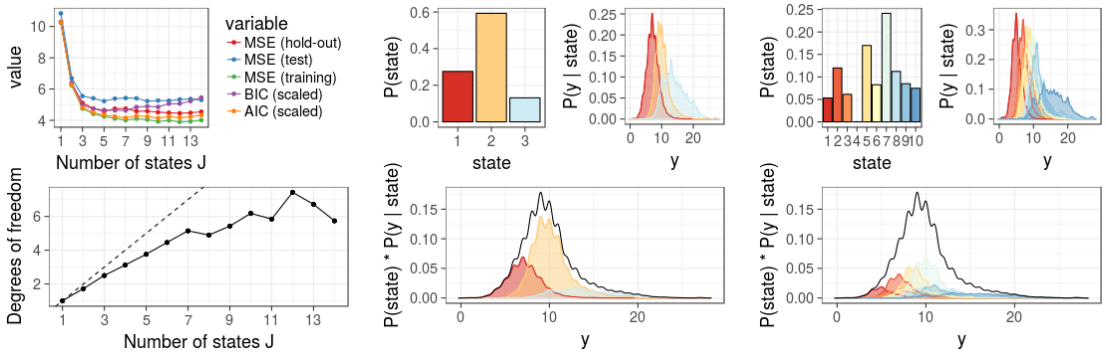
5.2.1 Interpreting the predictive state space for Abalone

One way to intepret 3 states in the lifetime of a shell is as $s_1 =$ "$infant''$, $s_2 =$ "$adult''$ (reproductive), and $s_3 =$ "$senescent''$ (non-reproductive) (see also Rogers-Bennet et al., 2007). This also clearly visible in the estimated predicted distributions in Figure 3a (Fig. 3b shows the $J = 10$ estimate for comparison). Figure 4 shows the corresponding Kernel smoothing matrices.

We can further explore the "infant" → "adult" → "senescent" state interpretations by looking at the categorical "sex" covariate – which contains an "infant", "female", and "male" category. We did not use this variable as a predictor, and also the TensorFlow

Table 2: Estimates for logistic PRESS with 3 predictive states: MLE for β and state-conditional average features.

	$\mathbb{E}(\mathbf{X})$	$\mathbb{E}(\mathbf{X} \mid s_1)$	$\mathbb{E}(\mathbf{X} \mid s_2)$	$\mathbb{E}(\mathbf{X} \mid s_3)$	$\hat{\boldsymbol{\beta}}_{s_1}$	$\hat{\boldsymbol{\beta}}_{s_2}$	$\hat{\boldsymbol{\beta}}_{s_3}$
intercept					-1.85	3.85	-2.00
length	0.52	0.38	0.57	0.59	3.32	-1.57	-1.75
diameter	0.41	0.29	0.45	0.47	-1.71	1.70	0.01
height	0.14	0.10	0.15	0.17	-1.73	-0.12	1.85
weight.w	0.82	0.32	0.99	1.16	-15.71	-4.40	20.11
weight.s	0.36	0.15	0.44	0.42	14.60	4.15	-18.75
weight.v	0.18	0.07	0.22	0.24	1.85	1.44	-3.29
weight.sh	0.24	0.09	0.28	0.38	-4.90	1.25	3.65



(a) Model selection (BIC and AIC have been scaled to match y-axis of MSE)

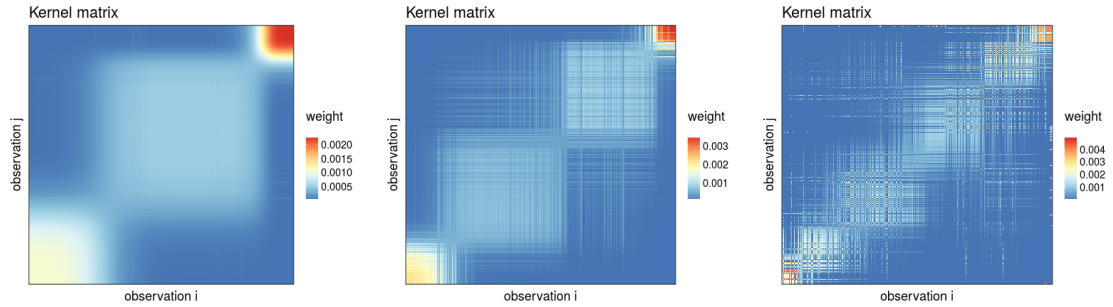
(b) $J = 3$ states ($\hat{\nu} = 2.51$)

(c) $J = 10$ states ($\hat{\nu} = 6.18$)

Figure 3: Model fit for Abalone dataset: model selection and estimates with marginal and conditional distribution for y, *state*, and $y \mid state$.

datasets do not contain this variable. Hence we obtained the original data from the UCI repository[7] and re-fit a 3 state model (again without the "sex" co-variate). Out-of-sample metrics and state-conditional distributions are essentially the same as for the TensorFlow datasets. Hence while the individual observations change, the aggregate model estimates and insights are comparable to the TensorFlow datasets. Figure 5 shows how the embedding space is related to the categorization of shells in "infant", "female", or "male". The left figure depicts the probability simplex $\Delta^{(3)}$, where each point in the scatterplot corresponds to the embedding space $\mathbf{w}_i = \epsilon(\mathbf{x}_i)$. The right figure shows the J conditional distribution $P(sex \mid s_j)$, which makes it clear that for predicting age it is important to distinguish between "infant" vs. "female / male", but not between "female" and "male". In fact, PRESS provides a better three-category variable for predicting age: the three predictive states s_1, s_2, and s_3.

[7] `https://archive.ics.uci.edu/ml/datasets/abalone`



(a) $J = 3$ states ($\hat{\nu} = 2.51$) **(b)** $J = 4$ states ($\hat{\nu} = 3.12$) **(c)** $J = 10$ states ($\hat{\nu} = 6.18$).

Figure 4: Optimal smoothing matrix for predicting abalone ring data. For better visualization rows and columns have been reordered according to increasing predicted number of rings from the 3 state model (since data is iid, the reordering does not change results).

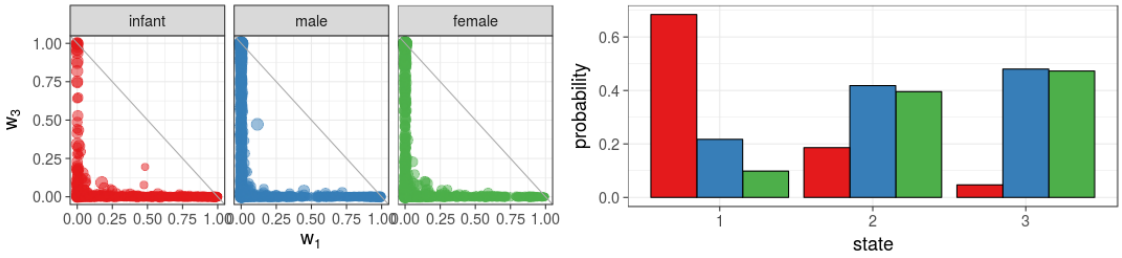


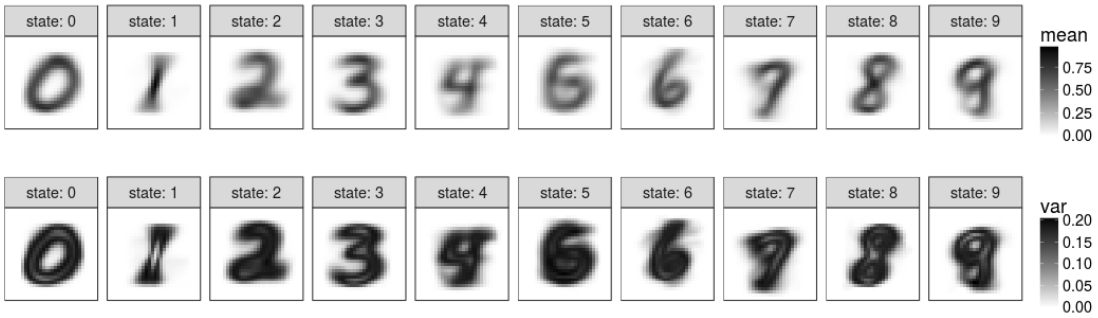
Figure 5: Predictive state space embedding of Abalone dataset for $J = 3$ states ($\hat{\nu} = 2.51$). Only w_1 and w_3 are shown since $w_2 = 1 - w_1 - w_3$ by definition.

5.3 MNIST dataset

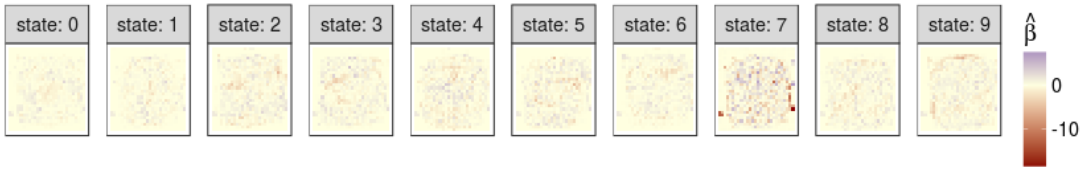
The MNIST dataset is commonly used for benchmarking classification methods. However, it is an ideal example for PRESS regression as well since: a) $J = 10$ digits is *known*; b) predictive state space summary statistics, $\mathbb{E}(\mathbf{X} \mid s_j)$ and $Var(\mathbf{X} \mid s_j)$, can be visualized and easily interpreted; c) it demonstrates the ability of variable selection for a large number of covariates $p = 784$ (pixels in a 28×28 image) and d) it proves to be easily scalable ($N_{train} = 60,000$, $N_{test} = 10,000$). Especially c) and d) are often computationally challenging for any traditional smoothing method or even SVMs with non-linear kernel function.

For training we again just use logistic regressions with J vectors, $\boldsymbol{\beta}_j \in \mathbb{R}^p$, where each entry of $\boldsymbol{\beta}_j$ corresponds to one pixel of the 28×28 image (plus intercept). We set $J = 10$ and $\nu_{smooth} = 10$ with a small $\mu > 0$ penalty parameter – since we *know* that the true function is a step function with a step at each digit $0, \ldots, 9$.

Figure 6 shows that PRESS recovers interpretable predictive states and the excellent quality of images suggests that predictions work well as well. Figure 6b shows the coefficient estimates for all 10 states. As coefficient estimates are quite heavy-tailed it distorts the color



(a) State-conditional summary statistics: $\mathbb{E}(\mathbf{X} \mid \mathbf{W}_j)$ and $Var(\mathbf{X} \mid \mathbf{W}_j)$



(b) Parameter estimates $\hat{\boldsymbol{\beta}}_j$

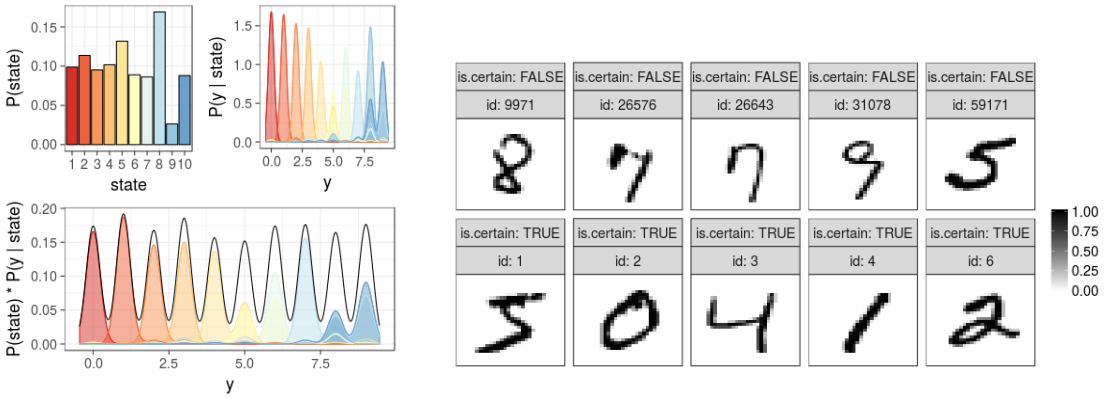
(c) Parameter estimates after heavy-tail removal using Lambert W × Gaussian transformation

Figure 6: Statistical inference for logistic PRESS model with $J = 10$ states for MNIST dataset

scale and does not allow much insight. We thus remove heavy tails from the coefficients using the bijective Lambert W × Gaussian transformation (Goerg, 2015) estimated from all coefficients jointly. The transformed parameters still have the same mean and keeps sign and monotonicity of coefficients, but do not have heavy tails anymore. Hence color coded images now do indeed show interesting patterns (Fig. 6c). The state conditional distributions $p(y \mid s_j)$ in Figure 7 show again that PRESS can correctly recover the best predictions as the conditional distributions peak around each digit, with small variation.

5.3.1 Intepreting the MNIST predictive state space

Predicting the value of a handwritten digit is easy if the image has a clear handwriting. We can revisit the measure of uncertainty η_i in (18) to rank images by how certain (low η_i) vs. uncertain (high η_i) their predictive state is. Figure 7b shows the top/bottom 5 images and indeed some of the images in the top row are even for a human hard to decipher.



(a) Minimal sufficient predictive estimates with marginal state and state-conditional predictive distributions

(b) Handwritten digits with high and low entropy η_i for digit $\rightarrow$ state mapping: top panel shows images with uncertain state mapping; bottom panel examples of images with certain state mapping

Figure 7: Prediction: Logistic PRESS for MNIST dataset

5.4 Discussion of application results

We want to emphasize that PRESS is fully probabilistic, able to predict and estimate for large N and p without running into the curse of dimensionality, useful for statistical inference as it is straightforward to intepret using regression coefficients on observed – not hidden – features, yet it outperforms highly complex models that are often hard to optimize, suffer from curse of dimensionality, or are difficult or impossible to interpret.

We plan to run extensive simulation studies to compare PRESS on a variety of datasets with tuned alternative methods. The applications suggest that PRESS is well-suited for high-dimensional regression. We also want to explore the options of adding hidden layers to PRESS – which has not yet proven to improve performance.

6 Summary & Discussion

In this work we introduce *predictive state smoothing (PRESS)*, a novel semi-parametric kernel regression method for high-dimensional data. PRESS is a metric learner, which determines that kernel function which gives the best predictions for $y \in \mathbb{R}$ given $\mathbf{x} \in \mathbb{R}^p$. It is not only statistically optimal in a theoretical sense, but also computationally efficient as prediction and estimation can rely on the kernel trick and thus compute predicted values linearly in N (instead of N^2 for typical kernel regression methods). The method also scales well in the number of variables and we propose a LASSO adaptation to perform variable selection for the $p \gg N$ case. We present algorithms for maximum likelihood estimation

as well as to optimize LOOCV MSE, which can be easily implemented in data flow graphs such as TensorFlow. PRESS compares well with state-of-the-art smoothing and regression methods, yet it remains interpretable and can be used for statistical inference as shown in the abalone and MNIST dataset.

6.1 Key properties

PRESS combines advantages of several popular regression techniques:

non-linear dependencies for optimal prediction: PRESS estimates the minimal sufficient statistic for predicting y from $\mathbf{x}$. The feature space is mapped onto an optimally predictive state space $\mathcal{S}$, which generates non-linear dependencies. Moreover, the mapping $\epsilon : \mathcal{X} \to \mathcal{S}$ can be non-linear as well, e.g., we use a multilayer neural net as one example of a deep PRESS .

linear smoother, fast cross-validation: It is linear in $\mathbf{y}$, which allows us to rely on theory and properties of linear smoothers. In particular, leave-one-out cross validation (LOOCV) can be computed on the training data only without actually running a proper cross-validation procedure.

variable selection: Being semi-parametric allows standard hypothesis testing and penalization techniques such as LASSO or Ridge. It is particularly noteworthy that it is a kernel smoother that can easily accommodate the $p \gg N$ case.

avoid curse of dimensionality: PRESS performs variable selection and tuning as part of the kernel matrix estimation, hence scales well with large p. Using the kernel trick it avoids the full $N \times N$ Kernel matrix computation for prediction and estimation. This is a major advantage compared to traditional kernel smoothers, which run into statistical and computational scaling issues for even moderate p or N.

mix of discrete and continuous covariates: PRESS can handle discrete and continuous covariates.

scalability for large N and large p: We present efficient algorithms based on stochastic gradient descent that can be easily implemented in data flow graphs such as TensorFlow. The proposed algorithm solve the joint optimization problem of findings the states and estimating $\epsilon : \mathcal{X} \to \mathcal{S}$ including variable selection.

family of smoothers: PRESS is a family of methods depending on the classifier used to model ϵ. In this work we use logistic PRESS and a variant of a deep PRESS .

Acknowledgments

I want to thank Christoph Best, Joseph Kelly, Jim Koehler, and Jing Kong for helpful feedback on this work; Hernan Moraldo, Pedro Nascimento, and Ashish Saxena for suggestions to improve the TensorFlow implementation; and Gabriel Singer for expert advice on the abalone dataset.

While the specific PRESS methodology and algorithms in this work were derived entirely while I was working at Google, I also want to thank Cosma Shalizi and Larry Wasserman for their insights, discussion, and advice in a previous, related collaboration on "Lebesgue Smoothing" (Goerg et al., 2012).

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. Software available from tensorflow.org.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Boots, B., Gordon, G. J., and Gretton, A. (2013). Hilbert space embeddings of predictive state representations. In Nicholson, A. and Smyth, P., editors, *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence, UAI 2013, Bellevue, WA, USA, August 11-15, 2013*. AUAI Press.

Cook, R. D. and Li, B. (2002). Dimension reduction for conditional mean in regression. *The Annals of Statistics*, 30(2):455–474.

Dawid, A. P. (1979). Conditional Independence in Statistical Theory. *Journal of the Royal Statistical Society. Series B (Methodological)*, 41(1):1–31.

Goerg, G. M. (2015). The Lambert Way to Gaussianize Heavy-Tailed Data with the Inverse of Tukeys h Transformation as a Special Case. *The Scientific World Journal*, 2015:1–16. doi:10.1155/2015/909231.

Goerg, G. M. and Shalizi, C. R. (2013). Mixed LICORS: A Nonparametric Algorithm for Predictive State Reconstruction. In *JMLR Workshop and Conference Proceedings*, volume 31. JMLR.org.

Goerg, G. M., Shalizi, C. R., and Wasserman, L. (2012). Lebesgue smoothing. Technical report, Carnegie Mellon University.

Golay, J., Leuenberger, M., and Kanevski, M. F. (2016). Feature selection for regression problems based on the morisita estimator of intrinsic dimension: Concept and case studies. *CoRR*, abs/1602.00216.

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA.

Hofmann, T., Schlkopf, B., and Smola, A. J. (2008). Kernel methods in machine learning. *The Annals of Statistics*, 36(3):1171–1220.

Jänicke, H. and Scheuermann, G. (2009). Knowledge Assisted Visualization: Steady Visualization of the Dynamics in Fluids Using epsilon-machines. *Comput. Graph.*, 33(5):597–606.

Kulis, B. (2013). Metric learning: A survey. *Foundations and Trends in Machine Learning*, 5(4):287–364.

Lauritzen, S. L. (1974a). On the Interrelationships Among Sufficiency, Total Sufficiency and Some Related Concepts. Technical report, Stanford University, Department of Statistics.

Lauritzen, S. L. (1974b). Sufficiency, prediction and extreme models. *Scandinavian Journal of Statistics*, 1(3):128–134.

Lauritzen, S. L., Barndorff-Nielsen, O. E., Dawid, A. P., Diaconis, P., and Johansen, S. (1984). Extreme point models in statistics [with discussion and reply]. *Scandinavian Journal of Statistics*, 11(2):65–91.

Li, K.-C. (1991). Sliced inverse regression for dimension reduction. *Journal of the American Statistical Association*, 86(414):316–327.

Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. The MIT Press.

R Core Team (2015). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Rogers-Bennet, L., Rogers, D. W., and Schultz, S. A. (2007). Modeling growth and mortality of red abalone (halliotis rufescens) in northern california. *Journal of Shellfish Research*, 26(3).

Shalizi, C. R. (2003). Optimal Nonlinear Prediction of Random Fields on Networks. In Morvan, M. and Rémila, É., editors, *Discrete Models for Complex Systems, DMCS'03*, volume AB of *DMTCS Proceedings*, pages 11–30. Discrete Mathematics and Theoretical Computer Science.

Shalizi, C. R. and Crutchfield, J. P. (2001). Computational mechanics: Pattern and prediction, structure and simplicity. *Journal of statistical physics*, 104(3):817–879.

Shalizi, C. R. and Shalizi, K. L. (2004). Blind Construction of Optimal Nonlinear Recursive Predictors for Discrete Sequences. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, UAI '04, pages 504–511, Arlington, Virginia, United States. AUAI Press.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell Systems Technical Journal*, 27:379–423,623–656.

Sober, E. (2002). Instrumentalism, parsimony, and the Akaike framework. *Philosophy of Science*, 69(S):112–123.

Tibshirani, R. (1994). Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288.

Wang, H. and Xia, Y. (2008). Sliced regression for dimension reduction. *Journal of the American Statistical Association*, 103(482):811–821.

Wasserman, L. (2006). *All of Nonparametric Statistics (Springer Texts in Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Weinberger, K. Q. and Tesauro, G. (2007). Metric learning for kernel regression. In Meila, M. and Shen, X., editors, *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics (AISTATS-07)*, volume 2, pages 612–619. Journal of Machine Learning Research - Proceedings Track.

A Appendix

A.1 Proofs

The proofs of Lemma 2.1 and Corollary 2.2 are completely analogous to proofs of Theorem 1 and 2 in Shalizi (2003). For sake of completeness we replicate the proofs for the regression setting. The interested reader is referred to Shalizi (2003) for several other important properties of predictive states for stochastic processes.

Proof of Lemma 2.1. The conditional distribution of y given state s is the average over all conditional distributions $P(y \mid \mathbf{x})$ for which $\mathbf{x}$ is in state s. Thus,

$$P(y \mid s) = \int_{\chi \in \epsilon^{-1}(s)} P(y \mid \mathbf{x} = \chi) \cdot P(\mathbf{x} = \chi \mid s) d\chi. \tag{44}$$

By construction, $P(y \mid \chi)$ is the same for all χ in the domain of the integral. Hence we can pick out a representative element in $\epsilon^{-1}(s)$, e.g., $\mathbf{x}$, and take it out of the integral

$$P(y \mid s = \epsilon(\mathbf{x})) = P(y \mid \mathbf{x}) \int_{\chi \in \epsilon^{-1}(s)} P(\mathbf{x} = \chi \mid s) d\chi \tag{45}$$

$$= P(y \mid \mathbf{x}), \tag{46}$$

where (46) holds as $P(\chi \mid s)$ is a proper probability distribution that integrates to 1. Hence $\epsilon(\mathbf{x})$ is sufficient to predict y.

□

Proof of Corollary 2.2. Assume there is another sufficient statistic $R = \eta(\mathbf{x}) \neq \epsilon(\mathbf{x})$. If we can show that there is always a function h which maps $h : R \mapsto S = \epsilon(\mathbf{x})$, then this implies that ϵ is minimal sufficient.

Since η is sufficient for predicting y, $\eta(\mathbf{x}) = \eta(\tilde{\mathbf{x}})$ if and only if $P(y \mid \tilde{\mathbf{x}}) = P(y \mid \mathbf{x})$. This implies that also $\epsilon(\mathbf{x}) = \epsilon(\tilde{\mathbf{x}})$. Thus all $\mathbf{x}$ with the same value of η also have the same $\epsilon(\mathbf{x})$, and thus S can be determined from R. Hence the required function h exists.

□

Proof of Lemma 4.3. If all $\mathbf{w}_i$ have this property, then it is straightforward to see that the crossproduct of column j_1 and j_2 is a sum of $1 \cdot 0$ and $0 \cdot 1$ terms for any (j_1, j_2) pair. Hence the vectors are orthogonal.

For the reverse direction assume the opposite is true, and there exists at least one $\mathbf{w}_i$ with a non-deterministic state mapping. That is there are at least two entries $0 < \mathbf{W}_{i,j_1}, \mathbf{W}_{i,j_2} < 1$ and $\mathbf{W}_{i,j_1} + \mathbf{W}_{i,j_2} = 1$. Without loss of generality say $i = 1$. The cross product between

column j_1 and j_2 equals

$$\langle \mathbf{W}_{j_1}, \mathbf{W}_{j_2} \rangle = \sum_{i=1}^{N} \mathbf{W}_{i,j_1} \mathbf{W}_{i,j_2} \geq \mathbf{W}_{1,j_1} \cdot \mathbf{W}_{1,j_2} > 0, \tag{47}$$

which contradicts the orthogonality assumption. The first inequality holds since $\mathbf{W}_{i,j} \geq 0$; the second strict inequality holds since by assumption the first observation has a non-zero probability of being in (at least) two states with probabilities $\mathbf{W}_{i,j_1} > 0$ and $\mathbf{W}_{i,j_2} > 0$, respectively. □

Proof of Corollary 4.2. The trace is invariant under cyclic permutations, i.e., $\mathrm{tr}\,(ABC) = \mathrm{tr}\,(CAB)$. Hence

$$\mathrm{tr}\,(\mathbf{S}) = \mathrm{tr}\left(\mathbf{W} \times \mathbf{D}(\mathbf{W}) \times \mathbf{W}^{\intercal}\right) = \mathrm{tr}\left(\mathbf{W}^{\intercal} \times \mathbf{W} \times \mathbf{D}(\mathbf{W})\right), \tag{48}$$

The matrix $\mathbf{D} \in \mathbb{R}J \times J$ is diagonal with $\mathbf{D}_{j,j} = \frac{1}{\sigma_j}$ and $\mathbf{W}^{\intercal} \times \mathbf{W} \in \mathbb{R}^{J \times J}$ has the squared ℓ_2 norm of $\mathbf{W}_j$ in its diagonal. Hence, (48) satisfies

$$\mathrm{tr}\left(\mathbf{W}^{\intercal} \times \mathbf{W} \times \mathbf{D}(\mathbf{W})\right) = \sum_{j=1}^{J} \frac{\|\mathbf{W}_j\|_2^2}{\|\mathbf{W}_j\|_1} \leq J, \tag{49}$$

since $\mathbf{W}_{i,j}^2 \leq \mathbf{W}_{i,j} \leq 1$ with equality if and only if each $\mathbf{w}_i$ is deterministic, since $\mathbf{W}_{i,j}^2 = \mathbf{W}_{i,j}$ if and only if $\mathbf{W}_{i,j} = 0$ or 1.

If states are deterministically obtained from $\mathbf{x}_i$, i.e., $\mathbf{W}_{i,j} = 0$ or 1, then

$$\mathrm{tr}\,(\mathbf{S}) = \sum_{j=1}^{J} \frac{\sum_{i=1}^{N} \mathbf{W}_{i,j}^2}{\sum_{i=1}^{N} \mathbf{W}_{i,j}} = \sum_{j=1}^{J} \frac{\sum_{i|\mathbf{W}_{i,j}=1} 1^2}{\sum_{i|\mathbf{W}_{i,j}=1} 1} = \sum_{j=1}^{J} 1 = J. \tag{50}$$

If each $\mathbf{x}_i$ is mapped to J states uniformly at random, $\mathbf{w}_i = (\frac{1}{J}, \ldots, \frac{1}{J})$, then

$$\mathrm{tr}\,(\mathbf{S}) = \sum_{j=1}^{J} \frac{\sum_{i=1}^{N} (1/J)^2}{\sum_{i=1}^{N} 1/J} = \sum_{j=1}^{J} \frac{N/J^2}{N/J} = \sum_{j=1}^{J} (1/J) = 1 \tag{51}$$

□