Online Learning for Prediction via Covariance Fitting: Computation, Performance and Robustness

Anonymous authors
Paper under double-blind review

Abstract

We consider the online learning of predictors based on a covariance model of the outcomes. The model parameters are often learned using cross-validation or maximum-likelihood techniques. However, neither technique is suitable when training data arrives in a streaming fashion. Here we consider a covariance-fitting method to learn the model parameters, which was initially developed for spectral estimation. We show that this results in a computationally efficient online learning method in which the resulting predictor can be updated sequentially. We prove that, with high probability, its out-of-sample error approaches the minimum achievable level at a root-n rate, where n is the number of data samples. Moreover, we show that the resulting predictor enjoys two robustness properties. First, it minimizes the out-of-sample error with respect to the least favourable distribution within a given Wasserstein distance from the empirical distribution. Second, it is robust against errors in the covariate training data. We illustrate the performance of the proposed method in a numerical experiment.

1 Introduction

We consider scenarios in which we observe a *stream* of randomly distributed data

$$\mathcal{D}_n = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}.$$

Given covariate $\mathbf{x}_{n+1}$ in $\mathcal{X}$, our goal is to predict the outcome y_{n+1} in a bounded range $\mathcal{Y} \subset \mathbb{R}$. A large class of predictors (also known as linear smoothers) can be described as a weighted combination of observed outcomes

$$\widehat{y}(\mathbf{x}) = \sum_{i=1}^{n} w_i(\mathbf{x}) y_i, \tag{1}$$

where $\mathbf{x}$ denotes any test point and the weights $\{w_i(\mathbf{x})\}$ are to be learned from $\mathcal{D}_n$. The sensitivity of such a predictor function to noise in the training data is often characterized by how closely $\widehat{y}(\mathbf{x}_i)$ is to y_i and quantified by its 'effective' degrees of freedom (Ruppert et al., 2003; Wasserman, 2006; Hastie et al., 2009):

$$0 < \ df_n \triangleq \sum_{i=1}^{n} w_i(\mathbf{x}_i) \ \leq n, \tag{2}$$

where $w_i(\mathbf{x}_i)$ is an in-sample weight. The effective degrees of freedom are often tuned to avoid overfitting the weights to the irreducible noise in the training data with the aim of achieving good out-of-sample performance. This is includes using distribution-free cross-validation or distribution-based maximum likelihood methods. However, these techniques do not readily work with streaming data.

In this paper, we consider an alternative method using a covariance-based criterion first proposed in the context of spectral estimation (Stoica et al., 2010a;b). We show that this method

- enables online tuning of the predictor,

- approaches an optimal out-of-sample performance at a root-n rate,
- enjoys to two different robustness properties.

For illustration of the online learning method, we include a numerical experiment.

2 Problem formulation

To determine the set of weights $\{w_i(\mathbf{x})\}$ in (1), y is often modeled as a zero-mean stochastic process with a specified covariance function (Bishop, 2006; Rasmussen & Williams, 2006; Stein, 2012). We consider a covariance function parameterized using bounded features $\{\phi_k(\mathbf{x})\}$ of $\mathbf{x}$:

$$\mathrm{Cov}[y, y'|\mathbf{x}, \mathbf{x}'; \boldsymbol{\lambda}] = \lambda_0 \delta(\mathbf{x}, \mathbf{x}') + \sum_{k=1}^{d} \lambda_k \phi_k(\mathbf{x}) \phi_k(\mathbf{x}'), \tag{3}$$

where $\mathbf{x}$ and $\mathbf{x}'$ are two arbitrary covariates and $\delta(\mathbf{x}, \mathbf{x}')$ is the Kronecker delta function. For compactness, we let $\boldsymbol{\lambda}$ denote the set of $d+1$ nonnegative covariance parameters and $\boldsymbol{\phi}(\mathbf{x}) = [\phi_1(\mathbf{x}), \ldots, \phi_d(\mathbf{x})]^\top$ (Ruppert et al., 2003; Rahimi et al., 2007; Stein, 2012; Hensman et al., 2017; Solin & Särkkä, 2020). Under this model, the minimum-mean-square-error predictor can be expressed as a linear smoother (1):

$$\widehat{y}(\mathbf{x}; \boldsymbol{\lambda}) = \boldsymbol{w}^\top(\mathbf{x}; \boldsymbol{\lambda}) \mathbf{y}, \quad \text{where } \mathbf{y} = [y_1, \ldots, y_n]^\top \tag{4}$$

and the vector of n weights is given by

$$\boldsymbol{w}(\mathbf{x}; \boldsymbol{\lambda}) = \mathbf{C}_{\boldsymbol{\lambda}}^{-1} \boldsymbol{\Phi} \boldsymbol{\Lambda} \boldsymbol{\phi}(\mathbf{x}) \tag{5}$$

and the covariance matrix

$$\mathbf{C}_{\boldsymbol{\lambda}} = \boldsymbol{\Phi} \boldsymbol{\Lambda} \boldsymbol{\Phi}^\top + \lambda_0 \mathbf{I}_n \succ \mathbf{0} \tag{6}$$

where

$$\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{\phi}^\top(\mathbf{x}_1) \\ \vdots \\ \boldsymbol{\phi}^\top(\mathbf{x}_n) \end{bmatrix}$$

is the matrix of features and $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_d)$. For any *fixed* $\boldsymbol{\lambda}$, the predictor (4) can be updated sequentially for each new point in a data stream.

The predictor function above includes a variety of penalized regression methods (see the references cited above). The degrees of freedom of (4) is controlled by $\boldsymbol{\lambda}$ and we have that (Stoica & Stanasila, 1982; Ruppert et al., 2003):

$$\begin{aligned} 0 < df_n(\boldsymbol{\lambda}) &= \mathrm{tr}\left\{ \boldsymbol{\Phi} \boldsymbol{\Lambda} \boldsymbol{\Phi}^\top \mathbf{C}_{\boldsymbol{\lambda}}^{-1} \right\} \\ &= \mathrm{tr}\left\{ \boldsymbol{\Phi} \boldsymbol{\Lambda} \boldsymbol{\Phi}^\top (\boldsymbol{\Phi} \boldsymbol{\Lambda} \boldsymbol{\Phi}^\top + \lambda_0 \mathbf{I}_n)^{-1} \right\} \leq \min(n, d) \end{aligned} \tag{7}$$

In principle, $\boldsymbol{\lambda}$ can be learned using cross-validation or maximum likelihood methods. However, these methods have problems in the online learning setting. First, for each additional training data point $(\mathbf{x}_{n+1}, y_{n+1})$, the parameters $\boldsymbol{\lambda}$ will need to be re-learned using augmented dataset $\mathcal{D}_{n+1}$ and (4) recomputed from scratch. Second, learning $\boldsymbol{\lambda}$ via these methods is a non-convex problem that can be riddled with multiple minima. Third, using data-dependent parameters $\boldsymbol{\lambda}$ in (4) does not readily provide any out-of-sample prediction performance guarantees.

With these issues in mind, we will investigate computational and theoretical properties of learning $\boldsymbol{\lambda}$ via a covariance-fitting approach.

3 Learning via covariance fitting

We propose to learn $\boldsymbol{\lambda}$ from $\mathcal{D}_n$ by fitting the model covariance matrix $\mathbf{C}_{\boldsymbol{\lambda}}$ in (5) to the empirical covariance matrix $\mathbf{y}\mathbf{y}^\top$. Specifically, we will use the following fitting criterion, known as SPICE,

$$\boxed{\boldsymbol{\lambda}^\circ = \underset{\boldsymbol{\lambda}\geq\mathbf{0}}{\arg\min}\ \left\|\mathbf{y}\mathbf{y}^\top - \mathbf{C}_{\boldsymbol{\lambda}}\right\|^2_{\mathbf{C}_{\boldsymbol{\lambda}}^{-1}},} \tag{8}$$

which was first proposed in the context of spectral estimation (Stoica et al., 2010a;b). Since this criterion is convex in $\boldsymbol{\lambda}$, we can be sure that a global minimizer can be determined.

3.1 Sequential computation

Let $\boldsymbol{\lambda}_n^\circ$ denote the learned parameters using $\mathcal{D}_n$, with an associated predictor function $\widehat{y}(\mathbf{x};\boldsymbol{\lambda}_n^\circ)$. Since data is obtained in an online manner, we wish to compute $\widehat{y}(\mathbf{x};\boldsymbol{\lambda}_{n+1}^\circ)$ sequentially, given a new training data point $(\mathbf{x}_{n+1}, y_{n+1})$.

Theorem 1 *The predictor function $\widehat{y}(\mathbf{x};\boldsymbol{\lambda}_{n+1}^\circ)$ can be updated from $\widehat{y}(\mathbf{x};\boldsymbol{\lambda}_n^\circ)$ in a constant runtime $\mathcal{O}\left(d^2\right)$. The total memory requirement of the method is also constant and in the order of $\mathcal{O}\left(d^2\right)$.*

Proof 1 *We first note that the predictor (4) has an equivalent form*

$$\widehat{y}(\mathbf{x};\boldsymbol{\lambda}) = \boldsymbol{\phi}^\top(\mathbf{x})\underbrace{\boldsymbol{\Lambda}\boldsymbol{\Phi}^\top\mathbf{C}_{\boldsymbol{\lambda}}^{-1}\mathbf{y}}_{\widehat{\boldsymbol{\theta}}(\boldsymbol{\lambda})} \tag{9}$$

For later use, we also note that (9) is invariant to a uniform rescaling of $\boldsymbol{\lambda}$, i.e., $\widehat{\boldsymbol{\theta}}(\boldsymbol{\lambda}) \equiv \widehat{\boldsymbol{\theta}}(c\boldsymbol{\lambda})$ for all $c > 0$.

Furthermore, using the matrix inversion lemma, it can be shown that $\widehat{\boldsymbol{\theta}}(\boldsymbol{\lambda})$ equals the minimizer of the following augmented criterion,

$$V(\boldsymbol{\theta},\boldsymbol{\lambda}) = \frac{1}{\lambda_0}\|\mathbf{y}-\boldsymbol{\Phi}\boldsymbol{\theta}\|_2^2 + \boldsymbol{\theta}^\top\boldsymbol{\Lambda}^{-1}\boldsymbol{\theta} + tr\{\mathbf{C}_{\boldsymbol{\lambda}}\}, \tag{10}$$

where the last term will be used below. It can also be shown that the minimizer of the concentrated criterion $V(\widehat{\boldsymbol{\theta}}(\boldsymbol{\lambda}),\boldsymbol{\lambda})$ equals $c\boldsymbol{\lambda}^\circ$, that is a rescaling of the covariance-based parameters (8). See Zachariah & Stoica (2015, Appendix A). Thus minimizing $V(\boldsymbol{\theta},\boldsymbol{\lambda})$ with respect to both $\boldsymbol{\theta}$ and $\boldsymbol{\lambda}$ yields $\widehat{y}(\mathbf{x};\boldsymbol{\lambda}^\circ) \equiv \boldsymbol{\phi}^\top(\mathbf{x})\widehat{\boldsymbol{\theta}}(\boldsymbol{\lambda}^\circ)$. By changing the order of the minimization, we obtain the solution

$$\lambda_k(\boldsymbol{\theta}) = \begin{cases} \frac{\|y-\boldsymbol{\Phi}\boldsymbol{\theta}\|_2}{\sqrt{n}}, & k=0 \\ \frac{|\theta_i|}{\sqrt{n}\psi_k}, & k=1,\ldots,d \end{cases} \tag{11}$$

where $\psi_k = \sqrt{\frac{1}{n}\sum_{i=1}^n \phi_k^2(\mathbf{x}_i)}$. Inserting (11) into (10) yields the following equivalent convex cost function

$$\underset{\boldsymbol{\theta}}{\arg\min}\ \sqrt{\frac{1}{n}\|\mathbf{y}-\boldsymbol{\Phi}\boldsymbol{\theta}\|_2^2} + \frac{1}{\sqrt{n}}\|\boldsymbol{\psi}\odot\boldsymbol{\theta}\|_1, \tag{12}$$

where $\boldsymbol{\psi} = [\psi_1\ \cdots\ \psi_d]^\top$ and $\odot$ denotes the element-wise product. This is a weighted square-root LASSO problem that can be solved in a runtime on the order of $\mathcal{O}(d^2)$ using variables of fixed dimension that are recursively updated. See Zachariah & Stoica (2015) for more details.

Solving (12) when data arrives in a streaming fashion requires storing the recursively updated quantities:

$$\mathbf{A}^n = \frac{1}{n}\sum_{i=1}^n \boldsymbol{\phi}(\mathbf{x}_i)\boldsymbol{\phi}^\top(\mathbf{x}_i), \quad \mathbf{b}^n = \frac{1}{n}\sum_{i=1}^n \boldsymbol{\phi}^\top(\mathbf{x}_i)y_i, \quad \kappa^n = \frac{1}{n}\sum_{i=1}^n y_i^2,$$

Thus the memory requirement is dominated by storing the $d\times d$-matrix $\mathbf{A}^n$.

For completeness, the code for computing $\widehat{y}(\mathbf{x};\boldsymbol{\lambda}^\circ)$ is provided at `TOAPPEAR`.

3.2 Out-of-sample performance

We have shown that $\widehat{y}(\mathbf{x};\boldsymbol{\lambda})$ in (4) with covariance-fitted parameters $\boldsymbol{\lambda}^\circ$ can be updated sequentially. While this determines the degrees of freedom (7) in an online manner, it remains to be investigated how covariance-fitting affects out-of-sample prediction performance as measured by the mean-squared error,

$$\textsc{Mse} = \mathbb{E}\left[(y - \widehat{y}(\mathbf{x}))^2\right], \tag{13}$$

where expectation is taken with respect to an unknown distribution $p(\mathbf{x}, y)$. That is, how closely we can predict y_{n+1} from $\mathbf{x}_{n+1}$ when they are drawn from $p(\mathbf{x}, y)$?

We first note that all predictors of the form (4) and (5) belong to the following class of predictor functions (see (9))

$$\mathcal{F} \triangleq \left\{ \; f(\mathbf{x}) = \sum_{k=1}^{d} \phi_k(\mathbf{x})\theta_k \; : \; \boldsymbol{\theta} \in \mathbb{R}^d \right\}, \tag{14}$$

where $\{\phi_k(\mathbf{x})\}$ are the features in (3). Next, we show how $\widehat{y}(\mathbf{x};\boldsymbol{\lambda}^\circ)$ performs as compared to the best predictor functions in $\mathcal{F}$.

Theorem 2 *If the data points $(\mathbf{x}_i, y_i)$ are drawn independently and identically (i.i.d.), then the out-of-sample error of $\widehat{y}(\mathbf{x};\boldsymbol{\lambda}^\circ)$ is given by*

$$\boxed{\mathbb{E}\left[(y - \widehat{y}(\mathbf{x};\boldsymbol{\lambda}^\circ))^2\right] \leq \min_{\widehat{y}\in\mathcal{F}} \mathbb{E}\left[(y - \widehat{y}(\mathbf{x}))^2\right] \; + \; K\sqrt{\frac{1}{n}\ln\frac{2(d+1)^2}{\varepsilon}} \; + \; b_n} \tag{15}$$

with probability of at least $1-\varepsilon$, where K is a constant and b_n is bounded as $\mathcal{O}(n^{-3/4})$. That is, with high probability, the out-of-sample error approaches the minimum achievable error at a root-n rate. Note that number of features d only increases the second term at a logarithmic rate.

Proof 2 *Let $\widehat{y}$ be any predictor in $\mathcal{F}$ and let $\mathcal{R}(\widehat{y}) = \mathbb{E}\left[(y - \widehat{y}(\mathbf{x}))^2\right]$. Then we can express the out-of-sample mean-square error in the following way:*

$$\mathcal{R}(\widehat{y}) \equiv \mathbb{E}\left[\left(y - \boldsymbol{\phi}^\top(\mathbf{x})\boldsymbol{\theta}\right)^2\right] = \mathbb{E}\left[\begin{bmatrix}\boldsymbol{\theta}\\ -1\end{bmatrix}^\top \underbrace{\begin{bmatrix}\boldsymbol{\phi}(\mathbf{x})\\ y\end{bmatrix}}_{\mathbf{z}} \begin{bmatrix}\boldsymbol{\phi}(\mathbf{x})\\ y\end{bmatrix}^\top \underbrace{\begin{bmatrix}\boldsymbol{\theta}\\ -1\end{bmatrix}}_{\widetilde{\boldsymbol{\theta}}}\right] = \widetilde{\boldsymbol{\theta}}^\top \boldsymbol{\Sigma} \widetilde{\boldsymbol{\theta}}, \tag{16}$$

where $\boldsymbol{\Sigma} = \mathbb{E}[\mathbf{z}\mathbf{z}^\top]$. Similarly, the in-sample error can be expressed as $\mathcal{R}_n(\widehat{y}) = \widetilde{\boldsymbol{\theta}}^\top \widehat{\boldsymbol{\Sigma}} \widetilde{\boldsymbol{\theta}}$, where $\widehat{\boldsymbol{\Sigma}} = n^{-1}(\mathbf{z}_1\mathbf{z}_1^\top + \cdots + \mathbf{z}_n\mathbf{z}_n^\top)$. The gap between in- and out-of-sample errors can be bounded as:

$$\begin{aligned} |\mathcal{R}_n(\widehat{y}) - \mathcal{R}(\widehat{y})| &= |\widetilde{\boldsymbol{\theta}}^\top(\widehat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma})\widetilde{\boldsymbol{\theta}}| \\ &\leq \sum_{i=1}^{d+1}\sum_{j=1}^{d+1} |\widetilde{\theta}_i\|\widetilde{\theta}_j| \, |\widehat{\boldsymbol{\Sigma}}_{ij} - \boldsymbol{\Sigma}_{ij}| \\ &\leq (\|\boldsymbol{\theta}\|_1 + 1)^2 \cdot \underbrace{\max_{i,j} |\widehat{\boldsymbol{\Sigma}}_{ij} - \boldsymbol{\Sigma}_{ij}|}_{\widetilde{\sigma}} \end{aligned} \tag{17}$$

Next, we bound $\widetilde{\sigma}$ (see also Greenshtein & Ritov (2004)). Since y and $\boldsymbol{\phi}(\mathbf{x})$ are bounded random variables, we have that $|z_i z_j| \leq B$ for some B and using Hoeffding's inequality

$$\Pr\left\{|\widehat{\boldsymbol{\Sigma}}_{ij} - \boldsymbol{\Sigma}_{ij}| \geq \sigma\right\} \leq 2\exp\left(-\frac{n\sigma^2}{2B^2}\right) \tag{18}$$

Combining this result with the union bound over all $(d+1)^2$ variables in $\widetilde{\sigma}$, we have that

$$\Pr\{\widetilde{\sigma} \geq \sigma\} \leq (d+1)^2 \cdot 2\exp\left(-\frac{n\sigma^2}{2B^2}\right) \triangleq \varepsilon \tag{19}$$

Consequently, we can replace $\widetilde{\sigma}$ by

$$\sigma = B\sqrt{\frac{2}{n}}\sqrt{\ln\frac{2(d+1)^2}{\varepsilon}} \tag{20}$$

in (17) so that

$$|\mathcal{R}_n(\widehat{y}) - \mathcal{R}(\widehat{y})| \leq (\|\boldsymbol{\theta}\|_1 + 1)^2\sigma \tag{21}$$

holds for any predictor in $\mathcal{F}$ with a probability of at least $1-\varepsilon$.

Let us now study two predictor functions in $\mathcal{F}$: an optimal predictor $y^\star$ and the learned predictor, denoted $\widehat{y}^\circ$. Both belong to $\mathcal{F}$ and will therefore have two correponding parameters which we denote $\boldsymbol{\theta}^\star$, which is fixed, and $\boldsymbol{\theta}^\circ$, which depends on $\mathcal{D}_n$. Since $\mathbf{z}$ and $\boldsymbol{\Sigma}$ are bounded, the parameters are also bounded such that $\|\boldsymbol{\theta}^\star\|_1, \|\boldsymbol{\theta}^\circ\|_1 \leq P$ for some P. Using (21), we have that

$$\mathcal{R}(\widehat{y}) - (P+1)^2\sigma \leq \mathcal{R}_n(\widehat{y}) \leq \mathcal{R}(\widehat{y}) + (P+1)^2\sigma \tag{22}$$

for any predictor function in $\mathcal{F}$. The parameter $\boldsymbol{\theta}^\circ$ minimizes the criterion in (12), and therefore

$$\sqrt{\mathcal{R}_n(\widehat{y}^\circ)} + n^{-1/2}\|\boldsymbol{\psi} \odot \boldsymbol{\theta}^\circ\|_1 \leq \sqrt{\mathcal{R}_n(y^\star)} + n^{-1/2}\|\boldsymbol{\psi} \odot \boldsymbol{\theta}^\star\|_1,\ \forall n$$

After rearranging, we have

$$\begin{aligned}\sqrt{\mathcal{R}_n(\widehat{y}^\circ)} - \sqrt{\mathcal{R}_n(y^\star)} &\leq n^{-1/2}(\|\boldsymbol{\psi} \odot \boldsymbol{\theta}^\star\|_1 - \|\boldsymbol{\psi} \odot \boldsymbol{\theta}^\circ\|_1) \\ &\leq n^{-1/2}\|\boldsymbol{\psi} \odot \boldsymbol{\theta}^\star\|_1 \\ &\leq n^{-1/2}\beta P,\end{aligned} \tag{23}$$

where $\beta = \|\boldsymbol{\psi}\|_\infty$. Multiplying both sides of the equality by the positive quantity $(\sqrt{\mathcal{R}_n(\widehat{y}^\circ)} + \sqrt{\mathcal{R}_n(y^\star)})$, we have

$$\begin{aligned}\mathcal{R}_n(\widehat{y}^\circ) - \mathcal{R}_n(y^\star) &\leq (\sqrt{\mathcal{R}_n(\widehat{y}^\circ)} + \sqrt{\mathcal{R}_n(y^\star)})n^{-1/2}\beta P \\ &\leq (2\sqrt{\mathcal{R}_n(y^\star)} + n^{-1/2}\beta P\ )n^{-1/2}\beta P,\end{aligned} \tag{24}$$

where the second inequality follows from using (23). Finally, by definition $\mathcal{R}(y^\star) \leq \mathcal{R}(\widehat{y}^\circ)$ and we have that

$$\begin{aligned}\mathcal{R}(\widehat{y}^\circ) &\leq \mathcal{R}_n(\widehat{y}^\circ) + (P+1)^2\sigma \\ &\leq \mathcal{R}_n(y^\star) + (P+1)^2\sigma + (2\sqrt{\mathcal{R}_n(y^\star)} + n^{-1/2}\beta P\ )n^{-1/2}\beta P \\ &\leq \mathcal{R}(y^\star) + 2(P+1)^2\sigma + (2\sqrt{\mathcal{R}(y^\star) + (P+1)^2\sigma} + n^{-1/2}\beta P\ )n^{-1/2}\beta P \\ &= \mathcal{R}(y^\star) + 2(P+1)^2B\sqrt{2}\cdot\sqrt{\frac{1}{n}\ln\frac{2(d+1)^2}{\varepsilon}} + \mathcal{O}(n^{-3/4}),\end{aligned} \tag{25}$$

with a probability of at least $1-\varepsilon$, where (22) was used in the first and third inequality and (24) was used in the second inequality.

3.3 Distributional robustness

In the previous section, we saw that the out-of-sample Mse of $\widehat{y}(\mathbf{x}; \boldsymbol{\lambda}^\circ)$ approaches the minimum achievable Mse at a root-n rate. We will now see that this predictor also provides robustness against distributional uncertainty for finite n, which lends it a noteworthy interpretation.

Recall that the distribution $p(\mathbf{x}, y)$ in (13) is unknown. Using n i.i.d. samples, we can define a predictor in $\mathcal{F}$ that minimizes the Mse under the *least favourable distribution* among all plausible distributions that are

consistent with the data. Such a predictor is called 'distributionally robust', see, e.g., Duchi & Namkoong (2018). To formalize a set of plausible distributions, we begin by noting that the MSE of any predictor in $\mathcal{F}$ can be expressed as

$$\mathbb{E}\left[(y - \widehat{y}(\mathbf{x}))^2\right] \equiv \mathbb{E}_p\Big[(y - \underbrace{\boldsymbol{\phi}^\top \boldsymbol{\theta}}_{=\widehat{y}\in\mathcal{F}})^2\Big], \tag{26}$$

using the joint distribution $p(\boldsymbol{\phi}, y)$ where $\boldsymbol{\phi} = \boldsymbol{\phi}(\mathbf{x})$ is a random variable in $\mathbb{R}^d$. Since p is unknown, we consider *all* distributions within some given divergence from the empirical distribution

$$p_n(\boldsymbol{\phi}, y) = \frac{1}{n}\sum_{i=1}^{n} \delta(\boldsymbol{\phi} - \boldsymbol{\phi}_i,\ y - y_i) \tag{27}$$

That is, the set of distributions

$$\{p : D(p_n, p) \leq \epsilon_n\}, \tag{28}$$

where $D(p_n, p)$ is some divergence measure. A distributionally robust predictor minimizes the MSE under the least-favourable distribution in (28), viz.

$$\max_{p\,:\,D(p_n,p)\leq\epsilon_n} \quad \mathbb{E}_p\left[(y - \boldsymbol{\phi}^\top\boldsymbol{\theta})^2\right] \tag{29}$$

Several different divergence measures $D(p_n, p)$ have been considered in the literature, including Kullback-Leibler divergence, chi-square divergence, and so on. One popular divergence measure is the Wasserstein distance (Blanchet et al., 2019), which is defined as

$$D(p_n, p) = \inf_{\pi}\ \mathbb{E}_\pi\left[c(\boldsymbol{\phi}, y,\ \boldsymbol{\phi}', y')\right], \tag{30}$$

where $c(\boldsymbol{\phi}, y,\ \boldsymbol{\phi}', y')$ is a nonnegative cost function and π is a joint distribution over $(\boldsymbol{\phi}, y,\ \boldsymbol{\phi}', y')$ whose marginals equal $p_n(\boldsymbol{\phi}, y)$ and $p(\boldsymbol{\phi}', y')$, respectively. Thus $D(p_n, p)$ can be interpreted as measuring the expected cost of moving probability mass from one distribution to the other.

Theorem 3 *Suppose we standardize the feature matrix $\boldsymbol{\Phi}$ in (5) so that its columns have unit norm. Then the predictor $\widehat{y}(\mathbf{x}; \boldsymbol{\lambda}^\circ)$ minimizes the out-of-sample error (26) with respect to the least favourable distribution among all distributions within a Wasserstein distance of ϵ_n from the empirical distribution p_n. The distance $D(p_n, p)$ is given by (30), with a cost function*

$$c(\boldsymbol{\phi}, y,\ \boldsymbol{\phi}', y') = \begin{cases} \|\boldsymbol{\phi} - \boldsymbol{\phi}'\|_\infty^2 & y = y', \\ \infty & \textit{otherwise.} \end{cases} \tag{31}$$

and $\epsilon_n = n^{-2}$. Thus, $\widehat{y}(\mathbf{x}; \boldsymbol{\lambda}^\circ)$ is robust against distributional uncertainties in the features $\boldsymbol{\phi}$ which may be high-dimensional. Note that the size of the distribution set ϵ_n shrinks with n.

Proof 3 *By normalizing the columns of the feature matrix $\boldsymbol{\Phi}$, (12) becomes*

$$\arg\min_{\boldsymbol{\theta}} \sqrt{\frac{1}{n}\|\mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\theta}\|_2^2} + \frac{1}{\sqrt{n}}\|\boldsymbol{\theta}\|_1. \tag{32}$$

Using Theorem 1 in Blanchet et al. (2019), it follows that the resulting predictor minimizes (29) with divergence $\epsilon_n = n^{-2}$.

3.4 In-sample robustness

When learning the predictor $\widehat{y}(\mathbf{x}; \boldsymbol{\lambda}^\circ)$ it is possible that the observed covariates themselves are subject to errors so that the dataset is:

$$\widetilde{\mathcal{D}}_n = \{(\widetilde{\mathbf{x}}_1, y_1), \ldots, (\widetilde{\mathbf{x}}_n, y_n)\}$$

Then the true feature vector $\boldsymbol{\phi}_i = \boldsymbol{\phi}(\mathbf{x}_i)$ can be viewed as a perturbed version of the observed vector $\widetilde{\boldsymbol{\phi}}_i = \boldsymbol{\phi}(\widetilde{\mathbf{x}}_i)$, where the perturbation $\boldsymbol{\delta}_i = \boldsymbol{\phi}_i - \widetilde{\boldsymbol{\phi}}_i$ is unknown (aka. errors-in-variables). This problem leads to yet another interpretation of the predictor $\widehat{y}(\mathbf{x}; \boldsymbol{\lambda}^\circ)$.

Theorem 4 *Consider the bounded set of possible in-sample perturbations:*

$$\mathcal{S}_n = \left\{ \boldsymbol{\delta}_1, \ldots, \boldsymbol{\delta}_n \ : \ \mathbb{E}_{p_n}\left[\delta_k^2\right] \leq n^{-1}\, \mathbb{E}_{p_n}\left[\widetilde{\phi}_k^2\right],\ \forall k = 1, \ldots, d \right\}$$

The predictor $\widehat{y}(\mathbf{x}; \boldsymbol{\lambda}^\circ)$ *minimizes the in-sample root-*Mse *under the least-favourable perturbations in* $\mathcal{S}_n$*:*

$$\max_{\{\boldsymbol{\delta}_i\} \in \mathcal{S}_n} \sqrt{\mathbb{E}_{p_n}\left[(y - (\widetilde{\boldsymbol{\phi}} + \boldsymbol{\delta})^\top \boldsymbol{\theta})^2\right]}, \tag{33}$$

where $\widehat{y} = (\widetilde{\boldsymbol{\phi}} + \boldsymbol{\delta})^\top \boldsymbol{\theta} \in \mathcal{F}$*.*

Proof 4 *The problem in (33) can be written as:*

$$\max_{\{\boldsymbol{\delta}_i\} \in \mathcal{S}} \frac{1}{\sqrt{n}} \|\mathbf{y} - (\widetilde{\boldsymbol{\Phi}} + \boldsymbol{\Delta})\boldsymbol{\theta}\|_2, \quad \textit{where } \boldsymbol{\Delta} \ = \ \begin{bmatrix} \boldsymbol{\delta}_1^\top \\ \vdots \\ \boldsymbol{\delta}_n^\top \end{bmatrix} \tag{34}$$

Let $[\boldsymbol{\Delta}]_k$ *denote the* k^{th} *column of the matrix* $\boldsymbol{\Delta}$*. We can then upper bound the error as*

$$\begin{aligned} \max_{\{\boldsymbol{\delta}_i\} \in \mathcal{S}} \frac{1}{\sqrt{n}} \left\| \mathbf{y} - \widetilde{\boldsymbol{\Phi}}\boldsymbol{\theta} - \sum_{k=1}^{d} [\boldsymbol{\Delta}]_k \theta_k \right\|_2 &\leq \max_{\{\boldsymbol{\delta}_i\} \in \mathcal{S}} \frac{1}{\sqrt{n}} \|\mathbf{y} - \widetilde{\boldsymbol{\Phi}}\boldsymbol{\theta}\|_2 + \frac{1}{\sqrt{n}} \sum_{k=1}^{d} \|[\boldsymbol{\Delta}]_k \theta_k\|_2, \\ &\leq \frac{1}{\sqrt{n}} \|\mathbf{y} - \widetilde{\boldsymbol{\Phi}}\boldsymbol{\theta}\|_2 + \max_{\{\boldsymbol{\delta}_i\} \in \mathcal{S}} \frac{1}{\sqrt{n}} \sum_{k=1}^{d} \|[\boldsymbol{\Delta}]_k\|_2 |\theta_k|, \\ &\leq \frac{1}{\sqrt{n}} \|\mathbf{y} - \widetilde{\boldsymbol{\Phi}}\boldsymbol{\theta}\|_2 + \frac{1}{\sqrt{n}} \sum_{k=1}^{d} \sqrt{\mathbb{E}_{p_n}[\widetilde{\phi}_k^2]} |\theta_k|. \end{aligned} \tag{35}$$

where the bound is attainable when

$$[\boldsymbol{\Delta}]_k = \sqrt{\mathbb{E}_{p_n}[\widetilde{\phi}_k^2]} \frac{\mathbf{y} - \widetilde{\boldsymbol{\Phi}}\boldsymbol{\theta}}{||\mathbf{y} - \widetilde{\boldsymbol{\Phi}}\boldsymbol{\theta}||_2}. \tag{36}$$

But the bound is of the same form as the cost function in (12). Thus solving problem (12) implies the minimization of (34). See also Xu et al. (2009, Theorem. 1).

4 Numerical Experiment

In the previous sections we have showed several computational and theoretical properties of the predictor function $\widehat{y}(\mathbf{x}; \boldsymbol{\lambda}^\circ)$ which we shall call the Spice-predictor. In this section we present a numerical experiment for sake of illustration.

We observe a stream of n samples generated by the following (unknown) process

$$\begin{aligned} \mathbf{x} &\sim \text{Uniform}([0, 10]^2), \\ y|\mathbf{x} &\sim \text{GP}(0, k(\mathbf{x}, \mathbf{x}') + \sigma^2 \delta(\mathbf{x}, \mathbf{x}')), \end{aligned} \tag{37}$$

where

$$k(\mathbf{x}, \mathbf{x}') \ = \ \sigma^2 \left(1 + \frac{\sqrt{3}}{l} \|\mathbf{x} - \mathbf{x}'\|_2\right) \exp\left(-\frac{\sqrt{3}}{l} \|\mathbf{x} - \mathbf{x}'\|_2\right).$$

with noise variance $\sigma = 2$ and scale $l = 7$. In other words, $\mathbf{x}$ is a two-dimensional covariate drawn from a uniform distribution and y is drawn from a Gaussian process with zero mean and a Matérn covariance function. A realization of the above GP and n training data points are shown in Figures 1a and 1e.

We consider a class $\mathcal{F}$ with $d = 100$ Laplace basis functions $\{\phi_k(\mathbf{x})\}$ (Solin & Särkkä, 2020). Note that this corresponds to a misspecified covariance model (3). We are interested in the online learning of a predictor

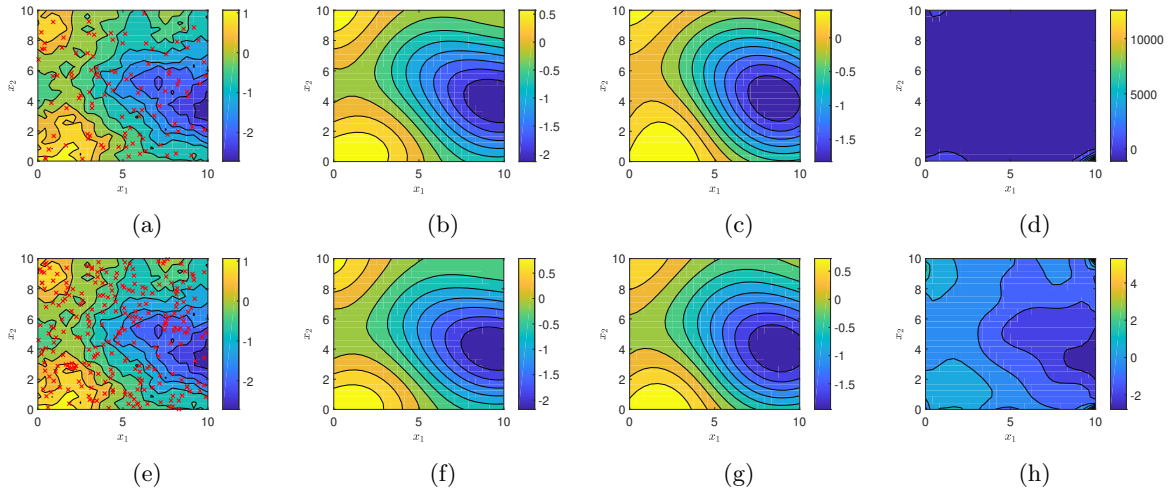

Figure 1: Contour plots. First column shows a realization of y in (37) along with sampling patterns $\{\mathbf{x}_i\}_{i=1}^n \in \mathcal{X}$ for $n = 100$ and $n = 250$ (top and bottom rows, respectively). Second, third and fourth columns show the contour plots of the SPICE-predictor $\widehat{y}(\mathbf{x}; \boldsymbol{\lambda}^\circ)$, ridge regression and the LS-predictor $\widehat{y}(\mathbf{x})$, respectively. All three predictors belong to $\mathcal{F}$.

function in $\mathcal{F}$, and use the least-squares (LS) and ridge regression methods as baseline references. Both methods can be implemented in an online fashion, but the latter requires fixing a regularization parameter. Here we simply set this parameter to 0.1 based on visual inspection.

For illustration, consider the predictions produced by the LS, ridge regression and SPICE methods produce in Figure 1. As expected, the LS provides poor results at these sample sizes. Ridge regression with a fixed regularization parameter and SPICE with adaptively learned parameters appear to perform similarly here. To evaluate their out-of-sample errors, we compare the MSE against that of the oracle predictor based on the unknown Gaussian process (GP) in (37). Table 1 shows that the out-of-sample error of SPICE lower than that of LS and ridge regression, and that the chosen class $\mathcal{F}$ predicts the GP in (37) well.

Following the discussion of effective degrees of freedom df_n in Ruppert et al. (2003), we also provide a comparison between LS, SPICE and the oracle GP predictors in Figure 2. While LS attains the maximum df_n at $n = 100$, SPICE moderates its growth rate in a data-adaptive and online manner. The degrees of freedom of the oracle predictor increases gracefully and remains below its maximum value, even when n increases beyond d.

	MSE/MSE*		
n	LS	RIDGE	SPICE
50	4.38×10^4	1.71	1.11
100	21.12	1.47	1.09
250	1.47	1.19	1.06
500	1.11	1.06	1.02

Table 1: Mean-square error (MSE) for LS and SPICE methods, normalized by MSE* of an oracle predictor which is given the unknown covariance function in (37). For a given set of training data $\mathcal{D}_n$, we compute the averaged squared error over 250 test points. The mean of this error is the MSE and was approximated using 100 different realizations of $\mathcal{D}_n$.

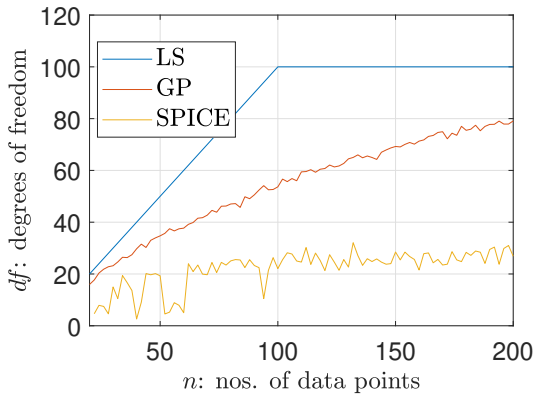

Figure 2: Plot of degrees of freedom df_n against number of data points n for LS, SPICE and oracle GP predictors.

5 Conclusion

We have proposed using a covariance-fitting criterion for learning a linear smoother predictor that, unlike maximum likelihood or cross-validation, enables the predictor to be updated as data points arrive in a streaming fashion. In addition of being trained online, its out-of-sample error approaches the minimum achievable level at root-n rate. It is also robust to distributional uncertainties and errors in the covariate training data. The performance of the proposed method was illustrated in a numerical experiment.

References

Christopher Bishop. *Pattern recognition and machine learning*. Springer, 2006.

Jose Blanchet, Yang Kang, and Karthyek Murthy. Robust wasserstein profile inference and applications to machine learning. *Journal of Applied Probability*, 56(3):830–857, 2019.

John Duchi and Hongseok Namkoong. Learning models with uniform performance via distributionally robust optimization. *arXiv preprint arXiv:1810.08750*, 2018.

Eitan Greenshtein and Ya'Acov Ritov. Persistence in high-dimensional linear predictor selection and the virtue of overparametrization. *Bernoulli*, 10(6):971–988, 2004.

Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

James Hensman, Nicolas Durrande, Arno Solin, et al. Variational fourier features for gaussian processes. *J. Mach. Learn. Res.*, 18(151–1), 2017.

Ali Rahimi, Benjamin Recht, et al. Random features for large-scale kernel machines. In *NIPS*, volume 3, pp. 5. Citeseer, 2007.

Carl Edward Rasmussen and Chris Williams. *Gaussian processes for machine learning*, volume 1. MIT press Cambridge, 2006.

David Ruppert, Matt P Wand, and Raymond J Carroll. *Semiparametric regression*. Number 12. Cambridge university press, 2003.

Arno Solin and Simo Särkkä. Hilbert space methods for reduced-rank gaussian process regression. *Statistics and Computing*, 30(2):419–446, 2020.

Michael L Stein. *Interpolation of spatial data: some theory for kriging*. Springer Science & Business Media, 2012.

P. Stoica and O. Stanasila. Some spectral properties of the matrix b/a. *Bul. Inst. Politehnic Bucuresti, ser. Electro.*, 44(3):3–8, 1982.

Petre Stoica, Prabhu Babu, and Jian Li. New method of sparse parameter estimation in separable models and its use for spectral analysis of irregularly sampled data. *IEEE Transactions on Signal Processing*, 59 (1):35–47, 2010a.

Petre Stoica, Prabhu Babu, and Jian Li. Spice: A sparse covariance-based estimation method for array processing. *IEEE Transactions on Signal Processing*, 59(2):629–638, 2010b.

Larry Wasserman. *All of nonparametric statistics*. Springer Science & Business Media, 2006.

Huan Xu, Constantine Caramanis, and Shie Mannor. Robust regression and lasso. In *Advances in Neural Information Processing Systems*, pp. 1801–1808, 2009.

Dave Zachariah and Petre Stoica. Online hyperparameter-free sparse estimation method. *IEEE Transactions on Signal Processing*, 63(13):3348–3359, 2015.