Machine Learning 33, 5–39 (1998)

Prequential and Cross-Validated Regression Estimation

DHARMENDRA S. MODHA
IBM Almaden Research Center, San Jose, CA 95120-6099, USA

ELIAS MASRY
Department of Electrical and Computer Engineering,
University of California at San Diego, La Jolla, CA 92093-0407, USA

Editor: David Haussler

Abstract. Prequential model selection and delete-one cross-validation are data-driven methodologies for choosing between rival models on the basis of their predictive abilities. For a given set of observations, the predictive ability of a model is measured by the model's accumulated prediction error and by the model's average-out-of-sample prediction error, respectively, for prequential model selection and for cross-validation. In this paper, given i.i.d. observations, we propose nonparametric regression estimators—based on neural networks—that select the number of "hidden units" (or "neurons") using either prequential model selection or delete-one cross-validation. As our main contributions: (i) we establish rates of convergence for the integrated mean-squared errors in estimating the regression function using "off-line" or "batch" versions of the proposed estimators and (ii) we establish rates of convergence for the time-averaged expected prediction errors in using "on-line" versions of the proposed estimators. We also present computer simulations (i) empirically validating the proposed estimators and (ii) empirically comparing the proposed estimators with certain novel prequential and cross-validated "mixture" regression estimators.

Keywords: regression estimation, prequential model selection, cross-validation, neural networks, rates of convergence, mixture regression, integrated mean-squared error, time-averaged expected prediction error

1. Introduction

Let $\{X_i, Y_i\}_{i=-\infty}^{\infty}$ be independent and identically distributed (i.i.d.) random variables on a probability space $(\Omega, \mathcal{F}, P)$ such that X_0 takes values in $\mathbb{R}^q$ and Y_0 takes values in $\mathbb{R}$. Define the regression function as

$$s(x) = E[Y_0 \mid X_0 = x], \quad x \in \mathbb{R}^q.$$

Given n observations $\{X_i, Y_i\}_{i=1}^{n}$, we are interested in estimating the regression function s.

We do not assume that the regression function s is a member of a finite-dimensional parametric model, hence it is natural to estimate s using a countable sequence of finite-dimensional parametric models with increasing dimensions, say $\{S_m\}_{m \in \mathcal{M}}$, which approximates s more accurately as the dimension m increases. As an example, S_m may represent

a class of neural networks with m hidden units and $\mathcal{M}$ may represent the set of natural numbers. In practice, for finite number of observations n, it is common to estimate s using a model $S_m, m \in \mathcal{M}_n$, where the set of dimensions $\mathcal{M}_n \subset \mathcal{M}$ grows with the sample size n at an appropriate rate. As an example, $\mathcal{M}_n$ may be $\{1, 2, \ldots, M_n\}$ for some finite number M_n.

Statistical risk in estimating the regression function s using a finite-dimensional parametric model $S_m, m \in \mathcal{M}_n$, has two components: approximation error and estimation error. Roughly speaking, a model with a larger dimension has a smaller approximation error but a larger estimation error, whereas a model with a smaller dimension has a smaller estimation error but a larger approximation error. The problem of model selection is to empirically select the finite-dimensional parametric model (from the permissible collection of models $\{S_m\}_{m \in \mathcal{M}_n}$) that achieves the best tradeoff between the competing approximation error and estimation error components—and, consequently, achieves the smallest possible statistical risk in estimating s. For previous theoretical work on model selection in the context of nonparametric regression estimation, see, for example, Barron (1991, 1994), Barron, Birgé, & Massart (1996), Birgé & Massart (1994b), Baum & Haussler (1989), Haussler (1992), Lugosi & Nobel (1995), Lugosi & Zeger (1996), McCaffrey & Gallant (1994), Modha & Masry (1996, 1998), Rissanen (1989), Shen & Wong (1994), Vapnik (1982, 1995), and White (1989).

In this paper, we study model selection in the context of nonparametric regression estimation using prequential model selection due to Dawid (1984, 1991, 1992) and Rissanen (1986a, 1986b, 1989) and also using delete-one cross-validation (or cross-validation for short) due to Mosteller & Tukey (1968) and Stone (1974, 1977).[1] Prequential and cross-validated regression estimators are attractive for practical application, in that, they require minimal inputs from the user: a set of observations $\{X_i, Y_i\}_{i=1}^n$, a suitable set of dimensions $\mathcal{M}_n$, and a sequence of finite-dimensional parametric models $\{S_m\}_{m \in \mathcal{M}_n}$. From this perspective, prequential model selection and cross-validation represent exciting steps towards automatic (or completely data-driven) model selection.

In this paper, we examine prequential and cross-validated regression estimators based on neural networks. As our main contributions: (i) we establish rates of convergence for the integrated mean-squared errors in estimating the regression function using "off-line" or "batch" versions of the prequential and the cross-validated estimators (Theorem 2.1) and (ii) we establish rates of convergence for the time-averaged expected prediction errors in using "on-line" versions of the prequential and the cross-validated estimators (Corollary 2.1). To the best of our knowledge, no such rates of convergence results have been previously established for prequential or cross-validated regression estimators in a nonparametric setting. We also present computer simulations (i) empirically validating the proposed estimators and (ii) empirically comparing the proposed estimators with certain novel prequential and cross-validated "mixture" regression estimators.

1.1. Prequential model selection

Prequential model selection is a data-driven methodology for choosing between rival models on the basis of their predictive abilities. In figure 1, we present a generic estimation scheme

Inputs: Sample size n and observations $\{X_i, Y_i\}_{i=1}^n$
a set of model dimensions $\mathcal{M}_n := \{1, 2, \ldots, M_n\}$
a sequence of finite-dimensional parametric models $\{S_m\}_{m \in \mathcal{M}_n}$

Estimation Scheme:
for $m := 1$ **to** M_n **step** 1
Choose a fixed initial estimator $\hat{s}_{(m,0)} \in S_m$
PREQ$(m, 1) := [Y_1 - \hat{s}_{(m,0)}(X_1)]^2$
for $j := 2$ **to** n **step** 1
compute the least-squares estimator based on $\{X_i, Y_i\}_{i=1}^{j-1}$
$\hat{s}_{(m,j-1)} := \arg\min_{g \in S_m} \left\{ \sum_{i=1}^{j-1} [Y_i - g(X_i)]^2 \right\} \in S_m$
update the prequential loss
PREQ$(m, j) :=$ PREQ$(m, j-1) + [Y_j - \hat{s}_{(m,j-1)}(X_j)]^2$
endfor;
endfor;

Output: Compute the model dimension and the prequential regression estimator, respectively, as

$$\hat{m}^{(p)} \equiv \hat{m}_n^{(p)} := \arg\min_{1 \le m \le M_n} \text{PREQ}(m, n) \tag{1}$$

$$\hat{s}_n^{(p)} \equiv \hat{s}_{(\hat{m}^{(p)},n)} := \arg\min_{g \in S_{\hat{m}^{(p)}}} \left\{ \sum_{i=1}^{n} [Y_i - g(X_i)]^2 \right\} \in S_{\hat{m}^{(p)}} \tag{2}$$

Figure 1. Scheme for computing the prequential regression estimator.

for computing prequential regression estimators. Intuitively, the term

$$[Y_j - \hat{s}_{(m,j-1)}(X_j)]^2$$

in figure 1 denotes the prediction error incurred on the next observation Y_j, given X_j, by a least-squares estimator with dimension m based on previous $(j-1)$ observations $\{X_i, Y_i\}_{i=1}^{j-1}$. Consequently, the prequential loss PREQ(m, n) in figure 1 represents the "accumulated prediction error" committed by a m-dimensional model on n observations $\{X_i, Y_i\}_{i=1}^n$, and prequential model selection chooses the dimension $\hat{m}_n^{(p)}$ that minimizes the accumulated prediction error.

The estimation scheme in figure 1 operates on a fixed set (or batch) of n observations $\{X_i, Y_i\}_{i=1}^n$, and hence is off-line. However, prequential model selection as conceived by Dawid (1984, 1991, 1992) is quint-essentially on-line—where the observations are assumed to arrive sequentially. Specifically, in the on-line case, for each $k \ge 1$, having seen k observations $\{X_i, Y_i\}_{i=1}^k$, one is interested in predicting Y_{k+1} given X_{k+1}. It is easy to apply the estimation scheme in figure 1 to the on-line case by observing that for each k one can use the prequential regression estimator $\hat{s}_k^{(p)}$ based on k observations $\{X_i, Y_i\}_{i=1}^k$. Specifically, $\hat{s}_k^{(p)}$ is obtained by replacing n by k in figure 1.

The notion of the accumulated prediction error is closely related to the notion of "predictive code-length" considered in Rissanen (1986b, 1989), Rissanen, Speed, & Yu (1992) and Yu & Speed (1992) and to the notion of "Shannon information gain" considered in Haussler, Kearns, & Schapire (1994).

For finite-dimensional Gaussian regression problem, Rissanen (1986a) has established order consistency of prequential model selection. However, our regression function may neither be linear (in observations or parameters) nor finitely parameterized. Furthermore, Rissanen did not address the issue of establishing rates of convergence for the statistical risk. Yu & Speed (1992) (also see Rissanen, Speed, & Yu, 1992) considered a prequential density estimator based on histograms and established almost sure rate of convergence for the excess code-length incurred by their estimator. Their results, however, are specific to density estimation using histograms.

In this paper, assuming that the regression function s satisfies a certain Fourier-transform type representation (Assumption 2.3), we examine a prequential regression estimator based on neural networks. We establish rates of convergence for the statistical risks in using off-line (Theorem 2.1) and on-line (Corollary 2.1) versions of this estimator.

1.2. Cross-validation

Cross-validation is a data-driven methodology for choosing between rival models on the basis of their predictive abilities. In figure 2, we present a generic estimation scheme for computing cross-validated regression estimators. Intuitively, the term

$$\left[Y_j - \hat{s}^{(j)}_{(m,n-1)}(X_j)\right]^2$$

in figure 2 denotes the prediction error incurred on the observation Y_j, given X_j, by a least-squares estimator with dimension m based on $(n-1)$ observations $\{X_i, Y_i\}_{i=1, i\neq j}^{n}$. Consequently, the cross-validation loss $\mathrm{CV}(m, n)$ in figure 1 represents the "average out-of-sample prediction error" committed by a m-dimensional model on n observations $\{X_i, Y_i\}_{i=1}^{n}$, and cross-validation chooses the dimension $\hat{m}^{(c)}_n$ that minimizes the average out-of-sample prediction error.

Unlike prequential model selection, cross-validation is inherently an off-line estimation scheme in that it operates on a fixed set (or batch) of n observations $\{X_i, Y_i\}_{i=1}^{n}$, as can be seen from figure 2. Nonetheless, it is possible to coerce cross-validation to operate in an on-line fashion by observing that for each k one can use the cross-validated regression estimator $\hat{s}^{(c)}_k$ based on k observations $\{X_i, Y_i\}_{i=1}^{k}$. Specifically, $\hat{s}^{(c)}_k$ is obtained by replacing n by k in figure 2.

The literature concerning cross-validation and its variants is rather vast, see, for example, Li (1987), Stone (1984), and Stone (1974, 1977); in this paper, we restrict attention to cross-validated model selection applied to sequences of parametric models such as neural networks. In a setting closely related to ours, White (1989) established weak consistency of cross-validated regression estimators based on neural networks without rates. We focus on obtaining rates of convergence. Birgé & Massart (1994a) established rates of convergence specifically for cross-validated projection density estimators based on linear models (such as

Inputs: Sample size n and observations $\{X_i, Y_i\}_{i=1}^n$
a set of model dimensions $\mathcal{M}_n := \{1, 2, \ldots, M_n\}$
a sequence of finite-dimensional parametric models $\{S_m\}_{m \in \mathcal{M}_n}$

Estimation Scheme:
for $m := 1$ **to** M_n **step** 1
for $j := 1$ **to** n **step** 1
delete j-th observation from $\{X_i, Y_i\}_{i=1}^n$
compute the least-squares estimator based on $\{X_i, Y_i\}_{i=1, i \neq j}^n$

$$\hat{s}_{(m,n-1)}^{(j)} := \arg\min_{g \in S_m} \left\{ \sum_{i=1, i \neq j}^n [Y_i - g(X_i)]^2 \right\} \in S_m$$

endfor;
compute the cross-validation loss

$$\mathrm{CV}(m, n) := \sum_{j=1}^n [Y_j - \hat{s}_{(m,n-1)}^{(j)}(X_j)]^2$$

endfor;

Output: Compute the model dimension and the cross-validated regression estimator, respectively, as

$$\hat{m}^{(c)} \equiv \hat{m}_n^{(c)} := \arg\min_{1 \leq m \leq M_n} \mathrm{CV}(m, n) \tag{3}$$

$$\hat{s}_n^{(c)} \equiv \hat{s}_{(\hat{m}^{(c)}, n)} := \arg\min_{g \in S_{\hat{m}^{(c)}}} \left\{ \sum_{i=1}^n [Y_i - g(X_i)]^2 \right\} \in S_{\hat{m}^{(c)}} \tag{4}$$

Figure 2. Scheme for computing the cross-validated regression estimator.

wavelets). Their results do not extend to cross-validated regression estimators. Recently, Kearns (1997) considered a cross-validation scheme (based on saving out a fraction of the available data as an independent test set) for model selection, and established rates of convergence for his estimators. Here, we focus on a different estimation scheme: delete-one cross-validation. His results do not extend to delete-one cross-validation.

Assuming that the regression function s satisfies a certain Fourier-transform type representation (Assumption 2.3), we examine a cross-validated regression estimator based on neural networks. We establish rates of convergence for the statistical risks in using off-line (Theorem 2.1) and on-line (Corollary 2.1) versions of this estimator.

This paper is organized as follows: In Section 2, we propose prequential and cross-validated regression estimators based on neural networks, and establish our main results (Theorem 2.1 and Corollary 2.1). We also compare the proposed estimators to certain complexity-regularized least-squares estimators. Inspired by Bayesian mixture density estimation of Dawid (1991), we propose in Section 2 certain prequential and cross-validated mixture regression estimators. In Section 3, we present the advertised computer simulation study. In Section 4, we examine prequential and cross-validated regression estimators based on a sequence of abstract parametric models, and establish abstract upper bounds on the integrated mean-squared errors in estimating s using these abstract estimators (Theorem 4.1). Theorem 2.1 follows by adapting Theorem 4.1 to neural networks. We note that Theorem 4.1

is fairly general, and may extend to prequential and cross-validated regression estimators based, for example, on wavelets, polynomials, splines, and Fourier series.

2. Regression estimation using neural networks

2.1. Neural networks

Assumption 2.1. *X_0 takes values in $[-1, 1]^q$.*

Assumption 2.2. *Y_0 takes values in $[-\xi, \xi]$ for some known $\xi > 0$.*

We assume that the regression function s satisfies the following Fourier-transform-type representation due to Barron (1993). For $w, x \in \mathbb{R}^q$, let $w \cdot x$ denote the usual inner product on $\mathbb{R}^q$ and let $\|w\|_1$ denote the ℓ^1-norm on $\mathbb{R}^q$.

Assumption 2.3. *There exists a complex valued function $\tilde{s}$ on $\mathbb{R}^q$ such that for $x \in [-1, 1]^q$, we have*

$$s(x) - s(0) = \int_{\mathbb{R}^q} (e^{iw\cdot x} - 1)\tilde{s}(w)\, dw$$

and that $\int_{\mathbb{R}^q} \|w\|_1 |\tilde{s}(w)|\, dw \le C' < \infty$ for some known $C' > 0$. Set $C = \max\{1, C'\}$.

For examples of functions satisfying Assumption 2.3, we refer the interested reader to Barron (1993). A function satisfying Assumption 2.3 can be approximated, without the curse of dimensionality (in terms of approximation error rates), using the sequence of parametric models based on neural networks presented below. Let $\phi : \mathbb{R} \to [0, 1]$ denote a Lipschitz continuous sigmoidal function such that its tails approach the tails of the unit step function at least polynomially fast.[2]

Assumption 2.4. *The function $\phi : \mathbb{R} \to [0, 1]$ is such that*
(a) *$\phi(u) \to 1$ as $u \to \infty$ and $\phi(u) \to 0$ as $u \to -\infty$.*
(b) *$|\phi(u)| \le 1$ and $|\phi(u) - \phi(v)| \le |u - v|$ for all $u, v \in \mathbb{R}$.*
(c) *$|\phi(u) - 1_{\{u>0\}}| \le A_1'/|u|^{A_2}$ for $u \in \mathbb{R}\backslash\{0\}$, and for some $A_1', A_2 > 0$. Set $A_1 = \max\{1, A_1'\}$.*

Define

$$\tau_m = 2^{(2A_2+1)/A_2}\, A_1^{1/A_2}\, m^{(A_2+1)/(2A_2)}, \tag{5}$$

where A_1 and A_2 are as in Assumption 2.4. For dimension m, let

$$\bar{S}_m = \left\{ c_0 + \sum_{j=1}^{m} c_j \phi(a_j \cdot x - b_j) \;\middle|\; c_0 \in [-\xi, \xi],\ \sum_{j=1}^{m} |c_j| \le C, \text{ and } ||a_j||_1, |b_j| \le \tau_m \text{ for } 1 \le j \le m \right\}, \tag{6}$$

denote the class of neural networks with m hidden units (or neurons) where τ_m denotes the rate at which the hidden unit weights, namely a_j and b_j, are allowed to grow as a function of m. It follows from Assumption 2.2 that, for all $x \in [-1, 1]^q$, $|s(x)| \leq \xi$. Define a clipped subset of $\bar{S}_m$ as

$$S_m = \{[g \vee (-\xi)] \wedge \xi \mid g \in \bar{S}_m\}, \tag{7}$$

where $\vee = \max$ and $\wedge = \min$. Define a set of dimensions as

$$\mathcal{M}_n = \left\{m \mid 1 \leq m \leq \kappa_1 \sqrt{n/(\ln n)^{\kappa_2}} \equiv M_n\right\}, \tag{8}$$

where $\ln = \log_e$ and the constants

$$\kappa_1 \geq 1 \tag{9}$$

and

$$\kappa_2 = \begin{cases} 0 & \text{for prequential estimation} \\ 1 & \text{for cross-validated estimation} \end{cases} \tag{10}$$

are selected with hindsight to establish the rates of convergence given in Theorem 2.1.

2.2. Prequential and cross-validated regression estimators

Given the set of model dimensions $\mathcal{M}_n$ defined in (8) and the parametric model class defined in (7), compute the prequential regression estimator $\hat{s}_n^{(p)}$ by proceeding as in figure 1 and compute the cross-validated regression estimator $\hat{s}_n^{(c)}$ by proceeding as in figure 2.

We now establish rates of convergence for the integrated mean squared errors of the estimators $\hat{s}_n^{(p)}$ and $\hat{s}_n^{(c)}$. Let P_X denote the marginal distribution of X_0.

Theorem 2.1. *Suppose Assumptions* 2.1, 2.2, 2.3, *and* 2.4 *hold. Then, the following bounds hold for each* $n \geq 2$.

(prequential regression estimation)

$$E \int_{\mathbb{R}^q} \left[s(x) - \hat{s}_n^{(p)}(x)\right]^2 dP_X(x) \leq (\text{constant}) \frac{\ln n}{\sqrt{n}}. \tag{11}$$

(cross-validated regression estimation)

$$E \int_{\mathbb{R}^q} \left[s(x) - \hat{s}_n^{(c)}(x)\right]^2 dP_X(x) \leq (\text{constant}) \sqrt{\frac{\ln n}{n}}. \tag{12}$$

The proof can be found in Section 4.2. The abstract upper bounds established in Section 4 (see Theorem 4.1) are a key step in establishing Theorem 2.1.

So far, we have dealt with the off-line versions of the prequential and the cross-validated regression estimators. We now consider on-line versions of these estimators. In the on-line case, for each $k \geq 1$, having seen k observations $\{X_i, Y_i\}_{i=1}^k$, one is interested in predicting Y_{k+1} given X_{k+1}. We can apply the estimation schemes in figures 1 and 2 to the on-line case by observing that for each k one can use the prequential regression estimator $\hat{s}_k^{(p)}$ or the cross-validated regression estimator $\hat{s}_k^{(c)}$ based on k observations $\{X_i, Y_i\}_{i=1}^k$.

In the on-line case, one is interested in measuring the performance of the sequence of prequential regression estimators $\{\hat{s}_k^{(p)}\}_{k\geq 1}^n$ or the sequence of cross-validated regression estimators $\{\hat{s}_k^{(c)}\}_{k\geq 1}^n$ and not in measuring the performance of the regression estimators $\hat{s}_n^{(p)}$ or $\hat{s}_n^{(c)}$ as considered in Theorem 2.1. Hence, in the on-line case, an appropriate measure of performance is the time-averaged expected prediction error.

We now establish upper bounds on the time-averaged expected prediction errors of the sequences of estimators $\{\hat{s}_k^{(p)}\}_{k\geq 1}^n$ and $\{\hat{s}_k^{(c)}\}_{k\geq 1}^n$.

Corollary 2.1. *Suppose Assumptions* 2.1, 2.2, 2.3, *and* 2.4 *hold. Then*, *the following bounds hold for each* $n \geq 2$.

(prequential regression estimation)

$$\frac{1}{n}\sum_{k=1}^{n} E\left[Y_{k+1} - \hat{s}_k^{(p)}(X_{k+1})\right]^2 < E[Y_0 - s(X_0)]^2 + (\text{constant})\,\frac{\ln n}{\sqrt{n}}. \tag{13}$$

(cross-validated regression estimation)

$$\frac{1}{n}\sum_{k=1}^{n} E\left[Y_{k+1} - \hat{s}_k^{(c)}(X_{k+1})\right]^2 < E[Y_0 - s(X_0)]^2 + (\text{constant})\sqrt{\frac{\ln n}{n}}. \tag{14}$$

The proof can be found in Section 4.2.

Observe that the left-hand sides of (13) and (14) represent the time-averaged expected prediction errors of the sequence of prequential estimators $\{\hat{s}_k^{(p)}\}_{k\geq 1}^n$ and the sequence of cross-validated estimators $\{\hat{s}_k^{(c)}\}_{k\geq 1}^n$, respectively. The first-terms on the right-hand sides of (13) and (14) represent the smallest possible expected prediction error. Thus, one may interpret Corollary 2.1 as establishing finite-sample upper bounds on the *excess* time-averaged expected prediction errors of $\{\hat{s}_k^{(p)}\}_{k\geq 1}^n$ and $\{\hat{s}_k^{(c)}\}_{k\geq 1}^n$.

Remark 2.1. In this remark, fix the model dimension m. Suppose that we have computed the prequential loss $\mathrm{PREQ}(m, n)$ and the cross-validation loss $\mathrm{CV}(m, n)$ using figures 1 and 2, respectively. Now, suppose that we have one additional observation $\{X_{n+1}, Y_{n+1}\}$ and would like to compute $\mathrm{PREQ}(m, n+1)$ and $\mathrm{CV}(m, n+1)$.

To calculate the prequential loss, we first compute the least-squares estimator

$$\hat{s}_{(m,n)} = \arg\min_{g\in S_m} \left\{ \sum_{i=1}^{n} [Y_i - g(X_i)]^2 \right\} \in S_m,$$

and then write

$$\mathrm{PREQ}(m, n+1) = \mathrm{PREQ}(m, n) + \left[Y_{n+1} - \hat{s}_{(m,n)}(X_{n+1})\right]^2. \tag{15}$$

To calculate the cross-validation loss, we first compute $(n+1)$ least-squares estimators

$$\hat{s}^{(j)}_{(m,n)} = \arg\min_{g \in S_m} \left\{ \sum_{i=1, i\neq j}^{n+1} [Y_i - g(X_i)]^2 \right\} \in S_m, \;\; j = 1, 2, \ldots, (n+1),$$

and then write

$$\mathrm{CV}(m, n+1) = \sum_{j=1}^{n+1} \left[Y_j - \hat{s}^{(j)}_{(m,n)}(X_j)\right]^2 \tag{16}$$

Observe that it is not possible to use the cross-validation loss $\mathrm{CV}(m, n)$ in computing $\mathrm{CV}(m, n+1)$ in (16). For each new observation we must compute the cross-validation loss from scratch. In contrast, for the prequential loss, for each new observation we only need to compute the second quantity on the right-hand side of (15). Hence, in an on-line setting, prequential model selection is computationally more efficient.

Note that, for both prequential model selection and cross-validation, the model dimensions $\hat{m}^{(p)}_{n+1}$ and $\hat{m}^{(c)}_{n+1}$ must be recomputed using (1) and (3), respectively, and cannot be written as recursive updates to the previous model dimensions $\hat{m}^{(p)}_n$ and $\hat{m}^{(c)}_n$.

Remark 2.2 (*upper bounds on the number of hidden units*). Note that the prequential regression estimator selects the number of hidden units, in a data-driven fashion, from the range

$$1 \leq m \leq \sqrt{n}.$$

Similarly, the cross-validated regression estimator selects the number of hidden units, in a data-driven fashion, from the range

$$1 \leq m \leq \sqrt{n/(\ln n)}.$$

The choices of the upper limits of the above ranges (roughly $\sqrt{n}$) are the appropriate values needed to establish the rates of convergence given in Theorem 2.1 for the specific class of regression functions satisfying Assumption 2.3. These upper bounds arise from the upper bounds on the complexity of the model class S_m that we derive in the proof of Theorem 2.1. For the class of regression functions satisfying Assumption 2.3, the imposition of a cap on the number of hidden units limits the data-driven search for the model dimension, and is thus computationally appealing. For a different smoothness assumption on the regression function, a different model class and a correspondingly different set of dimensions must be used to obtain the appropriate rate of convergence under that assumption.

Remark 2.3 (*computational complexity of nonlinear least-squares*). The least-squares estimators $\hat{s}_{(m,j-1)}$ and $\hat{s}^{(j)}_{(m,n-1)}$ employed in figures 1 and 2, respectively, are clearly and unambiguously defined—and hence exist. Implicitly, while establishing the rates of convergence results in Theorem 2.1 and Corollary 2.1, we assumed that these least-squares estimators can indeed be computed. Such an assumption is the very basis for applying Vapnik's empirical risk minimization theory to neural networks, and has been widely used in the literature dealing with rates of convergence results for neural networks and other models, see, for example, Barron (1994), Barron, Birgé, & Massart (1996), Breiman (1993), Haussler (1992), Kearns (1997), Lugosi & Nobel (1995), Lugosi & Zeger (1996), McCaffrey & Gallant (1994), Modha & Masry (1996, 1998), Vapnik (1982, 1995), and White (1989).

In the context of neural networks, the problem of finding the (nonlinear) least-squares estimators is known to be computationally intractable, see, for example, Jones (1997). In practice, it is common to use heuristic "approximations" to the least-squares estimators obtained by repeatedly applying the error backpropagation algorithm (Sarkar, 1995) started from a number of initial weights. To quote Kearns (1997): "The extent to which the theory presented here applies to such heuristics will depend in part on the extent to which they approximate training error minimization for the problem under consideration". In any case, the backpropagation algorithm seems to work quite well in our simulation study presented in Section 3.

Remark 2.4 (*comparison with complexity-regularized least squares estimator*). Assuming that Assumptions 2.1, 2.2, 2.3, and 2.4 hold, Barron (1994) proposed a complexity-regularized least squares estimator, say $\hat{s}^{(b)}_n$, based on neural networks, and established that

$$E \int_{\mathbb{R}^q} \left[s(x) - \hat{s}^{(b)}_n(x) \right]^2 dP_X(x) \leq (\text{constant}) \sqrt{\frac{\ln n}{n}}. \tag{17}$$

It follows from (11) and (17) that the rate of convergence achieved by the prequential regression estimators is within a logarithmic factor of that achieved by the complexity-regularized regression estimator. Similarly, it follows from (12) and (17) that the rate of convergence achieved by the cross-validated regression estimator is identical to that achieved by the complexity-regularized regression estimator.

Suppose that the underlying regression function s is parametric, that is, is a member of a finite-dimensional parametric model, say $S_{m^\star}$. In this case, it is known that the complexity-regularized regression estimator is *order-universal*, that is, the complexity-regularized regression estimator, which does not know the true dimension $m^\star$, delivers the same rate of integrated mean-squared error as that delivered by an estimator that knows the true dimension. See Barron (1994), Barron, Birgé, & Massart (1996), and Modha & Masry (1998). Establishing order-universality results for prequential and cross-validated regression estimators currently remains an open problem.

Remark 2.5 (*almost sure convergence*). An important open problem is to establish that the excess time-averaged prediction error of the sequence of prequential estimators $\{\hat{s}^{(p)}_k\}^n_{k\geq 1}$

converges almost surely to zero, that is,

$$\frac{1}{n}\sum_{k=1}^{n}\left\{\left[Y_{k+1}-\hat{s}_k^{(p)}(X_{k+1})\right]^2-\left[Y_{k+1}-s(X_{k+1})\right]^2\right\}\to 0 \text{ almost surely as } n\to\infty.$$

In addition, it would be interesting to determine the almost sure rate of convergence. Finally, similar results will also be interesting for the sequence of cross-validated estimators $\{\hat{s}_k^{(c)}\}_{k\geq 1}^n$.

2.3. *Prequential and cross-validated mixture regression estimators*

In the context of density estimation, Dawid (1991) has proposed a Bayesian mixture density estimator. Here, we extend his ideas to regression estimation.

For $m \geq 1$, let $\log_2^\star(m) = \lceil \log_2(m)\rceil + \lceil \log_2\lceil \log_2(m)\rceil\rceil + \cdots$ where the sum involves only the non-negative terms. Intuitively, $\log_2^\star(m)$ denotes the number of bits in a self-delimiting code for the integer m. For $m \geq 1$, let $Q(m) = c^\star 2^{-\log_2^\star(m)}$ denote a prior density on the set of natural numbers, where the normalization constant $c^\star$ ensures that $\sum_{m\geq 1} Q(m) = 1$. The precise value of $c^\star$ is not required in the sequel. The prior density Q is a slightly modified version of the universal prior density proposed by Rissanen (1983).

Suppose we are given a set of observations $\{X_i, Y_i\}_{i=1}^n$, a suitable set of dimensions $\mathcal{M}_n = \{1, 2, \ldots, M_n\}$, and a sequence of finite-dimensional parametric models $\{S_m\}_{m\in\mathcal{M}_n}$.

Define the prequential mixture regression estimator as

$$\hat{s}_n^{(pmix)} = \sum_{m=1}^{M_n} \alpha_{m,n}\,\hat{s}_{(m,n)}, \tag{18}$$

where $\hat{s}_{(m,n)} \in S_m$ denotes the least-squares estimator based on observations $\{X_i, Y_i\}_{i=1}^n$ and the weights $\alpha_{m,n}$ are defined as

$$\alpha_{m,n} = \frac{\alpha'_{m,n}}{\sum_{k=1}^{M_n}\alpha'_{k,n}}. \tag{19}$$

where for $k = 1, 2, \ldots, M_n$,

$$\begin{aligned}\alpha'_{k,n} &= Q(k)\prod_{j=1}^{n}\frac{1}{\sqrt{2\pi\hat{\sigma}^2_{(k,j-1)}}}\exp\left(-\left[Y_j-\hat{s}_{(k,j-1)}(X_j)\right]^2\Big/\left(2\hat{\sigma}^2_{(k,j-1)}\right)\right)\\ &= \left(\frac{c^\star}{(2\pi)^{n/2}}\right)2^{-\log_2^\star(k)}\prod_{j=1}^{n}\frac{1}{\sqrt{\hat{\sigma}^2_{(k,j-1)}}}\exp\left(-\left[Y_j-\hat{s}_{(k,j-1)}(X_j)\right]^2\Big/\left(2\hat{\sigma}^2_{(k,j-1)}\right)\right)\end{aligned} \tag{20}$$

where $\hat{s}_{(k,j-1)} \in S_k$ denotes the least-squares estimator based on observations $\{X_i, Y_i\}_{i=1}^{j-1}$ and $\hat{\sigma}^2_{(k,0)}$ is set to a constant value, say 1, and, for $j > 1$, $\hat{\sigma}^2_{(k,j-1)}$ denotes the residual

least-squares error

$$\hat{\sigma}^2_{(k,j-1)} = \frac{1}{j-1}\sum_{i=1}^{j-1}\left[Y_i - \hat{s}_{(k,j-1)}(X_i)\right]^2. \tag{21}$$

Observe that for computational purposes we can safely replace the constant factor $c^\star/(2\pi)^{n/2}$ in (20) by 1, since it appears in both the numerator and denominator of (23).

Now, define the cross-validated mixture regression estimator as

$$\hat{s}_n^{(cmix)} = \sum_{m=1}^{M_n} \beta_{m,n}\,\hat{s}_{(m,n)}, \tag{22}$$

where $\hat{s}_{(m,n)} \in S_m$ denotes the least-squares estimator based on observations $\{X_i, Y_i\}_{i=1}^n$ and the weights $\beta_{m,n}$ are defined as

$$\beta_{m,n} = \frac{\beta'_{m,n}}{\sum_{k=1}^{M_n}\beta'_{k,n}}. \tag{23}$$

where for $k = 1, 2, \ldots, M_n$,

$$\begin{aligned}\beta'_{k,n} &= Q(k)\prod_{j=1}^{n}\frac{1}{\sqrt{2\pi\left(\hat{\sigma}^{(j)}_{(k,n)}\right)^2}}\exp\left(-\left[Y_j - \hat{s}^{(j)}_{(k,n)}(X_j)\right]^2\Big/\left(2\left(\hat{\sigma}^{(j)}_{(k,n)}\right)^2\right)\right)\\ &= \left(\frac{c^\star}{(2\pi)^{n/2}}\right)2^{-\log_2^\star(k)}\prod_{j=1}^{n}\frac{1}{\sqrt{\left(\hat{\sigma}^{(j)}_{(k,n)}\right)^2}}\exp\left(-\left[Y_j - \hat{s}^{(j)}_{(k,n)}(X_j)\right]^2\Big/\left(2\left(\hat{\sigma}^{(j)}_{(k,n)}\right)^2\right)\right)\end{aligned} \tag{24}$$

where $\hat{s}^{(j)}_{(k,n)} \in S_k$ denotes the least-squares estimator based on observations $\{X_i, Y_i\}_{i=1,i\neq j}^n$ and $(\hat{\sigma}^{(j)}_{(k,n)})^2$ denotes the residual least-squares error

$$\left(\hat{\sigma}^{(j)}_{(k,n)}\right)^2 = \frac{1}{n-1}\sum_{i=1,i\neq j}^{n}\left[Y_i - \hat{s}^{(j)}_{(k,n)}(X_i)\right]^2. \tag{25}$$

Observe that for computational purposes we can safely replace the constant factor $c^\star/(2\pi)^{n/2}$ in (24) by 1, since it appears in both the numerator and denominator of (23).

Establishing rates of convergence for the statistical risks of the prequential and the cross-validated mixture regression estimators currently remains an open problem. In the next section, we empirically assess the performance of these estimators.

3. Computer simulations

We now empirically demonstrate the performance of prequential model selection and cross-validation via two simple, simulated examples. We also empirically compare prequential

and cross-validated regression estimators with prequential and cross-validated mixture regression estimators.

For various appealing computer simulation studies demonstrating prequential model selection (i) in the context of ARMA order selection (see Dawid, 1991; Rissanen, 1989); (ii) in the context of density estimation using histograms (see Dawid, 1991; Rissanen, Speed, & Yu, 1992); (iii) in the context of linear least-squares regression (see Rissanen, 1986a); and (iv) in the context of nonlinear ARMA order selection using neural networks (see Rissanen, 1994; Lehtokangas et al., 1996).

Throughout this section, we let the sigmoidal function ϕ to be the logistic sigmoidal function, namely

$$\phi(u) = \frac{1}{1+\exp(-u)}.$$

3.1. Learning a smooth regression function

We generated $n = 300$ independent samples $\{X_i, Y_i\}_{i=1}^n$, where, for $i = 1, 2, \ldots, n$, each X_i was uniformly distributed in the interval $[-1, 1]$ and each

$$Y_i = s(X_i) + \sigma Z_i,$$

where $\sigma = 0.15$, $Z_i \sim \mathcal{N}(0, 1)$, and the regression function s was selected to be a fifth-degree polynomial

$$s(x) = 7(x-1)(x-0.5)(x-0.25)(x+0.5)(x+1). \tag{26}$$

Clearly, $E[Y_i \mid X_i] = s(X_i)$. Our goal is to learn the true regression function s from the observations $\{X_i, Y_i\}_{i=1}^n$. We plot the observations $\{X_i, Y_i\}_{i=1}^n$ and the regression function s in figure 3.

We compute the prequential regression estimator by proceeding as in figure 1 with the set of dimensions $\mathcal{M}_n \equiv \mathcal{M}_{300} = \{1, 2, \ldots, 20\}$ and the set of parametric models in (7). To reduce the amount of computation, we divided the set of observations $\{X_i, Y_i\}_{i=1}^n$ into 12 consecutive blocks each of size $\ell = 25$. Specifically, we use the following modified "block" prequential estimation scheme:

for $m := 1$ **to** M_n **step** 1
  Choose a fixed initial estimator $\hat{s}_{(m,0)} \in S_m$
  $\mathrm{PREQ}(m, \ell) := \sum_{k=1}^{\ell} [Y_k - \hat{s}_{(m,0)}(X_k)]^2$
  for $j := 2\ell$ **to** n **step** ℓ
    compute the least-squares estimator
      $\hat{s}_{(m,j-\ell)} := \arg\min_{g \in S_m} \left\{ \sum_{i=1}^{j-\ell} [Y_i - g(X_i)]^2 \right\} \in S_m$
    update the prequential loss
      $$\mathrm{PREQ}(m, j) := \mathrm{PREQ}(m, j-\ell) + \sum_{k=1}^{\ell} \left[Y_{j-\ell+k} - \hat{s}_{(m,j-\ell)}(X_{j-\ell+k}) \right]^2$$
  endfor;
endfor;

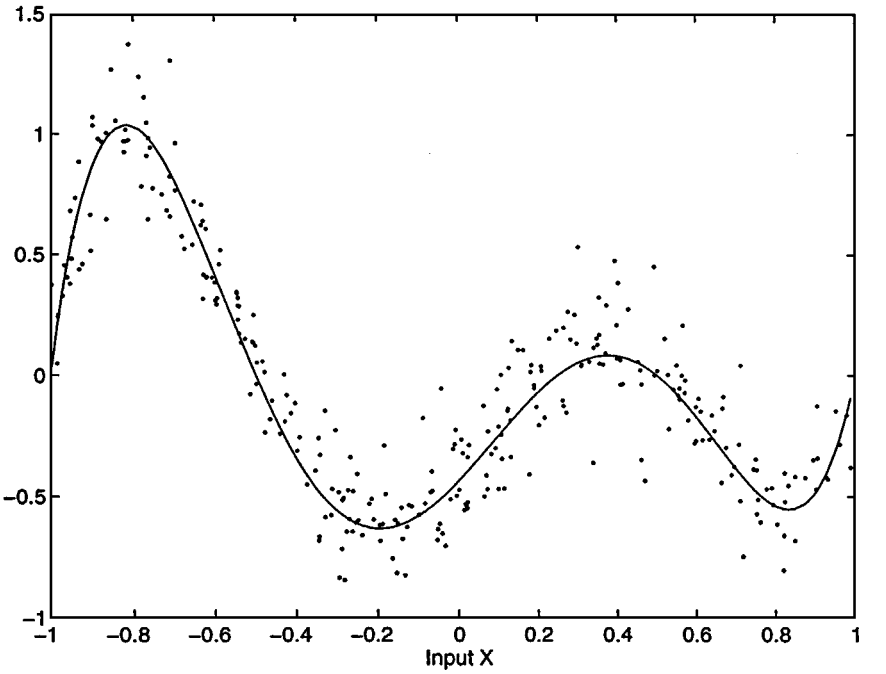

Figure 3. Noisy observations and the true regression function.

The computations were done using the neural networks toolbox in MATLAB, which includes a routine ("initff") for selecting the initial estimator $\hat{s}_{(m,0)}$. The least-squares step in the above estimation scheme was computed using traditional error backpropagation with learning rate $= (0.15)/(j - \ell)$. In other words, the learning rate was lowered with the increasing number of observations. Furthermore, to compute $\hat{s}_{(m,j-\ell)}$ we used $\hat{s}_{(m,j-2\ell)}$ as the initial starting point for the backpropagation procedure.

In figure 4, we plot the prequential loss PREQ(m, n) and the least-squares loss $\sum_{i=1}^{n}[Y_i - \hat{s}_{(m,n)}(X_i)]^2$ as a function of the number of hidden units m. It can be seen from figure 4 that the least-squares loss essentially decreases with increasing m, and hence is not a good yardstick for model selection. On the other hand, the prequential loss decreases initially and then increases with increasing m. The prequential loss achieves a clear minimum at $\hat{m}_n^{(p)} = 6$.

In figure 5, we plot the true regression function and the prequential regression estimator corresponding to $\hat{m}_n^{(p)}$. It can be seen from figure 5 that there is an excellent agreement between the true regression function and the prequential regression estimator.

We compute coefficients $\{\alpha_{m,n}\}_{m=1}^{20}$ of the prequential mixture regression estimator by proceeding essentially as in (19), (20), and (21). (In fact, we slightly modified these equations to reflect that block size $\ell = 25$. For the sake of brevity, we omit the presentation of the modified estimation scheme.) We plot $\{\alpha_{m,n}\}_{m=1}^{20}$ in figure 6. It can be seen that the

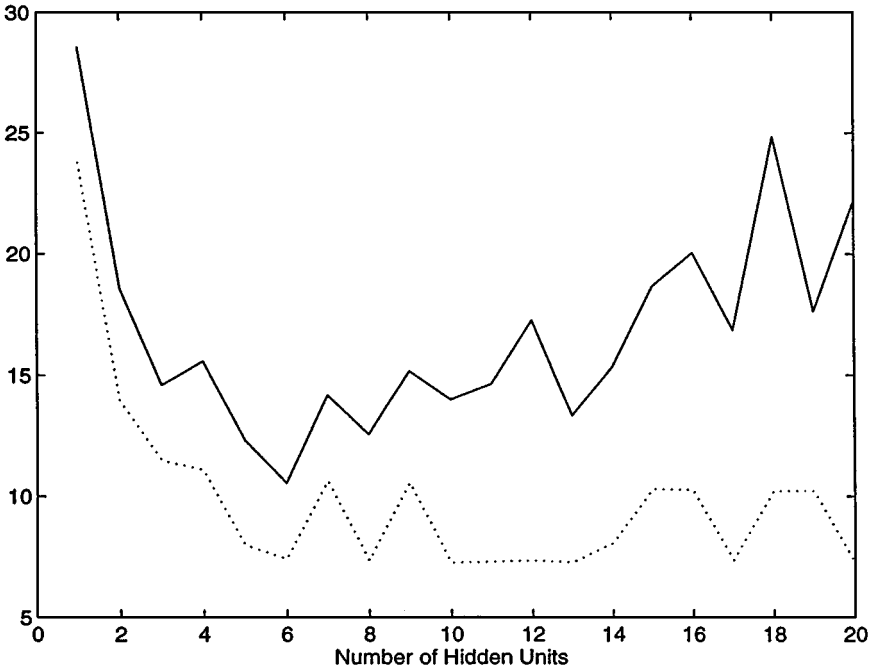

Figure 4. Prequential loss (solid line) and least-squares loss (dotted line).

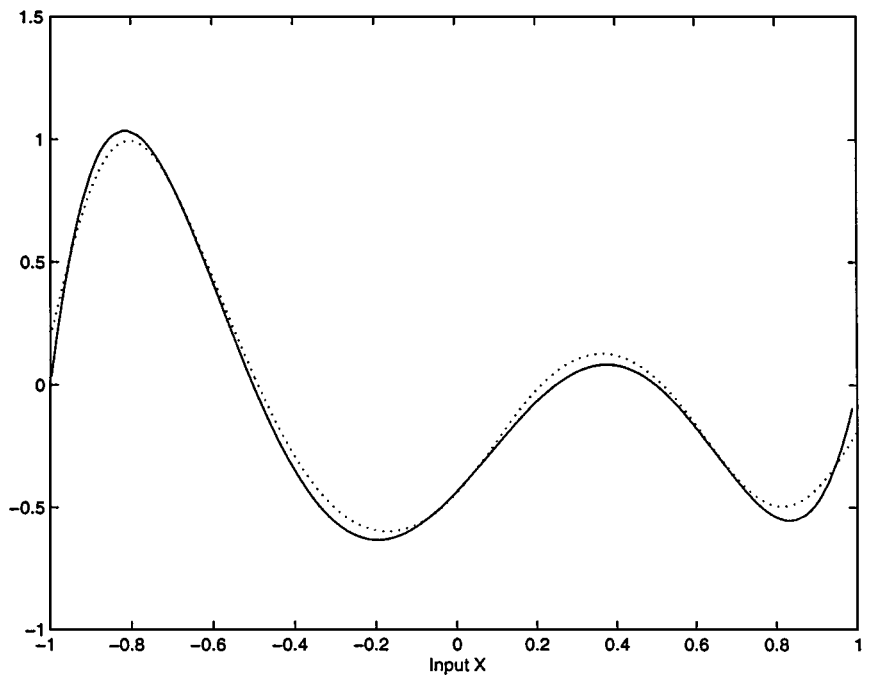

Figure 5. The true regression function (solid line) and the prequential regression estimator (dotted line). In this case, the prequential mixture regression estimator is identical to the prequential regression estimator.

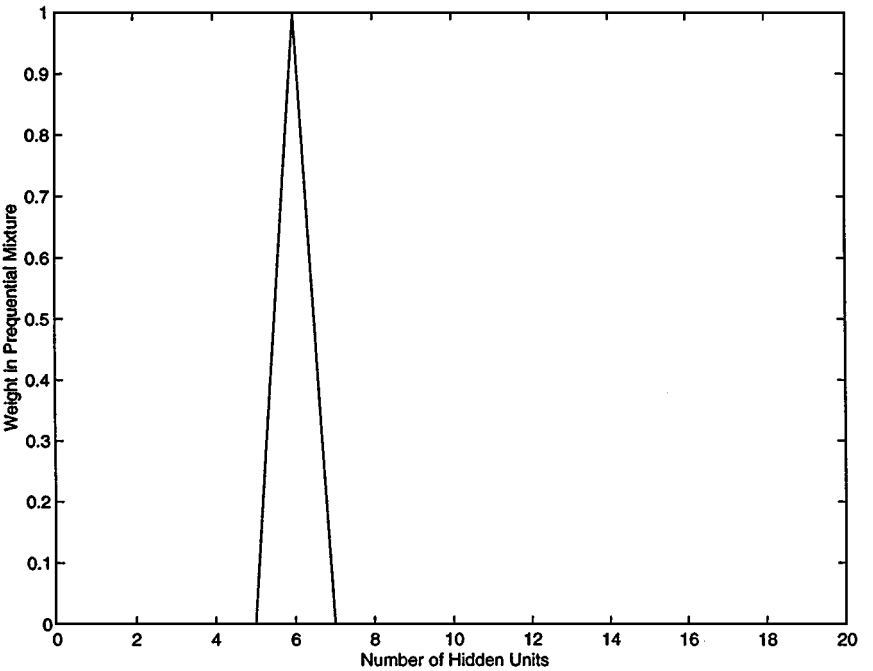

Figure 6. Weights of various hidden units in the prequential mixture regression estimator.

prequential mixture regression estimator assigns all the weight to the neural network with $m = 6$ hidden units. Consequently, in this case, the prequential mixture regression estimator is identical to the prequential regression estimator.

We now compute the cross-validated regression estimator by proceeding as in figure 2 with the set of dimensions $\mathcal{M}_n \equiv \mathcal{M}_{300} = \{1, 2, \ldots, 20\}$ and the set of parametric models as in (7). To reduce the amount of computation involved, we divided the set of observations $\{X_i, Y_i\}_{i=1}^n$ into 12 consecutive blocks each of size $\ell = 25$. Specifically, we use the following modified "block" cross-validated estimation scheme:

for $m := 1$ **to** M_n **step** 1
  for $j := \ell$ **to** n **step** ℓ
    delete observations $\{X_k, Y_k\}_{k=j-\ell+1}^{j}$ from $\{X_i, Y_i\}_{i=1}^n$
    compute the least-squares estimator based on $\{X_i, Y_i\}_{i=1, i \notin \{j-\ell+1, \cdots, j\}}^n$
      $\hat{s}_{(m,n-1)}^{(j-\ell+1,\cdots,j)} := \arg\min_{g \in S_m} \left\{ \sum_{i=1, i \notin \{j-\ell+1,\cdots,j\}}^n [Y_i - g(X_i)]^2 \right\} \in S_m$
  endfor;
  compute the cross-validation loss
    $\mathrm{CV}(m, n) := \sum_{j'=1, j=j'\ell}^{\lfloor n/\ell \rfloor} \sum_{k=1}^{\ell} [Y_{j-\ell+k} - \hat{s}_{(m,n-\ell)}^{(j-\ell+1,\cdots,j)}(X_{j-\ell+k})]^2$
endfor;

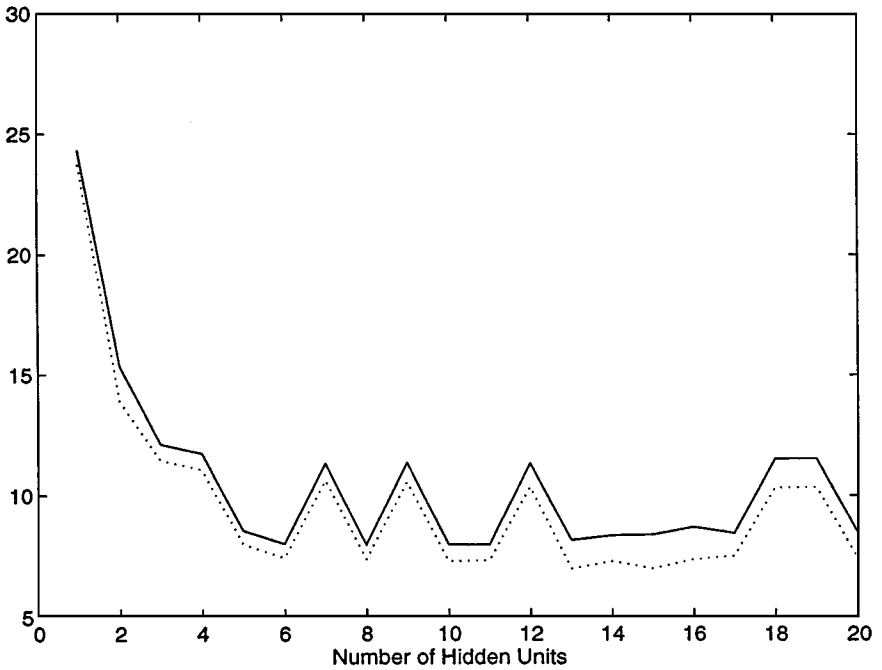

Figure 7. Cross-validation loss (solid line) and least-squares loss (dotted line).

The computations were done using the neural networks toolbox in MATLAB. The least-squares step in the above estimation scheme was computed using traditional error back-propagation with learning rate $= (0.15)/(n - \ell)$. For each fixed dimension m, the same initial starting weight was used for all different j. In figure 7, we plot the cross-validation loss $\mathrm{CV}(m, n)$ and the least-squares loss $\sum_{i=1}^{n}[Y_i - \hat{s}_{(m,n)}(X_i)]^2$ as a function of the number of hidden units m. Although not evident to the naked eye from figure 7, the cross-validation loss achieves a rather shallow minimum at $\hat{m}_n^{(c)} = 8$. Please see Remark 3.1 for a comparison of figures 4 and 7.

In figure 8, we plot the true regression function and the cross-validated regression estimator corresponding to $\hat{m}_n^{(c)}$. It can be seen from figure 8 that there is a fairly good agreement between the true regression function and the cross-validated regression estimator.

We compute coefficients $\{\beta_{m,n}\}_{m=1}^{20}$ of the cross-validated mixture regression estimator by proceeding essentially as in (23), (24), and (25). (In fact, we slightly modified these equations to reflect that block size $\ell = 25$. For the sake of brevity, we omit the presentation of the modified estimation scheme.) We plot $\{\beta_{m,n}\}_{m=1}^{20}$ in figure 9. It can be seen that the cross-validated mixture regression estimator assigns a large weight to the neural network with $m = 8$ hidden units and relatively small weights to the rest.

The cross-validated mixture regression estimator corresponding to coefficients $\{\beta_{m,n}\}_{m=1}^{20}$ is plotted in figure 8. Once again, there is an excellent agreement between the true regression function and the cross-validated mixture regression estimator.

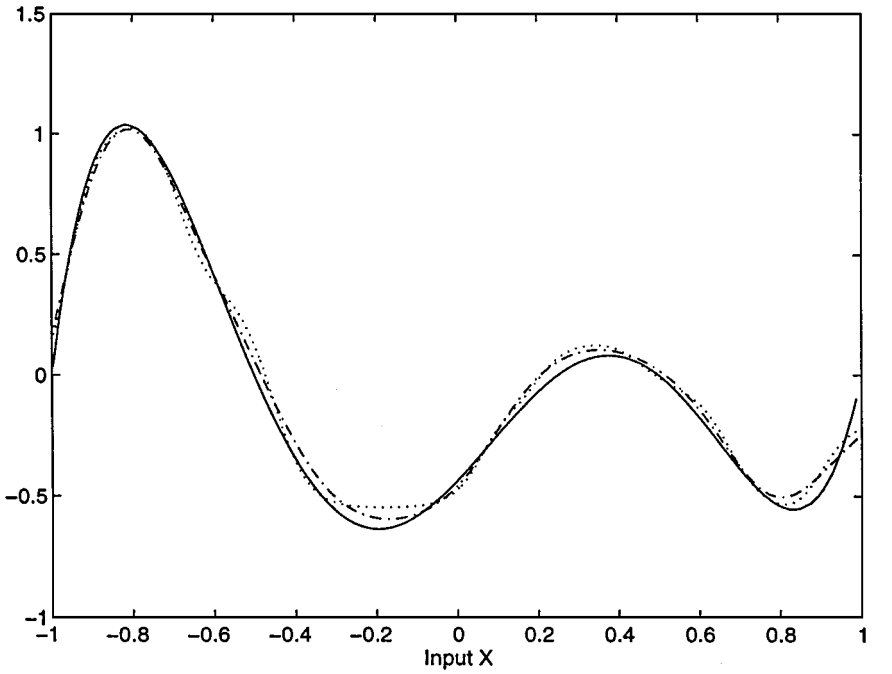

Figure 8. The true regression function (solid line), the cross-validated regression estimator (dotted line), and the cross-validated mixture regression estimator (dash-dotted line).

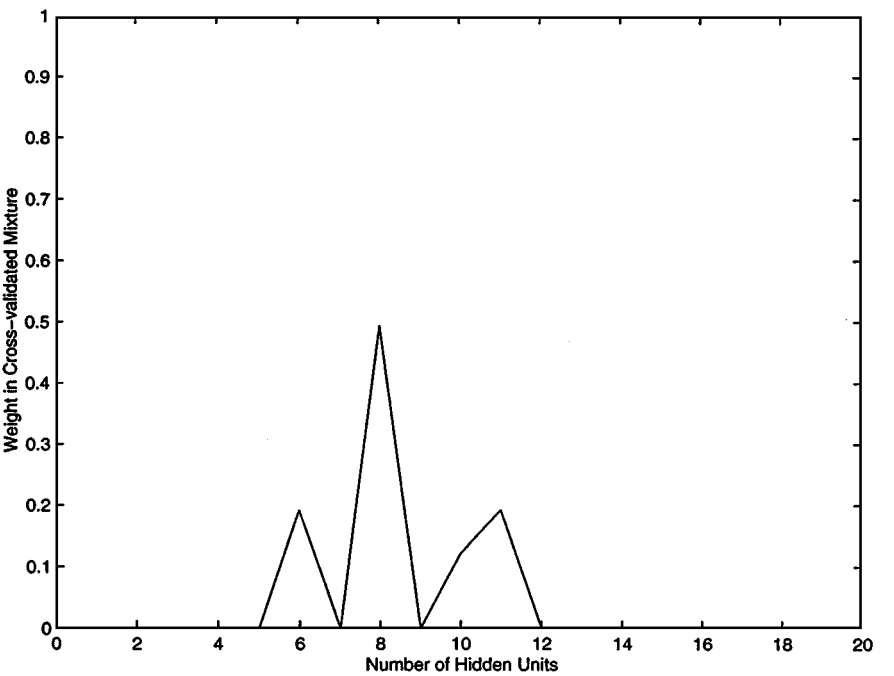

Figure 9. Weights of various hidden units in the cross-validated mixture regression estimator.

3.2. Learning a piecewise continuous regression function

We now present simulation results for a piecewise continuous regression function with two discontinuities, namely

$$s'(x) = \begin{cases} s(x) - 1 & \text{if } -1 \le x \le -0.2, \\ s(x) + 1 & \text{if } -0.2 < x \le 0.4, \\ s(x) - 1 & \text{if } 0.4 < x \le 1, \end{cases} \tag{27}$$

where the function s is as in (26). We generated $n = 300$ independent samples $\{X_i, Y_i\}_{i=1}^n$, where, for $i = 1, 2, \ldots, n$, each X_i was uniformly distributed in the interval $[-1, 1]$ and each

$$Y_i = s'(X_i) + \sigma Z_i,$$

where $\sigma = 0.15$, $Z_i \sim \mathcal{N}(0, 1)$, and s' is as in (27). We plot the observations $\{X_i, Y_i\}_{i=1}^n$ and the regression function s' in figure 10. We computed the prequential regression estimator, the prequential mixture regression estimator, the cross-validated regression estimator, and the cross-validated mixture regression estimator by proceeding exactly as in the previous subsection except with $\mathcal{M}_n \equiv \mathcal{M}_{300} = \{1, 2, \ldots, 18\}$.

We found that the behavior of the prequential loss and the cross-validation loss was essentially the same as that depicted in figures 4 and 7, respectively. We omit the plots for

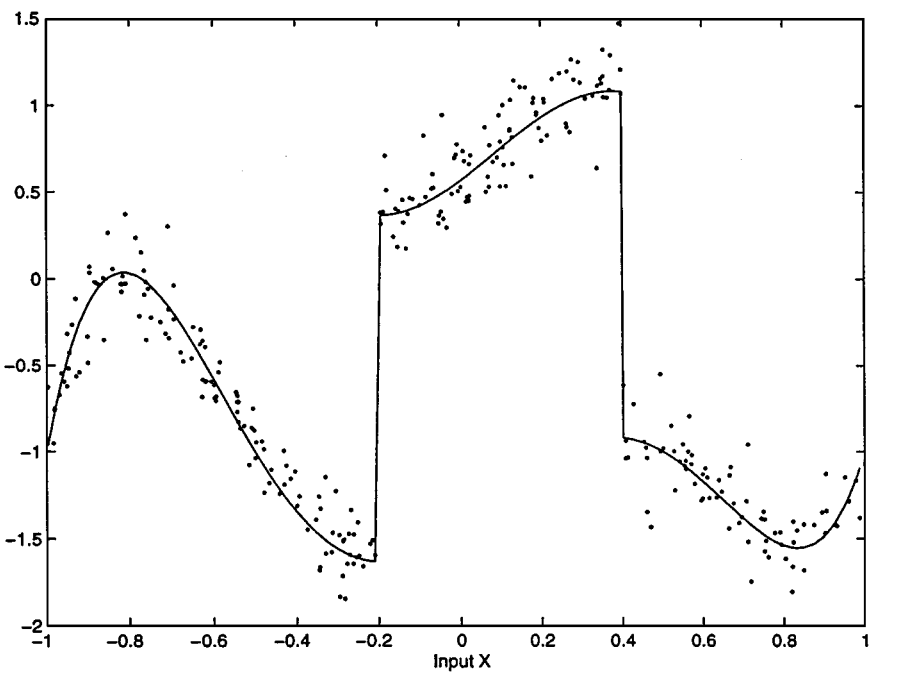

Figure 10. Noisy observations and the true regression function.

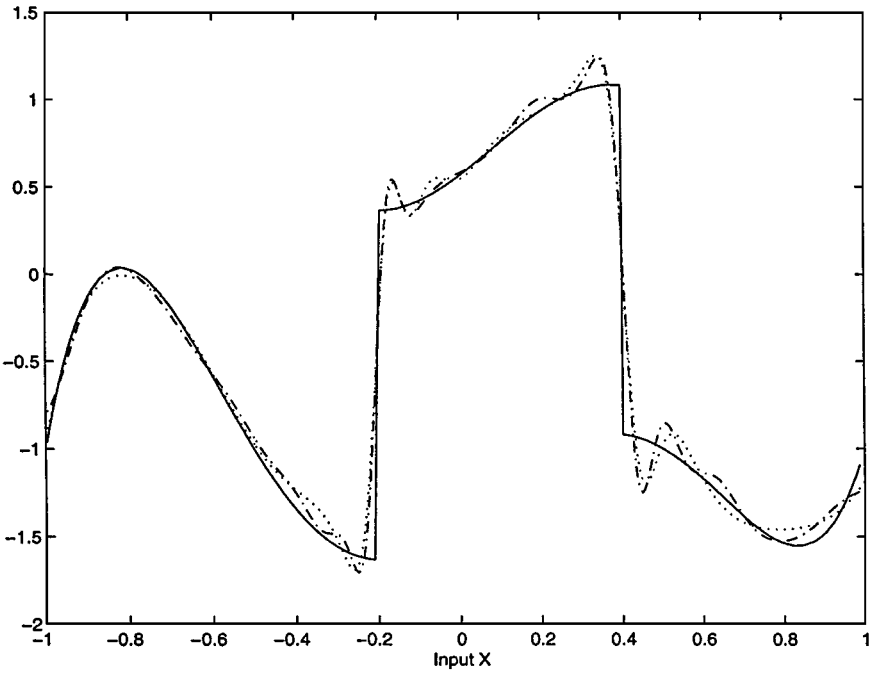

Figure 11. The true regression function (solid line), the prequential regression estimator (dotted line), and the cross-validated regression estimator (dash-dotted line).

brevity. The prequential loss achieves a minimum at $\hat{m}_n^{(p)} = 12$, while the cross-validation loss achieves a minimum at $\hat{m}_n^{(c)} = 18$.

In figure 11, we plot the true regression function, the prequential regression estimator corresponding to $\hat{m}_n^{(p)}$, and the cross-validated regression estimator corresponding to $\hat{m}_n^{(c)}$. It can be seen from figure 11 that there is an excellent agreement between the true regression function and both the estimators.

In figures 12 and 13, we plot the coefficients $\{\alpha_{m,n}\}_{m=1}^{18}$ and the coefficients $\{\beta_{m,n}\}_{m=1}^{18}$, respectively. It can be seen from figure 12 that the prequential mixture regression estimator assigns all the weight to the neural network with $m = 12$ hidden units. Also, it can be seen from figure 13 that the cross-validated mixture regression estimator assigns a large weight to the neural network with $m = 16$ hidden units and relatively small weights to the rest.

3.3. Discussion

Remark 3.1 (*comparisons between prequential and cross-validated regression estimators*). Theoretically, both the prequential and the cross-validated regression estimators enjoy similar rates of convergence (see Theorem 2.1 and Corollary 2.1). Also, empirically, it can be

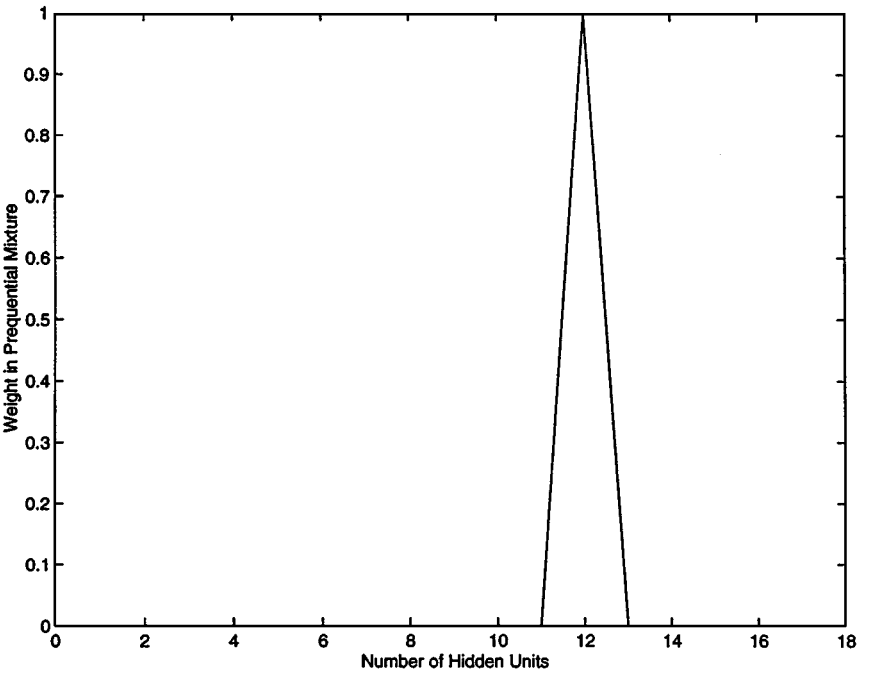

Figure 12. Weights of various hidden units in the prequential mixture regression estimator.

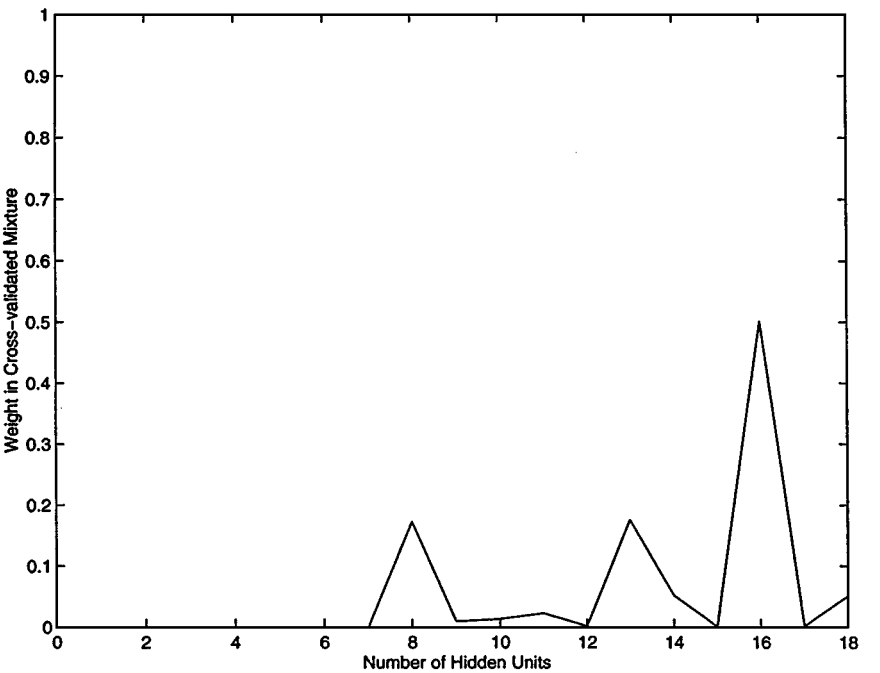

Figure 13. Weights of various hidden units in the cross-validated mixture regression estimator.

seen from figures 5, 8, and 11 that in both the examples there is an excellent agreement between the true regression function and both the estimators.

Nonetheless, it can be seen from figures 4 and 7, figures 6 and 9, and also from figures 12 and 13 that prequential model selection generates a clear and unequivocal estimate of the model dimension while cross-validation does not. For example, in figure 7, the cross-validation loss achieved by the neural network with $m = 20$ hidden units is only marginally higher than that achieved by the neural network with $m = 8$ hidden units. However, the estimator corresponding to $m = 20$ is a considerably poorer estimate of the true regression function than the estimator corresponding to $m = 8$. This leads us to conjecture that prequential model selection is order-consistent and that cross-validation is not order-consistent. Furthermore, in the above examples, we found that computing the cross-validated regression estimator was roughly ℓ-times (where $\ell = 12$) more expensive than computing the prequential regression estimator. In this light, we believe that in practice prequential model selection is more reliable and useful than cross-validation.

Remark 3.2. Observe that Theorem 2.1 assumes that the block size $\ell = 1$, whereas, in this section, we used a computationally more convenient block size $\ell = 25$. A close examination of our proofs reveals that the rates of convergence results established in Theorem 2.1 continue to hold if ℓ grows slower than n.

Remark 3.3. Statistically speaking, it would be desirable to generate a number of independent data sets, and to compute the prequential and the cross-validated regression estimators for each of the data set. One would then plot average results over these multiple (Monte Carlo) runs—and not simply plot results over a single data set as we do. However, the training of neural networks being computationally expensive, we limit our experiments to a single data set. This is not an uncommon practice in neural network literature. Moreover, it can be argued that, in practice the user will most often have only one set of observations.

Remark 3.4. Consider the following generic problem. Given a fixed dimension m and observations $\{X_i, Y_i\}_{i=1}^n$, we are interested in computing the least-squares estimator

$$\hat{s}_{(m,n)} = \arg\min_{g \in S_m} \left\{ \sum_{i=1}^{n} [Y_i - g(X_i)]^2 \right\}.$$

Directly computing $\hat{s}_{(m,n)}$ using traditional error backpropagation is a computationally expensive exercise. We empirically found that the following procedure—inspired by the prequential approach—is computationally much more efficient.

Choose a fixed initial estimator $\hat{s}_{(m,0)} \in S_m$
for $j := 2\ell$ **to** n **step** ℓ
    compute the least-squares estimator starting from $\hat{s}_{(m,j-2\ell)}$
        $\hat{s}_{(m,j-\ell)} := \arg\min_{g \in S_m} \left\{ \sum_{i=1}^{j-\ell} [Y_i - g(X_i)]^2 \right\} \in S_m$
endfor;

where we lower the learning rate of the backpropagation procedure with the increasing sample size j. Intuitively, the sequential least-squares minimization scheme outlined above is a heuristic for successively refining the error surface on which gradient descent is performed.

4. Abstract estimation framework and derivations

In this section, we estimate the regression function s using prequential and cross-validated estimators based on an abstract list of parametric models, say $\{S_m\}_{m\in\mathcal{M}_n}$, where $\mathcal{M}_n$ is allowed to grow as a function of the sample size n. We establish deterministic upper bounds on the integrated mean-squared errors of these estimators in Theorem 4.1. Theorem 4.1 is then employed to establish Theorem 2.1. This approach leads to shorter proofs and simultaneously permits a considerable degree of generality. For example, Theorem 4.1 is not limited to neural networks, but may also apply to wavelets, polynomials, splines, and Fourier series.

We now briefly sketch the main ideas involved in the proof of Theorem 4.1. Following Vapnik (1982, 1995), exponential probability bounds derived from Bernstein and Hoeffding inequalities (Hoeffding, 1963) have become a standard tool in obtaining rates of convergence results for nonparametric estimators, see, for example, Barron (1991), Barron, Birgé, & Massart (1996), Haussler (1992), Lugosi & Nobel (1995), McCaffrey & Gallant (1994), and Modha & Masry (1996, 1997). Here, we utilize one such exponential probability bound (see (42)) derived in Barron, Birgé, & Massart (1996)—referred to as BBM hereafter. It should be noted that our results are not an obvious consequence of the result in BBM. To be sure, their work is focussed entirely on establishing rates of convergence results for various complexity-regularized estimators; they considered neither the prequential nor the cross-validated regression estimators. We now briefly explain the purpose and the mode of action of various technical assumptions and highlight the main steps leading to the proof of Theorem 4.1 which is quite lengthy.

1. The exponential bound (see (42)) requires that for each fixed $m \in \mathcal{M}_n$, the set S_m is not too "fat." This condition is precisely captured in Assumptions 4.1 and 4.2. The exponential bounds are derived in BBM using the classical Bernstein inequality (Hoeffding, 1963). The boundedness condition required by the Bernstein inequality is furnished in Assumptions 2.2 and 4.3.
2. We harness the exponential bounds in (42) for the analysis of prequential regression estimators in (47) by using the fact that the least-squares loss is upper bounded by the prequential loss (see Lemma 4.1). Also, see Lemma 4.2 where we establish that the least-squares loss is upper bounded by the cross-validation loss.
   Figures 4 and 7 serve as an empirical validation of Lemmas 4.1 and 4.2, respectively.
3. The technical condition in Assumption 4.4 allows us to employ a simple probability inequality in Lemma A.6 of Modha and Masry (1996) to derive the bounds in (49).
4. Finally, we exploit special structures of the prequential loss and the cross-validation loss, respectively, in sequences of inequalities (51) and (53) to complete the proof of Theorem 4.1.

Having derived the abstract bounds in Theorem 4.1, we adapt these bounds to estimators based on neural networks by employing Example 4.1 (see (54) and step (a) of (57)). Finally, since Assumptions 2.1, 2.3, and 2.4 hold, by appealing to the approximation error bounds derived in Corollary 1 of Barron (1994) in step (b) of (57), we complete the proof of Theorem 2.1.

4.1. Abstract estimation framework

We introduce an abstract sequence of finite-dimensional models. Let $\mathcal{M}_n$ denote an arbitrary collection of model dimensions. For each fixed dimension $m \in \mathcal{M}_n$, let $S_m \subset L^2(P_X)$ denote a finite-dimensional parametric family of functions, where P_X denotes the marginal distribution of X_0. For example, see (7) and (8). D_m denotes the number of parameters necessary to describe the elements of S_m in the following precise sense.

Assumption 4.1. *For each sample size n and for each dimension m, there exists constants* $1 \leq B_{m,n} < \infty$, $1 \leq D_m < \infty$, *and* $0 < r_m < \infty$ *such that for each* $\delta > 0$ *and for each ball* $\mathcal{B} \subset S_m$ *with radius* $\sigma \geq [5\delta \vee (D_m/n)^{1/2}]$ *there exists a finite set* $T = T(m, \delta, \mathcal{B}) \subset \mathcal{B}$ *with*

$$\text{cardinality}(T) \leq (B_{m,n}\sigma/\delta)^{D_m}$$

such that

$$\sup_{g \in \mathcal{B}} \inf_{f \in T} \|g - f\|_\infty \leq r_m \delta.$$

Assumption 4.1 is essentially Assumption $\mathrm{M}'_{2,\infty}$ of BBM specialized to least-squares regression estimation. More generally, it may be possible to replace Assumption 4.1 by Assumption M of BBM, however, for the sake of simplicity and brevity we do not pursue such generalizations here.

Example 4.1 (*neural networks*). It follows from Sections 3.2.2 and 5.5 of BBM that Assumption 4.1 holds for the parametric family of functions S_m based on neural networks (see (6) and (7)) with

$$r_m = 1, \ B_{m,n} = (8eC)\left[\sqrt{n/D_m} \vee (1/5)\right], \text{ and } D_m = 1 + m(q+2). \tag{28}$$

For each $m \in \mathcal{M}_n$, let

$$\mathcal{L}_m = \frac{5}{2} \ln\left[12 B_{m,n}(1 + r_m\sqrt{D_m/n})\right]. \tag{29}$$

A certain factor L_m, which is roughly the logarithm of the number of parametric families with dimension D_m included in the set $\{S_m\}_{m \in \mathcal{M}_n}$, will also be needed below.

Assumption 4.2 (Assumption S of BBM). *There exists a family of weights* $\{L_m\}_{m\in\mathcal{M}_n}$ *such that* $1 \le L_m < \infty$ *for each* $m \in \mathcal{M}_n$, *and*

$$\sum_{m\in\mathcal{M}_n} \exp[-L_m D_m] = \Sigma < \infty.$$

Example 4.1 (*continued*). Assumption 4.2 holds for S_m based on neural networks with

$$L_m = 1. \tag{30}$$

Assumption 4.3. *The elements of* $\mathcal{S}_m$ *are uniformly bounded by* ξ, *where* ξ *is as in Assumption* 2.2.

Example 4.1 (*continued*). It follows from (7) that Assumption 4.3 holds for neural networks.

Assumption 4.4. *For each fixed* $n \ge 1$, *the set of dimensions* $\mathcal{M}_n$ *is such that*

$$\sup_{m\in\mathcal{M}_n} D_m < \infty, \quad \sup_{m\in\mathcal{M}_n} L_m < \infty, \quad \sup_{m\in\mathcal{M}_n} \mathcal{L}_m < \infty.$$

Example 4.1 (*continued*). It follows from (8), (9), (10), (28), (30), and (29) that Assumption 4.4 holds for neural networks.

Define abstract prequential estimator $\hat{s}_n^{(p)}$ as in (1) and (2) (see figure 1) and abstract cross-validated estimator $\hat{s}_n^{(c)}$ as in (3) and (4) (see figure 2), where $\{S_m\}_{m\in\mathcal{M}_n}$ represents a sequence of abstract finite-dimensional models. We now establish deterministic upper bounds on the integrated mean-squared errors of these estimators.

Theorem 4.1. *Suppose that Assumptions* 2.2, 4.1, 4.2, 4.3, *and* 4.4 *hold. For each* m, *let* s_m *be such that* $d^2(s_m, s) = \inf_{t\in S_m} d^2(t, s)$. *Let* κ_5, κ_6, *and* λ *denote positive constants. Then, for each* $n \ge 2$, *the following upper bounds hold.*

(**prequential regression estimation**)

$$\begin{aligned} E\int_{\mathbb{R}^q} \left[s(x) - \hat{s}_n^{(p)}(x)\right]^2 dP_X(x) \\ \le \inf_{m\in\mathcal{M}_n} \left\{\kappa_5 d^2(s_m, s) + [\kappa_6(2+\mathcal{L}_m)(\ln(n-1)+1) + (\lambda\mathcal{L}_m \vee 1) + \lambda L_m]\frac{D_m}{n}\right\} \\ + E\left[[(\lambda\mathcal{L}_{\hat{m}} \vee 1) + \lambda L_{\hat{m}}]\frac{D_{\hat{m}}}{n}\right] + \frac{8\xi^2 + 4.1\Sigma\lambda}{n}, \end{aligned} \tag{31}$$

where $\hat{m} = \hat{m}^{(p)}$ *and* $\hat{s}_n^{(p)} = \hat{s}_{(\hat{m},n)}$.

(cross-validated regression estimation)

$$E\int_{\mathbb{R}^q}\left[s(x)-\hat{s}_n^{(c)}(x)\right]^2 dP_X(x)$$
$$\leq \inf_{m\in\mathcal{M}_n}\left\{\kappa_5 d^2(s_m,s)+[2\kappa_6(2+\mathcal{L}_m)+(\lambda\mathcal{L}_m\vee 1)+\lambda L_m]\frac{D_m}{n}\right\}$$
$$+E\left[[(\lambda\mathcal{L}_{\hat{m}}\vee 1)+\lambda L_{\hat{m}}]\frac{D_{\hat{m}}}{n}\right]+\frac{4.1\Sigma\lambda}{n}, \tag{32}$$

where $\hat{m}=\hat{m}_n^{(c)}$ *and* $\hat{s}_n^{(c)}=\hat{s}_{(\hat{m},n)}$.

The proof can be found in the next sub-section.

Remark 4.1. For the reader's convenience, we note that our symbols $B_{m,n}$, $\mathcal{L}_m$, and $\chi^2(m,m')$ correspond, respectively, to the symbols B'_m, $\mathcal{L}'_m$, and $x^2(m,m')$ in BBM.

4.2. Derivations

For the sake of brevity, throughout this subsection, we write $Z_j=(X_j,Y_j)$, $j=1,2,\ldots,n$, write

$$\gamma(Z_j,g)=[Y_j-g(X_j)]^2, \tag{33}$$

write

$$\gamma_n(g)=n^{-1}\sum_{j=1}^{n}\gamma(Z_j,g), \tag{34}$$

and write

$$d^2(g,f)=\int_{\mathbb{R}^d}[g(x)-f(x)]^2\,dP_X(x), \tag{35}$$

where g and f are functions in $L^2(P_X)$. The following two simple lemmas are important steps in establishing Theorem 4.1.

Lemma 4.1. *For each* $n\geq 1$ *and for each* m,

$$\sum_{j=1}^{n}\gamma\left(Z_j,\hat{s}_{(m,n)}\right)\leq\sum_{j=1}^{n}\gamma\left(Z_j,\hat{s}_{(m,j-1)}\right).$$

Proof: Observe that for any $i\geq 1$, since $\hat{s}_{(m,n)}$ minimizes the least-squares error $\sum_{j=1}^{n}\gamma(Z_j,\cdot)$ we have that

$$\sum_{j=1}^{i}\gamma\left(Z_j,\hat{s}_{(m,i)}\right)\leq\sum_{j=1}^{i-1}\gamma\left(Z_j,\hat{s}_{(m,i-1)}\right)+\gamma\left(Z_i,\hat{s}_{(m,i-1)}\right). \tag{36}$$

The lemma follows by applying (36) n-times with $i=n,n-1,\ldots,1$. □

Lemma 4.2. *For each $n \geq 2$ and for each m,*

$$\sum_{j=1}^{n} \gamma\big(Z_j, \hat{s}_{(m,n)}\big) \leq \sum_{j=1}^{n} \gamma\big(Z_j, \hat{s}^{(j)}_{(m,n-1)}\big).$$

Proof: Observe that for any $1 \leq i \leq n$, since $\hat{s}_{(m,n)}$ minimizes the least-squares error $\sum_{j=1}^{n} \gamma(Z_j, \cdot)$ we have that

$$\sum_{j=1}^{n} \gamma\big(Z_j, \hat{s}_{(m,n)}\big) \leq \sum_{j=1}^{n} \gamma\big(Z_j, \hat{s}^{(i)}_{(m,n-1)}\big), \tag{37}$$

and since $\hat{s}^{(i)}_{(m,n-1)}$ minimizes the least-squares error $\sum_{j=1, j\neq i}^{n} \gamma(Z_j, \cdot)$ we have that

$$\sum_{j=1, j\neq i}^{n} \gamma\big(Z_j, \hat{s}^{(i)}_{(m,n-1)}\big) \leq \sum_{j=1, j\neq i}^{n} \gamma\big(Z_j, \hat{s}_{(m,n)}\big). \tag{38}$$

And, now it follows from (37) and (38) that

$$\gamma\big(Z_i, \hat{s}_{(m,n)}\big) \leq \gamma\big(Z_i, \hat{s}^{(i)}_{(m,n-1)}\big). \tag{39}$$

The lemma now follows by applying (39) n-times with $i = 1, 2, \ldots, n$. □

Proof of Theorem 4.1: We first establish bounds on the risk of the prequential estimator $\hat{s}^{(p)}_n = \hat{s}_{(\hat{m}^{(p)},n)}$. For the sake of brevity, we write $\hat{s} = \hat{s}^{(p)}_n$ and $\hat{m} = \hat{m}^{(p)}$. Let a fixed dimension $m \in \mathcal{M}_n$ be a given. For any $m' \in \mathcal{M}_n$, write

$$n\chi^2(m, m') = \theta + \big(n\sigma_m^2 \vee n\sigma_{m'}^2\big) \vee \lambda(L_m D_m \vee L_{m'} D_{m'}) \tag{40}$$

where $\theta \geq 0$, L_m and $L_{m'}$ are as in Assumption 4.2, D_m and $D_{m'}$ are as in Assumption 4.1, and σ_m^2 and $\sigma_{m'}^2$ are obtained from

$$\sigma_m^2 = [\lambda \mathcal{L}_m \vee 1]\frac{D_m}{n}, \tag{41}$$

where $\lambda > 0$ is an appropriate constant arising in Proposition 7 of BBM. Precise value of λ is not important in implementing the estimators considered in this paper.

By proceeding as in the proof of Theorem 9 in BBM, it can be checked that our Assumptions 2.2 and 4.3 imply that Assumption Lip of BBM holds.

Now, since our Assumptions 4.1 and 4.2, and Assumption Lip of BBM hold, we have from Eq. (5.17) of BBM that, for any fixed $s_m \in S_m$,

$$P\left\{ \sup_{m' \in \mathcal{M}_n} \sup_{g \in S_{m'}} \frac{\{\gamma_n(s_m) - E[\gamma_n(s_m)]\} - \{\gamma_n(g) - E[\gamma_n(g)]\}}{d^2(s, g) \vee d^2(s_m, s) \vee \chi^2(m, m')} > \frac{1}{2} \right\} \leq 4.1\Sigma \exp(-\theta/\lambda), \tag{42}$$

where $\gamma_n(s_m)$ and $\gamma_n(g)$ are obtained from (34) and $d^2(s,g)$ and $d^2(s_m,s)$ are obtained from (35). For completeness, we note that (42) is obtained from Eq. (5.17) of BBM by suitably adapting the latter to the specific case of least-squares regression estimation. Since (42) holds for any fixed $s_m \in S_m$, from now onwards let s_m be such that $d^2(s_m,s) = \inf_{t\in S_m} d^2(t,s)$. Now, for any $f \in L^2(P_X)$,

$$\begin{aligned}
&E[\gamma_n(s) - \gamma_n(f)] \\
&= n^{-1} E \sum_{j=1}^{n} \{[Y_j - s(X_j)]^2 - [Y_j - f(X_j)]^2\} \\
&= -n^{-1} \sum_{j=1}^{n} E\{[s(X_j) - f(X_j)]^2 - 2[s(X_j) - Y_j][f(X_j) - s(X_j)]\} \\
&= -E[s(X_j) - f(X_j)]^2 + 2n^{-1} \sum_{j=1}^{n} E\{E[(s(X_j) - Y_j) \mid X_j][f(X_j) - s(X_j)]\} \\
&\stackrel{(a)}{=} -d^2(s, f). \qquad (43)
\end{aligned}$$

Thus, it follows from (43) that

$$\begin{aligned}
E[\gamma_n(g)] - E[\gamma_n(s_m)] &= E[\gamma_n(g) - \gamma_n(s)] - E[\gamma_n(s_m) - \gamma_n(s)] \\
&= d^2(s,g) - d^2(s,s_m), \qquad (44)
\end{aligned}$$

and from (42) and (44) that

$$P\left\{ \sup_{m'\in\mathcal{M}_n} \sup_{g\in S_{m'}} \frac{\gamma_n(s_m) - \gamma_n(g) + d^2(s,g) - d^2(s,s_m)}{d^2(s,g) \vee d^2(s_m,s) \vee \chi^2(m,m')} > \frac{1}{2} \right\} \le 4.1\Sigma \exp(-\theta/\lambda). \qquad (45)$$

Now, it follows from (45) with $m' = \hat{m}$ and $g = \hat{s}$ that

$$\begin{aligned}
P\{2(\gamma_n(s_m) - \gamma_n(\hat{s}) + d^2(s,\hat{s}) - d^2(s,s_m)) \ge d^2(s,\hat{s}) + d^2(s_m,s) + \chi^2(m,\hat{m})\} \\
\le 4.1\Sigma \exp(-\theta/\lambda). \qquad (46)
\end{aligned}$$

Observe that

$$\begin{aligned}
\gamma_n(\hat{s}) = \frac{1}{n}\sum_{j=1}^{n} \gamma(Z_j,\hat{s}) = \frac{1}{n}\sum_{j=1}^{n} \gamma\big(Z_j, \hat{s}_{(\hat{m},n)}\big) &\stackrel{(a)}{\le} \frac{1}{n}\sum_{j=1}^{n} \gamma\big(Z_j, \hat{s}_{(\hat{m},j-1)}\big) \\
&\stackrel{(b)}{\le} \frac{1}{n}\sum_{j=1}^{n} \gamma\big(Z_j, \hat{s}_{(m,j-1)}\big) \qquad (47)
\end{aligned}$$

where (a) follows from Lemma 4.1 and (b) follows from (1). Write

$$W_{m,n} = d^2(s,\hat{s}) - 3d^2(s_m,s) - \sigma_m^2 - \sigma_{\hat{m}}^2 - \frac{\lambda L_m D_m + \lambda L_{\hat{m}} D_{\hat{m}}}{n} - 2[\gamma_n(s) - \gamma_n(s_m)] - 2n^{-1}\sum_{j=1}^{n}\left[\gamma\left(Z_j, \hat{s}_{(m,j-1)}\right) - \gamma(Z_j,s)\right],$$

and observe that we have added and subtracted $2\gamma_n(s) = 2n^{-1}\sum_{j=1}^{n}\gamma(Z_j,s)$. We now have from (40), (46), and (47) that

$$P\{W_{m,n} \geq \theta/n\} \leq 4.1\Sigma \exp(-\theta/\lambda). \tag{48}$$

For each fixed $n \geq 1$ and for each $m \in \mathcal{M}_n$, it follows from Assumptions 2.2 and 4.3 that

$$\left| d^2(s,\hat{s}) - 3d^2(s_m,s) - 2[\gamma_n(s) - \gamma_n(s_m)] - \frac{2}{n}\sum_{j=1}^{n}\left[\gamma\left(Z_j, \hat{s}_{(m,j-1)}\right) - \gamma(Z_j,s)\right] \right| < \infty,$$

and, since $B_{m,n}$, D_m, r_m, and L_m are finite constants (see Assumptions 4.1 and 4.2), we have that

$$\sigma_m^2 + \frac{\lambda L_m D_m}{n} < \infty.$$

Also, for each fixed $n \geq 1$, we have from Assumption 4.4 that

$$\sigma_{\hat{m}}^2 + \frac{\lambda L_{\hat{m}} D_{\hat{m}}}{n} < \infty.$$

Consequently, we have that $|W_{m,n}| < F_{m,n} < \infty$ a.s. for some finite constant $F_{m,n}$. This implies that $E|W_{m,n}| < \infty$, hence we now have from Lemma A.6 of Modha & Masry (1996) and from (48) that

$$E[W_{m,n}] \leq \int_0^\infty P\{W_{m,n} \geq \theta'\}\, d\theta' \leq 4.1\Sigma \int_0^\infty \exp(-n\theta'/\lambda)\, d\theta' = \frac{4.1\Sigma\lambda}{n}.$$

More explicitly, we have that

$$E[d^2(s,\hat{s})] \leq 3d^2(s_m,s) + \sigma_m^2 + \frac{\lambda L_m D_m}{n} + E\left[\sigma_{\hat{m}}^2 + \frac{\lambda L_{\hat{m}} D_{\hat{m}}}{n}\right] + 2E[\gamma_n(s) - \gamma_n(s_m)] + \frac{2}{n}E\left[\sum_{j=1}^{n}\left[\gamma\left(Z_j, \hat{s}_{(m,j-1)}\right) - \gamma(Z_j,s)\right]\right] + \frac{4.1\Sigma\lambda}{n}. \tag{49}$$

We have from (43) that

$$E[\gamma_n(s) - \gamma_n(s_m)] = -d^2(s, s_m) \leq 0. \tag{50}$$

Also, we have that

$$\begin{aligned}
&\frac{2}{n} E \sum_{j=1}^{n} \left[\gamma\left(Z_j, \hat{s}_{(m,j-1)}\right) - \gamma(Z_j, s)\right] \\
&= \frac{2}{n} \sum_{j=1}^{n} E\left\{\left[\hat{s}_{(m,j-1)}(X_j) - s(X_j)\right]^2 - 2[s(X_j) - Y_j]\left[s(X_j) - \hat{s}_{(m,j-1)}(X_j)\right]\right\} \\
&\overset{(a)}{=} \frac{2}{n} \sum_{j=1}^{n} E\left[\hat{s}_{(m,j-1)}(X_j) - s(X_j)\right]^2 \\
&= \frac{2}{n} \sum_{j=1}^{n} E\left[E\left[\left(\hat{s}_{(m,j-1)}(X_j) - s(X_j)\right)^2 \mid X_1, X_2, \ldots, X_{j-1}\right]\right] \\
&= \frac{2}{n} \sum_{j=1}^{n} E\left[d^2\left(\hat{s}_{(m,j-1)}, s\right)\right] \\
&= \frac{2}{n} E\left[d^2\left(\hat{s}_{(m,0)}, s\right)\right] + \frac{2}{n} \sum_{j=2}^{n} E\left[d^2\left(\hat{s}_{(m,j-1)}, s\right)\right] \\
&\overset{(b)}{\leq} \frac{8\xi^2}{n} + \frac{2}{n} \sum_{j=2}^{n} \kappa_9' \left\{ d^2(s, s_m) + \kappa_9 \frac{(2 + \mathcal{L}_m) D_m}{j-1} \right\} \\
&\overset{(c)}{\leq} \frac{8\xi^2}{n} + 2\kappa_9' d^2(s, s_m) + \frac{\kappa_6 (2 + \mathcal{L}_m) D_m}{n} (1 + \ln(n-1))
\end{aligned} \tag{51}$$

where (a) follows since

$$\begin{aligned}
&E\left\{\left[s(X_j) - Y_j\right]\left[s(X_j) - \hat{s}_{(m,j-1)}(X_j)\right]\right\} \\
&= E\left\{E\left[(s(X_j) - Y_j) \mid X_1, X_2, \ldots, X_j\right]\left[s(X_j) - \hat{s}_{(m,j-1)}(X_j)\right]\right\} \\
&= E\left\{E\left[(s(X_j) - Y_j) \mid X_j\right]\left[s(X_j) - \hat{s}_{(m,j-1)}(X_j)\right]\right\} \\
&= 0;
\end{aligned}$$

(b) the bound on the first term follows from Assumptions 2.2 and 4.3 and, since Assumptions 2.2, 4.1, 4.2, and 4.3 hold, the bound on the second term follows from Theorem 9 of BBM where κ_9 and κ_9' are positive constants; (c) follows since, for each $n \geq 2$,

$$\sum_{j=2}^{n} \frac{1}{j-1} = \sum_{j=1}^{n-1} \frac{1}{j} \leq 1 + \int_1^{n-1} \frac{1}{a}\, da = 1 + \ln(n-1).$$

Also, we write $\kappa_6 = 2\kappa_9'\kappa_9$.

After simple algebraic manipulations, it follows from (41), (49), (50), and (51) that

$$
\begin{aligned}
&E\int_{\mathbb{R}^q}\left[s(x)-\hat{s}_n^{(p)}(x)\right]^2 dP_X(x)\\
&\quad\le \kappa_5 d^2(s_m,s)+[\kappa_6(2+\mathcal{L}_m)(\ln(n-1)+1)+(\lambda\mathcal{L}_m\vee 1)+\lambda L_m]\frac{D_m}{n}\\
&\qquad + E\left[[(\lambda\mathcal{L}_{\hat{m}}\vee 1)+\lambda L_{\hat{m}}]\frac{D_{\hat{m}}}{n}\right]+\frac{8\xi^2+4.1\Sigma\lambda}{n},
\end{aligned}
\tag{52}
$$

The desired bound on the risk of the prequential estimator, namely (31), now follows by observing that the left-hand side of (52) does not depend on m.

The bound on the risk of the cross-validated estimator (32) follows in a similar fashion, but by employing Lemma 4.2 instead of Lemma 4.1 and by employing (3) instead of (1) in step (47), and by observing that

$$
\begin{aligned}
&\frac{2}{n}E\sum_{j=1}^{n}\left[\gamma\left(Z_j,\hat{s}_{(m,n)}^{(j)}\right)-\gamma(Z_j,s)\right]\\
&\quad=\frac{2}{n}\sum_{j=1}^{n}E\left\{\left[\hat{s}_{(m,n)}^{(j)}(X_j)-s(X_j)\right]^2-2[s(X_j)-Y_j]\left[s(X_j)-\hat{s}_{(m,n)}^{(j)}(X_j)\right]\right\}\\
&\quad\overset{\text{(a)}}{=}\frac{2}{n}\sum_{j=1}^{n}E\left[\hat{s}_{(m,n)}^{(j)}(X_j)-s(X_j)\right]^2\\
&\quad=\frac{2}{n}\sum_{j=1}^{n}E\left[E\left[\left(\hat{s}_{(m,n)}^{(j)}(X_j)-s(X_j)\right)^2\,\middle|\,X_1,\ldots,X_{j-1},X_{j+1},\ldots,X_n\right]\right]\\
&\quad=\frac{2}{n}\sum_{j=1}^{n}E\left[d^2\left(\hat{s}_{(m,n)}^{(j)},s\right)\right]\\
&\quad\overset{\text{(b)}}{\le}\frac{2}{n}\sum_{j=1}^{n}\kappa_9'\left\{d^2(s,s_m)+\kappa_9\frac{(2+\mathcal{L}_m)D_m}{n-1}\right\}\\
&\quad=2\kappa_9' d^2(s,s_m)+\frac{\kappa_6(2+\mathcal{L}_m)D_m}{n-1}
\end{aligned}
\tag{53}
$$

where (a) follows since

$$
\begin{aligned}
&E\left\{[s(X_j)-Y_j]\left[s(X_j)-\hat{s}_{(m,n)}^{(j)}(X_j)\right]\right\}\\
&\quad=E\left\{E\left[(s(X_j)-Y_j)\,\middle|\,X_1,X_2,\ldots,X_n\right]\left[s(X_j)-\hat{s}_{(m,n)}^{(j)}(X_j)\right]\right\}\\
&\quad=E\left\{E\left[(s(X_j)-Y_j)\,\middle|\,X_j\right]\left[s(X_j)-\hat{s}_{(m,n)}^{(j)}(X_j)\right]\right\}\\
&\quad=0;
\end{aligned}
$$

(b) since Assumptions 2.2, 4.1, 4.2, and 4.3 hold the bound follows from Theorem 9 of BBM for each $n\ge 2$, where κ_9 and κ_9' are positive constants. Also, as before, write $\kappa_6=2\kappa_9'\kappa_9$.

The proof of Theorem 4.1 is now complete. □

Proof of Theorem 2.1: We first establish bound (11) on the risk of the prequential estimator. Throughout this proof, let symbols $K_1, K_2, \ldots,$ represent generic positive constants.

It follows from Assumption 2.2 and from Example 4.1 that all the hypotheses of Theorem 4.1 hold, hence we have from (31) that

$$\begin{aligned} & E \int_{\mathbb{R}^q} \left[s(x) - \hat{s}_n^{(p)}(x) \right]^2 dP_X(x) \\ & \leq \inf_{m \in \mathcal{M}_n} \left\{ \kappa_5 d^2(s_m, s) + \frac{[\kappa_6(2 + \mathcal{L}_m)(\ln(n-1)+1) + (\lambda \mathcal{L}_m \vee 1) + \lambda L_m] D_m}{n} \right\} \\ & \quad + E\left[[(\lambda \mathcal{L}_{\hat{m}} \vee 1) + \lambda L_{\hat{m}}] \frac{D_{\hat{m}}}{n} \right] + \frac{8\xi^2 + 4.1\Sigma\lambda}{n}, \end{aligned} \tag{54}$$

where, for brevity, we write $\hat{m} = \hat{m}^{(p)}$.

For each $n \geq 2$, it follows from (8) and (30) that

$$\sup_{m \in \mathcal{M}_n} L_m = 1,$$

and from (8), (28), and (29) that

$$\begin{aligned} \sup_{m \in \mathcal{M}_n} \mathcal{L}_m & \leq \sup_{m \in \mathcal{M}_n} \left\{ \frac{5}{2} \ln \left[12(8en) \sqrt{\frac{n}{1 + m(q+2)}} \left(1 + \sqrt{\frac{1 + m(q+2)}{n}} \right) \right] \right\} \\ & \leq K_2 \ln n. \end{aligned}$$

Consequently, for each $n \geq 2$, we have—after some algebraic manipulations—that

$$\sup_{m \in \mathcal{M}_n} [(\lambda \mathcal{L}_m \vee 1) + \lambda L_m] \leq K_3 \ln n, \tag{55}$$

$$\sup_{m \in \mathcal{M}_n} [\kappa_6(2 + \mathcal{L}_m)(\ln(n-1)+1)] \leq K_4 (\ln n)^2. \tag{56}$$

For brevity, write

$$M_n = \kappa_1 \sqrt{n/(\ln n)^{\kappa_2}}.$$

Now,

$$\begin{aligned} & \inf_{m \in \mathcal{M}_n} \left\{ \kappa_5 d^2(s, s_m) + \frac{[\kappa_6(2 + \mathcal{L}_m)(\ln(n-1)+1) + (\lambda \mathcal{L}_m \vee 1) + \lambda L_m] D_m}{n} \right\} \\ & \overset{(a)}{\leq} \inf_{1 \leq m \leq M_n} \left\{ \kappa_5 d^2(s, s_m) + K_5 \frac{(\ln n)^2 D_m}{n} \right\} \\ & \overset{(b)}{\leq} \inf_{1 \leq m \leq M_n} \left\{ \frac{K_7}{m} + \frac{K_6 (\ln n)^2 m}{n} \right\} \\ & \overset{(c)}{\leq} K_8 \frac{\ln n}{\sqrt{n}} \end{aligned} \tag{57}$$

where (a) follows from (55) and (56); (b) follows from Corollary 1 of Barron (1994) by utilizing Assumptions 2.1, 2.3, and 2.4, and (5) and follows from (28); and (c) follows by setting $m = \lfloor \frac{\sqrt{n}}{\ln n} \rfloor$ and by checking that, for each $n \geq 2$, $1 \leq \lfloor \frac{\sqrt{n}}{\ln n} \rfloor \leq M_n = \kappa_1 \sqrt{n/(\ln n)^{\kappa_2}}$ if we set $\kappa_2 \leq 2$ and if we set $\kappa_1 \geq 1$ (as prescribed in (9)).

Now,

$$E\left[[(\lambda \mathcal{L}_{\hat{m}} \vee 1) + \lambda L_{\hat{m}}] \frac{D_{\hat{m}}}{n}\right] \overset{(a)}{=} E\left[[(\lambda \mathcal{L}_{\hat{m}} \vee 1) + \lambda L_{\hat{m}}] \frac{1 + \hat{m}(q+2)}{n}\right]$$
$$\overset{(b)}{\leq} E[(\lambda \mathcal{L}_{\hat{m}} \vee 1) + \lambda L_{\hat{m}}] \frac{K_9 (\ln n)^{-\kappa_2/2}}{\sqrt{n}}$$
$$\overset{(c)}{\leq} K_{10} \frac{(\ln n)^{1-\kappa_2/2}}{\sqrt{n}}, \tag{58}$$

where (a) follows from (28); (b) follows, for each $n \geq 2$, since the number of hidden units $\hat{m}$ is selected to be in the range $1 \leq \hat{m} \leq \kappa_1 \sqrt{n/(\ln n)^{\kappa_2}}$; and (c) follows from (55).

The desired rate of convergence for the prequential regression estimator based on neural networks, namely (11), now follows from from (54), (57), and (58) if we set $\kappa_2 = 0$ (as prescribed in (10)).

The desired rate of convergence for the cross-validated regression estimator based on neural networks, namely (12), follows similarly if we set $\kappa_1 \geq 1$ and $\kappa_2 = 1$. □

Proof of Corollary 2.1: We first establish bound (13) on the excess time-averaged expected prediction error of the sequence of prequential estimators $\{\hat{s}_k^{(p)}\}_{k \geq 1}^n$.

$$\frac{1}{n} \sum_{k=1}^{n} E\left[Y_{k+1} - \hat{s}_k^{(p)}(X_{k+1})\right]^2 - E[Y_0 - s(X_0)]^2$$
$$\overset{(a)}{=} \frac{1}{n} \sum_{k=1}^{n} \left(E\left[Y_{k+1} - \hat{s}_k^{(p)}(X_{k+1})\right]^2 - E[Y_{k+1} - s(X_{k+1})]^2\right)$$
$$\overset{(b)}{=} \frac{1}{n} \sum_{k=1}^{n} E\left[s(X_{k+1}) - \hat{s}_k^{(p)}(X_{k+1})\right]^2$$
$$= \frac{1}{n} E\left[s(X_2) - \hat{s}_1^{(p)}(X_2)\right]^2 + \frac{1}{n} \sum_{k=2}^{n} E\left[s(X_{k+1}) - \hat{s}_k^{(p)}(X_{k+1})\right]^2$$
$$\overset{(c)}{\leq} \frac{4\xi^2}{n} + (\text{constant}) \frac{1}{n} \sum_{k=2}^{n} \frac{\ln k}{\sqrt{k}}$$
$$\overset{(d)}{<} \frac{4\xi^2}{n} + (\text{constant}) \frac{\ln n}{\sqrt{n}} \tag{59}$$

where (a) follows since we assume the random variables $\{X_i, Y_i\}_{i=-\infty}^{\infty}$ to be i.i.d.; (b) follows by probabilistic manipulations similar to those carried out in step (a) of (51); (c) the bound on the first term follows from Assumptions 2.2 and 4.3 and, since Assumptions 2.1, 2.2, 2.3, and 2.4 hold, the bound on the second term follows from Theorem 2.1; (d) follows

since, for each $n \geq k \geq 2$,

$$\sum_{k=2}^{n} \frac{\ln k}{\sqrt{k}} \leq (\ln n) \sum_{k=2}^{n} \frac{1}{\sqrt{k}} < (\ln n) \int_{1}^{n} \frac{1}{\sqrt{a}}\, da = (\ln n)(\sqrt{n} - 1).$$

The desired result for the sequence of cross-validated regression estimators, namely (14), follows similarly. □

Acknowledgments

The authors are grateful to two anonymous referees and the Associate Editor, Professor David Haussler, for their numerous constructive suggestions which significantly improved this paper. This work was supported by the National Science Foundation under Grant DMS-97-03876.

Notes

1. As an important aside, we point out prequential model selection procedure for least-squares regression estimation problem represents one manifestation of Dawid's prequential principle—which can be applied to a variety of statistical problems. Also, note that Rissanen refers to prequential model selection as predictive minimum description length principle.
2. Note that we may let ϕ to be the cosine function, a wavelet ridge function (Hornik et al., 1994; Yukich, Stinchcombe, & White, 1995), or the hinged hyperplane (Breiman, 1993) by using Proposition 7 of Barron, Birgé, & Massart (1996) and by appropriately modifying Assumptions 2.3 and 2.4 in each case.

References

Barron, A.R. (1991). Complexity regularization. In G. Roussas (Ed.), *Proceedings NATO Advanced Study Institute on Nonparametric Functional Estimation*. Dordrecht, The Netherlands: Kluwer Academic Publishers.

Barron, A.R. (1993). Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Trans. Inform. Theory, 39*(3), 930–945.

Barron, A.R. (1994). Approximation and estimation bounds for artificial neural networks. *Machine Learning, 14*, 115–133.

Barron, A.R., Birgé, L., & Massart, P. (1996). Risk bounds for model selection via penalization. *Probab. Theory Relat. Fields* (to appear).

Baum, E., & Haussler, D. (1989). What size net gives valid generalization? *Neural Computation, 1*(1), 151–160.

Birgé, L., & Massart, P. (1994a). From model selection to adaptive estimation. *Probab. Theory Relat. Fields* (to appear).

Birgé, L., & Massart, P. (1994b). *Minimum contrast estimators on sieves*. Technical Report. Université Paris-Sud.

Breiman, L. (1993). Hinging hyperplanes for regression, classification, and function approximation. *IEEE Trans. Inform. Theory, 39*(3), 999–1013.

Dawid, A.P. (1984). Statistical theory: The prequential approach. *J.R. Statist. Soc. A, 147*(2), 278–292.

Dawid, A.P. (1991). Prequential data analysis. In M. Ghosh, & P.K. Pathak (Eds.), *Current issues in statistical inference*. Hayward, CA: Institute of Mathematical Statistics.

Dawid, A.P. (1992). Prequential analysis, stochastic complexity, and Bayesian inference. In J.M. Bernardo, J.O. Berger, A.P. Dawid, & A.F.M. Smith (Eds.), *Bayesian statistics*. Oxford University Press.

Haussler, D. (1992). Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Information and Computation, 100*, 78–150.

Haussler, D., Kearns, M., & Schapire, R.E. (1994). Bounds on the sample complexity of Bayesian learning using information theory and the VC dimension. *Machine Learning, 14*(1), 83–113.

Hoeffding, W. (1963). Probability inequalities for sums of bounded random variables. *J. Amer. Statist. Assoc., 58,* 13–30.

Hornik, K., Stinchcombe, M.B., White, H., & Auer, P. (1994). Degree of approximation results for feedforward networks approximating unknown mappings and their derivatives. *Neural Comput., 6*, 1262–1275.

Jones, L.K. (1997). L.K. Jones, The computational intractability of training sigmoidal neural networks. *IEEE Trans. Inform. Theory, 43*(1), 167–173.

Kearns, M. (1997). A bound on the error of cross validation using the approximation and estimation rates, with consequences for the training-test split. *Neural Computation, 9,* 1143–1161.

Lehtokangas, M., Saarinen, J., Huuhtanen, P., & Kaski, K. (1996). Predictive minimum description length criterion for time series modeling with neural networks. *Neural Computation, 8,* 583–593.

Li, K.-C. (1987). Asymptotic optimality for C_p, C_L, cross-validation, and generalized cross-validation: Discrete index set. *Ann. Statist., 15*, 958–975.

Lugosi, G., & Nobel, A. (1995). Adaptive model selection using empirical complexities. Submitted for publication.

Lugosi, G., & Zeger, K. (1996). Concept learning using complexity regularization. *IEEE Trans. Inform. Theory, 42*(1), 48–54.

McCaffrey, D.F., & Gallant, A.R. (1994). Convergence rates for single hidden layer feedforward networks. *Neural Networks, 7,* 147–158.

Modha, D.S., & Masry, E. (1996). Minimum complexity regression estimation with weakly dependent observations. *IEEE Trans. Inform. Theory, 42*, 2133–2145.

Modha, D.S., & Masry, E. (1998). Memory-universal prediction of stationary random processes. *IEEE Trans. Inform. Theory, 44*, 117–133.

Mosteller, F., & Tukey, J.W. (1968). Data analysis, including statistics. In G. Lindzey & E. Aronson (Eds.), *Handbook of social psychology* (Vol. 2). Reading, MA: Addison-Wesley.

Rissanen, J. (1983). A universal prior for integers and estimation by minimum description length. *Ann. Statist., 11*(2), 416–431.

Rissanen, J. (1986a). A predictive least-squares principle. *IMA J. Math. Contr. Inform., 3*, 211–222.

Rissanen, J. (1986b). Complexity of strings in the class of Markov sources. *IEEE Trans. Inform. Theory, 32*(4), 526–532.

Rissanen, J. (1989). *Stochastic complexity in statistical inquiry*. Teaneck, NJ: World Scientific Publishers.

Rissanen, J. (1994). Information theory and learning. In P. Smolensky, M.C. Mozer, & D.E. Rumelhart (Eds.), *Mathematical perspectives on neural networks*. Hilldale, NJ: L. Erlbaum Associates.

Rissanen, J., Speed, T., & Yu, B. (1992). Density estimation by stochastic complexity. *IEEE Tran. Inform. Theory, 38*(2), 315–323.

Sarkar, D. (1995). Methods to speed up error back-propagation learning algorithm. *ACM Comput. Surveys, 27*(4), 519–542.

Shen, X., & Wong, W.H. (1994). Convergence rates of sieves estimates. *Ann. Statist., 22*, 580–615.

Stone, C.J. (1984). An asymptotically optimal window selection rule for kernel density estimates. *Ann. Statist., 12*, 1285–1297.

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *J.R. Statist. Soc. B, 36*, 111–133.

Stone, M. (1977). An asymptotic equivalence of choice of model by cross-validation and Akaike's criterion. *J.R. Statist. Soc. B, 39*, 44–47.

Vapnik, V.N. (1982). *Estimation of dependences based on empirical data*. New York: Springer-Verlag.

Vapnik, V.N. (1995). *The nature of statistical learning theory*. New York: Springer-Verlag.

White, H. (1989). Connectionist nonparametric regression: Multilayer feedforward networks can learn arbitrary mappings. *Neural Networks, 3*, 535–549.

Yukich, J.E., Stinchcombe, M.B., & White, H. (1995). Sup-norm approximation bounds for networks through probabilistic methods. *IEEE Trans. Inform. Theory. 41*(4), 1021–1027.

Yu, B., & Speed, T. (1992). Data compression and histograms. *Probab. Theory Relat. Fields, 92*, 195–229.

Received October 24, 1996
Accepted October 15, 1997
Final Manuscript March 3, 1998