Published in final edited form as:

Stat Sin. 2016 April ; 26(2): 841–860. doi:10.5705/ss.202014.0068.

Prediction-based Termination Rule for Greedy Learning with Massive Data

Chen Xu¹, Shaobo Lin², Jian Fang², and Runze Li¹

¹The Pennsylvania State University

²Xi'an Jiaotong University

Abstract

The appearance of massive data has become increasingly common in contemporary scientific research. When sample size n is huge, classical learning methods become computationally costly for the regression purpose. Recently, the orthogonal greedy algorithm (OGA) has been revitalized as an efficient alternative in the context of kernel-based statistical learning. In a learning problem, accurate and fast prediction is often of interest. This makes an appropriate termination crucial for OGA. In this paper, we propose a new termination rule for OGA via investigating its predictive performance. The proposed rule is conceptually simple and convenient for implementation, which suggests an $O(\sqrt{n/\log n})$ number of essential updates in an OGA process. It therefore provides an appealing route to conduct efficient learning for massive data. With a sample dependent kernel dictionary, we show that the proposed method is strongly consistent with an $O(\sqrt{\log n/n})$ convergence rate to the oracle prediction. The promising performance of the method is supported by both simulation and real data examples.

Key words and phrases

Forward regression; Massive data; Greedy algorithms; Kernel methods; Nonparametric regression; Sparse modeling

1 Introduction

In modern scientific research, collecting data of unprecedented sizes is increasingly frequent. When the amount of data is huge, many traditional modeling strategies become infeasible due to the high computational burden. Developing effective and efficient approaches for analyzing massive data has been a recent focus in statistical learning and data mining. See Li, Lin, and Li (2013), National Research Council (2013), and Rajaraman, Leskovec, and Ullman (2011).

Chen Xu, The Methodology Center, The Pennsylvania State University, State College, PA 16801, USA, cux10@psu.edu

Shaobo Lin and Jian Fang, School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, China, ssxvkihc.lsb@stu.xjtu.edu.cn, jianfang@stu.xjtu.edu.cn

Runze Li, The Methodology Center, Department of Statistics, The Pennsylvania State University, State College, PA 16801, USA, rli@stat.psu.edu

The content is solely the responsibility of the authors and does not necessarily represent the official views of NIDA or NSFC.

Unlike data collected from designed experiments, massive data sets are sometimes available from uncontrolled natural mechanisms. To avoid potential model mis-specifications, semiparametric or nonparametric methods may be preferred. In regression analysis, one popular strategy is to build a regression model based on a set of pre-specified basis functions (dictionary). In particular, learning from a sample dependent kernel dictionary (SDKD) has received a great deal of attention, due to its high modeling flexibility and implementation feasibility (Shi, Feng, and Zhou (2011), Wu and Zhou (2008)). Specifically, let $z = \{(y_i, \mathbf{x}_i); i = 1, \dots, n\}$ be n independent observations from model

$$Y = f^*(X) + \varepsilon, \quad (1)$$

where Y is a response variable, X is a p -dimensional covariate vector, and ε is an observational noise with mean zero. We assume that X is random and independent of ε . The unknown function $f^*(X) = E(Y|X)$ then denotes the expectation of Y given X . The goal of learning is to explore the relationship $f(X) : X \rightarrow Y$ such that $f(X)$ is a good prediction of Y . Let $\mathcal{E}(f) = E(|f(X) - Y|^2)$ denote the predictive error between $f(X)$ and Y . It is known that $\mathcal{E}(f)$ is minimized at $f = f^*$. Thus, f^* serves as a conceptual oracle prediction. A good prediction f is expected to be close to f^* . In the SDKD-based approach, the dictionary is set by

$$D_z = \{K(\mathbf{x}_i, \cdot); i = 1, \dots, n\}, \quad (2)$$

where $K(\cdot, \cdot) : R^p \times R^p \rightarrow R$ is a continuous, symmetric, and nonnegative definite kernel function. The learning task is conducted by exploring f from the functional space linearly spanned by the elements in D_z that is,

$H_z = \text{span}\{D_z\} = \{g(\cdot) = \sum_{i=1}^n \theta_i K(\mathbf{x}_i, \cdot) \text{ for } \theta_i \in R\}$. In this paper, we are particularly interested in learning from D_z with a large sample size n (e.g. $n > 10,000$).

For the learning purpose, various regularization methods have been proposed in the past decades. These methods specify $f(X)$ by minimizing a regularized empirical loss. Examples include but are not limited to the regularized least squares methods (RLS; Girosi, Jones, and Poggio (1995), De Vito, Caponnetto, and Rosasco (2005)) and the support vector machine (SVM; Cortes and Vapnik (1995), Steinwart and Christmann (2008)). Although these methods are effective for learning, they are typically developed for the situation where n is moderate. When the regularization methods are applied to datasets with large sample sizes, solving the associated optimization problems often involves the inversion of a huge information matrix. This brings great numerical hurdles in both algorithmic design and memory storage. In addition, the large scale of D_z makes tuning these methods a challenging task.

In recent works, the orthogonal greedy algorithm (OGA) has been revitalized as a computationally convenient alternative for learning (Barron et al. (2008), Chen, Li, and Pan (2013a), Chen et al. (2013b)). As a different learning paradigm, it starts from a null model and explores $f(X)$ based on a series of expanding subspaces. It therefore provides a stepwise

learning framework without the need to handle the huge information matrix at all steps. Such an “expansion” strategy, in contrast with regularization, brings many potential computational benefits for learning in large- n situations.

In the literature, OGA and its variants have been widely used for signal sparse approximation and recovery. See, for examples, Devore and Temlyakov (1996), Tropp (2004), Cai and Wang (2011), Wang (2009), and Zhang (2009, 2011). However, it has not drawn much attention for statistical learning with focus on prediction. Different from approximation, an appropriate termination rule is crucial for OGA-based learning. This is analogous to the tuning parameter selection in the regularization methods. Essentially, a termination rule for OGA determines the number of updating steps k that is needed for an effective prediction. To be specific, let $\hat{f}_k$ denote the prediction obtained from a k -step OGA. It is known that an overly large k causes overfitting and slows down the learning process. Meanwhile, a parsimonious k often makes $\hat{f}_k$ an unreliable prediction due to the inadequate training. Barron et al. (2008) suggested picking a k that minimizes a penalized empirical risk

$$k' = \arg \min_{k \in \mathbb{N}_n} \left\{ \sum_{i=1}^n (\hat{f}_k(x_i) - y_i)^2 + \kappa k \log(n) \right\}, \quad (3)$$

where $\mathbb{N}_n = \{1, 2, \dots, n\}$ and κ is a positive constant. The final prediction $\hat{f}_{k'}$ therefore comes from a compromise between goodness of fit and model complexity. Although $\hat{f}_{k'}$ is shown to be consistent, a full OGA running (n -step) is often needed before k' can be determined. Thus, implementing (3) can be time consuming for the large n situations. Moreover, as Barron et al. (2008) recommended a large value for κ , rule (3) tends to select a k' that may be overly small in application. Addressing the same issue under a slightly different framework, Chen et al. (2013a, 2013b) proposed terminating OGA when

$$\sum_{i=1}^n (\hat{f}_k(x_i) - y_i)^2 + n \|\hat{f}_k\|_{l_1} \leq \sum_{i=1}^n y_i^2 \quad (4)$$

is satisfied with $\|\hat{f}\|_{l_1} = \inf \{ \sum_j |\theta_j| : \hat{f} = \sum_j \theta_j K(x_j, \cdot) \}$. Despite its theoretical feasibility, (4) often leads to a long updating procedure, which makes it inefficient for the massive data analysis; see Section 3.2 for more dissections. Recently, Ing and Lai (2011) studied the predictive performance of OGA in high-dimensional linear model. They suggested that, if $\hat{f}^*$ is linear of X and has a weak sparse representation, OGA may only need $k \ll n$ updates for an accurate prediction. However, this encouraging result is not directly applicable to non-parametric statistical learning.

The existing literature stimulates us to seek a new termination rule that gears OGA to learning with massive data. In particular, we conjecture that a sensible k can be selected directly based on a discrepancy measure between $\hat{f}_k$ and the oracle $\hat{f}^*$. Compared with the aforementioned methods, such a strategy may provide more straightforward insights on k , which helps to conduct an efficient learning procedure. In this spirit, we propose a prediction-based termination rule (PTR) for using OGA in learning with massive data. The new rule treats $\mathcal{L}(\hat{f}_k) = E(\hat{f}_k - \hat{f}^*)^2$ (i.e., generalization error) as a natural discrepancy

measure for $\hat{f}_k$ and selects a k^* such that $\mathcal{L}(\hat{f}_{k^*})$ is properly bounded. Specifically, it suggests $k^* = O(\sqrt{n/\log n})$ for OGA-based learning, which efficiently leads to a prediction with an $O(\sqrt{\log n/n})$ convergence rate to f^* . Since $k^* \ll n$ for the large n situations, PTR makes the associated OGA highly computationally attractive. The scale of k^* also suggests a sparse kernel basis for estimating a general f^* . To some extent, this finding provides a theoretical justification for the sparse model assumption that is commonly used in signal recovery and feature selection. Under mild conditions, we further show that the OGA procedure implemented by PTR is strongly consistent to the oracle prediction f^* . The promising performance of the new method is supported by a series of simulations and real data examples.

The rest of this paper is organized as follows. In Section 2, we propose the new PTR for OGA and assess its theoretical properties. In Section 3, we show numerical studies to demonstrate the good performance of PTR, especially in large- n situations. Finally, we conclude in Section 4 with some useful remarks. The proofs of theorems are given in the Appendix.

2 Termination for the OGA-based Learning

2.1 The OGA Framework

Following earlier notation, suppose that $Y \in [-M, M] \subset \mathbb{R}^1$ for some positive constant M and $X \in \mathcal{X} \subset \mathbb{R}^p$ for some compact set $\mathcal{X}$ with a positive Lebesgue measure. Recall that the goal of learning is to find a good prediction of Y by analyzing a set of training samples $z = \{(y_i, \mathbf{x}_i); i \in \mathbb{N}_n\}$.

OGA is a stepwise method that seeks for the best prediction from a series of expanding subspaces of $\mathcal{F} : \mathcal{X} \rightarrow \mathbb{R}^1$. It outputs the final prediction when the updating procedure meets a certain termination criterion. Before it is applied to statistical learning, OGA was also known as orthogonal matching pursuit or projection pursuit regression (Friedman and Stuetzle (1981), Mallat and Zhang (1993), Pati, Rezaiifar, and Krishnaprasad (1993)). One may refer to Temlyakov (2003) for more history about this method.

For the convenience of presentation, we introduce additional notation as follows. For any given z and $f, g \in \mathcal{F}$, we define the empirical inner product and the empirical norm, respectively, by

$$\langle f, g \rangle_n = \frac{1}{n} \sum_{i=1}^n f(\mathbf{x}_i)g(\mathbf{x}_i), \quad \|f\|_n^2 = \frac{1}{n} \sum_{i=1}^n |f(\mathbf{x}_i)|^2.$$

Also, we treat an arbitrary vector $\mathbf{u} = (u_1, \dots, u_n)^T \in \mathbb{R}^n$ as a degenerated function on $\mathcal{X}$, so that

$$\langle f, \mathbf{u} \rangle_n = \frac{1}{n} \sum_{i=1}^n f(\mathbf{x}_i) u_i(\mathbf{x}_i) = \frac{1}{n} \sum_{i=1}^n u_i f(\mathbf{x}_i).$$

For a given kernel function $K(\cdot, \cdot)$, we set

$$D_z^* = \left\{ \frac{K(\mathbf{x}_i, \cdot)}{\|K(\mathbf{x}_i, \cdot)\|_n} : i \in \mathbb{N}_n \right\}$$

as a normalized dictionary based on z . This forms a candidate learning space

$$\mathcal{F}_z = \{T_M[f] : f \in \text{span}\{D_z^*\}\},$$

where

$$T_M[f(\cdot)] = \begin{cases} \text{sign}[f(\cdot)]M, & |f(\cdot)| > M \\ f(\cdot), & |f(\cdot)| \leq M \end{cases}$$

is a truncating operator.

With above settings, we present the OGA-based learning procedure as the following algorithm.

Algorithm 1

(OGA)

Input: $z, M, K(\cdot, \cdot), k^s$

Output: $\hat{f}_k^s$

Compute D_z^* and set $k = 1, V_0 = \emptyset, \mathbf{r}_0 = \mathbf{y} = (y_1, \dots, y_n)^T$.

When $k \leq k^s$, recursively do substeps **a**–**d**:

a Update the active set by $V_k = V_{k-1} \cup \{g_k\}$, where

$$g_k = \arg \max_{g \in D_z^*} |\langle \mathbf{r}_k - \mathbf{1}, g \rangle_n|.$$

b Let $\mathbf{g}_k = (g_1, \dots, g_k)^T$. Compute

$$\hat{\beta}_k = \arg \min_{\beta_k} \|\mathbf{y} - \mathbf{g}_k^T \beta_k\|_n^2,$$

and update the estimate by

$$f_k = \mathbf{g}_k^T \hat{\boldsymbol{\beta}}_k.$$

c Update the residual by

$$\mathbf{r}_k = \mathbf{y} - (f_k(\mathbf{x}_1), f_k(\mathbf{x}_2), \dots, f_k(\mathbf{x}_n))^T.$$

d Increase the number of steps k by one.

Output $f_k^* = T_M[f_k]$.

Note that OGA never selects the same atom twice because a selected atom is orthogonal to the current residual. Compared with the regularization method, OGA is essentially a forward-searching procedure, which avoids the direct storage and operations on the full model. This framework brings many computational benefits for learning in large- n situations.

2.2 Termination Strategy

Although OGA provides an appealing route for learning, an appropriate k^* needs to be determined in applications. As mentioned in Section 1, the efficiency of OGA relies heavily on the choice of k^* , which indicates the number of basis functions included in f_k^* . For statistical learning, a good choice of k^* should make $\hat{f}_k^*$ an accurate prediction of Y . Thus, it is favorable to select a k^* such that $\hat{f}_k^*$ achieves a high predictive accuracy.

To be specific, suppose that $Z = (Y, X)$ follows from an unknown distribution ρ on $[-M, M] \times \mathcal{X} \subset \mathbb{R}^{p+1}$. We use ρ_X to denote the marginal distribution of X . For an arbitrary function $f(X)$, we define

$$\|f\|_{\rho_X}^2 = \int_{\mathcal{X}} f(X)^2 d\rho_X$$

as a norm with respect to ρ_X . Under model (1), the predictive error of an arbitrary $f(X)$ can be naturally measured by

$$\mathcal{E}(f) = E(|f(X) - Y|^2), \quad (5)$$

which attains its minimum at $f = f^*$. Therefore,

$$\mathcal{L}(\hat{f}_k) = \mathcal{E}(\hat{f}_k) - \mathcal{E}(f^*) = \|\hat{f}_k - f^*\|_{\rho_X}^2 \quad (6)$$

serves as a discrepancy measure between $\hat{f}_k$ and the oracle f^* in terms of the predictive power. Accordingly, an ideal k^* should be selected such that $\mathcal{L}(\hat{f}_k)$ is minimized at $k = k^*$. In the machine learning community, term (6) is often referred to as the generalization error.

Although an optimal k^* is desirable, it is only conceptual because the oracle f^* is unknown. One feasible idea is then to find a good k^* such that $\mathcal{L}(\hat{f}_k^*)$ is properly controlled. In the literature, several rules have been proposed for choosing a k^* such that $\hat{f}_k^*$ has a balanced goodness of fit and model complexity. In this work, we propose selecting a sensible k^* through bounding $\mathcal{L}(\hat{f}_k)$ directly. This strategy may help to provide more straightforward insights on k^* for the prediction purpose.

Since $\hat{f}_k$ is obtained based on n independent and identically distributed (i.i.d.) samples of Z , $\mathcal{L}(\hat{f}_k)$ is random with respect to probability measure ρ^n . To provide some guidance on choosing k^* , we derive a probability bound for $\mathcal{L}(\hat{f}_k)$ in the following proposition.

Proposition 1—*Let $\hat{f}_k$ be the k -step OGA output defined by Algorithm 1. Then, for any $0 < \delta < 1$ and $h \in \text{span}\{D_z^*\}$, when n is sufficiently large, the following inequality holds with probability at least $1 - \delta$*

$$\mathcal{L}(\hat{f}_k) \leq C \left\{ \|f^* - h\|_{\rho_X}^2 + \log \frac{2}{\delta} \left(\frac{\|h\|_{L_1}^2}{k} + \frac{\|h\|_{\infty}^2 + k \log n}{n} \right) \right\}, \quad (7)$$

where C is a positive constant and $\|\cdot\|_{\infty}$ denotes the function L^∞ norm.

The proof is given in the Appendix. Proposition 1 implies that, with a high confidence level, the generalization error of $\hat{f}_k$ is bounded by a quantity depending on k and h . This result extends Theorem 3.1 of Barron et al. (2008) in the sense that the error bound holds for $\hat{f}_k$ with any $k \geq 1$. It reveals an intrinsic predictive property for an arbitrary k -step OGA and therefore serves as a useful tool for choosing an appropriate k^* . Since we assume $\|f^*\|_{\infty} \leq M$, (7) attains its minimum with an h that is bounded above. Thus, a sensible k^* may be selected based on (7) with $\|h\|_{L_1} = T$ for some constant $T > 0$. Heuristically, it is reasonable to set

$$k^* = \arg \min_k \left\{ \frac{T^2}{k} + \frac{k \log n}{n} \right\} = T \sqrt{\frac{n}{\log n}}, \quad (8)$$

which minimizes (7) in k with $\|h\|_{L_1} = T$. Because (8) is derived from a probability bound of $\mathcal{L}(\hat{f}_k)$, we name it as the prediction-based termination rule (PTR).

Different from other rules in the literature, PTR provides a straightforward insight on the scale of k^* . When n is large, PTR tends to pick a k^* that is much smaller than n . The sensible choice on k^* makes the associated OGA procedure computationally attractive. The proposed termination rule indicates that, for prediction, only a few OGA updates may be needed. This amounts to suggesting a sparse kernel basis that is essential for effective learning. With such a sparse basis, researchers may further interpret the model parameters and conduct fast predictions for the future responses. Also, PTR is convenient for implementation, because it does not involve large-scale ad hoc evaluations in the OGA process. The simplicity gives the new rule an additional advantage for the analysis of massive data.

The choice of T in (8) should reflect the prior information on f^* . Our empirical experience shows that choosing $T = 1 + \log p$ is usually adequate for conducting accurate prediction. The term $\log p$ is added to reflect the intuition that higher dimensionality of X often leads to longer OGA updating to achieve an accurate prediction. Since OGA is computationally less demanding, it is convenient to tune a proper T for a specific case. In our simulation studies, the value of T is selected among a few candidates based on high-dimensional Akaike information criterion (HDAIC; Ing and Lai (2011)); see Section 3.1 for more details. This empirical choice on T works reasonably well in our numerical examples, and thus we recommend it for the practical implementation of PTR.

2.3 Consistency of PTR

We now provide some theoretical justifications for using PTR in the OGA-based learning. Specifically, we recommend the use of Algorithm 1 with k^* decided by PTR (8) and refer to it as the PTR-based OGA procedure. In statistical learning, a good prediction rule is expected to approach the oracle f^* arbitrarily close as the sample size increases. Such a property is typically referred to as consistency, which is especially desirable for the analysis of massive data with a large n . It is therefore important to first investigate whether the proposed PTR leads to a consistent prediction.

For learning with a SDKD, the effectiveness of a method relies on the sampling scheme as well as the choice of kernel $K(\cdot, \cdot)$. For the convenience of discussion, let $\mathcal{C}(\mathcal{X})$ be the family of continuous functions on $\mathcal{X}$ and $\text{supp}(\cdot)$ denotes the support set of a measure. We assess the performance of PTR under the following technical conditions.

- C1** $\text{supp}(\rho_X) = \mathcal{X}$.
- C2** Let $\mathcal{H}_K = \overline{\text{span}}\{K_X : K_X = K(X, \cdot), X \in \mathcal{X}\}$ be the reproducing kernel Hilbert space associated with $K(\cdot, \cdot)$. For any $\omega > 0$ and $f \in \mathcal{C}(\mathcal{X})$, there exists a $g \in \mathcal{H}_K$ such that $\|f - g\|_{\rho_X} \leq \omega$.

Condition C1 guarantees that every element of $\mathcal{X}$ has a positive probability to be sampled, so that $\mathcal{X}$ is an effective sampling space for random variable X on $\mathcal{X}$. Condition C2 corresponds to the notion of universal kernel proposed by Micchelli, Xu, and Zhang (2006), which requires that any continuous function on $\mathcal{X}$ can be closely approximated within $\mathcal{H}_K$. It serves as a prerequisite for approaching an arbitrary f^* from $\mathcal{F}_2 \subset \mathcal{H}_K$. Under our model setup, many choices of $K(\cdot, \cdot)$ naturally satisfy this condition, which include the popular Gaussian kernel as a special case. One may refer to Micchelli, Xu, and Zhang (2006) for more discussions on the universal kernel.

For the asymptotic analysis, we associate both dictionary D_z^* and prediction $\hat{f}_k^*$ with sample size n . The consistency of PTR is established via the following theorem.

Theorem 1—Let $\hat{f}_k^*$ be the output obtained from Algorithm 1 with k^* decided by GTR. Suppose that Conditions C1-C2 are satisfied, then

$$P \left\{ \lim_{n \rightarrow \infty} \mathcal{L}(\hat{f}_{k^*}) = 0 \right\} = 1.$$

The proof is given in the Appendix. Theorem 1 shows that, as $n \rightarrow \infty$, $\hat{f}_{k^*}$ necessarily converges to the oracle prediction in an almost sure sense. Since $\mathcal{X}$ is an arbitrary compact set with a positive Lebesgue measure in $\mathbb{R}^p$, Theorem 1 applies to a wide range of distributions on X . The result established here is closely related to the notion of universal strong consistency defined by Györfy et al. (2002)

Remark 1: Theorem 1 is established under the assumption that both Y and X are bounded, which might be a little restrictive. Similar assumptions have been commonly used in the literature of kernel-based statistical learning (see e.g., Barron et al. (2008), Chen et al. (2013b)). By using truncation techniques, this assumption may be further relaxed by requiring only $E(Y^2) < \infty$. However, such a relaxation may need more mathematical tools and is likely to make the proof lengthy. For a conceptually more focused manuscript, we regretfully leave this issue for the future research.

Under our model setup, the following corollary is an immediate result of Theorem 1, which shows that $\hat{f}_{k^*}$ has a diminishing generalization risk as the sample size increases.

Corollary 1—*Under conditions of Theorem 1, as $n \rightarrow \infty$,*

$$E[\mathcal{L}(\hat{f}_{k^*})] \rightarrow 0.$$

Proof: Note that $f^*(X) = E(Y|X) \in [-M, M]$ and $\hat{f}_{k^*}(X) \in \mathcal{F}_z^*$. We have $\mathcal{L}(f^*)$ bounded by $4M^2$. The corollary is implied by the almost sure convergence of $\mathcal{L}(\hat{f}_{k^*})$ from Theorem 1.

2.4 Convergence Rate

While the consistency justifies the effectiveness of $\hat{f}_{k^*}$, it provides less insight on the efficiency of a learning procedure. To further assess the proposed PTR, in this subsection, we investigate the convergence rate of $E[\mathcal{L}(\hat{f}_{k^*})]$. In general, a specific rate can not be derived for an arbitrary f^* without proper assumptions. To facilitate our proofs, we assess the efficiency of $\hat{f}_{k^*}$ under the following condition.

C3 When n is sufficiently large, there exists a $h' \in \text{span}(D_z^*)$ such that

$$\|h'\|_{l_1} \leq B \quad \text{and} \quad \|f^* - h'\|_{\infty} \leq Bn^{-r}$$

for some positive constants B and r .

Condition C3 corresponds to the $L_{1,r}$ space introduced in Barron et al. (2008). Similar conditions have been commonly used for investigating OGA in sparse approximation (see, e.g., Devore and Temlyakov(1996)). Intuitively, C3 requires that f^* can be approximated

within $\text{span}\{D_z^*\}$ at a certain rate. It is satisfied when f^* has a representation in $\mathcal{H}_K$ with a bounded ℓ_1 norm and a power-decayed tail. This condition is purely technical, which might be stringent for situations where the dimensionality of X is high. We only use C3 to gain some theoretical understanding of the proposed PTR and do not intend to make this condition the weakest possible. Readers may refer to Barron et al. (2008) for more detailed discussions about this condition.

Under Condition C3, we derive an $O(\sqrt{\log n/n})$ convergence rate for $\hat{f}_k^*$ as the following theorem.

Theorem 2—*Suppose that Condition C3 is satisfied with $r > 0.5$, then*

$$E[\mathcal{L}(\hat{f}_{k^*})] = O(\sqrt{\log n/n}).$$

The proof is in the Appendix. It is known that when an n -step OGA is used for approximating a known function, the essential convergence rate is $O(n^{-1/2})$, which can not be generally improved without further conditions (Devore and Temlyakov (1996)). Thus, Theorem 2 implies that, up to an $O(\sqrt{\log n})$ term, the PTR-based OGA approaches the unknown f^* almost as efficiently as a full OGA run. We suspect that the $O(\sqrt{\log n/n})$ rate might be nearly optimal for a SDKD-based learning, as additional cost needs to be paid for the truncated dictionary as well as the observational noise.

Remark 2: The convergence rate obtained from Theorem 2 is applicable to PTR (8) with an arbitrary scale parameter $T > 0$. In practice, we suggest tuning an appropriate T from a finite set of candidate values. Thus, this rate is also generally applicable to PTR with a user-specified T .

2.5 Implementation Issue

With the aid of PTR, OGA is geared to conducting efficient learning with massive data. In this subsection, we provide an efficient implementation procedure for OGA.

Recall that, in Algorithm 1, computing $\hat{\beta}_k$ from step **b** involves a least squares problem. Its solution is given by

$$\hat{\beta}_k = (Q_k^T Q_k)^{-1} Q_k^T y, \quad (9)$$

where Q_k is a $n \times k$ matrix with the element in the i th row and j th column being $g_j(x_i)$. When a long OGA run is needed, a direct implementation of (9) might be numerically less efficient. To further reduce computational burden, an iterative form of $\hat{\beta}_k$ is often used in practice. Specifically, let $Q_k^+ = (Q_k^T Q_k)^{-1} Q_k^T$ and $Q_k = (q_1, \dots, q_k)$ with $q_j = (g_j(x_1), \dots, g_j(x_n))^T$ for $j \in \mathbb{N}_k$. We present this result as the following proposition.

Proposition 2—In the updating procedure of Algorithm 1, for $k \geq 1$, $\hat{\beta}_k$ defined in step **b** is in form of

$$\hat{\beta}_{k+1} = \begin{pmatrix} \hat{\beta}_k - c_k Q_k^+ q_{k+1} \\ c_k \end{pmatrix},$$

where $c_k = q_{k+1}^T r_k [q_{k+1}^T (I - Q_k Q_k^+) q_{k+1}]^{-1}$

Readers may refer to Elad (2010) or Sturm and Christensen (2012) for the proof. Proposition 2 provides a convenient tool for implementing OGA. Specifically, given Q_k^+ at the k -th step, $\hat{\beta}_{k+1}$ can be efficiently computed without complex operations. This numerical advantage further supports the OGA-based learning in large- n situations. Based on Proposition 2, we advocate the following algorithm as an efficient implemental procedure for Algorithm 1.

Algorithm 2

(fast-OGA).

Input: $z, M, K(\cdot, \cdot), k^*$

Output: $\hat{f}_k^*$

Compute D_z^* and set $k = 1, r_1 = y, V_1 = \{g_1\}$, where

$$g_1 = \arg \max_{g \in D_z^*} | \langle r_1, g \rangle_n |.$$

Let

$$\hat{\beta}_1 = \frac{\langle y, g_1 \rangle_n}{\langle g_1, g_1 \rangle_n}, \quad Q_1^+ = \frac{n^{-1} q_1^T}{\langle g_1, g_1 \rangle_n}, \quad f_1 = \hat{\beta}_1 g_1$$

When $k \geq k^*$, recursively do the following **a–d** substeps:

a Let $V_{k+1} = V_k \cup \{g_{k+1}\}$, where

$$g_{k+1} = \arg \max_{g \in D_z^*} | \langle r_k, g \rangle_n |.$$

b Set

$$a_k = \frac{1}{q_{k+1}^T [I - Q_k Q_k^+] q_{k+1}}, \quad c_k = a_k q_{k+1}^T r_k.$$

update

$$\hat{\beta}_{k+1} = \begin{pmatrix} \hat{\beta}_k - c_k \mathbf{Q}_k^+ \mathbf{q}_{k+1} \\ c_k \end{pmatrix},$$

$$\mathbf{Q}_{k+1}^+ = \begin{pmatrix} \mathbf{Q}_k^+ \{ \mathbf{I} - a_k \mathbf{q}_{k+1} \mathbf{q}_{k+1}^T [\mathbf{I} - \mathbf{Q}_k \mathbf{Q}_k^+] \} \\ a_k \mathbf{q}_{k+1}^T [\mathbf{I} - \mathbf{Q}_k \mathbf{Q}_k^+] \end{pmatrix},$$

$$\text{and } f_{k+1} = \mathbf{g}_{k+1}^T \hat{\beta}_{k+1}.$$

c Update

$$\mathbf{r}_{k+1} = \mathbf{y} - \mathbf{Q}_{k+1} \hat{\beta}_{k+1}.$$

d Increase the number of steps k by one.

Output $\hat{f}_k^* = T_M\{f_k\}$.

3 Numerical Studies

We evaluate the finite sample performance of PTR through both simulation and real data examples. In particular, under the OGA framework, we compare PTR with other popular termination rules in terms of both computational efficiency and predictive accuracy. We further reveal the benefits of the proposed method through conducting comparisons between the PTR-based OGA and the regularization methods. All numerical studies are implemented by MATLAB 8.2 on a windows workstation with 8-core 3.07GHz CPUs.

3.1 Simulations

In the simulation studies, we assess the performance of PTR on a hypothetical learning problem with $p = 1$. Specifically, we generate independent observations $\mathbf{z} = \{(y_i, \mathbf{x}_i), i \in \mathbb{N}_n\}$ based on model (1) with

$$f^*(x) = \begin{cases} \frac{\sin(20x-10)}{20x-10}, & x \in [0, 0.5) \cup (0.5, 1] \\ 1, & x=0.5 \end{cases}$$

and $\varepsilon \sim N(0, 0.1)$. We adopt two sampling schemes: in the first scheme, we sample X from a uniform distribution on $[0, 1]$; in the second scheme, we sample X from a truncated normal distribution on $[0, 1]$ with mean 0.5 and standard deviation 0.25. These two schemes represent proportional and disproportional sampling designs. Under each sampling scheme, we further set $n = 1000, 5000, 10000$ and estimate f^* through analyzing $\mathbf{z}$.

To learn f^* , we apply the proposed PTR to OGA based on a Gaussian kernel

$$K(\mathbf{x}_1, \mathbf{x}_2) = \exp(-\|\mathbf{x}_1 - \mathbf{x}_2\|_2^2 / \tau^2), \quad (10)$$

with $\tau = 0.1$ and $\|\cdot\|_2$ denoting the L_2 norm. To facilitate computation, we implement OGA by Algorithm 2 and set PTR with $T \in \mathcal{T} = \{(1 + \log p)h/20; h = 1, \dots, 20\}$. The final k^* for OGA is decided by the PTR with a $T \in \mathcal{T}$ that minimizes

$$\text{HDAIC}(T) = n \log(\|\mathbf{y} - \hat{f}_{k(T)}(X)\|_n^2) + 2k(T) \log n,$$

where $k(T)$ denotes the largest integer that is smaller than the PTR value with respect to T . For comparison, we also carry out OGA based on termination rules (3) and (4). For (3), Barron et al. (2008) suggested using $\kappa = 2568M^4(a+5)$, where $a - 1$ indicates the total number of basis functions taken to be n^a . We follow their recommendation by setting $M = 1$ and $a = 1$ according to our setup. Moreover, we compare PTR with the termination rules used in Theorem 7 of Cai and Wang (2011) and Theorem 4 of Zhang (2009), which were proposed for using OGA in sparse recovery. For simplicity, we refer to the aforementioned four rules as TR1, TR2, TR3, and TR4 respectively, which are based on their orders of presentation. As a benchmark, we further report the results from the support vector regression (SVR), which is a popular SVM for the regression analysis. In our simulation studies, we implement SVR by MATLAB package LIBSVM, where a default 0.1-insensitive loss is used and λ is similarly tuned by HDAIC.

The performance of each method is evaluated by its goodness of fit, predictive accuracy, model complexity, and computational cost. Specifically, we measure the goodness of fit through the training root mean squared error (Tr-RMSE), which is given by

$$\text{Tr-RMSE}(\hat{f}) = \left\{ \frac{1}{n} \sum_{i=1}^n |\hat{f}(\mathbf{x}_i) - y_i|^2 \right\}^{1/2}.$$

To assess the predictive accuracy, we generate an independent testing set $z \stackrel{\sim}{=} \{(\tilde{\mathbf{y}}_i, \tilde{\mathbf{x}}_i), i \in N_T \text{ for } T = 1000\}$ from $Y = f^*(X)$ with $X \sim U[0, 1]$ and compute the testing root mean squared error (Ts-RMSE)

$$\text{Ts-RMSE}(\hat{f}) = \left\{ \frac{1}{T} \sum_{i=1}^T |\hat{f}(\tilde{\mathbf{x}}_i) - \tilde{y}_i|^2 \right\}^{1/2}.$$

We denote the model complexity by k , which is reported as the number of basis functions included in each fitted model. For the efficiency comparison, we record the absolute computing time (in second) for conducting each method and denote it by TIME.

The simulation results are summarized in Tables 1–2, where the statistics are averaged based on 300 independent repetitions. For Tr-RMSE and Ts-RMSE, we provide the corresponding

standard errors in brackets. From these results, we observe that the performance of OGA varies drastically according to different termination rules. For TR1, a null model ($k = 0$) is consistently selected, which results in an improper prediction with both high Tr-RMSE and Ts-RMSE. In addition, since TR1 selects k based on a full OGA run, the whole procedure requires a high computational cost. Unlike the stringency of TR1, method TR2 leads OGA to the other extreme, where k is chosen as the sample size n . Despite the satisfactory Ts-RMSE, TR2 is computationally costly for the large n cases and brings little interpretive value in the fitted model. For TR3 and TR4, a small k is picked for OGA and thus the procedure is computationally less intensive. However, as reflected by the high Ts-RMSE, these rules fail to provide effective predictions in our setup. In contrast, the performance of PTR is generally satisfactory. In each case, it picks a proper $k \ll n$ and accurately predicts Y with a high efficiency. Compared with SVR, PTR suggests the models with higher sparsity, which are helpful to conduct fast predictions for future responses. Moreover, it requires about 8 times less computational cost in $n = 10,000$ cases. Such a numerical advantage makes PTR attractive for massive data analysis.

As a concrete illustration, in Figure 1 we show an OGA procedure in terms of Tr-RMSE, Ts-RMSE, and TIME, which is based on a typical example for $n = 1000$ under the uniform sampling scheme. From Figure 1, it is clear that as the number of steps k increase, the OGA estimator better fits the training data. However, an overly large k does not help to improve the predictive accuracy, which also leads to a high computational cost. In this example, the proposed method picks $k = 9$, which balances the predictive accuracy well with the computational cost. The effectiveness of PTR is further illustrated in Figure 2, where a close match is observed between the oracle f^* and the prediction obtained by the PTR-based OGA.

3.2 Remarks for TR1 and TR2

In our simulation studies, we observe that both TR1 and TR2 are less effective in suggesting an appropriate k for OGA. In this subsection, we provide further remarks regarding these two termination rules.

Recall that TR1 requires the minimization of (3). Thus, for picking a $k' > 0$, it requires that

$$\begin{aligned} \|\hat{f}_k - \mathbf{y}\|_n^2 + \kappa \frac{k \log(n)}{n} &\geq \|\hat{f}_{k'} - \mathbf{y}\|_n^2 + \kappa \frac{k' \log(n)}{n} \\ &\geq \kappa \frac{k' \log(n)}{n} \end{aligned} \quad (11)$$

for any $k < k'$. Noting that $\|\hat{f}_k - \mathbf{y}\|_n^2 \leq 4M^2$, (11) implies

$$(k' - k) \leq \frac{4nM^2}{\kappa \log(n)}. \quad (12)$$

Taking $\kappa = 2568M^4(a + 1)$ into (12), we have one necessary condition for $k' > 0$ to be

$$k' \leq \frac{n}{642M^2(a+5)\log n}. \quad (13)$$

However, even in the simple case where $k' = 3$, $M = 1$, and $a = 1$, (13) holds only when n is approximately over 140,000. Thus, in applications, TR1 is likely to be overly stringent for OGA-based learning.

For TR2, the OGA procedure is terminated when (4) is satisfied. However, our simulation examples reveal that (4) is difficult to meet in applications. To gain some insight, one may consider a hypothetical situation as follows. Suppose that $Y = f^*(X) = \theta\phi(X)$, where θ is a model parameter and $\phi \in D_z^*$ is the basis function selected by OGA with $k = 1$. Under this situation, we have the right hand side of (4) bounded by $n\theta^2$, while the left hand side of (4) equals $n|\theta|$ for any $k = 1$. Thus, when $\theta \in (-1, 1)$, condition (4) can be never satisfied and TR2 necessarily leads to $k = n$. This heuristic example indicates that TR2 may lead to long OGA updating for huge n cases and thus is not suitable for massive data analysis.

3.3 Real Data Examples

We now assess the proposed method on seven real-world datasets from a variety of disciplines. Table 3 contains the sample sizes and dimensions of these datasets. Readers may refer to <http://archive.ics.uci.edu/ml/datasets.html> for background and detailed information. For each dataset, we denote by $y_{(j)}$ the j th order statistics of the response $y = \{y_i, i \in \mathbb{N}_n\}$ and normalize the observations by

$$\tilde{y}_i = \frac{y_i - y_{(1)}}{y_{(n)} - y_{(1)}}.$$

The dataset is then randomly split into two parts: a training set with size $n_0 = \lfloor 4n/5 \rfloor$ and a testing set with size $n_1 = n - n_0$, where $\lfloor \cdot \rfloor$ denotes the integer rounding operator. We apply the PTR-based OGA on the training set with the same setting used in simulations and assess its performance based on the testing set. For the learning purpose, the Gaussian kernel is again used, where σ is tuned within $[2^{-5}, 2^5]$ based on a few pilot OGA runs on the full data. The results are summarized in Table 4 with the same indices as Table 1, where the reported statistics are based on 300 independent replications. For comparison, the performance of SVR is also reported. We exclude methods TR1-TR4 from these analyses due to implementation issues.

From Table 4, we observe that both PTR and SVR perform reasonably well in terms of predictive power. This is reflected from their low Tr-RMSE values in all cases. Although SVR tends to be efficient when n is small, it is computationally costly for the large n cases. In comparison, the PTR-based OGA provides accurate predictions with models of high sparsity, where the promising numerical advantages are also observed.

4 Concluding Remarks

In this paper, we developed a new termination rule PTR for OGA-based statistical learning. In the proposed method, the number of updating steps of OGA is determined such that the generalization error of the associated estimator is properly bounded. The new rule is conceptually simple and convenient for implementation, which gears the OGA method to the analysis of massive data. With a proper SDKD, we showed that the PTR-based OGA procedure is strongly consistent and achieves an $O(\sqrt{\log n/n})$ convergence rate to the oracle prediction. In applications, PTR accelerates the OGA learning process by suggesting a sparse model with only $k^* = O(\sqrt{n/\log n})$ basis functions. The promising performance of proposed method has been supported by both simulation and real data examples.

The proposed PTR also reveals that it is sufficient to use a sparse kernel basis to estimate a general regression function. This implies that a good learning model may be directly built from a subset of SDKD with at most k^* elements. This helps to conduct more efficient learning procedures in large- n situations. We leave this interesting topic for the further research.

Acknowledgments

The authors are grateful to the Editor, the AE, and two anonymous referees for their insightful comments and helpful advices, which led to a substantially improved article. This work was supported by National Institute on Drug Abuse (NIDA) grant P50-DA10075 and National Natural Science Foundation of China (NSFC) grant 11131006.

References

1. Aronszajn N. Theory of reproducing kernels. Trans Amer Soc. 1950; 68:337–404.
2. Barron A, Cohen A, Dahmen W, DeVore R. Approximation and learning by greedy algorithm. The Annals of Statistics. 2008; 36:64–94.
3. Bennett, C.; Sharpley, R. Interpolation of Operators. Academic Press; Boston: 1988.
4. Cai T, Wang L. Orthogonal matvhing pursuit for spare signal recovery with noise. IEEE Trans on Information Theory. 2011; 57:4680–4688.
5. Chen H, Li L, Pan Z. Learning rates of multi-kernel regression by orthogonal greedy algorithm. Journal of Statistical Planning and Inference. 2013a; 143:276–282.
6. Chen H, Zhou Y, Tang Y, Li L, Pan Z. Convergence rate of the semi-supervised greedy algorithm. Neural Networks. 2013b; 44:44–50. [PubMed: 23563285]
7. Cortes C, Vapnik V. Support-vector networks. Machine Learning. 1995; 20 (3):273.
8. De Vito E, Caponnetto A, Rosasco L. Model selection for regularized least-squares algorithm in learning theory. Found Comput Math. 2005; 5:59–85.
9. Devore R, Temlyakov V. Some remarks on greedy algorithms. Adv Comput Math. 1996; 5:173–187.
10. Elad, M. Sparse and Redundant Representations. Springer; New York: 2010.
11. Friedman JH, Stuetzle W. Projection pursuit regressions. J Amer Stat Soc. 1981; 76:817–823.
12. Girosi F, Jones M, Poggio T. Regularization theory and neural networks architectures. Neural Computation. 1995; 7:219–269.
13. Györfy, L.; Kohler, M.; Krzyzak, A.; Walk, H. A Distribution-Free Theory of Nonparametric Regression. Springer; Berlin: 2002.
14. Ing C, Lai T. A stepwise regression method and consistent model selection for high-dimentional sparse linear models. Statistica Sinica. 2011; 21:1473–1513.

15. Li R, Lin D, Li B. Statistical inference in massive data sets. *Applied Stochastic Models in Business and Industry*. 2013; 28:339–409.
16. Maiorov V, Ratsaby J. On the degree of approximation by manifolds of finite pseudo-dimension. *Constr Approx*. 1999; 15:291–300.
17. Mallat S, Zhang Z. Matching pursuits with time-frequency dictionaries. *IEEE Trans Signal Processing*. 1993; 41:3397–3415.
18. Mendelson S, Vershinin R. Entropy and the combinatorial dimension. *Invent Math*. 2003; 125:37–55.
19. Micchelli C, Xu Y, Zhang H. Universal kernels. *Journal of Machine Learning Research*. 2006; 7:2651–2667.
20. National Research Council. *Frontiers in Massive Data Analysis*. The National Academies Press; Washington, D.C: 2013.
21. Pati, YC.; Rezaifar, R.; Krishnaprasad, PS. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. *Proc. 27th Annu. Asilomar Conf. Signals, Systems and Computers*; Pacific Grove, CA. 1993. p. 40–44.
22. Rudin, W. *Real and Complex Analysis*. McGraw-Hill; New York: 1974.
23. Rajaraman, A.; Leskovec, J.; Ullman, J. *Mining of Massive Datasets*. 2. Cambridge University Press; Cambridge: 2011.
24. Shi L, Feng Y, Zhou D. Concentration estimates for learning with $L - 1$ -regularizer and data dependent hypothesis spaces. *Appl Comput Harmon Anal*. 2011; 31:286–302.
25. Steinwart, I.; Christmann, A. *Support Vector Machines*. Springer; New York: 2008.
26. Sturm, B.; Christensen, M. Comparison of orthogonal matching pursuit implementations. *The 20th European Signal Processing Conference Proceedings*; 2012. p. 220–224.
27. Temlyakov V. Nonlinear methods of approximation. *Foundations of Comp Math*. 2003; 3:33–107.
28. Tropp J. Greed is good: Algorithmic results for spars approximation. *IEEE Trans on Information Theory*. 2004; 50:2231–2242.
29. Wang H. Forward regression for ultrahigh dimensional variable screening. *Journal of the American Statistical Association*. 2009; 104:1512–1524.
30. Wu Q, Zhou D. Learning with sample dependent hypothesis space. *Comput Math Appl*. 2008; 56:2896–2907.
31. Zhang T. On the consistency of feature selection using greedy least squares regression. *Journal of Machine Learning Research*. 2009; 10:555–568.
32. Zhang T. Sparse recovery with orthogonal matching pursuit under RIP. *IEEE Trans on Information Theory*. 2011; 57:6215–6221.
33. Zhou D, Jetter K. Approximation with polynomial kernels and SVM classifiers. *Adv Comput Math*. 2006; 25:323–344.

Appendix: Proof of Theoretical results

To facilitate our proofs, we first introduce a few technical lemmas. Specifically, let $\mathcal{G}$ be an arbitrary set of functions (function space). We use $\mathcal{N}_\varepsilon(\mathcal{G}, \nu)$ to denote the covering number of $\mathcal{G}$ by balls of radius ε with respect to a measure ν . The lemmas are presented as follows.

Lemma 1

Let $\mathcal{G}$ be a function space defined on a random variable Z . Suppose that, for some constants $C_1, C_2 \geq 0$, we have $|g(Y) - E[g(Y)]| \leq C_1$ and $E[g(Y)^2] \leq C_2 E[g(Y)]$ for any $g \in \mathcal{G}$. Then, for any $\varepsilon > 0$,

$$P \left\{ \sup_{g \in \mathcal{G}} \frac{E[g(Z)] - \frac{1}{n} \sum_{i=1}^n g(z_i)}{\sqrt{E[g(Z)] + \varepsilon}} > \sqrt{\varepsilon} \right\} \leq \mathcal{N}_\varepsilon(\mathcal{G}, \|\cdot\|_\infty) \exp \left\{ -\frac{n\varepsilon}{2C_2 + \frac{2C_1}{3}} \right\},$$

where $\{z_1, \dots, z_n\}$ is an i.i.d sample from Z and $\|\cdot\|_\infty$ is the function L^∞ norm.

Lemma 1 is a direct result from Lemma 2 of Zhou and Jetter (2006), which provides a useful probability concentration inequality to bound a function of random variable.

Lemma 2

Let $\mathcal{V}_k$ be a k -dimensional function space defined on $\mathcal{X}$. Suppose that there exists a constant T such that $|v(x)| \leq T$ for any $v \in \mathcal{V}_k$ and $x \in \mathcal{X}$. Then

$$\log \mathcal{N}_\varepsilon(\mathcal{V}_k, \|\cdot\|_2) \leq ck \log \frac{T}{\varepsilon},$$

where c is a positive constant and $\|\cdot\|_2$ denotes the function L^2 norm.

Lemma 2 is implied by Corollary 2 of Mendelson and Vershynin (2003) together with Property 1 of Maierov and Ratsaby (1999). It shows that the covering number of a bounded functional space can be also bounded properly.

Lemma 3

Let $\mathbf{y} = (y_1, \dots, y_n)^T$ and $\hat{f}_k$ be the k -step estimator defined in Algorithm 1. Then, for any $h \in \text{span}\{D_z^*\}$ and $k \in \mathbb{N}_n$,

$$\|\mathbf{y} - \hat{f}_k\|_n^2 \leq \|\mathbf{y} - h\|_n^2 + \frac{4\|h\|_{l_1}^2}{k},$$

where $\|h\|_{l_1} = \inf \left\{ \sum_{i=1}^n |\theta_i| : h = \sum_{i=1}^n \theta_i K(\mathbf{x}_i, \cdot) / \|K(\mathbf{x}_i, \cdot)\|_n \right\}$.

The proof of Lemma 3 is similar to Theorem 2.3 of Barron et al. (2008). It shows a nice property of the OGA estimator in terms of the empirical approximation error.

Proof of Proposition 1

Recall that the generalization error of $\hat{f}_k$ is defined as

$$\mathcal{L}(\hat{f}_k) = \mathcal{E}(\hat{f}_k) - \mathcal{E}(f^*),$$

where $\mathcal{E}(f) = E(|f(X) - Y|^2)$ for $f \in \mathcal{F}$. Let $\mathcal{E}_n(f) = \|y - f\|_n^2 = \frac{1}{n} \sum_{i=1}^n (y_i - f(x_i))^2$. Then, for an arbitrary $h \in \text{span}\{D_z^*\}$, $\mathcal{L}(\hat{f}_k)$ can be decomposed by

$$\mathcal{L}(\hat{f}_k) = \mathcal{D} + \mathcal{P} + \mathcal{S}, \quad (14)$$

where

$$\begin{aligned} \mathcal{D} &= \mathcal{E}(h) - \mathcal{E}(f^*) = \|h - f^*\|_{\rho_X}^2, \\ \mathcal{P} &= \mathcal{E}_n(\hat{f}_k) - \mathcal{E}_n(h), \\ \mathcal{S} &= \mathcal{E}_n(h) - \mathcal{E}(h) + \mathcal{E}(\hat{f}_k) - \mathcal{E}_n(\hat{f}_k). \end{aligned} \quad (15)$$

By Lemma 3, we readily have

$$\mathcal{P} \leq \frac{4\|h\|_{l_1}^2}{k}. \quad (16)$$

We proceed to prove the theorem by deriving a probability bound for $\mathcal{S}$. Specifically, we further decompose $\mathcal{S}$ by

$$\mathcal{S} = \mathcal{S}_1 + \mathcal{S}_2, \quad (17)$$

where

$$\begin{aligned} \mathcal{S}_1 &= \{\mathcal{E}_n(h) - \mathcal{E}_n(f^*)\} - \{\mathcal{E}(h) - \mathcal{E}(f^*)\}, \\ \mathcal{S}_2 &= \{\mathcal{E}(\hat{f}_k) - \mathcal{E}(f^*)\} - \{\mathcal{E}_n(\hat{f}_k) - \mathcal{E}_n(f^*)\}. \end{aligned}$$

Let us first work on $\mathcal{S}_1$ in (17). Define

$$\begin{aligned} J(Y, X) &= [Y - h(X)]^2 - [Y - f^*(X)]^2 \\ &= [f^*(X) - h(X)][2Y - h(X) - f^*(X)]. \end{aligned}$$

Clearly, we have

$$\mathcal{S}_1 = \frac{1}{n} \sum_{i=1}^n J(y_i, x_i) - E[J(Y, X)].$$

In the model setup, we assume $|Y| \leq M$, which implies that

$$|J| \leq (M + \|h\|_\infty)(3M + \|h\|_\infty) \leq (3M + \|h\|_\infty)^2.$$

Let $\xi = (3M + \|H\|_\infty)^2$. It is then easy to show that

$$|J - E(J)| \leq 2\xi \quad \text{and} \quad E(J^2) \leq \mathcal{D}\xi \quad (18)$$

with $\mathcal{D}$ defined in (15). The bounds in (18) together with Bernstein inequality (Shi, Feng, and Zhou (2011)) imply that

$$\mathcal{S}_1 \leq \frac{4\xi \log \frac{1}{\delta}}{3n} + \sqrt{\frac{2\xi \mathcal{D} \log \frac{1}{\delta}}{n}} \leq \frac{7\xi \log \frac{2}{\delta}}{3n} + \frac{\mathcal{D}}{2} \quad (19)$$

with probability at least $1 - \delta/2$ for any $\delta \in (0, 1)$.

We now turn to bound $\mathcal{S}_2$ in (17). Recall that V_k in Algorithm 1 is the active set formed by the k basis functions from a k -step OGA procedure. Let $\mathcal{F}_k = \{T_M[v] : v \in \text{span}\{V_k\}\}$ and g be an arbitrary element from

$$\mathcal{G}_k = \{g(X, Y) = \{f(X) - Y\}^2 - \{f^*(X) - Y\}^2, f \in \mathcal{F}_k\}.$$

Since both $|Y|$ and $|f^*|$ are bounded by M , it is straightforward to show that $|g| \leq 8M^2$ and $|g - E(g)| \leq 16M^2$. Also, we have

$$\begin{aligned} E(g^2) &= E \left[\{f(X) - f^*(X)\}^2 \{f(X) - Y + f^*(X) - Y\}^2 \right] \\ &\leq 16M^2 E(g). \end{aligned}$$

Thus, Lemma 1 becomes applicable to $\mathcal{G}_k$ with $C_1 = C_2 = 16M^2$. Note that

$$E(g) + \mathcal{L}(f) = \mathcal{E}(f) - \mathcal{E}(f^*), \quad \frac{1}{n} \sum_{i=1}^n g(y_i, x_i) = \mathcal{E}_n(f) - \mathcal{E}_n(f^*)$$

for some corresponding $f \in \mathcal{F}_k$. This together with Lemma 1 implies that

$$\sup_{f \in \mathcal{F}_k} \left\{ \frac{\mathcal{L}(f) - \{\mathcal{E}_n(f) - \mathcal{E}_n(f^*)\}}{\sqrt{\mathcal{L}(f) + \varepsilon}} \right\} \leq \sqrt{\varepsilon} \quad (20)$$

with probability at least

$$1 - \mathcal{N}_{\varepsilon/4}(\mathcal{G}_k, \|\cdot\|_\infty) \exp \left\{ -\frac{3n\varepsilon}{128M^2} \right\}.$$

Note that, for any $f_1, f_2 \in \mathcal{F}_k$ and the corresponding $g_1, g_2 \in \mathcal{G}_k$, we have

$$\begin{aligned}\|g_1 - g_2\|_\infty &= \max_{x,y} |(f_1(x) - y)^2 - (f_2(x) - y)^2| \\ &\leq 4M \|f_1 - f_2\|_\infty,\end{aligned}$$

where (x, y) denotes an arbitrary realization from (X, Y) . This implies that

$$\begin{aligned}\mathcal{N}_{\varepsilon/4}(\mathcal{G}_k, \|\cdot\|_\infty) &\leq \mathcal{N}_{\varepsilon/(16M)}(\mathcal{F}_k, \|\cdot\|_\infty) \\ &\leq \mathcal{N}_{\varepsilon/(16M)}(\mathcal{F}_k, \|\cdot\|_2) \\ &\leq \exp\left\{ck \log \frac{16M^2}{\varepsilon}\right\},\end{aligned}\quad (21)$$

where the last inequality follows from Lemma 2 with $T = M$. By (20) and (21), we have

$$P\left\{\mathcal{J}_2 \leq \frac{1}{2}\mathcal{L}(\hat{f}_k) + \varepsilon\right\} \geq 1 - \exp\left\{ck \log \frac{16M^2}{\varepsilon} - \frac{3n\varepsilon}{128M^2}\right\}. \quad (22)$$

To further specify (22), let

$$h(\varepsilon) = ck \log \frac{16M^2}{\varepsilon} - \frac{3n\varepsilon}{128M^2}$$

and ε_0 be the value of ε such that $h(\varepsilon_0) = \log(\delta/2)$ for the same δ used in (19). It can be shown that, by choosing

$$\varepsilon_1 = \omega \frac{k \log n + \log \frac{2}{\delta}}{n}$$

with some constant $\omega > 0$, we have $h(\varepsilon_1) \leq h(\varepsilon_0)$. Since $h(\cdot)$ is a decreasing function, this implies $\varepsilon_1 \leq \varepsilon_0$, and therefore

$$P\left\{\mathcal{J}_2 \leq \frac{1}{2}\mathcal{L}(\hat{f}_k) + \varepsilon_1\right\} \geq 1 - \delta/2. \quad (23)$$

Combining the results from (19) and (23), we have

$$P\left\{\mathcal{J} \leq \frac{\mathcal{D} + \mathcal{L}(\hat{f}_k)}{2} + \frac{7\xi \log \frac{2}{\delta}}{3n} + \varepsilon_1\right\} \geq 1 - \delta. \quad (24)$$

Inequality (24) together with (15) and (16) further implies that, with probability at least $1 - \delta$,

$$\begin{aligned}\mathcal{L}(\hat{f}_k) &\leq 3\|f^*-h\|_{\rho_X}^2 + \frac{8\|h\|_{l_1}^2}{k} + \frac{14\xi\log\frac{2}{\delta}}{3n} + 2\varepsilon_1 \\ &\leq 3\|f^*-h\|_{\rho_X}^2 + \frac{8\|h\|_{l_1}^2}{k} + \frac{28\log\frac{2}{\delta}\|h\|_{\infty}^2}{3n} + \frac{2\omega k\log n + 6M^2 + \log\frac{2}{\delta}}{n}.\end{aligned}$$

Noting $2\log(2/\delta) > 1$, we then have, for a sufficiently large n ,

$$\begin{aligned}\mathcal{L}(\hat{f}_k) &\leq 3\|f^*-h\|_{\rho_X}^2 + \frac{16\log\frac{2}{\delta}\|h\|_{l_1}^2}{k} + \frac{28\log\frac{2}{\delta}\|h\|_{\infty}^2}{3n} + \frac{4\omega\log\frac{2}{\delta}k\log n}{n} \\ &\leq C \left[\|f^*-h\|_{\rho_X}^2 + \log\frac{2}{\delta} \left(\frac{\|h\|_{l_1}^2}{k} + \frac{\|h\|_{\infty}^2}{n} + \frac{k\log n}{n} \right) \right]\end{aligned}$$

with probability at least $1 - \delta$, where $C = \max\{16, 4\omega\}$. This completes the proof of Proposition 1.

Proof of Theorem 1

Let $\mathcal{H}_{\infty} = \lim_{n \rightarrow \infty} \text{span}\{D_z^*\}$. For an arbitrary $h \in \mathcal{H}_{\infty}$, we decompose $\mathcal{L}(\hat{f}_k)$ by

$$\mathcal{L}(\hat{f}_k) = B_1 + B_2 + B_3 + B_4, \quad (25)$$

where

$$\begin{aligned}B_1 &= \|h - \mathbf{y}\|_n^2 - \mathcal{E}(h), & B_2 &= \mathcal{E}(\hat{f}_{k^*}) - \|\hat{f}_{k^*} - \mathbf{y}\|_n^2, \\ B_3 &= \mathcal{E}(h) - \mathcal{E}(f^*), & B_4 &= \|\hat{f}_k - \mathbf{y}\|_n^2 - \|h - \mathbf{y}\|_n^2.\end{aligned}$$

Since $\mathcal{L}(\hat{f}_k) \rightarrow 0$, the theorem is proved if

$$P \left\{ \lim_{n \rightarrow \infty} B_j \leq 0 \right\} = 1 \quad (26)$$

for $j = 1, 2, 3, 4$. By the strong law of large numbers, (26) readily holds for B_1 . Thus, it suffices to show (26) for B_2, B_3 , and B_4 .

We first show (26) for B_2 . Let

$$\mathcal{G}' = \{g(X, Y) = [f(X) - Y]^2 : f \in \mathcal{F}_k\}$$

with $\mathcal{F}_k$ same defined as in the proof of Proposition 1. Since $|Y| \leq M$, it is straight-forward to show that, for any $g \in \mathcal{G}'$,

$$|g| \leq 4M^2, \quad |g - E(g)| \leq 8M^2, \quad E(g^2) \leq 4M^2 E(g)$$

Thus, by applying Lemma 1 to $\mathcal{G}'$ with $C_1 = C_2 = 8M^2$ and some arbitrary $\varepsilon > 0$, we have

$$\sup_{f \in \mathcal{F}_k} \left\{ \frac{\mathcal{E}(f) - \|f - \mathbf{y}\|_n^2}{\sqrt{\mathcal{E}(f)} + \varepsilon} \right\} > \sqrt{\varepsilon} \quad (27)$$

with probability at most

$$\mathcal{N}_{\varepsilon/4}(\mathcal{G}', \|\cdot\|_\infty) \exp \left\{ -\frac{3n\varepsilon}{64M^2} \right\}.$$

Following the same arguments in (21), we have

$$\mathcal{N}_{\varepsilon/4}(\mathcal{G}', \|\cdot\|_\infty) \leq \exp \left\{ ck \log \frac{16M^2}{\varepsilon} \right\}$$

for some positive constant c . This together with (27) implies that

$$\mathcal{E}(\hat{f}_k) - \|\hat{f}_k - \mathbf{y}\|_n^2 > [\varepsilon(4M^2 + \varepsilon)]^{1/2} \quad (28)$$

with probability at most

$$P_k = \exp \left\{ ck \log \frac{16M^2}{\varepsilon} - \frac{3n\varepsilon}{64M^2} \right\}. \quad (29)$$

By setting $k = k^* = T \sqrt{n/\log n}$ with some constant $T > 0$, we have $\sum_{n=1}^{\infty} P_{k^*} < \infty$. Thus, by Borel-Cantelli lemma, (28) and (29) imply that

$$P \left\{ \lim_{n \rightarrow \infty} B_2 \leq [\varepsilon(4M^2 + \varepsilon)]^{1/2} \right\} = 1. \quad (30)$$

Since ε is arbitrary, (30) further implies that (26) holds for B_2 .

We now proceed to show (26) for B_3 and B_4 . Since $|\hat{f}^*(X)| \leq M$, we have $\|\hat{f}^*\|_{\rho_X} \leq M$. By Theorem A.1 of Györfy et al. (2002), for any $\varepsilon' > 0$, there exists a $\hat{f} \in \mathcal{C}(\mathcal{X})$ such that $\|\hat{f} - \hat{f}^*\|_{\rho_X} \leq \varepsilon'$. Also, Condition C1 implies that $\mathcal{H}_\infty$ is dense in H_K . These results together with Condition C2 imply that, for any $\varepsilon > 0$, there exists a $h_\varepsilon \in \mathcal{H}_\infty$ such that

$$\|h_\varepsilon - f^*\|_{\rho_X}^2 \leq \varepsilon. \quad (31)$$

By choosing $h = h_\varepsilon$ in (25), we have (26) holds for B_3 due to the arbitrariness of ε . Meanwhile, by setting $k = k^*$, Lemma 3 implies that

$$B_4 \leq \frac{4\|h_\varepsilon\|_{l_1}^2}{k^*}. \quad (32)$$

Since D_z^* is a normalized dictionary, (31) implies that $\|h_\varepsilon\|_{\ell_1} < \infty$. Thus, the right hand side of (32) goes to zero as $n \rightarrow \infty$, which implies that (26) holds for B_4 . The theorem is therefore proved.

Proof of Theorem 2

Proposition 1 implies that, for any $h \in \text{span}\{D_z^*\}$ and n large enough,

$$\mathcal{L}(\hat{f}_k) \leq C \left\{ \|f^* - h\|_{\rho_X}^2 + \log \frac{2}{\delta} \left(\frac{\|h\|_{l_1}^2}{k} + \frac{\|h\|_{\infty}^2 + k \log n}{n} \right) \right\}$$

with probability at least $1 - \delta$ for $\delta \in (0, 1)$. When Condition C3 is satisfied with $r > 0.5$, we have $\|H\|_{\ell_1} \leq B$ and $\|f^* - H\|_{\rho_X} \leq \|f^* - H\|_{\infty} \leq Bn^{-1/2}$ for some $h' \in \text{span}\{D_z^*\}$. Since $K(\cdot, \cdot)$ is continuous and $\mathcal{X}$ is compact, Condition C3 also implies that $\|h'\|_{\infty}^2$ is bounded by some positive constant B' . Based on these results, we have

$$\mathcal{L}(\hat{f}_k) \leq C \left\{ B^2 n^{-1} + \log \frac{2}{\delta} \left(\frac{B^2}{k} + \frac{B' + k \log n}{n} \right) \right\}$$

with probability at least $1 - \delta$. By setting $k = k^* = T(n/\log n)^{1/2}$, we have

$$P \left\{ \mathcal{L}(\hat{f}_k) > C' \log \frac{2}{\delta} \sqrt{\frac{\log n}{n}} \right\} \leq \delta$$

for some generic positive constant C' with a sufficiently large n . Let $t = C' \log \frac{2}{\delta} (\log n / n)^{1/2}$, we then have

$$\begin{aligned}
E[\mathcal{L}(\hat{f}_k)] &= \int_0^\infty P\{\mathcal{L}(\hat{f}_k) > t\} dt \\
&\leq \int_0^\infty 2 \exp \left\{ -\frac{t}{C'} \sqrt{\frac{n}{\log n}} \right\} dt \\
&\leq 2C' \sqrt{\frac{\log n}{n}}.
\end{aligned}$$

The theorem is therefore proved.

Figure 1.
An illustration example for the uniform sampling scheme with $n = 1000$.

Figure 2.
Plot of the PTR-based OGA for the illustration example.

Table 1

Simulation results under the uniform sampling scheme

Methods	Tr-RMSE	Ts-RMSE	TIME	<i>k</i>
<i>n</i> = 1000				
PTR	.099 _(.002)	.010 _(.002)	0.2	9
TR1	.502 _(.011)	.389 _(.012)	52.2	0
TR2	.099 _(.002)	.014 _(.002)	52.2	1000
TR3	.106 _(.002)	.039 _(.006)	0.4	22
TR4	.139 _(.007)	.096 _(.010)	0.3	9
SVR	.099 _(.002)	.019 _(.003)	1.4	315
<i>n</i> = 5000				
PTR	.099 _(.001)	.006 _(.001)	1.9	10
TR1	.402 _(.005)	.390 _(.011)	1256.3	0
TR2	.099 _(.001)	.006 _(.001)	1256.3	5000
TR3	.103 _(.001)	.024 _(.004)	1.9	18
TR4	.111 _(.003)	.048 _(.007)	1.8	15
SVR	.099 _(.001)	.007 _(.001)	15.7	1584
<i>n</i> = 10000				
PTR	.100 _(.001)	.004 _(.001)	6.2	9
TR1	.403 _(.003)	.389 _(.011)	16533.1	0
TR2	.010 _(.001)	.004 _(.001)	16533.1	10000
TR3	.102 _(.001)	.021 _(.003)	6.1	17
TR4	.107 _(.002)	.038 _(.005)	6.1	15
SVR	.099 _(.002)	.005 _(.001)	44.3	3163

Table 2

Simulation results under the truncated normal sampling scheme

Methods	Tr-RMSE	Ts-RMSE	TIME	<i>k</i>
<i>n</i> = 1000				
PTR	.099 _(.002)	.017 _(.004)	0.2	9
TR1	.502 _(.011)	.389 _(.012)	52.2	0
TR2	.099 _(.002)	.023 _(.004)	52.2	1000
TR3	.104 _(.003)	.041 _(.009)	0.4	17
TR4	.142 _(.006)	.098 _(.012)	0.3	8
SVR	.099 _(.002)	.019 _(.005)	1.4	317
<i>n</i> = 5000				
PTR	.099 _(.001)	.009 _(.002)	1.9	10
TR1	.501 _(.005)	.388 _(.011)	1256.3	0
TR2	.099 _(.001)	.008 _(.002)	1256.3	5000
TR3	.103 _(.001)	.028 _(.005)	1.8	16
TR4	.118 _(.002)	.049 _(.008)	1.8	13
SVR	.099 _(.001)	.009 _(.002)	15.3	1584
<i>n</i> = 10000				
PTR	.100 _(.001)	.007 _(.001)	6.1	10
TR1	.501 _(.003)	.391 _(.012)	16533.1	0
TR2	.099 _(.001)	.006 _(.001)	16533.1	10000
TR3	.105 _(.001)	.023 _(.004)	6.2	16
TR4	.106 _(.002)	.039 _(.006)	6.2	14
SVR	.100 _(.001)	.007 _(.001)	45.2	3171

Table 3

Specifications of real-world regression datasets

Data sets	Sample size (n)	Dimension (p)
Boston housing	506	13
ENB2012	768	8
Bank	4100	8
Delta ailerons	7130	5
Delta elevator	9517	6
House8L	22784	8
CASP	45730	9

Table 4

Results for analyzing the real-world datasets

Data sets	PTR				SVR			
	Tr-RMSE	Ts-RMSE	TIME	k	Tr-RMSE	Ts-RMSE	TIME	k
Boston h.	.073 _(.003)	.081 _(.011)	0.5	24	.070 _(.002)	.079 _(.015)	0.3	62
ENB2012	.067 _(.002)	.069 _(.003)	0.6	16	.067 _(.001)	.069 _(.002)	0.6	56
Bank	.043 _(.001)	.043 _(.001)	1.8	38	.045 _(.001)	.045 _(.001)	7.3	94
Delta ail.	.038 _(.001)	.039 _(.001)	3.3	23	.041 _(.001)	.042 _(.001)	9.4	147
Delta ele.	.054 _(.001)	.053 _(.001)	5.6	52	.054 _(.001)	.054 _(.002)	87.2	489
House8L	.062 _(.001)	.064 _(.002)	32.4	104	.072 _(.001)	.074 _(.002)	232.4	1408
CASP	.049 _(.001)	.052 _(.002)	188.7	168	.053 _(.001)	.056 _(.002)	536.5	1276