Statistica Sinica **26** (2016), 841-860
doi:http://dx.doi.org/10.5705/ss.202014.0068

PREDICTION-BASED TERMINATION RULE FOR GREEDY LEARNING WITH MASSIVE DATA

Chen Xu[1], Shaobo Lin[2], Jian Fang[2] and Runze Li[3]

University of Ottawa[1], Xi'an Jiaotong University[2]
and The Pennsylvania State University[3]

Abstract: The appearance of massive data has become increasingly common in contemporary scientific research. When the sample size n is huge, classical learning methods become computationally costly in regression analysis. 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 simulation and data examples.

Key words and phrases: Forward regression, greedy algorithms, kernel methods, massive data, 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 computational infeasible. Developing effective and efficient approaches to analyze 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), for example.

Unlike data collected from designed experiments, massive data sets are sometimes available only 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 $z = \{(y_i, \boldsymbol{x}_i); i = 1, \ldots, n\}$ be n independent observations from the model

$$Y = f^*(X) + \epsilon, \tag{1.1}$$

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 goal of learning is to explore the relationship $f(X) : X \to Y$ 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 $f^*(X) = E(Y|X)$. 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 as

$$D_z = \{K(\boldsymbol{x}_i, \cdot) : i = 1, \ldots, n\}, \tag{1.2}$$

where $K(\cdot, \cdot)$: $R^p \times R^p \to R$ 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, $H_z = \text{span}\{D_z\} = \{g : g(\cdot) = \sum_{i=1}^n \theta_i K(\boldsymbol{x}_i, \cdot)$ 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 learning purposes, various regularization methods have been proposed. 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 causes special problems for algorithmic design and memory storage. In addtion, 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). 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 a 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 a focus on prediction. In OGA-based learning, an appropriate termination rule is crucial. This is analogous to 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

$$k' = \arg\min_{k \in \mathbb{N}_n} \Big\{ \sum_{i=1}^{n} (\hat{f}_k\,(\boldsymbol{x}_i) - y_i)^2 + \kappa k \log(n) \Big\}, \tag{1.3}$$

where $\mathbb{N}_n = \{1, 2, \ldots, n\}$ 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 (1.3) can be time consuming in large n situations. Moreover, as Barron et al. (2008) recommended a large value for κ, rule (1.3) tends to select a k' that may be overly small in application. Addressing the same issue under a slightly different framework, Chen, Li, and Pan (2013a), Chen et al. (2013b) proposed terminating OGA when

$$\sum_{i=1}^{n} (\hat{f}_k\,(\boldsymbol{x}_i) - y_i)^2 + n \left\| \hat{f}_k \right\|_{l_1} \leq \sum_{i=1}^{n} y_i^2 \tag{1.4}$$

is satisfied with $\|f\|_{l_1} = \inf\{\sum_i |\theta_i| : f = \sum_i \theta_i K(\boldsymbol{x}_i, .)\}$. Despite its theoretical feasibility, (1.4) often leads to a long updating procedure that makes it inefficient for massive data analysis; see Section 3.2. Recently, Ing and Lai (2011) studied the predictive performance of OGA in high-dimensional linear models. They suggested that, if f^* is linear in 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 prompts 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 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. This rule treats $\mathcal{L}(\hat{f}_k) =$

$E(\hat{f}_k - f^*)^2$ (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 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 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 remarks. The proofs of theorems are given in the online Supplementary Material.

2. Termination for the OGA-based Learning

2.1. The OGA framework

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

OGA is a stepwise method that seeks the best prediction from a series of expanding subspaces of $\mathcal{F} : \mathcal{X} \to R^1$. 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 can 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

$$< f, g >_n = \frac{1}{n}\sum_{i=1}^{n} f(\boldsymbol{x}_i)g(\boldsymbol{x}_i), \quad \|f\|_n^2 = \frac{1}{n}\sum_{i=1}^{n} |f(\boldsymbol{x}_i)|^2.$$

We treat an arbitrary vector $\boldsymbol{u} = (u_1, \ldots, u_n)^T \in R^n$ as a degenerated function on $\mathcal{X}$, so that

$$< f, \boldsymbol{u} >_n = \frac{1}{n}\sum_{i=1}^{n} f(\boldsymbol{x}_i)u_i(\boldsymbol{x}_i) = \frac{1}{n}\sum_{i=1}^{n} u_i f(\boldsymbol{x}_i).$$

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

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

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

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

where

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

is a truncating operator.

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

Algorithm 1 (OGA)**.**

```
Input: $z$, $M$, $K(\cdot,\cdot)$, $k^*$
Output: $\hat{f}_{k^*}$
Compute $D_z^*$ and set $k = 1$, $V_0 = \varnothing$, $\boldsymbol{r}_0 = \boldsymbol{y} = (y_1, \ldots, y_n)^T$.
When $k \le k^*$, recursively do substeps a-d:
    a Update the active set by $V_k = V_{k-1} \cup \{g_k\}$, where
          $g_k = arg \max_{g \in D_z^*} |< \boldsymbol{r}_{k-1}, g >_n|$.
    b Let $\boldsymbol{g}_k = (g_1, \ldots, g_k)^T$. Compute
          $\hat{\boldsymbol{\beta}}_k = arg \min_{\boldsymbol{\beta}_k} \|\boldsymbol{y} - \boldsymbol{g}_k^T \boldsymbol{\beta}_k\|_n^2$,
      and update the estimate by $f_k = \boldsymbol{g}_k^T \hat{\boldsymbol{\beta}}_k$.
    c Update the residual by $\boldsymbol{r}_k = \boldsymbol{y} - (f_k(\boldsymbol{x}_1), f_k(\boldsymbol{x}_2), ..., f_k(\boldsymbol{x}_n))^T$.
    d Increase the number of steps $k$ by one.
Output $\hat{f}_{k^*} = T_M[f_k]$.
```

Here 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 that avoids direct storage and operations on the full model. This framework brings many computational benefits to 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. The efficiency of OGA relies heavily on the choice of k^*, which indicates the number of basis functions included in $\hat{f}_{k^*}$. For statistical learning, a good choice of k^* should make $\hat{f}_{k^*}$ an accurate prediction of Y.

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

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

as a norm with respect to ρ_X. Under model (1.1), the predictive error of an arbitrary $f(X)$ can be measured by

$$\mathcal{E}(f) = E(|f(X) - Y|^2), \tag{2.1}$$

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

$$\mathcal{L}(\hat{f}_k) = \mathcal{E}(\hat{f}_k) - \mathcal{E}(f^*) = \|\hat{f}_k - f^*\|_{\rho_X}^2 \tag{2.2}$$

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 (2.2) 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. 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 $\mathcal{L}(\hat{f}_k)$.

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

$$\mathcal{L}(\hat{f}_k) \le C\Big\{\|f^* - h\|_{\rho_X}^2 + \log\frac{2}{\delta}\Big(\frac{\|h\|_{l_1}^2}{k} + \frac{\|h\|_\infty^2 + k\log n}{n}\Big)\Big\}, \tag{2.3}$$

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

The proof is given in the online Supplementary Material. 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 $\|f^*\|_\infty \leq M$, (2.3) attains its minimum with an h that is bounded above. Thus, a sensible k^* may be selected based on (2.3) with $\|h\|_{l_1} \leq T$ for some constant $T > 0$. Heuristically, it is reasonable to set

$$k^* = \arg\min_k \left\{ \frac{T^2}{k} + \frac{k \log n}{n} \right\} = T\sqrt{\frac{n}{\log n}}, \tag{2.4}$$

which minimizes (2.3) in k with $\|h\|_{l_1} = T$. Because (2.4) is derived from a probability bound of $\mathcal{L}(\hat{f}_k)$, we name it as the prediction-based termination rule (PTR).

As opposed to 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 of 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 can further interpret the model parameters and conduct fast predictions for the future responses. PTR is convenient for implementation, because it does not involve large-scale ad hoc evaluations in the OGA process. Simplicity gives the new rule an additional advantage for the analysis of massive data.

The choice of T in (2.4) should reflect the prior information on f^*. Our empirical experience suggests that choosing $T \leq 1 + \log p$ is usually adequate for conducting accurate prediction. The term $\log p$ is added to reflect the intuition that the 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 was selected among a few candidates based on the 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 Algorithm 1 with k^* decided by PTR (2.4), 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 closely as the sample size increases. Such a property is typically referred to as consistency. 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 convenience, let $\mathcal{C}(\mathcal{X})$ be the family of continuous functions on $\mathcal{X}$ and $\text{supp}(\cdot)$ denote the support set of a measure. We assume the following technical conditions are satisfied.

C1 $\text{supp}(\rho_X) = \mathcal{X}$.

C2 Let $\mathcal{H}_K = \overline{\text{span}}\{K_X : K_X = K(X,.), X \in \mathcal{X}\}$ 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 $\mathcal{X}$ is an effective sampling space for random variable X. Condition C2 is the notion of universal kernel proposed by Micchelli, Xu, and Zhang (2006). Under our model setup, many choices of $K(\cdot,\cdot)$ satisfy this condition; they include the Gaussian kernel as a special case. One can refer to Micchelli, Xu, and Zhang (2006) for more discussion on universal kernels.

For asymptotic analysis, we associate both dictionary D_z^* and prediction $\hat{f}_{k^*}$ with sample size n.

Theorem 1. *Let $\hat{f}_{k^*}$ be the output obtained from Algorithm* 1 *with k^* decided by GTR. If* C1 *and* C2 *are satisfied, then $P\{\lim_{n\to\infty} \mathcal{L}(\hat{f}_{k^*}) = 0\} = 1$.*

The proof is given in the online Supplementary Material. Theorem 1 shows that $\hat{f}_{k^*}$ converges to the oracle prediction in an almost sure sense. Since $\mathcal{X}$ is an arbitrary compact set with a positive Lebesgue measure in R^p, Theorem 1 applies to a wide range of distributions on X. Our result 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 likely requires a lengthy proof. We leave this issue for future research.

Under our model setup, $\hat{f}_{k^*}$ has a diminishing generalization risk as the sample size increases.

Corollary 1. *Under conditions of Theorem* 1, *as $n \to \infty$, $E[\mathcal{L}(\hat{f}_{k^*})] \to 0$.*

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 $\mathcal{L}(f^*)$.

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, we investigate the convergence rate of $E[\mathcal{L}(\hat{f}_{k^*})]$ under the following technical 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)). It is satisfied when f^* has a representation in $\mathcal{H}_K$ with a bounded l_1 norm and a power-decayed tail. The condition might be too stringent when the dimensionality of X is high. We only use it to gain some understanding of the proposed PTR and do not intend to make this condition the weakest possible. See Barron et al. (2008) for more detailed discussion about C3.

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

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

The proof is given in the online Supplementary Material. 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 cannot be generally improved without further conditions (Devore and Temlyakov (1996)). 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 assessed for the truncated dictionary as well as the observational noise.

Remark 2. The convergence rate in 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 it.

In Algorithm 1, computing $\hat{\boldsymbol{\beta}}_k$ from step **b** involves a least squares problem with the solution

$$\hat{\boldsymbol{\beta}}_k = (\boldsymbol{Q}_k^T \boldsymbol{Q}_k)^{-1} \boldsymbol{Q}_k^T \boldsymbol{y}, \tag{2.5}$$

where $\boldsymbol{Q}_k$ is a $n \times k$ matrix with the element in the ith row and jth column being $g_j(\boldsymbol{x}_i)$. When a long OGA run is needed, a direct implementation of (2.5) might be numerically less efficient. To further reduce computational burden, an iterative form of $\hat{\boldsymbol{\beta}}_k$ is often used in practice. Specifically, let $\boldsymbol{Q}_k^+ = (\boldsymbol{Q}_k^T \boldsymbol{Q}_k)^{-1} \boldsymbol{Q}_k^T$ and $\boldsymbol{Q}_k = (\boldsymbol{q}_1, \ldots, \boldsymbol{q}_k)$ with $\boldsymbol{q}_j = (g_j(\boldsymbol{x}_1), \ldots, g_j(\boldsymbol{x}_n))^T$ for $j \in \mathbb{N}_k$.

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

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

where $c_k = \boldsymbol{q}_{k+1}^T \boldsymbol{r}_k [\boldsymbol{q}_{k+1}^T (\boldsymbol{I} - \boldsymbol{Q}_k \boldsymbol{Q}_k^+) \boldsymbol{q}_{k+1}]^{-1}$.

See Elad (2010) or Sturm and Christensen (2012) for the proof. Proposition 2 provides a convenient tool for implementing OGA. Specifically, given $\boldsymbol{Q}_k^+$ at the k-th step, $\hat{\boldsymbol{\beta}}_{k+1}$ can be efficiently computed without complex operations. 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* $k = 1$, $\boldsymbol{r}_1 = \boldsymbol{y}$, $V_1 = \{g_1\}$, *where*

$$g_1 = \arg\max_{g \in D_z^*} |< \boldsymbol{r}_1, g >_n| .$$

Let

$$\hat{\boldsymbol{\beta}}_1 = \frac{< \boldsymbol{y}, g_1 >_n}{< g_1, g_1 >_n}, \quad \boldsymbol{Q}_1^+ = \frac{n^{-1} \boldsymbol{q}_1^T}{< g_1, g_1 >_n}, \quad f_1 = \hat{\boldsymbol{\beta}}_1 g_1.$$

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

a *Let* $V_{k+1} = V_k \cup \{g_{k+1}\}$*, where* $g_{k+1} = \arg\max_{g \in D_z^*} |< \boldsymbol{r}_k, g >_n|$.

b *Set*

$$a_k = \frac{1}{\boldsymbol{q}_{k+1}^T[\boldsymbol{I} - \boldsymbol{Q}_k\boldsymbol{Q}_k^+]\boldsymbol{q}_{k+1}}, \quad c_k = a_k\boldsymbol{q}_{k+1}^T\boldsymbol{r}_k.$$

Update

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

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

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

c *Update* $\boldsymbol{r}_{k+1} = \boldsymbol{y} - \boldsymbol{Q}_{k+1}\hat{\boldsymbol{\beta}}_{k+1}$.

d *Increase the number of steps* k *by one.*

Output $\hat{f}_{k^*} = T_M[f_k]$.

3. Numerical Studies

We evaluated the finite sample performance of PTR with simulations and data examples. We compared PTR with other popular termination rules in terms of both computational efficiency and predictive accuracy. We conducted comparisons between the PTR-based OGA and regularization methods. All numerical studies were implemented by MATLAB 8.2 on a windows workstation with 8-core 3.07GHz CPUs.

3.1. Simulations

We assessed the performance of PTR on a hypothetical learning problem with $p = 1$. We generated independent observations $z = \{(y_i, \boldsymbol{x}_i), i \in \mathbb{N}_n\}$ based on model (1.1) 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 $\epsilon \sim N(0, 0.1)$. We adopted two sampling schemes: in the first scheme, we sampled X from a uniform distribution on $[0, 1]$; in the second scheme, we sampled 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 set $n =$1,000, $5,000$, $10,000$ and estimated f^* through analyzing z.

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

$$K(\boldsymbol{x}_1, \boldsymbol{x}_2) = \exp\Big(- \frac{\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2^2}{\tau^2} \Big), \tag{3.1}$$

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

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

where $k(T)$ is the largest integer smaller than the PTR value with respect to T. For comparison, we also carried out OGA based on termination rules (1.3) and (1.4). For (1.3), Barron et al. (2008) suggested using $\kappa = 2,568M^4(a+5)$, where $a \geq 1$ indicates the total number of basis functions taken to be n^a. We followed their recommendation by setting $M = 1$ and $a = 1$ according to our setup. Moreover, we compared 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. We refer to these rules as TR1, TR2, TR3, and TR4, respectively, based on their orders of presentation. As a benchmark, we further report the results from the support vector regression (SVR). In our simulation studies, we implemented SVR by MATLAB package LIBSVM, where a default 0.1-insensitive loss was used and λ was similarly tuned by HDAIC.

The performance of each method was evaluated on its goodness of fit, predictive accuracy, model complexity, and computational cost. We measured goodness of fit through the training root mean squared error

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

To assess the predictive accuracy, we generated an independent testing set $\tilde{z} = \{(\tilde{y}_i, \tilde{\boldsymbol{x}}_i), i \in N_T \text{ for } T = 1,000\}$ from $Y = f^*(X)$ with $X \sim U[0,1]$ and computed the testing root mean squared error

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

The model complexity, k, was reported as the number of basis functions included in each fitted model. For efficiency comparisons, we recorded the absolute computing time (in seconds) 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

Table 1. Simulation results under the uniform sampling scheme.

Methods	Tr-RMSE	Ts-RMSE	TIME	k
		$n = 1,000$		
PTR	0.099 (0.002)	0.010 (0.002)	0.2	9
TR1	0.502 (0.011)	0.389 (0.012)	52.2	0
TR2	0.099 (0.002)	0.014 (0.002)	52.2	1,000
TR3	0.106 (0.002)	0.039 (0.006)	0.4	22
TR4	0.139 (0.007)	0.096 (0.010)	0.3	9
SVR	0.099 (0.002)	0.019 (0.003)	1.4	315
		$n = 5,000$		
PTR	0.099 (0.001)	0.006 (0.001)	1.9	10
TR1	0.402 (0.005)	0.390 (0.011)	1,256.3	0
TR2	0.099 (0.001)	0.006 (0.001)	1,256.3	5,000
TR3	0.103 (0.001)	0.024 (0.004)	1.9	18
TR4	0.111 (0.003)	0.048 (0.007)	1.8	15
SVR	0.099 (0.001)	0.007 (0.001)	15.7	1,584
		$n = 10,000$		
PTR	0.100 (0.001)	0.004 (0.001)	6.2	9
TR1	0.403 (0.003)	0.389 (0.011)	16,533.1	0
TR2	0.010 (0.001)	0.004 (0.001)	16,533.1	10,000
TR3	0.102 (0.001)	0.021 (0.003)	6.1	17
TR4	0.107 (0.002)	0.038 (0.005)	6.1	15
SVR	0.099 (0.002)	0.005 (0.001)	44.3	3,163

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 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 models with higher sparsity; these models are useful for conducting 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 an illustration, in Figure 1 we show an OGA procedure in terms of Tr-RMSE, Ts-RMSE, and TIME based on a typical example for n =1,000 under the

Table 2. Simulation results under the truncated normal sampling scheme.

Methods	Tr-RMSE	Ts-RMSE	TIME	k
		$n = 1,000$		
PTR	0.099 (0.002)	0.017 (0.004)	0.2	9
TR1	0.502 (0.011)	0.389 (0.012)	52.2	0
TR2	0.099 (0.002)	0.023 (0.004)	52.2	1,000
TR3	0.104 (0.003)	0.041 (0.009)	0.4	17
TR4	0.142 (0.006)	0.098 (0.012)	0.3	8
SVR	0.099 (0.002)	0.019 (0.005)	1.4	317
		$n = 5,000$		
PTR	0.099 (0.001)	0.009 (0.002)	1.9	10
TR1	0.501 (0.005)	0.388 (0.011)	1,256.3	0
TR2	0.099 (0.001)	0.008 (0.002)	1,256.3	5,000
TR3	0.103 (0.001)	0.028 (0.005)	1.8	16
TR4	0.118 (0.002)	0.049 (0.008)	1.8	13
SVR	0.099 (0.001)	0.009 (0.002)	15.3	1,584
		$n = 10,000$		
PTR	0.100 (0.001)	0.007 (0.001)	6.1	10
TR1	0.501 (0.003)	0.391 (0.012)	16,533.1	0
TR2	0.099 (0.001)	0.006 (0.001)	16,533.1	10,000
TR3	0.105 (0.001)	0.023 (0.004)	6.2	16
TR4	0.106 (0.002)	0.039 (0.006)	6.2	14
SVR	0.100 (0.001)	0.007 (0.001)	45.2	3,171

uniform sampling scheme. Here as the number of steps k increase, the OGA estimator better fits the training data, but an overly large k does not help to improve predictive accuracy and leads to a high computational cost. In this example, the proposed method picks $k = 9$, which leads to high predictive accuracy with a low 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 observed 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.

For picking a $k' > 0$, TR1 requires that

$$\|\hat{f}_k - \boldsymbol{y}\|_n^2 + \kappa\frac{k\log(n)}{n} \geq \|\hat{f}_{k'} - \boldsymbol{y}\|_n^2 + \kappa\frac{k'\log(n)}{n} \geq \kappa\frac{k'\log(n)}{n} \tag{3.2}$$

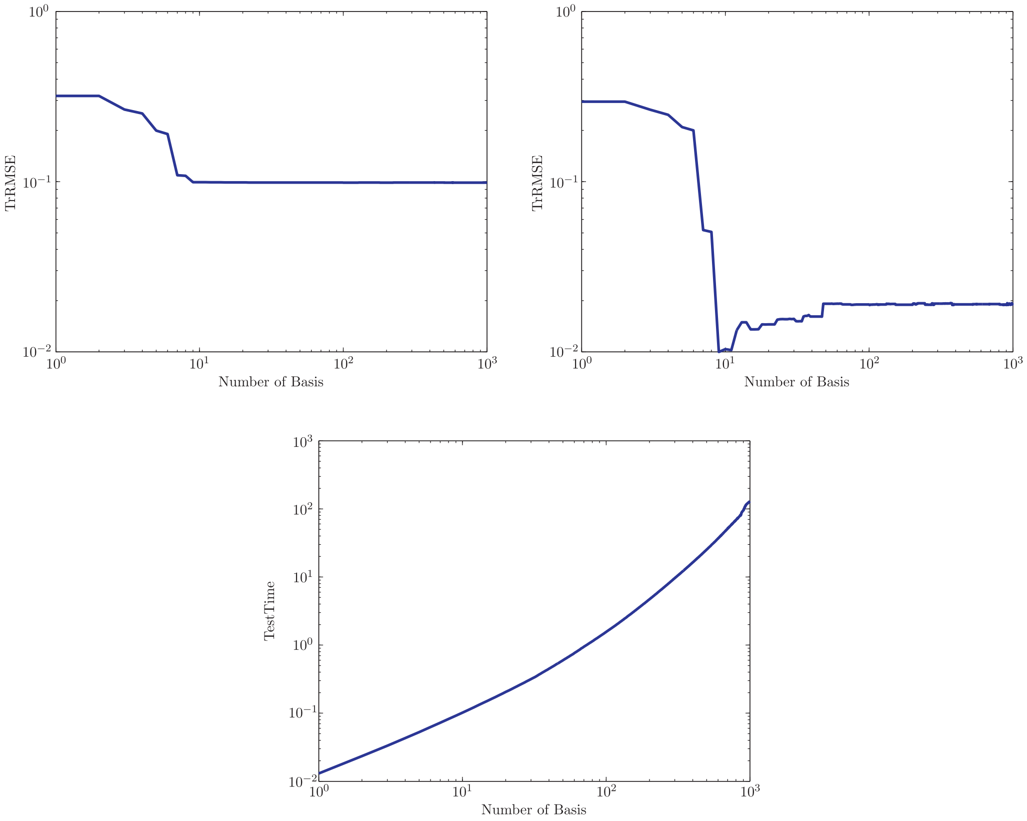

Figure 1. An illustration example for the uniform sampling scheme with n =1,000.

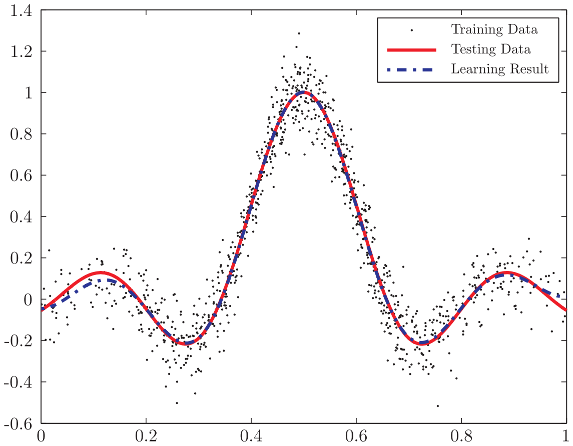

Figure 2. Plot of the PTR-based OGA for the illustration example.

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

for any $k < k'$. As $\|\hat{f}_k - \boldsymbol{y}\|_n^2 \leq 4M^2$, (3.2) implies

$$(k' - k) \leq \frac{4nM^2}{\kappa \log(n)}. \tag{3.3}$$

Taking $\kappa = 2568M^4(a+1)$ into (3.3), we have a necessary condition for $k' > 0$ to be

$$k' \leq \frac{n}{642M^2(a+5)\log n}. \tag{3.4}$$

However, even in the simple case where $k' = 3$, $M = 1$, and $a = 1$, (3.4) 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 (1.4) is satisfied, but our simulation examples suggest that (1.4) is difficult to meet in applications. To gain some insight, suppose that $Y = f^*(X) = \theta\varphi(X)$, where θ is a model parameter and $\varphi \in D_z^*$ is the basis function selected by OGA with $k = 1$. In this situation, the right hand side of (1.4) bounded by $n\theta^2$, while the left hand side of (1.4) is $n|\theta|$ for any $k \geq 1$. Thus, when $\theta \in (-1, 1)$, condition (1.4) can be never satisfied and TR2 necessarily leads to $k = n$. This simple example indicates that TR2 can lead to long OGA updating for huge n cases and thus is not suitable for massive data analysis.

3.3. Data examples

We now assess the proposed method on seven datasets from a variety of disciplines. Table 3 contains their sample sizes and dimensions. Refer to `http://archive.ics.uci.edu/ml/datasets.html` for background and detailed information. For each dataset, we denote by $y_{(i)}$ the ith order statistics of the response $\boldsymbol{y} = \{y_i, i \in \mathbb{N}_n\}$ and normalize the observations by

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

Table 4. Results for analyzing the real-world datasets.

	PTR				**SVR**			
Data sets	Tr-RMSE	Ts-RMSE	TIME	k	Tr-RMSE	Ts-RMSE	TIME	k
Boston h.	$0.073_{(0.003)}$	$0.081_{(0.011)}$	0.5	24	$0.070_{(0.002)}$	$0.079_{(0.015)}$	0.3	62
ENB2012	$0.067_{(0.002)}$	$0.069_{(0.003)}$	0.6	16	$0.067_{(0.001)}$	$0.069_{(0.002)}$	0.6	56
Bank	$0.043_{(0.001)}$	$0.043_{(0.001)}$	1.8	38	$0.045_{(0.001)}$	$0.045_{(0.001)}$	7.3	94
Delta ail.	$0.038_{(0.001)}$	$0.039_{(0.001)}$	3.3	23	$0.041_{(0.001)}$	$0.042_{(0.001)}$	9.4	147
Delta ele.	$0.054_{(0.001)}$	$0.053_{(0.001)}$	5.6	52	$0.054_{(0.001)}$	$0.054_{(0.002)}$	87.2	489
House8L	$0.062_{(0.001)}$	$0.064_{(0.002)}$	32.4	104	$0.072_{(0.001)}$	$0.074_{(0.002)}$	232.4	1408
CASP	$0.049_{(0.001)}$	$0.052_{(0.002)}$	188.7	168	$0.053_{(0.001)}$	$0.056_{(0.002)}$	536.5	1276

The dataset was 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 $\lfloor . \rfloor$ denotes the integer rounding operator. We applied the PTR-based OGA to the training set with the same setting used in simulations and assessed its performance based on the testing set. For the learning purpose, the Gaussian kernel was used, where σ was 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. 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, both PTR and SVR perform reasonably well in terms of predictive power. This is seen in their low Tr-RMSE values in all cases. Although SVR tends to be efficient when n is small, it is computationally costly for large n cases. In comparison, the PTR-based OGA provides accurate predictions with models of high sparsity, where promising numerical advantages are also observed.

4. Concluding Remarks

We have 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 data examples.

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

Supplementary Materials

The proofs of Proposition 1 and Theorems 1−2 are in the online supplementary material.

Acknowledgement

The authors are grateful to the Editor, the Associate Editor, and two referees for their insightful comments and constructive suggestions that have led to a substantially improved article. This work was supported by National Institute on Drug Abuse (NIDA) grants P50-DA10075, P50-DA039838, National Science Foundation (NSF) grant DMS-1512422, and National Natural Science Foundation of China (NSFC) grant 11131006. The content is solely the responsibility of the authors and does not necessarily represent the official views of NIDA, NSF, or NSFC.

References

Barron, A., Cohen, A. Dahmen, W., and DeVore, R. (2008). Approximation and learning by greedy algorithm. *Ann. Statist.* **36**, 64-94.

Cai, T. and Wang, L. (2011). Orthogonal matching pursuit for spare signal recovery with noise. *IEEE Trans. Inform. Theory* **57**, 4680-4688.

Chen, H., Li, L. and Pan, Z. (2013a). Learning rates of multi-kernel regression by orthogonal greedy algorithm. *J. Statist. Plann. Inference* **143**, 276-282.

Chen, H., Zhou, Y., Tang, Y., Li, L. and Pan, Z. (2013b). Convergence rate of the semi-supervised greedy algorithm. *Neural Networks* **44**, 44-50.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learn.* **20**, 273.

De Vito, E., Caponnetto, A. and Rosasco, L. (2005). Model selection for regularized least-squares algorithm in learning theory. *Found. Comput. Math.* **5**, 59-85.

Devore, R. and Temlyakov, V. (1996). Some remarks on greedy algorithms. *Adv. Comput. Math.* **5**, 173-187.

Elad, M. (2010). *Sparse and Redundant Representations.* Springer, New York.

Friedman, J. H. and Stuetzle, W. (1981). Projection pursuit regressions. *J. Amer. Statist. Soc.* **76**, 817-823.

Girosi, F., Jones, M. and Poggio, T. (1995). Regularization theory and neural networks architectures. *Neural Computation* **7**, 219-269.

Györfy, L., Kohler, M., Krzyzak, A. and Walk, H. (2002). *A Distribution-Free Theory of Nonparametric Regression.* Springer, Berlin.

Ing, C. and Lai, T. (2011). A stepwise regression method and consistent model selection for high-dimentional sparse linear models. *Statist. Sinica* **21**, 1473-1513.

Li, R., Lin, D. and Li, B. (2013). Statistical inference in massive data sets. *Appl. Stoch. Models in Business and Industry* **28**, 339-409.

Maiorov, V. and Ratsaby, J (1999). On the degree of approximation by manifolds of finite pseudo-dimension. *Constr. Approx.* **15**, 291-300.

Mallat, S. and Zhang, Z. (1993). Matching pursuits with time-frequency dictionaries. *IEEE Trans. Signal Process.* **41**, 3397-3415.

Mendelson, S. and Vershinin, R. (2003). Entropy and the combinatorial dimension. *Invent. Math.* **125** , 37-55.

Micchelli, C., Xu, Y. and Zhang, H. (2006). Universal kernels. *J. Machine Learn. Res.* **7**, 2651-2667.

National Research Council (2013). *Frontiers in Massive Data Analysis.* The National Academies Press, Washington, D.C.

Pati, Y. C., Rezaiifar, R. and Krishnaprasad, P. S. (1993). Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. *Proc. 27th Annu. Asilomar Conf. Signals, Systems and Computers*, 40-44. Pacific Grove, CA.

Rajaraman, A., Leskovec, J. and Ullman, J. (2011). *Mining of Massive Datasets.* 2nd edition. Cambridge University Press, Cambridge.

Shi, L., Feng, Y. and Zhou, D. (2011). Concentration estimates for learning with $L-1$-regularizer and data dependent hypothesis spaces. *Appl. Comput. Harmon. Anal.* **31**, 286-302.

Steinwart, I. and Christmann, A. (2008). *Support Vector Machines.* Springer, New York.

Sturm, B. and Christensen, M. (2012). Comparison of orthogonal matching pursuit implementations. *The 20th European Signal Processing Conference Proceedings*, 220-224.

Temlyakov, V. (2003). Nonlinear methods of approximation. *Found. Comp. Math.* **3**, 33-107.

Tropp, J. (2004). Greed is good: Algorithmic results for spares approximation. *IEEE Trans. Inform. Theory* **50**, 2231-2242.

Wang, H. (2009). Forward regression for ultrahigh dimentional variable screening. *J. Amer. Statist. Assoc.* **104**, 1512-1524.

Wu, Q. and Zhou, D. (2008). Learning with sample dependent hypothesis space. *Comput. Math. Appl.* **56**, 2896-2907.

Zhang, T. (2009). On the consistency of feature selection using greedy least squares regression. *J. Machine Learn. Res.* **10**, 555-568.

Zhang, T. (2011). Sparse recovery with orthogonal matching pursuit under RIP. *IEEE Trans. Inform. Theory* **57**, 6215-6221.

Zhou, D. and Jetter, K. (2006). Approximation with polynomial kernels and SVM classifiers. *Adv. Comput. Math.* **25**, 323-344.

Department of Mathematics and Statistics, University of Ottawa, Ottawa, ON, Canada K1N 6N5.

E-mail: cx3@uottawa.ca

School of Mathematics and Statistics, Xi'an Jiaotong Univeristy, Xi'an, Shaanxi, China 710049.

E-mail: ssxvkihc.lsb@stu.xjtu.edu.cn

School of Mathematics and Statistics, Xi'an Jiaotong Univeristy, Xi'an, Shaanxi, China 710049.

E-mail: jianfang@stu.xjtu.edu.cn

The Methodology Center, Department of Statistics, The Pennsylvania State University, University Park, PA 16802, USA.

E-mail: rli@stat.psu.edu

(Received October 2014; accepted July 2015)