Stochastic Feature Selection with Annealing and Its Applications to Streaming Data

Lizhe Sun and Adrian Barbu*

Abstract

Feature selection is an important topic in high-dimensional statistics and machine learning, for prediction and understanding the underlying phenomena. It has many applications in computer vision, natural language processing, bioinformatics, etc. However, most feature selection methods in the literature have been proposed for offline learning, and the existing online feature selection methods have theoretical and practical limitations in true support recovery. This paper proposes two novel online feature selection methods by stochastic gradient descent with a hard thresholding operator. The proposed methods can simultaneously select the relevant features and build linear regression or classification models based on the selected variables. The theoretical justification is provided for the consistency of the proposed methods. Numerical experiments on simulated and real sparse datasets show that the proposed methods compare favorably with state-of-the-art online methods from the literature.

Keywords: Variable Selection, Streaming Data, Stochastic Algorithm, Annealing Procedure, Big Data Learning

*To whom correspondence should be addressed: Adrian Barbu. Barbu is Professor, Department of Statistics, Florida State University, Tallahassee, FL 32306. Email: abarbu@fsu.edu. Sun is Assistant Professor, School of Statistics, Shanxi University of Finance and Economics, Taiyuan, Shanxi, 030006.

1 Introduction

Feature selection is an important research topic in high-dimensional statistics and machine learning. By removing the irrelevant and redundant features, feature selection methods improve the prediction accuracy, enhance the model interpretability, and reduce the computational burden. Therefore, variable selection methods have many applications in real data analysis, such as computer vision, text mining, and bioinformatics.

Most existing feature selection methods in the literature are restricted to the offline learning setting. In this offline scenario, all the features and observations are collected in advance for analysis. The ℓ_1 based method (Tibshirani, 1996), $\ell_1 + \ell_2$ based method (Zou and Hastie, 2005), non-convex penalized methods (Fan and Li, 2001; Zhang, 2010) and ℓ_0 based methods (Barbu et al., 2017; She et al., 2023) are the classical regularized methods proposed for variable selection in the offline learning. However, in real-world applications, the offline methods may not work when addressing large-scale streaming data, where the observations arrive sequentially with the time t. Even considering the data storage problem, a large dataset may not fit in the computer memory for training. An example of streaming data is the 120-day URL data for malicious website detection (Ma et al., 2009). In this dataset, the training instances are collected daily. There are about two million observations from 120 days and more than three million features. The other one is the Avazu click-through dataset (Juan et al., 2016), containing more than 12 million observations collected from 10 days of advertising logs, and 1 million features. In these cases, conventional offline variable selection methods are computationally expensive and memory-demanding. Therefore, using conventional offline feature selection techniques for these datasets is difficult.

In the online learning scenario, some online feature selection methods are proposed to exploit the sparse structure of the coefficient vector. Two main frameworks based on convex optimization are proposed in the literature. One is the Forward-Backward-Splitting method (Duchi and Singer, 2009), constructing an online feature selection framework by using the online proximal gradient (OPG) method (Duchi et al., 2010). The other one is Xiao's Regularized Dual Averaging (RDA) framework (Xiao, 2010), extending the primal-dual sub-gradient method (Nesterov, 2009) to the online case. A variant of the RDA method is developed by Agarwal et al. (2012). Then, an online statistical inference method is proposed based on the RDA framework (Han et al., 2024). These methods apply convex-relaxation approaches to use the ℓ_1-norm as a sparsity-inducing penalty. Greedy-based online feature selection methods such as truncated online gradient descent (TOGD) (Langford et al., 2009; Fan et al., 2018), first/second-order online feature selection methods (FOFS/SOFS) (Wang et al., 2014; Wu et al., 2017) are also developed in the literature.

Another research direction is the streaming feature selection method (Wu et al., 2010; Yang et al., 2016). In this scenario, one feature arrives once while all the training examples are available before the learning process starts. The goal is to select a subset of important features and then build an appropriate model on them. Unlike conventional online learning, in this novel online scenario, we cannot select all true features and train a model for prediction until all features are disclosed. This paper assumes observations arrive sequentially with time. Therefore, we will not consider algorithms such as Wu et al. (2010) and Yang et al. (2016) for comparison.

However, the existing online feature selection methods have some limitations in true support recovery. Although these proposed methods can induce sparse solutions and improve the model interpretability by constructing confidence intervals (Han et al., 2024) for coefficients, it is hard to recover the support of true features even under mild assumptions on the data matrix $\mathbf{X}$ because the sparsity level is difficult to control using penalized methods such as the ℓ_1 penalty in the streaming data case. At each step of the iteration, since the stochastic gradient has a dramatic change, a fixed tuning parameter λ may not be able to address the variable selection problem. Additionally, compared to the offline gradient descent, the stochastic gradient descent (SGD) method has a lower convergence rate $\mathcal{O}(1/\sqrt{T})$ (Shalev-Shwartz and Ben-David, 2014), which may lead to problems with true support recovery. To solve these issues, novel online feature selection methods are proposed in this paper for large-scale or high-dimensional datasets such as the URL dataset and the Avazu click-through dataset. The main contributions of this paper are:

1. Two novel methods are proposed for online feature selection. Compared to the existing online variable selection methods, the proposed methods can simultaneously recover the support of the true features for datasets with strongly correlated features and learn a linear regression or classification model on the selected features;

2. A theoretical analysis of the coefficient consistency and true feature recovery is provided for the proposed stochastic gradient descent with truncation (SGDT) methods, under some standard assumptions;

3. The empirical performance of the proposed methods is verified by conducting numerical experiments on simulated and real data. These experiments reveal that the proposed methods have a higher true support recovery and prediction accuracy than many existing online feature selection methods.

The remaining part of the paper is organized as follows. Section 2 introduces the notation and setup. Section 3 proposes the novel online feature selection methods. Section 4 provides the theoretical guarantees for the proposed methods. Section 5 evaluates the performance of the proposed methods by numerical experiments and real data analysis. Section 6 presents a brief conclusion of this paper and a discussion on future work.

2 Setup and Notation

Suppose that a sequence of independent training instances $\mathbf{z}_i = (\mathbf{x}_i, y_i)$, $i = 1, 2, \cdots$, where $\mathbf{x}_i \in \mathbb{R}^p$ and $y_i \in \mathbb{R}$, are generated from a bounded random input-output pair $\mathbf{Z} = (\mathbf{X}, Y)$. In the online setting, these training instances arrive one at a time. The expectation of the loss function is defined by

$$\mathcal{L}(f) = \mathbb{E}\ell(f(\mathbf{X}), Y), \tag{1}$$

where f is the link from the input $\mathbf{X}$ to the output Y, and $\ell : \mathbb{R}^p \to \mathbb{R}$ is the loss function. The loss function $\mathcal{L}$ is called the population loss. Since the dimension p may be large in practical applications, this paper considers the sparse learning problem for linear models. Hence, the population loss (1) is rewritten to the parametric form

$$\mathcal{L}(\boldsymbol{\beta}) = \mathbb{E}\ell(\mathbf{X}\boldsymbol{\beta}, Y), \tag{2}$$

where $\boldsymbol{\beta}$ is the coefficient vector. To estimate the coefficient $\boldsymbol{\beta}$ with a sparsity level k, we may solve the following constrained optimization problem based on the empirical version of the loss (2) by

$$\operatorname*{argmin}_{\boldsymbol{\beta}\in\mathbb{R}^p} L(\boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^{n} \ell(\mathbf{x}_i^T\boldsymbol{\beta}, y_i),\ \ \|\boldsymbol{\beta}\|_0 \le k.$$

In the online scenario, the coefficient vector $\boldsymbol{\beta}$ is estimated sequentially, one example at a time. Using a sequence of observations $\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_{t-1}$, we can learn the coefficient vector $\boldsymbol{\beta}_t$. The empirical loss function of the linear regression is

$$L(\boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \mathbf{x}_i^T\boldsymbol{\beta})^2, \tag{3}$$

and the empirical loss function of the logistic regression model for classification is

$$L(\boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^{n}\log\{1 + \exp(-y_i\mathbf{x}_i^T\boldsymbol{\beta})\}, \tag{4}$$

where each label $y_i \in \{-1, +1\}$ is binary. In the following, we use $\ell_t(\boldsymbol{\beta})$ as the simplified notation for the loss function $\ell(\mathbf{x}_t^T\boldsymbol{\beta}, y_t)$, where $t = 1, 2, \cdots$.

Then, we establish the notation formally. Vectors are lowercase bold letters, such as $\mathbf{x} \in \mathbb{R}^d$, and scalars are lowercase letters, e.g. $x \in \mathbb{R}$. A sequence of vectors is denoted by subscripts, i.e. $\mathbf{w}_1, \mathbf{w}_2, \cdots$, and the entries in a vector are denoted by non-bold subscripts, like w_j. Matrices are upper case bold letters, such as $\mathbf{M} \in \mathbb{R}^{d\times d}$, and random variables are upper case letters, such as Z. Given a vector $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \cdots, \gamma_n)^T \in \mathbb{R}^n$, we define vector norms: $\|\boldsymbol{\gamma}\|_1 = \sum_{i=1}^n |\gamma_i|$, $\|\boldsymbol{\gamma}\| = \sqrt{\sum_{i=1}^n \gamma_i^2}$ and $\|\boldsymbol{\gamma}\|_0 = \#\{j : \gamma_j \neq 0\}$. Finally, $\nabla_{\mathbf{x}} f(\mathbf{x})$ is the gradient vector of $f(\mathbf{x})$ with respect to $\mathbf{x}$.

3 Methodology

Two novel online feature selection methods are proposed in this section. One is the Stochastic Feature Selection with Annealing (SFSA), and the other is a simple version without an annealing procedure, called Stochastic Gradient Descent with Truncation (SGDT).

3.1 Stochastic Feature Selection with Annealing

The new method proposed here is motivated by the proposed feature selection with annealing (FSA) algorithm (Barbu et al., 2017; She et al., 2023). Like the FSA for offline learning, the proposed SFSA can simultaneously solve the feature selection problem and learn the linear regression or classification models for prediction.

According to the description in Barbu et al. (2017), the key points in this variable selection algorithm are: (1) using an annealing procedure to lessen the greediness in reducing the dimensionality from p to k, (2) gradually removing the most irrelevant variables to facilitate computation.

The proposed method starts with an initialized coefficient vector $\boldsymbol{\beta}_1$, generally $\boldsymbol{\beta}_1 = \mathbf{0}$, and then alternates two basic steps: one step updating the parameters based on the observation $\mathbf{z}_t = (\mathbf{x}_t, y_t)$ by the stochastic gradient descent

$$\boldsymbol{\beta}_{t+1} = \boldsymbol{\beta}_t - \eta \frac{\partial \ell_t(\boldsymbol{\beta}_t)}{\partial \boldsymbol{\beta}},$$

where η is the learning rate. The other step is a feature selection step that removes some unimportant variables based on the ranking of feature importance measure $v_j = \sqrt{(\mathbf{s}_j - \bar{\mathbf{x}}_j^2)}|\beta_j|$, $j = 1, 2, \cdots, p$.

There are two main differences between the proposed stochastic feature selection with annealing and the offline counterpart from Barbu et al. (2017). First, in our approach, the gradient is approximated using one observation or a mini-batch of observations, while Barbu et al. (2017) uses all data to construct the gradient. More importantly, the data is normalized beforehand in Barbu et al. (2017). At the same time, in our approach, we work directly with the non-normalized data and then use the estimated standard error of the variables to measure their importance, as described in Remark 4.1. The algorithm is summarized in Algorithm 1.

In general, we will input a mini-batch of observations rather than one at a time, but the one-at-a-time situation can always be recovered by setting the mini-batch size to 1. Also, an annealing schedule M_t is used, representing the number of features kept at time t by

$$M_t = k + (p-k)\max\{0, \frac{T-t}{t\mu + T}\}, t = 1, 2, \cdots, T,$$

in which k is the desired sparsity level, μ is the annealing parameter in this model and T is the maturity time for this schedule, when exactly k features are selected.

A longer maturity time uses more observations, therefore it usually has better feature selection capabilities. In practice, as the number of true features k and the total number of observations are unknown, one can use multiple maturity times in parallel, as illustrated in Figure 1, for many values of k. Therefore, we will always have a "current" set of selected features of a desired sparsity while building a better one as more data becomes available.

Finally, we emphasize that Algorithm 1 is just based on simple SGD. We also combine

Algorithm 1 Stochastic Feature Selection with Annealing

```
Input: Training data $\mathbf{z}_t = (\mathbf{x}_t, y_t)$ arriving one at a time, learning rate $\eta$, sparsity level
$k$, annealing parameter $\mu$, maturity time $T$, and the loss function $\ell_t(\boldsymbol{\beta})$, $t = 1, 2, \cdots$.
Output: Trained coefficient vector $\boldsymbol{\beta}_t$ with $\|\boldsymbol{\beta}_t\|_0 \leq k$.
Initialize $\boldsymbol{\beta}_1 = \mathbf{0}$, sample mean vector $\bar{\mathbf{x}} = \mathbf{0}$, and sample variance vector $\mathbf{s} = \mathbf{0}$.
for $t = 1$ to $\infty$ do
    Receive an observation $\mathbf{z}_t$.
    Update $\bar{\mathbf{x}} \leftarrow \frac{t-1}{t}\bar{\mathbf{x}} + \frac{1}{t}\mathbf{x}_t,\ \mathbf{s} \leftarrow \frac{t-1}{t}\mathbf{s} + \frac{1}{t}\mathrm{diag}(\mathbf{x}_t\mathbf{x}_t^T)$.
    Update $\boldsymbol{\beta}_{t+1} \leftarrow \boldsymbol{\beta}_t - \eta\frac{\partial \ell_t(\boldsymbol{\beta}_t)}{\partial \boldsymbol{\beta}}$.
    if $t \leq T$ then
        Keep only the $M_t$ features with the highest values of
            $$v_j = \sqrt{(\mathbf{s}_j - \bar{\mathbf{x}}_j^2)}|\beta_j|,\ j = 1, 2, \cdots, p,$$
        and renumber them $1, ..., M_t$.
    end if
end for
```

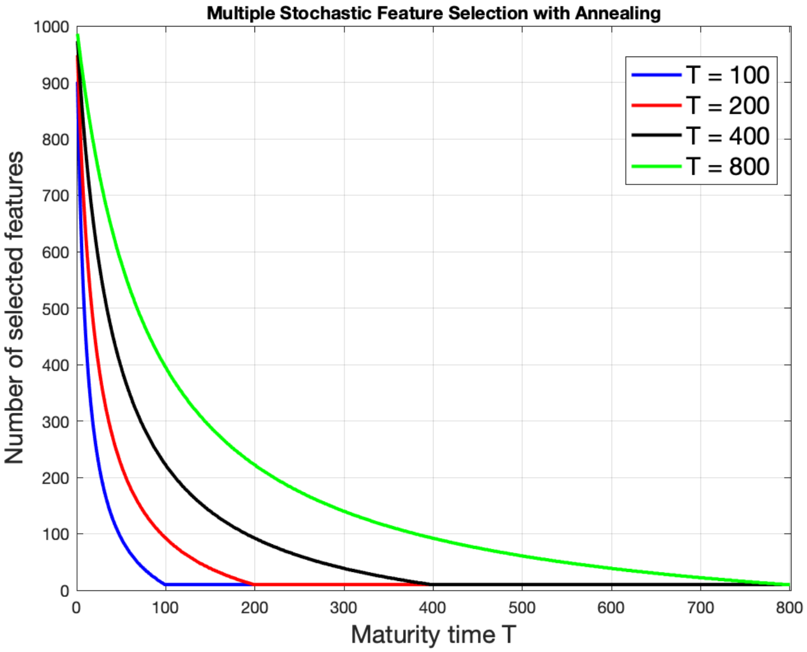

Figure 1: Annealing schedules with multiple maturity times T for SFSA with $k = 10$.

the SFSA algorithm with momentum, Nesterov accelerated gradient (Sutskever et al., 2013) or Adam (Kingma and Ba, 2015) optimization methods. These techniques will be evaluated in the experimental section 5.

3.2 Stochastic Gradient Descent with Truncation

From the SFSA method, we can see that using online gradient descent, one can select features by keeping the k coefficients with the largest importance measure $v_j = \sqrt{(\mathbf{s}_j - \bar{\mathbf{x}}_j^2)}|\beta_j|$. We consider a special case for online feature selection where we do not use an annealing procedure with online gradient descent but select the coefficients with k largest importances at time T. The prototype algorithm is described in Algorithm 2.

In the literature, some similar TOGD methods were proposed in Langford et al. (2009), Fan et al. (2018), and Wang et al. (2014), based on different loss functions. However, these methods truncate the coefficients vector at each time t, which may mislead the algorithms in selecting irrelevant features, especially when the features have strong correlations. The

Algorithm 2 Stochastic Gradient Descent with Truncation (SGDT)

```
Input: Training data $\mathbf{z}_t = (\mathbf{x}_t, y_t)$ arriving one at a time, learning rate $\eta$, sparsity level
$k$, maturity time $T$, and the loss function $\ell_t(\boldsymbol{\beta})$, $t = 1, 2, \cdots$.
Output: Trained coefficient vector $\boldsymbol{\beta}_t$ with $\|\boldsymbol{\beta}_t\|_0 \le k$.
Initialize $\boldsymbol{\beta}_1 = 0$, sample mean $\bar{\mathbf{x}} = 0$, and $\mathbf{s} = 0$.
for $t = 1$ to $\infty$ do
    Receive an observation $\mathbf{z}_t$.
    Update $\bar{\mathbf{x}} \leftarrow \frac{t-1}{t}\bar{\mathbf{x}} + \frac{1}{t}\mathbf{x}_t, \mathbf{s} \leftarrow \frac{t-1}{t}\mathbf{s} + \frac{1}{t}\text{diag}(\mathbf{x}_t\mathbf{x}_t^T)$
    Update $\boldsymbol{\beta}_{t+1} \leftarrow \boldsymbol{\beta}_t - \eta\frac{\partial \ell(\boldsymbol{\beta}_t)}{\partial \boldsymbol{\beta}_t}$
    if $t \ge T + 1$ then
        Keep only the $k$ features with highest values of
            $$v_j = \sqrt{(\mathbf{s}_j - \bar{\mathbf{x}}_j^2)}|\beta_j|,\ j = 1, 2, \cdots, p.$$
    end if
end for
```

proposed Algorithm 2 can select the subset of features according to the largest values of the variable importance $v_j = \sqrt{(\mathbf{s}_j - \bar{\mathbf{x}}_j^2)}|\beta_j|$ after T time steps, which will improve the accuracy for feature selection.

4 Theoretical Analysis

This section provides the theoretical justification for the SGDT method. When the iteration time T is large enough, and the true signal values for $\boldsymbol{\beta}^*$ are strong enough, the index of largest k^* values of $|\boldsymbol{\beta}_T|$ is the true support of the $\boldsymbol{\beta}^*$. All the features are assumed to be normalized in the dataset. Therefore, the absolute value of the coefficient vector $\boldsymbol{\beta}$ replaces the variable importance $\mathbf{v}$ in the feature selection step. As presented in Algorithm 1 and

2, the updated procedure by SGD method is

$$\boldsymbol{\beta}_{t+1} = \boldsymbol{\beta}_t - \eta \frac{\partial \ell_t(\boldsymbol{\beta}_t)}{\partial \boldsymbol{\beta}}.$$

Let the $\boldsymbol{\beta}^*$ be the minimizer for the population loss (2), where $\text{supp}(\boldsymbol{\beta}^*) = S_{\boldsymbol{\beta}^*}$ and $\|\boldsymbol{\beta}^*\|_0 = k^*$. It is not hard to verify that $\boldsymbol{\beta}^*$ is the true coefficient vector for the linear regression model or the Logistic regression model. Before providing the main theorems, we present some assumptions as follows.

> **Assumption 1** (Constrained Conditions for Coefficients Vector). There exist constants R_1 and R_2 satisfied $\|\boldsymbol{\beta}^*\| \leq R_1$ and $\|\boldsymbol{\beta}\| \leq R_2$. We define the closed convex set $\mathcal{W} := \{\boldsymbol{\beta} : \|\boldsymbol{\beta}\| \leq R_2\}$. For any $\boldsymbol{\beta} \in \mathcal{W}$, it is not hard to verify that $\|\boldsymbol{\beta} - \boldsymbol{\beta}^*\|$ is bounded by the constant $R = R_1 + R_2$.

> **Assumption 2** (Strongly Convexity for Population Loss Function). For any $\boldsymbol{\beta} \in \mathcal{W}$, where $\mathcal{W}$ is a closed convex set, the expectation of the loss function $\mathcal{L}(\boldsymbol{\beta})$ is strongly convex. The definition is that there is existing a constant $\lambda > 0$ satisfied that $\mathcal{L}(\boldsymbol{\beta}) - (\lambda/2)\|\boldsymbol{\beta}\|^2$ is a convex function.

> **Assumption 3** (Bounded Gradients for the Loss Function). Given a bounded random input-output pair $\mathbf{Z} = (\mathbf{X}, Y)$, for any $\boldsymbol{\beta} \in \mathcal{W}$, the gradient of the loss function $\ell(\mathbf{X}\boldsymbol{\beta}, Y)$ is bounded by a constant G such as $\|\nabla_\beta \ell(X\boldsymbol{\beta}, Y)\| \leq G$.

Since the random variables $\mathbf{X}$ and Y are bounded, Assumption 3 holds for typical loss functions, e.g., in the linear regression (3) and the logistic regression (4).

Proposition 4.1. *Suppose that the assumptions 1-3 hold. Let $\boldsymbol{\beta}^*$ be the minimizer for the loss function $\mathcal{L}(\boldsymbol{\beta})$ and $\boldsymbol{\beta}^*$ be the k^*-sparse vector, thus $\|\boldsymbol{\beta}^*\|_0 = k^*$. Let $\boldsymbol{\beta}_t$ be the SGD coefficient vector at iteration t, and $\ell(\boldsymbol{\beta}_t)$ be a differentiable convex function on a closed convex set $\mathcal{W}$. Then for the learning rate $\eta > 0$, we have*

$$\|\boldsymbol{\beta}_{t+1} - \boldsymbol{\beta}^*\|^2 \leq (1 - 2\lambda\eta)\|\boldsymbol{\beta}_t - \boldsymbol{\beta}^*\|^2 + \eta^2 G^2 + 2\eta\langle \boldsymbol{\beta}^* - \boldsymbol{\beta}_t, \nabla \ell(\boldsymbol{\beta}_t) - \nabla \mathcal{L}(\boldsymbol{\beta}_t)\rangle.$$

Proof. The updated procedure for the gradient descent method is

$$\boldsymbol{\beta}_{t+1} = \boldsymbol{\beta}_t - \eta \nabla \ell_t(\boldsymbol{\beta}_t).$$

Then we have

$$\begin{aligned}
\|\boldsymbol{\beta}_{t+1} - \boldsymbol{\beta}^*\|^2 &= \|\boldsymbol{\beta}_t - \eta \nabla \ell(\boldsymbol{\beta}_t) - \boldsymbol{\beta}^*\|^2 \\
&\leq \|\boldsymbol{\beta}_t - \boldsymbol{\beta}^*\|^2 + \eta^2 \|\nabla \ell(\boldsymbol{\beta}_t)\|^2 - 2\eta \langle \boldsymbol{\beta}_t - \boldsymbol{\beta}^*, \nabla \ell(\boldsymbol{\beta}_t) \rangle \\
&\leq \|\boldsymbol{\beta}_t - \boldsymbol{\beta}^*\|^2 + \eta^2 \|\nabla \ell(\boldsymbol{\beta}_t)\|^2 - 2\eta \langle \boldsymbol{\beta}_t - \boldsymbol{\beta}^*, \nabla \mathcal{L}(\boldsymbol{\beta}_t) \rangle + 2\eta \langle \boldsymbol{\beta}_t - \boldsymbol{\beta}^*, \nabla \mathcal{L}(\boldsymbol{\beta}_t) - \nabla \ell_t(\boldsymbol{\beta}_t) \rangle \\
&\leq \|\boldsymbol{\beta}_t - \boldsymbol{\beta}^*\|^2 + \eta^2 G^2 - 2\eta \langle \boldsymbol{\beta}_t - \boldsymbol{\beta}^*, \nabla \mathcal{L}(\boldsymbol{\beta}_t) - \nabla \mathcal{L}(\boldsymbol{\beta}^*) \rangle \\
&\quad + 2\eta \langle \boldsymbol{\beta}_t - \boldsymbol{\beta}^*, \nabla \mathcal{L}(\boldsymbol{\beta}_t) - \nabla \ell_t(\boldsymbol{\beta}_t) \rangle.
\end{aligned}$$

Since $\nabla \mathcal{L}(\boldsymbol{\beta}^*) = 0$ and the $\mathcal{L}(\boldsymbol{\beta})$ is strongly convex function, then we have

$$\langle \boldsymbol{\beta}_t - \boldsymbol{\beta}^*, \nabla \mathcal{L}(\boldsymbol{\beta}_t) - \nabla \mathcal{L}(\boldsymbol{\beta}^*) \rangle \geq \lambda \|\boldsymbol{\beta}_t - \boldsymbol{\beta}^*\|^2.$$

As a result, we have

$$\|\boldsymbol{\beta}_{t+1} - \boldsymbol{\beta}^*\|^2 \leq (1 - 2\lambda\eta) \|\boldsymbol{\beta}_t - \boldsymbol{\beta}^*\|^2 + \eta^2 G^2 + 2\eta \langle \boldsymbol{\beta}^* - \boldsymbol{\beta}_t, \nabla \ell_t(\boldsymbol{\beta}_t) - \nabla \mathcal{L}(\boldsymbol{\beta}_t) \rangle.$$

□

Denote the error term by $\epsilon(\boldsymbol{\beta}_t) = \nabla \ell_t(\boldsymbol{\beta}_t) - \nabla \mathcal{L}(\boldsymbol{\beta}_t)$, $t = 1, 2, \cdots, T$. According the the assumption 3, the the error term $\epsilon(\boldsymbol{\beta}_t)$ is a martingale difference sequence bounded by $\|\epsilon(\boldsymbol{\beta}_t)\| \leq 2G$. Then, Theorem 4.1 is described in the following.

Theorem 4.1. *Let $\boldsymbol{\beta}_1 = \mathbf{0}$ be the initial values for $t = 1$. With the same notations as Proposition 4.1 and suppose that assumptions 1 to 3 hold, the SFSA coefficient vector $\boldsymbol{\beta}_{T+1}$ satisfies*

$$\|\boldsymbol{\beta}_{T+1} - \boldsymbol{\beta}^*\|^2 \leq (1 - 2\lambda/T^\alpha)^T \|\boldsymbol{\beta}^*\|^2 + \frac{G^2}{2\lambda T^\alpha} + \frac{2}{T^\alpha} \sum_{t=1}^{T} \langle \boldsymbol{\beta}^* - \boldsymbol{\beta}_t, \epsilon(\boldsymbol{\beta}_t) \rangle, \tag{5}$$

if the learning rate $\eta = 1/T^\alpha$ is fixed, where $1/2 < \alpha < 1$. And the convergence rate for $\mathbb{E}\|\boldsymbol{\beta}_{T+1} - \boldsymbol{\beta}^*\|^2$ is $\mathcal{O}(1/T^\alpha)$.

Proof. By using the conclusion of the proposition 4.1 T times, let η be a fixed learning rate, then the following inequality holds by

$$\begin{aligned}
& \|\boldsymbol{\beta}_{T+1} - \boldsymbol{\beta}^*\|^2 \\
\leq \ & (1-2\lambda\eta)^T\|\boldsymbol{\beta}^*\|^2 + \sum_{t=0}^{T-1}(1-2\lambda\eta)^t\eta^2G^2 + 2\eta\sum_{t=0}^{T-1}(1-2\lambda\eta)^t\langle\boldsymbol{\beta}^* - \boldsymbol{\beta}_t, \nabla\ell(\boldsymbol{\beta}_t) - \nabla\mathcal{L}(\boldsymbol{\beta}_t)\rangle \\
\leq \ & (1-2\lambda\eta)^T\|\boldsymbol{\beta}^*\|^2 + \sum_{t=0}^{+\infty}(1-2\lambda\eta)^t\eta^2G^2 + 2\eta\sum_{t=0}^{T-1}(1-2\lambda\eta)^t\langle\boldsymbol{\beta}^* - \boldsymbol{\beta}_{T-t}, \epsilon(\boldsymbol{\beta}_{T-t})\rangle \\
\leq \ & (1-2\lambda\eta)^T\|\boldsymbol{\beta}^*\|^2 + \frac{\eta G^2}{2\lambda} + 2\eta\sum_{t=1}^{T}\langle\boldsymbol{\beta}^* - \boldsymbol{\beta}_t, \epsilon(\boldsymbol{\beta}_t)\rangle.
\end{aligned}$$

Let $\eta = 1/T^\alpha$ and $1/2 < \alpha < 1$, then we have

$$\|\boldsymbol{\beta}_{T+1} - \boldsymbol{\beta}^*\|^2 \leq (1-2\lambda/T^\alpha)^T\|\boldsymbol{\beta}^*\|^2 + \frac{G^2}{2\lambda T^\alpha} + \frac{2}{T^\alpha}\sum_{t=1}^{T}\langle\boldsymbol{\beta}^* - \boldsymbol{\beta}_t, \epsilon(\boldsymbol{\beta}_t)\rangle.$$

Taking the expectation on both sides, the following inequality holds by

$$\mathbb{E}\|\boldsymbol{\beta}_{T+1} - \boldsymbol{\beta}^*\|^2 \leq (1-2\lambda/T^\alpha)^T\|\boldsymbol{\beta}^*\|^2 + \frac{G^2}{2\lambda T^\alpha}.$$

When $T \to +\infty$, the first term converges with the convergence rate $\mathcal{O}(1/\exp(T^{1-\alpha}))$, and the second term converges with the rate $\mathcal{O}(1/T^\alpha)$. Therefore, the conclusion holds. □

Remark 4.1. *The above proposition and theorem assume that the data is normalized. If that is not the case, we can maintain the running averages* $\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i \in \mathbb{R}^p$ *and* $\mathbf{s} = \mathrm{diag}\{\frac{1}{n}\sum_{i=1}^n \mathbf{x}_i\mathbf{x}_i^T\} \in \mathbb{R}^p$ *by updating them one example at a time. Then, we can use the standard error:* $\hat{\boldsymbol{\sigma}}_{\mathbf{x}} = \mathbf{s}_{\mathbf{x}} - \hat{\boldsymbol{\mu}}_{\mathbf{x}}^2 \in \mathbb{R}^p$ *to perform SFSA or SGDT on non-normalized data using the thresholding* $\Theta_k(\hat{\boldsymbol{\sigma}}_{\mathbf{x}}\boldsymbol{\beta})$ *instead of* $\Theta_k(\boldsymbol{\beta})$*.*

In the conclusion of Theorem 4.1, the first term $(1-2\lambda/T^\alpha)^T\|\boldsymbol{\beta}^*\|^2$ on the right side of (5) will vanish when $T \to +\infty$ because of $\lim_{T\to+\infty}(1-2\lambda/T^\alpha)^T = \exp\{-T^{1-\alpha}\}$. Here, T is the iteration times for the stochastic gradient algorithms. Since in these theorems, we assume that the training instances arrive one at a time, T also represents the total sample size. For the second term on the right side, since we assume the learning rate is $\eta = 1/T^\alpha$, the term $G^2/2\lambda T^\alpha$ also can vanish. Finally, the variable selection problem depends on the random error term. We will give the assumptions and conclusion for variable selection in the following Corollary 4.1.

Corollary 4.1. *With the same notations as Proposition 4.1 and Theorem 4.1 and suppose that the assumptions 1 to 3 hold, given a constant $\omega > 0$ and the iterative time T, when the fixed learning rate is $\eta = 1/T^\alpha$, where $1/2 < \alpha < 1$ if the minimum absolute value of true $\boldsymbol{\beta}^*$ satisfies that*

$$|\beta^*_{\min}| > (1 - 2\lambda/T^\alpha)^T \|\boldsymbol{\beta}^*\|^2 + \frac{G^2}{2\lambda T^\alpha} + 4GR\frac{\sqrt{2\omega}}{T^{\alpha-1/2}},$$

then, with at least a probability $1 - \exp(-\omega)$, the index of all the true variables can be selected.

Proof. Since we have

$$\mathbb{E}[\nabla \ell(\boldsymbol{\beta}_t)] = \nabla \boldsymbol{\mathcal{L}}(\boldsymbol{\beta}_t),$$

the $\langle \boldsymbol{\beta}^* - \boldsymbol{\beta}_t, \epsilon(\boldsymbol{\beta}_t)\rangle$, $t = 1, 2, \cdots, T$, is a martingale difference sequence. This martingale difference can be bounded by

$$|\langle \boldsymbol{\beta}^* - \boldsymbol{\beta}_t, \epsilon(\boldsymbol{\beta}_t)\rangle| \leq \|\boldsymbol{\beta}^* - \boldsymbol{\beta}_t\| \|\epsilon(\boldsymbol{\beta}_t)\| \leq 2GR.$$

According to the Azuma inequality, with the probability $1-\exp(-\omega)$, we have the following inequality:

$$\sum_{t=1}^{T} \langle \boldsymbol{\beta}^* - \boldsymbol{\beta}_t, \epsilon(\boldsymbol{\beta}_t)\rangle \leq 2GR\sqrt{2T\omega}.$$

Therefore, with the probability $1 - \exp(-\omega)$, the following inequality holds by

$$\|\boldsymbol{\beta}_{T+1} - \boldsymbol{\beta}^*\|^2 \leq (1 - 2\lambda/T^\alpha)^T \|\boldsymbol{\beta}^*\|^2 + \frac{G^2}{2\lambda T^\alpha} + 4GR\frac{\sqrt{2\omega}}{T^{\alpha-1/2}}.$$

□

Compared to $\beta^*_{\min} = \mathcal{O}(\sqrt{\frac{\log p}{n}})$ in the true support recovery theorem for the offline high-dimensional setting (Loh and Wainwright, 2017), we assume the number of predictors p is fixed, even though p may be very large, and the iterative time $T \to +\infty$. Since only one observation or a mini-batch of observations is used at one time, the iterative time $T \to +\infty$ means that the sample size $n \to +\infty$. Our theoretical justification shows that with a high probability, if the iterative time T is large enough, while the signal strength $\beta^*_{\min}$ can be arbitrarily small, we can select the support of the true variables by using the index of largest k^* values of $|\boldsymbol{\beta}_{T+1}|$, introduced by the $(T+1)$-th iteration of the SGD algorithm. In this case, the sparsity level k^* does not change the variable selection accuracy.

5 Experiments and Real Data Analysis

Numerical experiments and real data analysis are presented to evaluate the performance of the proposed methods in this section. First, the experimental results on large sparse simulated datasets are presented to compare the performance of the proposed methods with the other state-of-the-art methods in linear regression and classification cases for prediction and variable selection. Then, the results on large real datasets are presented, and the performance of various online feature selection methods is compared. All experiments are run on a desktop computer with Core i7 - 8700k CPU and 32Gb memory.

5.1 Experiments with Simulated Data

In this experiment, we use uniformly correlated data generated as follows: given a scalar α, we generate $z_i \sim \mathcal{N}(0,1)$, then we generate an observation $\mathbf{x}_i$:

$$\mathbf{x}_i = \alpha z_i \mathbf{1}_{p\times 1} + \mathbf{u}_i, \text{ with } \mathbf{u}_i \sim \mathcal{N}(0, \mathbf{I}_p).$$

We generate i.i.d. observations by this way to obtain the $n \times p$ data matrix $\mathbf{X} = [\mathbf{x}_1^T, \cdots, \mathbf{x}_n^T]^T$, where $\mathbf{x}_i \in \mathbb{R}^p$ for $i = 1, 2, \cdots, n$. It is easy to verify that the correlation between any pair of predictors is $\alpha^2/(1+\alpha^2)$. We set $\alpha = 1$ in our simulation thus the correlation between any two features is 0.5. Then, the dependent response $\mathbf{y}$ is generated from the following linear regression model:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta}^* + \boldsymbol{\eta}, \text{ with } \boldsymbol{\eta} \sim \mathcal{N}(0, \mathbf{I}_n), \tag{6}$$

and for classification:

$$\mathbf{y} = \text{sign}(\mathbf{X}\boldsymbol{\beta}^* + \boldsymbol{\eta}), \text{ with } \boldsymbol{\eta} \sim \mathcal{N}(0, \mathbf{I}_n), \tag{7}$$

where $\boldsymbol{\beta}^*$ is a p-dimensional sparse parameter vector. The true coefficients are $\beta_j^* = 0$ except $\beta_{10j^*}^* \neq 0$, $j^* = 1, 2, \cdots, k^*$. A linear model cannot perfectly separate the classification data because of the random noise term. The simulation is based on the following setting: $p = 10000$ and $k^* = 100$. We considered the signal strength $\beta_{10j^*}^* = 1$ for strong signal and $\beta_{10j^*}^* \in [0.05, 1]$ increasing linearly with j from 0.05 to 1 for weak signal. The sample size n varies from 5×10^3 to 3×10^5, and the learning rates are $\eta = 0.0001$ for linear regression and $\eta = 0.01$ for classification. The annealing parameters are $\mu = 10$ in the regression case and $\mu = 5$ in the classification case for the proposed SFSA and related methods.

We evaluate two classical online feature selection methods for comparison for regression and classification: the OPG (Duchi and Singer, 2009) and RDA (Xiao, 2010) methods. In both frameworks, we consider ℓ_1 regularization for regression and $\ell_1 + \ell_2$ regularization for classification. In the regression setting, we also compare with the Regularization Annealed epoch Dual Averaging (RADAR) method proposed by Agarwal et al. (2012). For classification, besides the OPG and RDA methods, the simulation includes the first-order online feature selection (FOFS) and the second-order online feature selection (SOFS) methods (Wang et al., 2014; Wu et al., 2017).

The sparsity controlling parameters in the simulation are tuned to obtain k^* variables.

This can be done directly for SFSA, FOFS, and SOFS methods, and indirectly through the regularization parameter for the OPG, RDA, and RADAR methods. In OPG and RDA, we used 200 values of λ on an exponential grid and chose the λ that induces the $\hat{k}$ non-zero features, where $\hat{k}$ is the largest number of non-zeros features smaller than or equal to k^*, the number of true features. Since RADAR only induces approximately sparse coefficients, we set the small values to zero to select variables. All the experiments are replicated 20 times in the simulations.

The following criteria are used in the numerical experiments: the true variable detection rate (DR), the root mean square error (RMSE) on the test data for regression, the area under the ROC curve (AUC) on the test data for classification, and the running time (Time) of the various algorithms. The variable detection rate DR is defined as the average number of true variables correctly detected by an algorithm divided by the number of true variables. So when $S_{\boldsymbol{\beta}}$ is the set of selected variables and $S_{\boldsymbol{\beta}^*}$ are the true variables, then we have

$$DR = \frac{E(|S_{\boldsymbol{\beta}} \cap S_{\boldsymbol{\beta}^*}|)}{|S_{\boldsymbol{\beta}^*}|}.$$

5.1.1 Experimental results for regression

We evaluate the empirical performance of SFSA for the regression task. The performance of various algorithms is presented in Table 1. Considering the detection rate (DR), the SFSA, SFSA-AG, and SGDT algorithms are much better than the OPG, RDA, and RADAR methods. When the sample size n increases, our proposed methods find all true features and nothing else, 100% of the time. Also, the proposed algorithms have less computational time because they can directly control the desired sparsity level. In contrast, the OPG and RDA methods cannot recover the support of the true features. Because of the need to vary the regularization parameter to control the sparsity level, these algorithms are computationally expensive. The RADAR method performs better than the OPG and RDA

Table 1: Simulation experiments for online regression averaged over 20 runs.

	Variable Detection Rate DR (%)						RMSE						Time(s)					
n	SFSA	SFSA-AG	SGDT	OPG	RDA	RADAR	SFSA	SFSA-AG	SGDT	OPG	RDA	RADAR	SFSA	SFSA-AG	SGDT	OPG	RDA	RADAR
$p = 10000, k = 100$, strong signal $\beta = 1$, learning rate = 0.0001, mini-batch = 25																		
5×10^3	56.75	43.80	**98.05**	1.30	0.80	3.75	**60.07**	62.61	97.14	100.0	100.1	87.55	0.048	0.052	0.021	165.1	464.1	321.6
10^4	84.30	73.10	**100**	1.20	1.25	6.45	**48.74**	50.90	96.05	100.9	100.9	88.92	0.096	0.103	0.041	331.8	917.2	647.0
2×10^4	**100**	**100**	**100**	1.30	1.10	11.45	**39.93**	41.54	91.80	100.3	100.4	90.92	0.164	0.179	0.082	495.7	1822	1253
$p = 10000, k = 100$, weak signal β increase from 0.05 to 1, learning rate = 0.0001, mini-batch = 25																		
10^4	67.95	59.80	**84.55**	1.05	0.95	5.05	**25.68**	26.81	50.44	53.06	53.07	51.61	0.095	0.102	0.041	339.8	375.4	680.1
3×10^4	91.35	90.60	**93.40**	1.05	1.40	23.75	**16.37**	16.95	46.54	52.99	53.00	49.75	0.259	0.275	0.122	837.8	909.9	2109
10^5	**98.30**	98.20	98.25	0.95	1.30	71.60	**7.89**	8.10	34.84	52.44	52.46	32.24	0.738	0.804	0.409	2013	2117	6913
3×10^5	**100**	**100**	**100**	0.85	1.35	-	**2.06**	2.08	2.47	52.83	52.86	-	2.187	2.404	1.093	4853	4167	-

Table 2: Comparison between SFSA, SFSA-AG (AG), SFSA-Adam (Adam), SGDT, and other online algorithms for classification averaged 20 runs.

	Variable Detection Rate DR(%)								AUC								Time(s)							
n	SFSA	AG	Adam	SGDT	FOFS	SOFS	OPG	RDA	SFSA	AG	Adam	SGDT	FOFS	SOFS	OPG	RDA	SFSA	AG	Adam	SGDT	FOFS	SOFS	OPG	RDA
$p = 10000, k = 100$, strong signal $\beta = 1$, learning rate = 0.01, mini-batch = 25																								
3×10^3	66.60	62.40	54.55	**88.55**	8.70	1.25	1.15	1.70	0.841	0.822	0.781	**0.934**	0.578	0.507	0.502	0.506	0.037	0.040	0.044	0.014	0.268	0.005	103.5	150.0
10^4	86.35	82.60	74.35	**100**	15.75	0.6	1.90	3.95	0.945	0.931	0.901	**0.992**	0.654	0.499	0.504	0.513	0.124	0.132	0.148	0.046	0.876	0.016	342.1	451.3
3×10^4	**100**	**100**	**100**	**100**	35.20	1.0	1.75	11.0	**0.996**	**0.996**	**0.996**	**0.996**	0.756	0.497	0.500	0.560	0.293	0.318	0.381	0.138	2.412	0.050	1029	1198
$p = 10000, k = 100$, weak signal β increase from 0.05 to 1, learning rate = 0.01, mini-batch = 25																								
3×10^3	48.35	45.95	40.60	**63.50**	9.05	1.25	1.05	1.50	0.842	0.825	0.780	**0.912**	0.610	0.505	0.503	0.505	0.037	0.040	0.045	0.014	0.272	0.005	101.8	161.3
10^4	69.25	67.25	62.10	**81.05**	16.40	0.7	1.30	4.65	0.944	0.933	0.912	**0.974**	0.713	0.500	0.499	0.522	0.123	0.131	0.149	0.046	0.855	0.017	340.3	474.7
3×10^4	92.60	**92.50**	92.20	91.85	33.90	1.0	1.60	12.90	**0.989**	**0.989**	**0.989**	**0.989**	0.835	0.496	0.498	0.594	0.296	0.319	0.381	0.138	2.294	0.049	1029	1246
10^5	**98.60**	**98.55**	98.05	96.95	60.65	0.9	1.70	27.85	**0.991**	**0.991**	**0.990**	**0.990**	0.947	0.506	0.506	0.762	0.925	1.012	1.228	0.460	6.258	0.164	3412	3858

methods but not as good as the proposed methods. Considering the test RMSE for the proposed methods, the RMSE values based on the test data are smaller when the sample size n is larger. However, for the existing OPG and RDA methods, the RMSE values for the test data do not converge when the sample size n is large.

5.1.2 Experimental results for classification

Similarly, the empirical performance of SFSA and its variants AG and Adam for classification is evaluated in this subsection. The experimental results are shown in Table 2. First, we analyze the variable detection rate. In binary classification, it is more challenging to select the true features than in regression. First, we observe that we need more training instances in classification than in regression to recover all true features. Then, for weaker signal strength, given the largest sample size $n = 3 \times 10^5$, our SFSA algorithms and the variants cannot select all true features, even though their performance is much better than the regularized based methods and standard online feature selection methods, FOFS and SOFS. Similar to the regression problem, the regularized-based methods OPG and RDA cannot recover the support of true features for the dataset with strongly correlated features. As for the latest algorithms in the literature, FOFS and SFOS (Wang et al., 2014; Wu et al., 2017) from the literature cannot detect the true features either.

Then we consider the test AUC (Area under the ROC curve for the test data) as a criterion. Because the proposed SFSA methods and SGDT can recover most of the true features, it is clear why the test AUC values for SFSA methods and SGDT are larger than the existing methods, i.e., the proposed methods perform better than the current methods on classification accuracy. In other words, since the existing methods may not select most of the true features, the models learned by the current methods cannot predict the test data as accurately as the proposed method.

The analysis of time complexity is the same as in regression. Because of the parameter tuning problem, regularized-based methods are computationally expensive. By contrast, greedy-based methods that directly control the sparsity level have a huge advantage in computational time.

5.2 Simulations on Large Sparse Datasets

To run simulations of the size of real large-scale datasets and to further verify the classification performance of the current methods in the literature, such as FOFS, SOFS, OPG, and RDA algorithms, we also performed numerical experiments on large, sparse simulated

Table 3: Comparison between SFSA, SFSA-AG, SGDT, and other online methods for the simulated sparse classification dataset.

					Variable Detection Rate DR (%)							AUC						
Dataset	n	p	k	non-zero	SFSA	SFSA-AG	SGDT	FOFS	SOFS	OPG	RDA	SFSA	SFSA-AG	SGDT	FOFS	SOFS	OPG	RDA
$\mathbf{X}_1$	10^5	10^4	100	200	**100**	**100**	**100**	99.95	97	70.45	83.30	**0.923**	**0.923**	0.918	0.917	0.768	0.712	0.814
$\mathbf{X}_2$	10^5	2×10^4	200	400	**100**	**100**	**100**	99.70	97.15	67.75	86.88	**0.919**	**0.919**	0.913	0.910	0.758	0.671	0.818
$\mathbf{X}_3$	10^5	10^5	500	500	**100**	**100**	**100**	99.87	98.47	71.74	-	**0.918**	**0.918**	0.917	0.899	0.749	0.642	-

data similar to the ones described in Wu et al. (2017).

First, we generate three large sparse datasets $\mathbf{X}_1$, $\mathbf{X}_2$, and $\mathbf{X}_3$, where each observation is a sparse vector with 200, 400, and 500 nonzero entries at random locations. Each nonzero entry is generated from the i.i.d Gaussian distribution $\mathcal{N}(0,1)$. The label $\mathbf{y} \in \{-1,+1\}$ is generated from the following noiseless linear model:

$$\mathbf{y} = \text{sign}(\mathbf{X}\boldsymbol{\beta}^*),$$

where the true parameter vector $\boldsymbol{\beta}^*$ is sparse with 100, 200, and 500 nonzero entries, respectively. The nonzero entries are sampled from the uniform distribution $\mathcal{U}(0,1)$. These three datasets are large-scale but less challenging since all the predictors are independent and the classes are separable.

In this large sparse data simulation, we evaluate the algorithms regarding the true variable detection rate (DR) and the area under the ROC curve (AUC) on test data. Based on the previous experiment, we removed the SFSA-Adam algorithm from the large sparse data simulation because its performance is similar to SFSA-AG. The simulation results are shown in Table 3.

These experimental results verify that all the proposed methods can detect the true features quite well if the features are independent. However, compared to the proposed greedy-based algorithms, the regularized-based methods such as OPG and RDA still suffer lower variable detection rates and prediction accuracy. Considering the proposed greedy-based methods, the proposed SFSA-based methods outperform the SGDT method in true feature recovery and prediction. The experimental results for the large sparse datasets prove our proposed SFSA methods can address various datasets and that our implementation of the competing algorithms performs similarly to that reported in Wu et al. (2017).

5.3 Real Data Experiments

We use the proposed SFSA and SGDT methods for high-dimensional and large-scale real datasets. The FOFS and SOFS methods are also considered for comparison. The OPG, RDA, and RADAR methods are not included in these real datasets analyses since the numerical results in Table 3 show that the OPG, RDA, and RADAR methods are time-consuming when addressing large-scale datasets. Additionally, they may not perform very well in the classification. The first dataset is the URL dataset (Ma et al., 2009), also analyzed using the SOFS method in Wu et al. (2017). The URL dataset is a large-scale and high-dimensional dataset. This dataset has more than 3 million features and 2 million observations. The observations are collected to predict whether a website is malicious or not based on a large number of features. Since the URL instances are obtained day-by-day from a large Web server, it makes sense to apply online learning methods to this dataset. Following Ma et al. (2009), we used the streaming data from day 0 to day 99 as the training data and used the data on day 100 to evaluate the performance of the models we trained. The AUC values are reported for the data on day 100. There are 16,000 variables selected in this dataset, about 0.5% of the total features. The Avazu click-through dataset (Juan et al., 2016) is a large-scale dataset as well. It has 1 million features and more than 12 million observations. We selected 1,000 variables in this dataset, about 0.1% of the total features. There are 10 million observations used for training and more than 2 million observations used for testing the model.

Then, the proposed methods are applied to two ultra-high dimensional datasets. The

Table 4: Comparison between SFSA, SFSA-AG, SGDT, FOFS, and SOFS for the real datasets. The AUC values for test data are presented for these methods.

Dataset	n	p	SFSA	SFSA-AG	SGDT	FOFS	SOFS
URL	2,396,130	3,231,961	**0.998**	**0.998**	0.995	0.995	0.986
Avazu	1,000,000	12,642,186	**0.630**	0.602	0.627	0.623	0.623
News	19,996	1,355,191	**0.883**	0.882	0.881	0.875	0.736
Rcv1	20,242	47,236	0.972	0.972	**0.973**	0.938	0.782
Gisette	6,000	5,000	**0.983**	**0.983**	0.971	0.961	0.636

first one is the News dataset (Keerthi and DeCoste, 2005), which has 19,996 observations and 1,355,191 features. The second one is the Rcv1 dataset (Lewis et al., 2004), which has 20,242 observations and 47,236 features. In this paper, we used 18,000 observations to train the models and the rest to evaluate the model. There are 1% of total variables selected in these datasets.

The last dataset is the Gisette dataset (Guyon et al., 2004). It is not a very large dataset or not a high-dimensional dataset, but the number of features p and the number of samples n are very close. This is a good dataset to test whether the proposed methods not only can handle the high-dimensional data but can address the conventional settings in which n and p are very close or n is larger than p.

The results for the real data analysis are shown in Table 4. The evaluation criterion is the AUC values for the test data. A larger AUC value means a better classification performance. We can observe that the models trained by the proposed methods perform better than those learned by the existing methods. On all five datasets, the proposed SFSA method is better than the conventional methods from the literature. The performance of the proposed SGDT is also great, being the best method for the Rcv1 dataset and outperforming the existing techniques on four out of five datasets.

6 Conclusion

In this paper, we propose two online feature selection methods: stochastic feature selection with annealing (SFSA) and stochastic gradient descent with truncation (SGDT). Compared to the existing online feature selection methods, the proposed methods can recover the support of the true features for the high-dimensional and large-scale datasets, even when there is a strong correlation between features. Moreover, they can select features and estimate the parameters simultaneously. The theoretical justification provides the convergence and consistency of the estimated coefficients and theoretical guarantees of true feature support recovery. Then, the empirical performance of the proposed methods is evaluated and compared with other state-of-the-art online methods. Based on the results of experiments and real data analysis, the proposed methods have excellent performance on real and simulated datasets.

However, further study is necessary for the proposed stochastic feature selection with the annealing method. First, we do not consider the model drift problem, a common issue in streaming data learning. Second, the stochastic feature selection with annealing can also be used for offline learning. One can subsample the data and approximate the gradient using a mini-batch, then select the M_t features by the higher $|\beta_j|$ at each epoch t. The related theoretical analysis needs to be investigated. Finally, the proposed stochastic feature selection method can be used in a sparse deep neural network model. Generally, neural networks are trained by stochastic gradient descent (SGD) and its variants. Thus, it would be interesting to study if SFSA can replace SGD for training a neural network model, especially for non-vision data with irrelevant features.

Declarations

Conflict of interest

The authors declare that they have no conflict of interest.

Funding

Adrian Barbu's research was partially supported by DARPA ARO grant W911NG-16-1-0579. Lizhe Sun's research was partially supported by the National Social Science Fund of China grant 24&ZD183.

Data availability

The real datasets used in this paper are publicly available as specified in Section 5.3.

Code availability

The code will be made publicly available on `https://github.com/barbua/` upon paper acceptance.

Word Count

The total word count is 4825.

Authors' contributions

Sun and Barbu contributed to the conception and design of the study. Sun proved the theoretical results. Material preparation, data collection, and analysis were performed by Sun. The first draft of the manuscript was written by Sun and was revised by Sun and Barbu. All authors have read and approved the final manuscript.

References

Agarwal, A., Negahban, S. N., and Wainwright, M. J. (2012), "Stochastic optimization and sparse statistical recovery: optimal algorithms for high dimensions," in *NIPS*, p. 1538 – 1546.

Barbu, A., She, Y., Ding, L., and Gramajo, G. (2017), "Feature selection with annealing for computer vision and big data learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39, 272–286.

Duchi, J. and Singer, Y. (2009), "Efficient online and batch learning using forward backward splitting," *Journal of Machine Learning Research*, 10, 2899–2934.

Duchi, J. C., Shalev-Shwartz, S., Singer, Y., and Tewari, A. (2010), "Composite objective mirror descent," in *COLT*, pp. 14–26.

Fan, J., Gong, W., Li, C. J., and Sun, Q. (2018), "Statistical sparse online regression: a diffusion approximation perspective," in *AISTATS*, pp. 1017–1026.

Fan, J. and Li, R. (2001), "Variable selection via nonconcave penalized likelihood and its oracle properties," *Journal of the American Statistical Association*, 96, 1348–1360.

Guyon, I. M., Gunn, S. R., Ben-Hur, A., and Dror, G. (2004), "Result analysis of the NIPS 2003 feature selection challenge," in *NIPS*.

Han, R., Luo, L., Lin, Y., and Huang, J. (2024), "Online inference with debiased stochastic gradient descent," *Biometrika*, 111, 93–108.

Juan, Y., Zhuang, Y., Chin, W.-S., and Lin, C.-J. (2016), "Field-aware factorization machines for CTR prediction," in *Proceedings of the 10th ACM Conference on Recommender Systems*, ACM, pp. 43–50.

Keerthi, S. S. and DeCoste, D. (2005), "A modified finite Newton method for fast solution of large scale linear SVMs," *Journal of Machine Learning Research*, 6, 341–361.

Kingma, D. P. and Ba, J. (2015), "Adam: a method for stochastic optimization," in *ICLR*, vol. 5.

Langford, J., Li, L., and Zhang, T. (2009), "Sparse online learning via truncated gradient," *Journal of Machine Learning Research*, 10, 777–801.

Lewis, D. D., Yang, Y., Rose, T. G., and Li, F. (2004), "RCV1: A new benchmark collection for text categorization research," *Journal of Machine Learning Research*, 5, 361–397.

Loh, P.-L. and Wainwright, M. J. (2017), "Support recovery without incoherence: A case for nonconvex regularization," *The Annals of Statistics*, 45, 2455 – 2482.

Ma, J., Saul, L. K., Savage, S., and Voelker, G. M. (2009), "Identifying suspicious URLs: an application of large-scale online learning," in *ICML*, pp. 681–688.

Nesterov, Y. (2009), "Primal-dual subgradient methods for convex problems," *Mathematical Programming*, 120, 221–259.

Shalev-Shwartz, S. and Ben-David, S. (2014), *Understanding Machine Learning: From Theory to Algorithms*, Cambridge University Press.

She, Y., Shen, J., and Barbu, A. (2023), "Slow kill for big data learning," *IEEE Transactions on Information Theory*, 69, 5936–5955.

Sutskever, I., Martens, J., Dahl, G., and Hinton, G. (2013), "On the importance of initialization and momentum in deep learning," in *ICML*, pp. 1139–1147.

Tibshirani, R. (1996), "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, 267–288.

Wang, J., Zhao, P., Hoi, S. C., and Jin, R. (2014), "Online feature selection and its applications," *IEEE Transactions on Knowledge and Data Engineering*, 26, 698–710.

Wu, X., Yu, K., Wang, H., and Ding, W. (2010), "Online streaming feature selection," in *ICML*, pp. 1159–1166.

Wu, Y., Hoi, S. C., Mei, T., and Yu, N. (2017), "Large-scale online feature selection for ultra-high dimensional sparse data," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 11, 48.

Xiao, L. (2010), "Dual averaging methods for regularized stochastic learning and online optimization," *Journal of Machine Learning Research*, 11, 2543–2596.

Yang, H., Fujimaki, R., Kusumura, Y., and Liu, J. (2016), "Online feature selection: A limited-memory substitution algorithm and its asynchronous parallel variation," in *SIGKDD*, ACM, pp. 1945–1954.

Zhang, C.-H. (2010), "Nearly unbiased variable selection under minimax concave penalty," *Annals of Statistics*, 894–942.

Zou, H. and Hastie, T. (2005), "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67, 301–320.