Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2017) 25: 163 – 171

doi:10.3906/elk-1501-98

# An online approach for feature selection for classification in big data

**Nasrin Banu NAZAR, Radha SENTHILKUMAR***
Department of Information Technology, Madras Institute of Technology, Chromepet, Chennai, Tamil Nadu, India

**Received:** 14.01.2015 • **Accepted/Published Online:** 06.12.2015 • **Final Version:** 24.01.2017

**Abstract:** Feature selection (FS), also known as attribute selection, is a process of selection of a subset of relevant features used in model construction. This process or method improves the classification accuracy by removing irrelevant and noisy features. FS is implemented using either batch learning or online learning. Currently, the FS methods are executed in batch learning. Nevertheless, these techniques take longer execution time and require larger storage space to process the entire dataset. Due to the lack of scalability, the batch learning process cannot be used for large data. In the present study, a scalable efficient Online Feature Selection (OFS) approach using the Sparse Gradient (SGr) technique was proposed to select the features from the dataset online. In this approach, the feature weights are proportionally decremented based on the threshold value, which results in attaining zeros for the insignificant features' weights. In order to demonstrate the efficiency of this approach, an extensive set of experiments was conducted using 13 real-world datasets that range from small to large size. The results of the experiments showed an improved classification accuracy of 15%, which is considered to be significant when compared with the existing methods.

**Key words:** Data analysis, data preprocessing, big-data analytics, feature selection, online learning

## 1. Introduction

Feature selection (FS) is an important data processing step used extensively in data analytics and machine learning problems. The objective of FS is to remove redundant and irrelevant features from the original dataset. By removing these, the complexity in the computation is subdued and generalization capability of the model can be improved. Many of the existing FS methods use the batch learning model. These models assume that the entire dataset is available for processing. However, in real-world scenarios, the user may not have sufficient storage capacity to keep track of the entire dataset, and also collecting and maintaining such data is cumbersome and expensive. Moreover, the batch mode FS methods perform many epochs to acquire the best feature subset and this leads to an increase in computational time. Thus, it is seen that the batch feature selection methods may not be well suited for big-data problems.

On the other hand, online feature selection (OFS) is a challenging problem of processing data online, i.e. processing one instance at a time. In this paper, a supervised binary classification problem is considered for demonstrating the sparse gradient (SGr) method to select features online. In this approach, for each data instance, a perceptron learning model is trained to learn feature weights. The SGr method is applied to feature weights and weights of irrelevant features are proportionally reduced to zero based on a threshold value. By following this gradual decrement step, SGr induces sparsity in the feature weights. Finally, the features that are gaining nonzero weights are considered as important features and can be used in classification in big-data

*Correspondence: radhasenthil@annauniv.edu

problems to improve the classification performance. The rest of the paper is organized as follows: Section 2 presents a brief analysis on related works in the literature. Section 3 introduces the problem and discusses the proposed approach. In Section 4, we experimentally demonstrate the efficiency of the proposed approach, and Section 5 lists the possible extensions of the paper.

## 2. Related work

Based on dataset labeling, FS methods can be classified as supervised [1], unsupervised [2], or semisupervised. Supervised FS methods are further categorized into filter, wrapper, and embedded methods. Filter methods [3,4] utilize metrics like distance, mutual information, and consistency measures to rank features and select top-ranked features, but filter-based methods do not consider the classification performance and require an entire dataset to process. Wrapper methods [5] initially generate a candidate feature subset and exploit the learning model's performance to evaluate the generated candidate feature subset, but the generation of a candidate feature subset for evaluation may lead to an exhaustive search in the feature space. On the other hand, in embedded models, FS is done within a learning model. In other words, the results of a learning model are employed to evaluate and select features. The proposed OFS using the SGr method belongs to the embedded FS category.

In online learning, one of the classical and widely used models is the perceptron model. Based on perceptrons many margin-based classification methods are proposed. For example, the confidence weighted learning algorithm [6] uses second-order information about the data to update feature weights. Most of the online learning algorithms require that all features of data instances be available for processing, while the number of instances can vary. The execution time for these algorithms is less when compared to batch learning methods.

In the literature, streaming and online FS methods are gaining importance in recent days. Most of the streaming feature selection methods [7] assume that the number of training instances is fixed, while the number of features can vary. For example, grafting [8] and alpha-investing [9] belong to this category. These methods assume that features are arriving sequentially. However, in real-world scenarios, instances arrive sequentially while the number of features remains constant. In this paper, the proposed method addresses a feature selection problem where instances arrive sequentially.

The problem of selecting features online is addressed by the OFS algorithm [10,11]. It uses a simple truncation method to eliminate the features that have small weight, but the presence of noisy data may lead to the removal of important features. Moreover, these methods require the number of features to be selected as a parameter, but in many real-world situations the user may not know a sufficient number of features to avail good accuracy. In this paper, the proposed method is designed to select the features automatically without demanding any strict limitations on the number of features to be selected. The main aim of FS methods is to select a subset of features without degradation in the performance (classification performance). However, without sufficient background knowledge about the problem and dataset, the user could not decide about the number of features that do not degrade the performance. Because of this uncertainty about the number of features to select, it is always superior to make this selection of number of features an automated process. In this way, our method is different from the existing OFS [10,11] works.

Unlike the feature selection method for data streams [12], which deals with streaming data, our proposed methods assume that the data are available already, but because of storage and computation time constraints the data are accessed online. Moreover, our proposed methods address the classification problem where [12]

proposed the FS method to monitor data. Since the classification problem has been taken for demonstration, our proposed methods approach the problem in a supervised manner.

Our method is related to the truncated gradient method [13]. The truncated gradient descent method is used to induce sparsity in the feature weights, but this technique does not address the feature selection problem. In this paper, we present an approach that selects features from the dataset online by inducing sparsity in the feature weights and compare its efficiency with other online feature selection methods.

## 3. Online feature selection using sparse gradient

### 3.1. Problem setting

In this paper, FS for an online binary classification paradigm is proposed. Let $D$ be the dataset from which features are to be selected. $D$ consists of $M$ instances $(x_m, y_m)$ where $0 < m \leq M$, $x_m \in R^N$ where $N$ is the number of features. $y_m$ is the label for the instance $x_m$ and $y_m \in \{-1, 1\}$. For each instance $x_m$, feature weight vector $w_m \in R^N$ is learned and used to classify the instance. Later, depending on the result of classification, $w_m$ is updated to $w_{m+1}$, but the problem is to select only $s$ features, instead of all the $N$ features, where $s < N$, such that the accuracy obtained while using $s$ features is equal to or better than using all the $N$ features. This particular selection is done online so this problem is called OFS.

### 3.2. OFS method

OFS methods accept the dataset one instance at a time. For each instance, a linear classifier, or in other words a weight vector, is learned and the function $sign(w_m^{'} \times x_m)$ is used to predict the class label of the instance. Later, the target class and predicted class are compared. When the method misclassifies, the weight vector is updated using the following stochastic gradient rule:

$$w_{m+1} = w_m - \alpha C'\left(\langle w_m, x_m \rangle, y_m\right). \tag{1}$$

In Eq. (1), $C'(*,*)$ is the gradient cost function and $\alpha$ is the learning rate. After this update, the weight vector is imposed to a L2 ball so that the norm of the weight vector is controlled. This procedure is denoted as an online learning algorithm (OLA). Later, the SGr method is applied to the feature weights to induce sparsity.

The existing OFS algorithm uses a truncation method instead of the SGr method to select features. In the existing method, for each data instance, the truncation method is applied to the feature weights after calculating them. This truncation method keeps only the top $Q$ number of features that gained large weights. All other $N - Q$ features are reduced to zero. This method is not robust to noisy data.

### 3.3. Sparse gradient method (SGr)

SGr is a gradient method that induces sparsity in the weight vector of features. SGr is adaptive in nature such that it can be called for each instance or once for every K instances. In the SGr method, a threshold value of reduction ($\vartheta$) and a reduction amount ($\sigma$) should be predefined, by which the weight of features is reduced. SGr reduces the feature weights that are within the $\vartheta$ range by $\sigma$ amount. SGr uses the logic that if a feature does not gain enough weight to cross the threshold value $\vartheta$ for multiple instances, its value will be gradually decreased to zero. If a feature weight goes across zero then, that feature weight will remain zero. Thus, the

SGr method is less aggressive. The SGr function is given below:

$$SGr\left(W,\vartheta,\sigma\right)=\left\{\begin{array}{cc}\max(0,W-\sigma) & \text{if}\,0<W<\vartheta\\ \min(0,W+\sigma) & \text{if}\,0>W>-\vartheta\\ W & otherwise\end{array}\right\}. \tag{2}$$

In SGr, $\vartheta$ and $\sigma$ play a major role in selecting features. If the $\sigma$ value is too large then even the weights of important features may reach zero, leading to their omission. At the same time, if the $\sigma$ value is very small, then the weights of unimportant features may not reach zero, leading to their inclusion in the classifier and thus reducing the generalization capability of the classifier. Also, if the $\vartheta$ value is large, then many features will fall into the reduction range, leading to their exclusion, whereas if $\vartheta$ is very small, then the number of selected features will be large and lead to the inclusion of unimportant features. This is because if $\vartheta$ is small, only a few features will get their weights reduced to zero, so values of $\sigma$ and $\vartheta$ are to be chosen empirically depending upon the problem and the dataset to be processed. The pseudocode for the OFS (with SGr) and SGr is given in Methods 1 and 2, respectively.

**Method 1:** OFS with sparse gradient (OFSSGD)

**Input:** $\boldsymbol{W,\vartheta,\sigma,D}$

**for** each instance in $\boldsymbol{D}$

  $\boldsymbol{W = OLA(D)}$

  $\boldsymbol{W = SGr(W,\vartheta,\sigma)}$

**end for**

**Method 2**: Sparse gradient (SGr)

**Input:** $\boldsymbol{W,\vartheta,\sigma}$

**for** each *weight* $\boldsymbol{w_i}$ *in* $W$ **do**

  **if** $\boldsymbol{w_i > 0}$ **and** $\boldsymbol{w_i < \vartheta}$ **then**

    $\boldsymbol{w_i = max\{w_i - \sigma, 0\}}$

  **else if** $\boldsymbol{w_i < 0}$ **and** $\boldsymbol{w_i > -\vartheta}$ **then**

    $\boldsymbol{w_i = min\{w_i + \sigma, 0\}}$

  **end if**

**end for**

This SGr can be used with any learning model that uses gradient descent to induce sparsity in the weight vector of features. The proposed SGr considers the features that have weights larger than $\vartheta$ as important features because their contribution for classification is high. During learning, a feature is eliminated if the

weight of the feature become zero because of the SGr method. Thus, after processing many data instances, the size of the feature set will be reduced. This characteristic leads to the fact that if the number of training instances is large then features can be selected with more accuracy.

## 4. Experimental results

In this section, to demonstrate the efficacy of the SGr method, an extensive set of experiments are conducted. Real-world benchmark datasets ranging from small to large in size from the UCI Machine Learning Repository (https://archive.ics.uci.edu/ml/datasets.html), given in Table 1, are used in the experiments. The SGr method that is applied for every instance is implemented in OFS using Sparse Gradient Descent (OFSSGD) method and its variant that uses the SGr method for every $K^{th}$ instance is implemented in OFSSGDK. In the experiments, K is set to 5. The results of these 2 methods are compared with the existing OFS algorithm, which was discussed in [11].

**Table 1.** List of datasets used

| Dataset | # OF INSTANCES | # of dimensions |
|---|---|---|
| Zoo | 100 | 16 |
| Heart-Cleveland | 295 | 46 |
| Vote | 434 | 16 |
| German | 1000 | 24 |
| Svmguide3 | 1243 | 21 |
| Splice | 3175 | 60 |
| Spambase | 4601 | 57 |
| Magic04 | 19,020 | 10 |
| A8a | 32,561 | 123 |
| Kddcup08 | 102,294 | 125 |
| Ijcnn1 | 191,681 | 22 |
| Cod-rna | 488,565 | 8 |
| Covtype | 581,012 | 54 |

### 4.1. Experimental setup

For a fair comparison, the regularization parameter and the learning rate are set to 0.01 and 0.2, respectively. The dataset is normalized before it is processed. The $\vartheta$ and $\sigma$ values are set to 15% of the mean of the dataset and 0.2, respectively. A vector of $\vartheta$ values is used instead of a single constant value to denote the fact that different features may be in different ranges, and to consider each feature's importance we need a $\vartheta$ value that is different for each feature. At the same time, it is worth mentioning that our proposed methods provide efficient results even with constant $\vartheta$ value. These parameter values are chosen after conducting the experiments with different possible values for these parameters. The experiments were executed 20 times with different order of input data to show the stability of the proposed approach. All the results were reported by taking the average of the results collected over these 20 runs.

### 4.2. Evaluation of predictive performance

Table 2 shows the mistakes rate with execution time and Figure 1 shows the accuracy obtained by the proposed methods on medium-sized datasets. As mentioned in Section 3, the misclassified instance is counted as a mistake and the overall accuracy of the dataset is calculated by the ratio of number of correctly predicted instances

to total number of instances. From the results, it is observed that the proposed OFSSGD and OFSSGDK outperform the results of the existing OFS algorithm. In the experiments, these methods selected 30%–50% of features from the original features set.

**Table 2.** Comparison between algorithms on medium-sized datasets.

| Dataset | Algorithm | Execution time | Number of mistakes | Number of features selected |
|---|---|---|---|---|
| German | OFS | 0.0294 | 449 ± 82.7 | 2 |
| | OFSSGD | 0.0075 | 341.6 ± 14.6 | 14.8 ± 1.64 |
| | OFSSGDK | 0.0062 | 336.55 ± 5.5 | 19.15 ± 2.05 |
| Svmguide3 | OFS | 0.0338 | 400.95 ± 66.80 | 2 |
| | OFSSGD | 0.0089 | 341.80 ± 11.11 | 11.35 ± 1.56 |
| | OFSSGDK | 0.0072 | 333.70 ± 8.87 | 13.9 ± 1.25 |
| Magic04 | OFS | 0.3973 | 6023.45 ± 1342.37 | 1 |
| | OFSSGD | 0.1245 | 5533.15 ± 1421.50 | 2.10 ± 0.96 |
| | OFSSGDK | 0.1000 | 4685.85 ± 674.39 | 3.50 ± 0.82 |
| splice | OFS | 0.0705 | 735.4 ± 68.31 | 6 |
| | OFSSGD | 0.0213 | 608.75 ± 45.41 | 54.75 ± 2.38 |
| | OFSSGDK | 0.0183 | 591.8 ± 31.8 | 57.10 ± 1.68 |
| A8a | OFS | 1.1543 | 9424.4 ± 2545.8 | 12 |
| | OFSSGD | 0.3268 | 8019.35 ± 92.50 | 44.55 ± 2.54 |
| | OFSSGDK | 0.2534 | 7969.80 ± 79.64 | 57.45 ± 3.88 |
| spambase | OFS | 0.0974 | 913.15 ± 157.87 | 6 |
| | OFSSGD | 0.0237 | 445.60 ± 19.06 | 35.7 ± 3.14 |
| | OFSSGDK | 0.0218 | 415.65 ± 17.18 | 39.85 ± 2.58 |

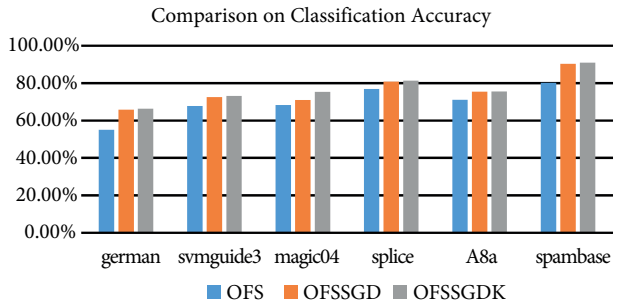


**Figure 1.** Comparison of classification accuracy with medium-sized datasets.

Table 3 shows the mistakes rate provided by the proposed methods on small datasets. Figure 2 shows the classification accuracy obtained by the proposed methods with small datasets. For small datasets like vote, zoo, and heart-Cleveland, OFSSGD and OFSSGDK resulted in good improvement in the accuracy (reduced number of mistakes), while the number of features selected is nearly three-fourths of the original features set. This is because the training instances are much fewer and the SGr is applied to the instances only a few times. This leads to a reduction of weights for small sets of features.

It is noted that the results of OFSSGDK are better than OFSSGD as the OFSSGDK method calls the SGr method once for every $K^{th}$ instance to perform the average reduction in weights. Thus, OFSSGDK is robust to noisy data, whereas OFSSGD may not be robust to noisy data.

**Table 3.** Comparison between algorithms on small datasets.

| Dataset | Algorithm | Execution time | Number of mistakes | Number of features selected |
|---|---|---|---|---|
| vote | OFS | 0.0092 | 86.15 ± 28.91 | 2 |
| | OFSSGD | 0.0033 | 46.50 ± 7.85 | 14.65 ± 1.04 |
| | OFSSGDK | 0.0031 | 45.25 ± 6.85 | 15.15 ± 0.87 |
| zoo | OFS | 0.0027 | 15.45 ± 15.04 | 2 |
| | OFSSGD | 0.0009 | 3.35 ± 2.15 | 15.15 ± 0.81 |
| | OFSSGDK | 0.0008 | 3.40 ± 1.93 | 15.7 ± 0.65 |
| Heart-Cleveland | OFS | 0.0099 | 98.75 ± 12.38 | 5 |
| | OFSSGD | 0.0035 | 82.65 ± 7.66 | 39.6 ± 2.13 |
| | OFSSGDK | 0.0028 | 82.40 ± 7.42 | 43.5 ± 1.23 |

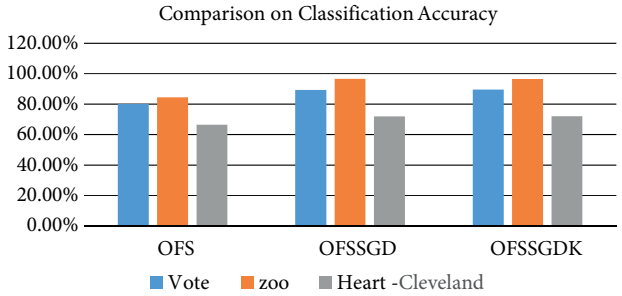


**Figure 2.** Comparison of classification accuracy with small datasets.

In the experiments, large datasets like Kddcup08, Ijcnn1, cod-rna, and Covtype, which have more than 100,000 training instances, are also used to emphasize that the proposed methods result in a significant amount of improvement in the classification accuracy for large datasets. This is shown in Figure 3. Table 4 shows the mistakes rate of the proposed methods with these large datasets. For the Ijcnn1 dataset the OFSSGD selected 8 more features than OFS and provided 13.77% improvement in the accuracy. For the Kddcup08 dataset OFSSGDK provided 11% improvement in the accuracy over OFS by selecting 10 more features. At the same time, OFSSGD selected only 3 more features than OFS and resulted in an accuracy improvement of 9.96%. It is also observed that the OFSSGD and OFSSGDK achieve these better results approximately 4 times faster than the existing OFS algorithm.

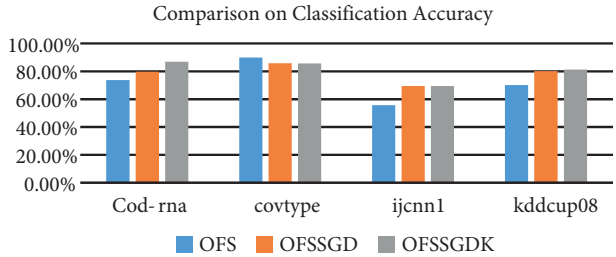


**Figure 3.** Comparison of classification accuracy with large datasets.

These results showcase that the proposed OFSSGD and OFSSGDK methods are better than the OFS algorithm, as our methods do not require any parameters to be given by the user to select features. Though the difference between the number of features that are selected by the existing method and the proposed methods

is high, it is emphasized that the number of features that are selected by the proposed methods is not received as a parameter and is chosen by the method itself. This automated selection of number of features to be selected ensures that the optimal or near optimal feature subset is selected without degrading the classification performance.

**Table 4.** Comparison between algorithms on large datasets.

| Dataset | Algorithm | Execution time | Number of mistakes | Number of features selected |
|---|---|---|---|---|
| Cod-rna | OFS | 8.6036 | 128,717.8 ± 634.5 | 1 |
| | OFSSGD | 2.5286 | 98,982.65 ± 95718 | 3.15 ± 1.69 |
| | OFSSGDK | 2.1922 | 63,875.8 ± 26180.2 | 4.45 ± 1.09 |
| Covtype | OFS | 18.3977 | 63,009.2 ± 6029.54 | 5 |
| | OFSSGD | 4.9755 | 82,721.1 ± 3272.45 | 22.65 ± 1.34 |
| | OFSSGDK | 3.9345 | 83,562.05 ± 1512.9 | 23.50 ± 1.19 |
| Ijcnn1 | OFS | 5.1762 | 84,890.7 ± 9142.12 | 2 |
| | OFSSGD | 1.4128 | 58,571.7 ± 4023.18 | 10.85 ± 1.63 |
| | OFSSGDK | 1.1679 | 58,189.8 ± 3521.61 | 15.60 ± 1.60 |
| Kddcup08 | OFS | 3.5720 | 30,564.10 ± 2244.3 | 13 |
| | OFSSGD | 1.1569 | 20,376.75 ± 513.13 | 16.2 ± 1.79 |
| | OFSSGDK | 0.8568 | 19,161.10 ± 750.99 | 23.65 ± 2.58 |

## 5. Conclusion and future work

In this paper, OFS methods for classification problems are proposed. These proposed methods use the SGr method to select features. This SGr method proportionally decrements the feature weight in the classifier based on a threshold, and the features with zero weights are considered as unimportant features and eliminated from the final selected features set. The experimental results provide evidence that our proposed methods select more features than the existing method and provide considerable improvement in the classification accuracy. The main advantage of the proposed method over the existing method is the automated selection of number features to select. This ensures that even the end user does not have any domain knowledge or information about the dataset and our methods select the optimal or near optimal feature subset without degrading the classification performance.

As a future extension, the proposed SGr method can be used with any online learning algorithms. This method can be extended to multiclass classification. The applications of the method can include fields like health-informatics and image processing that generate large amounts of data.

## References

[1] Liu H, Wu X, Zhang S. A new supervised feature selection method for pattern classification. Comput Intell 2014; 30: 342-361.

[2] Zhu P, Zuo W, Zhang L, Hu Q, Shiu SCK. Unsupervised feature selection by regularized self-representation. Pattern Recogn 2015; 48: 438-446.

[3] Freeman C, Kulic D, Basir O. An evaluation of classifier-specific filter measure performance for feature selection. Pattern Recogn 2015; 48: 1812-1826.

[4] Sarkar C, Cooley S, Srivasta J. Robust feature selection technique using rank aggregation. Appl Artif Intell 2014; 28: 243-257.

[5] Chyzhyk D, Savio A, Grana M. Evolutionary ELM wrapper feature selection for Alzheimer's disease CAD on anatomical brain MRI. Neurocomputing 2014; 128: 73-80.

[6] Crammer K, Dredze M, Pereira F. Exact convex confidence-weighted learning. In: 23rd Annual Conference on Advances in Neural Information Processing Systems; 6–12 December 2009; Vancouver, Canada. pp. 345-352.

[7] Wu X, Yu K, Ding W, Wang H, Zhu X. Online feature selection with streaming features. IEEE T Pattern Anal 2013; 35: 1178-1192.

[8] Perkins S, Theiler J. Online feature selection using grafting. In: 20th International Conference on Machine Learning; 21–24 August 2003; Washington, DC, USA. pp. 592-599.

[9] Zhou J, Foster D, Stine R, Ungar L. Streaming feature selection using alpha-investing. In: ACM 2005 Eleventh SIGKDD International Conference on Knowledge Discovery and Data Mining; 21–24 August 2005; Chicago, IL, USA. New York, NY, USA: ACM. pp. 384-393.

[10] Hoi SCH, Wang J, Zhao P, Jin R. Online feature selection for mining big data. In: ACM 2012 First International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications; 12–16 August 2012; Beijing, China. New York, NY, USA: ACM. pp. 93-100.

[11] Wang J, Zhao P, Hoi SCH, Jin R. Online feature selection and its applications. IEEE T Knowl Data En 2014; 26: 698-710.

[12] Kogan J. Feature selection over distributed data streams. In: Yada K, editor. Data Mining for Service. Heidelberg, Germany: Springer-Verlag, 2014. pp. 11-26.

[13] Langford J, Li L, Zhang T. Sparse online learning via truncated gradient. J Mach Learn Res 2009; 10: 719-743.