Online Streaming Feature Selection via Multi-Conditional Independence and Mutual Information Entropy

Hongyi Wang1, Dianlong You2,*

1Department of Commerce and Trade, Qinhuangdao Vocational and Technical College, No. 90, Lianfeng North Road, Beidaihe District, Qinhuangdao, Hebei 066101 China
2School of Information Science and Engineering, Yanshan University, No. 438, West Section of Hebei Street, Qinhuangdao, Hebei 066004, China

ABSTRACT

The goals of feature selection are to remove redundant and irrelevant features from high-dimensional data, extract the “optimal feature subset” of the original feature space to improve the classification accuracy, and reduce the time complexity. Traditional feature selection algorithms are based on static feature spaces that are difficult to apply in dynamic streaming data environments. Existing works, such as Alpha-investing and Online Streaming Feature Selection (OSFS), and Scalable and Accurate OnLine Approach (SAOLA), have been proposed to serve the feature selection with streaming feature, but they have drawbacks, including low prediction accuracy and a large number of selected features if the streaming features exhibit characteristics such as low redundancy and high relevance. To address the limitations of the abovementioned works, we propose the algorithm of Online Streaming Feature Selection via Conditional dependence and Mutual information (OSFSCM) for streaming feature, which is found to be superior to Alpha-investing and OSFS for datasets with low redundancy and high relevance. The efficiency of the proposed OSFSCM algorithm is validated through a performance test on widely used datasets, e.g., NIPS 2003 and Causality Workbench. Through extensive experimental results, we demonstrate that OSFSCM significantly improves the prediction accuracy and requires fewer selected features compared with Alpha-investing and OSFS.

1. INTRODUCTION

Dimensionality reduction can combat the curse of dimensionality [1–4]. Feature selection chooses an “optimal feature subset” by following a certain criterion while removing redundant and irrelevant features of the class attribute during classification from the original high-dimensional feature set [2,5–7]. This process maintains relevant features for learning and provides faster and more cost-effective predictors with respect to efficiency and effectiveness [8].

In real-world applications, many datasets have very high-dimensional and increasing feature spaces [9,10], such as the datasets from the real-time monitoring and analysis of environments [11], capturing CCTV streaming images [12], collecting features for intrusion detection [13,14], real-time updates of popular topics in Microblog and Twitter [15,16], and real-time spam filtering of email [17]. In such streaming feature spaces, the new, individual features flow into the model dynamically and must be considered sequentially. As an emerging research direction during the past few years, online feature selection with streaming features offers a complementary algorithmic methodology to enrich online feature selection [9,18,19] that addresses high-dimensionality in big data analyses. This methodology assumes that sequential features flow into the model dynamically, and the feature selection is performed when features arrive. This approach is different from classical online learning in which the feature space remains consistent while samples flow in sequentially [19,20]. The representative classical methods [19] include Grafting, Alpha-investing, online streaming feature selection (OSFS), Online Feature Selection with Group Structure (OGFS), and scalable and accurate online approach (SAOLA).

Feature sets with a continuous flow of new features, such as in medical diagnosis, relevant features might be generated with low redundancy. This scenario results in many selected features being generated in the candidate feature set. Experiments from the previously mentioned studies demonstrate that feature selection is restricted in these datasets. For example, the Grafting algorithm is weak at handling streaming features because the entire feature space must be known. The algorithm of Alpha-investing has low prediction accuracy and is unstable. Although the OSFS algorithm offers high prediction accuracy for these types of datasets, the running time increases exponentially with increasing of selected features, so OSFS becomes very slow. The SAOLA algorithm offers good efficiency in its running time while having a few features, but its prediction accuracy is not excellent.
Motivated by the above analysis of existing studies of online feature selection with streaming features, we investigate the problem with a novel algorithm called Online Streaming Feature Selection via Conditional dependence and Mutual (OSFSCM) information to solve the following challenges: (1) how to select the feature subset in a streaming feature space with low redundancy and obtain high relevancy, (2) how to provide effective continuous mechanisms to discover the pattern of the running time among all generated features with increasing feature volumes, and (3) how to validate the performance.

The contributions of this paper that extend beyond the existing methods include our proposal of a novel algorithm for streaming feature selection, an analysis of its theoretical properties, and a validation of its empirical performance through extensive experiments.

The remainder of the paper is organized as follows: Section 2 surveys the related work and Section 3 introduces notations, definitions, and a framework for streaming feature filtering as well as the proposal of our OSFSCM algorithm and analysis. Section 4 reports our experimental results and describes its application in a real scenario, followed by our conclusion in Section 5.

2. RELATED WORK

Feature selection is simple, interpretable, and essential when working with high-dimensional data [18,19]. Based on the available label information, feature selection can include supervised, unsupervised, and semi-supervised methods. From the perspective of a selection strategy, these algorithms can include a wrapper, filter, or embedded method [19].

Online feature selection for streaming features is an important branch of feature selection [4,8,21] that focuses on scenarios where the number of features changes over time, which must be processed in real-time and not after all the features arrive. Representative algorithms of this field include Grafting, Alpha-investing, OSFS, OGFS, and SAOLA.

Grafting [22] is an embedded feature selection method applied to streaming features with irrelevant and redundant features. However, Grafting is weak for this approach because it must know about the entire feature space to select a good value for its regularization parameter. Alpha-investing [23] can handle infinitely large feature sets but also must transform the initial features based on prior knowledge of candidate features. Only judging the relevance of features can give many redundant features in the candidate set resulting in a reduction of classification accuracy. OSFS [24] uses the $G^2$ test to denote conditional independence or dependence between features from which it identifies irrelevant and redundant features. While OSFS can remove these irrelevant and redundant features from streaming features, when the number of weak relevant features increases, the running time increases exponentially. OGFS [25] leverages the group structure information as a priori knowledge for feature selection and generates a better feature subset through two stages of intra-group and inter-group feature selection. However, OGFS must select a few parameters in advance, which is relatively difficult without prior information. SAOLA [15] can handle an extremely high-dimensional feature space by using a strategy of online pairwise comparisons. However, SAOLA only judges if the features are redundant under a single condition and cannot remove all redundant features. Besides, the algorithm alone cannot obtain the judgment threshold of the optimal relevance between features.

Based on the above observations, we propose an efficient OSFSCM algorithm for feature selection in a streaming feature space that offers low redundancy and high relevance.

3. THE OSFSCM ALGORITHM FOR ONLINE FEATURE SELECTION

3.1. Notations and Definitions

The selected features of the class attribute represent an optimal feature subset [24,26] containing all strongly relevant and nonredundant features [24], as shown in Figure 1. The optimal feature subset is named the Markov blanket of the class attribute, and this criterion only removes the attributes that are unnecessary for inclusion into the feature set because they are irrelevant or redundant [7,24,27]. We consider this problem in the context of online feature selection with streaming features where $S$ is the feature space set containing all available features.

Table 1 lists all symbols and notations used in this paper. Assume that $x_i$ denotes the $i$th new incoming feature at time $t_i$, $S_{t_i-1}$ is the selected feature set at time $t_{i-1}$, $S_{t_i}$ is the feature space set containing all available features.

### Table 1: Notations with mathematical meanings

| Notation | Mathematical Meanings |
|----------|-----------------------|
| $S$      | Feature space set     |
| $f$      | Feature $f, f \in S$  |
| $CFS$    | Candidate feature set at current time |
| $t$      | A time point          |
| $T$      | Class attribute       |
| $P(x)$   | Event probability of feature $x$ |
| $P(.)$   | Conditional probability |
| $\rho$   | A threshold           |
| $MB(T)$  | Markov blanket of $T$ |
| $a \perp b$ | $a$ is independent of $b$ |
factorizations of the allowed joint probability distribution. If $x$ and $y$ are dependent in the condition of $S_i$, then we write $P(x, y | S_i) = P(x | S_i)P(y | S_i)$.

Definition 3 (Markov blanket [15,29,30], MB). A Markov blanket of the class attribute $T$, denoted as $MB(T)$, is a minimal set of features that makes $\forall x \in S - MB(T) \cup \{T\}$, such that $x \perp T | MB(T)$. $MB(T)$ is the set of parents, children, and children’s parents of $T$ [31]. These Markov blankets can eliminate a conditionally independent feature without increasing the distance from the desired distribution. The Markov blanket criterion only removes the attributes that are unnecessary based on those that are irrelevant to the class attribute and redundant given other attributes [26].

Definition 4 (Strong relevance [24]). A feature $x$ is strongly relevant to the class attribute, $T$, iff $\forall S_i \subseteq S - \{x\}$, such that $P(T | S_i, x) \neq P(T | S_i)$.

Definition 5 (Weak relevance [24]). A feature $x$ is weakly relevant to the class attribute, $T$, iff $\exists S_i \subseteq S - \{x\}$, such that $P(T | x, S_i) \neq P(T | S_i)$.

Definition 6 (Irrelevance [24]). A feature $x$ is irrelevant to a class attribute, $T$, iff $\forall S_i \subseteq S - \{x\}$, s.t. $P(T | S_i, x) = P(T | S_i)$.

Definition 7 (Redundant features [24]). A feature $x$ is redundant to the class attribute $T$ iff it is weakly relevant to $T$ and has a $MB(x)$ that is a subset of $MB(T)$.

3.2. Using Mutual Information to Filter Partial Redundant Features

Mutual information theory and Fisher’s z-test are used to judge the correlation between features [15]. The evaluation standard of entropy information quantity, also known as Shannon entropy, uses a numerical value to express the uncertainty degree of a random variable. The entropy of feature $Y$ is defined as

$$H(Y) = - \sum_{y_i \in Y} P(y_i) \log_2 P(y_i)$$

(1)

Conditional entropy refers to the uncertainty degree of another variable after it is known, i.e., the dependence degree of the variable on the known variable. Assuming the feature $Z$ is known, the conditional entropy of $Y$ under the $Z$ condition is defined as

$$H(Y | Z) = \sum_{z \in Z} P(z) \sum_{y_i \in Y} P(y_i | z) \log_2 P(y_i | z)$$

(2)

Conditional mutual information is the degree of correlation between two features under the condition that a certain feature is known. The expression of conditional mutual information is established according to Eqs. (3) and (4) [15].

$$I(X; Y | Z) = H(X | Z) - H(X | YZ)$$

$$= H(X | Z) + H(Y | Z) - H(Z, Y, Z) - H(Z)$$

(3)

When a new feature $f$ arrives, to determine if it is relevant for the class attribute $T$, the algorithm calculates a correlation threshold $\delta$, such that if $I(f; T) > \delta$ ($0 \leq \delta \leq 1$), and $f_i \in CFS - \{f\}$, then $f$ is considered a relevant feature for the class attribute $T$. Meanwhile, if Eq. (4) is hold [15], the $f$ term is considered a redundant feature, so it is then filtered and not considered in the next process.

$$I(f; T) \geq I(f_i; T) \text{ and } I(f; f_i) \geq I(f_i; T)$$

(4)

3.3. Using Fisher’s z-Test to Filter Irrelevant and Partially Redundant Features with Continuous Data

Fisher’s z-test calculates the correlation degree between features [15], as in shown in Eq. (5). In the Gaussian distribution $N(\mu, \Sigma)$, after the feature subset $S$ is provided, the expression of the partial correlation coefficient $r_{(f,T|S)}$ between the feature $f_i$ and the class attribute $T$ is expressed as [15].

$$r_{(f,T|S)} = \frac{r_{(f,T)} - r_{(f,S)}r_{(T,S)}}{\sqrt{1 - r_{(S,T)}^2}}$$

(5)

In Fisher’s z-test, under the null hypothesis of the conditional independence between the feature $f_i$ and the class attribute $T$ of the given feature subset $S$, $r_{(f_i,T|S)} = 0$. Suppose $\alpha$ is a given significance level, and $\rho$ is a $p$-value returned by Fisher’s z-test. Under the null hypothesis of the conditional independence of $f_i$ and $T$, if $\rho > \alpha$, $f_i$ and $T$ are not related to each other when the subset $S$ is given. If $\rho \leq \alpha$, then $f_i$ and $T$ are relevant to each other.

3.4. Using the G$^2$ Test to Filter Irrelevant and Partially Redundant Features with Discrete Data

As an alternative to the $X^2$ test, the $G^2$ test is a statistic defined as Eq. (6) [24]

$$G^2 = 2 \sum_{i,j,k} s_{abc}^{ijk} \ln \frac{s_{abc}^{ijk}c_{jk}}{s_{ab}^{ik}c_{bk}}$$

(6)

where $s_{abc}^{ijk}$ represents the number of features satisfying $f_i = a, f_j = b,$ and $f_k = c$ in a dataset, and $s_{ab}^{ik}, s_{bk}^{ic}, s_{bk}^{ic}$ are defined similarly. If three features $f_i, f_j,$ and $f_k$, and $f_i, f_j, f_k$ are conditionally independent given $f_k$. Under the condition where $\alpha$ has a given significance level of 0.05, then $\rho$ is the $p$-value returned and $f_i \perp f_j | f_k$ defines the null hypothesis ($H_0$) of $f_i$ and $f_j$ is conditionally independent given $f_k$ iff $\rho > \alpha$. Otherwise, for ($H_1$), $f_i$ and $f_j$ are nonconditionally independent given $f_k$ iff $\rho \leq \alpha$, i.e. $f_i \not\perp f_j | f_k$.

3.5. The OSFSCM Algorithm and Analysis

In this section, we propose a new approach for feature filtering via multi-conditional independence and mutual information entropy to process data with streaming features. In this approach, the data stream is fixed, while the features continue arriving as each feature is evaluated. The process of feature selection can be performed in three phase, as shown in Table 2. First, the irrelevant features are filtered by the nonconditional independence and leaving the
relevant features. Second, part of the redundant features is further discarded from the weakly relevant features by filtering through mutual information. Finally, the remaining redundant features are discarded by filtering through conditional independence.

According to the above approach, The OSFSCM algorithm for streaming feature filtering is proposed, as is shown in Algorithm 1.

Algorithm 1: The OSFSCM algorithm

1. Initialization
   (1) Selected features set \( CFS = \emptyset \), class attribute \( T \)
2. Analysis of irrelevant features: filtering via non-conditional independence
   (2) Get a new feature \( f \);
   (3) If \( f \) is an irrelevant feature, discard \( f \), enter step (2);
   (4) If not, \( CFS = CFS \cup \{ f \} \), enter step 3;
3. Analysis of redundant features 1: the filtering via mutual information
   (5) If \( \exists x \in CFS \), \( x \) is the redundant feature in the filtering via mutual information, discard \( x \), and enter step (2); Once all features in \( CFS \) is non-redundant according to Eq. (4); then, step (6);
4. Analysis of redundant features 2: the filtering via conditional independence
   (6) If \( \exists y \in CFS \), subSet \( CFS \setminus \{ y \} \), \( f \) is the redundant feature according to Eq. (5) or Eq. (6), discard \( y \);
5. Repeat steps 2–5 until no new features or stopping criteria is met;
6. Output the selected features \( CFS \).

In the OSFSCM algorithm, \( CFS \) is a candidate feature set at the current time, and \( f \) is the new feature. In step 2, nonconditional independence filtering is executed. If it returns an irrelevant feature of class attribute \( T \), then the feature \( f \) is discarded. Otherwise, the feature \( f \) goes into further analysis of the redundant features.

The filtering of the redundant features is divided into two sequential steps. Step 3 includes the filtering via mutual information and step 4 filters via conditional independence.

For filtering via mutual information, according to Eq. (4), a new feature \( f \) is relevant with \( T \), \( \exists x \in CFS \setminus \{ f \} \), if \( I(x; T) \geq I(f; T) \) and \( I(f; x) \geq I(f; T) \), then \( x \) can be removed from \( CFS \) until any feature in \( CFS \setminus \{ f \} \) is nonredundant in the condition of \( f \).

For filtering via conditional independence, the candidate feature set includes the new feature \( f \) and the expression of \( y \perp T | \text{subSet} \) is determined on the condition of \( \text{subSet} \subseteq CFS - \{ y \} \). If \( y \perp T | \text{subSet} \), then the feature \( y \) is redundant, and \( y \) is discarded from \( CFS \). Through continuous looping of steps 3–4, all redundant features in \( CFS \) are discarded due to the inclusion of new features.

The OSFSCM uses the notation \( x \perp T | S, S \subseteq CFS - \{ x \} \) to denote the conditional independence. To measure \( x \perp T | S \), OSFSCM uses the \( p \)-value returned by the \( G^2 \) test for discrete data and Fisher's \( z \)-test for continuous data of its measurements with significance levels of 0.05 or 0.01, where the former threshold is used in this paper.

### 3.6. The Time Complexity of OSFSCM

The complexity of the OSFSCM algorithm depends on the tests of nonconditional independence filtering, mutual information filtering, and conditional independence filtering. Assume that \(|N|\) is the number of features arrived, and \(|N|\) also is the number of remaining features before filtering via non-conditional independence, \(|S_1|\) is the number of remaining features before filtering via mutual information, and \(|CFS|\) is the number of remaining features before filtering via conditional independence. For OSFSCM, we use the \( k \)-greedy search strategy with \( k = 3 \). The complexity of each filtering phase is shown in Table 3. Therefore, the complexity is represented as \( O\left(\frac{|N|}{|CFS|} \cdot \frac{|S_1|}{|CFS|} \cdot 3\right) \). Therefore, the search times exponentially decline. Meanwhile, numerous irrelevant and redundant features exist in the large data condition, so the size of \(|CFS|\) is very small. The time complexity determined by the number of features within \( CFS \) and the worst-case complexity is \( O\left(\frac{|N|}{3^{|CFS|}}\right) \) when the size of \(|S_1|\) is \(|N|\) during step (5) and \(|CFS|\) is \(|N|\) during step (6). With new features continuously arriving, the complexity of OSFSCM becomes very high. This means that almost all features are strongly related with class attributes are, and there are almost no irrelevant features and redundant features in the dataset. Few features are filtered out by steps 2 to 5. Obviously, this situation nearly does not exist in real flow datasets.

### 4. EXPERIMENTAL RESULTS

#### 4.1. Experiment Setup

We empirically evaluate the performance of the OSFSCM algorithm on 14 benchmark datasets listed in Table 4. All experiments are performed on a computer with an Intel(R) Xeon(R) CPU E3-1505M 3.0 GHz with 32 G RAM.

The arcene, colon, ionosphere, and leukemia datasets are sourced from the NIPS 2003 feature selection challenge [8] and a frequently studied public microarray datasets (wdbc). We also downloaded the datasets from the Causality Workbench that includes sy1va, lung, cina0, regel1, lucas0, marti1, and lucap0. The cina0 is a marketing dataset derived from census data. The regel1 is a genomics dataset for studying the causes of lung cancer. The marti1 is obtained from the data generative process of simulated genomic data. The lucas0 is a lung cancer simple set, and lucap0 is a lung cancer set with probes, which are used for modeling a medical application for the

| Phase of filtering | Cost |
|-------------------|------|
| Phase1: filtering via nonconditional independence | \( O(|N|) \) |
| Phase2: filtering via mutual information | \( O(|S_1| \cdot |CFS|) \) |
| Phase3: filtering via conditional independence | \( O\left(\frac{|CFS|}{3} \cdot |CFS| \right) \) |

OSFSCM, online streaming feature selection via conditional dependence and mutual information.

| Table 3 | The time complexity of OSFSCM algorithm. |
|---------|-----------------------------------------|

Phase of filtering | Cost |
|-------------------|------|
| Phase1: filtering via nonconditional independence | \( O(|N|) \) |
| Phase2: filtering via mutual information | \( O(|S_1| \cdot |CFS|) \) |
| Phase3: filtering via conditional independence | \( O\left(\frac{|CFS|}{3} \cdot |CFS| \right) \) |
diagnosis, prevention, and cure of lung cancer. The number of features available in these datasets ranges from 11 to 10000, and the number of samples varies from 72 to 145252. In particular, the number of features of seven datasets is larger than the number of samples, including marti1, reged1, lung, prostate_GE, leukemia, arcene, and Smk_can_187. These 14 datasets cover a wide range of real-world application domains, including gene expressions, ecology, and casual discovery, making the construction of feature selection very challenging. We preprocess the data by turning to a standard score and deleting the same columns, such as in leukemia and other datasets.

Our analysis of the OSFSCM algorithm compares against the state-of-the-art online feature selection algorithms, Alpha-investing, and OSFS, using 10-fold cross-validation on each training dataset. First, we compared the prediction accuracy of OSFSCM with that of the state-of-the-art using 12 classifiers, including Decision Tree, KNN, SVM, and Ensemble, as implemented in MATLAB. Second, we analyzed the number of selected features and the running time for each algorithm. Third, the OSFSCM algorithm is applied to a real-world data scenario and compared with the other algorithms.

4.2. Comparison of OSFSCM with Two Online Algorithms

The algorithms are implemented in Library of Online Streaming Feature Selection (LOFS) [18], an open-source library available in MATLAB 2017. To evaluate the selected features in the experiments, we use twelve available classifiers in the app of “classification learner” in MATLAB 2017, including Decision Tree, KNN, SVM, and Ensemble, as implemented in MATLAB. Second, we analyzed the number of selected features and the running time for each algorithm. Third, the OSFSCM algorithm is applied to a real-world data scenario and compared with the other algorithms.

Table 4 | The benchmark datasets used for algorithm evaluation.

| Dataset   | #Features | #Instances | Dataset   | #Features | #Instances |
|-----------|-----------|------------|-----------|-----------|------------|
| wdbc      | 30        | 569        | marti1    | 1024      | 500        |
| colon     | 2000      | 62         | reged1    | 999       | 500        |
| lucas0    | 11        | 2000       | lung      | 3312      | 203        |
| sylva     | 216       | 13086      | prostate_GE | 5966   | 102        |
| ionosphere| 34        | 351        | leukemia  | 7066      | 72         |
| cina0     | 132       | 16033      | arcene    | 10000     | 100        |
| lucap0    | 143       | 2000       | Smk_can_187 | 19993  | 187        |

#Features: the number of features; #Instances: the number of instances.

Table 5 | Parameter settings in 12 classifiers.

| Classifier Parameters | Parameter Values | Parameter Values | Parameter Values |
|-----------------------|------------------|------------------|------------------|
|                       | Complex Tree     | Medium Tree      | Simple Tree      |
| Maximum number of splits | 100              | 20               | 10               |
| Split criterion        | Gini’s diversity index | Off             | 10               |
| &gt; surrogate decision splits &gt; Maximum surrogates per node
|                       | Linear SVM       | Quadratic SVM    | Cubic SVM        |
| Kernel function        | Linear           | Quadratic        | Cubic            |
| Box constraint level    | 1                | Auto             |                  |
| Kernel scale mode       | 1                |                  |                  |
| Manual kernel scale     |                  |                  |                  |
| Fine KNN               | 10               | 10               | 10               |
| Distance metric         | Euclidean        | Euclidean        | Minkowski        |
| Distance weight         | Equal            | 1                |                  |
| Box constraint level     | Auto             |                  |                  |
| Kernel scale mode       | 1                |                  |                  |
| Manual kernel scale     |                  |                  |                  |
| Bagged Trees            | Bag              | Subspace         | RUSBoosted Trees |
| Learner type            | Decision tree    | Discriminant     | RUSBoost         |
| Maximum number of splits | 266              | 20               | 20               |
| Number of learners      | 30               |                  |                  |
| Learning rate           | 0.1              |                  |                  |
| Subspace dimension      | 1                |                  |                  |
models. After training multiple models, compare these algorithms’ performance side-by-side in these classifiers.

As described above, we evaluate OSFSCM against the others based on prediction accuracy, the number of selected features, and the running time. In the following, we perform statistical comparisons to analyze the prediction accuracies further.

### 4.2.1. Prediction Accuracy

Figure 2 summarizes the prediction accuracies with 12 classifiers applied to 14 datasets during online learning. We conduct these tests using $G^2$ for discrete data and Fisher’s z-test for continuous data at an $\alpha = 0.05$ significance level. The prediction accuracies of OSFSCM and OSFS are higher compared with Alpha-investing for most of the 5–14 datasets for these classifiers. OSFSCM achieves a higher accuracy for all the classifiers except for RUSBoosted Trees. With some datasets, as seen in Figure 2, the accuracies of the classifiers are reduced too much, such as in leukemia, marti1, and reged1.

In addition, from the three curves in Figure 2, the prediction accuracies from Alpha-investing, OSFS, and OSFSCM are nearly the same for some datasets, including six datasets using Complex Tree, three each using Medium Tree, Liner SVM, Quadratic SVM, Cubic SVM, Fine KNN, Medium KNN, and Subspace discriminant, two datasets with Cubic KNN, Bagged Trees, and RUSBoosted Trees, and one dataset with Simple Tree. For the wdbc, lucas0, and sylva datasets, the prediction accuracies of these classifiers, as well as RUSBoosted Trees, remain nearly the same.

#### 4.2.2. Number of Selected Features and Running Time

To further examine these three algorithms, Table 6 lists the performance of each with respect to the 14 datasets comprised of different numbers of features.

- **Summary of the numbers of selected features**

  While the prediction accuracy of OSFSCM is higher than Alpha-investing and OSFS for most datasets, as previously described in Figure 2, another observation is apparent in Table 6. The number of selected features from OSFSCM is more than Alpha-investing and OSFS for many datasets. This result is due to the following reasons:

  - **For the Alpha-investing algorithm**: From its low prediction accuracy, the ability of the algorithm to mine features is low, so part of the Markov blanket elements cannot be obtained.
  - **For the OSFS algorithm**: During the redundant feature analysis phase, the non-redundant features could be discarded under the condition of redundant features due to its low prediction accuracy, so fewer features are selected.

![Figure 2](image_url)

*Figure 2* | Prediction accuracies of the three algorithms on 14 datasets using classifiers on the selected features. The x-axis labels represent the 14 datasets as (1) wdbc, (2) colon, (3) lucas0, (4) sylva, (5) ionosphere, (6) cina0, (7) lucap0, (8) marti1, (9) reged1, (10) lung, (11) prostate_GE, (12) leukemia, (13) arcene, and (14) Smk_can_187.
For the OSFSCM algorithm: A large number of selected features is potential due to its significant outperformance over OSFS and Alpha-investing in mining the elements from the Markov blanket, enabling it to find more elements.

• Summary of the running times

The running time performances for the three algorithms are also reported in Table 6. The Alpha-investing is faster than OSFS and OSFSCM for all datasets because it only considers the newest feature added, and never considered the discarded features again. This approach also results in low prediction accuracy (as seen in Figure 2) with the classifiers, as with the datasets colon, lucap0, reged1, leukemia, and Smk_can_187.

The performances of the running times in OSFSCM and OSFS are very different, as seen in the results in Table 6, with OSFS being much faster than OSFSCM on the datasets of wdbc, colon, lucas0, ionosphere, lung, prostate_GE, arcene, leukemia, and Smk_can_187.

On the other hand, OSFSCM is much faster compared with OSFS on the datasets of sylva, cina0, lucap0, reged1, and lung, as highlighted in bold. These datasets include features with higher feature selection, as shown in Table 4, and have low redundancy with high strong and weak relevance because the number of candidate features significantly influences the runtime for the two algorithms.

4.3. Application in a Real Scenario

The PEMS-SF dataset from the UCI website is selected for our real scenario test for algorithm evaluation. This dataset contains 440 instances and 138672 features based on 963 sensors recordings between January 1, 2008, and September 30, 2009, of daily lane occupancy rates on highways. Each feature represents a lane occupancy rate from a sensor for a day (between 0 and 1) with a classification label of 1 through 7, representing Monday to Sunday. The dataset deleted data on public holidays and anomalies observed on March 8 and March 9, 2008. This evaluation only considers the OSFSCM, OSFS, and SAOLA algorithms because the Alpha-investing algorithm cannot run in the PEMS-SF dataset. The significance level $\alpha$ is set to 0.05, and the experiment performs multiple 10-fold cross-validations.

Table 7 reports the running time of the OSFSCM algorithm on the PEMS-SF dataset. As the features flow in, OSFSCM removes many irrelevant and redundant features. First, irrelevant features can be removed by the filtering of nonconditional independence, followed by the removal of redundant features by the filtering of mutual information and conditional independence. Finally, the approximate Markov blanket of the classification label is obtained by the three filters.

Compared with the low prediction accuracy of Alpha-investing and a high running time cost (greater than three days) of OSFS on the

| Dataset       | Alpha-Investing | OSFS     | OSFSCM       |
|---------------|-----------------|----------|--------------|
|               | #Features       | Time(s)  | #Features    | Time(s)  | #Features       | Time(s)  |
| wdbc          | 20              | 0.0138   | 3            | 0.1577   | 3              | 0.0567   |
| colon         | 1               | 0.0663   | 3            | 0.6778   | 4              | 0.0582   |
| lucas0        | 4               | 0.0008   | 4            | 0.0142   | 4              | 0.0061   |
| sylva         | 70              | 1.6717   | 18           | 247.9366 | 22             | 32.5611  |
| ionosphere    | 10              | 0.0147   | 4            | 0.1315   | 7              | 0.0669   |
| cina0         | 8               | 0.1046   | 22           | 721.3638 | 28             | 62.1947  |
| lucap0        | 10              | 0.0197   | 36           | 1.43E+07 | 41             | 72.5523  |
| martil        | 28              | 0.116    | 1            | 0.1081   | 1              | 0.1045   |
| reged1        | 1               | 0.0417   | 13           | 121.2839 | 13             | 27.3695  |
| lung          | 45              | 0.7523   | 11           | 420.5678 | 20             | 85.3698  |
| prostate_GE   | 12              | 0.4308   | 3            | 7.7915   | 4              | 3.1189   |
| leukemia      | 1               | 0.4346   | 3            | 12.7647  | 5              | 8.3657   |
| arcene        | 8               | 1.4139   | 5            | 20.8445  | 6              | 5.8624   |
| Smk_can_187   | 6               | 2.7929   | 4            | 42.8323  | 7              | 12.1648  |

OSFS, online streaming feature selection; OSFSCM, online streaming feature selection via conditional dependence and mutual information.

| Algorithm   | Time(s)  | #Features       | Ratio (%) | Filtering via Nonconditional Independence | Filtering via Mutual Information | Filtering via Conditional Independence |
|-------------|----------|-----------------|-----------|-------------------------------------------|----------------------------------|--------------------------------------|
| OSFSCM      | 476.3568 | 25              | 14295     | 98                                        | 21                               |                                       |
|             | 50       | 31461           | 110       | 33                                        |                                  |                                       |
|             | 75       | 46636           | 150       | 35                                        |                                  |                                       |
|             | 100      | 61618           | 210       | 41                                        |                                  |                                       |

OSFSCM, online streaming feature selection via conditional dependence and mutual information.
Using benchmark datasets, we compared the OSFS algorithm with conditional dependence and mutual information. We explored OSFS through a proposed algorithm called OSFSCM. In Figure 3, the classification accuracies of OSFSCM on Decision Tree (simple tree), SVM (qualitative SVM), KNN (cosine KNN), and embedded (bagged trees) are higher compared with SAOLA, with differences of 8.63%, 3.11%, 10.03%, and 2.81%, respectively.

### 5. CONCLUSIONS

We explored OSFS through a proposed algorithm called OSFSCM designed with conditional dependence and mutual information. Using benchmark datasets, we compared the OSFSCM algorithm with two state-of-the-art online feature selection methods, and our empirical results demonstrated the following: (1) The prediction accuracy of OSFSCM is higher compared with Alpha-investing and OSFS for many datasets. (2) The efficacy of the OSFSCM algorithm is high for these datasets, especially those with low redundancy and high relevance. (3) The number of selected features from OSFSCM is greater compared with Alpha-investing and OSFS for many datasets. (4) The validity of the OSFSCM algorithm is verified by tests using a real-world dataset scenario.

### ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the article.

### REFERENCES

[1] J. Tang, S. Alelyani, H. Liu, Feature selection for classification: a review, in: Data Classification: Algorithms and Applications, CRC Press, 2014, p. 37.
[2] J. Li, K. Cheng, S. Wang, F. Morstatter, R.P. Trevino, J. Tang, H. Liu, Feature selection: a data perspective, ACM Comput. Surv. 50 (2017), 94.
[3] V. Kumar, Feature selection: a literature review, Smart Comput. Rev. 4 (2014), 211–229.
[4] D. You, X. Wu, L. Shen, Y. He, X. Yuan, Z. Chen, S. Deng, C. Ma, Online streaming feature selection via conditional independence, Appl. Sci. 8 (2018), 2548.
[5] J. Yang, X. Guo, N. An, A. Wang, K. Yu, Streaming feature-based causal structure learning algorithm with symmetrical uncertainty, Inf. Sci. 467 (2018), 708–724.
[6] N. Alluaimi, M.M. Masud, M.A. Serhani, N. Zaki, Streaming feature selection algorithms for big data: a survey, Appl. Comput. Inform. (2019).
[7] K. Yu, H. Chen, Markov boundary-based outlier mining, IEEE Trans. Neural Netw. Learn. Syst. 30 (2019), 1259–1264.
[8] D. You, X. Wu, L. Shen, S. Deng, Z. Chen, C. Ma, Q. Lian, Online feature selection for streaming features using self-adaption sliding-window sampling, IEEE Access. 7 (2019), 16088–16100.
[9] K. Yu, W. Ding, D.A. Simovici, H. Wang, J. Pei, X. Wu, Classification with streaming features: an emerging-pattern mining approach, ACM Trans. Knowl. Discov. Data. 9 (2015), 1–31.
[10] J. Cai, J. Luo, S. Wang, S. Yang, Feature selection in machine learning: a new perspective, Neurocomputing. 300 (2018), 70–79.
[11] X. Wu, X. Zhu, G.-Q. Wu, W. Ding, Data mining with big data, IEEE Trans. Knowl. Data Eng. 26 (2014), 97–107.
[12] X. Jia, B.-C. Kuo, M. Crawford, Feature mining for hyperspectral image classification, Proc. IEEE. 101 (2013), 676–697.
[13] F. Zhao, J. Zhao, X. Niu, S. Luo, Y. Xin, A filter feature selection algorithm based on mutual information for intrusion detection, Appl. Sci. 8 (2018), 1535.
[14] S. Dong, X. Wang, Research on network intrusion data based on KNN and feature extraction algorithm, in: Q. Zhou, Y. Gan, W. Jing, X. Song, Y. Wang, Z. Lu (Eds.), Data Science, Springer, Singapore, 2018, pp. 182–191.
[15] K. Yu, X. Wu, W. Ding, J. Pei, Scalable and accurate online feature selection for big data, ACM Trans. Knowl. Discov. Data. 11 (2016), 16.
[16] W. Xie, F. Zhu, J. Jiang, E.-P. Lim, K. Wang, Topicsketch: real-time bursty topic detection from twitter, IEEE Trans. Knowl. Data Eng. 28 (2016), 2216–2229.
[17] D. Wang, D. Irani, C. Pu, Evolutionary study of web spam: Webb Spam Corpus 2011 versus Webb Spam Corpus 2006, in: 2012 8th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), Pittsburgh, USA, 2012, pp. 40–49.
[18] K. Yu, W. Ding, X. Wu, LOFS: a library of online streaming feature selection, Knowl. Based Syst. 113 (2016), 1–3.

**Figure 3** Classification accuracies on the PEMS-SF dataset using four classifiers.
[19] X. Hu, P. Zhou, P. Li, J. Wang, X. Wu, A survey on online feature selection with streaming features, Front. Comput. Sci. 12 (2018), 479–493.

[20] J. Wang, P. Zhao, S. C. Hoi, R. Jin, Online feature selection and its applications, IEEE Trans. Knowl. Data Eng. 26 (2014), 698–710.

[21] D. You, X. Wu, C. Ma, L. Shen, S. Deng, Z. Chen, Online feature selection for streaming features with high redundancy using sliding-windowsampling, in Proceedings - 9th IEEE International Conference on Big Knowledge (ICBK), Singapore, 2018, pp. 205–212.

[22] S. Perkins, J. Theiler, Online feature selection using grafting, in Proceedings of the 20th International Conference on Machine Learning (ICML-03), Washington D.C, USA, 2003, pp. 592–599. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.13.9658

[23] J. Zhou, D. Foster, R. Stine, L. Ungar, Streaming feature selection using alpha-investing, in Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, Chicago, IL, USA, 2005, pp. 384–393.

[24] X. Wu, K. Yu, W. Ding, H. Wang, X. Zhu, Online feature selection with streaming features, IEEE Trans. Pattern Anal. Mach. Intell. 35 (2013), 1178–1192.

[25] J. Wang, M. Wang, P. Li, L. Liu, Z. Zhao, X. Hu, X. Wu, Online feature selection with group structure analysis, IEEE Trans. Knowl. Data Eng. 27 (2015), 3029–3041.

[26] D. Koller, M. Sahami, Toward Optimal Feature Selection, in Proceedings of the Thirteenth International Conference on Machine Learning (ICML’96), San Francisco, CA, USA, 1996, pp. 284–292. https://dl.acm.org/doi/10.5555/3091696.3091731

[27] H. Wang, Z. Ling, K. Yu, X. Wu, Towards efficient and effective discovery of Markov blankets for feature selection, Inf. Sci. 509 (2020), 227–242.

[28] R. Kohavi, G.H. John, Wrappers for feature subset selection, Artif. Intell. 97 (1997), 273–324.

[29] K. Yu, X. Wu, W. Ding, Y. Mu, H. Wang, Markov blanket feature selection using representative sets, IEEE Trans. Neural Netw. Learn. Syst. 28 (2017), 2775–2788.

[30] Z. Ling, K. Yu, H. Wang, L. Liu, W. Ding, X. Wu, BAMB: a balanced Markov blanket discovery approach to feature selection, ACM TIST. 10 (2019), 52:1–52:25.

[31] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, in: M. Morgan, D. Pearl (Eds.), Morgan Kaufmann publishers, San Francisco, CA, USA, 2014, pp. 96–104. https://dl.acm.org/doi/book/10.5555/52121