Pattern Analysis and Applications (2019) 22:949–963
https://doi.org/10.1007/s10044-018-0690-7

THEORETICAL ADVANCES

Online streaming feature selection: a minimum redundancy, maximum significance approach

Mohammad Masoud Javidi[1] · **Sadegh Eskandari**[1,2]

Received: 15 December 2016 / Accepted: 2 February 2018 / Published online: 10 February 2018

Abstract
All the traditional feature selection methods assume that the entire input feature set is available from the beginning. However, online streaming features (OSF) are integral part of many real-world applications. In OSF, the number of training examples is fixed while the number of features grows with time as new features stream in. A critical challenge for online streaming feature selection (OSFS) is the unavailability of the entire feature set before learning starts. OS-NRRSAR-SA is a successful OSFS algorithm that controls the unknown feature space in OSF by means of the rough sets-based significance analysis. This paper presents an extension to the OS-NRRSAR-SA algorithm. In the proposed extension, the redundant features are filtered out before significance analysis. In this regard, a redundancy analysis method based on functional dependency concept is proposed. The result is a general OSFS framework containing two major steps, (1) online redundancy analysis that discards redundant features, and (2) online significance analysis, which eliminates non-significant features. The proposed algorithm is compared with OS-NRRSAR-SA algorithm, in terms of compactness, running time and classification accuracy during the features streaming. The experiments demonstrate that the proposed algorithm achieves better results than OS-NRRSAR-SA algorithm, in every way.

Keywords Feature selection · Rough set theory · Online streaming feature selection · Functional dependency

1 Introduction

In many practical machine learning tasks, we encounter a very large feature space with thousands of irrelevant and/or redundant features [4, 13, 24–26]. Feature selection is an important pre-processing step to cope with the course of dimensionality. The task of feature selection is to select a small subset of most important and discriminative input features. Traditional feature selection methods consider that all input features are available from the beginning. However, incrementally update knowledge in data mining is getting more and more popular. The volume of data is growing at an unprecedented rate, both in the number of features and instances [23, 29]. Online streaming features (OSF) is the incrementally data growing scenario, where the number of instances is fixed while feature set grows with time. There are several scenarios where the feature space is unknown or even infinite and therefore the OSF consideration is inevitable. For example:

- In bioinformatic and clinical machine learning problems, acquiring the entire set of features for every training instance is expensive due to the high cost laboratory experiments [38].
- In texture-based image segmentation problems, the number of different texture filters can be infinite and therefore acquiring the entire feature set is infeasible [12, 29, 40].
- In statistical relational learning, an agent may search over the space of SQL queries to augment the base set of candidate features found in the tables of a relational database. The number of candidate features generated by such a method is limited by the amount of CPU time available to run SQL queries. Generating 100,000 features can easily take 24 CPU hours, while millions of features may be

✉ Mohammad Masoud Javidi
javidi@uk.ac.ir

Sadegh Eskandari
eskandari@math.uk.ac.ir; eskandari@guilan.ac.ir

[1] Shahid Bahonar University of Kerman, Kerman, Iran

[2] Faculty of Mathematical Sciences, University of Guilan, Rasht, Iran

irrelevant due to the large numbers of individual words in text [35].
- In Twitter, trending topics keep changing over time, and thus the dimensionality of data is changed dynamically. When a new top topic appears, it may come with a set of new keywords, which usually serve as key features to identify new hot topics [44].

A rudimentary approach in these scenarios is to wait a long time for all features to become available and then carry out the feature selection process. This approach is infeasible for most of the OSF scenarios. Another approach is to take the set of all features seen at each time step and then apply an standard feature selection technique, starting afresh each time. However, this approach is very inefficient, especially when the set of features only increases by one every time step. A more efficient and rational approach is to design an online streaming feature selection (OSFS) method which selects a best subset from so far seen features and updates it on the fly whenever new features stream in.

There are also some scenarios where the entire feature space is accessible, but feature streaming offers many advantages. In deep learning, the recently widely used method in machine learning, thousands (and or millions) of features could be generated by the network. Exhaustive searching over such a large feature space is very expensive or even infeasible. OSF can be considered as an integral part of such deep networks. Many emerging applications today, such as social media services, high-resolution images and document analysis, consume data of extremely high dimensionality [43, 45]. For example, the educational data mining data set from KDD CUP 2010 has about 29 million features. Therefore, scalability of feature selection algorithms is a must in such scenarios. Traditional feature selection algorithms need to access the entire feature set on the training data and perform a global search for the best feature at each round. Accordingly, batch methods cannot be highly scalable for high-dimensional data applications and online feature selection algorithms will be required [46].

OSFS is a less studied subject as it is a new problem in the era of big data. However, we believe that with the fast growing data dimensionality, OSFS can be considered as an important candidate for huge data pre-processing. Any OSFS method should satisfy three critical conditions [10]: first, it should not require any domain knowledge about feature space, because the full feature space is unknown or inaccessible. Second, it should allow efficient incremental updates in selected features, specifically when we have a limited amount of computational time available in between each feature arrival. Third, it should be as accurate as possible at each time instance.

Motivated by these challenges, several research efforts have been made to address OSFS. Perkins and Theiler proposed an online grafting algorithm for this problem, which treats the feature selection task as part of a regularized risk minimization problem [29]. An extension of this algorithm is adopted in [12] for edge detection. While the online grafting algorithm is able to handle streaming features, choosing a suitable threshold requires information about the global feature space. Moreover, this algorithm suffers from the so-called nesting effect [30]. Ungar et al. [35] proposed a streamwise regression algorithm, called information-investing. In this algorithm, a newly generated feature is added to the model if the entropy reduction is greater than the cost of the feature coding. Zhou et al. [47] proposed α -investing, a very similar algorithm to information-investing, which uses the p value of the generated feature as a criterion for adding it to the model. Similar to online grafting, these algorithms suffer from the nesting effect. The fast-OSFS, proposed by Wu et al. [40], is the first algorithm that tries to satisfy all the OSFS critical conditions. This algorithm contains two major steps: (1) online relevance analysis that discards irrelevant features and (2) online redundancy analysis, which eliminates redundant features. Although successful in selecting most informative features and avoiding nesting effect, fast-OSFS uses conditional independence tests which need a large number of training instances, especially when the number of features contributed in test grows with time. Therefore, adopting this algorithm on data sets with limited number of instances does not generate reliable results.

Rough set (RS) theory, introduced by Pawlak [28], is a growing mathematical tool to express information in data by means of boundary region of a set. The main advantage of this tool is that it requires no human input or domain knowledge other than the given data set [15, 16, 27, 36]. This property makes the RS theory an ideal candidate for OSFS. Wang et al. [37] proposed a dimension incremental attribute reduction algorithm called DIA-RED. This algorithm maintains a RS-based entropy value of the current selected subsets and updates this value whenever new conditional features are added to the data set. While DIA-RED is able to handle streaming scenarios, experiments in [10] show that this algorithm is not applicable effectively to real-world data sets. Eskandari and Javidi [10] proposed OS-NRRSAR-SA algorithm, which adopts the classical RS-based feature significance concept to eliminate irrelevant features in OSF scenarios. To significance analysis, we need to generate elementary subsets based on all the selected features. This causes a computational problem when the size of the selected subsets is not small enough during features streaming. This paper presents a method which based on the initial work in [10], filters out redundant features before significance analysis. In this regard, a redundancy analysis method based on functional dependency concept is proposed. The result is a general OSFS framework containing two major steps: (1) online redundancy analysis that discards redundant features

and (2) online significance analysis, which eliminates non-significant features.

The remainder of this paper is structured as follows: Sect. 2 summarizes the theoretical background and ideas of RS along with a look at functional dependency concepts. Section 3 discusses the proposed OSFS framework and presents a new OSFS algorithm, called OSFS-MRMS. Section 4 reports experimental results, and Sect. 5 concludes the paper.

2 Rough set

Rough set theory, introduced by Pawlak [28], proposes a mathematical approach to express vagueness by means of boundary region of a set. The main advantage of this implementation of vagueness is that it requires no human input or domain knowledge other than the given dataset [27, 36]. This section describes the fundamentals of the theory.

2.1 Information system and indiscernibility

An information system is a pair $\mathrm{IS} = (U, F)$, where U is a non-empty finite set of objects called the universe and F is a non-empty finite set of features such that $f : U \to V_f$, for every $f \in F$. The set V_f is called the value set or domain of f. A decision system is an information system of the form $\mathrm{IS} = (U, F, d)$, where d is called the decision feature.

For any set $B \subseteq F \cup \{d\}$, the B-indiscernibility relation is defined as:

$$\mathrm{IND}_{\mathrm{IS}}(B) = \{(x, y) \in U \times U | \forall f \in B, f(x) = f(y)\} \quad (1)$$

If (x, y) belongs to $\mathrm{IND}_{\mathrm{IS}}(B)$, x and y are said to be indiscernible according to the feature subset B. Equivalence classes of the relation $\mathrm{IND}_{\mathrm{IS}}(B)$ are denoted $[x]_B$ and referred to as B-elementary sets. The partitioning of U into B-elementary subsets is denoted by $U/\mathrm{IND}_{\mathrm{IS}}(B)$ or simply U/B. Generating such a partition is a common computational routine that affects the performance of any rough set-based operation. The general procedure PARTITION to compute U/B is displayed in Fig. 1.

The time complexity of PARTITION is $\Theta(|B||P||U|)$, where $|P|$ is the number of generated B-elementary subsets. If none of the objects in U are indiscernible according to B, the number of B-elementary subsets is $|U|$ and therefore the worst-case complexity of PARTITION is $O(|B||U|^2)$. Figure 2, from [10], demonstrates the ratio $|P|/|U|$ from application viewpoint. The figure on the left (a) shows the effect(s) of the number of features and instances on the number of generated partitions. The datasets for this figure have 30 uniformly distributed binary features. The figure shows the fast decrease in the ratio $|P|/|U|$, once $|B|$ becomes smaller than a threshold. The threshold is different for each data set and the larger the $|U|$ the larger the threshold. The figure on the right shows the effects of data sparseness (bias of feature values to a special value) on $|P|$. The data sets for this figure are all binary with 30 features and 1000 instances, but different in terms of sparseness (sparseness of 50% means uniformly distributed feature values). As it can be seen from this figure, the more sparse the data set, the higher the possibility of the objects become indiscernible and therefore the ratio $|P|/|U|$ is significantly small, even for large values of $|B|$.

PARTITION(U, B)
U: the universe of objects
B: a subset of features

```
1: P ← {}
2: for each x in U do
3:     [x]_B ← {x}
4:     for each y in U − {x} do
5:         if ( x(B) = y(B))
6:             [x]_B ← [x]_B ∪ {y}
7:             U ← U − {y}
8:         end if
9:     end for
10:    P ← P ∪ {[x]_B}
11: end for
12: return P
```

Fig. 1 The partitioning algorithm to generate elementary subsets

2.2 Lower and upper approximations

Two fundamental concepts of rough set are the lower and upper approximations of sets. Let $B \subseteq F$ and $X \subseteq U$, the B-lower and B-upper approximations of X are defined as follows:

$$\underline{B}X = \{x | [x]_B \subseteq X\} \quad (2)$$

$$\overline{B}X = \{x | [x]_B \cap X \neq 0\} \quad (3)$$

The $\underline{B}X$ and $\overline{B}X$ approximations define information contained in B [27]. If $x \in \underline{B}X$, it certainly belongs to X, but if $x \in \overline{B}X$, it may or may not belong to X.

By the definition of $\underline{B}X$ and $\overline{B}X$, the objects in U can be partitioned into three parts, called the positive, boundary and negative regions.

$$\mathrm{POS}_B(X) = \underline{B}X \quad (4)$$

$$\mathrm{BND}_B(X) = \overline{B}X - \underline{B}X \quad (5)$$

$$\mathrm{NEG}_B(X) = U - \overline{B}X \quad (6)$$

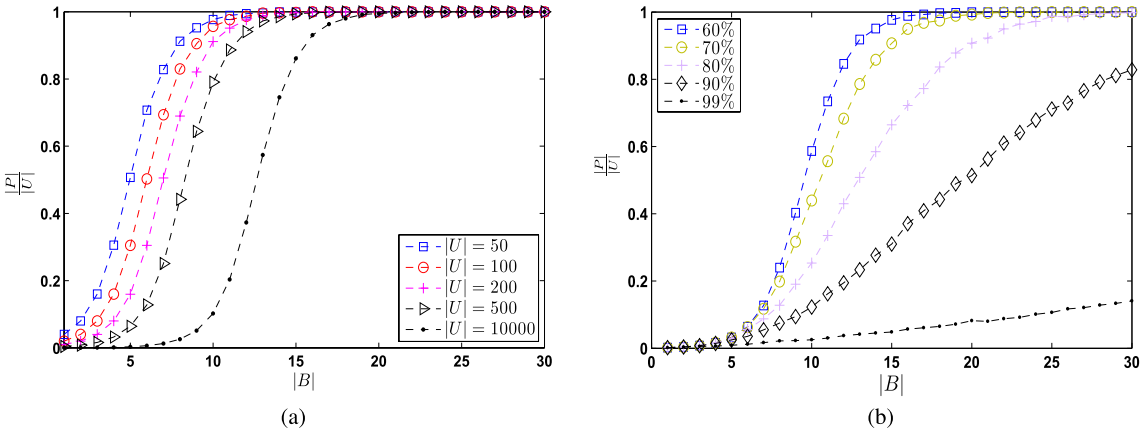

Fig. 2 **a** The effects of $|B|$ and $|U|$ on $|P|$, **b** the effects of sparseness on $|P|$

2.3 Dependency

Discovering dependencies between attributes is an important issue in data analysis. Let D and C be subsets of $F \cup \{d\}$. For $0 \leq k \leq 1$, it is said that D depends on C in the kth degree (denoted $C \Rightarrow_k D$), if

$$k = \gamma(C, D) = \frac{|\text{POS}_C(D)|}{|U|}, \tag{7}$$

where

$$\text{POS}_C(D) = \bigcup_{X \in U/D} \underline{C}X$$

is called a positive region of the partition U/D with respect to C. This region is the set of all elements of U that can be uniquely classified to blocks of the partition U/D, by means of C.

The rough functional dependency of D and C ($C \Rightarrow D$) is an special case of dependency where $\gamma(C, D) = 1$. In this case, we say that all values of attributes from D are uniquely determined by the values of attributes from C. The rough functional dependencies satisfy Armstrong's axioms [1]. Let X, Y, Z and W be arbitrary subsets of attributes, and the rough functional dependency has the following properties:

1. *Reflexivity* If $Y \subseteq X$, then $X \Rightarrow Y$
2. *Augmentation* If $Z \subseteq W$ and $X \Rightarrow Y$, then $X \cup W \Rightarrow Y \cup Z$
3. *Transitivity* If $X \Rightarrow Y$ and $Y \Rightarrow Z$, then $X \Rightarrow Z$
4. *Pseudo-transitivity* If $X \Rightarrow Y$ and $Y \cup W \Rightarrow Z$, then $X \cup W \Rightarrow Z$
5. *Union* If $X \Rightarrow Y$ and $X \Rightarrow Z$, then $X \Rightarrow Y \cup Z$
6. *Decomposition* If $X \Rightarrow Y \cup Z$, then $X \Rightarrow Y$ and $X \Rightarrow Z$

2.4 Reduct

Two different definitions are introduced for the reduct concept in the literature: (1) the indiscernibility relation preserving definition and (2) the dependency preserving definition. The former defines a reduct for a given information system $\text{IS}(U, C)$ (or decision system $\text{DS}(U, C, D)$) as a minimal set of attributes $R \subseteq C$ such that $\text{IND}_{\text{IS}}(R) = \text{IND}_{\text{IS}}(C)$. The later, on the other hand, defines a reduct for a given decision system $\text{DS}(U, C, D)$ as a minimal set of attributes $R \subseteq C$ such that $\gamma(R, D) = \gamma(C, D)$. In our work, the later is considered as a base for reduct analysis.

An optimal reduct is a reduct with minimum cardinality. The intersection of all reducts contains those attributes that cannot be eliminated and is called the core. Finding a minimal reduct is NP-hard [34], because all possible subsets of conditional features must be generated to retrieve such a reduct. Therefore, finding a near optimal has generated much of interest [17, 18, 21].

2.5 Rough set extensions

Traditional rough set-based attribute reduction (RSAR) has three shortcomings which make it ineffective in real-world applications [20, 21, 27]. Firstly, it only operates effectively with data sets containing discrete values and therefore it is necessary to perform a discretization step for real-valued attributes, secondly, RSAR is highly sensitive to noisy data, and finally, RSAR methods examine only the information contained within the lower approximation of a set ignoring the information contained in the boundary region.

Several extensions to the original theory have been proposed to overcome such shortcomings. Three well-known extensions are variable precision rough set (VPRS) [48], tolerance rough set model (TRSM) [33], fuzzy rough set

(FRS) [8, 20], decision-theoretic rough set [42], and game-theoretic rough set [14].

In addition to rough set extensions, there are also some modifications, which do not change classical rough set principles. The dependency notion in classical rough set is redefined in [27] and [16] to deal with useful information that may be contained in the boundary region.

3 The proposed OSFS framework

In this section, we first define online streaming features. Then we review notations of feature significance and feature redundancy and make a theorem to deal with feature redundancy in streaming features. Finally, propose a general framework to implement the significance and redundancy concepts for feature selection with streaming features.

Suppose that $\mathrm{DS}_t = (A_t, F_t, d)$ is a decision system at time t where $A_t = \{x_{1_t}, x_{2_t}, \ldots, x_{N_t}\}$, $F_t = \{f_1, f_2, \ldots, f_{M_t}\}$ and d is a decision feature. In online streaming features (OSF), new conditional features flow in one by one over time, while the number of objects in A remains fixed. In other words, for every time $t' > t$, $M_{t'} \geq M_t$ while $N_{t'} = N_t$.

Because we do not have access to the full feature space in the online streaming features context, we need to gradually build a reduct over time based on features seen so far. A rudimentary approach is to take the set of all features seen at each time step and then apply an standard traditional feature selection technique, starting afresh each time. However, a more rational approach is to design an algorithm which keeps a best subset from so far seen features and updates it on the fly whenever new features stream in. Here, we will review notions of feature significance and then define the feature redundancy concept. We will use the significance and redundancy notions to propose a general framework to update selected subset in OSF. In the definitions below, $\mathrm{DS} = (A, F, d)$ represents a decision system, where A, F, and d represent the universe, the full set of conditional features, and the decision feature, respectively. Moreover, $F - \{f\}$ represents the feature subset excluding the single feature f.

Definition 1 (*non-significant feature* [34]) A feature $f \in F$ is a non-significant feature for $\mathrm{DS} = (A, F, d)$ iff

$$\sigma_{(F,d)}(f) = \frac{\gamma(F,d) - \gamma(F - \{f\}, d)}{\gamma(F,d)} = 0 \tag{8}$$

Definition 2 (*non-significant feature subset* [34]) A feature subset $F' \subseteq F$ is a non-significant feature subset for $\mathrm{DS} = (A, F, d)$ iff

$$\sigma_{(F,d)}(F') = \frac{\gamma(F,d) - \gamma(F - F', d)}{\gamma(F,d)} = 0 \tag{9}$$

Significance analysis is a tool for measuring the effect of removing an attribute, or a subset of attributes, from a decision system on the positive region defined by that decision system. The more the significance of an attribute (set), the higher the change in dependency is. If the significance is 0, then the attribute (set) is dispensable and can be eliminated from the decision system.

Definition 3 (*redundant feature*) A feature $f \in F$ is a redundant feature for $\mathrm{DS} = (A, F, d)$ iff $\exists F' \subseteq F - \{f\}$ s.t. $F' \Rightarrow f$, otherwise it is non-redundant.

Definition 4 (*redundant feature subset*) A feature subset $C \subset F$ is a redundant subset for $\mathrm{DS} = (A, F, d)$ iff $\exists F' \subseteq F - C$ s.t. $F' \Rightarrow C$, otherwise it is non-redundant.

Redundant features can be completely described using some other features in the conditional feature set, and therefore they can be eliminated without loosing any useful information.

By Definitions 1–4, we propose an OSFS framework that contains two major steps: (1) online redundancy analysis that discards redundant features, and (2) online significance analysis, which eliminates non-significant features from the features selected so far (see Fig. 3).

4 The proposed OSFS algorithm

Algorithm 1 represents the proposed algorithm which implements the OSFS framework. The algorithm starts with an empty selected subset R. Then it waits for a new incoming feature (line 3). Once a new feature f is provided, the algorithm tests the consistency of the current decision system. If it is not consistent, the first phase of the algorithm triggers. This phase calculates two values; (1) the increase of the dependency value, when f is added to current subset, and (2) the noise-resistent dependency of d on f. If at least one

1: **Initialization**
   Selected feature subset $R = \{\}$
2: **Online Streaming Feature**
   a) If a new feature f is available, then get f
   b) Otherwise, wait for a new feature.
3: **Online Redundancy Analysis**
   Discard feature subsets in R, became redundant due to the presence of f, by Definitions 3 and 4.
4: **Online Significance Analysis**
   Discard feature subsets in R, became non-significant due to the presence of f by definitions 1 and 2.
5: Alternate steps 2-4 until a stopping criteria is satisfied.
6: Output the selected subset R.

Fig. 3 The proposed OSFS framework

of these values is nonzero, the current subset is updated to include f (line 6); otherwise, f is simply rejected. However, if the current decision system is consistent before arriving f, the second and third phases are triggered, respectively. In the second phase (lines 9–17), the algorithm checks to see if there exists any current reduct subset, which becomes redundant due to the presence of f. If such subset exists, and its size is larger than one, then the subset can be replaced with f (lines 12–14). Moreover, if only one feature (say f') becomes redundant due to f, then one of the features f and f' is removed based on the noise-resistant measure value (lines 28–30). The third phase (lines 18–25) removes the non-significant features, with a methodology similar to the second phase.

Different stopping criterions can be adopted to control the algorithm execution. If the size of the streaming data set is known, the algorithm can keep running to see the last feature. (No further features are available.) However if we have no knowledge about the feature space (including maximum number of features), then the algorithm can stop once a pre-defined accuracy is satisfied or a maximum number of iterations is reached.

Algorithm 1 OSFS-MRMS(d)

```
    d: The decision feature
 1: R = ∅
 2: while stopping criterion is not met do
 3:    f = GET-NEW-FEATURE()

    %First Phase: The Consistency Test%
 4:    if γ(R,d) ≠ 1 then
 5:       if (γ(R∪{f},d) − γ(R) > 0) ∨ (ρ({f},d) > 0)) then
 6:          R = R∪{f}
 7:       end if
 8:    else
    %Second Phase: The Redundancy Analysis%
 9:       added = false
10:       R1 = R∪{f}
11:       A = REDUNDANT(R1, f, k)
12:       if |A| > 1 then
13:          R1 = R1 − A
14:          added = true
15:       else if |A| = 1 then
16:          X = A
17:       end if
    %Third Phase: The Significance Analysis%
18:       B = NON-SIGNIFICANT(R1, f, d)
19:       if |B| > 1 then
20:          R1 = R1 − B
21:          added = true
22:       else if |B| = 1 then
23:          Y = B
24:       end if
25:    end if

26:    if added then
27:       R = R1
28:    else if arg min_{z∈X∪Y∪{f}} (ρ({z},d)) ≠ f then
29:       R = R1 − X − Y
30:    end if
31: end while
```

4.1 The REDUNDANT routine in OSFS

Procedure 2 uses the notation REDUNDANT routine to identify features, which became redundant due to the new incoming feature. Several implementations of the routine can be adopted based on the relative importance of the reduct size compared with the time required to locate the redundant features. Finding the redundant subset with maximum size is an expensive task, because we need to consider all feature subsets, and for each subset we need to investigate its redundancy using other subsets. Algorithm 3 represents an efficient sequential backward elimination procedure to find features that became redundant due to presence of a new incoming feature f.

Algorithm 2 REDUNDANT(R,f,k)

```
    R: The feature set
    f: The newly added feature
    k: The maximum subset size
 1: B = {}
 2: T = R − {f}
 3: while |T| ≠ 0 do
 4:    g = RANDOM{f_i ∈ T, i = 1,2,...,|T|}
 5:    size ← 0
 6:    isRedundant = 0
 7:    while size < k and !isRedundant do
 8:       size = size + 1
 9:       for Each S ⊆ R − {f} − {g} s.t. |S| = size do
10:          if γ(S∪{f},g) = 1 then
11:             isRedundant = 1
12:          end if
13:       end for
14:    end while
15:    if isRedundant then
16:       B = B∪{g}
17:       R = R − {g}
18:    end if
19:    T ← T − {g}
20: end while
21:
22: return B
```

At each step, the algorithm considers a random feature g that has not already been evaluated and drops the feature out if it is redundant based on available features. Testing the redundancy of g consists of finding a subset S of already non-eliminated features, such that $(S \cup \{f\}) \Rightarrow g$. The algorithm uses a bottom-up process to find such a subset. Thus, it considers subsets of size one in the first step, subsets of size two in the next step, and so on. The maximum subset size is controlled by the parameter k. The larger the value of this parameter, the algorithm is more successful in locating redundant features, but on the other hand, the time complexity of the algorithm is greater. Therefore, a trade-off is needed between the success of the algorithm and its time complexity. Our empirical studies show that even small values of k, such as 3 or 4, yield satisfactory results.

4.2 The NON-SIGNIFICANT routine in OSFS

As the REDUNDANT routine, several implementations can be adopted for NON-SIGNIFICANT routine, based on the relative importance of the reduct size compared with the time required to locate the non-significant features. Algorithm 4.2, which represents a very efficient routine, is proposed in our previous work [10], which uses a sequential backward elimination mechanism.

Algorithm 3 NON-SIGNIFICANT(R,f,d)

```
R: The feature set
f: The newly added feature
d: The decision feature
B = {}
T = R − {f}
while |T| ≠ 0 do
  g = RANDOM{f_i ∈ T, i = 1,2,...,|T|}
  if σ_(R,d)(g) = 0 then
    B = B ∪ {g}
    R = R − {g}
  end if
  T = T − {g}
end while

return B
```

Starting from full reduct, at each step the method considers a random feature that has not already been evaluated and drops the feature out if it is non-significant based on available feature subset. Because of random consideration of features, different executions of this method may return different non-significant subsets and therefore a good heuristic would be to executing the method k times and selecting the result with maximum size. Experiments in [10] show that $k = 3$ causes satisfactory results.

4.3 Some properties of selected subset by OSFS

Let $F_t = \{f_1, f_2, \ldots, f_{M_t}\}$ be the set of features that have arrived until the time t, such that f_i arrives before f_j if and only if $i < j$. Here, we prove two important theorems about our proposed OSFS algorithm. The first theorem is about the consistency-preserving property of the proposed algorithm. If the decision system becomes consistent at a time t, it will remain consistent at any time $t' > t$. The second theorem gives a better insight about the selected subsets size changes during the time.

For convenience, we list some important mathematic notations that are employed in this paper in Table 1.

Lemma 1 (monotonicity of γ [19]) *Suppose that $R \subseteq F$ is a subset of conditional attributes, $f \in F$ is an arbitrary conditional attribute, and d is the decision attribute. Then $\gamma(R \cup \{f\}, d) \geq \gamma(R, d)$.*

The following lemma implies that removing a redundant feature from a consistent decision system preserves the consistency of that decision system.

Lemma 2 *Let* DS = *(A, F, d) be a consistent decision system ($\gamma(F, d) = 1$) and $G \subset F$ be a redundant attribute set. Then $\gamma(F - G, d) = 1$.*

Proof Based on definition of the functional dependency, $F \Rightarrow d$. Because G is a redundant subset for DS, $\exists F' \subseteq F - G$ s.t. $F' \Rightarrow G$. Based on reflexivity property of

Table 1 List of important notations

Notation	Description	Notation	Description
IS	Information system	DS	Decision system
U	Set of all objects	F	Total feature s
d	Decision feature	f	A single feature
V_f	Value set or domain of feature f	$IND_{IS}(B)$	$IND_{IS}(B) = \{(x,y) \in U \times U \mid \forall f \in B,\ f(x) = f(y)\}$
$[x]_B$	$[x]_B = \{y \in U \mid (x,y) \in IND_{IS}(B)\}$	U/B	partitioning of U into B-elementary subsets
P	The set of all elementary subsets	$\underline{B}X$	$\underline{B}X = \{x \mid [x]_B \subseteq X\}$
$\bar{B}X$	$\bar{B}X = \{x \mid [x]_B \cap X \neq 0\}$	$POS_B(X)$	$POS_B(X) = \underline{B}X$
$BND_B(X)$	$BND_B(X) = \bar{B}X - \underline{B}X$	$NEG_B(X)$	$NEG_B(X) = U - \bar{B}X$
$\gamma(C, D)$	$\gamma(C,D) = \frac{\lvert POS_C(D)\rvert}{\lvert U\rvert}$	$C \Rightarrow D$	Rough functional dependency of D on C
x_t	The object x at time t	N_t	Number of objects at time t
A_t	$A_t = \{x_{1_t}, x_{2_t}, \ldots x_{N_t}\}$	M_t	Total number of features at time t
f_t	The feature that arrives at time t	F_t	$F_t = \{f_1, f_2, \ldots f_{M_t}\}$
$\rho(C, D)$	The noise resistance dependency of D on D	R_t	Selected subset at time t
$\sigma_{(F,d)}(f)$	$\sigma_{(F,d)}(f) = \frac{\gamma(F,d) - \gamma(F - \{f\}, d)}{\gamma(F,d)}$	$\sigma_{(F,d)}(F')$	$\sigma_{(F,d)}(F') = \frac{\gamma(F,d) - \gamma(F - F', d)}{\gamma(F,d)}$

the functional dependency, $F - G \Rightarrow F'$ and we can conclude that $F - G \Rightarrow G$, based on transitivity property. Now considering that $X = F - G, Y = G, W = F - G$ and $Z = \{d\}$, we can conclude from the pseudo-transitivity property that $F - G \Rightarrow d$ and therefore $\gamma(F - G, d) = 1$, which means that the decision system is consistent using $F - G$.□

Theorem 1 *Let R_t be the selected feature subset using OSFS at time t. IF $\gamma(R_t, d) = 1$, then $\forall t' \geq t, \gamma(R_{t'}, d) = 1$.*

Proof We prove this theorem by induction. For $t' = t$, the theorem holds by assumption. Let $\gamma(R_{t'=t+k}, d) = 1$ for a given $k \geq 1$. Suppose that a new feature f_{t+k+1} is streamed in at time $t' = t + k + 1$. Because the decision system is consistent using current selected feature subset R_{t+k}, the second and third phases of the algorithm will be triggered, respectively. The following cases can occur after the two phases:

1. $|A| > 1$ or $|B| > 1$: R_{t+k+1} will be one of the subsets (a) $R_{t+k} \cup \{f_{t+k+1}\} - A$, (b) $R_{t+k} \cup \{f_{t+k+1}\} - B$, and (c) $R_{t+k} \cup \{f_{t+k+1}\} - A - B$. Firstly, $R_{t+k} \cup \{f_{t+k+1}\}$ will remain consistent according to Lemma 1. Secondly, A is a redundant subset for $R_{t+k} \cup \{f_{t+k+1}\}$; therefore, $R_{t+k} \cup \{f_{t+k+1}\} - A$ will remain consistent according to Lemma 2. Finally, B is a non-significant feature subset for $R_{t+k} \cup \{f_{t+k+1}\}$ (or $R_{t+k} \cup \{f_{t+k+1}\} - A$). That is

$$\sigma_{(R_{t+k} \cup \{f_{t+k+1}\}, d)}(B) = \frac{\gamma\left(R_{t+k} \cup \{f_{t+k+1}\}, d\right) - \gamma\left(\left(R_{t+k} \cup \{f_{t+k+1}\}\right) - B, d\right)}{\gamma\left(R_{t+k} \cup \{f_{t+k+1}\}, d\right)} = 0$$

$$(\text{or } \sigma_{(R_{t+k} \cup \{f_{t+k+1}\} - A, d)}(B) = \frac{\gamma\left(R_{t+k} \cup \{f_{t+k+1}\} - A, d\right) - \gamma\left(\left(R_{t+k} \cup \{f_{t+k+1}\} - A\right) - B, d\right)}{\gamma\left(R_{t+k} \cup \{f_{t+k+1}\} - A, d\right)} = 0)$$

then

$$\gamma((R_{t+k} \cup \{f_{t+k+1}\}) - B, d) = \gamma(R_{t+k} \cup \{f_{t+k+1}\}, d) = 1$$

$$(\text{or } \gamma((R_{t+k} \cup \{f_{t+k+1}\} - A) - B, d) = \gamma(R_{t+k} \cup \{f_{t+k+1}\} - A, d) = 1)$$

2. $|A| \leq 1$ and $|B| \leq 1$: R_{t+k+1} will be one of the two subsets (a) R_{t+k} and (b) $R_{t+k} \cup \{f_{t+k+1}\} - X - Y$. In either case, it is obvious that $\gamma(R_{t+k+1}, d) = 1$.

□

Theorem 2 *Let R_t be the selected feature subset using OSFS algorithm at time t.*

(a) *If $\gamma(R_t, d) < 1$, then $\forall t' < t, |R_{t'}| \leq |R_t|$,*
(b) *If $\gamma(R_t, d) = 1$, then $\forall t' \geq t, |R_{t'}| \leq |R_t|$.*

Proof We prove (a) by contradiction. Suppose that $\exists t' < t$, such that $|R_{t'}| > |R_t|$. Then it is obvious that $R_{t'} \not\subset R_t$. Therefore, $\exists a \in R_{t'}$ s.t $a \notin R_t$ and hence $\exists t_1, t \leq t_1 < t'$, when a is removed from selected subset. Removing feature(s) from selected subset only occurs during the second phase of the algorithm, and this phase triggers if the decision system is consistent using selected subset. Therefore, $\gamma(R_{t_1}, d) = 1$. However, by Theorem 1, we have $\gamma(R_t, d) = 1$, which is a contradiction.□

In order to prove (b), we use induction. For $t' = t$, the theorem holds by assumption. Let $|R_{t+k}| \leq |R_t|$ for a given $k \geq 1$. Suppose that a new feature f_{t+k+1} is streamed in at time $t + k + 1$. Based on Theorem 1, the decision system is consistent using R_{t+k} and therefore the second and third phases of the algorithm will be triggered, respectively. As shown in proof of Theorem 1, R_{t+k+1} can be one of the subsets (a) $R_{t+k} \cup \{f_{t+k+1}\} - A$, (b) $R_{t+k} \cup \{f_{t+k+1}\} - B$, (c) $R_{t+k} \cup \{f_{t+k+1}\} - A - B$, (d) $R_{t+k} \cup \{f_{t+k+1}\} - X - Y$, and (e) R_{t+k}. Given that $\forall I \in \{A, B, X, Y\}$, $|I| \geq 1$, then $|R_{t+k+1}| \leq |R_{t+k}|$.

Let $F = \{f_1, f_2, \ldots, f_M\}$ be the set of features that have arrived so far. Assume that the data set has been non-consistent for the $|M_1|$ first incoming features and consistent for the remaining $|M_2|$ features, where $|M_1| + |M_2| = |M|$. The size of the selected subsets constitutes a sequence over time, that, starting from the first element (the size of the first subset), we will encounter elements in non-decreasing order until we reach the maximum element in the list, after which we will encounter elements in non-increasing order. The selected subset with maximum size is located after arriving f_{M_1} (based on Theorem 2).

4.4 The time complexity of OSFS

The time complexity of OSFS depends on the number of tests. Two types of tests are used in the algorithm: the γ-tests and the ρ-tests. As stated previously, the time required by this RS-based test can be attributed by the time that is required to generate equivalence classes (the PARTITION

algorithm). Suppose that at time t a new feature f_t be present to the OS-NRRSAR-RA algorithm and let R_t be the selected feature subset at this time. If the available decision system is not consistent using R_t, the first phase of the algorithm will be triggered. This phase includes constant number of ρ and γ tests. Therefore, the worst-case time complexity of this phase is $\Theta(|R_t||U|^2)$. However, if the data set is consistent, the redundancy and significance analysis phases will be triggered, respectively. The redundancy analysis phase (second phase) uses REDUNDANT routine in Fig. 2. This routine tests the redundancy of all features in R_t and the maximum number of subsets that are considered for each test is $\binom{|R_t|}{1} + \binom{|R_t|}{2} + \cdots + \binom{|R_t|}{k}$. Therefore, the worst-case time complexity of this phase is $|R_t|\binom{|R_t|}{1} + \binom{|R_t|}{2} + \cdots + \binom{|R_t|}{k}|U^2|$. On the other hand, the significance analysis phase (third phase) uses NON-SIGNIFICANT routine in Fig. 3, which has the worst-case time complexity of $O(|R_t|^2|U|^2)$.

Let $F = \{f_1, f_2, \ldots, f_M\}$ be the set of features that have arrived so far. As stated, The selected subset with maximum size is located after arriving f_{M_1}, and therefore, the worst-case time complexity of OSFS is

$$O\big((|M_1||R_{M_1}||U|^2) + (|M_2||R_{M_1}|\Theta_1|U|^2) + (|M_2||R_{M_1}|^2|U|^2)\big)$$

where

$$\Theta_1 = \binom{|R_{M-1}|}{1} + \binom{|R_{M-1}|}{2} + \cdots + \binom{|R_{M-1}|}{k}$$

Although the worst-case time complexity of the proposed algorithm is square with respect to the number of selected features, in many real-world applications, only a small number of features in a large feature space are predictive and relevant to decision feature [40]. Therefore $|R_t|$ (and hence $|R_{M_1}|$) is so small that its square does not affect the time complexity of the OSFS algorithm, significantly. Being squared with respect to $|U|$ (number of training instances) is because of the fact that we considered the worst-case time complexity of the PARTITION algorithm, for analysing the complexity of the proposed algorithm. However, as stated in Sect. 2.1, the time complexity of PARTITION routine tends to be linear, when the number of participating features is small. OSFS keeps the $|R_t|$ (number of selected features at time t) very small and therefore all the PARTITION calls (in γ and ρ tests) will be executed with small feature subsets. Therefore, $|P| << |U|$ and the PARTITION algorithm will be very time efficient. Moreover, most of the real-world large scale data sets are highly sparse, which means even faster executions of the PARTITION calls.

5 Experimental results

In this section, we show the performance of the proposed method. To do this, the proposed OSFS algorithm is compared with OS-NRRSAR-SA [10]. Table 2 summarizes the 12 high-dimensional data sets used in our experiments. For VOC 2007, which is an image classification data set, we extracted convolutional neural network (CNN)-based features from the penultimate layer of the VGG-VD [32] (4096 features) deep network. In order to provide OSF scenario, features are considered one by one. All the experiments are carried out on a DELL workstation with Windows 7, 2 GB memory and 2.4 GHz CPU. Two classifiers are employed for the classification of the data, J48 [31, 39] and kernel SVM with RBF kernel function [3]. For two class classification problems, average precision (AP %) is used as accuracy measure. For multi-class cases, we used the mean of the APs (mAP %) on different classes. In all the experiments, the maximum subset size (k) in REDUNDANT routine is set to be 3.

Because we do not have access to the full feature space, the streaming order of the features affects the final results. Therefore, in order to strengthen the comparison, the results are averaged over 30 different pre-generated random streaming orders for each data set.

5.1 Compactness

The selected subsets sizes during the features streaming are reported in Fig. 4. As it can be seen, the proposed OSFS-MRMS results in more compact reducts in most of the cases. Moreover, considering the selected subsets at the end of the streaming (100% of the features seen), this algorithm outperforms the OS-NRRSAR-SA for all

Table 2 Summary of the benchmark high-dimensional data sets

Data set	# Attributes	# Train	# Test	Type	Source
dorothea	100000	800	800	*C*	[5]
arcene	10000	100	700	*I*	[5]
dexter	20000	300	2000	*I*	[5]
madelon	500	2000	1800	*C*	[5]
sido0	4932	12678	10000	*C*	[7]
cina0	132	16033	10000	*I*, *C*	[7]
nova	16969	1754	17537	*C*	[6]
sylva	216	13086	130854	*I*, *C*	[6]
hiva	1617	3845	38449	*C*	[6]
arrhythmia	279	452	–	*C*, *I*, *R*	[2]
mf	649	2000	–	*I*, *R*	[2]
VOC 2007	Not specified	5011	4952	*R*	[11]

C categorical, *R* real and *I* integer

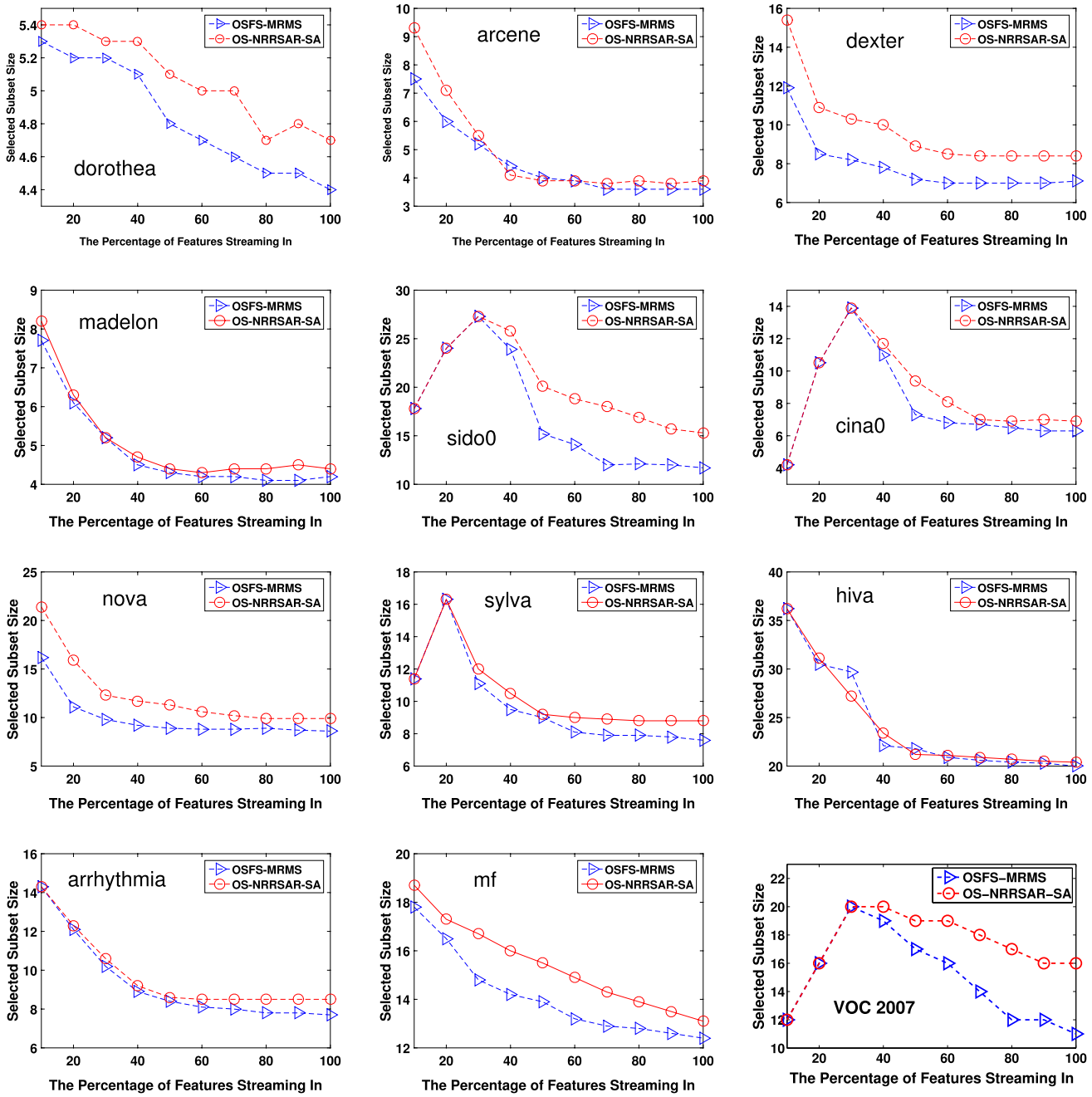

Fig. 4 Selected subsets size during features streaming

the data sets. The two algorithms are the same when the decision system is not consistent. Therefore, we expect the same selected subsets at the early stages of the streaming. We can see this phenomenon for *sido0*, *cina0* and *sylva* data sets, which did not become consistent using less then 10% of the streaming features.

5.2 Running time

Table 3 reports the running times of the two algorithms at the end of the features streaming. A hypothesis paired t test is carried out to compare the results on the 30 streaming orders. Let t_A, t_B be the set of running times of the methods A and B, respectively, for the 30 different streaming orders. We define the following two one-tailed t tests:

Table 3 Comparison of run times for OS-NRRSAR-SA and OSFS-MRMS

Data set	OS-NRRSAR-RA	OSFS-MRMS	t
dorothea	509.73	**486.92**	↓
arcene	82.28	**80.66**	=
dexter	572.75	**511.02**	↓
madelon	**87.03**	123.87	↑
sido0	961.99	**798.10**	↓
cina0	**62.82**	65.01	=
nova	214.84	**183.22**	↓
sylva	**782.09**	852.68	↑
hiva	**2311.89**	3027.84	↑
arrhythmia	123.10	**118.32**	=
mf	234.89	**163.01**	↓
voc 2007	3028	**2472**	↓

Dominant results are shown in bold

$$t_1 : \begin{cases} H_0 : \mu_{d_A} = \mu_{d_B} \\ H_1 : \mu_{d_A} > \mu_{d_B} \end{cases} \tag{10}$$

$$t_2 : \begin{cases} H_0 : \mu_{d_A} = \mu_{d_B} \\ H_1 : \mu_{d_B} > \mu_{d_A} \end{cases} \tag{11}$$

where μ_D is the population mean of set D.

Based on results of these tests, the variable t is defined as

$$t = \begin{cases} \uparrow & \text{if the null hypothesis } (H_0) \text{ in } t_1 \text{ is rejected} \\ \downarrow & \text{if the null hypothesis in } t_2 \text{ is rejected} \\ = & \text{if none of the null hypothesis in } t_1 \text{ and } t_2 \text{ is rejected} \end{cases} \tag{12}$$

We see that the proposed OSFS-MRMS is superior for six (*dorothea*, *dexter*, *sido0*, *nova*, *mf* and *voc 2007*) and inferior for three cases (*madelon*, *sylva* and *hiva*). The tests show that the mean running times of the two algorithms are not significantly different for *hiva* and *mf*. Although the redundancy analysis step in the proposed algorithm imposes an extra computational time, the smaller selected subsets during features streaming cause faster PARTITION routine executions for this algorithm.

5.3 Classification accuracy

The classification results, presented in Figs. 5 and 6, show that the proposed OSFS-MRMS algorithm performs very well and shows increase in classification accuracies for most of the tests. Compared with OS-NRRSAR-SA in terms of J48 classifier, OSFS-MRMS is superior in most of the cases except *dexter*, *sido0* and *mf*. The same comparison in terms of SVM classifier shows that our proposed algorithm won the tests for seven data sets *arcene*, *dexter*, *sido0*, *cina0*, *nova*, *sylva* and *hiva*. Table 4 reports the t test results on the classification accuracies of the two algorithms during features streaming.

According to the recorded accuracy values for each data set (10 measurements on 30 streaming orders), OSFS-MRMS outperforms the OS-NRRSAR-SA in 65 and 61% of the cases using J48 and SVM, respectively. Moreover, considering all the records, the average accuracy of the OSFS-MRMS is 2.28 and 2.16% higher, in terms of J48 and SVM, respectively.

6 Conclusions

This paper presented a method which based on the OS-NRRSAR-SA algorithm proposed in [10], filters out redundant features before significance analysis. In this regard, a redundancy analysis method based on functional dependency concept was proposed. The result was a general OSFS framework containing two major steps: (1) online redundancy analysis that discards redundant features and (2) online significance analysis, which eliminates non-significant features. To show the efficiency and accuracy of the proposed algorithm, it was compared with OS-NRRSAR-SA algorithm. Several high-dimensional data sets were used for comparisons, and their features considered one by one to simulate the true OSF scenarios. The compactness, running time and classification accuracy during the features streaming were the comparison terms. The experiments demonstrate that the proposed algorithm achieves better results than OS-NRRSAR-SA algorithm, for all evaluation terms.

The authors would like to propose the following subjects for future works on OSFS scenarios:

1. *OSFS with missing values* Feature vectors with missing values are common in remote sensing where incomplete data may occur when certain regions are covered by a subset of sensors. Data missing in clinical databases due to expense or difficulty of obtaining certain results, particularly when they are not routine clinical measurements, is another example. OSFS, where new incoming features have one or more missing values, can be considered as an important problem in dealing with OSFS problems.
2. *OSFS with streaming instances* In this paper, the main consideration was that the number of feature vectors is fixed. However, it is possible that a data set grows both in terms of number of features and instances.
3. *OSFS in deep learning* Deep learning is the recently widely used method in machine learning [22, 41]. We would have millions or billions features generated by

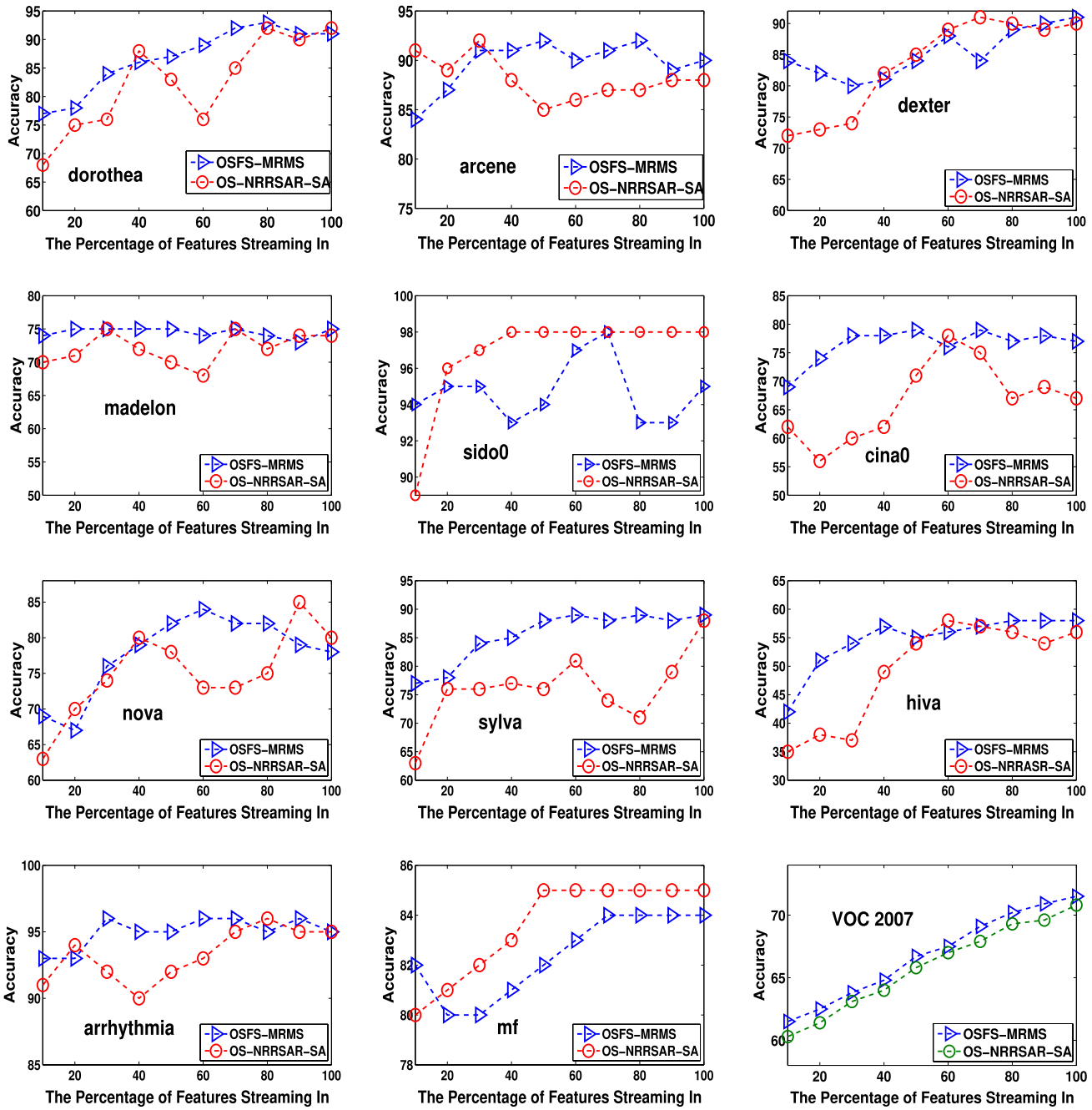

Fig. 5 J48 classification results of selected subsets during features streaming

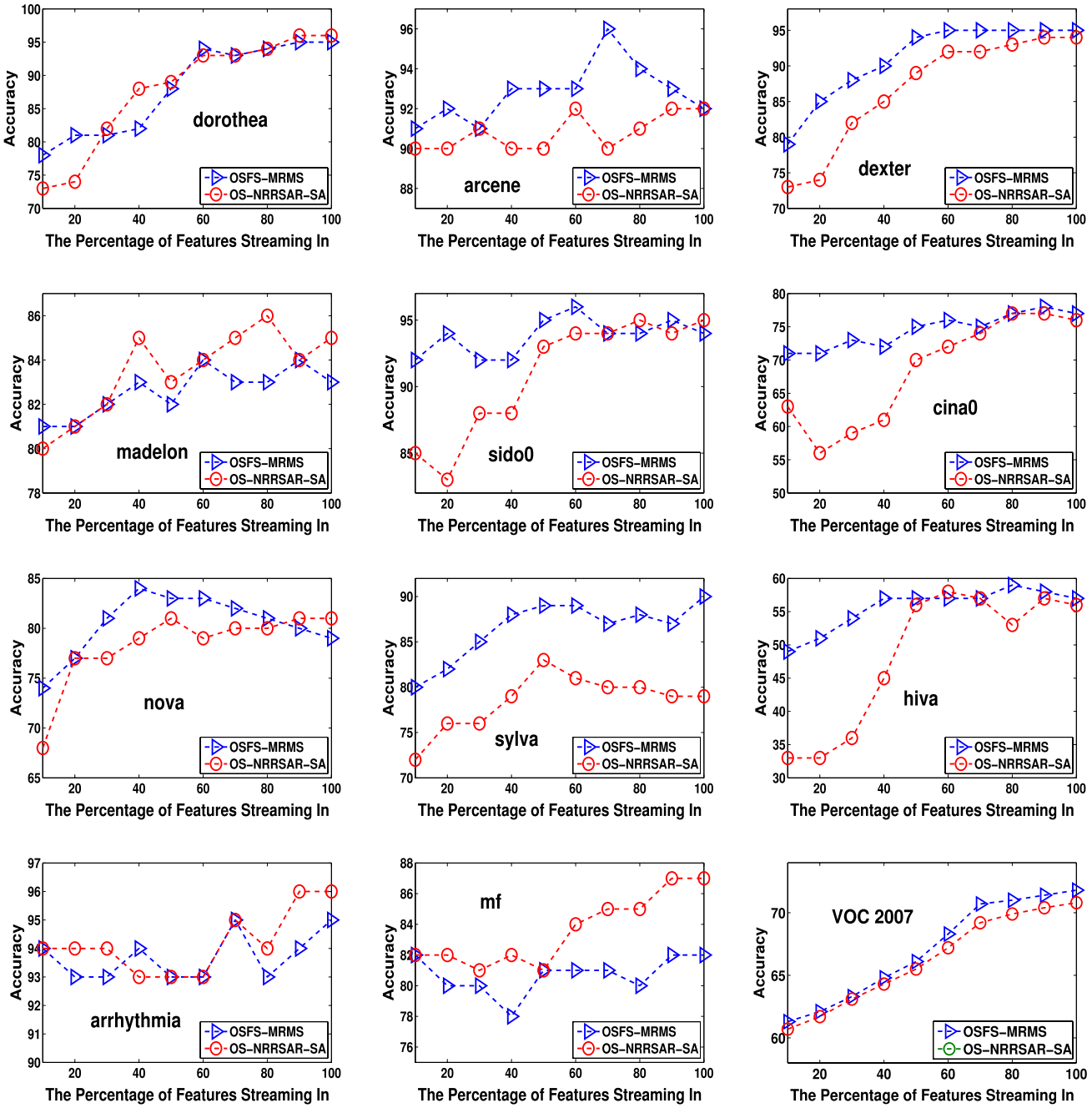

Fig. 6 SVM classification results of selected subsets during features streaming

Table 4 *t* test results on classification accuracies for OS-NRRSAR-SA and OSFS-MRMS during features streaming (J48/SVM)

Data set	The percentage of features streaming in									
	10	20	30	40	50	60	70	80	90	100
dorothea	↑/↑	↑/↑	↑/=	=/↓	↑/=	↑/=	↑/=	=/=	=/=	=/=
arcene	↓/↑	↓/↑	=/=	↑/↑	↑/↑	↑/↑	↑/↑	=/↑	=/↑	↑/=
dexter	↑/↑	↑/↑	↑/↑	=/↑	=/↑	=/↑	↓/↑	=/↑	=/=	=/=
madelon	↑/↑	↑/=	=/=	↑/↓	↑/↓	↑/=	=/↓	↑/↓	=/=	=/↓
sido0	↑/↑	↓/↑	↓/↑	↓/↑	↓/↑	↓/↑	=/=	↓/↓	↓/=	↓/↓
cina0	↑/↑	↑/↑	↑/↑	↑/↑	↑/↑	=/↑	↑/=	↑/=	↑/=	↑/=
nova	↑/↑	↓/=	=/↑	=/↑	↑/↑	↑/↑	↑/↑	↑/=	↓/=	=/↓
sylva	↑/↑	=/↑	↑/↑	↑/↑	↑/↑	↑/↑	↑/↑	↑/↑	↑/↑	=/↑
hiva	↑/↑	↑/↑	↑/↑	↑/↑	=/=	↓/=	=/=	↑/↑	↑/=	=/=
arrhythmia	↑/=	=/↓	↑/↓	↑/↑	↑/=	↑/=	=/=	↓/↓	↑/↓	=/↓
mf	↑/=	↓/↓	↓/↓	↓/↓	↓/=	↓/↓	↓/↓	↓/↓	↓/↓	↓/↓
voc 2007	↑/=	↑/=	=/=	=/=	=/=	=/=	↑/↑	↑/↑	↑/↑	↑/↑

deep networks. One approach would be to adopt FS to select most important features (weights) from a trained model [9]. This is the approach we adopted in this paper for PASCAL VOC data sets. Another approach would be to use OSFS as a construction or training part to reduce the tons of parameters for deep networks.

Acknowledgements The authors would like to thank professor Mahbano Tata for her comments that greatly improved the manuscript.

References

1. Beaubouef T, Petry F (2013) Incorporating rough data in database design for imprecise information representation. In: Skowron A, Suraj Z (eds) Rough sets and intelligent systems—Professor Zdzisław Pawlak in Memoriam, intelligent systems reference library, vol 43. Springer, Berlin, pp 137–155
2. Blake C, Merz CJ (1998) UCI Repository of machine learning databases. http://www.ics.uci.edu/mlearn/MLRepository.html. Accessed 06 March 2015
3. Chang CC, Lin CJ (2011) Libsvm: a library for support vector machines. ACM Trans Intell Syst Technol 2(3):27:1–27:27
4. Chao S, Cai J, Yang S, Wang S (2016) A clustering based feature selection method using feature information distance for text data. Springer, Cham, pp 122–132
5. Clopinet: feature selection challenge, NIPS 2003. http://clopinet.com/isabelle/Projects/NIPS2003/ (2003). Accessed 6 March 2015
6. Clopinet: performance prediction challenge, WCCI 2006. http://clopinet.com/isabelle/Projects/modelselect/ (2006). Accessed 6 March 2015
7. Clopinet: causation and prediction challenge, WCCI 2008. http://www.causality.inf.ethz.ch (2008). Accessed 6 March 2015
8. Dubois D, Prade H (1992) Putting rough sets and fuzzy sets together. In: SÅowiÅki R (ed) Intelligent decision support, theory and decision library, vol 11. Springer, Dordrecht, pp 203–232
9. Eskandari S, Akbas E (2017) Supervised infinite feature selection. CoRR abs/1704.02665. http://arxiv.org/abs/1704.02665
10. Eskandari S, Javidi M (2016) Online streaming feature selection using rough sets. Int J Approx Reason 69:35–57
11. Everingham M, Van Gool L, Williams CKI, Winn J, Zisserman A The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html
12. Glocer K, Eads D, Theiler J (2005) Online feature selection for pixel classification. In: Proceedings of the 22nd international conference on machine learning, Bonn, Germany
13. Gosztolya G, Tóth L (2017) A feature selection-based speaker clustering method for paralinguistic tasks. Pattern Anal Appl 1–12
14. Herbert JP, Yao J (2011) Game-theoretic rough sets. Fundam Inform 108(3–4):267–286
15. Javidi MM, Eskandari S (2016) Streamwise feature selection: a rough set method. Int J Mach Learn Cybern 1–10
16. Javidi MM, Eskandari S (2017) A noise resistant dependency measure for rough set-based feature selection. J Intell Fuzzy Syst 33(3):1–14
17. Jensen R, Shen Q (2001) A rough set-aided system for sorting www bookmarks. In: Proceedings of the first Asia-Pacific conference on web intelligence: research and development, WI'01, London, UK
18. Jensen R, Shen Q (2005) Fuzzy-rough data reduction with ant colony optimization. Fuzzy Sets Syst 149(1):5–20
19. Jensen R, Shen Q (2008) Computational intelligence and feature selection: rough and fuzzy approaches. Wiley, London
20. Jensen R (2004) Qiang Shen: semantics-preserving dimensionality reduction: rough and fuzzy-rough based approaches. IEEE Trans Knowl Data Eng 16(16):1457–1471
21. Jensen R, Tuson A, Shen Q (2014) Finding rough and fuzzy-rough set reducts with SAT. Inf Sci 255:100–120
22. Liu W, Zha ZJ, Wang Y, Lu K, Tao D (2016) *p*-Laplacian regularized sparse coding for human activity recognition. IEEE Trans Ind Electron 63(8):5120–5129
23. Li T, Ruan D, Geert W, Song J, Xu Y (2007) A rough sets based characteristic relation approach for dynamic attribute generalization in data mining. Knowl Based Syst 20(5):485–494
24. Li Z, Liu J, Tang J, Lu H (2015) Robust structured subspace learning for data representation. IEEE Trans Pattern Anal Mach Intell 37(10):2085–2098
25. Li Z, Liu J, Yang Y, Zhou X, Lu H (2014) Clustering-guided sparse structural learning for unsupervised feature selection. IEEE Trans Knowl Data Eng 26(9):2138–2150
26. Li Z, Tang J (2015) Unsupervised feature selection via nonnegative spectral analysis and redundancy control. IEEE Trans Image Process 24(12):5343–5355

27. Parthalain N, Shen Q, Jensen R (2010) A distance measure approach to exploring the rough set boundary region for attribute reduction. IEEE Trans Knowl Data Eng 22(3):305–317
28. Pawlak Z (1982) Rough sets. Int J Comput Inform Sci 11(5):341–356
29. Perkins S, Theiler J (2003) Online feature selection using grafting. In: International conference on machine learning. ACM Press, pp 592–599
30. Pudil P, Novovičová J, Kittler J (1994) Floating search methods in feature selection. Pattern Recogn Lett 15(11):1119–1125
31. Quinlan JR (1993) C4.5: programs for machine learning. Morgan Kaufmann Publishers Inc., San Francisco
32. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. Preprint arXiv:1409.1556
33. Skowron A, Stepaniuk J (1996) Tolerance approximation spaces. Fundam Inf 27(2–3):245–253
34. Swiniarski RW, Skowron A (2003) Rough set methods in feature selection and recognition. Pattern Recogn Lett 24(6):833–849
35. Ungar L, Zhou J, Foster D, Stine B (2005) Streaming feature selection using IIC. In: Proceedings of the 10th international conference on artificial intelligence and statistics
36. Walczak B, Massart D (1999) Rough sets theory. Chemometr Intell Lab Syst 47(1):1–16
37. Wang F, Liang J, Qian Y (2013) Attribute reduction: a dimension incremental strategy. Knowl Based Syst 39:95–108
38. Wang J, Zhao P, Hoi S, Jin R (2014) Online feature selection and its applications. IEEE Trans Knowl Data Eng 26(3):698–710. https://doi.org/10.1109/TKDE.2013.32
39. Witten IH, Frank E (2005) Data mining: practical machine learning tools and techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems). Morgan Kaufmann Publishers Inc., San Francisco
40. Wu X, Yu K, Ding W, Wang H, Zhu X (2013) Online feature selection with streaming features. IEEE Trans Pattern Anal Mach Intell 35:1178–1192
41. Yang X, Liu W, Tao D, Cheng J (2017) Canonical correlation analysis networks for two-view image recognition. Inf Sci 385–386:338–352
42. Yao Y (2007) Decision-theoretic rough set models. Springer, Berlin, pp 1–12
43. Yu K, Ding W, Simovici DA, Wang H, Pei J, Wu X (2015) Classification with streaming features: an emerging-pattern mining approach. ACM Trans Knowl Discov Data (TKDD) 9(4):30
44. Yu K, Ding W, Wu X (2016) Lofs: a library of online streaming feature selection. Knowl Based Syst 113:1–3
45. Yu K, Wang D, Ding W, Pei J, Small DL, Islam S, Wu X (2015) Tornado forecasting with multiple markov boundaries. In: Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, pp 2237–2246
46. Yu K, Wu X, Ding W, Pei J (2014) Towards scalable and accurate online feature selection for big data. In: IEEE international conference on data mining (ICDM). IEEE, pp 660–669
47. Zhou J, Foster D, Stine R, Ungar L (2005) Streaming feature selection using alpha-investing. In: Proceedings of the 11th ACM SIGKDD international conference on knowledge discovery in data mining, Chicago, IL, USA
48. Ziarko W (1993) Variable precision rough set model. J Comput Syst Sci 46(1):39–59