2013 IEEE 13th International Conference on Data Mining

Group Feature Selection with Streaming Features

Haiguang Li, Xindong Wu
Department of Computer Science
University of Vermont
{hli, xwu}@cems.uvm.edu

Zhao Li
TCL Research America
zhaoli01@tcl.com

Wei Ding
Department of Computer Science
University of Massachusetts Boston
ding@cs.umb.edu

***Abstract*—Group feature selection makes use of structural information among features to discover a meaningful subset of features. Existing group feature selection algorithms only deal with pre-given candidate feature sets and they are incapable of handling streaming features. On the other hand, feature selection algorithms targeted for streaming features can only perform at the individual feature level without considering intrinsic group structures of the features. In this paper, we perform group feature selection with streaming features. We propose to perform feature selection at the group and individual feature levels simultaneously in a manner of a feature stream rather than a pre-given candidate feature set. In our approach, the group structures are fully utilized to reduce the cost of evaluating streaming features. We have extensively evaluated the proposed method. Experimental results have demonstrated that our proposed algorithms statistically outperform state-of-the-art methods of feature selection in terms of classification accuracy.**

I. Introduction

Feature selection, a.k.a. variable selection, reduces the number of features to speed up the learning process, to improve learning accuracy, enhance generalization capability, and facilitate model interpretation. It hence has been an active research field for decades in data mining and machine learning, and has broad applications in text mining, genomic analysis, intrusion detection, and image retrieval [1]. Generally speaking, feature selection removes noisy, irrelevant, and redundant features, and selects relevant features from a given candidate feature set or a feature stream at the individual feature level.

In this paper we address the problem of feature selection with streaming features at both the group level and the individual feature level. We target at the problem that features possess certain group structures, which is typical in many real-world applications. One of the most common examples, the multi-factor Analysis of Variance (ANOVA), is a statistical technique to test the significant differences among multiple means. In this technique, the inferences are made by analyzing variances rather than means, as each mean (factor) is expressed through a group of variances (dummy variables) [2]. Another popular example is the categorical feature being represented as a group of dummy features [3]. A dummy feature is a design feature that takes a binary value 0 or 1 to indicate the absence or presence of a categorical value of a categorical feature. Obviously, each group of dummy features/variables corresponds to one measured feature/variable and is directly related to the measurement cost. Therefore, in such cases, feature/variable selection corresponds to the selection of groups rather than individual dummy features/variables. As generating features in different groups may require different procedures, measurements, domain knowledge, etc., the candidate features are very likely to appear in the form of a feature steam, in which features are generated dynamically and arrive one by one and group by group. This situation appears in many real-world applications. For example, several giga features with values in {A, C, G, T}, thus each feature being represented by a group of 4 dummy features, can be generated using the next-generation sequencing techniques only on one run [1]. The storage cost is very expensive to keep those features, and it is not practical to wait until all features have been generated before learning begins. Therefore, it could be far more preferable to generate candidate features one at a time for all observations [4], [5], [6].

Many existing algorithms can effectively perform feature selection from a given candidate feature set or a feature stream. However, without considering group structures, they always try to select features with sparsity (a small percentage) only at the individual feature level. Selecting features with sparsity at both the group level and the individual feature level is more preferable when group structures exist. For instance, in the above categorical feature example, 20 dummy features in 20 groups correspond to 20 measured features, while 20 features in 2 groups only need to measure two original features, and the measurement cost decreases sharply.

In this paper, a new algorithm named GFSSF is proposed for *g*roup *f*eature *s*election with *s*treaming *f*eatures. Unlike existing algorithms, GFSSF performs feature selection at both the group and individual feature levels simultaneously from the features generated and arrived so far to achieve accurate classification peformance. The main contributions of this paper are as follows:

1) Utilizing feature group structures, GFSSF effectively identifies relevant features from important groups and selects features with sparsity at both the group and individual feature levels.

2) If without using feature group structures, GFSSF treats each feature as an individual group and performs feature selection just at the individual feature level. Moreover, a user-specified reasonable group size can improve the time efficiency significantly.

3) GFSSF can be easily configured to perform feature selection at the group level, the individual feature level, or both.

4) Extensive experiments have demonstrated that our proposed method is superior to others on extensive benchmark datasets with or without group structures.

1550-4786/13 $31.00 © 2013 IEEE
DOI 10.1109/ICDM.2013.137

II. Related Work

A standard feature selection scenario assumes that all candidate features are available before feature selection takes place, where mutual information (MI) has attracted a lot of attention as a measure of relevance and redundancy among features. Battiti [7] defined the feature selection problem as the process of selecting the most relevant features from a set of candidate features, and proposed a selection method MIFS. Later, Kwak and Choi in [8] analyzed the limitations of MIFS and proposed a greedy selection method MIFS-U. A redundancy criterion was introduced in the min-redundancy max-relevance (mRMR) method in [9]. Normalized mutual information feature selection (NMIFS), an enhancement over MIFS, MIFS-U, and mRMR, was proposed in [10]; and Liu et al. proposed a feature selection algorithm based on dynamic mutual information in [11]. Recently, Brown et al. [12] proposed a unifying framework for information theoretic feature selection.

A streaming feature selection scenario assumes that the candidate features are generated dynamically and arrive one at a time instead of all candidate features being known in advance. Algorithms designed for the standard feature selection scenario cannot fit streaming features well due to the candidate features not being available at the beginning. Perkins and Theiler proposed the Grafting method in [4], which is an online feature selection method based on a stagewise gradient descent technique. Zhou et al. [13], [14] presented two algorithms, information-investing and α-investing, based on streamwise regression for streaming feature selection. Wu et al. [5] proposed an online streaming feature selection framework with two algorithms OSFS and fast-OSFS.

Group feature selection, the selection of important groups rather than individual features, is a new and interesting topic. Lasso [15] is a shrinkage and selection method for linear regression, which minimizes the sum of squared errors with the L_1 penalty on the sum of the absolute values of the coefficients. By extending the L_1 penalty of lasso to an intermediate between the L_1 and L_2 penalty [16], [17], Yuan and Lin in [2] proposed the group lasso model for selecting grouped variables for accurate prediction in regression. Later, Roth and Fischer [16] extended the group lasso to logistic regression models, which are especially suitable for high dimensional problems. Other extensions of group lasso include the group lasso for generalized linear models [3], the group lasso with overlap between groups [17], etc. The sparse group lasso criterion was proposed in [18], and can yield solutions that are sparse at both the group and feature levels.

In this paper, we improve the group feature selection from the standard feature selection scenario to the streaming feature selection scenario by exploiting entropy and mutual information in information theories.

III. Preliminary

Problem Definition: Given a candidate feature set $\mathbb{X}$ and the target feature Y, the task of feature selection is to find a subset of features $\mathbb{F} \subseteq \mathbb{X}$, such that $\mathbb{F}$ maximally represents Y with minimal residual uncertainty.

By extending the definitions of entropy and mutual information in information theories [19], [20] from variables to sets of variables, the uncertainty of Y is the entropy $H(Y)$, and the information shared by $\mathbb{F}$ and Y is their mutual information $I(\mathbb{F};Y)$. Therefore, given the information of X, X can share $I(X;Y)$ with Y, and the residual uncertainty of Y reduces to $H(Y|X) = H(Y) - I(X;Y)$. Furthermore, if the information of $\forall X \in \mathbb{F}$ is given, $\mathbb{F}$ can share $I(\mathbb{F};Y)$ with Y, and the residual uncertainty of Y is $H(Y|\mathbb{F})$. Ideally, if $I(\mathbb{F};Y) = H(Y)$, then knowing information of the features in $\mathbb{F}$ can determine Y directly as the residual uncertainty reduces to $H(Y|\mathbb{F}) = H(Y) - I(\mathbb{F};Y) = 0$.

Given a feature stream, in which the size p of the candidate feature set $\mathbb{X} = \{X_1, \cdots, X_p\}$ is unknown or even infinite, features arrive one at a time, and the number of observations n is constant [4], [13], [14], [5]. The n observations are independent and identically distributed. The values of the target feature Y for the n observations arrived at the very beginning. Suppose that these p features can be divided into q disjoint groups, then we can rewrite $\mathbb{X} = \{G_1; \cdots; G_q\}$, where $G_i = \{X_{\sum_{j=1}^{i-1}|G_i|+1}, \cdots, X_{\sum_{j=1}^{i}|G_i|}\}$ for $\forall i \in [1, q]$. Concretely, the task of group feature selection from streaming features is to seek a group set Γ that is sparse at both the group level and the individual feature level, by solving the following optimization problem

$$\min_{\Gamma}\{H(Y) - I(\Gamma;Y)\} + \{\lambda_1|\Gamma|_g + \lambda_2|\Gamma|_f\}, \qquad (1)$$

where $|\Gamma|_g$ and $|\Gamma|_f$ are the number of groups and the number of features in the selected group set Γ, respectively. The first part $H(Y) - I(\Gamma;Y)$ is the residual uncertainty of Y after knowing the information of features in Γ. The second part $\lambda_1|\Gamma|_g + \lambda_2|\Gamma|_f$, which controls the numbers of selected groups and features, is a penalty.

Definition 1. *[Irrelevance] Given two features X and Y, X is irrelevant to Y if and only if $I(X;Y) = 0$.*

Theorem 1. *Given the target feature Y and a newly arrived feature X from the feature stream, if X is irrelevant to Y, then X can be safely discarded.*

Proof: $\because I(X;Y) = H(X) + H(Y) - H(X,Y)$ [19] and $I(X;Y) = 0$ (given), $\therefore H(X,Y) = H(X) + H(Y)$, $\therefore X$ and Y are independent, $\therefore$ discarding X is lossless for Y. ■

Definition 2. *[Redundancy] Given two features X and Y, and a set of features $\mathbb{F}$, X is redundant to $\mathbb{F}$ for Y if and only if $I(X;Y|\mathbb{F}) = 0$.*

Theorem 2. *Given the target feature Y, and a set of selected features $\mathbb{F}$, if $\exists X \in \mathbb{F}$ **s.t** X is redundant to $\mathbb{F} \setminus \{X\}$ for Y, then X can be safely removed from $\mathbb{F}$.*

Proof: $\because I(\mathbb{F};Y) = I(\mathbb{F} \setminus \{X\} \cup \{X\};Y) = I(\mathbb{F} \setminus \{X\};Y) + I(X;Y|\mathbb{F}\setminus\{X\})$ and $I(X;Y|\mathbb{F}\setminus\{X\}) = 0$ (given), $\therefore I(\mathbb{F};Y) = I(\mathbb{F} \setminus \{X\};Y)$, $\therefore \mathbb{F} \setminus \{X\}$ can provide the same information for Y as $\mathbb{F}$, $\therefore$ removing X is lossless for Y. ■

Definition 3. *[Coverage] Given three features X, $\bar{X}$ and Y, X covers $\bar{X}$ on Y if and only if $I(\bar{X};Y|X) = 0$.*

Theorem 3. *Given the target feature Y, a newly arrived feature X, and a set of selected features $\mathbb{F}$, if $\exists \bar{X} \in \mathbb{F}$ **s.t** X covers $\bar{X}$ on Y, then $\bar{X}$ can be safely replaced by X in $\mathbb{F}$.*

Proof: $\because X$ covers $\bar{X}$ on Y, $\therefore I(X;Y) \geq I(\bar{X};Y)$ and $I(\bar{X};Y|X) = 0$. Furthermore, with $I(X;Y|\mathbb{F}) \geq I(\bar{X};Y|\mathbb{F})$ and based on $\mathbb{F}$, X can provide all information that $\bar{X}$ provides for Y. Therefore, this replacement is lossless for Y. Meanwhile, as a replacement, it will not change the number of features in $\mathbb{F}$. ■

Such a replacement may bring two potential benefits: (1) $\because I(X;Y|\mathbb{F}) \geq I(\bar{X};Y|\mathbb{F})$, $\therefore X$ has the potential to provide more new information for Y based on $\mathbb{F}$ than $\bar{X}$; (2) $\because \forall \hat{X} \in \mathbb{F}$ covered by X on Y can be safely removed (Theorem 2), therefore, this replacement has the potential to decrease the number of selected features.

Theorem 4. *Given the target feature Y, a newly arrived feature X from the feature stream, and a set of selected features $\mathbb{F}$, if X is redundant to $\mathbb{F}$ for Y and X cannot cover $\forall \bar{X} \in \mathbb{F}$ on Y, then X can be safely discarded.*

Proof: $\because X$ is redundant to $\mathbb{F}$ for Y, $\therefore$ adding X cannot provide any new information for Y based on $\mathbb{F}$. As X cannot cover $\forall \bar{X} \in \mathbb{F}$ on Y, $\therefore \nexists \bar{X} \in \mathbb{F}$ **s.t.** $\bar{X}$ can be safely removed from $\mathbb{F}$, $\therefore$ adding X into $\mathbb{F}$ will increase the number of selected features. Overall, discarding X is lossless for Y, and will avoid redundancy in $\mathbb{F}$. ■

IV. The Proposed Method

In this section, we propose our group feature selection with streaming features (GFSSF), which consists of feature level and group level selections. The group structures are obtained from domain knowledge or start with a user-specified group size for the sake of time efficiency.

A. Feature Level Selection

The feature level selection algorithm InGFSSF is given in Algorithm 1. It only processes features from the same group, and seeks the best feature subset from the arrived features so far. The selected feature subset $\mathbb{F}$ is initialized as an empty set in Step 2 if the latest arrived feature X is the first one in the group. Step 4 tests whether X is relevant to the target feature Y (Definition 1), and if failed, it is discarded directly (Theorem 1). *If* (Step 5) X is not redundant to $\mathbb{F}$ for Y, Step 6 adds it into $\mathbb{F}$ as it can provide new information for Y that any other formerly selected feature cannot; *Else if* (Step 7) X is redundant but it can cover some $X_i \in \mathbb{F}$ on Y (Definition 3), X_i is replaced by X in Step 8 (Theorem 3); Otherwise (Step 9), X is redundant to $\mathbb{F}$ for Y, and meanwhile, it cannot cover any $X_i \in \mathbb{F}$ on Y, therefore, it is discarded in Step 10 (Theorem 4). Once the new feature X is selected and added into $\mathbb{F}$, some of the other features in $\mathbb{F}$ may become redundant. Therefore, the '*while*' loop (Steps 12 – 14) removes any redundancy in $\mathbb{F}$ (Theorem 2). Finally, Step 16 returns the currently selected subset $\mathbb{F}$.

B. Group Level Selection

Algorithm 2 is the group level selection algorithm AgGFSSF, and seeks a set of groups that can cover as much uncertainty of the target feature Y as possible with a minimum cost (i. e. the penalty on the number of selected groups and the number of selected features). This algorithm is similar to Algorithm 1, and the only differences are the selection level and the penalty. In Step 6, the newly arrived group $\mathbb{F}$ is selected if the new information it provides for Y is more than the penalty that comes with it (the '*if*' test in Step 5). If replacing $\mathbb{F}_i \in \Gamma$ with $\mathbb{F}$ can lower Formula (1) (Step 7), Step 8 performs this replacement. After adding $\mathbb{F}$, the '*while*' loop (Steps 12 – 14) tries to lower Formula (1) by removing some formerly selected groups.

Algorithm 1 InGFSSF

Require:
X; Y; The group structures;
Ensure:
1: **if** {X is the first arrived feature of the group } **then**
2: $\mathbb{F} \leftarrow \emptyset$
3: **end if**
4: **if** $I(X;Y) \neq 0$ **then**
5: **if** $I(X;Y|\mathbb{F}) > 0$ **then**
6: $\mathbb{F} \leftarrow \mathbb{F} \cup \{X\}$
7: **else if** $\exists X_i \in \mathbb{F}$ **s.t.** $I(X_i;Y|X) = 0$ **then**
8: $\mathbb{F} \leftarrow \mathbb{F} \cup \{X\} \setminus \{X_i\}$
9: **else**
10: Goto Step (16)
11: **end if**
12: **while** $\exists X_i \in \mathbb{F}$ **s.t.** $I(X_i;Y|\mathbb{F} \setminus \{X_i\}) = 0$ **do**
13: $\mathbb{F} \leftarrow \mathbb{F} \setminus \{X_i\}$
14: **end while**
15: **end if**
16: **return** $\mathbb{F}$

Algorithm 2 AgGFSSF

Require:
$\mathbb{F}$; Y; The group structures;
Ensure:
1: **if** {$\mathbb{F}$ is the first arrived group} **then**
2: $\Gamma \leftarrow \emptyset$
3: **end if**
4: **if** $|\mathbb{F}|_f > 0$ **then**
5: **if** $I(\mathbb{F};Y|\Gamma) \geq \lambda_1 + \lambda_2 \times |\mathbb{F}|_f$ **then**
6: $\Gamma \leftarrow \Gamma \cup \{\mathbb{F}\}$
7: **else if** $\exists \mathbb{F}_i \in \Gamma$ **s.t** $I(\mathbb{F};Y|\Gamma \setminus \{\mathbb{F}_i\}) - I(\mathbb{F}_i;Y|\Gamma \setminus \{\mathbb{F}_i\}) \geq \lambda_2 \times (|\mathbb{F}|_f - |\mathbb{F}_i|_f)$ **then**
8: $\Gamma \leftarrow \Gamma \cup \{\mathbb{F}\} \setminus \{\mathbb{F}_i\}$
9: **else**
10: Goto Step (16)
11: **end if**
12: **while** $\exists \mathbb{F}_i \in \Gamma$ **s.t** $I(\mathbb{F}_i;Y|\Gamma \setminus \{\mathbb{F}_i\}) < \lambda_1 + \lambda_2 \times |\mathbb{F}_i|_f$ **do**
13: $\Gamma \leftarrow \Gamma \setminus \{\mathbb{F}_i\}$
14: **end while**
15: **end if**
16: **return** Γ

C. Framework of GFSSF

Algorithm 3 GFSSF

Require:
The feature stream; The group structures;
Ensure:
1: **repeat**
2: **repeat**
3: $X \leftarrow$ the newly arrived feature
4: $\mathbb{F} \leftarrow$ InGFSSF(X)
5: **until** {X is the last arrived feature of a group}
6: $\Gamma \leftarrow$ AgGFSSF($\mathbb{F}$)
7: **until** {Meet some stopping criteria}
8: **return** Γ

The framework of GFSSF is presented in Algorithm 3. With only one single pass over the feature stream GFSSF is

	MIFS	JMI	mRMR	RELIEF	α-invest.	Grafting	OSFS	Fast-OSFS	Lasso	Group Lasso	GFSSF$_\circ$	GFSSF$_\bullet$
WDBC	92.62(**06**)	92.12(10)	98.77(10)	**99.30**(11)	95.12(30)	92.81(12)	**99.30**(08)	94.05(11)	93.67(**06**)	97.36(10){4}	98.59(**06**)	**99.30**(**06**){3}
WPBC	62.63(05)	93.46(05)	62.63(05)	95.98(10)	86.87(04)	91.93(04)	85.87(03)	62.63(**01**)	81.86(06)	85.86(12){5}	97.47(05)	**98.99**(04){3}
IONOSPHERE	67.24(05)	83.86(10)	95.30(11)	91.58(12)	90.12(10)	88.05(04)	90.91(**03**)	87.56(04)	88.31(16)	90.59(21){5}	98.86(**03**)	**99.15**(05){2}
SPECTF	63.75(06)	94.25(08)	87.75(10)	95.00(08)	65.14(**03**)	91.75(21)	96.25(07)	95.31(11)	82.51(08)	95.00(19){3}	93.52(**03**)	**98.75**(07){2}
ARRHYTHMIA	85.40(10)	82.58(09)	95.02(12)	86.23(30)	91.59(**08**)	92.69(19)	95.80(13)	95.80(19)	79.13(17)	85.00(27){4}	94.47(**08**)	**96.35**(12){4}
DLBCL	—	—	—	74.32(15)	91.60(04)	85.42(05)	91.51(**02**)	91.51(27)	88.16(08)	91.25(19){7}	**92.21**(**02**)	**92.21**(**02**){2}
LUNG	—	—	—	79.92(16)	**100.0**(26)	91.96(05)	**100.0**(**02**)	97.92(03)	94.85(07)	97.94(07){3}	98.96(**02**)	**100.0**(03){2}
CNS	—	—	—	71.00(19)	61.67(**02**)	**95.00**(**02**)	**95.00**(04)	93.67(82)	87.70(12)	**95.00**(23){7}	**95.00**(**02**)	**95.00**(**02**){1}
ARCENE	—	—	—	68.72(20)	95.30(09)	88.15(09)	91.54(**02**)	89.73(09)	—	—	**99.60**(05)	**99.60**(04){4}
OVARIAN	—	—	—	65.32(36)	87.13(14)	87.22(15)	89.25(**02**)	86.12(04)	—	—	**94.02**(05)	**94.02**(04){2}
AVG.	77.33	89.25	88.49	82.74	86.44	90.49	93.58	89.02	87.52	92.25	96.27	**97.34**
WTL	0/0/10	0/0/10	0/0/10	0/1/9	0/1/9	0/1/9	0/3/7	0/0/10	0/0/10	0/1/9	0/4/6	**4/6/0**

TABLE I: Comparison of Classification Accuracies (%) Using 12 Algorithms on 10 Datasets without Group Structures

able to complete feature selection at the individual feature level and the group level simultaneously. Given a feature steam, the target feature Y and the group structures, it seeks a group set Γ to minimize Formula (1). Step 3 keeps receiving new features from the feature steam. Step 4 invokes the feature level selection algorithm InGFSSF (Algorithm 1) to process the newly arrived feature in the current group, where $\mathbb{F}$ is the currently selected feature set. Once all features of a group have arrived, the feature level selection for that group is done. Then in Step 6, the group level selection algorithm AgGFSSF (Algorithm 2) is invoked to process the new group, where Γ is the currently selected group set.

From Algorithms 1 and 2, there is no tolerance for irrelevant and redundant features in InGFSSF, while AgGFSSF is more aggressive in discarding groups with a large size yet little effect. Therefore, GFSSF always yields results with sparsity at both the group level and the individual feature level. On the one hand, if all candidate features are in a single group, GFSSF only performs selection at the individual feature level; on the other hand, it only performs selection at the group level if InGFSSF does nothing. Therefore, GFSSF can be easily set to perform selection at the individual feature level, the group level, or both by the group size and its parameters.

V. Experiments

A. Experimental Settings

We choose 10 well-known algorithms from the standard feature selection scenario (MIFS [7], JMI [21], mRMR [9], RELIEF [22] and Lasso [15]), the streaming feature selection scenario (Grafting [4], α-investing [13], OSFS [5], Fast-OSFS [5]), and the group feature selection scenario (Group Lasso [2]) for our experiments. Two variants of our proposed algorithm, GFSSF$_\circ$ and GFSSF$_\bullet$, are evaluated in our experiments. The group size of GFSSF$_\bullet$ is user specified or from the group structures among features in datasets, while GFSSF$_\circ$ sets its group size to infinity. Obviously, GFSSF$_\circ$ only performs feature level selection as all features fall into the same group.

The parameter λ of Grafting was chosen through cross-validation, α-investing exploited its default parameters, and the statistical significance level parameter α for both OSFS and Fast-OSFS was set to 0.05 or 0.01 whichever reached a better performance. Lasso [15] and Group Lasso [2] both adopted the LogReg($\bullet$) penalty function. The parameters of both GFSSF$_\circ$ and GFSSF$_\bullet$ were $\lambda_1 = \frac{H(Y)}{4|\mathbb{X}|_g}$ and $\lambda_2 = \frac{H(Y)}{4|\mathbb{X}|_f}$, where $H(Y)$ is the entropy of the target feature Y, and $|\mathbb{X}|_g$ and $|\mathbb{X}|_f$ are the number of arrived groups and the number of arrived features so far, respectively.

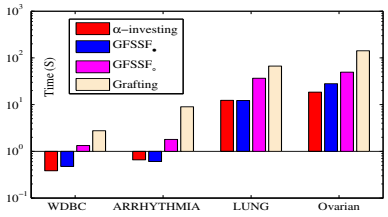


Fig. 1: Running Time of 4 Algorithms on 4 Datasets

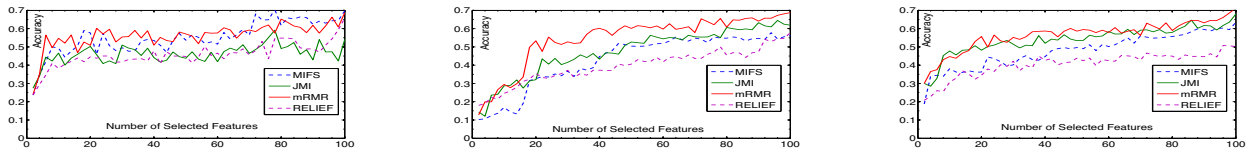
To validate conservative performance, 15 datasets were adopted in our experiments. 5 UCI benchmark datasets: WDBC, WPBC, IONOSPHERE, SPECTF and ARRHYTHMIA; 5 challenge datasets with relatively high feature dimensions downloaded from (http://mldata.org/repository): DLBCL (7,130 features, 77 instances), LUNG (7,130 features, 96 instances), CNS (7,130 features, 96 instances), ARCENE (10,000 features, 100 instances) and OVARIAN (15,155 features, 253 instances); 5 UCI datasets with generated group structures: HILL-VALLEY (400 features, 606 instances), NORTHIX (800 features, 115 instances), MADELON (2,000 features, 4,400 instances), ISOLET (2,468 features, 7,797 instances), and MULTI-FEATURES (2,567 features, 2,000 instances). The group structures were built by introducing dummy features: first, continuous features were discretized into nominal ones; second, each feature was replaced by 4 dummy features; and finally, for each dataset, a balanced training dataset was built by randomly selecting without replacement, and the rest were reserved for testing.

In our experiments, four popular classifiers, namely, Naive-Bayes [23], k-NN [24], C4.5 [25], and Randomforest [26], were chosen to test classification capability of the selected feature subset, and the best accuracy was selected as the result. To achieve impartial results, if an independent testing dataset is not provided, 10-fold cross-validation was adopted for each "algorithm–dataset" combination in verifying classification capability. The experiments were conducted on a computer with Windows 7, 3.33 GHz dual-core CPU, and 4GB memory.

B. Experimental Results

Results on Datasets without Group Structures: Table I lists the results about classification accuracies (%) and the number of selected features (in ($\bullet$)) on 10 datasets without group structures using 12 different algorithms.

Table I shows that the classification accuracies of GFSSF$_\circ$ and GFSSF$_\bullet$ are better than the others in most cases. One can



(a) Northix (b) Hill-Valley (c) Multi-Features

Fig. 2: Accuracies of 4 Other Algorithms on 3 Datasets with Group Structures

	Training Instances	α-investing	Grafting	OSFS	Fast-OSFS	Lasso	Group Lasso	GFSSF$_\circ$	GFSSF$_\bullet$
NORTHIX	30	67.32(94)[56]	61.24(92)[92]	70.13(**22**)[18]	70.13(**22**)[18]	64.17(53)[53]	68.51(97)[89]	70.31(78)[71]	**85.42**(29)[**13**]
	60	75.15(84)[82]	74.87(85)[79]	79.14(27)[24]	79.48(25)[21]	72.17(62)[62]	71.58(97)[81]	91.16(**21**)[19]	**93.62**(23)[**10**]
HILL–VALLEY	100	38.05(43)[37]	47.43(31)[**21**]	42.40(32)[25]	51.49(33)[30]	59.12(37)[36]	63.15(47)[43]	**73.92**(28)[27]	72.25(32)[24]
	200	66.28(61)[57]	55.38(43)[42]	47.45(32)[26]	48.45(32)[32]	75.51(27)[21]	73.15(33)[22]	78.16(**19**)[17]	**79.92**(26)[**14**]
	300	**87.27**(76)[72]	63.84(57)[37]	52.16(**18**)[18]	52.16(**18**)[18]	75.58(23)[21]	80.14(28)[18]	79.75(23)[18]	**87.27**(28)[**09**]
MULTI–FEATURES	200	22.15(13)[13]	49.34(13)[**12**]	40.24(**12**)[**12**]	63.48(13)[13]	69.21(17)[16]	64.45(17)[14]	**78.79**(18)[17]	74.52(22)[14]
	400	68.21(21)[17]	53.48(15)[14]	40.24(**12**)[**12**]	40.24(**12**)[**12**]	72.34(27)[19]	72.56(23)[15]	77.06(21)[17]	**78.97**(26)[14]
	600	**83.79**(26)[22]	62.48(17)[17]	49.61(**14**)[**14**]	49.61(**14**)[**14**]	77.92(33)[21]	83.74(28)[18]	77.73(23)[19]	82.97(28)[17]
	800	84.94(47)[29]	80.86(51)[27]	52.76(**13**)[**13**]	51.58(16)[16]	83.33(25)[17]	87.59(23)[18]	81.91(16)[11]	**88.58**(19)[**12**]
	1,000	90.76(94)[56]	91.42(98)[87]	71.88(**12**)[11]	71.88(**12**)[11]	85.79(31)[23]	90.55(25)[10]	87.16(18)[11]	**91.85**(19)[**06**]
MADELON	400	78.39(18)[15]	50.18(**12**)[**12**]	57.06(13)[13]	57.06(13)[13]	73.12(25)[16]	72.39(17)[13]	**82.44**(18)[16]	76.14(16)[13]
	800	81.12(22)[17]	75.38(15)[15]	64.09(**14**)[**13**]	64.09(**14**)[**13**]	81.66(26)[16]	80.75(19)[14]	84.13(18)[15]	**85.61**(19)[**13**]
	1,200	86.16(27)[27]	75.95(**16**)[16]	70.66(**16**)[15]	65.08(**16**)[14]	86.59(28)[17]	88.79(22)[14]	88.12(21)[16]	**89.32**(18)[**10**]
	1,600	86.94(28)[27]	80.86(21)[17]	72.69(**18**)[16]	72.69(**18**)[16]	86.00(21)[17]	85.61(24)[14]	**89.92**(22)[17]	89.11(18)[**08**]
	2,000	82.46(27)[14]	83.10(19)[16]	64.26(19)[16]	60.81(18)[16]	84.03(27)[21]	87.76(**12**)[05]	85.41(15)[07]	**90.31**(13)[**04**]
ISOLET	700	62.15(53)[39]	59.34(45)[41]	50.28(**22**)[**22**]	67.83(54)[53]	68.92(27)[26]	67.56(27)[24]	**74.95**(28)[27]	72.15(**22**)[**13**]
	1,400	64.61(48)[47]	64.74(23)[**12**]	51.46(**12**)[**12**]	51.46(**12**)[**12**]	71.64(27)[18]	75.62(23)[15]	76.63(21)[17]	**77.08**(26)[14]
	2,100	72.23(76)[62]	65.57(17)[17]	54.96(**14**)[14]	54.96(**14**)[14]	74.16(29)[25]	81.57(18)[08]	79.47(23)[19]	**82.97**(18)[**07**]
	2,800	**89.86**(98)[89]	79.98(51)[27]	62.65(**13**)[13]	62.15(16)[16]	81.34(26)[18]	88.25(24)[09]	83.19(17)[10]	89.18(19)[**08**]
	3,500	92.15(92)[90]	92.52(95)[87]	90.14(32)[28]	88.89(29)[27]	91.87(17)[13]	91.55(17)[09]	92.98(20)[19]	**96.36**(**15**)[**05**]

TABLE II: Comparison of Classification Accuracies (%) Using 8 Algorithms on 5 Datasets with Group Structures

easily observe that GFSSF$_\bullet$ gets the highest accuracies on all datasets, while GFSSF$_\circ$ gets 4 out of 10. Besides, the average accuracy of GFSSF$_\circ$ is higher than any other competitor, and even in the loss cases, it selects fewer features with competitive accuracies. For instance, on SPECTF, it only selects 3 features, the least number of features on that dataset, for an accuracy of 93.52%, while both RELIEF and JMI select 8, OSFS selects 7, Fast-OSFS selects 11, and Group Lasso selects 19.

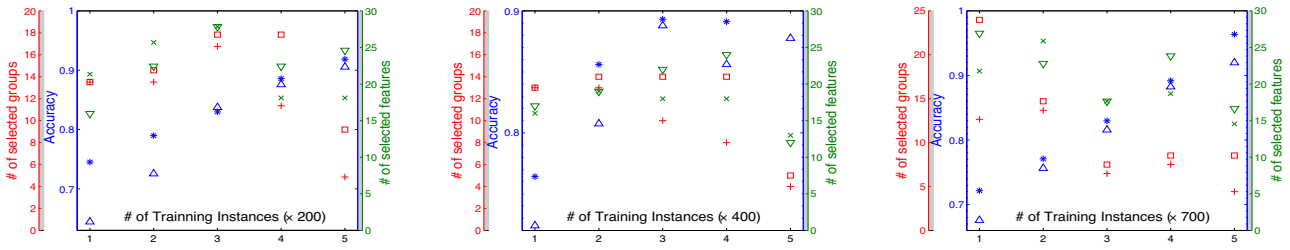
Meanwhile, Table I also shows that GFSSF$_\circ$ and GFSSF$_\bullet$ select very compact features. GFSSF$_\circ$ selects the least number of features from 7 out of 10 datasets, while GFSSF$_\bullet$ does 3 out of 10. And their max number of selected features is only 8 and 12, respectively. OSFS also selects compact features but with much lower accuracies. For instance, on ARCENE, OSFS selects 2 features with an accuracy of 91.54%, while GFSSF$_\bullet$ and GFSSF$_\circ$ achieve 99.60% with 4 and 5 features, respectively. Obviously, it is a good tradeoff to increase the accuracy from 91.54% to 99.60% with 2 or 3 more features.

As shown in Table I, MIFS, JMI, mRMR, Lasso and Group Lasso crash when the number of features is increased to a certain threshold. The running time of RELIEF depends on the user-specified number of selected features, in our experiments, and it crashed when the number of features is larger than 7000 and the user-specified number of selected features approaches 40, which indicates their poor scalabilities. Therefore, we did not compare our algorithms against them in running time. Since OSFS and Fast-OSFS were implemented in C and ours in Matlab, a direct time comparison is meaningless. Therefore, we only compare the running time of our algorithms with the most comparable algorithms α-investing and Grafting. As demonstrated in Figure 1, both GFSSF$_\circ$ and GFSSF$_\bullet$ can be applied effectively to very high dimensional datasets. For instance, they can process the Ovarian dataset, which contains 15115 features, within less than 30 seconds. Besides, GFSSF$_\bullet$ requires less running time than GFSSF$_\circ$ on the same dataset.

RESULTS ON DATASETS WITH GROUP STRUCTURES: For datasets with group structures, the standard feature selection algorithms MIFS, JMI, mRMR and RELIEF crash when running on the Madelon and Isolet datasets, and their performances on the other three datasets are shown in Figure 2. With the whole training sets, their best accuracies are only close to 0.7. Obviously, the standard feature selection algorithms cannot fit datasets with group structures vary well.

Table II lists the experimental results of classification accuracy (%), the number of selected features (in (•)), and the number of selected groups (in [•]) on 5 datasets with group structures using 8 feature selection algorithms. Several observations can be drawn from Table II. First, the accuracies of almost all algorithms increase with the number of training instances. Second, OSFS and Fast-OSFS select compact features, however, their accuracies are not as good. For instance, on the Multi-Features dataset with the entire training set, they both select the least number of features (12) but with the worst accuracy (71.88%). Third, α-investing and Grafting have good accuracies especially with a large training set, but they both tend to select too many features. For example, they select more than 90 features when using the whole training sets on the Multi-Feature and Isolet datasets while others only select about 30 features. Fourth, all algorithms except Group Lasso and GFSSF$_\bullet$ select too many groups, which means they try to select features from different groups. Although the selected subset may have a good enough accuracy, it's hard to interpret or just meaningless. Overall, the results of Group Lasso and GFSSF$_\bullet$ are the best as they can effectively utilize the group structures to guide their selection processes.

For a better visualization, Figure 3 presents the comparisons between Group Lasso (accuracy $\triangle$, number of selected features $\bigtriangledown$, number of selected groups $\square$) and GFSSF$_\bullet$ (accuracy $*$, number of selected features $\times$, number of selected groups $+$). Due to space limitations, only comparisons on



(a) Multi-Features (b) Madelon (c) Isolet

Fig. 3: GFSSF$_\bullet$ vs Group Lasso on 3 Datasets with Group Structures

the Multi-Features, Madelon, and Isolet datasets are shown in Figure 3. One can easily observe that our proposed method GFSSF$_\bullet$, has achieved higher accuracies yet with less features and groups, and clearly outperforms Group Lasso on all of the five datasets with group structures.

VI. Conclusion

In this paper, we have proposed a novel online algorithm for group feature selection with streaming features. The proposed algorithm, performing selection at both the group level and the individual feature level, is more efficient than existing algorithms which select only at the feature level. Besides, a user can specify the group size to infinity, which transfers it from group feature selection to traditional feature selection. Two variants of the proposed algorithm are compared with state-of-the-art algorithms in different models and scenarios. Our comprehensive experimental studies have demonstrated that the proposed method can select less features and groups for higher classification accuracies on datasets with or without group structures.

VII. Acknowledgment

This work was supported by the US National Science Foundation (NSF) under grant CCF-0905337 and Vermont Agency of Transportation under grant 000025425.

References

[1] H. Liu, H. Motoda, R. Setiono, and Z. Zhao, "Feature selection: An ever evolving frontier in data mining," *Journal of Machine Learning Research - Proceedings Track*, vol. 10, pp. 4–13, 2010.

[2] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society, Series B*, vol. 68, pp. 49–67, 2006.

[3] V. Roth and B. Fischer, "The group-lasso for generalized linear models: uniqueness of solutions and efficient algorithms," in *Proceedings of the 25th International Conference on Machine Learning*, 2008, pp. 848–855.

[4] S. Perkins and J. Theiler, "Online feature selection using grafting," in *Proceedings of the 20th International Conference on Machine Learning*, 2003, pp. 592–599.

[5] X. Wu, K. Yu, H. Wang, and W. Ding, "Online streaming feature selection," in *Proceedings of the 27th International Conference on Machine Learning*, 2010, pp. 1159–1166.

[6] J. Wang, Z. Zhao, X. Hu, Y. Cheung, M. Wang, and X. Wu, "Online group feature selection," in *Proceedings of 2013 International Joint Conference on Artificial Intelligence*, 2013, pp. 1757–1763.

[7] R. Battiti, "Using mutual information for selecting features in supervised neural net learning," *IEEE Transactions on Neural Networks*, vol. 5, pp. 537–550, 1994.

[8] N. Kwak and C. H. Choi, "Input feature selection for classification problems," *IEEE Transactions on Neural Networks*, vol. 13, no. 1, pp. 143–159, 2002.

[9] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.

[10] P. A. Estevez, M. Tesmer, C. A. Perez, and J. M. Zurada, "Normalized mutual information feature selection," *IEEE Transactions on Neural Networks*, vol. 20, no. 2, pp. 189–201, 2009.

[11] H. Liu, J. Sun, L. Liu, and H. Zhang, "Feature selection with dynamic mutual information," *Pattern Recognition*, vol. 42, no. 7, pp. 1330–1339, 2009.

[12] G. Brown, A. Pocock, M. J. Zhao, and M. Lujun, "Conditional likelihood maximisation: A unifying framework for information theoretic feature selection," *Journal of Machine Learning Research*, vol. 13, pp. 27–66, 2012.

[13] J. Zhou, D. Foster, R. Stine, and L. Ungar, "Streaming feature selection using alpha-investing," in *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, 2005, pp. 384–393.

[14] J. Zhou, D. P. Foster, R. A. Stine, and L. H. Ungar, "Streamwise feature selection," *Journal of Machine Learning Research*, vol. 7, pp. 1861–1885, 2006.

[15] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society, Series B*, vol. 58, pp. 267–288, 1996.

[16] L. Meier, S. V. D. Geer, P. Buhlmann, and E. T. H. Zurich, "The group lasso for logistic regression," *Journal of the Royal Statistical Society, Series B*, vol. 70, pp. 53–71, 2008.

[17] L. Jacob, G. Obozinski, and J. P. Vert, "Group lasso with overlap and graph lasso," in *Proceedings of the 26th International Conference on Machine Learning*, 2009, pp. 433–440.

[18] J. Friedman, T. Hastie, and R. Tibshirani, "A note on the group lasso and a sparse group lasso," Tech. Rep., 2010.

[19] S. Guiasu, *Information Theory with Applications*. McGraw-Hill, 1977.

[20] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379–423, 1948.

[21] H. Yang and J. Moody, "Feature selection based on joint mutual information," in *Proceedings of International ICSC Symposium on Advances in Intelligent Data Analysis*, 1999, pp. 22–25.

[22] K. Kira and L. A. Rendell, "The feature selection problem: traditional methods and a new algorithm," in *Proceedings of the 10th National Conference on Artificial Intelligence*, 1992, pp. 129–134.

[23] G. H. John and P. Langley, "Estimating continuous distributions in bayesian classifiers," in *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence*, 1995, pp. 338–345.

[24] D. W. Aha, D. Kibler, and M. K. Albert, "Instance-based learning algorithms," *Machine Learning*, vol. 6, no. 1, pp. 37–66, 1991.

[25] J. R. Quinlan, "Bagging, boosting, and c4.5," in *Proceedings of the 13th National Conference on Artificial Intelligence*, 1996, pp. 725–730.

[26] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.