Group Based Unsupervised Feature Selection

Kushani Perera[1(✉)], Jeffrey Chan[2], and Shanika Karunasekera[1]

[1] University of Melbourne, Melbourne, VIC 3010, Australia
bperera@student.unimelb.edu.au, karus@unimelb.edu.au
[2] RMIT University, Melbourne, VIC 3000, Australia
jeffrey.chan@rmit.edu.au

Abstract. Unsupervised feature selection is an important task in machine learning applications, yet challenging due to the unavailability of class labels. Although a few unsupervised methods take advantage of external sources of correlations within feature groups in feature selection, they are limited to genomic data, and suffer poor accuracy because they ignore input data or encourage features from the same group. We propose a framework which facilitates unsupervised filter feature selection methods to exploit input data and feature group information simultaneously, encouraging features from different groups. We use this framework to incorporate feature group information into Laplace Score algorithm. Our method achieves high accuracy compared to other popular unsupervised feature selection methods (∼30% maximum improvement of Normalized Mutual Information (NMI)) with low computational costs (∼50 times lower than embedded methods on average). It has many real world applications, particularly the ones that use image, text and genomic data, whose features demonstrate strong group structures.

Keywords: Unsupervised feature selection · Feature groups · $L_{1,1}$ norm minimisation.

1 Introduction

Feature selection is an important task in preparing high dimensional data for machine learning tasks. It improves the prediction accuracy and simplicity of the learning models and reduces the computational costs. Unlike deep learning methods, feature selection identifies the important features that can be interpreted by the humans when explaining AI decisions (E.g.: genes related to certain diseases [12]). Feature selection methods are of two types, supervised and unsupervised, based on the availability of class labels in data. Among them, unsupervised feature selection has wide applicability because data in most real world scenarios are unlabelled. For example, there is a vast amount of text and image data in the web, yet the label information, such as the subject of a tweet, the topic of an image is only rarely available. Due to the unavailability of labels,

H. W. Lauw et al. (Eds.): PAKDD 2020, LNAI 12084, pp. 805–817, 2020.
https://doi.org/10.1007/978-3-030-47426-3_62

unsupervised approach is more challenging than the supervised approach and achieving good accuracy remains a challenge.

Many unsupervised feature selection methods evaluate features using instance-feature data alone, which is available in the form of the data matrix [9, 14]. In contrast, recent work shows that features can be grouped according to various criteria and this group information can improve the usefulness of the feature selection [17]. For example, the nearby pixels in images can be grouped together considering the spatial locality to improve selection of pixels for image analysis. The words in document datasets can be grouped according to their semantics [13] to improve selection of words for document analysis. Genes in genomic data can be grouped using Gene Ontology information [3] to improve bio-marker identification for disease prediction and drug discovery. We show that considering this group structure can enable selection of a better feature subset in real world applications. In Sect. 4, we illustrate this using a concrete text data example.

In contrast to supervised feature selection [11], little work exist in unsupervised feature selection which exploits feature group information. The existing ones are limited to genomic data in which feature selection is limited to simple methods such as selecting the centroids of feature groups [3]. They do not use group information in combination with instance-feature data, which is also useful for feature selection. Hierarchical Unsupervised Feature Selection (HUFS) [17] uses feature group information together with instance-feature data to improve feature selection accuracy and is applicable for different data types. Like many state of the art feature selection methods, HUFS is also an embedded approach, yet embedded methods do not have a significant advantage in unsupervised feature selection due to the unavailability of class labels. Compared to embedded methods, filter methods are fast and produce more generic solutions [15]. Consequently, they are still popular in applications such as bio-marker identification [12] and have growing interest in big data applications [7,16,20].

We propose a framework which helps incorporating feature group information into unsupervised filter feature selection methods. To demonstrate the usefulness of our approach, we incorporate feature group information into Laplace Score (LS) algorithm [9], a well established feature selection method which achieves good accuracy with very low computational costs. We mathematically show that the proposed feature selection objective can be represented as a standard quadratic optimisation problem, such that standard optimisation algorithms can be used to solve the optimisation problem. However, quadratic programming optimisation algorithms are slow and cannot scale to larger problems which are typically encountered, hence we also propose a greedy optimisation method, *Group Laplace Score* (*GLS*), which is faster than quadratic optimisation algorithms, yet show comparable performance. *Through extensive experiments we show that GLS achieves high clustering performance with low computational costs*, compared to existing feature selection methods. Our main contributions are as follows.

- We propose a framework which facilitates unsupervised filter feature selection methods to exploit the knowledge about feature groups to achieve higher clustering performance.
- We use the proposed framework to incorporate feature group information into LS algorithm and propose a new feature selection algorithm, *GLS*.
- We experimentally show that *GLS* obtains significantly higher clustering performance than the existing feature selection algorithms.

2 Related Work

Many unsupervised feature selection methods, both similarity preserving (filter) [9,19] and embedded [6,8,10,14] methods, are based on input data alone and rarely take the advantage of the external sources of knowledge about feature group structures. The feature groups used by some feature selection methods are also formed with input data [15,18]. Some domain specific unsupervised methods [3] are proposed for selecting genes from different gene groups, yet they do not combine group based feature selection with instance-feature data which is also useful for feature selection. In contrast, HUFS uses feature group information to improve the instance-feature data based feature selection and is applicable for different data types. However, HUFS encourages features from the same group which is not effective in most real world applications [11]. In contrast, our method encourages features from different groups and we experimentally show that our method outperforms HUFS in terms of accuracy and efficiency. Compared to HUFS, our method requires less parameter tuning too.

3 Preliminaries

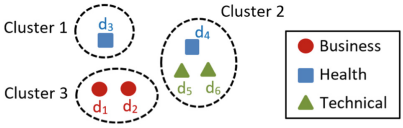
This section discusses some frequently used definitions and terms in the paper. $X \in \mathbb{R}^{n \times m}$ is the input data matrix, where n is the number of instances and m is the number of features in X. F is the set of all features in X, $S \subseteq F$ is the selected feature subset, $f_i \in F$ the i^{th} feature in X and k is the number of features to be selected. G_i is the set of features in i^{th} feature group and r is number of groups. Given a matrix $A \in \mathbb{R}^{n \times m}$, $a_{i,j}$, is its element in i^{th} row and j^{th} column. $L_{1,1}$ norm of A, $\|A\|_{1,1} = \sum_{i=1}^{n} \sum_{j=1}^{m} |a_{i,j}|$.

Definition 1. *The feature indicator matrix, $U \in \mathbb{R}^{m \times m}$, is a diagonal matrix whose i^{th} diagonal entry, $u_{i,i} = u_i = 1$ if the i^{th} feature in X is selected into S and $u_{i,i} = u_i = 0$ otherwise. $u_{i,j} = 0$ $(\forall\ i \neq j)$.*

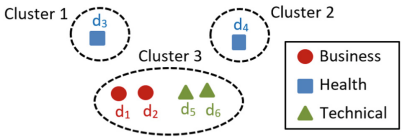
Definition 2. *Given that S is the selected feature subset and G_i is the set of features in i^{th} feature group, $w_i = \frac{No.\ of\ features\ in\ S\ and\ G_i}{No.\ of\ features\ in\ S} = \frac{|S \cap G_i|}{|S|}$.*

	d_1	d_2	d_3	d_4	d_5	d_6
Bank	**13**	**10**	0	0	0	**1**
Patient	0	0	**20**	0	0	0
Cell	0	0	0	**16**	0	0
Google	0	**1**	0	0	**13**	0
Class	**B**	**B**	**H**	**H**	**T**	**T**

(a) Example text dataset. Column (d_i): a document/instance, Row: a word/feature, Class: document type, B: Business, H: Health, T: Technical



(b) Cluster results for {Bank, Patient, Google}



(c) Cluster results for {Bank, Patient, Cell}

Fig. 1. Feature selection in the text dataset in Example 1

4 Motivation and Background

In this section, we demonstrate the importance of external feature group information for feature selection accuracy, using Reuters (RT) text dataset [1] as a concrete example. As the complete dataset is too large, we select only some instances and feature values which are helpful for the discussion.

Example 1: Figure 1a shows a part of the RT dataset in which the words are the features and documents (d_i) are the instances. Feature values represent the occurrence frequency of each word in each document. Each document is one of the three types: Business, Health, Technical, but in the unsupervised feature selection, the algorithm is not provided this. The feature selection problem is to select three features which achieves the best clustering performance.

The features which result in small distances between the same class instances and large distances between different class instances help the same class instances to get clustered together. For example, with respect to "Bank", business documents have lower distances between each other and large distances with the rest (Manhattan distance of 3 between d_1 and d_2 and 13 between d_1 and d_5). Therefore, "Bank" discriminates business documents from the rest. Similarly, "Google" and "Patient" discriminate some technical (d_5) and health (d_3) documents. {Bank, Patient, Google} collectively discriminate between different class instances from one another. Figure 1b shows the k-means (k = 3) cluster assignments for this feature subset. Only d_4 is assigned to a wrong cluster and cluster purities are 1,1, and 0.67. Clustering performance in terms of NMI [9] is 0.74.

In contrast, no feature in {Bank, Patient, Cell} discriminates between business and technical documents and "Patient" and "Cell" cause large distances between the health documents, the same class instances, leading to poor clustering performance. Figure 1c shows that d_4, d_5, d_6 are assigned to wrong clusters, resulting in impure clusters (cluster purities of 1, 1, and 0.5) compared to the previous case. Clustering performance in terms of NMI is 0.65. Therefore, {Bank, Patient, Google} is better compared to {Bank, Patient, Cell}. However, "Cell"

$$\begin{array}{c} \\ d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \end{array} \begin{array}{c} \begin{array}{cccccc} d_1 & d_2 & d_3 & d_4 & d_5 & d_6 \end{array} \\ \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 2 & 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & -1 & -1 & -1 & -1 & 4 \end{bmatrix} \end{array}$$

(a) Laplace Matrix (L)

$$\begin{array}{c} \\ b \\ p \\ c \\ g \end{array} \begin{array}{c} \begin{array}{cccc} b & p & c & g \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

(b) G

$$\begin{array}{c} \\ b \\ p \\ c \\ g \end{array} \begin{array}{c} \begin{array}{cccc} b & p & c & g \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

(c) U for {Bank, Patient, Cell}

$$\begin{array}{c} \\ b \\ p \\ c \\ g \end{array} \begin{array}{c} \begin{array}{cccc} b & p & c & g \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

(d) G' for {Bank, Patient, Cell}

Fig. 2. Matrices for the dataset in Example 1. b: Bank, p: Patient, c: Cell, g: Google

and "Google" have very similar feature value distributions, and class labels are not available for feature selection. Therefore, "Cell" and "Google" cannot be differentiated from one another using instance feature data alone. We show this using LS algorithm, which selects the features which best preserve the locality structure of the instances, as a concrete example.

LS Algorithm: Given that A is the adjacency matrix between the instances, D is the degree matrix and L is the Laplace matrix such that $L = D - A$, the Laplace score of a feature f, $l_f = \frac{\tilde{f}^T L \tilde{f}}{\tilde{f}^T D \tilde{f}}$, where $\tilde{f} = f - \mu_f$ and μ_f is the mean of f. LS objective for selecting k features is shown in Eq. (1). LS algorithm achieves this by selecting the features with k minimum Laplace scores. Figure 2a shows L for RT dataset, assuming a 1-Nearest Neighbour A. Laplace scores for "Bank", "Cell", "Patient" and "Google" are 0.39, 1.06, 1.06 and 1.1, respectively. The selected feature subset is therefore {Bank, Cell, Patient}, which is not optimal.

$$\min_S \sum_{\tilde{f} \in S} \frac{\tilde{f}^T L \tilde{f}}{\tilde{f}^T D \tilde{f}} \text{ subject to } |S| = k \tag{1}$$

Using Feature Group Information: Consider using Wordnet [13] as an external source of knowledge for Example 1. Wordnet shows a high semantic similarity (0.7) between "Cell" and "Patient", and low similarity between other feature pairs (0.1 between "Google" and "Bank"). Three feature groups can be created based on semantic similarity. Group 1: {Bank}, Group 2: {Patient, Cell}, Group 3: {Google}. Encouraging features from different groups results in {Bank, Patient, Google}, which is optimal. This is because *semantically similar words tend to occur in similar types of documents*. Consequently, words from different groups discriminate different types of documents from one another and result in lower distances between the same type of documents. For example, given "Patient", selecting "Google" (from a different group), results in a lower distance between d_3 and d_4 than selecting "Cell" (from the same group). Opposed to "Cell", "Google" also discriminates between business and technical documents.

5 Proposed Method

We propose a framework which facilitates the unsupervised filter feature selection methods to encourage features from different groups and use this framework to incorporate feature group information into LS algorithm. When the feature groups have different importance levels based on factors such as group size and group quality, more features are encouraged from the groups with higher importance. Proposed feature selection objective can be solved using quadratic optimisation methods, but we also propose a greedy approach, *GLS*, which achieves the same performance faster. In this paper, we focus on non-overlapped groups, yet the proposed method can easily be extended to overlapped groups as well.

Modelling Feature Group Information: We define $G \in \mathbb{R}^{m \times m}$, the feature group matrix. If f_i, $f_j \in F$ are in the same group, $g_{i,j} = g_{j,i} = 1$. Otherwise $g_{i,j} = g_{j,i} = 0$. $\forall i = 1, \ldots, m$, $g_{i,i} = 0$. G for Example 1 is shown in Fig. 2b. Multiplying G by U twice makes the rows and columns of G corresponding to the unselected features all zeros. This results in $G' = UGU \in \mathbb{R}^{m \times m}$, feature group matrix of the features in S. The number of zeros in G' increases when the features in S are from different feature groups and all the elements in $G' \geq 0$. Therefore, given that k features are to be selected, to encourage features from different feature groups, our objective is to select U to minimise $\|UGU\|_{1,1}$ subject to $\|U\|_{1,1} = k$.

Figure 2c and d show U and G' when $S = \{\text{Bank, Patient, Cell}\}$, for which $\|G'\|_{1,1} = 2$. When $S = \{\text{Bank, Patient, Google}\}$ U is a diagonal matrix where $diag(U) = [1, 1, 0, 1]$, $G' \in \mathbb{R}^{4 \times 4}$ is a matrix of all zeros and $\|G'\|_{1,1} = 0$. This shows that $\|UGU\|_{1,1}$ is minimal when the features are selected from different groups. When the feature groups have different importance levels, to encourage more features from the groups with higher importance, we set $g_{i,j} = g_{j,i} = \frac{1}{\alpha_i}$ (instead of 1), where α_i *is the weight of* G_i.

Input Data Based Feature Selection: We next propose a common framework to combine group based feature selection with any unsupervised filter feature ranking method. Let Q be a diagonal matrix, where, $q_{i,i} = l_i$, where l_i is the feature score of f_i, in terms of its capability to preserve the sample similarity. $Q' = UQU$ is the feature score matrix for selected features in S. Q' is a diagonal matrix in which $q'_{i,i} = l_i$ if $f_i \in S$ and $q'_{i,i} = 0$ otherwise. Given that $l_i \geq 0$, $\forall\ i$, the feature selection objective is to select U to minimise or maximise $\|UQU\|_{1,1}$ subject to $\|U\|_{1,1} = k$. Minimisation or maximisation is decided based on the algorithm used to compute l_i.

Theorem 1 shows that Laplace score is always non-negative and eligible for Q. Consequently, Eq. (1) can be reformulated as minimising $\|UQU\|_{1,1}$ subject to $\|U\|_{1,1} = k$, where l_i = Laplace score of f_i. For example, in Example 1, $diag(Q)$ = [0.39, 1.06, 1.06, 1.1]. When $S = \{\text{Bank, Patient, Cell}\}$, $diag(Q')$ = [0.39, 1.06, 1.06, 0] and $\|Q'\|_{1,1} = 2.51$. When $S = \{\text{Bank, Patient, Google}\}$, $diag(Q')$ = [0.39, 1.06, 0, 1.1] and $\|Q'\|_{1,1} = 2.55$. Therefore, minimal $\|UQU\|_{1,1}$ is achieved

for {Bank, Patient, Cell}, the same feature subset selected by LS algorithm. For the rest of the paper, we assume l_i is computed using Laplace score, therefore minimise $\|UQU\|_{1,1}$. Maximisation is equivalent to minimising -$\|UQU\|_{1,1}$.

Theorem 1. *Given that l_i is the Laplace score of $f_i \in F$, $l_i \geq 0, \forall\ i = 1, \cdots, m$.*

Proof. Because L and D are positive definite. Refer to this link[1] for the proof.

Feature Selection Objective: The feature selection objective which combines both group based feature selection and input data based feature selection is shown in Eq. (2). λ is a user defined parameter. In this paper, we assign a fixed value for λ. In future, we plan to iteratively decide λ value for each feature selected. Based on Theorem 2, we reformulate Eq. (2) into Eq. (3).

$$\min_U \|UQU\|_{1,1} + \lambda \|UGU\|_{1,1} \text{ subject to } \|U\|_{1,1} = k \tag{2}$$

$$\min_U \|U(Q + \lambda G)U\|_{1,1} \text{ subject to } \|U\|_{1,1} = k \tag{3}$$

Theorem 2. *Given $\lambda \geq 0$, $\|UQU\|_{1,1} + \lambda \|UGU\|_{1,1} = \|U(Q + \lambda G)U\|_{1,1}$*

Proof. Because $u_{i,j}$, $q_{i,j}$, $g_{i,j} \geq 0\ \forall\ i, j$. Refer to this link (See footnote 1) for the proof.

Given $u = [u_1, \cdots, u_m]^T$, where u_i is the i^{th} diagonal element of U, Theorem 3 shows that $\|U(Q + \lambda G)U\|_{1,1}$ can be reformulated as a quadratic function of u. Therefore, to solve Eq. (3), we use two approaches: (1) Standard Quadratic Programming (QP) methods (2) Greedy method (GLS algorithm). As the QP method, we use the MATLAB inbuilt "fmincon" function with "interior point" method, but omitted the details due to space limitations. Please refer to this link (See footnote 1) for details. The greedy method showed comparable accuracy to QP method, yet faster. Therefore, in this paper, we focus on the greedy method.

Theorem 3. *Given that $H = Q + \lambda G$, and u as defined above, $\|UHU\|_{1,1} = u^T H u = h(u)$, that is $\|UHU\|_{1,1}$ is a quadratic function of u.*

Proof. Please refer to this link (See footnote 1) for the proof.

Greedy Method: As discussed, $\|U(Q + \lambda G)U\|_{1,1} = \|UHU\|_{1,1} = h(u)$. At each Iteration t, *GLS* selects a feature, f_t, such that $f_t = \text{argmin}_{f_x \in S'_{t-1}} h(u_t) - h(u_{t-1})$, where u_{t-1} and u_t are the selected feature indicator vectors (u) after Iteration $(t-1)$ and t, respectively and S'_{t-1} is the unselected feature subset after Iteration $t-1$. According to Theorem 4, this is equivalent to selecting $f_t = \text{argmin}_{f_x \in S'_{t-1}} l_x + \lambda \frac{w_i}{\alpha_i}$, where f_x is any feature in S'_{t-1}, l_x is the Laplace score of f_x, G_i the feature group of f_x, α_i is the weight of G_i, $w_i = \frac{|S_{t-1} \cap G_i|}{S_{t-1}}$ and S_{t-1} is the selected feature subset after Iteration $t-1$. Therefore, as shown in Algorithm 1, *GLS* selects f_x to minimise this quantity (Line 5), which *avoids complex matrix multiplication operations.*

[1] https://sites.google.com/view/kushani/publications.

Algorithm 1: GLS algorithm

input : Dataset (X), Requested feature count (k), Group weights ($\alpha_1 \cdots \alpha_r$)
output: Selected feature subset (S)

1 $S' \leftarrow F$ in X; $S \leftarrow \emptyset$; $fCount \leftarrow 0$; $n_1 \cdots n_r \leftarrow 0$;
2 **while** $fCount < k$ **do**
3 **for** $x \in S'$ **do**
4 $i \leftarrow$ Group index of G_i where $x \in G_i$;
5 $score_x \leftarrow l_x + \lambda \frac{w_i}{\alpha_i}$;
6 **end**
7 $f_{min} \leftarrow \text{argmin}_{x \in S'}\ score_x$;
8 $S \leftarrow S + f_{min}$; $S' \leftarrow S' - f_{min}$;
9 $j \leftarrow$ Group index of G_j where $f_{min} \in G_j$;
10 n_j++; $w_j \leftarrow \frac{n_j}{|S|}$; $fCount$++;
11 **end**
12 return S;

Theorem 4. *Given that* S_{t-1}, S'_{t-1}, u_{t-1}, u_t, $f_x \in S'_{t-1}$, l_x, w_i *and* α_i *are as defined above,* $\text{argmin}_{f_x \in S'_{t-1}}\ h(u_t) - h(u_{t-1}) = \text{argmin}_{f_x \in S'_{t-1}}\ l_x + \lambda \frac{w_i}{\alpha_i}$.

Proof. Refer to this link (See footnote 1) for the proof.

Example 1 Revisited: We apply *GLS* for Example 1, given the feature groups created in Sect. 4. $\lambda = 1$, $\alpha_i = 1\ \forall = i$. *GLS* first selects "Bank" which has the minimum Laplace score (0.39). In Iteration 2, for all remaining features, $w_i = 0$. Therefore, *GLS* selects "Patient" or "Cell", which has next minimum Laplace score (1.06). Assume it selects "Patient". In Iteration 3, for "Cell" and "Google", $w_i = 0.5$ and 0, respectively and $l_i + \lambda\ \frac{w_i}{\alpha_i} = 1.56$ and 1.1, respectively. *GLS* selects "Google" which has minimal feature score. Therefore, the selected feature subset is {Bank, Patient, Google}, which is optimal according to Sect. 4.

Computation Complexity Analysis: Given F and S are as defined in Sect. 3, time complexity for computing the Laplace score is $O(|F|)$. The complexity of the iterative group based feature selection (Line 2–11 in Algorithm 1), is $O(|S||F|)$. As $|S| << |F|$, the time complexity of *GLS* is linear to $|F|$.

6 Experimental Evaluation

In this section, we discuss the experimental results obtained by *GLS* algorithm.

Datasets: We evaluate *GLS*, using real datasets, which are benchmark datasets used to test group based feature selection. Table 1 shows a summary of them. Yale, ORL and COIL20 have a 32×32 pixel map and USPS a 16×16 pixel map.

Feature Grouping: To introduce spatial locality information, which is not available from the input data matrix alone, we partition the pixel map of an image into $p \times p$ non overlapping squares. Each square is a feature group. Default p for USPS is 2 and 4 for other image datasets. In text data, pairwise semantic similarities between the words are found using WordNet [13] and words are clustered based on the similarity values, using spectral clustering. We use only 2,468 words, available in WordNet. Genes in genomic data are clustered based on Gene Ontology information as discussed in [3]. Number of groups is set to 0.04 of the original feature set based on the previous findings for MT dataset [3].

Table 1. Dataset description. m: # features, n: # instances, c: # classes

Dataset	m	n	c	**Type**	**Dataset**	m	n	c	**Type**
Multi-Tissue (MT) [2]	1,000	103	4	Genomic	Yale [5]	1,024	165	15	Image
CNS [2]	989	42	5	Genomic	ORL [5]	1,024	400	40	Image
DLBCL-B [2]	661	180	3	Genomic	COIL20 [4]	1,024	1,440	20	Image
Multi-B [2]	5,565	32	4	Genomic	USPS [4]	256	9,298	10	Image
Reuters (RT) [1]	3,068	294	6	Text					

Baselines: As baselines, we use LS algorithm and Spectral Feature Selection SPEC [19] as similarity preserving methods and Multi Cluster Feature Selection (MCFS) [6], Robust Unsupervised Feature Selection (RUFS) [14] and HUFS as embedded methods. RUFS has proven high performance compared to many existing embedded methods and HUFS uses feature group information similar to our method. RUFS and MCFS use two different approaches to control feature redundancy ($L_{2,1}$ norm vs. L_1 norm). k-medoid (KM) [3] is specific for genomic datasets, therefore, we use it with genomic data only. For HUFS, we consider the complete pixel hierarchy as described in [17].

Evaluation Criteria: We consider the clustering performance as the measure of feature selection accuracy and evaluate it in terms of NMI [9]. k-means is the cluster method used. It is run 20 times and we report the average *NMI*. SD is the standard deviation of NMI obtained for the 20 iterations. *Average accuracy of an algorithm in a dataset* is the average of the NMIs obtained for all the selected feature numbers in that dataset. We select features up to the point all algorithm accuracies converge. Algorithm run times are measured in seconds.

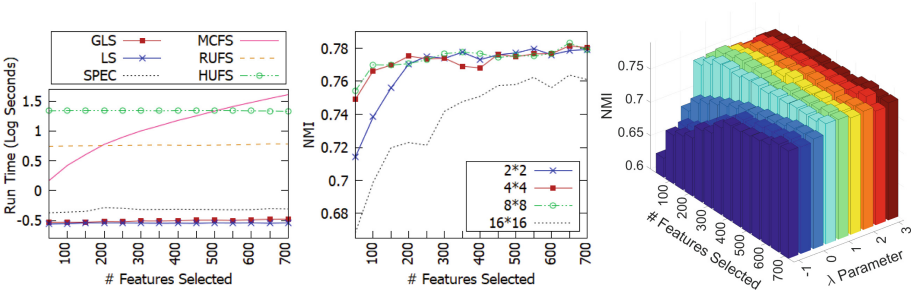
Experimental Setup: We split each dataset, 60% instances for training set and 40% for test test, using stratified random sampling method and remove the class labels from both. We perform feature selection on the training dataset and evaluate the clustering performance of the test set, using only the selected feature subset. By default, $\alpha_i = 1$ for all feature groups and $\lambda = 1$.

Table 2. Comparison of the clustering performances of different algorithms. Row 1: maximum NMI of each algorithm for each dataset. The highest maximum NMI for each dataset is in bold letters. Row 2 (±): SD corresponding to maximum NMI. Row 3 (x): the number of features at which the maximum NMI is achieved. Row 4: Algorithm rankings in terms of average accuracy (1 corresponds to the highest average accuracy)

	Yale	ORL	COIL20	USPS	RT	MT	CNS	DLBCL-B	Multi-B
GLS	**0.69** ±0.01	**0.82** ±0.01	**0.78** ±0.01	0.62 ±0.00	**0.34** ±0.03	**0.76** ±0.00	**0.71** ±0.04	0.49 ±0.02	**0.74** ±0.00
	(400)	(450)	(200)	(200)	(40)	(20)	(15)	(200)	(40)
	1	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
LS	0.67 ±0.02	0.82 ±0.01	0.78 ±0.01	**0.63** ±0.01	0.31 ±0.04	0.64 ±0.00	0.62 ±0.07	**0.5** ±0.02	0.69 ±0.00
	(300)	(900)	(850)	(150)	(35)	(120)	(120)	(180)	(50)
	3	5	5	2	2	5	5	4	3
SPEC	0.67 ±0.02	0.82 ±0.01	0.78 ±0.01	0.62 ±0.00	0.31 ±0.03	0.68 ±0.03	0.59 ±0.06	0.48 ±0.03	0.42 ±0.00
	(900)	(850)	(750)	(240)	(40)	(100)	(50)	(180)	(10)
	2	6	4	6	3	6	6	2	7
MCFS	0.67 ±0.01	0.82 ±0.01	0.78 ±0.01	0.62 ±0.00	0.32 ±0.04	0.76 ±0.00	0.66 ±0.04	0.27 ±0.04	0.71 ±0.01
	(750)	(750)	(450)	(240)	(85)	(60)	(130)	(180)	(15)
	4	3	3	3	4	2	3	3	2
RUFS	0.67 ±0.02	0.82 ±0.01	0.78 ±0.01	0.62 ±0.00	0.22 ±0.01	0.74 ±0.05	0.69 ±0.05	0.37 ±0.07	0.66 ±0.00
	(1000)	(700)	(350)	(240)	(5)	(30)	(10)	(300)	(50)
	6	2	2	5	6	4	2	7	4
HUFS	0.67 ±0.02	0.81 ±0.01	0.77 ±0.01	0.62 ±0.00	0.28 ±0.04	0.63 ±0.00	0.58 ±0.03	0.34 ±0.07	0.57 ±0.00
	(1000)	(650)	(900)	(240)	(90)	(140)	(110)	(240)	(55)
	5	4	6	4	5	3	4	5	6
KM	-	-	-	-	-	0.68 ±0.02	0.41 ±0.02	0.17 ±0.05	0.57 ±0.02
						(30)	(20)	(280)	(75)
						7	7	6	5

Experiment 1 evaluates the clustering performance of different algorithms for different numbers of selected features. **Experiment 2** evaluates the clustering performance of *GLS* in text and genomic data, for $\alpha_i = \frac{|G_i|}{|F|}$ and $\alpha_i = 1 \; \forall \; i$. This tests the *effect of group weights* on clustering performance. **Experiment 3** executes each feature selection algorithm 100 times and reports the log value of the average run time to evaluate the *algorithm efficiency*. **Experiment 4**

performs feature selection in image datasets for $p = 2, 4, 8, 16$. This tests the *effect of the group size* on the clustering performance. **Experiment 5** runs *GLS* for $\lambda \in$ [-1, 3]. This tests the *effect of* λ on the clustering performance.



(a) Execution time of different algorithms

(b) Different feature group sizes: $p \times p$ (p = 2,4,8,16)

(c) Different λ values

Fig. 3. *GLS* execution time and accuracy variation for different settings for COIL20

Experimental Results: Table 2 shows that *GLS achieves the highest NMI over baselines* in 7 out of 9 datasets. In ORL and COIL20, *GLS* achieves the highest NMI with a smaller number of features than baselines. *In all datasets, GLS has the highest average accuracy* (rank 1), yet the rankings of baselines vary across the datasets. *GLS*'s average NMI gain over SPEC in Multi-B dataset is ∼30%, which is its maximum NMI gain over baselines. Maximum NMI gain of *GLS* over the NMI obtained by the complete feature set is 3%, 1%, 1%, 2%, 10%, 11%, 4%, 12% and 24% for Yale, ORL, COIL20, USPS, RT, MT, CNS, DLBCL-B and Multi-B respectively. *GLS*'s average accuracy gains for $\alpha_i = \frac{|G_i|}{|F|}$ over $\alpha_i = 1$ are 0.3% and 3% in RT and DLBCL-B datasets, respectively. Due to space limitations, we omit the results graphs for Experiment 1 and 2. Please refer to this link (See footnote 1) to see all the results graphs. *GLS also has the lowest SD for clustering performance* for 7 out of 9 datasets. Figure 3a shows that *GLS has only little increase of run time than LS, which is significantly low compared to embedded methods*. For COIL20 dataset, the run time of *GLS* is ∼50, ∼20 and ∼70 times lower than the run time of MCFS, RUFS and HUFS. Figure 3b shows that compared to large and small feature groups (p = 2, 16), *GLS* performance for medium sized groups (p = 4, 8) is high. According to Fig. 3c, *clustering performance is less sensitive to* λ for $\lambda > 0$, yet significantly low for $\lambda \leq 0$.

Evaluation Insights: Compared to baselines, *GLS consistently shows high clustering performance* for all the datasets (highest average accuracy in all datasets and maximum accuracy in 7 out of 9 datasets), *with low computational costs* (∼50 times lower run time than embedded methods on average). *In*

all datasets, GLS achieves higher accuracy than using the complete feature set, with a comparatively smaller number of features. Higher accuracy obtained by weighted feature groups show that in some cases, knowledge about the importance level of different feature groups improves the accuracy of *GLS*. Low SD values for NMI show that *GLS produces more stable clusters and more precise performance results* than the baselines. Medium sized groups achieve higher accuracy because large and small groups more resemble the case of no groupings. This demonstrates the contribution of feature group information to achieve high accuracy. Low accuracy for $\lambda \leq 0$ supports our hypothesis that *selecting features from the same group is less effective than selecting from different groups. Less parameter tuning is required* for *GLS* as its accuracy is less sensitive to λ (> 0).

7 Conclusion

We propose a framework which facilitates exploiting feature group information by unsupervised feature selection methods and use this framework to incorporate feature group information into LS algorithm. We show that compared to baselines, the proposed method achieves high clustering performance for the datasets with feature group structures with low computational costs and requires less parameter tuning. Our future work includes using the proposed framework for unsupervised feature selection methods other than the LS algorithm.

Acknowledgements. This work is supported by the Australian Government.

References

1. 3 sources. http://mlg.ucd.ie/datasets/3sources.html. Accessed Nov 2019
2. Cancer program datasets. http://portals.broadinstitute.org/cgi-bin/cancer/datasets.cgi. Accessed Nov 2019
3. Acharya, S., Saha, S., Nikhil, N.: Unsupervised gene selection using biological knowledge: application in sample clustering. BMC Bioinform. **18**(1), 513 (2017)
4. Cai, D., He, X., Han, J.: Speed up kernel discriminant analysis. VLDB J. **20**(1), 21–33 (2011). https://doi.org/10.1007/s00778-010-0189-3
5. Cai, D., He, X., Hu, Y., et al.: Learning a spatially smooth subspace for face recognition. In: Proceedings of IEEE CVPR 2007, pp. 1–7 (2007)
6. Cai, D., Zhang, C., He, X.: Unsupervised feature selection for multi-cluster data. In: Proceedings of the 16th ACM SIGKDD, pp. 333–342. ACM (2010)
7. Cai, J., Luo, J., Wang, S., et al.: Feature selection in machine learning: a new perspective. Neurocomputing **300**, 70–79 (2018)
8. Guo, J., Zhu, W.: Dependence guided unsupervised feature selection. In: 32nd AAAI (2018)
9. He, X., Cai, D., Niyogi, P.: Laplacian score for feature selection. In: NIPS, pp. 507–514 (2006)
10. Hou, C., Nie, F., Li, X., et al.: Joint embedding learning and sparse regression: a framework for unsupervised feature selection. IEEE Trans. Cybern. **44**(6), 793–804 (2014). https://doi.org/10.1109/TCYB.2013.2272642

11. Kong, D., Fujimaki, R., Liu, J., et al.: Exclusive feature learning on arbitrary structures via $l_{1,2}$-norm. In: NIPS, pp. 1655–1663 (2014)
12. Lazar, C., Taminau, J., Meganck, S., et al.: A survey on filter techniques for feature selection in gene expression microarray analysis. IEEE/ACM TCBB **9**(4), 1106–1119 (2012). https://doi.org/10.1109/TCBB.2012.33
13. Miller, G.A.: WordNet: a lexical database for English. Commun. ACM **38**(11), 39–41 (1995). https://doi.org/10.1145/219717.219748
14. Qian, M., Zhai, C.: Robust unsupervised feature selection. In: IJCAI, pp. 1621–1627 (2013)
15. Sahu, B., Dehuri, S., Jagadev, A.K.: Feature selection model based on clustering and ranking in pipeline for microarray data. Inform. Med. Unlocked IMU **9**, 107–122 (2017)
16. Wang, L., Wang, Y., Chang, Q.: Feature selection methods for big data bioinformatics: a survey from the search perspective. Methods **111**, 21–31 (2016)
17. Wang, S., Wang, Y., Tang, J., et al.: Exploiting hierarchical structures for unsupervised feature selection. In: Proceedings of the 2017 SDM, pp. 507–515. SIAM (2017). https://doi.org/10.1137/1.9781611974973.57
18. Zaharieva, M., Breiteneder, C., Hudec, M.: Unsupervised group feature selection for media classification. Int. J. Multimed. Inf. Retr. **6**(3), 233–249 (2017). https://doi.org/10.1007/s13735-017-0126-y
19. Zhao, Z., Liu, H.: Spectral feature selection for supervised and unsupervised learning. In: Proceedings of the 24th ICML, pp. 1151–1157. ACM (2007)
20. Zou, Q., Zeng, J., Cao, L., et al.: A novel features ranking metric with application to scalable visual and bioinformatics data classification. Neurocomputing **173**, 346–354 (2016). https://doi.org/10.1016/j.neucom.2014.12.123