Utility metric for unsupervised feature selection

Amalia Villa, Abhijith Mundanad Narayanan, Sabine Van Huffel, Alexander Bertrand, Carolina Varon

***Abstract*—Feature selection techniques are very useful approaches for dimensionality reduction in data analysis. When these applications lack annotations, unsupervised methods are required for their analysis. These methods generally consist of two stages: manifold learning and subset selection. In the first stage, the underlying structures in the high-dimensional data are extracted, to further on replicate them with a subset of the features. This subset selection is generally performed making use of sparse regression techniques.**

In this work, the use of a backwards greedy approach based on the Utility metric for Unsupervised feature selection (U2FS) is proposed. This variable selection metric is demonstrated to achieve comparable results to the least absolute shrinkage and selection operator (LASSO) at a much lower computational cost. Furthermore, the effect of non-linearities in the stage of spectral embedding for manifold learning is explored, making use of a radial basis function (RBF) kernel. An alternative solution for the estimation of the kernel parameter for cases presenting high-dimensional data is presented. The proposed unsupervised method succeeds in selecting the correct features in a simulation environment, and its performance in benchmark datasets is comparable to the state-of-the-art, while reducing the computational time required.

***Index Terms*—Unsupervised feature selection, dimensionality reduction, manifold learning, kernel methods.**

I. Introduction

Many applications of data science require the study of highly multi-dimensional data. A high number of dimensions implies a high computational cost as well as a large amount of memory required. Furthermore, this often leads to problems related to the curse of dimensionality [1] and thus, to irrelevant and redundant data for machine learning algorithms [2]. Therefore, it is crucial to perform dimensionality reduction before analyzing the data.

There are two types of dimensionality reduction techniques. On the one hand, feature selection, where the aim is to keep a subset of the original features. On the other hand, transformation techniques define a new smaller set of features, which are derived from a combination of all features of the original set. Some examples of these are Principal Component Analysis (PCA) [3] and Independent Component Analysis(ICA) [4]. These methods lead to less interpretable results, in which the direct relationship between the features and the results is lost.

In this work, the focus is on unsupervised feature selectors. Since these methods do not rely on the availability of labels or annotations in the data, the information comes from the learning of the underlying structure of the data. Despite this challenge, the generalization capabilities of these methods are typically better than for supervised or semi-supervised methods [5].

One specific type of unsupervised feature selectors are those based on manifold and sparse learning [6]. These methods rely on graph theory to learn the underlying structures of the data [7]–[15]. However, to the best of our knowledge, none explores specifically the behavior of these methods with data presenting non-linear relationships between the features (i.e., dimensions). While the graph definition step can make use of kernels to tackle non-linearities, these can be heavily affected by the curse of dimensionality, since they are often based on a distance metric [16].

After the manifold learning stage, sparse regression is applied to score the relevance of the features in the structures present in the graph. These formulations make use of sparsity-inducing regularization techniques to provide the final subset of features selected, and thus, they are highly computationally expensive. These methods are often referred to as structured sparsity-inducing feature selectors (SSFS).

In this work, an efficient unsupervised feature selector is presented, with the goal of achieving comparable results to state-of-the-art approaches, while reducing the computational time. Thus, this method is recommended for applications in which the focus is on investing low computational resources, while obtaining interpretable and accurate results. The main novelty of this work is the use of the Utility metric for Unsupervised feature selection (U2FS). The utility metric was originally proposed in the framework of supervised learning [17] and has been used for channel selection in applications such as electroencephalography (EEG) [18], sensor networks [19], and microphone arrays [20]. Furthermore, motivated by the effect of the curse of dimensionality in kernel methods, a

Manuscript received on July 14, 2020; revised ... This work has received funding from FWO project G0A4918N. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 802895). This research received funding from the Flemish Government (AI Research Program). SVH, AB, AM, AV are affiliated to Leuven.AI - KU Leuven institute for AI, B-3000, Leuven, Belgium. This work was supported in part by Bijzonder Onderzoeksfonds KU Leuven (BOF): The effect of perinatal stress on the later outcome in preterm babies: C24/15/036, Prevalentie van epilepsie en slaapstoornissen in de ziekte van Alzheimer: C24/18/097. Agentschap Innoveren en Ondernemen (VLAIO) 150466: OSA+ and O& O HBC 2016 0184 eWatch . KU Leuven Stadius acknowledges the financial support of imec, and EU H2020 MSCA-ITN-2018: INtegrating Magnetic Resonance SPectroscopy and Multimodal Imaging for Research and Education in MEDicine (INSPiRE-MED), funded by the European Commission under Grant Agreement no. 813120. EU H2020 MSCA-ITN-2018: 'INtegrating Functional Assessment measures for Neonatal Safeguard (INFANS)', funded by the European Commission under Grant Agreement no. 813483. EIT 19263 – SeizeIT2: Discreet Personalized Epileptic Seizure Detection Device. Part of the resources and services used in this work were provided by the VSC (Flemish Supercomputer Center), funded by the Research Foundation - Flanders (FWO) and the Flemish Government.

A. Villa, A. Mundanad Narayanan, S. Van Huffel, A. Bertrand and C. Varon are with the STADIUS Center for Dynamical Systems, Signal Processing and Data Analytics, Department of Electrical Engineering. KU Leuven, Leuven B-3001, Belgium. (email: amalia.villagomez@esat.kuleuven.be; abhijith@esat.kuleuven.be; sabine.vanhuffel@esat.kuleuven.be; alexander.bertrand@esat.kuleuven.be; carolina.varon@esat.kuleuven.be). C. Varon is also with the Circuits and Systems (CAS) group, Delft University of Technology, The Netherlands.

new approximation of the model parameter in the radial basis function (RBF) kernel is proposed.

The rest of the paper is structured as follows. Section II summarizes previous algorithms on SSFS. In Section III the proposed U2FS method is described: first the manifold learning stage, together with the algorithm proposed for the selection of the kernel parameter; and further on, the utility metric is discussed and adapted to feature selection. The experiments performed in simulations and benchmark databases, as well as the results obtained are described in Section IV, and discussed in Section V. Section VI presents the conclusion.

II. Related Work

Spectral feature selection methods have become widely used in unsupervised learning applications for high-dimensional data. This is due to two reasons. On the one hand, the use of manifold learning guarantees the preservation of local structures present in the high-dimensional data. Additionally, its combination with feature selection techniques not only reduces the dimensionality of the data, but also guarantees interpretability.

Spectral feature selectors learn the structures present in the data via connectivity graphs obtained in the high-dimensional space [22]. The combination of manifold learning and regularization techniques to impose sparsity, allows to select a subset of features from the original dataset that are able to describe these structures in a smaller dimensional space.

Most of these algorithms can also be categorized as sparsity-inducing feature selectors, since they make use of sparsity-inducing regularization approaches to stress those features that are more relevant for data separation. The sparsity of these approaches is controlled by different statistical norms ($l_{r,p}$-norms), which contribute to the generalization capability of the methods, adapting them to binary or multi-class problems [23]. One drawback of these sparse regression techniques is that generally, they rely on optimization methods, which are computationally expensive.

The Laplacian Score [7] was the first method to perform spectral feature selection in an unsupervised way. Based on the Laplacian obtained from the spectral embedding of the data, it obtains a score based on locality preservation. SPEC [8] is a framework that contains this previous approach, but it additionally allows for both supervised or unsupervised learning, including other similarity metrics, as well as other ranking functions. These approaches evaluate each feature independently, without considering feature interactions. These interactions are, however, taken into account in Multi-Cluster Feature Selection (MCFS) [9], where a multi-cluster approach is defined based on the eigendecomposition of a similarity matrix. The subset selection is performed applying an l_1-norm regularizer to approximate the eigenvectors obtained from the spectral embedding of the data inducing sparsity. In UDFS [10] the l_1-norm regularizer is substituted by a $l_{2,1}$-norm to apply sample and feature-wise constraints, and a discriminative analysis is added in the graph description. In NDFS [11], the use of the $l_{2,1}$-norm is preserved, but a non-negative constraint is added to the spectral clustering stage.

The aforementioned algorithms perform manifold learning and subset selection in a sequential way. However, other methods tackle these simultaneously, in order to adaptively change the similarity metric or the selection criteria regarding the error obtained between the original data and the new representation. Examples of these algorithms are JELSR [12], SOGFS [13] and (R)JGSC [14], and all make use of an $l_{2,1}$-norm. Most recently, the SAMM-FS algorithm was proposed [15], where a combination of similarity measures is used to build the similarity graph, and the $l_{2,0}$-norm is used for regression.

In summary, several algorithms have been proposed in the literature, combining different spectral embeddings and sparse regression techniques. However, the methods available are not connected to the specific type of data they are suitable to, so the selection of one of these methods is challenging for the user [23]. Additionally, the bottleneck of these methods in terms of computational time is the feature subset selection stage based on sparse regression. In this work, the use of the utility metric for unsupervised feature selection provides a solution for cases where quick and interpretable results are needed, in cases with more samples than features and vice-versa.

III. Proposed Method

This section describes the proposed U2FS algorithm, which focuses on selecting the relevant features in an unsupervised way, at a relatively small computational cost. The method is divided in three parts. In Section III-A, the suggested manifold learning approach is explained, where an embedding based on binary weighting and the RBF kernel are used. In Section III-B a method to select the kernel parameter of the RBF kernel is proposed, specially designed for high-dimensional data. Once the manifold learning stage is explained, Section III-C presents the Utility metric as a new approach for subset selection.

A. Manifold learning considering non-linearities

Given is a data matrix $\mathbf{X} \in \mathbb{R}^{N \times d}$, with $\mathbf{X} = [\mathbf{x}_1; \mathbf{x}_2; \ldots; \mathbf{x}_N]$, $\mathbf{x}_i = [x_i^{(1)}, x_i^{(2)}, \ldots, x_i^{(d)}]$, $i = 1, \ldots, N$, N the number of data points, and d the number of features (i.e., dimensions) in the data. The aim is to learn the structure hidden in the d-dimensional data and approximate it with only a subset of the original features. In this paper, this structure will be identified by means of clustering, where the dataset is assumed to be characterized by c clusters.

In spectral clustering, the clustering structure of this data can be obtained by studying the eigenvectors derived from a Laplacian built from the original data [24][25]. The data is represented using a graph $G = (\mathscr{V}, \mathscr{E})$. $\mathscr{V}$ is the set of vertices $\mathbf{v}_i$, $i = 1, \ldots, N$ where $\mathbf{v}_i = \mathbf{x}_i$. $\mathscr{E} = \{e_{ij}\}$ with $i = 1, \ldots, N$ $j = 1, \ldots, N$ is the set of edges between the vertices where $\{e_{ij}\}$ denotes the edge between vertices v_i and v_j. The weight of these edges is determined by the entries $w_{ij} \geq 0$ of a similarity matrix $\mathbf{W}$. We define the graph as undirected. Therefore, the similarity matrix $\mathbf{W}$, is symmetric (since $w_{ij} = w_{ji}$, with the diagonal set to $w_{ii} = 0$).

Typically, $\mathbf{W}$ is computed after coding the pairwise distances between all N data points. There are several ways of doing this, such as calculating the k-nearest neighbours (KNN) for each point, or choosing the ε-neighbors below a certain distance [26].

In this paper, two similarity matrices are adopted inspired by the work in [9], namely a binary one and one based on an RBF kernel. The binary weighting is based on KNN, being $w_{ij} = 1$ if and only if vertex i is within the K closest points to vertex j. Being a non-parametric approach, the binary embedding allows to simply characterize the connectivity of the data.

Additionally, the use of the RBF kernel is considered, which is well suited for non-linearities and allows to characterize complex and sparse structures [24]. The RBF kernel is defined as $K(\mathbf{x}_i, \mathbf{x}_j) = exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma^2)$. The selection of the kernel parameter σ is a long-standing challenge in machine learning. For instance, in [9], σ^2 is defined as the mean of all the distances between the data points. Alternatively, a rule of thumb, uses the sum of the standard deviations of the data along each dimension [27]. However, the estimation of this parameter is highly influenced by the amount of features or dimensions in the data, making it less robust to noise and irrelevant features. In this work, a new and better informed method to approximate the kernel parameter is proposed (see section III-B).

The graph G, defined by the similarity matrix $\mathbf{W}$, can be partitioned into multiple disjoint sets. Given the focus on multi-cluster data of our approach, the k-Way Normalized Cut (*NCut*) Relaxation is used, as proposed in [28]. In order to obtain this partition, the degree matrix $\mathbf{D}$ of $\mathbf{W}$ must be calculated. $\mathbf{D}$ is a diagonal matrix for which each element on the diagonal is calculated as $D_{ii} = \sum_j W_{i,j}$. The normalized Laplacian $\mathbf{L}$ is then obtained as $\mathbf{L} = \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$, as suggested in [24]. The vectors $\mathbf{y}$ embedding the data in $\mathbf{L}$ can be extracted from the eigenvalue problem [29]:

$$\mathbf{L}\mathbf{y} = \lambda\mathbf{y} \tag{1}$$

Given the use of a normalized Laplacian for the data embedding, the vectors $\mathbf{y}$ must be adjusted using the degree matrix $\mathbf{D}$:

$$\alpha = \mathbf{D}^{1/2}\mathbf{y}, \tag{2}$$

which means that α is the solution of the generalized eigenvalue problem of the pair $\mathbf{W}$ and $\mathbf{D}$. These eigenvectors α are a new representation of the data, that gathers the most relevant information about the structures appearing in the high-dimensional space. The c eigenvectors, corresponding to the c highest eigenvalues (after excluding the largest one), can be used to characterize the data in a lower dimensional space [28]. Thus, the matrix $\mathbf{E} = [\alpha_1, \alpha_2, \ldots, \alpha_c]$ containing column-wise the c selected eigenvectors, will be the low-dimensional representation of the data to be mimicked using a subset of the original features, as suggested in [9].

B. Kernel parameter approximation for high-dimensional data

One of the most used similarity functions is the RBF kernel, which allows to explore non-linearities in the data. Nevertheless, the kernel parameter σ^2 must be selected correctly, to avoid overfitting or the allocation of all data points to the same cluster. This work proposes a new approach to approximate this kernel parameter, which will be denoted by $\hat{\sigma}^2$ when derived from our method. This method takes into account the curse of dimensionality and the potential irrelevant features or dimensions in the data.

As a rule of thumb, σ^2 is approximated as the sum of the standard deviation of the data along each dimension [27]. This approximation grows with the number of features (i.e. dimensions) of the data, and thus, it is not able to capture its underlying structures in high-dimensional spaces. Nevertheless, this σ^2 is commonly used as an initialization value, around which a search is performed, considering some objective function [27], [30].

The MCFS algorithm skips the search around an initialization of the σ^2 value by substituting the sum of the standard deviations by the mean of these [9]. By doing so, the value of σ^2 does not overly grow. This estimation of σ^2 suggested in [9] will be referred to as σ_0^2. A drawback of this approximation in high-dimensional spaces is that it treats all dimensions as equally relevant for the final estimation of σ_0^2, regardless of the amount of information that they actually contain.

The aim of the proposed approach is to provide a functional value of σ^2 that does not require any additional search, while being robust to high-dimensional data. Therefore, this work proposes an approximation technique based on two factors: the distances between the points, and the number of features or dimensions in the data.

The most commonly used distance metric is the euclidean distance. However, it is very sensitive to high-dimensional data, deriving unsubstantial distances when a high number of features is involved in the calculation [16]. In this work, the use of the Manhattan or taxicab distance [31] is proposed, given its robustness when applied to high-dimensional data [16]. For each feature l, the Manhattan distance δ_l is calculated as:

$$\delta_l = \frac{1}{N}\sum_{i,j=1}^{N} |x_{il} - x_{jl}| \tag{3}$$

Additionally, in order to reduce the impact of irrelevant or redundant features, a system of weights is added to the approximation of $\hat{\sigma}^2$. The goal is to only take into account the distances associated to features that contain relevant information about the structure of the data. To calculate these weights, the probability density function (PDF) of each feature is compared with a Gaussian distribution. Higher weights are assigned to the features with less Gaussian behavior, i.e. those the PDF of which differs the most from a Gaussian distribution. By doing so, these will influence more the final $\hat{\sigma}^2$ value, since they allow a better separation of the structures present in the data.

Figure 1 shows a graphical representation of this estimation. The dataset in the example has 3 dimensions or features: f_1, f_2 and f_3. f_1 and f_2 contain the main clustering information, as it can be observed in Figure 1a, while f_3 is a noisy version of f_1, derived as $f_3 = f_1 + 1.5n$, where n is drawn from a

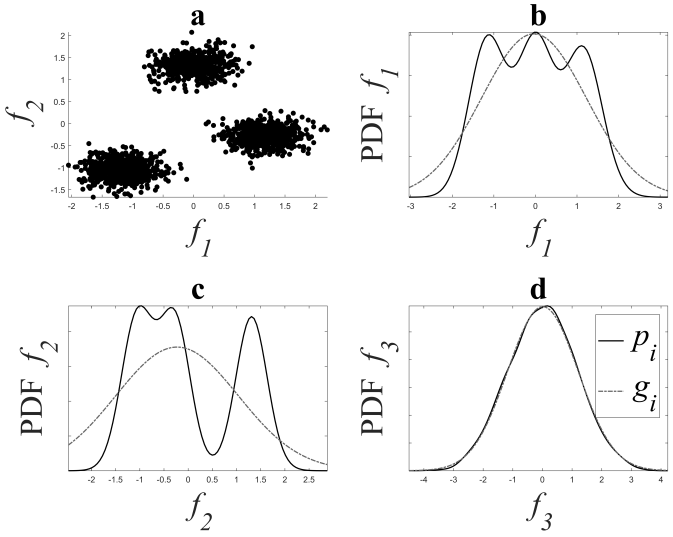

Fig. 1: Weight system for relevance estimation. In Figure 1a, f_1 and f_2 can be seen. 1b, 1c and 1d show in black the PDFs p_i of f_1, f_2 and f_3 respectively, and in grey dotted line their fitted Gaussian g_i.

normal distribution $\mathcal{N}(0,1)$. Figures 1b, 1c and 1d show in a continuous black line the PDFs derived from the data, and in a grey dash line their fitted Gaussian, in dimensions f_1, f_2 and f_3 respectively. This fitted Gaussian was derived using the Curve Fitting toolbox of Matlab™. As it can be observed, the matching of a Gaussian with an irrelevant feature is almost perfect, while those features that contain more information, like f_1 and f_2, deviate much more from a normal distribution.

Making use of these differences, an error, denoted ϕ_l, for each feature l, where $l = 1,\ldots,d$, is calculated as:

$$\phi_l = \frac{1}{H}\sum_{i=1}^{H}(p_i - g_i)^2, \tag{4}$$

where H is the number of bins in which the range of the data is divided to estimate the PDF (p), and g is the fitted Gaussian. Equation (4) corresponds to the mean-squared error (MSE) between the PDF of the data over feature l and its fitted Gaussian. From these ϕ_l, the final weights b_l are calculated as:

$$b_l = \frac{\phi_l}{\sum_{l=1}^{d}\phi_l} \tag{5}$$

Therefore, combining (3) and (5), the proposed approximation, denoted $\hat{\sigma}^2$, is derived as:

$$\hat{\sigma}^2 = \sum_{l=1}^{d} b_l \delta_l, \tag{6}$$

which gathers the distances present in the most relevant features, giving less importance to the dimensions that do not contribute to describe the structure of the data. The complete algorithm to calculate $\hat{\sigma}^2$ is described in Algorithm 1.

Algorithm 1 Kernel parameter approximation for high-dimensional data.

Input: Data $\mathbf{X} \in \mathbb{R}^{N\times d}$.
Output: Sigma parameter $\hat{\sigma}^2$
1: Calculate the Manhattan distances between the datapoints using Equation (3): vector of distances per feature δ_l.
2: Obtain the weights for each of the features using Equations (4) and (5): weights b_l.
3: Calculate $\hat{\sigma}^2$ using Equation (6).

C. Utility metric for feature subset selection

In Section III-A, a new representation $\mathbf{E}$ of the data based on the eigenvectors was built, which described the main structures present in the original high-dimensional data. The goal is to select a subset of the features which best approximates the data in this new representation. In the literature, this feature selection problem is formulated using a graph-based loss function and a sparse regularizer of the coefficients is used to select a subset of features, as explained in [14]. The main idea of these approaches is to regress the data to its low dimensional embedding along with some sparse regularization. The use of such regularization techniques reduces overfitting and achieves dimensionality reduction. This regression is generally formulated as a least squares (LS) problem, and in many of these cases, the metric that is used for feature selection is the magnitude of their corresponding weights in the least squares solution [9], [23]. However, the optimized weights do not necessarily reflect the importance of the corresponding feature as it is scaling dependent and it does not properly take interactions across features into account [17]. Instead, the importance of a feature can be quantified using the increase in least-squared error (LSE) if that feature was to be removed and the weights were re-optimized. This increase in LSE, called the 'utility' of the feature can be efficiently computed [17] and can be used as an informative metric for a greedy backwards feature selection procedure [17]–[19], as an alternative for (group-)LASSO based techniques. Under some technical conditions, a greedy selection based on this utility metric can even be shown to lead to the optimal subset [32].

After representing the dataset using the matrix $\mathbf{E} \in \mathbb{R}^{N\times c}$ containing the c eigenvectors, the following LS optimization problem finds the weights $\mathbf{p}$ that best approximate the data $\mathbf{X}$ in the c-dimensional representation in $\mathbf{E}$:

$$J = \min_{\mathbf{P}} \frac{1}{N}||\mathbf{X}\mathbf{p} - \mathbf{E}||_F^2 \tag{7}$$

where J is the cost or the LSE and $||.||_F$ denotes the Frobenius norm.

If $\mathbf{X}$ is a full rank matrix and if $N > d$, the LS solution $\hat{\mathbf{p}}$ of (7) is

$$\hat{\mathbf{p}} = \mathbf{R}_{\mathbf{XX}}^{-1}\mathbf{R}_{\mathbf{XE}}, \tag{8}$$

with $\mathbf{R}_{\mathbf{XX}} = \frac{1}{N}\mathbf{X}^T\mathbf{X}$ and $\mathbf{R}_{\mathbf{XE}} = \frac{1}{N}\mathbf{X}^T\mathbf{E}$.

The goal of this feature selection method is to select the subset of $s(<d)$ features that best represents $\mathbf{E}$. This feature selection problem can be reduced to the selection of the best $s(<d)$ columns of $\mathbf{X}$ which minimize (7). However, this is

inherently a combinatorial problem and is computationally unfeasible to solve. Nevertheless, several greedy and approximative methods have been proposed [13], [18], [23]. In the current work, the use of the utility metric for subset selection is proposed to select these best s columns.

The utility of a feature l of $\mathbf{X}$, in an LS problem like (7), is defined as the increase in the LSE J when the column corresponding to the l-th feature in $\mathbf{X}$ is removed from the problem and the new optimal weight matrix, $\hat{\mathbf{p}}_{-l}$, is re-computed similar to (8). Consider the new LSE after the removal of feature l and the re-computation of the weight matrix $\hat{\mathbf{p}}_{-l}$ to be J_{-l}, defined as:

$$J_{-l} = \frac{1}{N}||\mathbf{X}_{-l}\hat{\mathbf{p}}_{-l} - \mathbf{E}||_F^2 \tag{9}$$

where $\mathbf{X}_{-l}$ denotes the matrix $\mathbf{X}$ with the column corresponding to l-th feature removed. Then according to the definition, the utility of feature l, U_l is:

$$U_l = J_{-l} - J \tag{10}$$

A straightforward computation of U_l would be computationally heavy due to the fact that the computation of $\hat{\mathbf{p}}_l$ requires a matrix inversion of $\mathbf{X}_{-l}\mathbf{X}_{-l}^T$, which has to be repeated for each feature l.

However, it can be shown that the utility of the l-th feature of $\mathbf{X}$ in (10) can be computed efficiently without the explicit recomputation of $\hat{\mathbf{p}}_{-l}$ by using the following expression [17]:

$$U_l = \frac{1}{q_l}||\bar{\mathbf{p}}_l||_2, \tag{11}$$

where q_l is the l-th diagonal element of $\mathbf{R}_{\mathbf{XX}}^{-1}$ and p_l is the l-th row in $\hat{\mathbf{p}}$, corresponding to the l-th feature. The mathematical proof of (11) can be found in [17]. Note that $\mathbf{R}_{\mathbf{XX}}^{-1}$ is already known from the computation of $\hat{\mathbf{p}}$ such that no additional matrix inversion is required.

However, since the data matrix $\mathbf{X}$ can contain redundant features or features that are linear combinations of each other in its columns, it cannot be guaranteed that the matrix $\mathbf{X}$ in (7) is full-rank. In this case, the removal of a redundant column from $\mathbf{X}$ will not lead to an increase in the LS cost of (7). Moreover, $\mathbf{R}_{\mathbf{XX}}^{-1}$, used to find the solution of (7) in (8), will not exist in this case since the matrix $\mathbf{X}$ is rank deficient. A similar problem appears if $N < d$, which can happen in case of very high-dimensional data. To overcome this problem, the definition of utility generalized to a minimum l_2-norm selection [17] is used in this work. This approach eliminates the feature yielding the smallest increase in the l_2-norm of the weight matrix when the column corresponding to that feature were to be removed and the weight matrix would be re-optimized. Moreover, minimizing the l_2-norm of the weights further reduces the risk of overfitting.

This generalization is achieved by first adding an l_2-norm penalty β to the cost function that is minimized in (7):

$$J = \min_{\mathbf{p}} \frac{1}{2}||\mathbf{X}\mathbf{p} - \mathbf{E}||_F^2 + \beta||\mathbf{p}||_2^2 \tag{12}$$

where $0 < \beta \leqslant \mu$ with μ equal to the smallest non-zero eigenvalue of $\mathbf{R}_{\mathbf{XX}}$ in order to ensure that the bias added due to the penalty term in (12) is negligible. The minimizer of (12) is:

$$\hat{\mathbf{p}} = \mathbf{R}_{\mathbf{XX}\beta}^{-1}\mathbf{R}_{\mathbf{XE}} = (\mathbf{R}_{\mathbf{XX}} + \beta\mathbf{I})^{-1}\mathbf{R}_{\mathbf{XE}} \tag{13}$$

It is noted that (13) reduces to $\mathbf{R}_{\mathbf{XX}}^{\dagger}\mathbf{R}_{\mathbf{XE}}$ when $\beta \to 0$, where $\mathbf{R}_{\mathbf{XX}}^{\dagger}$ denotes the Moore-Penrose pseudo-inverse. This solution corresponds to the minimum norm solution of (7) when $\mathbf{X}$ contains linearly dependent columns or rows. The utility U_l of the l-th column in $\mathbf{X}$ based on (12) is [17]:

$$\begin{aligned} U_l &= \left(||\mathbf{X}_{-l}\hat{\mathbf{p}}_{-l} - \mathbf{E}||_2^2 - ||\mathbf{X}\hat{\mathbf{p}} - \mathbf{E}||_2^2\right) \\ &\quad + \beta\left(||\hat{\mathbf{p}}_{-l}||_2^2 - ||\hat{\mathbf{p}}||_2^2\right) \\ &= (J_{-l} - J) + \beta\left(||\hat{\mathbf{p}}_{-l}||_2^2 - ||\hat{\mathbf{p}}||_2^2\right) \end{aligned} \tag{14}$$

Note that if column l in $\mathbf{X}$ is linearly independent from the other columns, (14) closely approximates to the original utility definition in (10) as the first term dominates over the second. However, if column l is linearly dependent, the first term vanishes and the second term will dominate. In this case, the utility quantifies the increase in l_2-norm after removing the l-th feature.

To select the best s features of $\mathbf{X}$, a greedy selection based on the iterative elimination of the features with the least utility is carried out. After the elimination of each feature, a re-estimation of the weights $\hat{\mathbf{p}}$ is carried out and the process of elimination is repeated, until s features remain.

Note that the value of β depends on the smallest non-zero eigenvalue of $\mathbf{R}_{\mathbf{XX}}$. Since $\mathbf{R}_{\mathbf{XX}}$ has to be recomputed every time when a feature is removed, also its eigenvalues change along the way. In practice, the value of β is selected only once and fixed for the remainder of the algorithm, as smaller than the smallest non-zero eigenvalue of $\mathbf{R}_{\mathbf{XX}}$ before any of the features are eliminated [18]. This value of β will be smaller than all the non-zero eigenvalues of any principal submatrix of $\mathbf{R}_{\mathbf{XX}}$ using the Cauchy's interlace theorem [33].

The summary of the utility subset selection is described in Algorithm 2. Algorithm 3 outlines the complete U2FS algorithm proposed in this paper.

Algorithm 2 Utility metric algorithm for subset selection.

Input: Data $\mathbf{X}$, Eigenvectors $\mathbf{E}$, Number of features s to select
Output: s features selected

1: Calculate $\mathbf{R}_{\mathbf{XX}}$ and $\mathbf{R}_{\mathbf{XE}}$ as described in Equation (8).
2: Calculate β as the smallest non-zero eigenvalue of $\mathbf{R}_{\mathbf{XX}}$
3: **while** Number of features remaining is $> s$ **do**
4: Compute $\mathbf{R}_{\mathbf{XX}\beta}^{-1}$ and $\hat{\mathbf{p}}$ as described in (13).
5: Calculate the utility of the remaining features using (11)
6: Remove the feature f_l with the lowest utility.
7: Update $\mathbf{R}_{\mathbf{XX}}$ and $\mathbf{R}_{\mathbf{XE}}$ by removing the rows and columns related to that feature f_l.
8: **end while**

Algorithm 3 Unsupervised feature selector based on the utility metric (U2FS).

Input: Data $\mathbf{X}$, Number of clusters c, Number of features s to select

Output: s features selected

1: Construct the similarity graph $\mathbf{W}$ as described in Section III-A selecting one of the weightings:
   - Binary
   - RBF kernel, using σ_0^2
   - RBF kernel, using $\hat{\sigma}^2$ based on Algorithm 1
2: Calculate the normalized Laplacian $\mathbf{L}$ and the eigenvectors α derived from Equation (2).
   Keep the c eigenvectors corresponding to the highest eigenvalues, excluding the first one.
3: Apply the backward greedy utility algorithm 2.
4: Return the s features remaining from the backward greedy utility approach.

As it has been stated before, one of the most remarkable aspects of the U2FS algorithm is the use of a greedy technique to solve the subset selection problem. The use of this type of method reduces the computational cost of the algorithm. This can be confirmed analyzing the computational complexity of U2FS, where the most demanding steps are the eigendecomposition of the Laplacian matrix (step 2 of Algortihm 3), which has a cost of $O(N^3)$ [34], and the subset selection stage in step 3 of Algorithm 3. Contrary to the state-of-the-art, the complexity of U2FS being a greedy method depends on the number of features to select. The most computationally expensive step of the subset selection in U2FS is the calculation of the matrix $\mathbf{R}_{\mathbf{XX}}^{-1}$, which has a computational cost of $O(d^3)$. In addition, this matrix needs to be updated $d-s$ times. This update can be done efficiently using a recursive updating equation from [17] with a cost of $O(t^2)$, with t the number of features remaining in the dataset, i.e. $t = d-s$. Since $t < d$, the cost for performing $d-s$ iterations will be $O((d-s)d^2)$, which depends on the number of features s to be selected. Note that the cost of computing the least squares solution $\hat{\mathbf{p}}_{-l}$ for each l in (14) is eliminated using the efficient equation (11), bringing down the cost for computing the utility from $O(t^4)$ to $O(t)$ in each iteration. This vanishes with respect to the $O(d^3)$ term (remember that $t < d$). Therefore, the total asymptotic complexity of U2FS is $O(N^3 + d^3)$.

IV. Experimental results

Two sets of experiments were performed, namely a simulation study and a set of tests run on well-known benchmark databases. Both novelties presented in this paper are tested: the new estimation of the kernel parameter $\hat{\sigma}^2$ for the RBF embedding, and the utility metric applied to feature selection.

In order to compare the subset selection techniques, in the following experiments, MCFS is compared to the proposed approach U2FS. The MCFS formulation relies on LASSO for the sparse regression problem, which is solved using the least angle regression algorithm (LARS) [9], [36]. The code used to evaluate this results was published by the authors of MCFS[37]. For the manifold learning stage, two different weighting systems to build the similarity matrix, suggested in MCFS, are tested: the binary weighting and the RBF kernel, which in [9] is called "heat kernel". To verify the suggested estimation of $\hat{\sigma}^2$, the utility metric for subset selection is combined with an RBF kernel with the suggested $\hat{\sigma}^2$ and with the original approximation proposed in [9], denoted σ_0^2. In Table I, the 5 approaches considered are summarized.

TABLE I: Methods compared in the experiments

	Similarity measure	Subset selection
MCFS: Bin	KNN + binary weighting	LASSO
MCFS: RBF + σ_0^2	RBF kernel, σ_0^2	LASSO
U2FS: Bin	KNN + binary weighting	Utility metric
U2FS: RBF + σ_0^2	RBF kernel, σ_0^2	Utility metric
U2FS: RBF + $\hat{\sigma}^2$	RBF kernel, $\hat{\sigma}^2$	Utility metric

A. Simulations

A set of nonlinear toy examples typically used in clustering problems are proposed to test the different feature selection methods. In these experiments, the goal was to verify the correct selection of the original set of features. Figure 2 shows the toy examples considered, which are described by features f_1 and f_2 , and the final description of the datasets can be seen in Table II.

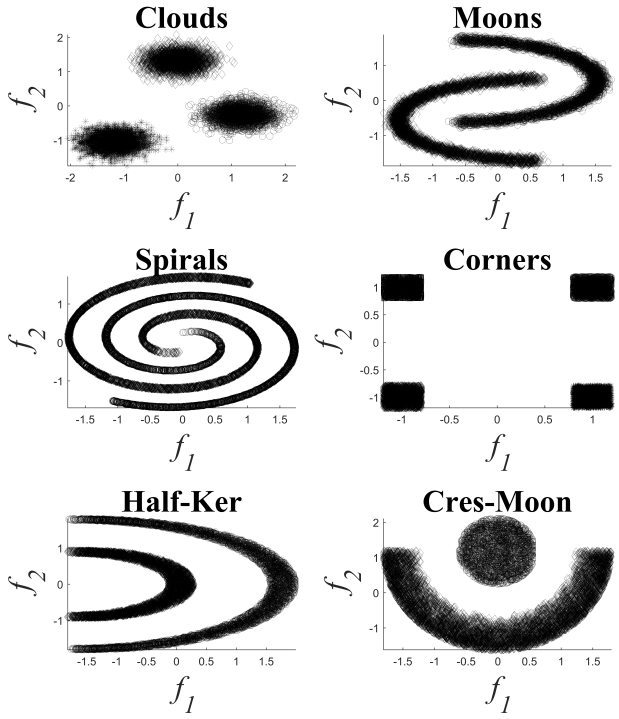

Fig. 2: Toy examples used for simulations.

All these problems are balanced, except for the last dataset Cres-Moon, for which the data is divided 25% to 75% between the two clusters. Five extra features in addition to the original f_1 and f_2 were added to each of the datasets in order to include redundant or irrelevant information:

TABLE II: Description of the toy example datasets.

	# samples	# classes
Clouds	9000	3
Moons	10000	2
Spirals	10000	2
Corners	10000	4
Half-Ker	10000	2
Cres-Moon	10000	2

- f_1' and f_2': random values extracted from two Pearson distributions characterized by the same higher-order statistics as f_1 and f_2 respectively.
- f_3' and f_4': Original f_1 and f_2 contaminated with Gaussian noise ($\nu\mathcal{N}(0,1)$), with $\nu = 1.5$.
- f_5': Constant feature of value 0.

The first step in the preprocessing of the features was to standardize the data using z-score to reduce the impact of differences in scaling and noise. In order to confirm the robustness of the feature selection techniques, the methods were applied using 10-fold cross-validation on the standardized data. For each fold a training set was selected using m-medoids, setting m to 2000 and using the centers of the clusters found as training samples. By doing so, the generalization ability of the methods can be guaranteed [27]. On each of the 10 training sets, the features were selected applying the 5 methods mentioned in Table I. For each of the methods, the number of clusters c was introduced as the number of classes presented in Table II. Since these experiments aim to evaluate the correct selection of the features, and the original features f_1 and f_2 are known, the number of features s to be selected was set to 2.

Regarding the parameter settings within the embedding methods, the binary was obtained setting k in the kNN approach to 5. For the RBF kernel embedding, σ_0^2 was set to the mean of the standard deviation along each dimension, as done in [9]. When using $\hat{\sigma}^2$, its value was obtained by applying the method described in Algorithm 1.

In terms of subset selection approaches, MCFS automatically sets the value of the regularization parameter required for the LARS implementation used [37]. For U2FS, β was set to the smallest non-zero eigenvalue of the matrix $\mathbf{R_{XX}}$ as described in Algorithm 2.

The performance of the algorithm is evaluated comparing the original set of features f_1 and f_2 to those selected by the algorithm. In these experiments, the evaluation of the selection results is binary: either the feature set selected is correct or not, regardless of the additional features f_i', for $i = 1,2,...,5$, selected.

In Table III the most common results obtained in the 10 folds are shown. The utility-based approaches, U2FS, always obtained the same results for all 10 folds of the experiments. On the contrary, the MCFS method provided different results for different folds of the experiment. For these cases, Table III shows the most common feature pair for each experiment, occurring at least 3 times.

As shown in Table III, the methods that always obtain the adequate set of features are based on utility, both with the binary weighting and with the RBF kernel and the suggested $\hat{\sigma}^2$. Since these results were obtained for the 10 folds, they confirm both the robustness and the consistency of the U2FS algorithm.

TABLE III: Results feature selection for toy examples

Method	**U2FS**			**MCFS**	
Embedding	**Bin**	**RBF** $+\sigma_0^2$	**RBF** $+\hat{\sigma}^2$	**Bin**	**RBF** $+\sigma_0^2$
Clouds	$\boldsymbol{f_1, f_2}$	f'_1, f'_4	$\boldsymbol{f_1, f_2}$	f'_1, f'_2	f'_1, f'_2
Moons	$\boldsymbol{f_1, f_2}$	f'_3, f'_4	$\boldsymbol{f_1, f_2}$	f'_1, f'_3	f'_1, f'_3
Spirals	$\boldsymbol{f_1, f_2}$	$\boldsymbol{f_1, f_2}$	$\boldsymbol{f_1, f_2}$	f_2, f'_2	f_2, f'_2
Corners	$\boldsymbol{f_1, f_2}$	f'_1, f'_2	$\boldsymbol{f_1, f_2}$	f_2, f'_2	f_2, f'_2
Half-Ker	$\boldsymbol{f_1, f_2}$	f_2, f'_3	$\boldsymbol{f_1, f_2}$	f_1, f'_3	f_1, f'_3
Cres-Moon	$\boldsymbol{f_1, f_2}$	f_1, f'_4	$\boldsymbol{f_1, f_2}$	f_2, f'_1	f_2, f'_2

B. Benchmark datasets

Additionally, the proposed methods were evaluated using 6 well-known benchmark databases. The databases considered represent image (USPS, ORL, COIL20), audio (ISOLET) and text data (PCMAC, BASEHOCK)[1], proposing examples with more samples than features, and vice versa. The description of these databases is detailed in Table IV.

TABLE IV: Description of the benchmark databases

	Data Type	**Samples**	**Features**	**Classes**
USPS	Images	9298	256	10
Isolet	Audio	1560	617	26
ORL	Images	400	1024	40
COIL20	Images	1440	1024	20
PCMAC	Text	1943	3289	2
BASEHOCK	Text	1993	4862	2

In these datasets, the relevant features are unknown. Therefore, the common practice in the literature to evaluate feature selectors consists of applying the algorithms, taking from 10 to 80% of the original set of features, and evaluating the accuracy of a classifier when trained and evaluated with the selected feature set [14]. The classifier used for this aim in other papers is k-Nearest Neighbors (KNN), setting the number of neighbors to 5.

These accuracy results are computed using 10-fold cross-validation to confirm the generalization capabilities of the algorithm. By setting m to 90% of the number of samples available in each benchmark dataset, m-medoids is used to select the m centroids of the clusters and use them as training set. Feature selection and the training of the KNN classifier are performed in these 9 folds of the standardized data, and the accuracy of the KNN is evaluated in the remaining 10% for testing. Exclusively for USPS, given the size of the dataset, 2000 samples were used for training and the remaining data was used for testing. These 2000 samples were also selected using m-medoids. Since PCMAC and BASEHOCK consist of binary data, these datasets were not standardized.

The parameters required for the binary and RBF embeddings, as well as β for the utility algorithm, are set as detailed in section IV-A.

Figure 3 shows the median accuracy obtained for each of the 5 methods. The shadows along the lines correspond to

[1]All datasets downloaded from http://featureselection.asu.edu/datasets.php

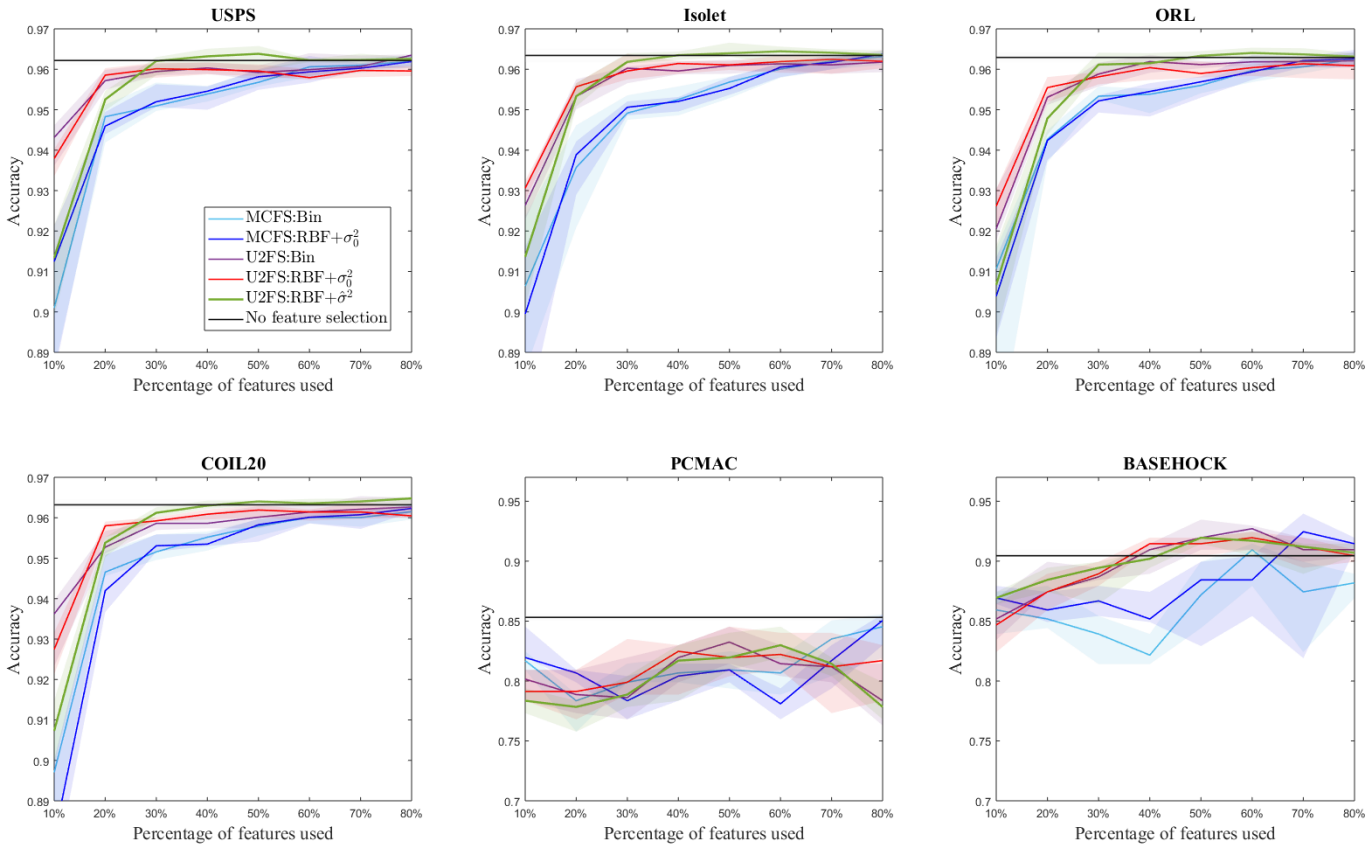

Fig. 3: Accuracy results for the benchmark databases, for selecting from 10 to 80% of the original number of features. The thick lines represent the median accuracy of the 10-fold cross-validation, and the shadows, the 25 and 75 percentile.

the 25 and 75 percentile of the 10 folds. As a reference, the accuracy of the classifier without using feature selection is shown in black for each of the datasets. Additionally, Figure 4 shows the computation time for both U2FS and MCFS applied on a binary weighting embedding. In this manner, the subset selection techniques, namely utility and LASSO respectively, can be evaluated regardless of the code efficiency of the embedding stage. Similarly to Figure 3, the computation time plots show in bold the median running time for each of the subset selection techniques, and the 25 and 75 percentiles around it obtained from the 10-fold cross-validation.

The difference in the trends of LASSO and utility in terms of computation time is due to their formulation. Feature selection based on LASSO, solved using the LARS algorithm in this case, requires the same computation time regardless of the number of features aimed to select. All features are evaluated together, and later on, an MCFS score obtained from the regression problem is assigned to them. The features with the higher scores are the ones selected. On the other hand, since the utility metric is applied in a backward greedy trend, the computation times change for different number of features selected. The lower the number of features selected compared to the original set, the higher the computation time. This is aligned with the computational complexity of the algorithm, described in Section III-C. In spite of this, it can be seen that even the highest computation time for utility is lower than the time taken using LASSO. The experiments were performed with 2x Intel Xeon E5-2640 @ 2.5 GHz processors and 64GB of working memory.

Additionally, in order to compare the results of U2FS with more methods from the state-of-the-art, the study proposed in [15] was replicated for two of the datasets mentioned above: COIL20 and USPS. For this study, 90 features were selected in a 10-fold cross-validation trend, with the training set selected at random. The evaluation of the features selected is done applying k-means, and measuring the clustering performance using clustering accuracy (ACA) and normalized mutual information (NMI). The results of this study are shown in Table V, taking as a reference U2FS with the RBF kernel and the proposed $\hat{\sigma}^2$. The results are in-line with other methods, which guarantees the applicability of U2FS to feature selection approaches.

V. Discussion

The results obtained in the experiments suggest that the proposed U2FS algorithm obtains comparable results to MCFS in all the applications suggested, taking less computational time. Nevertheless, the performance of the utility metric for feature selection varies for the different experiments presented and requires a detailed analysis.

From Table III, in Section IV-A, it can be concluded that the utility metric is able to select the correct features in an artificially contaminated dataset. Both the binary embedding and the RBF kernel with $\hat{\sigma}^2$ select the original set of features

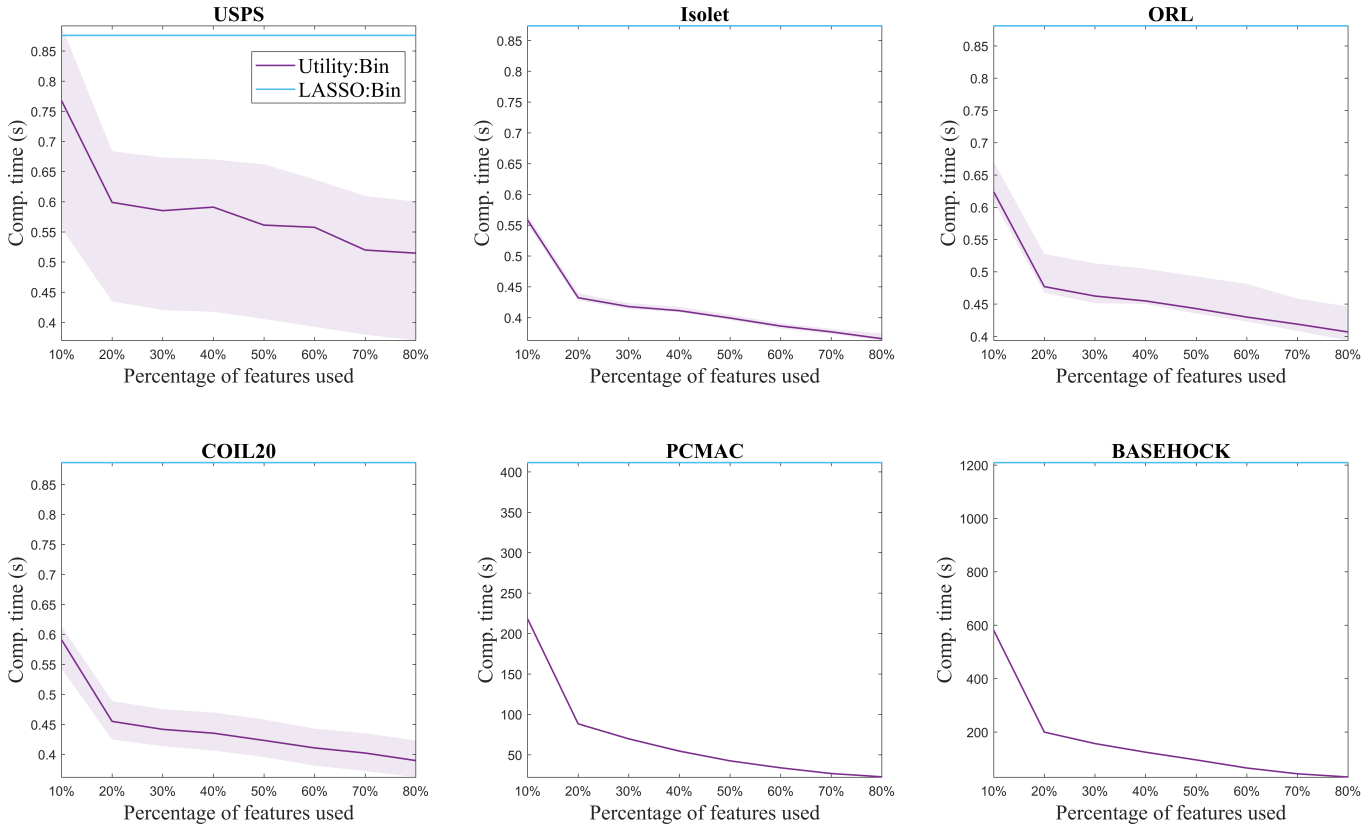

Fig. 4: Computation time for extracting from 10 to 80% of the original number of features for each of the benchmark databases.

TABLE V: Comparison of clustering accuracy (ACA) and Normalized mutual information (NMI) for 90 features selected.

	ACA		NMI	
	COIL20	**USPS**	**COIL20**	**USPS**
SPEC[8]	52.7±2.9	65.8±3	67.2±2.4	56.9±1.8
LapScore[7]	52.2±3.1	50.3±3.3	70.2±1.9	41.8±2.2
MCFS[9]	52.8±3.3	60.1±3.6	65.4±3.1	54.7±1.9
UDFS[10]	61.5±2.8	66.2±2.7	70.1±2.5	59.2±3.4
NDFS[11]	67.7±3.4	67.3±3	80.2±3.4	62.4±2.7
JELSR[12]	62±2.8	67.6±3.3	80±2.8	60.4±1.8
SOGFS[13]	66.2±2.9	69.4±2.9	81.3±3.4	64.2±2.4
SAMM-FS[15]	70±2.2	70±2.5	81.9±2	66.6±2.2
U2FS-RBF+$\hat{\sigma}^2$ (proposed)	54.4±3.6	70.1±3.7	69.3±1.7	61.6±1.5

for the 10 folds of the experiment. The stability in the results also applies for the RBF embedding with σ_0^2, which always selected the same feature pair for all 10 folds even though they are only correct for the spirals problem.

Therefore, considering the stability of the results, it can be concluded that U2FS is more robust in the selection of results, while MCFS is more unstable.

On the other hand, when considering the suitability of the features selected, two observations can be made. First of all, it can be seen that the lack of consistency in the MCFS approaches discards the selection of the correct set of features. Moreover, the wrong results obtained with both MCFS and utility methods for the RBF embedding using σ_0^2 reveal the drawback of applying this approximation of σ_0^2 in presence of redundant or irrelevant features. Since this value is calculated as the mean of the standard deviation of all the dimensions in the data, this measure can be strongly affected by irrelevant data, that could be very noisy and enlarge this sigma, leading to the allocation of all the samples to a mega-cluster.

While the use of the proposed approximation for $\hat{\sigma}^2$ achieves better results than σ_0^2, these are comparable to the ones obtained with the binary embedding when using the utility metric. The use of KNN to build graphs is a well-known practice, very robust for dense clusters, as it is the case in these examples. The definition of a specific field where each of the embeddings would be superior is beyond the scope of this paper. However, the excellence of both methods when combined with the proposed subset selection method only confirms the robustness of the utility metric, irrespective of the embedding considered.

For standardization purposes, the performance of the method was evaluated in benchmark databases. As it can be observed, in terms of the accuracy obtained for each experiment, U2FS achieves comparable results to MCFS for most of the datasets considered, despite its condition of greedy method.

In spite of this, some differences in performance can be observed in the different datasets. The different ranking of the methods, as well as the accuracy obtained for each of the databases can be explained taking into account the type of data under study and the ratio between samples and dimensions.

Taking into account the type of data represented by each test, it can be observed that for the ISOLET dataset, containing sound information, two groups of results are distinguishable. The group of U2FS results outperforms those derived from MCFS, which only reach comparable results for 60% of the features selected. These two groups of results are caused by the subset selection method applied, and not for the embedding, among which the differences are not remarkable.

In a similar way, for the case of the image datasets USPS, ORL and COIL20, the results derived from U2FS are slightly better than those coming from MCFS. In these datasets, similarly to the performance observed in ISOLET, accuracy increases with the number of features selected.

Regarding the differences between the proposed embeddings, it can be observed that the results obtained are comparable for all of them. Nonetheless, Figure 3 shows that there is a slight improvement in the aforementioned datasets for the RBF kernel with $\hat{\sigma}^2$, but the results are still comparable to those obtained with other embeddings. Moreover, this similarity in the binary and RBF results holds for the MCFS methods, for which the accuracy results almost overlap in Figure 3. This can be explained by the relation between the features considered. Since for these datasets the samples correspond to pixels, and the features to the color codes, a simple neighboring method such as the binary weighting is able to code the connectivity of pixels of similar colors.

The text datasets, PCMAC and BASEHOCK, are the ones that show bigger differences between the results obtained with U2FS and those obtained with MCFS. This can be explained by the amount of zeros present in the data, with which the utility metric is able to copes slightly better. The sparsity of the data leads to more error in the LASSO calculation, since more features end up having the same MCFS score, and among those, the order for selection comes at random. The results obtained with U2FS are more stable, in particular for the BASEHOCK dataset. For this dataset, U2FS even outperforms the results without feature selection if at least 40% of the features are kept.

In all the datasets proposed, the results obtained with MCFS show greater variability, i.e. larger percentiles. This is aligned with the results obtained in the simulations. The results for MCFS are not necessarily reproducible in different runs, since the algorithm is too sensitive to the training set selected. The variability of the U2FS methods is greater for the approaches based on the RBF kernel. This is due to the selection of the σ^2 parameter, which also depends on the training set. The tuning of this parameter is still very sensitive to high-dimensional and large-scale data, posing a continuous challenge for the machine learning community [38], [39].

Despite it being a greedy method, the utility metric proves to be applicable to feature selection approaches and to strongly outperform LASSO in terms of computational time, without significant reduction in accuracy. U2FS proves to be effective both in cases with more samples than features and vice versa. The reduction in computation time is clear, for all the benchmark databases described, and is particularly attractive for high-dimensional datasets. Altogether, our feature selection approach U2FS, based on the utility metric, and with the binary or the RBF kernel with $\hat{\sigma}^2$ is recommended due to its fast performance and its interpretability.

Additionally, the results of U2FS are shown to perform comparable with the state-of-the-art, as shown in Table V. The differences in performance come with the exhaustive search of combinations in the other methods, while U2FS is the only greedy approach proposed. Moreover, JELSR, SOFGS and SAMM-FS do not apply manifold learning and sparse regression sequentially, but simultaneously adapt both steps in a more complex methodology. Nevertheless, it can be observed that the results are not that different, in particular for the USPS dataset.

While this work is focused on the applicability of the utility metric for future selection in a setting comparable to MCFS, these other feature selection methods also make use of sparse regression for feature selection, with different embedding techniques and objective functions. Future work may study the limitations and impact of applying the utility metric to these other formulations. One of the most direct applications could be the substitution of group-LASSO for group-utility, in order to perform selections of groups of features [17]. This can be of interest in cases where the relations between features are known, such as in channel selection [18] or in multi-modal applications [40].

VI. Conclusion

This work presents a new method for unsupervised feature selection based on manifold learning and sparse regression. The main contribution of this paper is the formulation of the utility metric in the field of spectral feature selection, substituting other sparse regression methods that require more computational resources. This method, being a backward greedy approach, has been proven to obtain comparable results to the state-of-the-art methods with analogous embedding approaches, yet at considerably reduced computational load. The method shows consistently good results in different applications, from images to text and sound data; and it is broadly applicable to problems of any size: using more features than samples or vice versa.

Furthermore, aiming to show the applicability of our method to data presenting non-linearities, the proposed approach has been evaluated in simulated data, considering both a binary and an RBF kernel embedding. Given the sensitivity of the RBF kernel to high-dimensional spaces, a new approximation of the RBF kernel parameter was proposed, which does not require further tuning around the value obtained. The proposed approximation outperforms the rule-of-thumb widely used in the literature in most of the scenarios presented. Nevertheless, in terms of feature selection, the utility metric shows to perform robustly regardless of the embedding considered. This opens a new path for structure sparsity-inducing feature selection methods, which can benefit from this quick and efficient technique.

REFERENCES

[1] Michel Verleysen and Damien François. "The curse of dimensionality in data mining and time series prediction". In: *International Work-Conference on Artificial Neural Networks*. Springer. 2005, pp. 758–770.

[2] John Maindonald et al. "Pattern recognition and machine learning". In: *Journal of Statistical Software* 17.b05 (2007).

[3] Svante Wold, Kim Esbensen, and Paul Geladi. "Principal component analysis". In: *Chemometrics and intelligent laboratory systems* 2.1-3 (1987), pp. 37–52.

[4] Xing Jiang, Liqing Zhang, Qibin Zhao, et al. "ECG arrhythmias recognition system based on independent component analysis feature extraction". In: *TENCON 2006-2006 IEEE Region 10 Conference*. IEEE. 2006, pp. 1–4.

[5] Isabelle Guyon and André Elisseeff. "An introduction to variable and feature selection". In: *Journal of machine learning research* 3.Mar (2003), pp. 1157–1182.

[6] Dalton Lunga, Saurabh Prasad, Melba M Crawford, et al. "Manifold-learning-based feature extraction for classification of hyperspectral data: A review of advances in manifold learning". In: *IEEE Signal Processing Magazine* 31.1 (2013), pp. 55–66.

[7] Xiaofei He, Deng Cai, and Partha Niyogi. "Laplacian score for feature selection". In: *Advances in neural information processing systems*. 2006, pp. 507–514.

[8] Zheng Zhao and Huan Liu. "Spectral feature selection for supervised and unsupervised learning". In: *Proceedings of the 24th international conference on Machine learning*. 2007, pp. 1151–1157.

[9] Deng Cai, Chiyuan Zhang, and Xiaofei He. "Unsupervised feature selection for multi-cluster data". In: *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2010, pp. 333–342.

[10] Yi Yang, Heng Tao Shen, Zhigang Ma, et al. "L2, 1-norm regularized discriminative feature selection for unsupervised". In: *Twenty-Second International Joint Conference on Artificial Intelligence*. 2011.

[11] Zechao Li, Yi Yang, Jing Liu, et al. "Unsupervised feature selection using nonnegative spectral analysis". In: *Twenty-Sixth AAAI Conference on Artificial Intelligence*. 2012.

[12] Chenping Hou, Feiping Nie, Xuelong Li, et al. "Joint embedding learning and sparse regression: A framework for unsupervised feature selection". In: *IEEE Transactions on Cybernetics* 44.6 (2013), pp. 793–804.

[13] Feiping Nie, Wei Zhu, and Xuelong Li. "Structured Graph Optimization for Unsupervised Feature Selection". In: *IEEE Transactions on Knowledge and Data Engineering* (2019).

[14] Xiaofeng Zhu, Xuelong Li, Shichao Zhang, et al. "Robust joint graph sparse coding for unsupervised spectral feature selection". In: *IEEE transactions on neural networks and learning systems* 28.6 (2016), pp. 1263–1275.

[15] Rui Zhang, Feiping Nie, Yunhai Wang, et al. "Unsupervised feature selection via adaptive multimeasure fusion". In: *IEEE transactions on neural networks and learning systems* 30.9 (2019), pp. 2886–2892.

[16] Charu C Aggarwal, Alexander Hinneburg, and Daniel A Keim. "On the surprising behavior of distance metrics in high dimensional space". In: *International conference on database theory*. Springer. 2001, pp. 420–434.

[17] Alexander Bertrand. "Utility Metrics for Assessment and Subset Selection of Input Variables for Linear Estimation [Tips & Tricks]". In: *IEEE Signal Processing Magazine* 35.6 (2018), pp. 93–99.

[18] A. M. Narayanan and A. Bertrand. "Analysis of Miniaturization Effects and Channel Selection Strategies for EEG Sensor Networks With Application to Auditory Attention Detection". In: *IEEE Transactions on Biomedical Engineering* 67.1 (2020), pp. 234–244.

[19] Joseph Szurley, Alexander Bertrand, Peter Ruckebusch, et al. "Greedy distributed node selection for node-specific signal estimation in wireless sensor networks". In: *Signal Processing* 94 (2014), pp. 57–73.

[20] Joseph Szurley, Alexander Bertrand, and Marc Moonen. "Efficient computation of microphone utility in a wireless acoustic sensor network with multi-channel Wiener filter based noise reduction". In: *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2012, pp. 2657–2660.

[21] Saúl Solorio-Fernández, J Ariel Carrasco-Ochoa, and José Fco Martínez-Trinidad. "A review of unsupervised feature selection methods". In: *Artificial Intelligence Review* 53.2 (2020), pp. 907–948.

[22] Shuicheng Yan, Dong Xu, Benyu Zhang, et al. "Graph embedding and extensions: A general framework for dimensionality reduction". In: *IEEE transactions on pattern analysis and machine intelligence* 29.1 (2006), pp. 40–51.

[23] Jie Gui, Zhenan Sun, Shuiwang Ji, et al. "Feature selection based on structured sparsity: A comprehensive study". In: *IEEE transactions on neural networks and learning systems* 28.7 (2016), pp. 1490–1507.

[24] Ulrike Von Luxburg. "A tutorial on spectral clustering". In: *Statistics and computing* 17.4 (2007), pp. 395–416.

[25] Norman Biggs, Norman Linstead Biggs, and Biggs Norman. *Algebraic graph theory*. Vol. 67. Cambridge university press, 1993.

[26] Mikhail Belkin and Partha Niyogi. "Laplacian eigenmaps and spectral techniques for embedding and clustering". In: *Advances in neural information processing systems*. 2002, pp. 585–591.

[27] Carolina Varon, Carlos Alzate, and Johan AK Suykens. "Noise level estimation for model selection in kernel PCA denoising". In: *IEEE transactions on neural networks and learning systems* 26.11 (2015), pp. 2650–2663.

[28] Andrew Y Ng, Michael I Jordan, and Yair Weiss. "On spectral clustering: Analysis and an algorithm". In: *Advances in neural information processing systems*. 2002, pp. 849–856.

[29] Fan RK Chung and Fan Chung Graham. *Spectral graph theory*. 92. American Mathematical Soc., 1997.

[30] Carlos Alzate and Johan AK Suykens. "Multiway spectral clustering with out-of-sample extensions through weighted kernel PCA". In: *IEEE transactions on pattern analysis and machine intelligence* 32.2 (2008), pp. 335–347.

[31] Barbara E Reynolds. "Taxicab geometry". In: *Pi Mu Epsilon Journal* 7.2 (1980), pp. 77–88.

[32] Christophe Couvreur and Yoram Bresler. "On the optimality of the backward greedy algorithm for the subset selection problem". In: *SIAM Journal on Matrix Analysis and Applications* 21.3 (2000), pp. 797–808.

[33] Suk-Geun Hwang. "Cauchy's interlace theorem for eigenvalues of Hermitian matrices". In: *The American Mathematical Monthly* 111.2 (2004), pp. 157–159.

[34] Serafeim Tsironis, Mauro Sozio, Michalis Vazirgiannis, et al. "Accurate spectral clustering for community detection in mapreduce". In: *Advances in Neural Information Processing Systems (NIPS) Workshops*. Citeseer. 2013.

[35] Alexander Bertrand, Joseph Szurley, Peter Ruckebusch, et al. "Efficient calculation of sensor utility and sensor removal in wireless sensor networks for adaptive signal estimation and beamforming". In: *IEEE Transactions on Signal Processing* 60.11 (2012), pp. 5857–5869.

[36] Bradley Efron, Trevor Hastie, Iain Johnstone, et al. "Least angle regression". In: *The Annals of statistics* 32.2 (2004), pp. 407–499.

[37] Xiaofei He Deng Cai, Chiyuan Zhang. *Supervised/Unsupervised/Semi-supervised Feature Selection for Multi-Cluster/Class Data*. URL: http://www.cad.zju.edu.cn/home/dengcai/Data/MCFS.html (visited on 03/03/2020).

[38] Shen Yin and Jiapeng Yin. "Tuning kernel parameters for SVM based on expected square distance ratio". In: *Information Sciences* 370 (2016), pp. 92–102.

[39] Alaa Tharwat, Aboul Ella Hassanien, and Basem E Elnaghi. "A BA-based algorithm for parameter optimization of support vector machine". In: *Pattern Recognition Letters* 93 (2017), pp. 13–22.

[40] Lei Zhao, Qinghua Hu, and Wenwu Wang. "Heterogeneous feature selection with multi-modal deep neural networks and sparse group lasso". In: *IEEE Transactions on Multimedia* 17.11 (2015), pp. 1936–1948.