Feature and Instance Selection Through Discriminant Analysis Criteria

Fadi Dornaika (fdornaika@gmail.com)
Universidad del Pais Vasco Facultad de Ciencias de la Actividad Fisica y del Deporte
https://orcid.org/0000-0001-6581-9680

Abdelmalik Moujahid
University of the Basque Country: Universidad del Pais Vasco

Research Article

Keywords: Feature selection, instance selection, feature and instance selection, discriminant analysis, image classification

Posted Date: July 16th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-610854/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License
Feature and Instance Selection Through Discriminant Analysis Criteria

F. Dornaika$^{1,2}$ and A. Moujahid$^1$
$^1$ University of the Basque Country UPV/EHU, Spain
$^2$ IKERBASQUE, Basque Foundation for Science, Bilbao, Spain

Abstract

Feature selection and instance selection are two data preprocessing methods widely used in data mining and pattern recognition. The main goal is to reduce the computational cost of many learning tasks. Recently, joint feature and instance selection has been approached by solving some global optimization problems using meta-heuristics. This approach is not only computationally expensive, but also does not exploit the fact that the data usually has a structured manifold implicitly hidden in the data and its labels.

In this paper, we address joint feature and instance selection using scores derived from discriminant analysis theory. We present three approaches for joint feature and instance selection. The first scheme is a wrapper technique, while the other two schemes are filtering techniques. In the filtering approaches, the search process uses a genetic algorithm where the evaluation criterion is mainly given by the discriminant analysis score. This score depends simultaneously on the feature subset candidate and the best corresponding subset of instances. Thus, the best feature subset and the best instances are determined by finding the best score. The performance of the proposed approaches is quantified and studied using image classification with Nearest Neighbor and Support Vector Machine Classifiers. Experiments are conducted on five public image datasets. We compare the performance of our proposed methods with several state-of-the-art methods. The experiments performed show the superiority of the proposed methods over several baseline methods.

Keywords Feature selection, instance selection, feature and instance selection, discriminant analysis, image classification.

1 Introduction

Feature selection (FS) and instance selection (IS) have been an active research topic in preprocessing of datasets of images, videos, and texts as they promise fast performance and reduced complexity.
in classification problems. The importance of these tasks is increasingly recognized in most real-world datasets problems that are addressed in machine learning and autonomous systems. Feature Selection methods reduce the dimensionality of datasets by removing features that are considered as irrelevant or noisy for the learning task. This topic has received a lot of attention in machine learning and pattern recognition communities [4, 46, 68, 35, 37, 50, 1]. In any dataset, data can be seen as a collection of data points called instances. Each instance can describe a particular object or situation and is defined by a set of independent variables called features or attributes. Instance Selection refers to the task of selecting a useful subset of data called representatives that can summarize the whole dataset (images, videos or texts)[17, 34, 6, 20]. On the other hand, Feature Selection selects a subset of features that represents patterns from a larger set of features that are often mutually redundant, possibly irrelevant[59, 64, 18]. As a result the ability to select features from a large feature set can be critical for many computer vision tasks since the task will focus only on the important attributes of the dataset. Instance Selection and Feature Selection are used by many machine learning algorithms such as classification and clustering. In this regard, many methods for Instance Selection and Feature Selection have been proposed and have shown promising results [24, 32, 65, 56].

Accordingly, other approaches rely on the well-known statistical analysis technique Least Squares Regression (LSR) to perform various discriminative learning methods. In [63, 66], the authors propose a multiclass feature selection algorithm that embeds class label information into the LSR formulation so that the distances between classes can be increased. This was achieved by introducing a $\epsilon$-dragging technique that forces the regression targets of different classes to move in opposite directions. In [61], the authors introduce a new relaxed label regression method that aims to preserve the row sparsity consistency property of samples from the same class, so that the distances of regression responses between samples from the same class can be greatly reduced, imposing a novel inter-class sparsity regularisation term on the transformation. In contrast, other authors do not introduce an additional regularization term to capture the local structure of the data, but instead constrain the reconstruction error of the data by using low-rank preserving projection over a graph regularized reconstruction [60].

In this paper, we address the problem of features and instance selection. Generally speaking, it was shown that the best way for performing feature and instance selection is either to perform feature selection then instance selection, or to simultaneously perform both tasks. Both schemes
can have their own drawbacks and limitations. For instance, the first scheme (feature selection
then instance selection), the instance selection can be affected by the solution provided by the
subset of features found in a separate stage in advance. The simultaneous solution can be sub-
optimal and lacks of efficiency since very often the solution is obtained by a wrapper technique.
We propose a trade-off solution that can be seen as an in-between solution that is efficient and
accurate. The proposed method exploits the strengths of some feature subset selection and data
self-representativeness instance selection. A genetic algorithm based search scheme is deployed in
order to optimize a given criterion that characterizes the subset of features and the associated
selected instances. The proposed schemes are tested for the task of classification over different
datasets having small and high dimension. In addition, the proposed method will be compared
with the main trends used for combining feature and instance selection. Classification perfor-
mances will be quantified on several public image datasets in the two scenarios: original training
dataset, and (ii) pre-processed training dataset. Here, the pre-processed datasets means that the
training datasets undergoes feature and instance selection according to a given method.

The remainder of this paper is structured as follows. Section 2 reviews the concept of Feature
and Instance Selection. It briefly reviews the instance selection method (Sparse Modeling Repre-
sentative Selection (SMRS)). Section 3 introduces our proposed schemes for the joint feature and
instance selection. Section 4 presents our experiments and results. Section 5 concludes the paper.
Table 1 presents the main acronyms and notations used in the paper.

2 Related Work

This section briefly presents the feature and instance selection. It also reviews the instance selection
method based on Sparse Modeling Representation Selection.

2.1 Feature selection

Feature selection is one of the stage for preprocessing the data through the identification and
selection of a subset of $F$ features from the original data of $D$ features ($F < D$) without any
transformation[57]. In the domain of supervised learning, feature Selection attempts to maximize
the accuracy of the classifier, minimizing the related measurement costs by reducing irrelevant and
possibly redundant features [5, 40, 45, 26, 46, 68, 35, 37, 50, 1]. Feature selection reduces the
Table 1: Description of the main acronyms and notations used in this paper.

| Notation | Description |
|----------|-------------|
| FS       | Feature Selection |
| IS       | Instance Selection |
| FSIS     | Feature Selection + Instance Selection |
| FIS      | Simultaneous Feature and Instance Selection |
| LDA      | Linear Discriminant Analysis |
| LDE      | Local Discriminant Embedding |
| GA       | Genetic Algorithm |

| Notation | Description |
|----------|-------------|
| $N$      | Number of images or samples |
| $D$      | Dimension of samples |
| $C$      | Number of classes |
| $X$      | Data matrix $\in \mathbb{R}^{D \times N}$ |
| $x$      | Data sample or feature vector $\in \mathbb{R}^{D}$ |
| $F$      | Subset of selected features |
| $X_F$    | Data matrix $\in \mathbb{R}^{F \times N}$ after feature selection |
| $X_I$    | Data matrix $\in \mathbb{R}^{D \times M}$ after instance selection |
| $X_{FI}$ or $X_I$ | Data matrix $\in \mathbb{R}^{F \times M}$ after feature and instance selection |
| $W$      | Projection matrix $\in \mathbb{R}^{D \times F}$ or Selection matrix $\in \mathbb{B}^{D \times F}$ |
| $B$      | Coding matrix $\in \mathbb{R}^{N \times N}$ |
| $H$      | Binary chromosome $\in \mathbb{B}^{D}$ |

complexity and the associated computational cost and improves the probability that a solution will be comprehensible and realistic.

2.2 Instance selection

As data volumes continue to grow, many traditional methods and systems struggle to process such volumes of data to obtain actionable knowledge [62, 22]. Current classification algorithms suffer from memory requirements and excessive processing times that make them impractical when dealing with huge datasets [25]. These issues can be critical when using an inert learning algorithm such as the nearest neighbor rule, and can prevent results from being obtained. However, reducing the size of the dataset by selecting a representative subset has two main advantages. (1) It reduces the storage space required to store the data. (2) It speeds up the learning and classification algorithms.

In the last two decades, several instance selection methods have been proposed [3, 54, 16, 33, 48, 58].

2.3 Combining Feature and Instance Selection

Instead of treating the instance or feature selection problems separately, some research efforts have been devoted to the study of the dual problem IS and FS, which we will refer to as feature and instance selection (FIS). There is no disadvantage in addressing both problems simultaneously,
since features and instances can be selected independently: Classification accuracy is the only part of the problem that is determined by the selected data.

Many algorithms have been developed to tackle either instance or feature selection. However, few works have addressed the joint instance and feature selection problem [21, 51]. The simplest way to combine instance and feature selection is to perform one process at a time. When the first process is instance selection and the second is feature selection, the combination is called ISFS. The reverse case is referred to as FSIS.

In [30], the authors solve these two problems using a heuristic technique. They use two simultaneous Simulated Annealing algorithms where each problem is solved separately and they then compute the quality of both problems using the current solution of each process.

Much work has been done with genetic and evolutionary algorithms [2, 53]. In [53], the authors use a chromosome that is a binary vector whose elements are associated with all instances and all features. The search for optimal subsets of features and instances is transferred to finding the optimal chromosome. Very often genetic algorithms are invoked to solve these problems. In [44] the authors use the chromosome in a different way, they split it into two fields. The first field corresponds to the feature weights and the second field has binary values corresponding to the instances. In [31] the authors use boolean value coding to select features and instances. The fitness function used is set to the combination of the accuracy of the Nearest Neighbor classifier and a value that penalizes the cardinality of each set.

In [11], the authors use an intelligent multi-objective evolutionary design to solve the instance and feature selection problems. The goal is to search for a minimal subset of features and instances by maximizing the classification accuracy of the Nearest Neighbor classifier. In [47], the authors use a hybrid genetic algorithm with the dual goal of reducing instances and selecting features while achieving the highest classification score. In [27], the authors use a genetic algorithm to select a small number of instances along with only significant features by giving the change that excludes features a greater probability of excluding features from the solutions. In [49], the authors present a combination of instance and feature selection by using an adaptation to the Estimation of Distribution algorithm (EDA)named Genetic Algorithm. [41] In [12], the authors propose the Instance and Feature Selection method based on an evolutionary model to perform feature and instance selection in Nearest Neighbor classification. It is based on the use of three population types, where these types represent the FS problem, the IS problem, and the feature and instance
problem. Each chromosome in each population has a fitness function formed by the accuracy of
the NN classifier (obtained from the three NN classifiers) and the reduction ratio in features and
instances.

2.4 Review of Instance Selection based on SMRS

In this section, we briefly describe the efficient instance selection method described in [19] (namely,
the method Sparse Modeling Representative Selection (SMRS)). In what follows, large bold letters
denote matrices and small bold letters denote vectors. The problem formulation can be stated as
follows. Consider a set of data samples $T = \{x_1, \ldots, x_N\}$ in $\mathbb{R}^d$ arranged as columns of the data
matrix $X = [x_1, \ldots, x_N]$, where $d$ denotes the sample dimension. The goal is to select the most
representative samples within the sample set $T$. SMRS is a filtering method that uses the concept
of relevance ranking. In other words, the relevance score of each sample $x_i, i = 1, \ldots, N$ is first
estimated. Based on the sorted relevance scores, the most relevant prototypes are then selected
using a predefined threshold or fixed number of prototypes. The basic idea of [19] is to estimate
the coefficients for the self-representativeness of the data, from which the relevance score of each
sample can be derived. The basic assumption is that each data sample is equal or close to a linear
combination of some samples in the original data set. Mathematically, this assumption can be
written as follows:

$$x_i = X b_i, \quad i = 1, \ldots, N$$

where $b_i$ is a vector of coefficients. The above $N$ equations can be encapsulated in one single matrix
equation $X = X B$ where $B = [b_1, \ldots, b_N]$.

The unknown matrix can be estimated by minimizing the following criterion using some regu-
larization on the unknown coefficients:

$$\sum_{i=1}^{N} \|x_i - X b_i\|^2 = \|X - X B\|^2_F$$

where $\|A\|_F$ denotes the Frobenius norm of the matrix $A$, i.e. $\|A\|_F = (\sum_{i} \sum_{j} A_{ij}^2)^{1/2}$. The
regularization plays an important role in order to get useful coding coefficients. In particular, the
regularization becomes crucial whenever the system is undetermined (number of samples is greater
than the dimensionality of the sample). The SMRS method [19] estimates the matrix $B$ by adding
a regularization term given by the $L_{1,2}$ norm of $B$. Thus, the matrix $B$ is estimated by minimizing the following criterion:

$$
B = \arg \min_B \left( \frac{1}{2} \|X - XB\|^2 + \lambda \|B\|_{1,2} \right) \text{ s.t. } 1^T B = 1^T
$$

(1)

where $\lambda$ is a positive regularization parameter. The above criterion has two terms. The first is the least square error associated with the self-representativeness of data. The second is the $L_{1,2}$ norm of the matrix $B$, i.e., $\|B\|_{1,2} = \sum_j |b(j,1)|^2 + \ldots + |b(j,N)|^2$. Thus, the regularization term is the $L_{1,2}$ norm of $B$ (i.e., sum of $L_2$ norm of the rows of the matrix $B$). This norm imposes that the matrix of coefficients $B$ is block-sparse. It can be considered as an approximation to the $L_{0,2}$ norm of a matrix (the number of non-null rows). Thus, in essence, the optimization problem provided by (1) attempts to compute the unknown matrix that simultaneously provides the least reconstruction error and that has the least number of non-null rows (a low rank matrix). The affine constraint $1^T B = 1^T$ makes the selection of representatives invariant with respect to a global translation of the data. The optimization of (1) can be carried out using the Alternating Direction Method of Multipliers (ADMM) [8]. Thus, the relevance score of the $i^{th}$ sample is set to the $L_2$ norm of the $i^{th}$ row of $B$. It should be noted that in a supervised learning context, the same framework of SMRS can be applied to each class separately to find the most representative examples in each class.

3 Proposed Approaches

3.1 Motivation

As can be seen, the state-of-the-art methods for combining feature selection and instance selection can be mainly divided into two groups. In the first group, referred to as FSIS, the proposed algorithms perform feature selection followed by instance selection. In the second group, the two processes are performed simultaneously. These are referred to as FIS. A deep look into related work on FIS methods shows that the majority of these approaches, which attempt to solve these problems in a broad generic formulation, solve them by using specialized versions of genetic algorithms that divide the chromosome into two distinct regions, one for features and one for instances, and apply separate operators to each region. The main solution to FIS problems is given by evolutionary algorithms, which are able to achieve a solution where the fitness function depends mainly on the performance of a given classifier and on the reduction of features and instances. However,
the use of criteria derived from manifold learning paradigms has received much less attention. In this work, the joint selection of features and instances is solved by using some manifold learning criteria for the FS part and an efficient self-representation of the data for the IS part. Motivated by the fact that the combination FSIS can be generally more interesting than the combination IS + FS, when dealing with real-world problems [55], we propose three schemes for combining feature selection and instance selection. All three schemes use an evolutionary scheme for searching the FS solution part, but the IS part is solved and integrated within this evolutionary algorithm using a self-representation scheme for the data, thus avoiding the use of a specific chromosome for the instance selection part.

The three proposed schemes have some interesting properties in terms of effectiveness and efficiency. This is discussed in more detail in the next section. The first scheme is a wrapper technique. The second and third schemes are filtering techniques (i.e., their fitness function does not depend on a particular classifier). All the proposed schemes use a similar search strategy given by a genetic algorithm (GA). For the three proposed schemes, we use only one chromosome for feature selection and the final selection of both features and instances is guided by a fitness score that considers the candidate feature subset and the associated selected instances. We emphasize that there is no chromosome for the instances, since the selection of the instances is performed on the candidate feature subset using the efficient SMRS method.

The proposed methods are based on some collaboration between Feature Selection and Instance Selection. Moreover, the number of Features Selected is not fixed in the sense that the search process tries to find the best features from the data. In practice, the size of the selected features is generally between 40% and 60% of the original features.

### 3.2 Outline of the proposed schemes

In this section, we describe our three proposed methods for feature selection and instance selection. In all the proposed methods, a genetic algorithm is used to perform the search for the optimal feature chromosome. The first scheme is a wrapper technique, while the other two schemes do not depend on the classifier. Suppose that we have a training dataset represented by the data matrix \( \mathbf{X} \in \mathbb{R}^{D \times N} \), where \( \mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N] \), where each datum \( \mathbf{x}_i \), \( i = \{1, \ldots, N\} \), has \( D \) features.

Our goal is to use a genetic algorithm that maximizes a fitness function to select the features from each data, and then we select the instances using the method of SMRS. First, we build a binary
chromosome whose size is given by $D$. In this chromosome, a bit value of one in the chromosome representation means that the corresponding feature is included in the given subset, and the value of zero means that the corresponding feature is not included in the subset. Second, we create a fitness function that focuses on maximizing the performance for a given set of features and the corresponding selected instances. The relationship between features and instances is treated as something related to their quality for a supervised learning task. A textual description of the framework can be seen as follows: At the beginning, the complete sets of features and instances are set as initial solutions. There are two separate processes for selecting features and instances. Starting with the initial data, the new solution is generated using the feature selection meta-heuristic, where the features of the data are reduced, then the algorithm performs instance selection based on the feature subset, which results in a given solution for feature and instance selection. The goal is to find the optimal feature chromosome by maximizing the fitness function, which already quantifies the goodness of the feature subset and the instance subset.

In this work, we introduce three fitness functions that produce three different schemes. The first one (wrapper FSIS) based on classification accuracy, while the second and third fitness functions are based on LDA and LDE criteria, respectively. The classifiers used in our work are: Nearest Neighbor (NN) and Support Vector Machines (SVM).

### 3.3 Wrapper approach: W-FSIS

First, we build a binary chromosome $H$ for feature selection. The size of $H$ is $D$. Our goal is to use a genetic algorithm that optimize a fitness function that score both the subset of features associated to the current chromosome and the selected instances provided by the SMRS method. Second, we build a fitness function focused on maximizing the classification rate.

The flowchart of the proposed fitness function associated with the wrapper FSIS is illustrated in Figure 1. Since it is a wrapper technique, the fitness is mainly set to the rate of correct classification of the used classifier over some data with known labels. Due the specific problem that is given by a cooperative feature selection instance selection, the rate should actually score the goodness of the subset of features and the subset of the associated selected instances. Let $X_F \in \mathbb{R}^{F \times N}$ denote the data matrix $X$ after feature selection. That is to say $X_F$ is $X$ from which all features, that have their chromosome bit equal to zero, have been eliminated. Let $X_I \in \mathbb{R}^{F \times M}$ denote the output of instance selection given by the SMRS method when applied on $X_F$. Here $M$ denotes the number
Figure 1: Wrapper FSIS algorithm. This figure summarizes the design of the fitness function associated with the proposed wrapper FSIS algorithm. The fitness function of each feature selection chromosome is set to a blend of three classification errors obtained after carrying out the feature selection and the instance selection processes. In the proposed algorithm only the features have a chromosome that is optimized using a genetic algorithm.
of selected instances. In practice, \( M \) is user-defined. Let \( X_N \in \mathbb{R}^{F \times (N-M)} \) denote the non-selected instances in \( X_F \). We have \( X_F = X_I \cup X_N \).

Thus, based on the content of the current chromosome and the associated output of the SMRS method, we are able to generate from the training set the following data matrices: \( X_F \), \( X_I \), and \( X_N \). We then randomly split the training matrix \( X_F \) into two equal parts: \( X_{F1} \) and \( X_{F2} \). Based on the four parts \( X_{F1} \), \( X_{F2} \), \( X_I \), and \( X_N \), we can estimate three classification rates for a given classifier as it is shown in Figure 1. The first rate, denoted by \( R_{F1} \) is the classifier success rate when the part \( X_{F1} \) is used as a training set and the part \( X_{F2} \) is used as a test set. The second rate, denoted by \( r_{F2} \), is the classifier success rate when the part \( X_{F2} \) is used as a training set and the part \( X_{F1} \) is used as a test set. The third rate, denoted by \( R_I \), is the classifier success rate when the part \( X_I \) is used as a training set and the part \( X_N \) is used as a test set. By merging all these classification rates, the accuracy of the current chromosome, \( H \), can be given by the following formula:

\[
Acc(H) = \lambda R_I + (1 - \lambda) \left( \frac{R_{F1} + R_{F2}}{2} \right)
\]

where \( \lambda \) is a positive number between 0 and 1. The reduction rate of the chromosome \( H \) is:

\[
Reduction(H) = 1 - \frac{\#FeaturesSelected}{D}
\]

By merging the classification rate and the reduction rate, the fitness of the current chromosome, \( H \), can be evaluated by the following formula:

\[
Fitness(H) = \alpha . Acc(H) + (1 - \alpha) . Reduction(H)
\]

where \( \alpha \) is a positive number between zero and one. For tasks where accuracy can be the primary objective, this parameter can be set to a value very close to one.

As it can be seen, the above fitness function takes into account the quality of the selected features and that of the selected instances. Based on the fitness function, a genetic algorithm is invoked in order to estimate the optimal chromosome. The main steps of the proposed Wrapper FSIS method are given in Algorithm 1.

It is worth noting that for a given number of selected instances, the selected instances obtained by the SMRS method are linked to the content of the feature chromosome and thus can be
considered as deterministic samples.

| Data: Training dataset $X$ (a $D \times N$ matrix), number of selected instances per class $K$, and an evaluation function |
|---|
| Result: Set of $d_s$ selected features ($d_s < D$) and set of $K$ instances per class |
| - $t = 0$ |
| - Initialize $P(t) = \{H_1, H_2, \ldots, H_m\}$ |
| - Evaluate $P(t)$ |
| Repeat until (termination condition is satisfied) |
| - Select $P(t+1)$ from $P(t)$ |
| - Recombine $P(t+1)$ |
| - Evaluate the individuals in $P(t+1)$ using the fitness function (2) |
| - $t = t + 1$ |

**Algorithm 1:** Outline of the proposed Wrapper FSIS method.

### 3.4 Filter FSIS based on LDA criterion

#### 3.4.1 Background: Feature subset selection using LDA criterion

Appropriate classification of data points requires the data points belonging to different classes be ideally far apart from each other and those belonging to the same class should be as near as possible. The extents of such inter-class separability and intra-class nearness depends on the features of the data points.

Linear Discriminant Analysis (LDA) [29, 38] seeks a linear combination of features that characterizes or separates two or more classes of objects or events. Initially, LDA was used in order to estimate a linear projection of data. It turned out that the same criterion can be used in order to select a subset of features by replacing the projection matrix by a selection matrix [38]. Based on LDA criterion, the authors of [38] propose a trace ratio for feature subset selection. The LDA criterion (after projection or feature selection) favors a high within-class compactness and a large between-class separation. In the Fisher discriminant analysis, the corresponding $D \times D$ scatter matrices $S_w, S_b$ are defined as follows:

$$S_w = X(I - G(G^T G)^{-1}G^T)X^T = XLwX^T$$

$$S_b = X(G(G^T G)^{-1}G^T - \frac{1}{N}11^T)X^T = XLbX^T$$
where \(1 = [1, 1, \ldots, 1]^{T} \in \mathbb{R}^{N}\), \(G \in \mathbb{B}^{N \times C}\) is the label matrix, \(G_{ij} = 1\) if \(x_{i}\) is labeled as the \(j\)-th class and \(G_{ij} = 0\) otherwise. \(C\) is the number of classes.

By fixing the size of the feature subset to \(F\), the problem stated in [38] is to find the selection matrix \(W \in \mathbb{B}^{D \times F}\) (each column of \(W\) is formed by zeros except one element is one) that maximizes the following trace ratio:

\[
W = \arg \max_{W} \frac{tr(W^{T}X L_{b}X^{T}W)}{tr(W^{T}X L_{w}X^{T}W)}
\]  

(3)

where \(tr(.)\) denotes the trace of a matrix, \(X\) is the original data matrix, \(W \in \mathbb{R}^{D \times F}\) is the selection matrix, \(L_{b}\) is the between-class Laplacian matrix associated with the data \(X\), and \(L_{w}\) is the within-class Laplacian matrix. Given the number of selected features \(F\), the authors devise an iterative algorithm in order to obtain the optimal selected features by optimizing the above trace ratio.

In [36], the author showed that by normalizing the rows of the data matrix (i.e., the features over instances) using zero-mean unit variance normalization, the trace ratio optimization becomes simpler since the method is based on scoring individual features and on taking the subset of \(F\) features that have the highest scores. Furthermore, it was shown that the normalization process may lead to better performances.

### 3.4.2 Proposed LDA-FSIS

The wrapper technique described in the previous section must learn and test a classifier for each chromosome in order to compute the rates intervening in fitness. This process can be computationally intensive for large datasets. In this section, we propose a filtering approach that is able to provide a solution for feature and instance selection without the use of a classifier. Our filtering approach follows the main steps described in the previous section. However, the main difference is that the fitness function does not depend on a particular classifier. Therefore, the fitness computation is much more efficient since no classifier needs to be trained or tested. The flowchart of the proposed fitness function in conjunction with the filter FSIS method is shown in Figure 2.

Similarly to the wrapper method, for a given chromosome configuration, we can generate the matrix data \(X_{F} \in \mathbb{R}^{F \times N}\). Then, the SMRS method selects the most relevant instances in \(X_{F}\). This provides the data matrix \(X_{I} \in \mathbb{R}^{F \times M}\). It should be noted that \(F\) is imposed by the content of the current feature chromosome (it is equal to the number of bits having one) while \(M\) is imposed by the SMRS method or by the user.
Let $\hat{X}_I$ denote the row-wise normalized version of $X_I$. Each row in $X_I$ is normalized to have a zero mean and unit variance. From the current data matrix $X_I$, one can generate the $M \times M$ matrices $L_b$ and $L_w$. Therefore, the fitness function of the current feature chromosome is given by:

$$Fitness(H) = \frac{tr(\hat{X}_I L_b \hat{X}_I^T)}{tr(\hat{X}_I L_w \hat{X}_I^T)}$$  \hspace{1cm} (4)$$

A high value for the score given in Eq. (4) means that the subset of $F$ features and $M$ instances are good with respect to the class discrimination of the data matrix $X_I$. We emphasize the fact that the selected $M$ instances, that are represented by $X_I$, are already assumed to be the most relevant representatives in the set of instances $X_F$. Therefore, the fitness function defined by (4) is a good indicator about the goodness of the solution (features and instances) associated with the current chromosome.

A genetic algorithm is invoked in order to estimate the optimal solution where the fitness function is given by (4). The resulting scheme is coined LDA-FSIS.

### 3.5 FSIS based on LDE criterion

#### 3.5.1 Background: LDE criterion

Local Discriminant Embedding (LDE) [10] attempts to separate the submanifold of each class, and specifically derives the embedding for classification in a low-dimensional Euclidean space. LDE has a similar expression to LDA. However, the Laplacian matrices are computed differently. In fact, LDE is based on two graphs: The intra-class graph $G_w$ (intrinsic graph) and the inter-class graph $G_b$ (penalty). Let $l(x_i)$ be the class label of $x_i$. For each data sample $x_i$, two subsets, $N_w(x_i)$ and $N_b(x_i)$ are computed. $N_w(x_i)$ contains the neighbors that share the same label with $x_i$, while $N_b(x_i)$ contains the neighbors with different labels. A simple way to compute these two sets of neighbors associated with the local sample is to use two nearest neighbor graphs: one nearest neighbor graph for homogeneous samples (parameterized by $K_1$) and one nearest neighbor graph for heterogeneous samples (parameterized by $K_2$). Note that $K_1$ and $K_2$ can be different and are chosen with empirical values.

Each of the previously mentioned graphs, $G_w$ and $G_b$, is represented by its corresponding affinity (weight) matrix $W_w$ and $W_b$, respectively $W_b$. The entries of these symmetric matrices are defined by the following formulas:
Figure 2: Proposed manifold based fitness function for FSIS. This figure summarizes the design of the fitness function associated with the two proposed FSIS algorithms. The score of each feature selection chromosome quantifies the goodness of the selected features and instances in terms of class separability. In our work, we derive two fitness functions yielding two manifold based schemes: LDA-FSIS and LDE-FSIS. In the proposed algorithm only the features have a chromosome that is optimized using a genetic algorithm.
\[ \begin{align*}
S_{w,ij} &= \begin{cases} 
sim(x_i, x_j) & \text{if } x_j \in N_w(x_i) \text{ or } x_i \in N_w(x_j) \\
0, & \text{otherwise}
\end{cases} \\
S_{b,ij} &= \begin{cases} 
1 & \text{if } x_j \in N_b(x_i) \text{ or } x_i \in N_b(x_j) \\
0, & \text{otherwise}
\end{cases}
\end{align*} \]

where \( \sim(x_i, x_k) \) is a real value that encodes the similarity between \( x_i \) and \( x_k \). Without loss of generality, we assume that \( \sim(x_i, x_k) \) belongs to the interval \([0, 1]\). Simple choices for this function are the Kernel heat and the cosine.

The Laplacian matrices associated with the graph \( G_w \) and \( G_b \) are given by:

\[ \begin{align*}
L_w &= D_w - S_w \quad (7) \\
L_b &= D_b - S_b \quad (8)
\end{align*} \]

where \( D_w \) denotes the diagonal weight matrix, whose entries are column (or row, since \( S_w \) is symmetric) sums of \( S_w \).

The projection matrix \( W \) can be estimated by maximizing the following trace ratio:

\[ W = \arg \max_W \frac{\text{tr}(W^T X L_b X^T W)}{\text{tr}(W^T X L_w X^T W)} \quad (9) \]

### 3.5.2 Proposed LDE-FSIS

Similarly to LDA-FSIS, we can generate a fitness function based on LDE criterion. For a given chromosome configuration, we can generate the data matrix \( X_I \in \mathbb{R}^{F \times M} \). The fitness of a given feature chromosome becomes:

\[ \text{Fitnesss}(H) = \frac{\text{tr}(\hat{X}_I L_b \hat{X}_I^T)}{\text{tr}(\hat{X}_I L_w \hat{X}_I^T)} \quad (10) \]

where the Laplacian matrices \( L_w \) and \( L_b \) are given by (5), (6), (7), and (8).
4 Differences with existing work

4.1 Proposed W-FSIS scheme vs. simultaneous FIS

The majority of simultaneous FIS LDE is proposed to address some limitations of the global Linear Discriminant Analysis (LDA) method. It extends the concept of LDA to perform local discrimination. According to graph-based embedding paradigms, the schemes are based on the use of a chromosome, which is used to select both features and instances. The optimal chromosome (with binary or real values) is estimated using a wrapper method. Therefore, when the number of features and instances increases, these methods may face problems in terms of effectiveness and efficiency. On the other hand, our proposed wrapper method has two main differences. First, it uses only one chromosome for feature selection. The instance selection is performed using a data self-representation method, which provides the best subset of relevant instances given the subset of features.

Second, FIS algorithms are very often performed with wrapper techniques based on a genetic algorithm. These techniques can reduce the number of features and instances by penalizing low feature and instance reduction rates. There is a tradeoff between achieving high accuracy and high data reduction [43]. Their global fitness uses equilibrium parameters that can be difficult to tune. On the other hand, our scheme uses a different type of instance compression. By using SMRS instance selection, the number of selected instances can be either explicitly specified by the user or automatically estimated from the associated encoding matrix, since this matrix is a row-sparse matrix.

We emphasize the fact that whenever the number of original features becomes very large, the use of dimensionality reduction based on manifold learning such as Principal Component Analysis (PCA) can reduce the size of the feature chromosome from thousands of bits to just a few tenths. This property can benefit both our proposed schemes and the simultaneous FIS schemes. However, the simultaneous schemes cannot benefit from manifold dimension reduction techniques to reduce the size of the instance chromosome because its size is still equal to the number of instances in the original dataset. In our proposed scheme, there is no chromosome for the instances.
4.2 Proposed LDA-FSIS and LDE-FSIS schemes vs. sequential FSIS techniques

All filter methods adopting FS as a first step have two main shortcomings. First, the set of selected features is estimated independently of the instance selection step. Thus, the process of instance selection depends entirely on the solution found on the whole set of instance. Second, the filter methods require an a priori knowledge of the number of features to be selected. On the other hand, our proposed schemes (LDA-FSIS and LDE-FSIS) explore feature subset selection in a way that the instance selection intervenes in the scoring process of a given candidate feature subset. By doing so, there is an explicit interaction between FS and IS steps. Furthermore, due to the adopted search mechanism, our proposed schemes do not require prior knowledge about the number of selected features.

5 Performance evaluation

5.1 Datasets

Our experiments were conducted on five public image datasets, including handwritten digits, face images in which the pose of faces ranged from profile to frontal view, object images with a variety of complex geometry and reflectance properties, and image datasets described by high-level features.

In the following, we give a brief description of some of the features of these datasets.

- **USPS**: Handwritten Digit dataset consists of 11,000 grayscale images of the ten digits "0" to "9"; each class (digit) has 1100 images.
- **Sheffield**: It contains 575 facial images of 20 people. The poses of the faces range from profile to frontal views. The original image size is 112×92 pixels with 8-bit grey levels. In our experiments, the images are rescaled to 56×46 pixels.
- **COIL-20**: This dataset (Columbia Object Image Library) consists of 1440 images of 20 objects. Each object was subjected to 72 rotations (each object has 72 images). The objects exhibit a wide variety of complex geometry and reflectance properties.
- **Image Segmentation Dataset**: The image data is described by 19 numerically-valued

---

[^1]: [http://www.cs.nyu.edu/~roweis/data.html](http://www.cs.nyu.edu/~roweis/data.html)
[^2]: [https://www.sheffield.ac.uk/eee/research/iel/research/face](https://www.sheffield.ac.uk/eee/research/iel/research/face)
[^3]: [www.cs.columbia.edu/CAVE/software/softlib/coil-20.php](www.cs.columbia.edu/CAVE/software/softlib/coil-20.php)
[^4]: [https://archive.ics.uci.edu/ml/datasets/Image+Segmentation](https://archive.ics.uci.edu/ml/datasets/Image+Segmentation)
high-level attributes. The number of classes is 7. The whole dataset contains 2310 instances.

- **Extended Yale - part B**: It contains 38 human subjects, each with about 60 images. Lighting and facial expression variations are present in these images. The original size of the images was $192 \times 168$, in our experiments we scaled the images to $32 \times 32$.

### 5.2 Experimental setup

In this section, we compare the performance of the proposed methods (LDA-FSIS, LDE-FSIS, and W-FSIS) with that of state-of-the-art selection strategies. The performance is quantified by the classification accuracy (recognition rate) using the selected features and instances as training data. In our work, we adopted a commonly used protocol to evaluate the classification accuracy after selecting features and instances. First, each dataset is randomly divided into two parts: a training part and a testing part. Then, the training part is processed with a given selection method that selects a subset of features and a set of training instances. The test part is then recognized either with all training data and all features (in the tables representing the recognition results, this is called "All Data") or with the selected features and instances (in the tables, each method corresponds to a row). This automatic detection is performed by two classifiers: Nearest Neighbor (NN), and Support Vector Machines (SVM). Then, the recognition performance of each selection method or scheme can be quantified using the ground truth labels of the test set. Since the same training and test data are used, the difference in classification performance will almost certainly be due to the selection algorithm alone. For a fair comparison, all instance selection algorithms are used to provide the same number of selected instances.

We conduct three groups of experiments. In the first group, we compare the proposed methods with methods that use sequential processing, where feature selection is applied first and then instance selection is applied to the training sets with selected features. In the second group of experiments, we compare the proposed methods with methods that apply joint feature and instance selection. In the third group of experiments, we compare our proposed methods with some instance selection methods. We used the MATLAB genetic algorithm with the following main parameters: The number of generations is set to 20 and the population size is set to 150. For the first two sets of experiments, the data partitioning into training and test sets is given in Table 2. This table also

---

5 [http://vision.ucsd.edu/~leekc/ExtYaleDatabase/ExtYaleB.html](http://vision.ucsd.edu/~leekc/ExtYaleDatabase/ExtYaleB.html)
5.3 Experimental results

Table 2: Adopted training and testing parts for the five datasets. The last column depicts the percentage of selected instances with respect to the train part.

| Dataset      | Training | Testing | Selected Instances from Training |
|--------------|----------|---------|----------------------------------|
| Sheffield    | 93.9%    | 6.1%    | 18.5%                            |
| COIL-20      | 83.3%    | 16.7%   | 8.3%                             |
| USPS         | 83.3%    | 16.7%   | 4.0%                             |
| Segmentation | 90.9%    | 9.1%    | 10.0%                            |
| Extended Yale| 80.0%    | 20.0%   | 16.7%                            |

First group: Proposed schemes vs. FSIS schemes  For feature selection, we consider the following methods. Fisher method [23] is a well-known feature selection method. It computes a score for a feature as the ratio of interclass separation and intraclass variance, where features are evaluated independently, and the final feature selection occurs by aggregating the \( m \) top ranked ones. Feature Selection via SVM (FSV) [9] which is a wrapper method, where the feature selection process is injected into the training of an SVM by a linear programming technique. Mutual information (MutInf) method [67] is a filter method where the feature are first considered individually, ranked, and then a subset is extracted. The Maximum Relevance Minimum Redundancy (mRMR) algorithm [42] selects a subset of features having the most correlation with a class (relevance) and the least correlation between themselves (redundancy). The Relief algorithm [52] performs feature ranking based on the concept of margin that is computed for each feature.

We evaluate the classification performance of the methods in two scenarios. In the first scenario, we only apply feature selection using the whole set of training samples (FS). In this scenario, there is no instance selection. In the second scenario, we apply feature selection followed by instance selection where a given number of the most relevant samples are retained. In this scenario, all compared algorithms use the SMRS instance selection method. In our work we have adopted a commonly used protocol to evaluate the classification accuracy. We proceed as follows. The whole dataset is randomly split into two parts: a training part and a test part. For the first scenario, the training part is processed by a given feature selection method and the test part will be processed by selecting the features based on the found selected features selected from the training data. For the second scenario, feature selection is applied on the training data, then the obtained training
data (selected features) undergo an instance selection using the SMRS method.

Finally the recognition rate is quantified on the test data. The classification is carried out by two classifiers: Nearest Neighbor (NN) and Support Vector Machine (SVM). For the second scenario, and for a fair comparison, all compared algorithms are used such a way they use the same number of selected instances.

Tables 3-7 illustrate the obtained performance on the five image datasets. In these tables, the number of selected representatives per class by the IS method (SMRS) is depicted at the top of each table.

For the classic schemes (FS+IS) (shown in the first six rows in Tables 3-7, we used the 40 % top ranked features. The last row in each table illustrates the performance using all training data (i.e., all features and all samples). Best results are shown in bold. From these tables, we can observe the following.

- The FS methods that work on all samples can improve the classification performance; whereas for the FS+IS paradigms the performance decreases. This is very consistent with the essence of the objectives sought by FS and IS. The objective of FS methods is to reduce the features while maintaining or improving the final performance. The objective of IS methods is to reduce the instances while maintaining a performance close to the one obtained with the whole training dataset.

- For the FS methods without any instance selection, the proposed schemes LDA-FSIS and LDE-FSIS achieve start-of-the art performance and for some datasets they gave better results (the Segmentation and Extended Yale datasets).

- For the FS+IS methods, in general the proposed schemes outperformed the other competing methods.

- For the NN classifier, the proposed LDA-FSIS and LDE-FSIS outperformed the proposed Wrapper-FSIS scheme. For the SVM classifier, there is no clear conclusion about which scheme is the superior.

We emphasise that our three proposed methods automatically compute the number of selected features. We used a genetic algorithm that can automatically determine the suboptimal chromosome. In this chromosome, the number of bits taking the value of one is not predetermined and
is determined by the genetic algorithm performing a full feature subset selection (number and positions).

We also report the obtained performance of the competing method (FS + IS) on the dataset Extended Yale. In this case, the feature selection method is Fisher’s method and the instance selection method is SMRS. Figure 3 shows the color map of SVM performance as a function of selected attributes and selected representatives per class.

Figure 3: The SVM classifier’s performance versus the number of selected instances (per class) and the percentage of relevant attributes. The Extended Yale dataset was used. The method is given by the Fisher selection followed by the SMRS instance selection method.

**Second group: Proposed schemes vs. simultaneous FIS schemes** For the simultaneous feature and instance selection, we consider the following methods: (i) A wrapper method that follows a main trend for the simultaneous feature and instance selection. It is based on the classifier performance optimization, (ii) Instance and feature selection based on cooperative coevolution (IFS-coco) [12], and (iii) Feature Weighting and Instance Weighting (FWIW) [43]. According to the work described in [43] the simultaneous FWIW gave superior results to other schemes involving hard selection. Thus, in our comparison will use this variant.

Table 8 depicts a comparison between our proposed schemes and the above mentioned methods on the five datasets. The number of selected instances is the same for all compared methods that allow to fix this number. The latter is shown in the previous tables. As it can be seen, in general, the proposed LDA-FSIS and LDE-FSIS schemes have outperformed the competing simultaneous feature
and instance selection methods. This holds for the two classifiers. Our proposed methods LDA-FSIS and LDE-FSIS do not implement a simultaneous selection scheme. Instead, they optimize a score that depends on within-class and between-class graphs as well as on the best relevant instances for a given feature subset. Thus, our proposed schemes are still more efficient than the simultaneous feature and instance selection paradigms since the computational cost of the manifold-based fitness function is much less than carrying out an entire classifier training and testing. Regarding the compared wrapper methods (our proposed W-FSIS method and the competing method shown in the first row), the proposed W-FSIS outperformed the competing method in five configurations out of ten.

Third group: Proposed schemes vs. IS schemes

We also compare the proposed method with the following instance selection methods: Kmedoids [28], Collaborative Neighborhood Representation (CNR) [15], simple random selection of training data (Rand), Prototype Selection by Relevance (PSR) [39], DROP3 [62], Two Phase CNR (TPCNR) [13], Ensembles of instance selection methods based on feature subset (EIS) [7], SMRS [19], Decremental SMRS (D-SMRS) [15], and the nonlinear sparse modeling [14].

From the previous experiments, we can observe that the LDA-FSIS and the LDE-FSIS schemes are giving the best performance. In this group of experiments, we focus on the LDA-FSIS scheme. The number of selected instances is the same for all compared methods that allow to fix this number. The experimental setup adopted by the first two groups of experiments is slightly changed. Thus, we set the number of representatives per class for the Extended Yale, COIL-20, USPS, and Segmentation datasets to 12, 10, 50, and 60, respectively.

Table 9 depicts a comparison between our proposed LDA-FSIS and the mentioned instance selection methods on five datasets. We can observe that the proposed manifold-based scheme LDA-FSIS has outperformed all competing selection methods. This suggests that performing feature selection and instance selection using the proposed scheme can be better than using instance selection on the full original set of features.

Influence of the representative subset size. Figure 4 illustrates the recognition rate of the proposed method LDA-FSIS as a function of the number of representatives. The SVM classifier is used. We can observe that by increasing the number of representatives the classification performance improves. Note that the classification rate obtained with the whole training set (1000
### Table 3: Proposed schemes versus FSIS schemes on the Sheffield dataset.

| Method     | NN                  | SVM                  |
|------------|---------------------|----------------------|
| FS         | 98.7                | 98.1                 |
| FS + IS (K=5 samples/27) | 91.3                | 95.6                 |
| Fisher [23] | 98.7                | 91.3                 |
| FSV [9]    | 98.7                | 96.9                 |
| mRMR [67]  | 98.7                | 98.1                 |
| mutINF [42]| 98.1                | 91.3                 |
| Relief [52]| 98.1                | 92.3                 |
| LDE        | 98.1                | 89.5                 |
| LDA-FSIS   | 98.7                | 97.5                 |
| LDE-FSIS   | 98.7                | 96.2                 |
| W-FSIS     | 98.7                | 95.6                 |
| All data   | 98.1                | 98.1                 |

### Table 4: Proposed schemes versus FSIS schemes on the COIL-20 dataset.

| Method     | NN                  | SVM                  |
|------------|---------------------|----------------------|
| FS         | 100                 | 100                  |
| FS + IS (K=5 samples/60) | 87.9                | 92.5                 |
| Fisher [23] | 100                 | 100                  |
| FSV [9]    | 100                 | 100                  |
| mRMR [67]  | 100                 | 100                  |
| mutINF [42]| 100                 | 100                  |
| Relief [52]| 100                 | 93.3                 |
| LDE        | 100                 | 87.9                 |
| LDA-FSIS   | 100                 | 100                  |
| LDE-FSIS   | 100                 | 100                  |
| W-FSIS     | 100                 | 95.0                 |
| All data   | 100                 | 100                  |

### Table 5: Proposed schemes versus FSIS schemes on the USPS dataset.

| Method     | NN                  | SVM                  |
|------------|---------------------|----------------------|
| FS         | 96.6                | 97.8                 |
| FS + IS (K=20 samples/500) | 82.6                | 87.6                 |
| Fisher [23] | 96.6                | 97.8                 |
| FSV [9]    | 92.2                | 93.3                 |
| mRMR [67]  | 96.0                | 97.0                 |
| mutINF [42]| 96.4                | 97.2                 |
| Relief [52]| 95.2                | 96.6                 |
| LDE        | 96.0                | 97.4                 |
| LDA-FSIS   | 96.0                | 97.4                 |
| LDE-FSIS   | 95.8                | 97.2                 |
| W-FSIS     | 95.0                | 97.0                 |
| All data   | 94.6                | 98.0                 |
Table 6: Proposed schemes versus FSIS schemes on the Segmentation dataset.

| Method         | NN                  | SVM                  |
|----------------|---------------------|----------------------|
|                | FS                  | FS + IS (K=30 samples/300) | FS                  | FS + IS (K=30 samples/300) |
| Fisher [23]    | 90.9                | 75.7                 | 85.7                | 78.5 |
| FSV [9]        | 90.9                | 50.0                 | 79.5                | 63.3 |
| mRMR [67]      | 92.3                | 70.0                 | 80.9                | 72.8 |
| mutINF [42]    | 90.4                | 65.7                 | 71.9                | 71.4 |
| Relief [52]    | 93.1                | 70.1                 | 84.4                | 77.3 |
| LDE            | 92.8                | 72.8                 | 83.8                | 78.5 |
| **LDA-FSIS**   | **96.6**            | 86.1                 | **96.1**            | 88.6 |
| **LDE-FSIS**   | 95.2                | **86.6**             | **96.1**            | 89.5 |
| **W-FSIS**     | 94.7                | 83.8                 | 95.7                | **92.3** |
| All data       |                     |                      |                     | 95.2 |

Table 7: Proposed schemes versus FSIS schemes on the Extended Yale dataset.

| Method          | NN                  | SVM                  |
|-----------------|---------------------|----------------------|
|                 | FS                  | FS + IS (K=8 samples/48) | FS                  | FS + IS (K=8 samples/48) |
| Fisher [23]     | 80.0                | 59.5                 | 85.0                | 68.4 |
| FSV [9]         | 38.5                | 24.0                 | 80.0                | 60.9 |
| mRMR [67]       | 73.9                | 51.2                 | 82.2                | 69.8 |
| mutINF [42]     | 43.2                | 32.1                 | 82.2                | 64.5 |
| Relief [52]     | 71.1                | 50.8                 | 81.1                | 65.3 |
| LDE             | 45.9                | 31.0                 | 81.4                | 62.8 |
| **LDA-FSIS**    | **88.9**            | **69.8**             | **96.9**            | 81.4 |
| **LDE-FSIS**    | 88.3                | 62.3                 | 94.7                | 78.6 |
| **W-FSIS**      | 80.6                | 62.0                 | 95.2                | **82.2** |
| All data        |                      |                      |                     | 78.1                 | 98.6 |
Table 8: Comparison with simultaneous FIS methods. The used datasets are: Sheffield, COIL-20, USPS, Segmentation, and Extended Yale.

| Sheffield       | NN  | SVM |
|-----------------|-----|-----|
| FS&IS based on rate | 92.1| 96.4|
| IFS-coco [12]   | 93.4| 95.1|
| FWIW [43]       | 94.2| 97.1|
| LDA-FSIS        | 94.4| 97.5|
| LDE-FSIS        | 95.0| 96.2|
| W-FSIS          | 91.3| 95.6|

| COIL-20         | NN  | SVM |
|-----------------|-----|-----|
| FS&IS based on rate | 88.1| 93.4|
| IFS-coco [12]   | 89.5| 94.1|
| FWIW [43]       | 89.1| 94.3|
| LDA-FSIS        | 88.3| 92.5|
| LDE-FSIS        | 87.9| 91.6|
| W-FSIS          | 84.5| 95.0|

| USPS            | NN  | SVM |
|-----------------|-----|-----|
| FS&IS based on rate | 80.3| 85.4|
| IFS-coco [12]   | 81.1| 84.1|
| FWIW [43]       | 85.1| 91.3|
| LDA-FSIS        | 82.2| 88.0|
| LDE-FSIS        | 81.0| 86.2|
| W-FSIS          | 75.2| 80.2|

| Segmentation    | NN  | SVM |
|-----------------|-----|-----|
| FS&IS based on rate | 79.1| 83.5|
| IFS-coco [12]   | 82.3| 88.1|
| FWIW [43]       | 81.4| 88.7|
| LDA-FSIS        | 86.1| 88.6|
| LDE-FSIS        | 86.6| 89.5|
| W-FSIS          | 83.8| 92.3|

| Extended Yale   | NN  | SVM |
|-----------------|-----|-----|
| FS&IS based on rate | 59.4| 68.9|
| IFS-coco [12]   | 64.1| 71.3|
| FWIW [43]       | 67.3| 70.4|
| LDA-FSIS        | 69.8| 81.4|
| LDE-FSIS        | 62.3| 78.6|
| W-FSIS          | 62.0| 82.2|
Table 9: Comparison with IS methods. The used datasets are: Extended Yale, COIL-20, USPS, and Segmentation.

| Method                  | NN  | SVM | Rand | SVM |
|-------------------------|-----|-----|------|-----|
| **Extended Yale**       |     |     |      |     |
| Rand                    | 94.5| 97.7| 96.7 | 98.1|
| K-medoids [28]          | 93.1| 89.8| 96.3 | 97.1|
| PSR [39]                | 91.7| 92.1| 94.3 | 93.7|
| DROP3 [62]              | 88.8| 93.0| 92.9 | 90.8|
| EIS [7]                 | 88.8| 89.4| 93.9 | 97.7|
| CNR [15]                | 82.8| 89.4| 95.9 | 98.3|
| TPCNR [15]              | 85.8| 90.6| 96.1 | 98.8|
| SMRS [19]               | 80.9| 90.0| 93.9 | 96.8|
| D-SMRS [13]             | 89.0| 94.8| 95.2 | 96.0|
| Nonlinear Sparse Modeling [14] | | | | |
| **LDA-FSIS**            | 94.4| 99.1|      |     |
| **COIL-20**             |     |     |      |     |
| Rand                    | 89.1| 92.0| 91.2 | 93.7|
| K-medoids [28]          | 81.6| 84.6| 81.2 | 77.5|
| PSR [39]                | 71.7| 75.4| 89.6 | 93.7|
| DROP3 [62]              | 80.0| 83.6| 91.6 | 93.3|
| EIS [7]                 | 80.0| 90.0| 90.8 | 96.2|
| CNR [15]                | 89.3| 91.8| 92.0 | 96.6|
| TPCNR [15]              | 88.6| 90.1| 90.8 | 93.4|
| SMRS [19]               | 88.3| 94.3| 92.2 | 93.7|
| D-SMRS [13]             | 88.6| 93.4| 93.7 | 94.7|
| Nonlinear Sparse Modeling [14] | | | | |
| **LDA-FSIS**            | 94.4| 98.9|      |     |

| Method                  | NN  | SVM | Rand | SVM |
|-------------------------|-----|-----|------|-----|
| **USPS**                |     |     |      |     |
| Rand                    | 88.6| 92.4| 88.8 | 93.0|
| K-medoids [28]          | 88.8| 93.0| 82.8 | 80.0|
| PSR [39]                | 87.0| 89.4| 87.0 | 89.4|
| DROP3 [62]              | 85.8| 90.6| 85.8 | 90.6|
| EIS [7]                 | 80.9| 90.0| 80.9 | 90.0|
| CNR [15]                | 89.0| 94.8| 89.0 | 94.8|
| TPCNR [15]              | 81.6| 95.2| 81.6 | 95.2|
| SMRS [19]               | 82.6| 96.0| 82.6 | 96.0|
| D-SMRS [13]             | 81.6| 95.2| 81.6 | 95.2|
| Nonlinear Sparse Modeling [14] | | | | |
| **LDA-FSIS**            | 93.7| 98.0|      |     |
| **Segmentation**        |     |     |      |     |
| Rand                    | 90.8| 91.5| 92.0 | 93.7|
| K-medoids [28]          | 89.3| 91.8| 89.3 | 91.8|
| PSR [39]                | 86.9| 91.0| 86.9 | 91.0|
| DROP3 [62]              | 88.6| 90.1| 88.6 | 90.1|
| EIS [7]                 | 80.0| 83.6| 80.0 | 83.6|
| CNR [15]                | 88.3| 94.3| 88.3 | 94.3|
| TPCNR [15]              | 88.6| 93.4| 88.6 | 93.4|
| SMRS [19]               | 92.2| 93.7| 92.2 | 93.7|
| D-SMRS [13]             | 93.7| 94.7| 93.7 | 94.7|
| Nonlinear Sparse Modeling [14] | | | | |
| **LDA-FSIS**            | 94.4| 98.9|      |     |
examples per class) was 98.8%.

Figure 4: Effect of the number of selected instances (per class) on the final classification using the proposed LDA-FSIS with the USPS dataset.

6 Conclusion

This paper addresses the problem of joint feature and instance selection. We propose three schemes for combining feature selection and instance selection. All three schemes use an evolutionary scheme for finding the feature selection solution. The instance selection part is integrated into this evolutionary algorithm, which searches for the best subset of features. Instance selection is performed on candidate features using an efficient sparse modeling data self-representation scheme avoiding the use of a specific chromosome for the instance selection part. The first scheme is a wrapper technique that depends on the classifier used. The second and third schemes are filtering techniques that use fitness functions based on Linear Discriminant Analysis (LDA) and Local Discriminant Embedding (LDE) criteria, respectively.

The three proposed schemes have some interesting properties in terms of effectiveness and efficiency. The classifiers used in our work are: Nearest Neighbor (NN) and Support Vector Machines (SVM). The proposed schemes were compared with three categories of algorithms: (i) schemes that perform feature selection followed by instance selection, (ii) schemes that perform feature and instance selection simultaneously, and (iii) schemes that perform instance selection only. The experiments involved image classification using the selected features and instances. The results obtained by the proposed schemes seem to be more accurate than those obtained by the competing schemes. In particular, the two filtering schemes that adopt the manifold structure can be very
appealing.

Compliance with Ethical Standards:

This article complies with ethical standards.

Conflict of Interest:

The authors declare that they have no conflict of interest.

Ethical approval:

This article does not include studies with human participants or animals conducted by any of the authors.
References

[1] R. S. A. Aghazadeh, D. LeJeune, G. Dasarathy, A. Shrivastava, and R. Baraniuk. Ultra-large-scale feature selection using count-sketches. In ICML, 2018.

[2] H. Ahn and K. Kim. Bankruptcy prediction modeling with hybrid case-based reasoning and genetic algorithms approach. Applied Soft Computing, 9:599–607, 2009.

[3] A. Akinyelu and A. Adewumi. Improved instance selection methods for support vector machine speed optimization. Security and Communication Networks, 217, 2017.

[4] A. P. Angulo and K. Shin. Mrmr+ and Cfs+ feature selection algorithms for high-dimensional data. Applied Intelligence, 49(5):1954–1967, 2018.

[5] J. Apolloni, G. Leguizamón, and E. Alba. Two hybrid wrapper-filter feature selection algorithms applied to high-dimensional microarray experiments. Applied Soft Computing, 38:922–932, 2016.

[6] J. Bien and R. J. Tibshirani. Sparse estimation of a covariance matrix. Biometrika, 98:807–820, 2011.

[7] M. Blachnik. Ensembles of instance selection methods based on feature subset. Procedia Computer Science, 35:388–396, 2014.

[8] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends in Machine Learning, 3(1):1–122, 2011.

[9] P. S. Bradley and O. L. Mangasarian. Feature selection via concave minimization and support vector machines. In ICML, volume 98, pages 82–90, 1998.

[10] H.-T. Chen, H.-W. Chang, and T.-L. Liu. Local discriminant embedding and its variants. In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, volume 2, pages 846–853. IEEE, 2005.

[11] J.-H. Chen, H.-M. Chen, and S.-Y. Ho. Design of nearest neighbor classifiers: multi-objective approach. International Journal of Approximate Reasoning, 40(1-2):3–22, 2005.
[12] J. Derrac, S. Garcia, and F. Herrera. IFS-CoCo: instance and feature selection based on cooperative coevolution with nearest neighbor rule. *Pattern Recognition*, 43:2082–2105, 2010.

[13] F. Dornaika and I. K. Aldine. Decremental sparse modeling representative selection for prototype selection. *Pattern Recognition*, 48(11):3717–3727, November 2015.

[14] F. Dornaika and I. K. Aldine. Instance selection using non-linear sparse modeling. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(6):1457–1461, 2018.

[15] F. Dornaika, I. K. Aldine, and B. Cases. Exemplar selection using collaborative neighbor representation. In *Hybrid Artificial Intelligence Systems*, volume LNAI, 9121, 2015.

[16] F. Dornaika, I. K. Aldine, and A. Hadid. Kernel sparse modeling for prototype selection. *Knowledge-Based Systems*, 107(1):61–69, 2016.

[17] F. Dornaika and I. Kamal Aldine. Decremental sparse modeling representative selection for prototype selection. *Pattern Recogn.*, 48(11):3714–3727, Nov. 2015.

[18] W. Du, Z. Cao, T. Song, Y. Li, and Y. Liang. A feature selection method based on multiple kernel learning with expression profiles of different types. *BioData Mining*, 10(4), 2017.

[19] E. Elhamifar, G. Sapiro, and R. Vidal. See all by looking at a few: Sparse modeling for finding representative objects. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1600–1607. IEEE, 2012.

[20] E. Elhamifar and R. Vidal. Robust classification using structured sparse representation. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1873–1879, 2011.

[21] A. Fernández, C. J. Carmona, M. J. del Jesus, and F. Herrera. A pareto based ensemble with feature and instance selection for learning from multi-class imbalanced datasets. In *Proc. of the XVIII Conferencia de la Asociación Española para la Inteligencia Artificial (XVIII CAEPIA)*, pages 1316–1317, 2018.

[22] N. García-Pedrajas, J. A. R. Del Castillo, and D. Ortiz-Boyer. A cooperative coevolutionary algorithm for instance selection for instance-based learning. *Machine Learning*, 78(3):381–420, 2010.
[23] Q. Gu, Z. Li, and J. Han. Generalized fisher score for feature selection. *arXiv preprint arXiv:1202.3725*, 2012.

[24] S. Gunal and R. Edizkan. Subspace based feature selection for pattern recognition. *Inf. Sci.*, 178(19):3716–3726, Oct. 2008.

[25] P. Hernandez-Leal, J. A. Carrasco-Ochoa, J. F. Martínez-Trinidad, and J. A. Olvera-Lopez. Instancerank based on borders for instance selection. *Pattern Recognition*, 46(1):365–375, 2013.

[26] C. Hou, X. L. Nie, D. Yi, and Y. Wu. Joint embedding learning and sparse regression: A framework for unsupervised feature selection. *IEEE Trans. on Cybernetics*, 44(6):793–804, June 2014.

[27] H. Ishibuchi and T. Nakashima. Multi-objective pattern and feature selection by a genetic algorithm. In *Proceedings of the 2nd Annual Conference on Genetic and Evolutionary Computation*, pages 1069–1076. Morgan Kaufmann Publishers Inc., 2000.

[28] L. Kaufman and P. Rousseeuw. *Statistical Data Analysis based on the L1-Norm*, chapter 6. Clustering by means of medoids, pages 405–416. 1987.

[29] F. Keinosuke. Introduction to statistical pattern recognition. *Academic Press Inc*, 1990.

[30] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, et al. Optimization by simulated annealing. *science*, 220(4598):671–680, 1983.

[31] L. I. Kuncheva and L. C. Jain. Nearest neighbor classifier: Simultaneous editing and feature selection. *Pattern Recognition Letters*, 20(11):1149–1156, 1999.

[32] A. Kuri-Morales and F. Rodríguez-Erazo. A search space reduction methodology for data mining in large databases. *Eng. Appl. Artif. Intell.*, 22(1):57–65, Feb. 2009.

[33] E. Leyva, A. Gonzalez, and R. Perez. Knowledge-based instance selection: A compromise between efficiency and versatility. *Knowledge-Based Systems*, 47:65–76, 2013.

[34] Y. Li and L. Maguire. Selecting critical patterns based on local geometrical and statistical information. *IEEE transactions on pattern analysis and machine intelligence*, 33(6):1189–201, Jun 2011.
[35] H. Lim, J. Lee, and D.-W. Kim. Optimization approach for feature selection in multi-label classification. *Pattern Recognition Letters*, 89:25 – 30, 2017.

[36] Y. Liu, F. Nie, J. Wu, and L. Chen. Efficient semi-supervised feature selection with noise insensitive trace ratio criterion. *Neurocomputing*, 105:12 – 18, 2013.

[37] R. Mohamed, M. M. Yusof, and N. Wahidi. A comparative study of feature selection techniques for bat algorithm in various applications. In *MATEC Web of Conferences*, volume 150, 2018.

[38] F. Nie, S. Xiang, Y. Jia, C. Zhang, and S. Yan. Trace ratio criterion for feature selection. In *AAAI*, 2008.

[39] J. A. Olvera-Lopez, J. A. Carrasco-Ochoa, and J. F. Martínez-Trinidad. Prototype selection via prototype relevance. In *IberoAmerican Congress on Pattern Recognition, LNCS 5197*, 2008.

[40] S. Paul and S. Das. Simultaneous feature selection and weighting an evolutionary multi-objective optimization approach. *Pattern Recognition Letters*, 65:51–59, 2015.

[41] M. Pelikan and H. Mühlenbein. Marginal distributions in evolutionary algorithms. In *Proceedings of the International Conference on Genetic Algorithms Mendel*, volume 98, pages 90–95. Citeseer, 1998.

[42] H. Peng, F. Long, and C. Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence*, 27(8):1226–1238, 2005.

[43] J. Perez-Rodriguez, A. G. Arroyo-Pena, and N. Garcia-Pedrajas. Simultaneous instance and feature selection and weighting using evolutionary computation: Proposal and study. *Applied Soft Computing*, 37:416–443, 2015.

[44] J.-F. Ramirez-Cruz, O. Fuentes, V. Alarcon-Aquino, and L. Garcia-Banuelos. Instance selection and feature weighting using evolutionary algorithms. In *Computing, 2006. CIC’06. 15th International Conference on*, pages 73–79. IEEE, 2006.

[45] M. S. Raza and U. Qamar. An incremental dependency calculation technique for feature selection using rough sets. *Information Sciences*, 343-344:41–65, 2016.
[46] G. Roffo, S. Melzi, U. Castellani, and A. Vinciarelli. Infinite latent feature selection: A probabilistic latent graph-based ranking approach. arXiv:1707.07538, 2017.

[47] F. Ros, S. Guillaume, M. Pintore, and J. R. Chrétien. Hybrid genetic algorithm for dual selection. Pattern Analysis and Applications, 11(2):179–198, 2008.

[48] D. Ryu, J. Jang, and J. Baik. A hybrid instance selection using nearest-neighbor for cross-project defect prediction. Journal of Computer Science and Technology, 30(5):969–980, 2015.

[49] B. Sierra, E. Lazkano, I. Inza, M. Merino, P. Larrañaga, and J. Quiroga. Prototype selection and feature subset selection by estimation of distribution algorithms. a case study in the survival of cirrhotic patients treated with tips. In Conference on Artificial Intelligence in Medicine in Europe, pages 20–29. Springer, 2001.

[50] U. Staczyk, B. Zielosko, and L. C. Jain. Advances in Feature Selection for Data and Pattern Recognition. Springer, 2018.

[51] M. Suganthi and V. Karunakaran. Instance selection and feature extraction using cuttlefish optimization algorithm and principal component analysis using decision tree. Cluster Computing, 2018.

[52] Y. Sun, S. Todorovic, and S. Goodison. Local learning based feature selection for high dimensional data analysis. IEEE Transactions ON Pattern Analysis and Machine Intelligence, 32(9):1–18, 2010.

[53] J. Teixeira, R. Ferreira, and G. Lima. A novel approach for integrating feature and instance selection. In International Conference on Machine Learning and Cybernetics, 2008.

[54] J. A. Tropp. Column subset selection, matrix factorization, and eigenvalue optimization. In Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms, pages 978–986. Society for Industrial and Applied Mathematics, 2009.

[55] C. Tsai, W. Eberle, and C. Chu. Genetic algorithms in feature and instance selection. Knowledge-Based Systems, 39:240–247, 2013.

[56] C.-F. Tsai and J.-W. Wu. Using neural network ensembles for bankruptcy prediction and credit scoring. Expert Syst. Appl., 34(4):2639–2649, May 2008.
[57] G.-F. Tsai. Data pre-processing by genetic algorithms for bankruptcy prediction. *IEEE International Conference on Industrial Engineering and Engineering Management*, 2011.

[58] Y. Wang, S. Tang, Y.-D. Zhang, J.-T. Li, and D. Wang. Representative selection based on sparse modeling. *Neurocomputing*, 139:423 – 431, 2014.

[59] J. Wen, X. Fang, J. Cui, L. Fei, K. Yan, Y. Chen, and Y. Xu. Robust sparse linear discriminant analysis. *IEEE Transactions on Circuits and Systems for Video Technology*, 2018.

[60] J. Wen, N. Han, X. Fang, L. Fei, K. Yan, and S. Zhan. Low-rank preserving projection via graph regularized reconstruction. *IEEE Transactions on Cybernetics*, 49(4):1279–1291, 2019.

[61] J. Wen, Y. Xu, Z. Li, Z. Ma, and Y. Xu. Inter-class sparsity based discriminative least square regression. *Neural Networks*, 102:36–47, 2018.

[62] D. Wilson and T. Martinez. Reduction techniques for instance-based learning algorithms. *Machine Learning*, 38:257–286, 2000.

[63] S. Xiang, F. Nie, G. Meng, C. Pan, and C. Zhang. Discriminative least squares regression for multiclass classification and feature selection. *IEEE Transactions on Neural Networks and Learning Systems*, 23(11):1738–1754, 2012.

[64] W. Yang, D. Li, and L. Zhu. An improved genetic algorithm for optimal feature subset selection from multi-character feature set. *Expert Systems with Applications*, 38(3):2733–2740, 2011.

[65] Z.-X. Yin and J.-H. Chiang. Novel algorithm for coexpression detection in time-varying microarray data sets. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 5(1):120–135, Jan. 2008.

[66] G. Yuan, X. Chen, C. Wang, F. Nie, and L. Jing. Discriminative semi-supervised feature selection via rescaled least squares regression-supplement. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018.

[67] M. Zaffalon and M. Hutter. Robust feature selection using distributions of mutual information. In *Proceedings of the 18th International Conference on Uncertainty in Artificial Intelligence (UAI-2002)*, pages 577–584, 2002.

[68] X. Zhu, X. Li, S. Zhang, C. Ju, and X. Wu. Robust joint graph sparse coding for unsupervised spectral feature selection. *IEEE Trans. on Neural Networks and Learning Systems*, 28(6):1263–1274, June 2017.