Muñoz-Romero, S., Gómez-Verdejo, V. & Parrado-Hernández, E. (2017). A novel framework for parsimonious multivariate analysis. *Pattern Recognition, 71*, 173–186.

DOI: 10.1016/j.patcog.2017.06.004

A novel framework for parsimonious multivariate analysis

Sergio Muñoz-Romero[a,*], Vanessa Gómez-Verdejo[b], Emilio Parrado-Hernández[b]

[a] *Department of Signal Theory and Communications, Universidad Rey Juan Carlos, 28933 Fuenlabrada, Madrid, Spain*
[b] *Department of Signal Theory and Communications, Universidad Carlos III de Madrid, 28911 Leganés, Madrid, Spain*

Abstract

This paper proposes a framework in which a multivariate analysis method (MVA) guides a selection of input variables that leads to a sparse feature extraction. This framework, called parsimonious MVA, is specially suited for high dimensional data such as gene arrays, digital pictures, etc. The feature selection relies on the analysis of consistency in the behavior of the input variables through the elements of an ensemble of MVA projection matrices. The ensemble is constructed following a bootstrap that builds on an efficient and generalized MVA formulation that covers PCA, CCA and OPLS. Moreover, it allows the estimation of the relative relevance of each selected input variable. Experimental results point out that the features extracted by the parsimonious MVA have excellent discrimination power, comparing favorably with state-of-the-art methods, and are potentially useful to build interpretable features. Besides, the parsimonious feature extractor is shown to be robust against to parameter selection, as we all computationally efficient.

Keywords: Feature Selection, Dimensionality Reduction, Multivariate Analysis, Principal Component Analysis, Canonical Correlation Analysis, Orthonormalized Partial Least Squares

1. Introduction

Multivariate analysis (MVA) has become a keystone tool in the application of machine learning to solve pattern recognition problems. In a nutshell, MVA techniques preprocess the set of input variables to form a new, reduced set that captures the useful information and filters out redundancies and noise. Broadly used examples of MVA techniques are Principal Component Analysis (PCA) [1], Canonical Correlation Analysis (CCA) [2], Partial Least Squares (PLS) [3, 4], and Orthonormalized PLS (OPLS) [5, 6]. This dimensionality reduction capability becomes critical in domains in which the number of input variables is several orders of magnitude higher than the number of available data samples. Examples of these scenarios can be found in image classification, neuroimage, gene arrays processing, text classification, etc.

[*] Corresponding author.
Email addresses: sergio.munoz@urjc.es (Sergio Muñoz-Romero), vanessa@tsc.uc3m.es (Vanessa Gómez-Verdejo), emipar@tsc.uc3m.es (Emilio Parrado-Hernández)

From the application perspective, a main drawback of the use of MVA is that it obscures the interpretation of the machine learning outcome. For instance, consider a clinical application consisting in learning a score for a disease from a massive set of input variables (like outcomes of clinical essays, answers to questionnaires, etc). MVA would do an excellent job filtering non relevant information and merging redundancies into a compact set of features that would lead to a very good performance of a regression algorithm. However, it will not be possible, in general, to interpret the resulting model in terms of these original, valuable variables because they would appear melted within the features extracted by MVA.

The immediate choice for gaining interpretability can be to use a feature selection (FS) method that removes from the model definition all the non-informative input variables. An elegant way of providing this sparsity within linear models is with penalties in the cost function (see [7, 8] and references therein for a survey of sparse regularizations). In the bayesian framework, sparsity in the primal space is induced by the introduction of priors, such as the laplacian or the spike-lab, that drop irrelevant weights to zero [9, 10, 11]. Several extensions of these approaches have been applied to MVA algorithms; this is the case of the sparse PCA [12, 13, 14], the sparse CCA [14, 15], the sparse OPLS [16], or their bayesian formulations [17, 18, 19]. However, achieving sparsity in the coefficients of linear models does not lead, in general, to an easily interpretable solution from a MVA perspective. In MVA these coefficients form the projection matrix (the matrix that transforms the original variables in the new ones), and in order to gain in interpretability one needs that whole rows of the projection matrix become zero, and this property is not guaranteed by most of these approaches.

Parsimony, therefore, becomes a proper way of gaining interpretability through FS. Parsimony focuses on completely removing the participation of certain input variables in the definition of all the new features by zeroing whole rows of the projection matrix. Some works use iterative approaches to find the optimal subset of features which optimize a score related to parsimony [20]. Other alternatives induce parsimony through group Lasso regularisations [21, 22, 23], incurring in an exceedingly high computational cost due to their performing the optimisations in the primal space. Alternatively, other authors have proposed a solution to this problem by means of the $L_{2,1}$ regularisation [24], introducing both unsupervised [25, 26, 27] and supervised approaches [28, 29, 30, 31, 32, 33]. In particular, these methods yield a robust parsimonious feature extraction over a dual formulation, allowing its application in high dimensional problems [34]. However, this feature selection pursues to find out a subset of input variables that suffice to solve the problem at hand, involving the removal of not only noisy and but also redundant features.

The removal of redundancy hampers interpretability in those cases where each original input variable is not very informative by itself, like individual pixels in image classification applications. The training of humans in natural image classification tasks exploits the presence of patterns, regions, objects, etc. However, the input to computer image classification systems are pixels and each of them is managed in a isolated way. In cases in which the majority of pixels in the input images are not relevant for the classification (maybe it is based in small details present or not in the image), the automatic classifier will benefit from the previous application of an MVA or a feature selection preprocessing that select the most discriminative pixels. MVA plus automatic classification can lead to excellent results in terms of classification accuracy, but in exchange of a lack of interpretability of the results of the classification. The classifier becomes a black box that can not be used to gain further insight about those patterns and visual features that define the classification task. In our view the feature selection and feature extraction tasks hamper the interpretability because they not only remove irrelevant features, but also redundant ones. For those scenarios where some interpretability can be exchanged for accuracy, we propose to design feature filtering, as an alternative to feature selection and feature extraction that focuses on removing non-relevant input variables but preserving redundancy where this redundancy can help construct elaborated features with high semantic content.

Ensemble feature selection [35, 36, 37, 38, 39] is an elegant framework for a robust feature selection tailored to the particularities of each machine learning task. In essence these methods combine a pool of different classifiers, each trained on a bootstrapped version of the problem, and employ different strategies to determine the relevance of each feature by aggregating its role in the members of the ensemble [40]. If the bootstrapped training sets carry enough diversity, the selection can include redundancy. Following the classification of [41], there are two main streams in this framework, (1) feature selection for the ensemble, in which the final goal is to optimize the accuracy of the overall classifier resulting from the ensemble; and (2) feature selection by the ensemble, in which the ultimate goal of the method is to find a good set of selected features, and the bagging provides with consistency (in the sense of the goodness of the selected features) and robustness (with respect to the parameter selection) [42]. The work presented in this paper fits in this latter category. In this sense, [43] proposes an ensemble feature selection where the members of the ensemble are linear SVMs for regression with L_1 regularization and the features are selected according to their consistency in both sign and magnitude across the ensemble. The selected features were fed into a nonlinear SVM for regression. Our previous work in [44] goes an step further by using L_2 regularized linear SVMs classifiers in the ensemble. This enables to capture redundancy useful for unveiling clusters of voxels that explained a MRI classification.

Building on [44], this paper pursues to extend its feature selection by the ensemble, suited for binary classification, to a feature extraction and selection by the ensemble, that serves as a general framework for parsimonious, interpretable MVA. The contributions of this paper include:

- A methodology, based on an ensemble of feature extractors, which automatically captures the parsimony pattern underlying the input features. Moreover, this is accomplished without introducing any prior knowledge about this pattern.

- An estimation of the relevance of each feature for the task at hand, and the possibility of introducing this relevance in a further feature extraction stage.

- A regularised MVA that exploits the above estimated relevance. This regularisation forces that the contribution of each input variable in the MVA optimization becomes proportional to its relevance: informative (albeit redundant) input variables take a dominant role in the extracted features while noisy input variables end up with a negligible weight. Therefore, this regularised feature extraction method achieves accuracies comparable to those of a feature selection or extraction aimed at optimising accuracy.

In order to obtain efficient implementations, all the above contributions build on a general MVA formulation able to cover the most popular MVA approaches: PCA, CCA and OPLS. This formulation is based on weighted reduced rank regression problem [6], what yields results in a computational complexity given by the number of target variables. In comparison to other well-known MVA frameworks [45, 46] which scale with either the number of input dimensions or the number of input data (depending on the use of a primal or a dual formulation), this new formulation involves a severe computational cost reduction, since in most of the problems the number of output variables is even much smaller than the number of samples or input variables.

The remainder of the paper is as follows. Section 2 presents a general framework to formulate MVA problems suitable to introduce the consistency heuristic that performs the feature selection introduced in Section 3. Section 4 introduces parsimonious versions of PCA, CCA and OPLS that incorporate the characteristics described in this introduction. Section 5 shows experimental results with these algorithms in different problems, such as, face recognition and gene array classification. Finally Section 6 concludes the article.

2. A generalized formulation for regularized MVA

This section presents a generalized MVA formulation which covers the most well-known MVA methods: PCA, CCA and OPLS.

Let us consider a machine learning problem defined in terms of a collection of l input/output data pairs $\{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^{l}$. Observations are formed by d input variables $\mathbf{x}_n \in \mathbb{R}^d$ and targets $\mathbf{y}_n$ are formed by c output variables. This codification of the output serves for multiple output regression, classification, and unsupervised problems, such as, PCA.

- In the common single output regression case $c = 1$ and $\mathbf{y}_n$ is in fact an scalar $y_n \in \mathbb{R}$. In the multiple output regression case ($c \geq 2$) $\mathbf{y}_n \in \mathbb{R}^c$, $n = 1, \ldots, l$ and each element in $\mathbf{y}_n$ is the target in the corresponding regression problem.

- In the classification problems this notation means that targets $\mathbf{y}_n$ are binary indicator vectors [47]. Therefore, $c \geq 2$ (the number of classes must be greater or equal than 2) and $(\mathbf{y}_n)_i = 1$, if $\mathbf{x}_n$ belongs to class i and $(\mathbf{y}_n)_i = 0$, otherwise ($i = 1, \ldots, c$).

- In a PCA problem the target vector is directly the input vector as the task of the multivariate analysis is to come up with an optimal projection in the sense of achieving the best possible reconstruction the original input vectors.

The notation of the paper uses brackets to denote component of vectors or matrices, i.e., $(\mathbf{a}_n)_i$ refers to the i-th element of vector $\mathbf{a}_n$ and $(A)_{ij}$ is the element in position (i, j) of matrix A.

The input/output data pairs form two matrices: a input data matrix $X \in \mathbb{R}^{l \times d}$ and an $l \times c$ target matrix Y. For the remainder of the paper we consider, that matrices X and Y are centred, that is, their columns add up to zero. This centering allows us to define the sample covariance matrices $C_{XX} = l^{-1}X^TX$, $C_{YY} = l^{-1}Y^TY$, and $C_{XY} = l^{-1}X^TY$. The machine learning problems above presented (multiple regression, multiple classification, PCA) can be solved using formulations that lead to a solution that is a linear combination of the input variables. For example, the multiple regression case can be solved minimizing

$$\|Y - XM^T\|_F^2, \tag{1}$$

where $\|\cdot\|_F$ is the Frobenius norm operator. Each column of Y defines a single regression problem and each column of the $c \times d$ matrix M stores the coefficients of the corresponding linear regression model that minimizes the mean square error.

A general MVA problem extends the formulation in (1) with the introduction of an $d \times r$ projection matrix U that maps the input data onto a lower dimensional space with r features. The input matrix in this new space is $Z = XU$. The Least Square (LS) cost function defined with Y and Z is completed adding a constraint that enforces the orthogonality of the extracted features:

$$\mathcal{L}(W, U) = ||\left(Y - XUW^T\right)\Gamma^{\frac{1}{2}}||_F^2, \quad \text{s.t.} \;\; U^TX^TXU = I, \tag{2}$$

where W is the $c \times r$ solution matrix in the mapped data, Z, and matrix Γ allows us to recover the different MVA methods:

- In CCA Γ is the inverse of the sample covariance matrix of the target data $\Gamma = l(Y^TY)^{-1} = C_{YY}^{-1}$,

- in OPLS $\Gamma = I$ is the identity matrix,

- and in PCA $\Gamma = I$ and $Y = X$.

Following [6], the constraint $U^T X^T X U = I$ can be replaced by an equivalent one in W, $W^T \Gamma W = I$ (see Appendix A for a detailed derivation). Therefore, the optimization (2) becomes

$$\mathcal{L}(W, U) = || \left(Y - XUW^T \right) \Gamma^{\frac{1}{2}} ||_F^2, \quad \text{s.t.} \;\; W^T \Gamma W = I. \tag{3}$$

Since this work focuses on high dimensional problems ($d \gg l$), the optimization of (3) becomes computationally more efficient using its dual formulation. As the mapping matrix U happens to be a linear combination of the inputs, it can be written as $U = X^T A$, where A is a matrix with the dual variables or coefficients of these linear combinations. Defining $K_x = XX^T$ as the linear kernel (Gram) matrix of the input data, the optimization problem (3) can be reformulated as

$$\mathcal{L}(W, A) = || \left(Y - K_x A W^T \right) \Gamma^{\frac{1}{2}} ||_F^2, \quad \text{s.t.} \;\; W^T \Gamma W = I. \tag{4}$$

Problem (4) turns out to be ill-conditioned since K_x is positive semidefinite (notice that centering X gives at least an eigenvalue of K_x equal to 0). This situation is fixed by including a regularization term $||A||_F^2$, with its corresponding regularization parameter λ. These changes leave (2) as

$$\mathcal{L}(W, A) = || \left(Y - K_x A W^T \right) \Gamma^{\frac{1}{2}} ||_F^2 + \lambda ||A||_F^2, \quad \text{s.t.} \;\; W^T \Gamma W = I. \tag{5}$$

Then, (5) can be solved in two steps: first, fixing W and solving for A yields

$$A = (K_x K_x + \lambda I)^{-1} K_x Y \Gamma W \tag{6}$$

Second, we introduce (6) in (5) and after some algebra arrive at a cost function depending only on W:

$$\mathcal{L}(W) = \mathrm{Tr}\left\{\Gamma C_{YY}\right\} - \mathrm{Tr}\left\{ W^T \Gamma Y^T K_x (K_x K_x + \lambda I)^{-1} K_x Y \Gamma W \right\}, \quad \text{s.t.} \;\; W^T \Gamma W = I. \tag{7}$$

Problem (7) can be formulated as a generalized eigenvalue problem,

$$\Gamma Y^T K_x (K_x K_x + \lambda I)^{-1} K_x Y \Gamma W = \Gamma W \Sigma. \tag{8}$$

where Σ is a diagonal matrix with the corresponding eigenvalues. Defining $V = \Gamma^{\frac{1}{2}} W$ enables to rewrite (8) as a standard eigenvalue problem:

$$\Gamma^{\frac{1}{2}} Y^T K_x (K_x K_x + \lambda I)^{-1} K_x Y \Gamma^{\frac{1}{2}} V = V \Sigma. \tag{9}$$

Thus, once V is computed, A can be retrieved from

$$A = (K_x K_x + \lambda I)^{-1} K_x Y \Gamma^{\frac{1}{2}} V. \tag{10}$$

An immediate advantage of this formulation is that the eigenvalue problem given by (9) involves matrices of size c instead of the size l matrices of the standard kernel MVA implementations. This fact reduces drastically the computational cost in almost all cases since usually $c \ll l$: the number of classes in a classification problem or the single regression problems in a multiple regression case is usually much smaller than the number of training data.

3. Feature filtering guided by bagged MVAs

This section introduces a new FS method that exploits a heuristic to spot which features behave in a similar/consistent way in the definition of all the components of the mapping Z found by a MVA. Each observation $\mathbf{x}$ is mapped into $\mathbf{z}$ (the corresponding column of Z), following $\mathbf{z} = \mathbf{x}^T U$. Component wise

$$(\mathbf{z})_k = \sum_j (\mathbf{x})_j (U)_{kj}, \qquad k = 1, \ldots, r,$$

where r is the number of principal components found by the MVA method.

When the features forming each observation $\mathbf{x}$ are pixels, voxels, grayscale values, probabilities, etc, the role of each $(\mathbf{x})_j$ in the definition of $(\mathbf{z})_k$ has an interpretable meaning. A highly positive value of $(U)_{kj}$ means that input patterns with a high value of feature $(\mathbf{x})_j$ push towards a large (positive) $(\mathbf{z})_k$. Conversely, a highly negative value for $(U)_{kj}$ means that patterns with high $(\mathbf{x})_j$ push towards a highly negative component $(\mathbf{z})_k$. A small absolute value for $(U)_{kj}$ points out that input feature $(\mathbf{x})_j$ bears little relevance in the definition of the k-th component of the mapping. A dummy heuristic for an interpretable FS would be to carry out an MVA to discard irrelevant features and select features with similar behaviour across all the principal components; however, this strategy poses a risk of overfitting in high dimensional problems. As advised in [43, 44], this overfitting is dramatically alleviated by a bagging procedure [48].

The bagging consists in repeating P times the MVA each time but with different input matrices of size $m \times d$ obtained randomly sampling m rows of X. This way, one can obtain P slightly different projection matrices from the same scenario and capture robust consistency patterns in the behaviour of coefficients $(U)_{kj}$. This robuster heuristic consists in averaging the one described in the previous paragraph across all the P bagging iterations. In more detail, let us denote as U^p, $p = 1, \ldots, P$, the projection matrix provided by each bagging iteration and construct a consistency matrix, B, in the following way:

$$(B)_{kj} = \left| \sum_{p=1}^{P} \mathbb{I}((U^p)_{kj} > 0) - P/2 \right|, \quad \text{where } \mathbb{I}(\cdot) \text{ is the indicator function.} \tag{11}$$

Notice that a high value of $(B)_{kj}$ indicates that the behaviour of feature $(\mathbf{x})_j$ in the definition of $(\mathbf{z})_k$ is quite stable across all the bagging iterations. Our intuition indicates a physical relationship between $(\mathbf{x})_j$ and $(\mathbf{z})_k$. This way, we can spot the critical input features by adding the consistency values achieved across all the principal components:

$$(\mathbf{b})_j = \sum_{k=1}^{r} (B)_{kj}, \qquad j = 1, \ldots, d. \tag{12}$$

Vector $\mathbf{b}$ stores the overall consistency of each input feature. The final selection consists in sorting the input variables according to their decreasing overall consistency and select as relevant features the subset S whose consistency exceeds a certain threshold t. This threshold can be fixed using prior domain knowledge or cross validation.

The procedure described in the previous paragraphs selects input features that explain either with a positive correlation or with a negative one (high $(B)_{kj}$) a good number of the extracted features, that is, it selects those input features having a large value of $(\mathbf{b})_j$ what indicates that $(\mathbf{x})_j$ participates in many $(\mathbf{z})_k$.

Regarding the computational burden of this method, it is important to note that the generalized kernel MVA formulation presented in Section 2 leads to dramatical reductions in the cost of the bagging. Algorithm 1 shows that the bagging just involves subsampling two matrices and multiplying the resulting submatrices, instead of solving a generalized eigenvalue problem. Moreover, in cases in which K_x is too large and step 1 becomes computationally unaffordable, we could save memory and get additional computational cost reductions by applying a more aggressive subsampling and computing K_x inside the bagging loop using X^p (i.e., Line 1 of Algorithm 1 could be included inside the loop of Lines 2-5). Furthermore, this bagging scheme can be straightforwardly implemented in a map-reduce paradigm, providing an embarrassingly fast implementation.

4. Parsimonious MVA formulation from selected features

The previous section presents a FS method that, on the one hand, captures features with a consistent behaviour in the definition of all the components in the mapping. This translates into a parsimony structure: a sparsity pattern common to the definition of all the features. On the other hand, the outcome of the bagging can be used to somehow assess the relevance of each input feature: the higher the consistency across all the bagging iterations, the higher the relevance. These two outcomes are combined to yield the parsimonious MVA. This method is basically an MVA whose principal components are built using exclusively the input features selected by the FS method. Moreover, the contribution of each selected input variable in the extracted features is regularised with a term that is proportional to the relevance of the variable learned in the bagging.

The parsimonious MVA is applied to the set of selected features S (let $X_{\mathcal{S}}$ be the submatrix of X whose columns are the selected features). In most situations the number of selected features will be

Algorithm 1 PseudoCode FS guided by bagged MVA.

Input: number of rows to be sampled from X and A: m, number of bagging iterations: P, consistency threshold: t, training data: $X = [\mathbf{x}_1^T, \ldots, \mathbf{x}_l^T]^T$, $Y = [\mathbf{y}_1^T, \ldots, \mathbf{y}_l^T]^T$.

Output: S: Set of relevant features, $\{U^p\}_{p=1}^P$: Set of projection matrices

```
With the complete $X$, solve eigenvalue problem (9) and compute matrix $A$ with (10).
for $p = 1 \to P$ do
    Sample $m$ rows of $A$ and $X$: $A^p = [\mathbf{a}_{M_1}^T, \ldots, \mathbf{a}_{M_m}^T]^T$ and $X^p = [\mathbf{x}_{M_1}^T, \ldots, \mathbf{x}_{s_M}^T]^T$, being $M$ the subset of $m$ subsampled data.
    Obtain the primal eigenvectors $U^p = (X^p)^T A^p$.
end for
Obtain consistency matrix $B$ using (11) and add up consistencies using (12).
for $j = 1 \to d$ do
    if $(\mathbf{b})_j > t$ then
        $S = S \cup (\mathbf{x})_j$
    end if
end for
```

less than the number of data ($|S| < l$), therefore it is more convenient to start from the generalised MVA formulation in the primal space of (3), i.e.,

$$\mathcal{L}(W, U) = \| \left(Y - X_{\mathcal{S}} U W^T \right) \Gamma^{\frac{1}{2}} \|_F^2, \quad \text{s.t.} \;\; W^T \Gamma W = I. \tag{13}$$

This formulation is completed with a regularization term that resembles a $\ell_{2,1}$-penalization term. The proposed regularization consists in a $d \times d$ diagonal matrix Ω, where each of its elements, $(\Omega)_{jj}$, $j = 1, \ldots, d$, emphasizes or penalizes the corresponding selected input variable $(\mathbf{x})_j$ according to its relevance across the bagging iterations:

$$\Omega_{jj} = \frac{1}{2||\bar{\mathbf{u}}_j||_2}, \tag{14}$$

being $\| \cdot \|_2$ the Euclidean norm operator and $\bar{\mathbf{u}}_j = \frac{1}{P} \sum_{p=1}^P \mathbf{u}_j^p$ the averaged value of the j^{th} row of the projection matrix U^p obtained after P bagging iterations.

After the introduction of the proposed regularization (13) becomes

$$\mathcal{L}(W, U) = \| \left(Y - X_{\mathcal{S}} U W^T \right) \Gamma^{\frac{1}{2}} \|_F^2 + \lambda ||\Omega^{\frac{1}{2}} U||_F^2, \quad \text{s.t.} \;\; W^T \Gamma W = I. \tag{15}$$

Note that, unlike $\ell_{2,1}$-penalization term, this regularization factors have been learned by the bagging procedure, being more stable than other approaches which estimate them with an unique execution. In fact, as we will analyse in the experimental section, this regularisation endows the feature extraction with a strong robustness and stability against weaknesses that can arise during the FS, such as a bad selection of the threshold t or of the number of selected features (notice that the overall FS is in fact a combination of several FSs, one per each component of the mapping).

Now, to obtain the solution of (15), we can follow a similar process to that described in Section 2, but applied over the primal formulation; thus, U is given by

$$U = (C_{X_\mathcal{S} X_\mathcal{S}} + \lambda\Omega)^{-1} C_{X_\mathcal{S} Y} \Gamma^{\frac{1}{2}} V, \tag{16}$$

where $C_{X_\mathcal{S} X_\mathcal{S}} = l^{-1} X_\mathcal{S}^T X_\mathcal{S}$, $C_{X_\mathcal{S} Y} = l^{-1} X_\mathcal{S}^T Y$, $W = \Gamma^{-\frac{1}{2}} V$; and V is the solution of the following eigenvalue problem:

$$\Gamma^{\frac{1}{2}} C_{X_\mathcal{S} Y}^T (C_{X_\mathcal{S} X_\mathcal{S}} + \lambda\Omega)^{-1} C_{X_\mathcal{S} Y} \Gamma^{\frac{1}{2}} V = V\Sigma. \tag{17}$$

Finally, taking into account that $(C_{X_\mathcal{S} X_\mathcal{S}} + \lambda\Omega)^{-1} C_{X_\mathcal{S} Y} \Gamma^{\frac{1}{2}}$ has to be computed in both U and V equations, we can obtain an efficient implementation with some manipulations. For this purpose, we firstly propose to calculate this common matrix as

$$U' = (C_{X_\mathcal{S} X_\mathcal{S}} + \lambda\Omega)^{-1} C_{X_\mathcal{S} Y} \Gamma^{\frac{1}{2}}.$$

and, then, we can rewrite (17) and (16) in terms of U' as $\Gamma^{\frac{1}{2}} C_{X_\mathcal{S} Y}^T U' V = V\Sigma$ and $U = U'V$, respectively. Algorithm 2 summarizes the main steps of this approach.

Algorithm 2 PseudoCode Parsimonious MVA.

```
Input: number of rows to be sampled from $X$ and $A$: $m$, number of bagging iterations: $P$, consistency threshold: $t$,   training data: $X = [\mathbf{x}_1^T, \ldots, \mathbf{x}_l^T]^T$, $Y = [\mathbf{y}_1^T, \ldots, \mathbf{y}_l^T]^T$.
Output: $S$: Set of selected features,   $U$: projection matrix from the selected features.

Obtain $S$ and $\{U^p\}_{p=1}^P$ following Algorithm 1: FS guided by bagged MVA.
Obtain $\Omega$ from $\{U^p\}_{p=1}^P$ using (14).
Using the set of selected features ($X_\mathcal{S}$), obtain the projection vector matrix:
    (a) $U' = (C_{X_\mathcal{S} X_\mathcal{S}} + \lambda\Omega)^{-1} C_{X_\mathcal{S} Y} \Gamma^{\frac{1}{2}}$.
    (b) $\Gamma^{\frac{1}{2}} C_{X_\mathcal{S} Y}^T U' V = V\Sigma$.
    (c) $U = U'V$.
```

5. Experimental results

The previous sections have introduced a novel framework to develop parsimonious versions of classic MVA methods. Therefore, this section is devoted to an empirical evaluation of the capabilities of these new parsimonious MVAs in some classification problems. The main outcome of these enhanced MVAs is a set of extracted features, each constructed using as basis the same reduced set of the original input variables that form a parsimony pattern.

Then, the focus of the study presented in this section is three fold. First of all we will look at the accuracy achieved by a classifier fed with the extracted features. A high accuracy will point out that the features indeed acquire that information relevant for the problem at hand. The second part of the study concentrates on the stability of the parsimony pattern formed by the selected original variables across different realisations of the experiment with different training/testing partitions. One of our motivations for this work is to capture the parsimonious structure underlying the data that is responsible for the definition of the problem. The stability of this pattern opens doors for the design of high level features suitable for human interpretation. Finally, the last part of the study aims at a detailed analysis of each of the individual contributions that conform the framework, the feature selection, the feature extraction and the regularisation.

5.1. Experimental setup

We have selected six multiclass problems in which the number of input variables is significantly larger than the number of available observations $d \gg l$. Table 1 summarises their main characteristics. Two datasets are gene arrays obtained from [24]: Carcinoms and LUNG. The other four datasets are face recognition tasks: Yale, ORL, PIE from [49], and a preprocessed excerpt of “Labeled Faces in the Wild” (LFW)[1]. In the Yale, ORL, and PIE data sets, we have used the partitions available at `http://www.cad.zju.edu.cn/home/dengcai/Data/FaceData.html`.

Table 1: Summary of benchmark datasets: size of training ($\mathrm{l}_{\mathrm{train}}$) and test subsets ($\mathrm{l}_{\mathrm{test}}$), input dimension ($d$), number of categories (c) and number of training images per class (p).

Name	l_{train}	l_{test}	d	c
Carcinoms	139	35	9182	11
LUNG	162	41	3312	5
Yale ($p = 8$)	120	45	4096	15
ORL ($p = 8$)	320	80	4096	40
PIE ($p = 10$)	680	10874	1024	68
LWF ($p \geq 70$)	1030	258	1850	7

We have implemented parsimonious versions of the three feature extraction approaches included in our general MVA formulation that we term p-PCA, p-CCA and p-OPLS in the presentation of the

[1] `http://vis-www.cs.umass.edu/lfw/lfw-funneled.tgz`

results. For comparison purposes we include five baseline methods:

- RFS (Robust Feature Selection): the standard $L_{2,1}$ approach for feature selection [24] (it does not include any feature extraction process).
- L21SDA: a parsimonious feature extraction based on an $L_{2,1}$ norm regularization [30]. It can be regarded as a CCA with $L_{2,1}$ norm regularization.
- SRRR: whose formulation is equivalent to the OPLS extraction combined with an $L_{2,1}$ norm regularization [31].
- SCM (Simultaneous Capped L_2-norm loss and $L_{2,1}$-norm regularizer Minimization): a robust feature selection based on capped L_2 norm loss function and capped $L_{2,1}$ norm regularization [50] (it does not include any feature extraction process).
- DFS (Discriminative Feature Selection): a parsimonious Linear Discriminant Analysis (LDA) feature extraction based on an $L_{2,1}$ norm regularization [33].

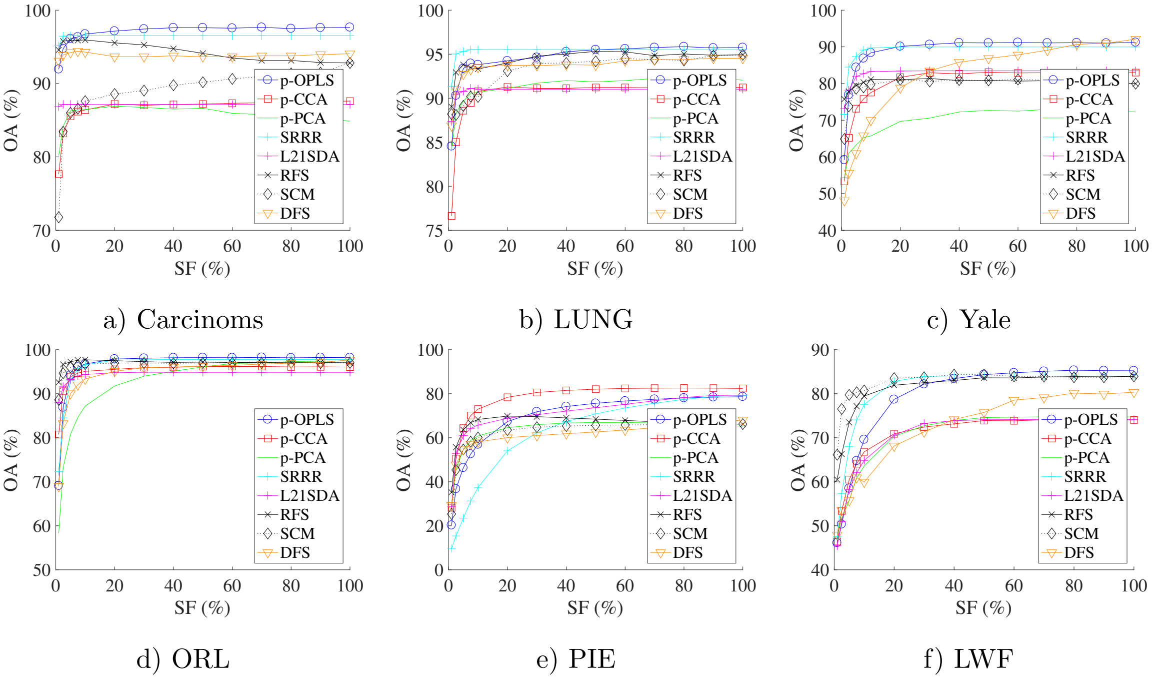
With respect to the hyperparameters of the training of the parsimonious MVAs, each bagging comprises a total number of $P = 10000$ projection matrices, each one learned with $m = 50\%$ of the training samples. We have checked empirically that these hyperparameters do not need special tuning (the observed results are very robust against reasonable selections of these parameters). The values of the regularization parameter λ in equations (5) and (15) have been crossvalidated for each problem within the following range: $\{10^{-6}, 5 \cdot 10^{-6}, 10^{-5}, 5 \cdot 10^{-5}, \ldots, 50, 100, 500, 1000\}$.

All the sets of selected or extracted features are used as input for classification stage implemented by linear SVM classifiers with their regularization parameter set to $C = 1$. The test results displayed in the sequel correspond to averages over 50 different train/test partitions with 80% of the dataset used for training. Hyperparameters were crossvalidated by averaging ten separate 80/20 random partitions of each training set.

5.2. Accuracy of the extracted features

This section analyzes the discriminative power of the features extracted by the parsimonious MVA in comparison to the $L_{2,1}$ baseline methods. For this purpose, we are going to use two evaluation measurements: (1) the Overall Accuracy (OA) or ratio of correct over total classifications; and (2) the Multiclass Area Under the Curve (MAUC) [51], which is an extension of the AUC to multi-class problems.

Figure 1 displays OA achieved by an SVM trained with the extracted features vs. the percentage of Selected Features (SF) defining the parsimony structure. In the supervised MVAs the number of



a) Carcinoms

b) LUNG

c) Yale

d) ORL

e) PIE

f) LWF

Figure 1: Overall accuracy (OA) vs. percentage of selected features (SF) for the different methods under study.

extracted features was set to the number of output classes minus one. For analogy, the number of principal components in p-PCA was also set to this number.

Methods RFS, SRRR and SCM achieve better performance with very low numbers of selected features in problems LUNG, Carcinom, Yale, ORL and LWF. This is mainly due to their focusing on spotting strictly relevant input variables. However, after the number of selected features goes beyond a certain value, the five baseline methods performance worsens since they are not able to handle redundant and irrelevant variables properly. However, the performance of parsimonious MVAs does not get worse as the number of features in the parsimony pattern increases. This characteristic endows these methods with great robustness against bad choices of the number of features, facilitating the tuning of this parameter. Besides, it is remarkable the high accuracy achieved by p-PCA, despite being an unsupervised approach. These results are corroborated with those of Table 2, which includes a comparison of these methods, in terms of MAUC, using only 50% of the features.

Figure 2 completes this study showing the performance of the methods in feature extraction. The contour plots show the classification accuracy for different numbers of extracted features (r, size of the input of the final classifier), and different sizes of the parsimony pattern (SF). In the colormap red indicates the highest accuracy and blue the lowest. For briefness, this analysis is carried out with problems Yale and ORL, although similar conclusions can be extracted using any of the other data sets.

Table 2: Evaluation of the different methods under study with a 50% of selected features in terms of MAUC.

	SBCCA	SBOPLS	SBPCA	L21SDA	SRRR	RFS	SCM	DFS
Carcinom	**0.997** ± 0.005	0.997 ± 0.005	0.996 ± 0.006	0.993 ± 0.010	0.992 ± 0.014	0.994 ± 0.012	0.993 ± 0.012	0.997 ± 0.006
LUNG	0.994 ± 0.008	0.992 ± 0.010	0.993 ± 0.010	0.993 ± 0.010	0.994 ± 0.006	**0.996** ± 0.004	0.995 ± 0.005	0.995 ± 0.005
Yale	**0.974** ± 0.010	0.964 ± 0.014	0.967 ± 0.012	0.972 ± 0.008	0.930 ± 0.017	0.965 ± 0.013	0.969 ± 0.012	0.963 ± 0.011
ORL	**0.999** ± 0.001	0.998 ± 0.003	0.997 ± 0.003	0.998 ± 0.003	0.999 ± 0.001	0.999 ± 0.002	0.998 ± 0.002	0.998 ± 0.002
PIE	0.971 ± 0.003	0.967 ± 0.004	0.920 ± 0.005	0.974 ± 0.002	0.972 ± 0.003	**0.981** ± 0.002	0.977 ± 0.003	0.949 ± 0.004
LWF	**0.548** ± 0.034	0.491 ± 0.019	0.470 ± 0.025	0.538 ± 0.030	0.546 ± 0.045	0.519 ± 0.015	0.539 ± 0.025	0.506 ± 0.043

The plots clearly show that the parsimonious MVAs provide the most accurate classifications with the lower number of selected features. This is specially remarkable in problem ORL; for this database, p-OPLS, p-CCA and p-PCA obtain its maximum performance with $r = 8$ extracted features, whereas L21SDA and SRRR need almost 40 and DFS around 20 features. With respect to the influence of the size of the parsimony pattern, the baseline methods suffer accuracy decreasing as SF increases for a wide range of values of the extracted features (r). However, the accuracy of the parsimonious MVAs never decreases as the parsimony pattern is enriched with more input features.

5.3. Stability analysis

Stability is a crucial property in a feature selection method. Stable methods would consistently select the adequate subset of features under different variations of the problem settings. These different conditions can be variations in the signal to noise ratio of the input features, in the values of parameters of the algorithms or different realizations of the training and test data. This section studies the stability of the parsimonious MVA using a synthetic classification problem in which the informative and the noisy features are known beforehand.

All datasets contain a total of 20 samples that belong to five output classes, and are generated in the following manner. First, 20 target vectors $\{\mathbf{y}_n\}_{n=1}^{20}$ are generated with equal probability from each output class. The $\{\mathbf{y}_n\}_{n=1}^{20}$ are encoded as 5 components vectors with 4 components set to -1 and a single component set to 1 indicating the correct output class (a 1 in the i-th position means the correct output class is the i-th one, $i = 1, \ldots, 5$).

Then an observation $\mathbf{x}_n$ formed by 2000 features is constructed for each $\mathbf{y}_n$, $n = 1, \ldots, 20$. The 2000 features come from three groups $\mathbf{x}_n = [\mathbf{f}_n^T, \mathbf{h}_n^T, \mathbf{g}_n^T]^T$:

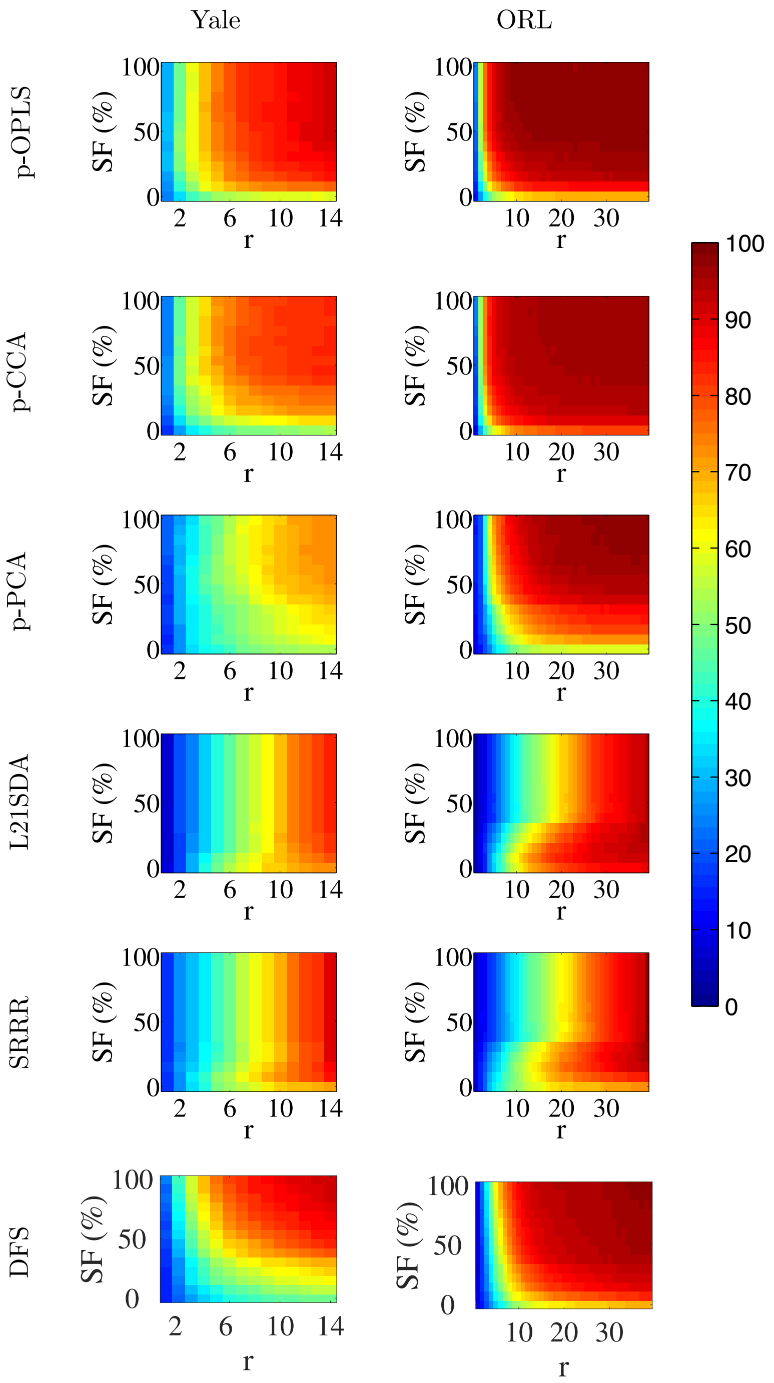


Figure 2: Accuracy (OA) of the different methods under study as a function of the size of the parsimony pattern (number of selected input features, SF) and of the number of extracted features (r).

1. The first subset of 200 input features ($\mathbf{f}_n$) are relevant since they are a linear combination of the target vector plus an additive Gaussian noise ($\mathbf{v}$) with zero mean and standard deviation

Table 3: Parameters explored during the stability analysis.

Parameter	Explored values
Level of noise over the redundant features (σ_r)	$\{10^{-5}, .1\}$
Regularization parameter of the feature extractor (λ)	$[10^{-6}, 10^{3}]$
Percentage of the selected features (SF)	$[1\%, 50\%]$.

$\sigma = 0.1$:

$$\mathbf{f}_n = W_r^T \mathbf{y}_n + \mathbf{v},$$

where W_r is a constant[2] 5×200 matrix with its elements randomly selected in $[0, 1]$.

2. The next 800 input features $\mathbf{h}_n$ are also informative variables but redundant as they are linear combinations of the relevant features plus additive Gaussian noise with zero mean and variance σ_r^2.
3. The remaining 1000 input features ($\mathbf{g}_n$) are drawn from independent Gaussian distributions with zero mean and unit variance, and do not take part in the construction of the target variables.

We have generated 948 different variations of the problem. Each variation is characterized by its own realization of the training set (with a particular value of σ_r), value of the regularization parameter λ and a percentage of features to select. Table 3 details the ranges of values analyzed for these parameters that configure the problem variations. Each of these situations was addressed with all the algorithms used throughout the experimental section: the proposed methods, p-OPLS and p-CCA, and the baseline approaches RFS, L21SDA, SRRR, SCM and DFS.

Figure 3 shows the results achieved by each method in the synthetic feature selection task. Each point in the scatter plots corresponds to one of the 948 situations. The x-axis sorts the simulations according to the percentage of features that each method had to select. The y-axis shows the percentage of these selected features that turned out to be actually informative, i.e., the percentage of the selected features in each simulation that came from the corresponding relevant and/or redundant groups of features.

Results are quite conclusive, the proposed methods are clearly more stable than the baseline ones. RFS, L21SDA, SRRR, SCM and DFS tend to include a significant percentage of non-informative features, while p-CCA and p-OPLS behaviors are close to ideal. Notice how for numbers of selected features smaller than 200 both p-CCA and p-OPLS achieve almost a perfect precision (almost all the selected features are informative).

[2] It changes in each realisation of the dataset, but it is constant within a same realisation.

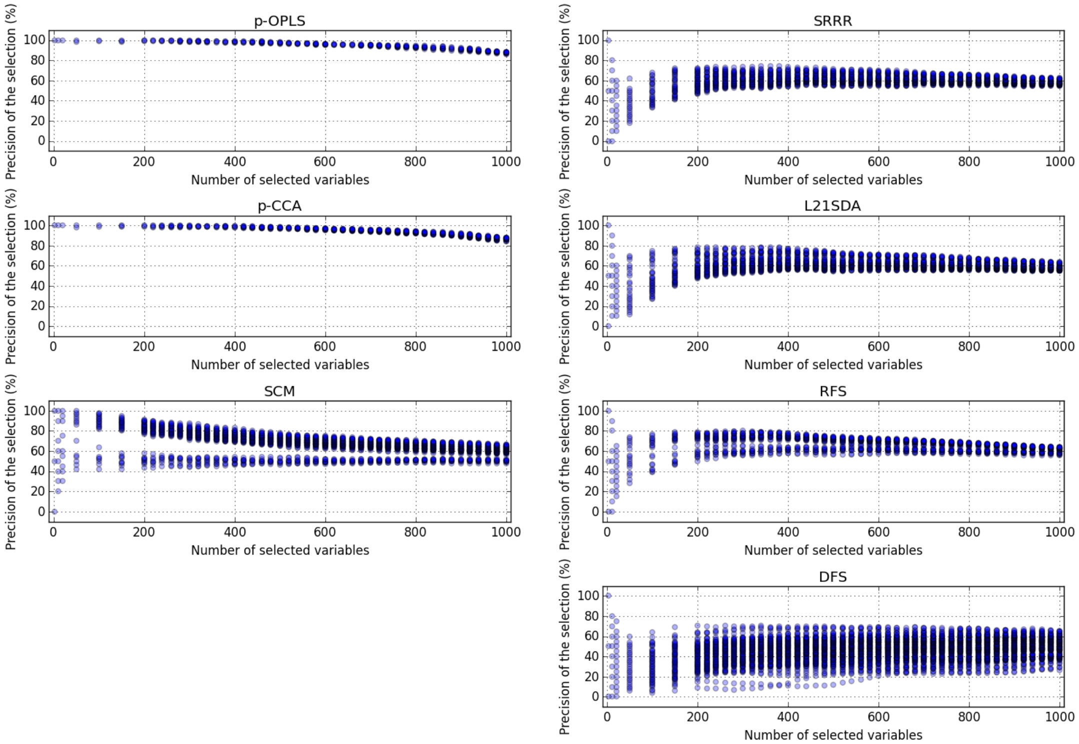


Figure 3: Stability analysis in a toy problem. Each point in the plots corresponds to one of the 948 data sets (each dataset constructed and analyzed with different parameters). The x-coordinate is the number of selected features, whilst the y-coordinate states the percentage of the features selected by the corresponding method that came from any of the actually informative groups.

5.4. Consistency of the parsimony pattern

The methods presented in this paper rely on the hypothesis that there exists an underlying parsimony pattern in the input variables. This section presents results showing that they indeed recover such pattern by analysing the consistency across the patterns retrieved in all the simulations. For this purpose we are going to focus on face recognition problems, since they enable to assess that the retrieved parsimony patterns are in fact meaningful for the interpretability of the learning (we will check that they are mainly formed by clusters of pixels around the eyes, nose, mouth and other visual salient features of the images).

Figure 4 shows the consistency in the selected input features across all the 50 simulations in problems Yale and ORL for different sparsity factors. Each pixel color is proportional to the number of simulations in which it was selected: red pixels were selected in all the simulations while dark blue pixels were never selected. The parsimony pattern is very stable across all the simulations, in fact, as the number of selected features increases, these methods include redundancies that help achieve a sharper definition of the parsimony pattern. On the contrary, the baseline feature selection methods

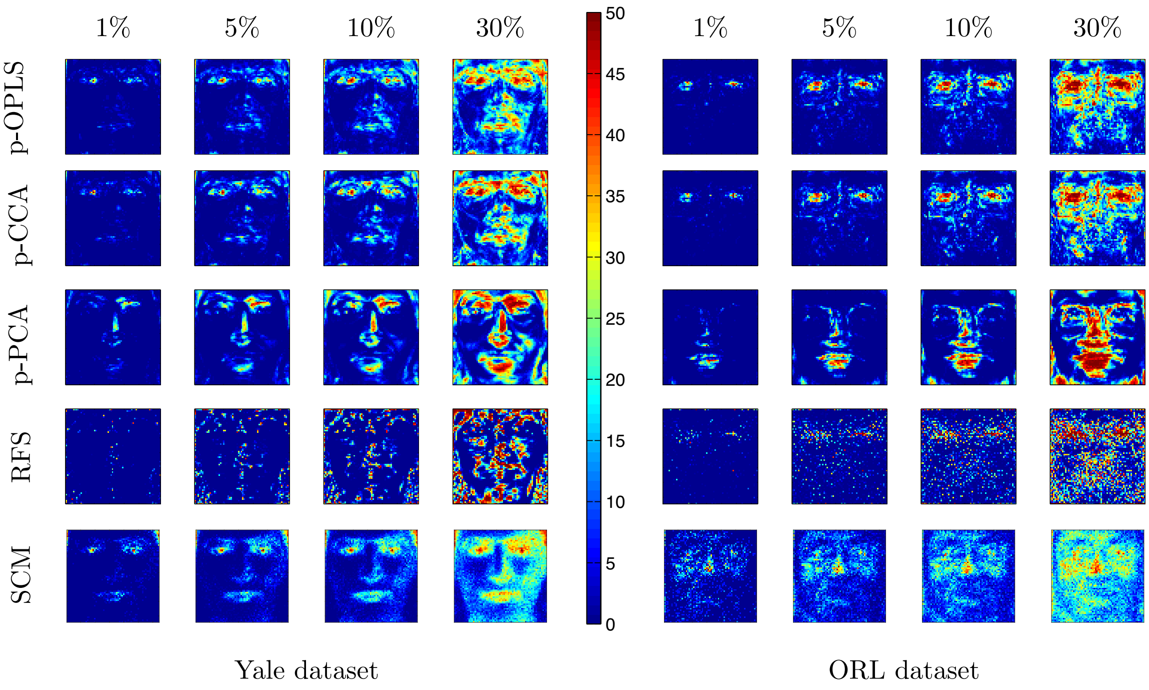


Figure 4: Feature selection masks for Yale and ORL datasets for different sparsity factors. Red color pixels are selected in all the runs; blue ones have never been selected.

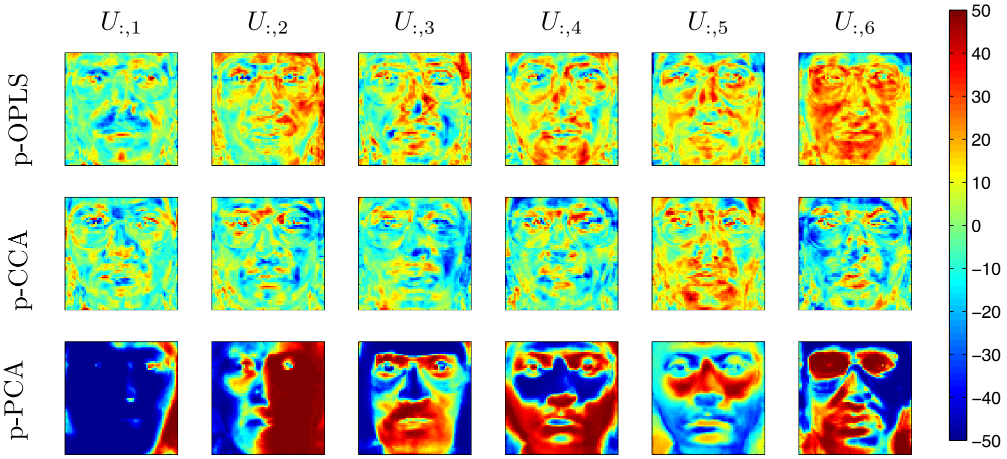


Figure 5: Sign consistency in the first 6 principal components of Yale. Red pixels are positive in all simulations, blue pixels are always negative. $U_{:,k}$ denotes the k-th column of U.

(RFS and SCM) present a less consistent selection. For instance, with SF=30%, the pattern of SCM presents many of the regions in the face in light-blue or green colours (i.e. these pixels were selected in about a half of the runs). In the case of the algorithm RFS the results look even worse. It yields a very scattered parsimony pattern that is difficult to interpret in terms of a face recognition task.

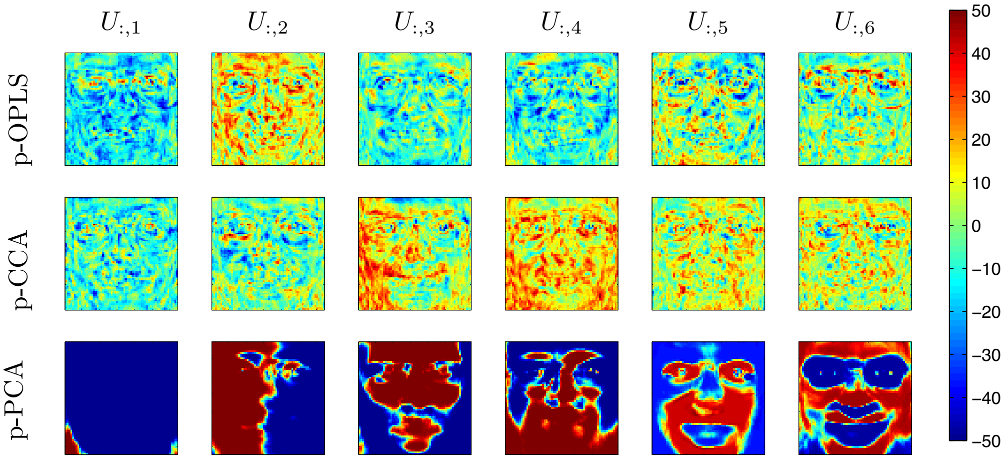


Figure 6: Sign consistency in the first 6 extracted features of ORL. Red pixels are positive in all simulations, blue pixels are always negative. $U_{:,k}$ denotes the k-th column of U.

The second hypothesis supporting the approach presented in this paper is that each input feature selected as member of the parsimony pattern presents a consistent behaviour in the definition of the extracted features that will be used in the final classification, and that this behaviour can be captured by a bagging of MVAs (see Section 3). Figures 5 and 6 show another point of view for the analysis of interpretability. Each mask corresponds to the sign consistency found for the first six extracted features when SF is set to 100% (each subplot is the corresponding column of U arranged in the same way as the input faces) in problems Yale and ORL. Red pixels correspond to elements $(U)_{kj}$ that ended up with a positive sign in all the simulations while blue pixels indicate that the corresponding $(U)_{kj}$ got a negative sign in all the simulations. The existence of highly consistent clusters of pixels with a same sign in all the features points out the existence of these latent meaningful features that conform the parsimony pattern.

Finally, to conclude this experimental analysis, Figure 7 shows the features extracted by the proposed parsimonious MVA methods, as well as baseline approaches L21SDA, SRRR and DFS, for one of the training/testing partitions of problems Yale and ORL. The recovered parsimony pattern comprises a 30% of the total input variables. It can also be noticed how the stability of the parsimony pattern in the proposed methods translates to the definition of the extracted features. These features are formed by clusters of selected input variables in areas relevant for the face recognition. Moreover, the coefficients in the same cluster receive weights of a similar value. However, the structure presented by the baseline methods is not so clear. This is specially remarkable in L21SDA and SRRR, since their coefficients are scattered across all the image, without a clear structure in terms of location and value of the weights.

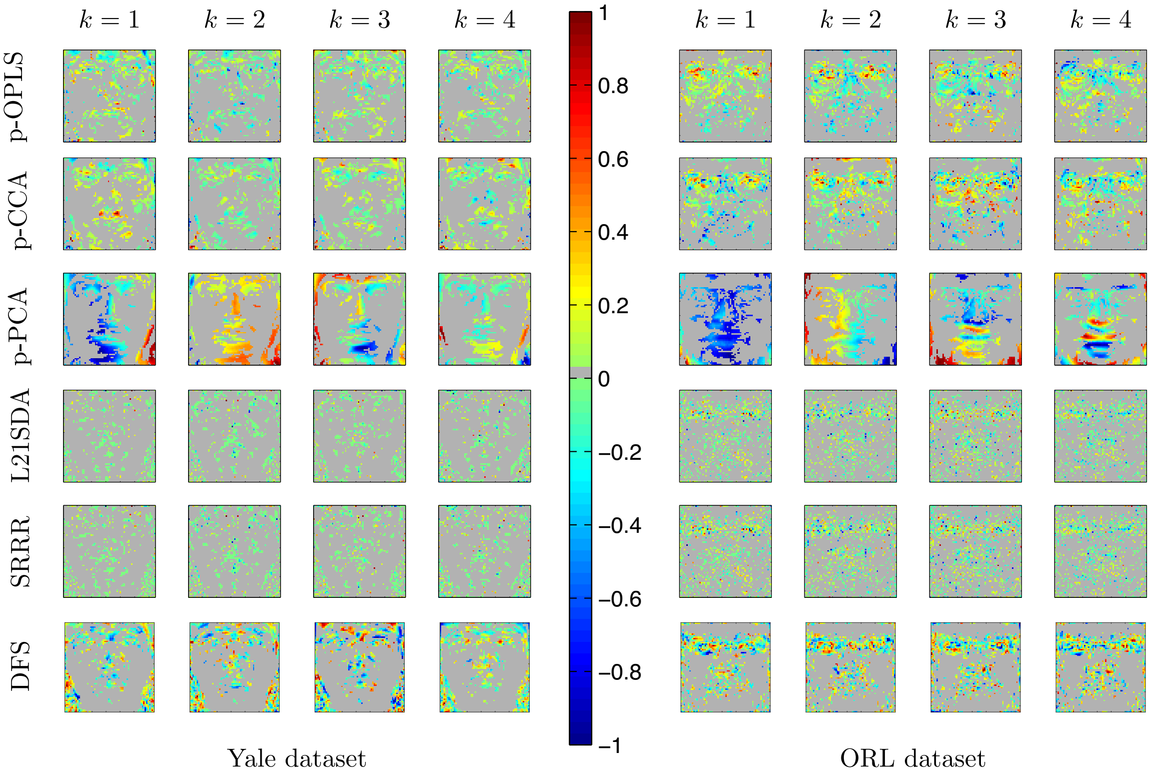


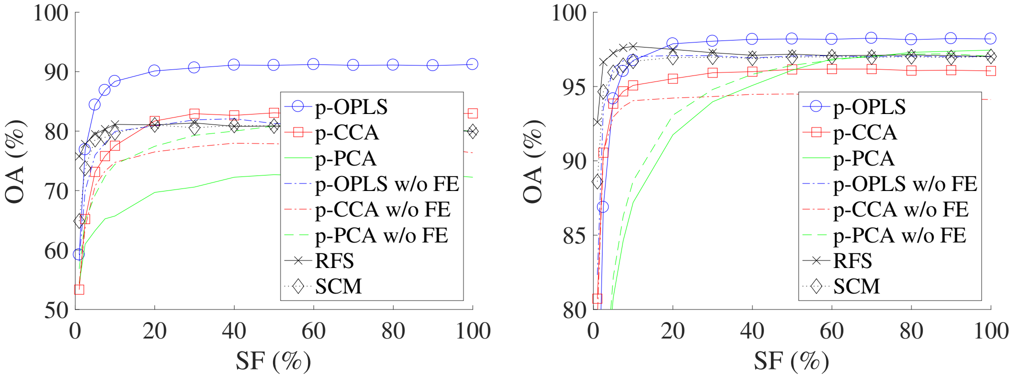
Figure 7: Values of the first 4 columns of U in one of the simulations for problems Yale and ORL. The columns have been arranged so that each coefficient U_{jk} appears in the position of the corresponding pixel $(\mathbf{x})_j$.

5.5. Analysis of the contributions of the feature selection and of the regularization

The remainder of the experimental work aims to give insights about the impact in the final results of the contributions introduced in Sections 3 (parsimonious feature selection) and 4 (regularization). Figure 8 shows the overall accuracy when the classifier is directly fed with the selected features, without the feature extraction stage (denoted as w/o FE in the figure). It is clear the poor performance provided when the selected features are used in a straight way and the accuracy improvement achieved due to the feature extraction process.

Finally, Figure 9 shows the accuracy obtained after performing the feature extraction with and without the regularization penalty introduced in Section 4.

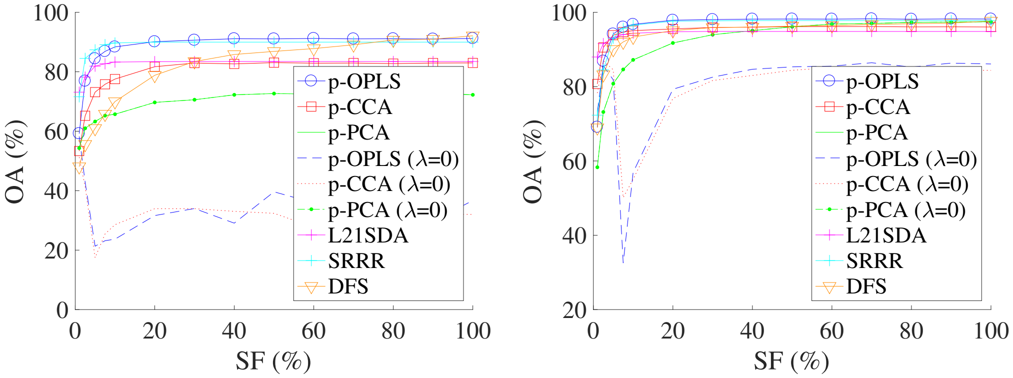
The discrimination capability of the features extracted without regularisation is very poor, not only in comparison with their standard versions, but also in comparison to L21SDA, SRRR and DFS. This is mainly caused by an overfitting effect which is alleviated by including the information of the relevance of each feature in the regularisation term. Notice that including this regularisation does not lead to an increment in the computational burden of the method, since this relevance is a collateral result of the bagging process. Besides, this regularisation endows the parsimonious MVA with robustness against



a) Yale

b) ORL

Figure 8: Overall accuracy (OA) vs. percentage of selected features (SF) for the different methods used as feature selectors.



a) Yale

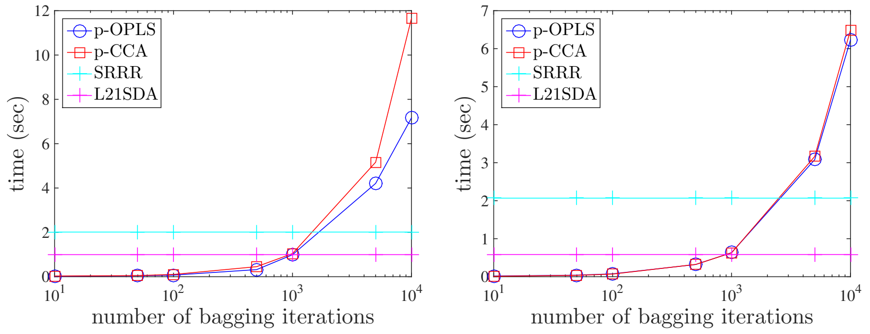
b) ORL

Figure 9: Overall accuracy (OA) vs. percentage of selected features (SF) for the different methods under study.

the number of selected features: once the maximum accuracy is achieved, this performance is kept 365 constant as the number of selected features in the parsimony pattern increases.

5.6. Computational complexity analysis

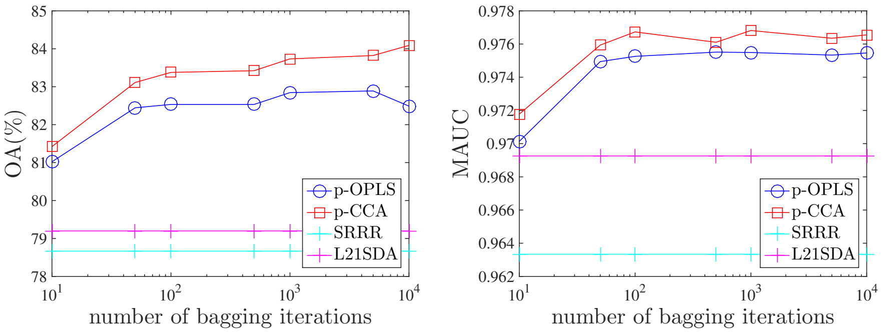
This last section completes the experimental analysis studying the computational cost required for the training of the proposed methods. As all the eigenvalue problems are solved in the dual space with a very reduced dimension (the size of the target vectors, c), the number of bagging iterations 370 becomes the dominant quantity impacting the discussion about the computational burden of the proposed algorithms. To be precise, an initial fixed cost of computing the $l \times l$ kernel matrices has to be added to the quantities presented in the remainder of the section. Figure 10 shows the evolution of the computation times with the number of bagging iterations for two problems: a) the synthetic one described in Subsection 5.3 and b) the Yale problem presented in Table 1. For this study, we



a) Synthetic problem

b) Yale

Figure 10: Computational time comparison.



a) OA

b) MAUC

Figure 11: Performance evolution with the number of bagging iteration for Yale dataset.

have only considered as reference approaches the most efficient methods, SRRR and L21SDA; DFS solves a generalized eigenvalue problem, i.e., two $\mathcal{O}(d^3)$ operations, requiring about 50 times more than our proposal with 1000 bagging iterations. All the simulations have been carried out on an ordinary MacBook Pro laptop.

As expected, Figure 10 shows that the computational cost of p-OPLS and p-CAA grows linearly with the number of iterations. It is remarkable that the computation time of the proposed methods compares favourably with the baseline methods up to 1000 bagging iterations.

Figure 11 shows the dependence of the method performance (both in terms of OA and MAUC) with the number of bagging iterations. Notice that using around 1000 iterations does not cause significant performance degradations. In fact, the number of bagging iterations could be reduced down to 100, without serious drops in accuracy (it is the same in terms of MAUC and it is slightly reduced in terms of OA).

6. Concluding remarks

This paper has introduced a framework for the implementation of parsimonious MVA, specially suited for very high dimensional problems. This novel framework develops the concept of feature filtering: to capture relevant and redundant variables and to discard the noisy ones. The core of the methodology is a bagging of MVAs. On the one hand, the bagging drives a feature selection based on the consistency of input variables in the definition of the features extracted by the MVAs. This feature selection recovers an underlying parsimony pattern formed by some of the input variables. On the other hand, the bagging allows to assess the relevance of each input variable for the solution of the problem. This relevance is used in a regularisation term that endows a final feature extraction with a strong robustness. Furthermore, the formulation developed to derive the MVAs is tailored to the bagging procedure, limiting its computational burden.

The experimental results point out that the proposed approach is robustness against to their parameter selection, is computationally efficient and the extracted features presents excellent discriminatory capabilities. Moreover, the retrieved parsimony patterns in face recognition problems are very stable across simulations. A further analysis of these patterns shows that the selected input variables appear in clusters with similar sign consistency. These clusters are located in areas that are relevant for the face recognition, what leads us to consider them as potential seeds for the definition of semantic features that help interpret the results of the learning procedure and gain insight about the problem at hand.

Acknowledgements

This work has been partly funded by the Spanish MINECO grant TEC2014-52289R and TEC2013-48439-C4-1-R. The authors want to thank the action editor and the reviewers for their valuable feedback.

Appendix A. Proof of equivalence between Equations (2) and (3)

This Appendix shows the equivalence between equations (2) and (3). The overall derivation comprises two main steps. The first step involves obtaining solutions for problems (2) and (3) independently; the second step shows that both solutions are equivalent.

To obtain a solution for (2), first fixing U in (2) and solving for W yields $W = C_{XY}^T U(U^T C_{XX} U)^{-1}$. This result is substituted back in (2) to form a new functional in which the constraints are introduced using Lagrange Multipliers:

$$\mathcal{L}(U, \Lambda) = || \left(Y - XU \left(U^T X^T X U\right)^{-1} U^T X^T Y\right) \Gamma^{\frac{1}{2}} ||_F^2 - \text{Tr}\{XU\Lambda U^T X^T\} + \text{Tr}\{\Lambda\}$$

where Λ is a square matrix of size r in which each element Λ_{ij} is the Lagrange Multiplier that corresponds with each constraint $U_i^T X^T X U_j = I_{ij}$.

The optimization of this functional can be carried out following a standard OPLS derivation that ends up in a generalized eigenvalue decomposition problem (GEV):

$$C_{XY}\Gamma C_{XY}^T U_{\text{GEV}} = C_{XX} U_{\text{GEV}} \Lambda_{\text{GEV}}, \tag{A.1}$$

where Λ_{GEV} is the diagonal matrix[3] containing the r largest generalized eigenvalues arranged in decreasing order, U_{GEV} is a matrix whose columns are the corresponding r leading generalized eigenvectors. Matrix U_{GEV} is a solution to (2) (in fact any matrix $U_R = U_{\text{GEV}} R$, where R is a rotation matrix, is also a solution of (2)). Moreover, U_{GEV} is C_{XX}-orthonormal (e.i., $U_{\text{GEV}}^T C_{XX} U_{\text{GEV}} = I$) due to the constraint of (2).

Now U_{GEV} is used to solve (2) for W:

$$W_{\text{GEV}} = C_{XY}^T U_{\text{GEV}}. \tag{A.2}$$

Premultiplying (A.1) by U_{GEV}^T and inserting (A.2) yields $W_{\text{GEV}}^T \Gamma W_{\text{GEV}} = \Lambda_{\text{GEV}}$, which demonstrates the orthogonality condition of W.

Besides, a pair of matrices $(U_{\text{EVD}}^T, W_{\text{EVD}}^T)$ that form a solution for (3) can be reached following a similar procedure to the one described above. Fixing W in (3) and solving for U yields

$$U_{\text{EVD}} = C_{XX}^{-1} C_{XY} \Gamma W (W^T \Gamma W)^{-1}. \tag{A.3}$$

With this expression replaced back in (3) and the constraints introduced using Lagrange Multipliers contained in matrix Λ, some algebraic manipulations lead to the new functional

$$\mathcal{L}(W, \Lambda) = \left|\left| \left(Y - X C_{XX}^{-1} C_{XY} \Gamma W (W^T \Gamma W)^{-1} W^T\right) \Gamma^{\frac{1}{2}} \right|\right|_F^2 - \text{Tr}\{W^T \Gamma W \Lambda\} + \text{Tr}\{\Lambda\}.$$

The solution to this optimization can be carried out by solving the following eigenvalue decomposition problem:

$$\Gamma C_{XY}^T C_{XX}^{-1} C_{XY} \Gamma W_{\text{EVD}} = \Gamma W_{\text{EVD}} \Lambda_{\text{EVD}}, \tag{A.4}$$

where Λ_{EVD} is a diagonal matrix with the r largest eigenvalues arranged in decreasing order.

The U_{EVD} that corresponds to W_{EVD} can be obtained from (A.3)

$$U_{\text{EVD}} = C_{XX}^{-1} C_{XY} \Gamma W_{\text{EVD}}. \tag{A.5}$$

[3] Notice that in the optimum the constraints that correspond with the non-diagonal elements of Λ ($U_i^T X^T X U_j = 0$) have a null contribution in the functional.

Moreover, premultiplying both terms of (A.4) by W_{EVD}^T and inserting (A.5) yields

$$U_{\text{EVD}}^T C_{XY} \Gamma W_{\text{EVD}} = \Lambda_{\text{EVD}},$$

since $W_{\text{EVD}} \Gamma W_{\text{EVD}} = I$ (constraint of (3)). Noticing also that according to (A.5), $C_{XY} \Gamma W_{\text{EVD}} = C_{XX} U_{\text{EVD}}$:

$$U_{\text{EVD}}^T C_{XX} U_{\text{EVD}} = \Lambda_{\text{EVD}}, \tag{A.6}$$

which demonstrates the orthogonality condition of the projected input data.

Since the columns of U_{EVD} and U_{GEV} span the same subspace of $\mathbb{R}^d$, they should verify $U_{\text{EVD}} = U_{\text{GEV}} A$, for some square and invertible matrix A of size r. Inserting this expression in (A.6), and realizing that the columns in U_{GEV} are C_{XX}-orthonormal (e.i., $U_{\text{GEV}}^T C_{XX} U_{\text{GEV}} = I$) yields

$$A^T U_{\text{GEV}}^T C_{XX} U_{\text{GEV}} A = A^T A = \Lambda_{\text{EVD}}.$$

Since Λ_{EVD} admits a Cholesky factorization, and this is unique, it can be written necessarily $A = A^T = \Lambda_{\text{EVD}}^{1/2}$ and

$$U_{\text{EVD}} = U_{\text{GEV}} \Lambda_{\text{EVD}}^{1/2}. \tag{A.7}$$

The next step shows the relationship between the regression coefficient matrices. Inserting (A.5) into (A.4) yields $C_{XY}^T U_{\text{EVD}} = W_{\text{EVD}} \Lambda_{\text{EVD}}$. Also, the use of (A.7) together with (A.2) shows that $W_{\text{GEV}} \Lambda_{\text{EVD}}^{1/2} = C_{XY}^T U_{\text{EVD}}$. Combining these two last equations, it is straightforward to arrive at

$$W_{\text{EVD}} = W_{\text{GEV}} \Lambda_{\text{EVD}}^{-1/2}. \tag{A.8}$$

The conclusion of the proof passes through showing that $\Lambda_{\text{EVD}} = \Lambda_{\text{GEV}} = \Lambda$, for which it is enough to use (A.8) together with condition $W_{\text{GEV}}^T W_{\text{GEV}} = \Lambda_{\text{GEV}}$. Resourcing also to the orthonormality condition of the columns of W_{EVD}:

$$W_{\text{GEV}}^T W_{\text{GEV}} = \Lambda_{\text{EVD}}^{1/2} W_{\text{EVD}}^T W_{\text{EVD}} \Lambda_{\text{EVD}}^{1/2} = \Lambda_{\text{EVD}} = \Lambda_{\text{GEV}}. \tag{A.9}$$

To summarize, the following relationships between (2) and (3) hold:

$$\Lambda_{\text{EVD}} = \Lambda_{\text{GEV}} (= \Lambda), \; U_{\text{EVD}} = U_{\text{GEV}} \Lambda^{1/2}, \; W_{\text{EVD}} = W_{\text{GEV}} \Lambda^{-1/2}.$$

Since Λ is diagonal, the columns of U_{GEV} and U_{EVD} must have the same direction, and differ only by a scaling factor. This finally concludes that Equations (2) and (3) are equivalent.

A similar equivalence demostration for the particular case $\Gamma = I$ is described in [16].

[1] K. Pearson, On lines and planes of closest fit to systems of points in space, Philosophical Magazine 2 (6) (1901) 559–572.

[2] H. Hotelling, Relations between two sets of variates, Biometrika 28 (1936) 321–377.

[3] H. Wold, Non-linear estimation by iterative least squares procedures, in: Research Papers in Statistics, Wiley, 1966, pp. 411–444.

[4] H. Wold, Estimation of principal components and related models by iterative least squares, in: Multivariate Analysis, Academic Press, 1966, pp. 391–420.

[5] K. Worsley, J. Poline, K. Friston, A. Evans, Characterizing the response of pet and fMRI data using multivariate linear models (MLM), Neuroimage 6 (1998) 305–319.

[6] G. C. Reinsel, R. P. Velu, Multivariate reduced-rank regression: Theory and applications, Springer New York, 1998.

[7] Z. Zhang, Y. Xu, J. Yang, X. Li, D. Zhang, A survey of sparse representation: Algorithms and applications, IEEE Access 3 (2015) 490–530.

[8] D. Bertsimas, A. King, R. Mazumder, Best subset selection via a modern optimization lens, The Annals of Statistics 44 (2) (2016) 813–852.

[9] K. P. Murphy, Machine Learning: A Probabilistic Perspective, The MIT Press, 2012, Ch. 13: Sparse linear models, pp. 421–478.

[10] T. Park, G. Casella, The bayesian lasso, Journal of the American Statistical Association 103 (482) (2008) 681–686.

[11] A. Armagan, D. Dunson, J. Lee, Generalized double pareto shrinkage, Statistica Sinica 1 (23) (2013) 119–143.

[12] H. Zou, T. Hastie, R. Tibshirani, Sparse principal component analysis, Journal of computational and graphical statistics 15 (2) (2006) 265–286.

[13] I. M. Johnstone, A. Y. Lu, On consistency and sparsity for principal components analysis in high dimensions, Journal of the American Statistical Association 104 (486) (2009) 682–693.

[14] D. M. Witten, R. Tibshirani, T. Hastie, A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis, Biostatistics (2009) 515–534.

[15] D. R. Hardoon, J. Shawe-Taylor, Sparse canonical correlation analysis, Machine Learning 83 (3) (2011) 331–353.

[16] S. Muñoz-Romero, J. Arenas-García, V. Gómez-Verdejo, Sparse and kernel OPLS feature extraction based on eigenvalue problem solving, Pattern Recognition 48 (5) (2015) 1797 – 1811.

[17] C. Archambeau, F. R. Bach, Sparse probabilistic projections, in: Advances in Neural Information Processing Systems 21, Curran Associates, Inc., 2009, pp. 73–80.

[18] Y. Guan, J. G. Dy, Sparse probabilistic principal component analysis., in: AISTATS, 2009, pp. 185–192.

[19] A. Klami, S. Virtanen, S. Kaski, Bayesian canonical correlation analysis, Journal of Machine Learning Research 14 (Apr) (2013) 965–1003.

[20] F. Nie, S. Xiang, Y. Jia, C. Zhang, S. Yan, Trace ratio criterion for feature selection, in: Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 2, AAAI'08, AAAI Press, 2008, pp. 671–676.

[21] P. Zhao, G. Rocha, B. Yu, The composite absolute penalties family for grouped and hierarchical variable selection, The Annals of Statistics (2009) 3468–3497.

[22] J. Friedman, T. Hastie, R. Tibshirani, A note on the group lasso and a sparse group lasso, arXiv preprint arXiv:1001.0736.

[23] L. F. S. Merchante, Y. Grandvalet, G. Govaert, An efficient approach to sparse linear discriminant analysis, in: Proc. 29th International Conference on Machine Learning (ICML-12), Omnipress, New York, NY, USA, 2012, pp. 1167–1174.

[24] F. Nie, H. Huang, X. Cai, C. H. Ding, Efficient and robust feature selection via joint $\ell_{2,1}$-norms minimization, in: Advances in Neural Information Processing Systems 23, Curran Associates, Inc., 2010, pp. 1813–1821.

[25] S. Xiaoshuang, L. Zhihui, G. Zhenhua, W. Minghua, Z. Cairong, K. Heng, Sparse principal component analysis via joint l 2, 1-norm penalty, AI (2013) 148–159.

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

[27] D. Wang, F. Nie, H. Huang, Unsupervised feature selection via unified trace ratio formulation and k-means clustering (track), in: Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2014, Nancy, France, September 15-19, 2014. Proceedings, Part III, Springer Berlin Heidelberg, Berlin, Heidelberg, 2014, pp. 306–321.

[28] B. Liu, B. Fang, X. Liu, J. Chen, Z. Huang, X. He, Large margin subspace learning for feature selection, Pattern Recognition 46 (10) (2013) 2798 – 2806.

[29] Z. Li, J. Liu, J. Tang, H. Lu, Robust structured subspace learning for data representation, IEEE Transactions on Pattern Analysis and Machine Intelligence 37 (10) (2015) 2085–2098.

[30] X. Shi, Y. Yang, Z. Guo, Z. Lai, Face recognition by sparse discriminant analysis via joint $l_{2,1}$-norm minimization, Pattern Recognition 47 (7) (2014) 2447–2453.

[31] L. Chen, J. Z. Huang, Sparse reduced-rank regression for simultaneous dimension reduction and variable selection, Journal of the American Statistical Association 107 (500) (2012) 1533–1545.

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

[33] H. Tao, C. Hou, F. Nie, Y. Jiao, D. Yi, Effective discriminative feature selection with nontrivial solution, IEEE Transactions on Neural Networks and Learning Systems 27 (4) (2016) 796–808.

[34] S. Muñoz-Romero, V. Gómez-Verdejo, J. Arenas-Garcia, Regularized multivariate analysis framework for interpretable high-dimensional variable selection, IEEE Computational Intelligence Magazine 11 (4) (2016) 24–35.

[35] T. Abeel, T. Helleputte, Y. Van de Peer, P. Dupont, Y. Saeys, Robust biomarker identification for cancer diagnosis with ensemble feature selection methods, Bioinformatics 26 (3) (2010) 392–398.

[36] B. Jin, A. Strasburger, S. J. Laken, F. A. Kozel, K. A. Johnson, M. S. George, X. Lu, Feature selection for fMRI-based deception detection, BMC Bioinformatics 10 (9) (2009) 1–7.

[37] P. Somol, J. Grim, J. Novovičová, P. Pudil, Improving feature selection process resistance to failures caused by curse-of-dimensionality effects, Kybernetika 47 (3) (2011) 401–425.

[38] X. Wang, X. Tang, Random sampling for subspace face recognition, International Journal of Computer Vision 70 (1) (2006) 91–104.

[39] H. Wang, T. Khoshgoftaar, A. Napolitano, A comparative study of ensemble feature selection techniques for software defect prediction, in: Proc. 9th International Conference on Machine Learning and Applications (ICMLA), 2010, pp. 135–140.

[40] D. Dernoncourt, B. Hanczar, J.-D. Zucker, Analysis of feature selection stability on high dimension and small sample data, Computational Statistics & Data Analysis 71 (2014) 681 – 693.

[41] L. I. Kuncheva, Combining Pattern Classifiers: Methods and algorithms, 2nd Edition, John Wiley & Sons, Inc., 2014, Ch. 9 Ensemble Feature Selection, pp. 290–325.

[42] N. Meinshausen, P. Buehlmann, Stability selection, arXiv:0809.2932.

[43] J. Bi, K. Bennett, M. Embrechts, C. Breneman, M. Song, Dimensionality reduction via sparse support vector machines, JMLR 3 (2003) 1229–1243.

[44] E. Parrado-Hernández, V. Gómez-Verdejo, M. Martínez-Ramón, J. Shawe-Taylor, P. Alonso, J. Pujol, J. M. Menchón, N. Cardoner, C. Soriano-Mas, Discovering brain regions relevant to obsessive–compulsive disorder identification through bagging and transduction, Medical image analysis 18 (3) (2014) 435–448.

[45] F. D. la Torre, A least-squares framework for component analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence 34 (6) (2012) 1041–1055.

[46] J. Arenas-Garcia, K. B. Petersen, G. Camps-Valls, L. K. Hansen, Kernel multivariate analysis framework for supervised subspace learning: A tutorial on linear and kernel multivariate methods, IEEE Signal Processing Magazine 30 (4) (2013) 16–29.

[47] D. J. Hsu, S. M. Kakade, J. Langford, T. Zhang, Multi-label prediction via compressed sensing, in: Advances in Neural Information Processing Systems 22, Curran Associates, Inc., 2009, pp. 772–780.

[48] L. Breiman, Random forests, Machine learning 45 (1) (2001) 5–32.

[49] D. Cai, X. He, J. Han, H.-J. Zhang, Orthogonal laplacianfaces for face recognition, IEEE Transactions on Image Processing 15 (11) (2006) 3608–3614.

[50] G. Lan, C. Hou, D. Yi, Robust feature selection via simultaneous capped ℓ_2-norm and $\ell_{2,1}$-norm minimization, in: 2016 IEEE International Conference on Big Data Analysis (ICBDA), 2016, pp. 1–5.

[51] K. Tang, R. Wang, T. Chen, Towards maximizing the area under the ROC curve for multi-class classification problems, in: Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI Press, 2011, pp. 483–488.