Infinite Feature Selection: A Graph-based Feature Filtering Approach

Giorgio Roffo, Simone Melzi, *Member, IEEE*, Umberto Castellani,
Alessandro Vinciarelli, *Member, IEEE*, and Marco Cristani, *Member, IEEE*

Abstract—We propose a filtering feature selection framework that considers subsets of features as paths in a graph, where a node is a feature and an edge indicates pairwise (customizable) relations among features, dealing with relevance and redundancy principles. By two different interpretations (exploiting properties of power series of matrices and relying on Markov chains fundamentals) we can evaluate the values of paths (i.e., feature subsets) of arbitrary lengths, eventually go to infinite, from which we dub our framework *Infinite Feature Selection* (Inf-FS). Going to infinite allows to constrain the computational complexity of the selection process, and to rank the features in an elegant way, that is, considering the value of any path (subset) containing a particular feature. We also propose a simple unsupervised strategy to cut the ranking, so providing the subset of features to keep. In the experiments, we analyze diverse settings with heterogeneous features, for a total of 11 benchmarks, comparing against 18 widely-known comparative approaches. The results show that Inf-FS behaves better in almost any situation, that is, when the number of features to keep are fixed a priori, or when the decision of the subset cardinality is part of the process.

Index Terms—Feature selection, filter methods, markov chains

1 Introduction

Over the last few decades, successful approaches to machine learning problems have been based initially on hand-crafted features (e.g., SIFT and HOG-like [1], [2], [3], [4], dictionary-based [5]) that evolved into automatically learned ones with the diffusion of deep learning models [6], [7], [8]. Through these advancements, feature selection (FS) still remains an active and growing research area that enables both dimensionality reduction and data interpretability, looking for features which are relevant and not redundant [9], [10], [11].

In this paper we introduce a fast graph-based feature filtering approach that ranks and selects features by considering the possible subsets of features as paths on a graph, and works in an unsupervised or supervised setup.

Our framework is composed by three main steps. In the first step, an undirected fully-connected weighted graph is built, where the node $\vec{v}_i$, $1 \leq i \leq n$, corresponds to the feature f_i, and each edge connecting $\vec{v}_i$ to $\vec{v}_j$ has associated a weight, or value, modeling the expectation that features f_i and f_j are relevant and not redundant. The weight comes from customizable pairwise relations among feature distributions, which can be easily crafted by the user, and, as a future perspective, learned directly from data. Here we present two instances of pairwise relations: one exploiting class information (Inf-FS$_S$), the other one being completely agnostic (Inf-FS$_U$).

In the second step, the weighted adjacency matrix associated to the graph is employed to assess the value of each feature (i.e., a node in the graph) while considering possible subsets of features (i.e., subsets of nodes) as they were paths of variable length. Two interpretations can be exploited: one comes from the properties of power series of matrices, the other one from the concept of absorbing Markov chain. In both the cases, we compute a vector which at the ith entry expresses the value (or probability) of having a particular feature in a subset of any length, summing for all the possible lengths, until infinite. Going to infinite allows us to reduce the computational complexity from $\mathcal{O}(n^3 lT)$ (n features, l path length, T samples) to $\mathcal{O}(n^3 T)$. For this reason, we dubbed our approach *Infinite Feature selection (Inf-FS)*. Ranking the values of the "infinite" vector gives the ordered importance of the features. In the third step, a threshold over the ranking is automatically selected by clustering over the ranked value. The rationale is to individuate at least two distributions, one which contains the features to keep with higher value, the other the ones to discard.

The proposed framework is compared against 18 comparative approaches of feature selection, with the goal of feeding the selected features into an SVM classifier.

As for the datasets, we selected 11 publicly available benchmarks to deal with diverse FS scenarios and challenges. In particular we consider five DNA microarray datasets for cancer classification (*Colon* [12], *Lymphoma* [13], *Leukemia* [13],

- *G. Roffo and A. Vinciarelli are with the School of Computing Science, University of Glasgow, G12 8QQ Glasgow, U.K. E-mail: {Giorgio.Roffo, Alessandro.Vinciarelli}@glasgow.ac.uk.*
- *M. Cristani is with the Department of Computer Science, University of Verona, 37129 Verona, Italy, and also with the Institute of Cognitive Sciences and Technologies (ISTC), 00185 Roma, Italy. E-mail: marco.cristani@univr.it.*
- *S. Melzi and U. Castellani are with the Department of Computer Science, University of Verona, 37129 Verona, Italy. E-mail: {simone.melzi, umberto.castellani}@univr.it.*

Manuscript received 5 Aug. 2019; revised 29 Feb. 2020; accepted 1 June 2020. Date of publication 16 June 2020; date of current version 3 Nov. 2021.
(Corresponding author: Giorgio Roffo.)
Recommended for acceptance by R. Feris.
Digital Object Identifier no. 10.1109/TPAMI.2020.3002843

Lung [14], *Prostate* [15]), handwritten character recognition (GINA [16]), general classification tasks from the NIPS feature selection challenge (MADELON, GISETTE [17], DEXTER [18]), and two classic object recognition datasets with convolutional neural networks (CNNs) features (PASCAL VOC 2007 [19] and CalTech 101 [20]).

One of the most interesting aspects shown in the experiments is the flexibility of Inf-FS, both in its unsupervised and supervised version: independently on the scenario (small-sample+high dimensional, unbalanced classes, severe inter-class overlap, noise) Inf-FS overcomes the competitors, and if not, it gives the second or third best performance, promoting itself as all-purpose feature selection strategy. Another important achievement is in the automatic thresholding, which is simple yet effective in deciding which features to keep on any dataset. Finally, Inf-FS operates also on neural features, acting over cues that have been the state-of-the-art for long time [21].

The proposed framework generalizes the previously published *Infinite Feature Selection* (Inf-FS) [22], [23] presented as an unsupervised filtering approach, explained by algebraic motivations. Here we introduce a supervised counterpart and a strategy to select a subset of features, supported by a novel alternative way to explain the Inf-FS thanks to Markov chains fundamentals.

The rest of the paper is organized as follows: Section 2 illustrates the related literature, including the comparative approaches we consider in this study. Section 3 introduces our approach showing how the fully-connected graph is built for both the unsupervised and supervised variants. Section 3.5 connects the proposed approach to the absorbing Markov chain framework, deriving the subset selection strategy. Extensive experiments are reported in Section 4, and, finally, in Section 5, conclusions are given and future perspectives are envisaged.

2 State Of The Art

Feature selection algorithms are partitioned into three main classes [24], [25]: *filters*, *wrappers* and *embedded* methods. *Filter* methods make use of the intrinsic properties of the data (e.g., correlation, variance, locality, information gain, etc.) to evaluate the value of a feature. In contrast, *wrapper* methods assign an importance score to each feature based on the performance of a predictor, which is considered as a black box. Finally, *embedded* methods include the feature selection process as part of an internal regression model which estimates the relationships among variables. Inf-FS belongs to the filter approaches, since it deals with the sole properties of the data, without relying on a specific predictor.

Within each of the above families of algorithms, FS techniques can be further classified into two sub-categories, *unsupervised* and *supervised*, depending on the use of class-label information in the selection process. In this paper we present an example of Inf-FS for both the scenarios.

Most of the feature selection algorithms evaluates an initial feature set, providing a ranking on them as a final output. Subsequently, the ranking is cut by *subset selection* strategies, commonly performed by cross-validation strategies on validation data [25].

The section overviews the three families of FS methods, separating their unsupervised and supervised versions.

2.1 Filter Methods

2.1.1 Unsupervised Approaches

In unsupervised scenarios, filter methods are mainly based on locality preserving principia found by clustering. The Laplacian Score (LS) [26] evaluates the value of a feature as its tendency to preserve spatial relationships which ensure intra-cluster proximity. Technically, LS constructs a nearest neighbor graph and ranks high those features that are consistent with Gaussian Laplacian matrix [26]. Similarly, in the multi-cluster feature selection approach (MCFS) [27], features are selected based on spectral analysis and solving a sparse regression problem, encouraging the formation of compact clusters. Local learning clustering (LLCFS) method [9] is a kernel learning method that weights features and exploits the weights to regularize the clustering. Uninformative features are left out before the clustering. These solutions, included in the experiments, are computationally expensive since rely on clustering. In contrast, our approach is faster since it only uses intrinsic properties of the data.

2.1.2 Supervised Approaches

A standard two-class filter method is *Relief* and its multi-class extension *Relief-F* [28]. In general, the strategy evaluates feature value differences between nearest neighbor pairs and scores features according to how well they contribute to the overall class separation. A common criticism of Relief is that it selects redundant subsets, since it is not controlling feature correlation. A solution is given by the minimum Redundancy and Maximum Relevancy (mRMR) algorithm [29], minimizing the redundancy and maximizing the relevance of the set of features. This is obtained by maximizing the joint mutual information (using Parzen Gaussian windows [30]) between the values of a given feature and the membership to a particular class. The mRMR suffers from an expensive computational cost (i.e., $\mathcal{O}(n^2T^3)$ where n is the number of features and T the number of samples [25], [29]). Another weakness of mRMR comes with the approximation of the mutual information, which is inaccurate when the number of training samples is small [30]. A faster filter approach is the Fisher score [31], which scores the features individually, according to the ratio of inter-class separation and intra-class variance.

Several algorithms employ mutual information to select the features. The simple method proposed in [32] estimates the mutual information between feature distributions and class labels. All the features are evaluated independently, one by one, obtaining a score used to do ranking. The recent Max-Relevance and Max-Independence (MRI) [33] introduces a relevancy additional constraint, by maximizing the classification accuracy while minimizing the redundancy between features. Other approaches such as CIFE [34], MIFS [35] and ICAP [36] quantify the redundancy (or dependency) among the set of feature distributions by proposing slightly different variations of the objective function, i.e., the conditional likelihood of the training labels. Similarly, the joint mutual information (JMI) [37] and conditional mutual information (CMIM) [38] may be included in this group. The common assumption behind all these methods is that independency among features can positively affect the classification performance.

The Inf-FS framework is attractive because, when computing the weighted adjacency matrix, allows to include inter/intra class reasoning without relying on a specific strategy: in fact, the supervised Inf-FS$_S$ proposed here makes use of a fast computation of the mutual information and the Fisher criterion, but other alternatives are possible. Especially in the case of large number of samples, mutual information may be dropped in favor of other relations faster to be computed. Another difference with Inf-FS is that the MI-based approaches take into account pairwise (feature-class label) dependencies, while our approach extends the 2-nd order to nth order by considering subsets of features as paths on a graph.

Recently, other graph-based approaches have been proposed such as the eigenvector centrality (ECFS) [39], [40] and the infinite latent feature selection (ILFS) [22], which is an extension of the unsupervised Inf-FS$_U$. The ECFS ranks features according to a centrality measure over the graph of features (eigenvector centrality), and should be considered a lighter version of Inf-FS$_U$, see Section 3 for further details. In ILFS, the features are grouped into token by probabilistic latent semantic analysis (PLSA), which in practice learns the weights of the adjacency graph of Inf-FS as to provide better class separability. Instead, our framework requires to explicitly craft the weights; despite the experiments show that our approach overcomes ILFS, we think learning the weights is a convenient direction, which we are interested at the present moment.

Summarizing, some advantages of using filter methods are:

- faster than wrapper and embedded methods;
- scalable;
- classifier independent (better generalization).

On the other hand, disadvantages are related to a generic lower performance if compared to supervised approaches, since filters are independent on the specific classifier.

2.2 Wrapper Approaches

2.2.1 Unsupervised Approaches

In the dependence-guided unsupervised feature selection (DGUFS) [41], feature selection is performed by graph-based clustering through the optimization of two terms: one term individuates dependence among samples by clustering, while the other term votes for features which minimize the intra-cluster variance. This approach is showed to be prone to local minima. The feature selection with adaptive structure learning (FSASL) [42] is an iterative approach that captures the global structure of data within a sparse representation framework, where the reconstruction coefficient is learned from the selected features. Its main drawback is the high computational complexity (see Table 1). Finally, the unsupervised feature selection with ordinal locality (UFSOL) [43] is a clustering-based approach that preserves the relative neighborhood proximities of the samples through distance-based clustering.

Similarly to our approach, these last three methods estimate inter-relationships among features, but in an iterative fashion which makes the entire process expensive and prone to local minima. Conversely, Inf-FS is one-shot.

TABLE 1
Feature Selection Approaches Considered in the Experiments of Section 4

Acronym	Type	Class	Comp. complexity
LLCFS [9]	f	u	N/A
LS [26]	f	u	N/A
MCFS [27]	f	u	N/A
Relief-F [28]	f	s	$\mathcal{O}(iTnG)$
MI [32]	f	s	$\mathcal{O}(T^2n^2)$
Fisher [31]	f	s	$\mathcal{O}(Tn)$
ECFS [39], [40]	f	s	$\mathcal{O}(Tn+n^2)$
ILFS [22]	f	s	$\mathcal{O}(n^{2.37}+in+T+G)$
CFS [44]	f	u	$\mathcal{O}(\frac{n^2}{2}T)$
UDFS [48]	f	u	N/A
DGUFS [41]	w	u	N/A
FSASL [42]	w	u	$\mathcal{O}(n^3+Tn^2)$
UFSOL [43]	w	u	$\mathcal{O}(iTGn^3)$
RFE [44]	w	s	$\mathcal{O}(T^2nlog_2n)$
FSV [11]	e	s	$\mathcal{O}(T^2n^2)$
LASSO (hinged) [52] (unhinged) [53]	e	s	$\mathcal{O}(T^2n^2)$
NHTP [56]	e	s	N/A
Inf-FS$_U$	f	u	$\mathcal{O}(n^3(1+T))$
Inf-FS$_S$	f	s	$\mathcal{O}(T^2+n^3(1+T))$

The methods follow the taxonomy of Section 2, and are characterized by type *(f=filter, w=wrapper, e=embedded),* class *(u = unsupervised, s = supervised) and* computational complexity. *As for the complexity, T is the number of samples, n is the number of initial features, i is the number of iterations in the case of iterative algorithms, and G is the number of classes.*

2.2.2 Supervised Approaches

The support vector machine with recursive feature elimination (RFE) [44] is a popular wrapper method that eliminates useless features in a sequential, backward fashion, ranking high a feature if it separates (by a linear SVM) samples of different classes. However, the performance of the RFE becomes unstable at some values of the filter-out factor (i.e., the number of features eliminated in each iteration) [45]. To overcome this weakness many different variants of RFE have been proposed, where the initial feature subset is selected using several SVM models with different filter-out factors, and in the second stage, features are selected by eliminating one feature at each iteration. For example, the sample weighting *SW SVM-RFE* [46] gives more weight to those samples that are close to the separating hyperplane. The *Ensemble SVM-RFE* [46] aggregates the results of several SVM-RFE selectors which are applied to randomized training data. Finally, the recursive cluster elimination (RCE) [47] has been introduced to overcome the RFE instability; it is a backward elimination algorithm that combines K-means to identify correlated clusters of features.

Some advantages of wrapper methods are:

- exploit the advantages of specific classifiers;
- in general, higher classification accuracy than filters.

The advantages of wrappers are also disadvantages, since they are suitable for some data only if their associated classifiers are, limiting the overall portability; additionally, wrappers tend to be computational expensive. On the contrary, Inf-FS is classifier agnostic, focusing only on intrinsic properties of data and their labels. We omit the RFE-X approaches in the experiments since they have been already shown to be inferior to Inf-FS in [23].

2.3 Embedded Methods

Embedded methods include the selection process as part of an internal regression model (e.g., L1, LASSO regularization, decision tree), and the overall ranking process is less prone to overfitting than wrappers.

2.3.1 Unsupervised Approaches

An example of unsupervised embedded method is the $L_{2,1}$-norm regularized discriminative feature selection for unsupervised learning (UDFS) [48]. UDFS optimizes an objective function representing a $L_{2,1}$-norm regularized minimization problem with orthogonal and locality preserving constraints [49] so that it simultaneously exploits discriminative information and feature correlations. However, such optimization problems are difficult to solve due to the nonsmooth objective function and non-convex constraints [49].

2.3.2 Supervised Approaches

In supervised learning scenarios, support vector machines play a role in many embedded approaches. The Feature Selection concaVe method (FSV) [11] generates a separating plane by maximizing the usual margin, minimizing at the same time the number of dimensions (= features) where the plane is defined. Another SVM-based feature selection approach minimizes the 0-norm with (L0) SVMs [50], encouraging sparsity. The least square regression (LSR) has also been frequently employed for feature selection. The classical embedding approach is the regression by LASSO [51], where feature selection takes place by selecting the variables that have nonzero weighting coefficients. For classification, LASSO is modified by exploiting a hinge loss (LASSO_h), which penalizes linearly with respect to the correct classification labels [52]. More recently, *unhinged* losses have shown to be more robust against biased estimates [53], which are a known issue of LASSO (LASSO_u). In the experiments we consider as comparative approaches both LASSO_h and LASSO_u.

Another way to avoid bias comes with non-convex optimization strategies, for example with hard-thresholding approaches, which work under the hypotheses of strong restricted convexity/smoothness of the function to be minimized. Recent hard thresholding approaches are GraHTP [54], [55] and NHTP [56], the latter included as comparative approach.

Advantages and disadvantages of using embedded methods are similar to those of wrappers (they depend on external techniques), however, they are less prone to over-fitting. Inf-FS is conceptually different, being a filter which *prepares* the data to a subsequent, independent classification step.

3 Our Approach

We propose two different versions of Inf-FS: the unsupervised Inf-FS_U and the supervised Inf-FS_S. In both the cases, we build upon a weighted undirected fully-connected graph $G = (V, E)$ with node set $V = \{\vec{v}_1, \dots, \vec{v}_n\}$ representing a set of n feature distributions $F = \{f_1, \dots, f_n\}$, and edge set E modeling relations among pairs of nodes (i.e., relations among distributions). In the following, the terms *feature* and *feature distribution* will be used interchangeably.

Let us represent G with its adjacency matrix A, where each of its elements $A(i,j)$, $1 \leq i,j \leq n$, models the confidence that features f_i and f_j (the nodes $\vec{v}_i$ and $\vec{v}_j$) are *both* good candidates to be selected, thanks to an associated weight function $\varphi(\cdot,\cdot)$:

$$A(i,j) = \varphi(\vec{v}_i, \vec{v}_j), \tag{1}$$

where $\varphi(\cdot,\cdot)$ is a positive, real-valued function defining the *value* of each edge. In the unsupervised version of our approach, referred as Inf-FS_U, the function $\varphi_U(\cdot,\cdot)$ is modeled as a function of both the variance and correlation of the features, while in its supervised form (Inf-FS_S), the function $\varphi_S(\cdot,\cdot)$ adds the class information using the Fisher criterion and the mutual information. It is worth noting that other types of functions can be built, with the only constraint that the higher the value of the function, the stronger the preference of selecting both the features.

3.1 Graph Building for Inf-FS_U

For the unsupervised scenario, $\varphi_U(\cdot,\cdot)$ is a weighted linear combination of two pairwise measures relating the features f_i and f_j, defined as:

$$\varphi_U(\vec{v}_i, \vec{v}_j) = \alpha E_{ij} + (1-\alpha)\overline{corr}_{ij}, \tag{2}$$

with E_{ij} indicating the maximal normalized standard deviation over the two distributions, i.e., $E_{ij} = max(\sigma_i, \sigma_j)$, where σ_i is the standard deviation over the samples $\{f_i\}$, normalized to the range [0,1] by the maximum standard deviation over the set F. The second term is the opposite of the correlation $\overline{corr}_{ij} = 1 - |Spearman(f_i, f_j)|$, with $Spearman$ indicating Spearman's rank correlation coefficient. The α is a loading coefficient $\in [0,\ 1]$, with its value being estimated during the experiments by cross validating on the training set for the classification tasks.

In practice, $\varphi_U(\cdot,\cdot) \in [0,1]$ analyzes two feature distributions, accounting for the maximal feature dispersion (the standard deviation) and how much they are uncorrelated (the Spearman rank correlation coefficient).

3.2 Graph Building for Inf-FS_S

The Inf-FS_S introduces measures which consider class membership information, where we assume to have G classes into play.

The function $\varphi_S(\vec{v}_i, \vec{v}_j)$ is formed by three factors: the first is the Fisher criterion [57]:

$$\tilde{h}_i = \frac{|\mu_{i,1} - \mu_{i,2}|^2}{\sigma_{i,1}^2 + \sigma_{i,2}^2}, \tag{3}$$

where $\mu_{i,g}$ and $\sigma_{i,g}$ are the mean and standard deviation, respectively, assumed by the ith feature when considering the samples of the gth class, $1 \leq g \leq G$. The multi-class generalization is given by:

$$h_i = \frac{\sum_{g=1}^{G}(\mu_{i,g} - \hat{\mu_i})^2}{E_i^2}, \tag{4}$$

where $\hat{\mu_i}$ and E_i denote the mean and standard deviation of the whole data set corresponding to the f_i feature (i.e., $E_i^2 = \sum_{g=1}^{G}(\sigma_{i,g})^2$). This is considering intra-class compactness and inter-class separation induced by different features. The final scores are normalized to have maximum 1 and minimum 0. The closer h_i to 1, the less redundant is the ith feature, since its domain does not overlap with the other ones.

The second factor is the normalized mutual information m_i between the features samples of the ith class and the class label [58]:

$$m_i = \sum_{y \in Y} \sum_{z \in f_i} p(z,y) log\Big(\frac{p(z,y)}{p(z)p(y)}\Big), \tag{5}$$

where Y is the set of class labels and $p(\cdot,\cdot)$ stands for the joint probability distribution. Its normalized version is obtained by normalizing over all the n computed values (one for each feature into play). In practice, m_i measures the amount by which the knowledge provided by the feature vector decreases the uncertainty about a class, summed over all the classes.

The third factor is the normalized standard deviation σ_i as computed for the unsupervised case.

The three factors are weighted linearly:

$$s_i = h_i\alpha_1 + m_i\alpha_2 + \sigma_i\alpha_3, \tag{6}$$

with $1 \leq i, j \leq n$. The parameters α_k are mixing coefficients, $0 \leq \alpha_k \leq 1$, $\sum_k \alpha_k = 1$, and their values have been estimated during the experiments by cross validating on the training set for the classification tasks. Summarizing, the score s_i indicates how much a feature is not redundant (Fisher criterion) and relevant (mutual information, standard deviation) w.r.t. the other classes.

Finally, the weights of the adjacency matrix A are obtained by coupling the correspondent s as follows:

$$\varphi_S(\vec{v}_i, \vec{v}_j) = A(i,j) = s_i s_j. \tag{7}$$

It is worth noting that the formulation above is just one among the many possible alternatives that computes the value of features i and j taken together. Studying how to estimate this value in an end-to-end fashion would be probably more effective, and is subject of current work.

3.3 Feature Ranking Procedure

The Inf-FS procedure can be explained in two ways: with the properties of power series of matrices, or borrowing from the concept of absorbing Markov chain. Next, the analysis with the power series of matrices is presented, while the Markov chain view is given at Section 3.5.

Let $\gamma = \{\vec{v}_0 = i, \vec{v}_1, \ldots, \vec{v}_{l-1}, \vec{v}_l = j\}$ denote a path of length l between nodes i and j, that is, features f_i and f_j, passing through generic nodes $\vec{v}_1, \ldots, \vec{v}_{l-1}$. Let us suppose that the length l of the path is less than the total number of nodes n in the graph. In this case, a path is simply a subset of the features.

We define the overall weight associated to γ as

$$\pi_\gamma = \prod_{k=0}^{l-1} A(\vec{v}_k, \vec{v}_{k+1}), \tag{8}$$

where π_γ is actually the value of the path accounting for all the features pairs that belong to it. There can be more than one path of length l connecting nodes i and j. Therefore, we define the set $\mathbb{P}_{i,j}^l$ as containing all the paths of length l between two nodes i and j. To estimate the overall contribution of all these paths, we calculate the following sum:

$$R_l(i,j) = \sum_{\gamma \in \mathbb{P}_{i,j}^l} \pi_\gamma, \tag{9}$$

which, following standard matrix algebra, gives:

$$R_l = A^l, \tag{10}$$

that is, the power iteration of the adjacency matrix A. R_l contains now cycles, and in our feature selection view, this is equivalent to evaluate each feature several times, possibly associated to itself in a self-cycle. This is a side effect that arises with this kind of network, but this possibility holds for all the features, and is taken into account by R_l.

We can evaluate the *single feature score* for the feature $x^{(i)}$ at a given path length l as

$$c_l(i) = \sum_{j \in V} R_l(i,j) = \sum_{j \in V} A^l(i,j). \tag{11}$$

In practice, Eq. (11) models the value of the feature $x^{(i)}$ when considered in whatever selection of l features; the higher $c_l(i)$, the better. Therefore, a possible strategy could be that of ordering the features decreasingly by c_l, taking the first m obtain a relevant set. Unfortunately, the computation of c_l is expensive, bounded by $(\mathcal{O}((l-1)\cdot n^3))$: in fact, l is of the same order of n, so the computation turns out to be $\mathcal{O}(n^4)$ and becomes impractical for large sets of features to select ($> 10K$); our approach addresses this issue 1) by expanding the path length to infinity $l \rightarrow \infty$ and 2) using notions from algebra to analytically solve the ranking problem in a computationally convenient way.

Eq. (11) estimates the score for feature f_i when injected in whatever subset of l features. Taking into account all the possible path lengths can be elegantly modeled by letting $l \rightarrow \infty$.

$$c(i) = \sum_{l=1}^{\infty} c_l(i) = \sum_{l=1}^{\infty}\left(\sum_{j \in V} R_l(i,j)\right). \tag{12}$$

Let C be the geometric series of adjacency matrix A:

$$C = \sum_{l=1}^{\infty} A^l, \tag{13}$$

It is worth noting that C can be used to obtain $c(i)$ as

$$c(i) = \sum_{l=1}^{\infty} c_l(i) = \left[\left(\sum_{l=1}^{\infty} A^l\right)\mathbf{e}\right]_i = [C\mathbf{e}]_i, \tag{14}$$

where $\mathbf{e}$ indicates a $1D$ vector of ones, and the square bracket indicates the extraction of an entry of the vector, specified by the index i.

The problem is, summing infinite A^l terms could lead to divergence; in which case, regularization is needed, in the form of generating functions [59], usually employed to assign a consistent value for the sum of a possibly divergent series. There are different forms of generating functions [60]. We define the generating function for the l-path as

$$\check{c}(i) = \sum_{l=1}^{\infty} r^l c_l(i) = \sum_{l=1}^{\infty} \sum_{j \in V} r^l R_l(i,j), \tag{15}$$

where r is a real-valued regularization factor, and r^l can be interpreted as the weight for paths of length l. The parameter r has been defined as $r = 0.9/\rho(A)$, with $\rho(A)$ spectral radius of A (more on this at Section 3.4), ensuring that the infinite sum converges.

From an algebraic point of view, $\check{c}(i)$ can be efficiently computed by using the convergence property of the geometric power series of a matrix (for a proof, see Section 3.4):

$$\check{C} = (\mathbf{I} - rA)^{-1} - \mathbf{I}, \tag{16}$$

Matrix $\check{C}$ encodes the partial scores of our set of features. The goodness of this measure is strongly related to the choice of parameters that define the adjacency matrix A.

We can obtain final relevancy scores for each feature by marginalizing this quantity:

$$\check{c}(i) = [\check{C}\mathbf{e}]_i. \tag{17}$$

Ranking in decreasing order the $\check{c}$ vector gives the output of the algorithm: a ranked list of features where the most discriminative and relevant features are positioned at the top of the list. The gist of the Inf-FS is to provide a score of importance for each feature as a function of the importance of its neighbors. See Algorithms 1 (unsupervised) and 2 (supervised) for a sketch of our approaches.

3.4 Choice of the Regularization Parameter r

In this section, we want to justify the correctness of the method in terms of convergence. The value of r (used in the generating function, and introduced in the previous section, Eq. (15)) can be determined by relying on linear algebra [61]. Let us define $\{\lambda_0, \ldots, \lambda_{n-1}\}$ as the eigenvalues of the matrix A; drawing from linear algebra, we can define the spectral radius $\rho(A)$ as:

$$\rho(A) = \max_{\lambda_i \in \{\lambda_0, \ldots, \lambda_{n-1}\}} \Big(|\lambda_i|\Big).$$

For the theory of convergence of the geometric series of matrices, we also have::

$$\lim_{l \to \infty} A^l = 0 \Longleftrightarrow \rho(A) < 1 \Longleftrightarrow \sum_{l=1}^{\infty} A^l = (\mathbf{I} - A)^{-1} - \mathbf{I}.$$

Furthermore, Gelfand's formula [62] states that for every matrix norm, we have:

$$\rho(A) = \lim_{k \longrightarrow \infty} ||A^k||^{\frac{1}{k}}.$$

This formula leads directly to an upper bound for the spectral radius of the product of two matrices that commutes, given by the product of the individual spectral radii of the two matrices, that is, for each pair of matrices A and B, we have:

$$\rho(AB) \leq \rho(A)\rho(B).$$

Starting from the definition of $\check{s}(i)$ and from the following trivial consideration:

$$r^l A^l = \left(r^l \mathbf{I}\right) A^l = [(r\mathbf{I})A]^l,$$

we can use Gelfand‘s formula on the matrices $r\mathbf{I}$ and A and thus obtain:

$$\rho\Big((r\mathbf{I})A\Big) \leq \rho(r\mathbf{I})\rho(A) = r\rho(A). \tag{18}$$

For the property of the spectral radius: $\lim_{l \to \infty}(rA)^l = 0 \Longleftrightarrow \rho(rA) < 1$. Thus, we can choose r, such as $0 < r < \frac{1}{\rho(A)}$; in this way we have:

$$\begin{aligned} 0 < \rho(rA) &= \rho\Big((r\mathbf{I})A\Big) \leq \rho(r\mathbf{I})\rho(A) \\ &= r\rho(A) < \frac{1}{\rho(A)}\rho(A) = 1 \end{aligned} \tag{19}$$

that implies $\rho(rA) < 1$, and so:

$$\check{C} = \sum_{l=1}^{\infty} (rA)^l = (\mathbf{I} - rA)^{-1} - \mathbf{I}.$$

This choice of r allows us to have convergence in the sum that defines $\check{c}(i)$. Particularly, in the experiments, we use $r = \frac{0.9}{\rho(A)}$, leaving it fixed for all the experiments.

Algorithm 1. Unsupervised Infinite Feature Selection

```
Input: $F = \{\vec{f}_1, \ldots, \vec{f}_n\}$, $\alpha$
Output: $\check{c}$ final scores for each feature
  + Building the graph
  for $i = 1 : n$ do
    for $j = 1 : n$ do
      $\sigma_{ij} = max(std(f_i), std(f_j))$;
      $\overline{corr}_{ij} = 1 - |Spearman(f_i, f_j)|$;
      $A(i,j) = \alpha\sigma_{ij} + (1-\alpha)\overline{corr}_{ij}$
    end for
  end for
  + Letting paths tend to infinite
  $r = \frac{0.9}{\rho(A)}$
  $\check{C} = (\mathbf{I} - rA)^{-1} - \mathbf{I}$
  $\check{c} = \check{C}\,\mathbf{e}$
  return $\check{c}$
```

3.5 An Alternative View of Inf-FS as Absorbing Random Walks

This section provides a different perspective of the proposed framework in terms of absorbing Markov chains and random walks.

Following standard theory on stochastic processes [63], any $m \times m$ transition matrix T of a discrete time, first-order Markov chain with m states can be written in the *canonical* form, which separates *absorbing states* (having probability of self-transition $= 1$) from transient ones by re-ordering rows and columns as follows:

$$T = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ R & \tilde{A} \end{bmatrix}, \tag{20}$$

where $\tilde{A}$ is the square submatrix of size $n \times n$ giving the transition probabilities from non-absorbing to non-absorbing states ($n \leq m$), R is the non-null rectangular submatrix of size $n \times k$ giving transition probabilities from non-absorbing to absorbing states ($k = m - n$), $\mathbf{I}$ is the identity matrix of size $k \times k$, and $\mathbf{0}$ is a rectangular matrix of zeros of size $k \times n$.

Algorithm 2. Supervised Infinite Feature Selection

```
Input: $F = \{f_1, \dots, f_n\}$ , $Y = \{1, \dots, G\}$ , $\alpha_1, \alpha_2, \alpha_3$
Output: $\check{c}$ final scores for each feature
  + Building the graph
  for $i = 1 : n$ do
    $h_i \frac{\sum_{k=1}^{K} (\mu_{i,k} - \hat{\mu_i})^2}{E_i^2}$;
    $m(i) = \sum_{y \in Y} \sum_{z \in f_i} p(z,y) log\left(\frac{p(z,y)}{p(z)p(y)}\right)$
    Compute $\sigma_i$;
    $s_i = h_i\alpha_1 + m_i\alpha_2 + \sigma_i\alpha_3$
  end for
  for $i = 1 : n$ do
    for $j = 1 : n$ do
      $A(i,j) = s_i s_j$
    end for
  end for
  + Letting paths tend to infinite
  $r = \frac{0.9}{\rho(A)}$
  $\check{C} = (\mathbf{I} - rA)^{-1} - \mathbf{I}$
  $\check{c} = \check{C}\,\mathbf{e}$
  return $\check{c}$
```

When $k > 0$, it means we have non-null probability of ending in a absorbing state, with R and $\tilde{A}$ that are both substochastic, meaning that summing (separately) over their rows gives at least one row less than 1; in the case of $k = 0$ we have that the matrices $R, \mathbf{I}, \mathbf{0}$ vanish, and the transition matrix $T = \tilde{A}$ is stochastic and has no absorbing state. In the following, we assume that all of the rows of $\tilde{A}$ are substochastic, so that necessarily there is at least one absorbing state, so that $k > 0$.

With the canonical form, it becomes easy to compute different quantities, all related to the probability of having a particular random walk associated to T. In particular, the probability of having a walk of l steps[1] from state i to state j, $1 \leq i, j \leq m$ is given by

$$T^l = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ (\mathbf{I} + \tilde{A} + \tilde{A}^2 + \ldots + \tilde{A}^{l-1})R & \tilde{A}^l \end{bmatrix}. \tag{21}$$

The fact that $\tilde{A}$ is substochastic in all its rows is a sufficient condition which tells us that its spectral radius is $\rho(\tilde{A}) < 1$ [64], which is the same condition that we required for the convergence of the infinite sum at Section 3.4, this implying $\tilde{A} = rA$. Therefore, let us suppose that $\tilde{A} = rA$, for a specific r which will be discussed next, and A built as described in Sections 3.2 and 3.1, so that $\tilde{A}(i,j)$ indicates the probability of choosing feature j after having selected i. Under this probabilistic view, the higher $\tilde{A}(i,j)$, the higher the complimentarity between j and i. Going from a (transient) state of $\tilde{A}$ into an absorbing state b, $1 \leq b \leq k$, driven by probability $\tilde{A}(i,b)$, would mean to end the feature selection process. Intuitively, a high $\tilde{A}(i,b)$ would mean that no other transient state (feature) j, $k+1 \leq j \leq k+n$ is complimentary w.r.t. i. Following this perspective, we may compute T^{∞} as containing the probability of going from two states in an infinite number of steps by rewriting Eq. (21) with $\tilde{A} \rightarrow \infty = \mathbf{0}$ and

$$T^{\infty} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ CR & \mathbf{0} \end{bmatrix}, \tag{22}$$

where the matrix

$$C = \mathbf{I} + \tilde{A} + \tilde{A}^2 + \ldots + \tilde{A}^{\infty} = (\mathbf{I} - \tilde{A})^{-1}. \tag{23}$$

At this point, interesting facts do emerge:

- the matrix C of Eq. (23) resembles the matrix of Eq. (16) $\check{C} = (\mathbf{I} - rA)^{-1} - \mathbf{I}$, with $\tilde{A} = rA$ and a difference given by the identity matrix I.
- In the Markov chain hypothesis, matrix C expresses with $C(i,j)$ the expected number of visits to transient state j starting from transient state i, before to go into an absorbing state. In our feature selection case, $C(i,j)$ could be seen as the length of the path enabled by feature i before to end the process of selection: a long path means that there is a pool of features, including necessarily i and j, which are strongly complimentary among each other (that have high probability to have transitions among themselves). In the same way, considering $c = C\mathbf{e}$, c_i indicates how much, in general, feature i enable long paths, irrespective of the arrival feature j. The longer the path, the more complimentary is the feature i with respect to all the other features.
- Unfortunately, the matrix A that we build with the procedures in Section 3.1 and Section 3.2, in general, could be not substochastic, neither could be their regularized versions rA of Section 3.4. In fact, Section 3.4 indicates a necessary and sufficient condition for making rA convergent to 0 at infinity, which is not sufficient for being substochastic.

The three observations above suggest a different, stronger regularization than the one expressed by Section 3.4 ($r = 0.9/\rho(A)$), in order to be compatible with the Markov chain paradigm; in practice, we need to have rA with $r = 0.9/r_{max}$, where $r_{max} = \max_i \sum_{j=1}^{n} A(i,j)$ is the max summation over the rows of the original matrix adjacency A. This makes rA both convergent to 0 at infinity, and substochastic, unlocking an alternative, more interpretable view of our selection process.

[1] Here step means a single iteration of the stochastic process modeled by the Markov chain.

TABLE 2
Datasets and the Challenges for the Feature Selection Scenario

Dataset	Ref.	#Samples	#Classes	#Feat.	*few train*	*unbal. (+/-)*	*overlap*	*noise*	*sparse*
COLON	[12]	62	2	2K	X	(40/22)	*n.s.*	X	
LEUKEMIA	[13]	72	2	7129	X	(47/25)	*n.s.*	X	
LUNG	[14]	181	2	12533	X	(31/150)	*n.s.*	X	
LYMPHOMA	[13]	45	2	4026	X	(23/22)	*n.s.*		
PROSTATE	[15]	102	2	6033	X	(50/52)	*n.s.*		
DEXTER	[18]	2600	2	20K		(1,3K/1,3K)	X	X	X
GISETTE	[17]	6000	2	5K		(3K/3K)	X	X	
GINA	[16]	3153	2	970		(1,5K/1,6K)	X		
MADELON	[17]	2000	2	500		(1K/1K)	X	X	
VOC 2007	[19]	10K	20	4096		X	X	X	
CalTech 101	[20]	10K	102	4096		X	X	X	

The abbreviation n.s. *stands for* not specified *(for example, in the DNA microarray datasets, no information on class overlap is given in advance).*

At the same time, with the above regularization, the $\check{C}$ of matrix Eq. (16) measuring the value of a couple of features at infinity can be computed as the C matrix at Eq. (23), and, consequently, the vectors to be ordered become $\check{c} = \check{C}\mathbf{e}$ and $c = C\mathbf{e}$.

It is worth noting that $\check{c}$ and c give rise to the same ranking, so choosing one regularization $r = 0.9/\rho(A)$ or the other $r = 0.9/r_{max}$, in practice, makes absolutely no difference: the two regularizations give just two different interpretations of the same process.

3.6 Selection of the Number of Features

The vector $\check{c}$ obtained by Eq. (17) contains at the ith entry, in term of power series of matrices, the cumulative cost of having a particular feature in any (possibly infinite) subset of features. Equivalently, in terms of Markov chain, c_i of Section 3.5 represents the expected number of selections of features which are complimentary to i that have been chosen before to finish the process of feature selection.

Ranking the c vector for feature filtering under the former perspective amounts to rank features which ensures paths of higher costs, where the cost, by construction, is higher for features which are relevant and redundant. Choosing the high-ranked features ensures to consider features of high value. In the Markov chain assumption, ranking the c vector amounts to promote features which are highly complimentary to each other.

Looking at how the values of $\check{c}$ (or, equivalently, c) are distributed will give a global view of the features into play. Experimentally, we have found that the features are bipartite (especially in the supervised case), expressing features which are useful for the classification process and features that carry few or no value. In other words, it is easy to spot a structure in this data, which can be extracted by a clustering procedure.

In this paper we propose to select a particular number of features, by considering the distribution of the $\{c_i\}$ values, and select by a clustering method the features which include the first ranked feature. Different clustering strategies can be taken into account: in our case, we consider 1D Mean-shift with automatic bandwidth selection [65], which showed to be highly effective in the experiments.

Future work will be devoted in looking for alternative ways to cluster the data: in particular, we spot few cases in which the Mean-shift was not working, due to Pareto-like distributions.

4 Experiments and Results

In this section, we compare our framework with several feature selection methods considering both recent approaches [22], [39], [40], [41], [43], [53], [56], as so as some established algorithms [11], [27], [28], [31], [32], [44], [48], [52]. Methods are selected to cover the three different families presented in Section 2, i.e., filter, wrapper and embedded approaches. Tab. 1 lists the methods included in the experiments, reporting their *type* (*f* = filters, *w* = wrappers, *e* = embedded methods), and their *class* (*s* = supervised or *u* = unsupervised). Additionally, the table shows the computational complexity whereas it has been provided.

The experiments are performed on 11 different publicly available benchmarks, whose characteristics are summarized in Table 2. The benchmarks allow to evaluate the proposed approach on supervised classification problems, focusing first on small-sample, high-dimensional scenarios, studying the strengths and weaknesses of the unsupervised and supervised Inf-FS on heterogeneous datasets, dealing then with features produced by deep learning algorithms. All of these experiments evaluate the feature selection approaches when they are constrained to provide a definite number b of features; different b's are considered (see in the following sections). In addition, we evaluate the automatic subset selection capability, where the optimal number of features has also to be decided. A conclusive statistics shows the Inf-FS framework as the most versatile and effective general-purpose algorithm among the considered competitors. All of the (MATLAB) code is available at https://github.com/giorgioroffo/Infinite-Feature-Selection.

4.1 Challenge 1: Small-Sample, High-Dimensional

Treating few samples described by many features is a traditional feature selection challenge. For example, in the medical field [66] observations are often difficult to collect (e.g., in the case of rare diseases), while the number of measurements performed on each sample can easily reach the order of thousands (e.g., set of DNA sequences). The small-sample, high-dimensional scenario holds in many other fields like business

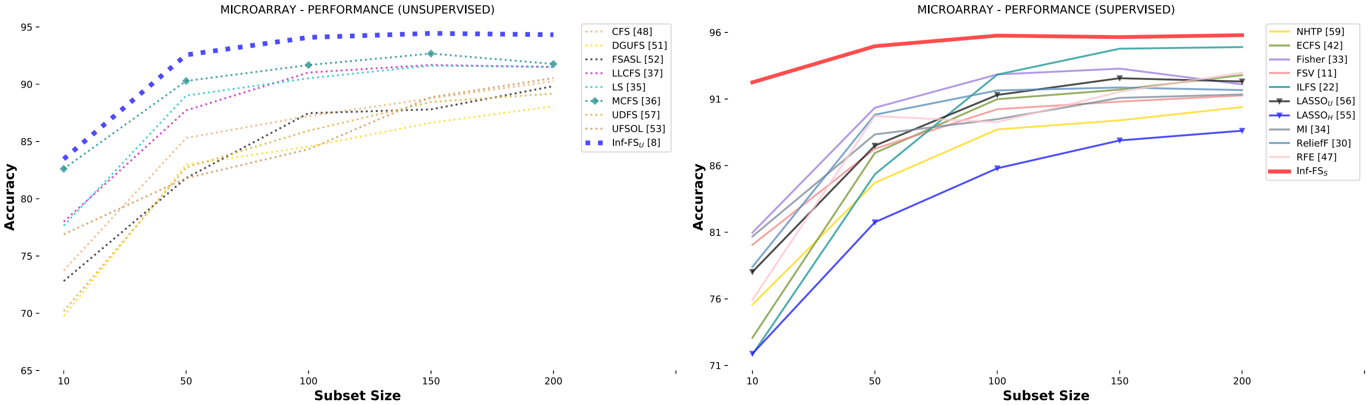


Fig. 1. Classification results on the small-sample, high-dimensional challenge. On the left, the average performance curves for unsupervised approaches, and on the right, supervised methods are shown. In all of the cases, the performance is measured at different numbers of selected features (on the x-axis).

intelligence [67], geoscience [68] and the automatic analysis of behavioural cues and social signals [69], [70]).

Here we consider five widely used small-sample, high-dimensional 2-class microarray datasets: *Colon* [12], *Lymphoma* [13], *Leukemia* [13], *Lung* [14], and *Prostate* [15]. They have been chosen for their variability in terms of number of features (from 2000 to 12533, see Tab. 2) which characterize 45 to 181 samples, because they deal with balanced and unbalanced classes, and because they are widely used in the literature. An exhaustive list of microarray small-sample, high-dimensional datasets can be found in *https://bit.ly/2OSlOfv*, while an essay on generic microarray datasets can be found in [71].

The experimental protocol consists in splitting the samples of the dataset in 70 percent for training and 30 percent for testing. The training procedure consists in building the matrix A as described in Sections 3.1 and 3.2. In the case of Inf-FS$_S$, the class labels are taken into account, while in the unsupervised case they are ignored. After the training, a selection of the ranked features is considered, by keeping the top-b features, with b variable. The selected features are used to train a linear SVM, where a 5-fold cross-validation on training data is used to set the best C regularization parameter. The same experimental protocol has been applied to all the comparative feature selection approaches.

The number b of selected features varies (i.e., $b =$10, 50, 100, 150, and 200) in order to show the performance at different regimes. The performance is specified in terms of classification accuracy. In order to avoid any bias induced by a particularly favourable split, this procedure is repeated 20 times by shuffling the data (keeping training and testing separated) and the results are averaged over the trials. A cross-validation is carried out on each training partition of the datasets to select the $\{\alpha\}$ parameters introduced in Sections 3.1 and 3.2.

Fig. 1 depicts the results: on the *left*, the average performance obtained over all of the datasets by the unsupervised approaches are reported; on the *right*, supervised approaches are shown.

On Fig. 1 (left and right), it can be seen that in both the unsupervised and supervised case, the performance improves substantially with the number of the selected features up to a knee around 50 features; after 150 features, in general, the performance tends to saturate. On the left, it can be seen that Inf-FS$_U$ outperforms the existing methods with a mild but consistent average gap. On the right, Inf-FS$_S$ achieves definitely the best performance, in particular when the number of selected features is fixed to be small (from 10 to 100).

Comparing Inf-FS$_U$ and Inf-FS$_S$ (Fig. 1, left and right) one can see that, in general, Inf-FS$_S$ works better than Inf-FS$_U$, since it uses class-label information to guide the FS process. Nonetheless, it is worth knowing (no curves are reported here) that on some datasets (COLON, LEUKEMIA and LUNG) the performance of the two approaches is comparable. This interesting aspect will be further discussed in Sections 4.4 and 4.3.

4.2 Challenge 2: Inf-FS$_U$ VS Inf-FS$_S$

This section compares the supervised and unsupervised versions of Inf-FS. Essentially, the difference between the two approaches consists of the type of functions used for weighting the graph. In fact, Inf-FS$_U$ does not employ any class-label information according to Eq. (2), while Inf-FS$_S$ is a combination of three different terms, two of them making use of the class labels (Fisher criterion and mutual information, see Eq. (6)). When the difficulty of a classification problem depends on classes that overlap, Inf-FS$_S$ can naturally favour those features that best represent the explanatory factors of the dissimilarity among the classes. On the other side, Inf-FS$_S$ suffers when features are severely correlated, even if they are representative for a specific class. In this case, variance and correlation computed by Inf-FS$_U$ do represent a very convenient option.

To validate these considerations, we consider four additional datasets from the *NIPS* feature selection challenge, namely: DEXTER [18], GISETTE, MADELON [17] and GINA [16]. GISETTE and GINA present severely overlapped classes. Indeed, the GISETTE dataset [72] has instances of “4” and “9”, two confusable handwritten digits (i.e., two overlapped classes) extracted from the MNIST data [73]. Features consist of normalized pixels and quantities derived from their combination.

The task of GINA is again handwritten digit recognition, but in this case, the two classes are *even and odd* 2-digit numbers. Obviously, only the unit digit is informative. In addition to the overlapping issues among the single digits (which are taken again from the MNIST data), a further consistent overlap is caused by the digits indicating the tens.

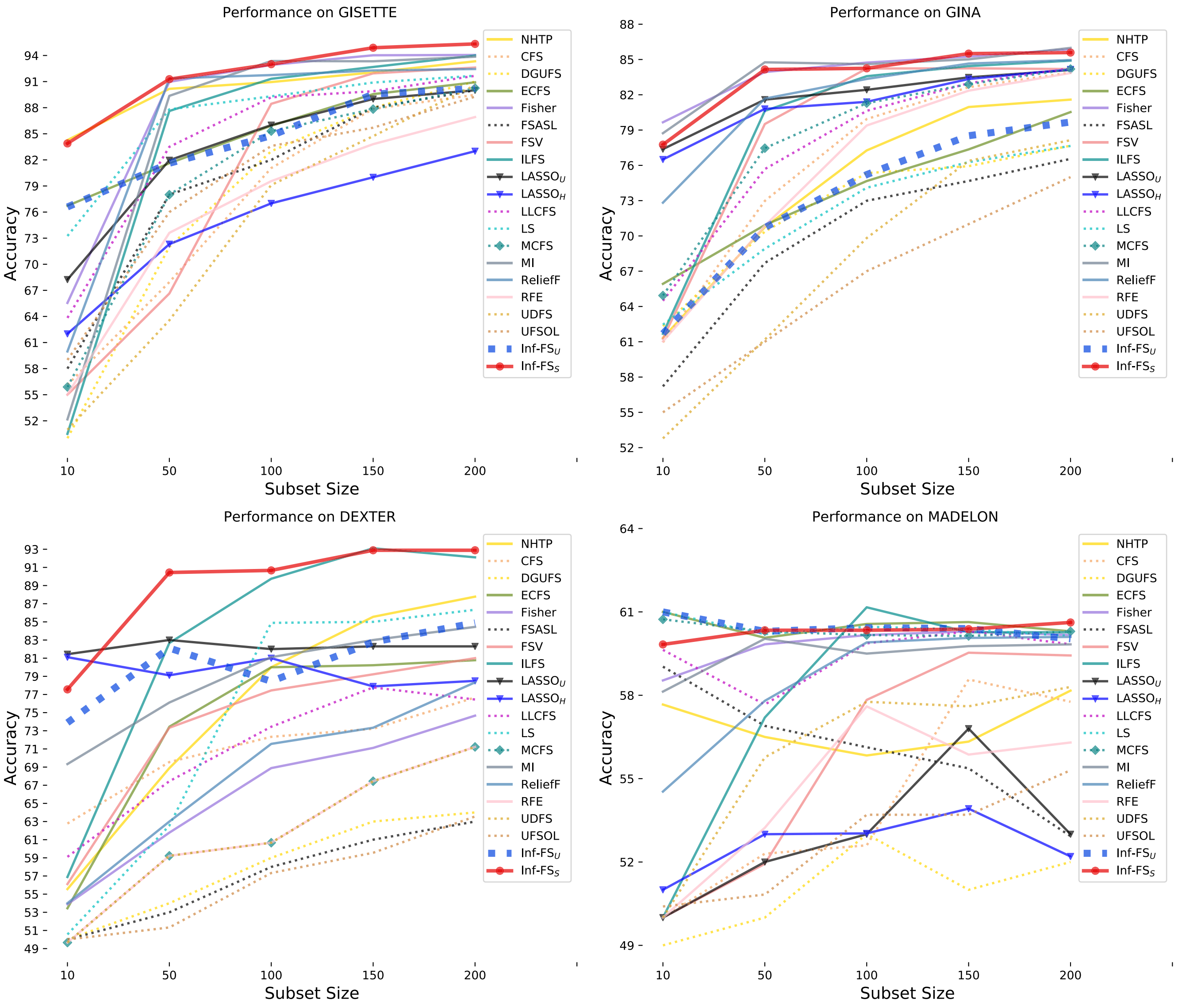


Fig. 2. Comparison between Inf-FS$_U$ and Inf-FS$_S$. All the supervised approaches are reported by solid lines and the unsupervised ones by dotted lines. Results are expressed in terms of classification accuracy (%).

As for a dataset with non-descriptive features, we selected the DEXTER dataset [18], composed by sparse continuous bag-of-words histograms, extracted from the Reuters text categorization benchmark [72]. Noise is coming from 10,053 distractors (features having no discriminative power) put voluntarily in the dataset.

A benchmark where Inf-FS$_U$ should perform comparably if not superior to Inf-FS$_S$ is MADELON [17]. In fact, MADELON is an artificial dataset containing data points grouped in 32 clusters placed on the vertices of a five-dimensional hypercube and randomly labelled +1 or -1. The five dimensions constitute 5 informative features. 15 linear combinations of those features were added to form a set of 20 (redundant) informative features. Based on those 20 features one must separate the examples into the 2 classes (corresponding to the +1, -1 labels). A number of distractor features (480) called “probes” have no predictive power. Other than this, correlated features are present.

The results are shown in Fig. 2. In general, Inf-FS$_S$ outperforms Inf-FS$_U$ on DEXTER, GINA and GISETTE and achieves a absolute top performance in most of the cases. On the other hand, Inf-FS$_U$ achieves a better performance on MADELON at 10 features w.r.t. the supervised counterpart, by discarding the several correlated features in the set, and behaves comparably with Inf-FS$_S$ at the other regimes.

Considering each dataset separately, on GISETTE (Fig. 2 top-left) Inf-FS$_S$ betters all the comparative approaches when using 10 features, having NHTP close to its performance, while in the other supervised cases the gap is substantial. Unsupervised approaches do comparably to supervised ones when it comes to 10 features, but this is probably due to the fact that 10 features are definitely too few over the 5K which are originally available, and where many of them are probably equally useful. In fact, when the number of allowed features is growing (150, 200), it is visible that most supervised approaches better the unsupervised ones. Among the unsupervised approaches, our Inf-FS$_U$ ranks approximately third after LLCFS [9] and LS [26], since the former is driven by variance and correlation, and this does not allow to unveil features which are overlapped among classes. Notably, LLCFS [9] and LS [26] select features which are locality

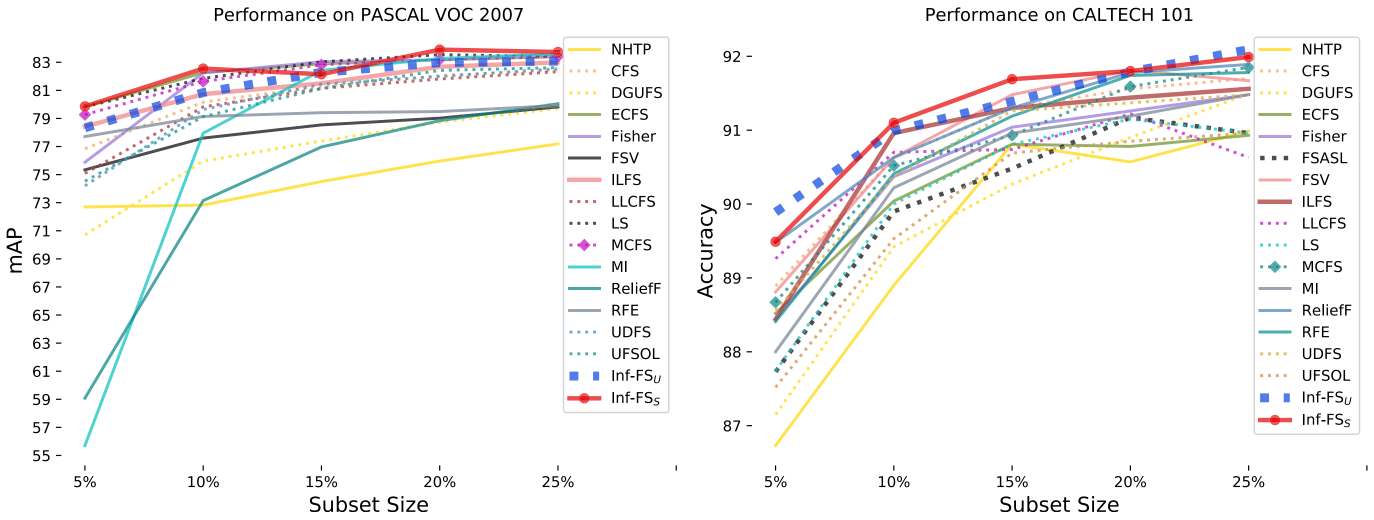


Fig. 3. Performance achieved for the image classification task reported in terms of mAP (VOC 2007) and classification accuracy (Caltech-101) while selecting the first 5, 10, 15, 20, and 25 percent features. Solid lines individuate supervised feature selection approaches, dotted lines indicate unsupervised approaches.

preserving, i.e., which agree on a clustering over the data. We may think that this clustering is capable to naturally separating the digits data, providing a more powerful solution than Inf-FS$_U$.

On GINA instead (Fig. 2 top-right), supervised approaches show at just 10 features a consistent advantage over the unsupervised methods. Here, Inf-FS$_S$ is on pair with the mutual information MI [32] and the Fisher approach [31]. In fact, Inf-FS$_S$ contains both of them in the adjacency matrix A (see Section 3.2), and they are useful to highlight features that do not overlap across classes, i.e., which are non linearly correlated with the class information. Inf-FS$_U$ gives here the worst performances, ranking approximately fourth with respect to slower and more complex approaches (MCFS [27], LLCFS [9], DGUFS [41]) which once again exploit the hypothesis that data is organized in multiple clusters which we are ignoring with Inf-FS$_U$.

DEXTER (Fig. 2 bottom-left) has the highest number of features (20K) so that restricting to only 10-200 features opens to many equivalent selections, which anyway are better individuated by INf-FS$_S$ (among the supervised approaches, except the 10 features case where LASSO shows to be better) and by INf-FS$_U$ (among the unsupervised approaches, on pair with LS [26] which is better at 100-200 features).

On MADELON we already have discussed above the results of Fig. 2 (bottom-right).

4.3 Challenge 3: Feature Selection on CNN Features

Applying feature selection on deep learning-based cues is a recent trend in image recognition [74], [75]. In fact, recent studies show that feature learning and deep learning are not immune to produce redundant or introduce useless information in the learned representations. For example, [75] proposed a generic framework for network compression and acceleration where CNNs are pruned by removing neurons with least importance, resulting in more robust networks. Neuron importance scores (usually associated to the last layer of the network, before classification) are computed by Inf-FS$_U$ as a function of the importance of all the other neurons in the layer.

In this subsection, we evaluate the performance of the proposed approach on features learned by the very deep ConvNet [21] framework, where the pre-trained model used for the ImageNet Large-Scale Visual Recognition Challenge 2014 (ILSVRC) is adopted. We use the 4,096-dimension activations of the last layer as image descriptors (L2-normalized afterwards), and we focus on the CALTECH 101 and PASCAL VOC-2007 datasets. These datasets allow for a systematic testing of the feature selection approaches taken into account in this paper, in a reasonable amount of time. We omit to choose other benchmarks (Imagenet for example) since for some of the comparative methods (LASSO and MCFS) the running time for a single trial is exceeding the week. Indeed, for each comparative approach, we perform a total of 200 runs.

According to the experimental protocol provided by the VOC challenge, a one-vs-rest SVM classifier is trained for each class (where cross-validation is used to find the best parameter C) and evaluated independently. Fig. 3 reports the performance curves obtained with the 18 feature selection approaches (solid lines for supervised approaches, dotted lines for unsupervised ones). In this case, the goal was to investigate the classification while keeping the first 5, 10, 15, 20, 25 percent of the features, corresponding to 205, 410, 614, 819 and 1024 characteristics.

From Fig. 3 (Left), it can be seen that the supervised Inf-FS_S reaches good performance in general, with a slightly superior performance w.r.t. the eigenvector centrality-based approach (ECFS). In general, the supervised approaches are organized into two groups, the most performing ones are the INFFS, ECFS, that, together with MI and ILFS gives an increase in the classification performance when adding more features. The other supervised approaches (RFE, FSV and RELIEF) seems to have a lower trend. Viceversa, all of the unsupervised approaches are more consistent among themselves, with Inf-FS positioning in the top 3 positions after LS and LLCFS. In the case of CALTECH 101 it is easy to see that the task is easier, with all of the approaches positioning in a narrow band of performance. Notably, Inf-FS$_S$ and Inf-FS$_U$ are on pair at the top position.

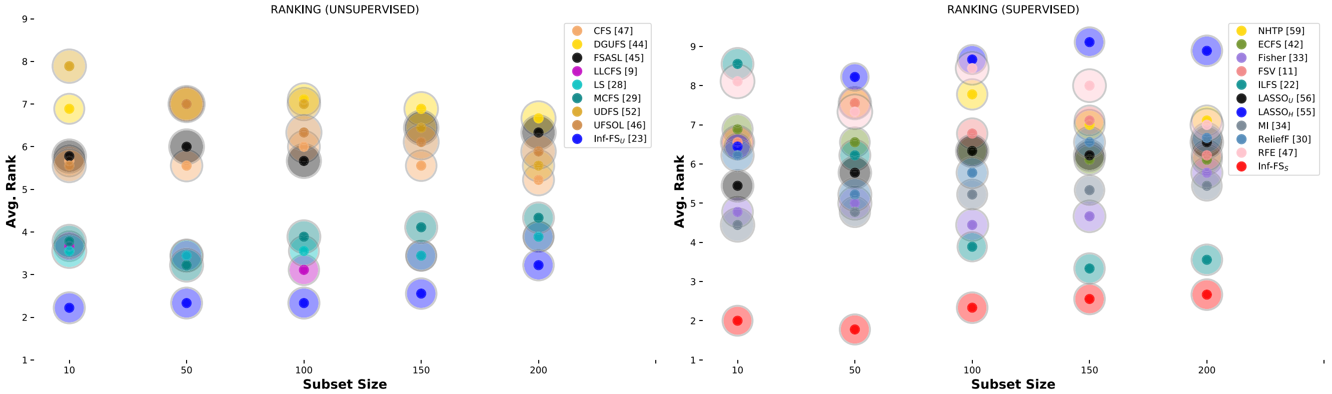


Fig. 4. Bubble plot showing the average ranking performance (y-axis) overall the datasets while increasing the number of selected features for the unsupervised approaches (left) and supervised ones (right). The area of each circle is proportional to the variance of the ranking.

On the PASCAL 2007, we performed an additional experiment, aimed at exploring the performances when spanning the number of features retained from 5 to 100 percent (Fig. 5). The idea is to check how much difference holds when keeping a small number of features with respect to the whole set. In fact, feature selection approaches often represent a compromise between admitting a lower classification performance at the price of a faster time of task execution [25]. We apply both Inf-FS$_S$ and Inf-FS$_U$. Noteworthy, both of the approaches provide features subsets leading to a performance (mAP) superior to the one obtained with the entire pool. In particular, with 25 percent of features, Inf-FS$_S$ raises the classification performances of barely 1 percentage point (83.8 percent against 83.1 percent fo the full set). Better performances are obtained in the range of 25-45 percent. The Inf-FS$_S$ shows that there is a 10 percent of features ranked last which cause a slight bending of the performances (see the 90-100 percent range). Inf-FS$_U$ has a similar behaviour, but lower in mAP score: The peak is at 45 percent of features (83.6 percent). To further explore the behavior of the approach in the range of best performance (25-45 percent for Inf-FS$_S$ and 35-45 percent for Inf-FS$_U$) we perform a fine-grained cardinality analysis reaching the absolute best of Inf-FS$_S$ at 31.5 percent features (84.18 percent mAP) and 36.5 percent for Inf-FS$_U$ (83.91 percent mAP).

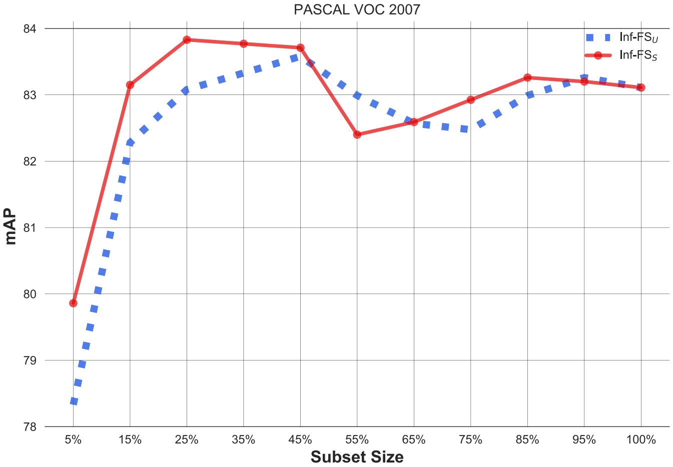


Fig. 5. Varying the cardinality of the selected features on VOC 2007. Mean average precision instead of classification accuracy is provided here.

4.4 The Versatility of Inf-FS$_U$ and Inf-FS$_S$

In this section we want to summarize the diverse experiments carried out so far, demonstrating that one of the most valuable merit of the Inf-FS framework is that *it applies favorably on every genre of feature selection scenario*. To this sake, we set up in Fig. 4 two bubble-plots showing the average ranking (the lower, the better) for each compared approach (y-axis), considering all of the used datasets (except CALTECH 101 and PASCAL VOC where LASSO did not apply, and where we evaluated different numbers of features), separating the unsupervised and supervised approaches that we have considered in the experiments.

In practice, the ranking represents the position of an approach (as classification accuracy) with respect to all the others. In the case a given approach has the best accuracy for a given benchmark, its rank on that benchmark is 1, in the case it gives the second-best accuracy the rank is 2, and so on. The average ranking shows how an approach, independently on the accuracy score, is *generically better* than the others, exhibiting a relative ordering.

The average ranking is computed with respect to different subsets of features (x-axis), and is enriched by the standard deviation in the ranking (how consistently an approach had a particular rank), depicted by the size of the blob (the larger the size, the higher the ranking variance).

The figures convey a clear message, since both Inf-FS unsupervised and supervised have the best rank, with a variance of 0.23 which indicates a stable behavior of both the approaches. Notably, Inf-FS$_S$ is definitely the most effective choice when it comes to few features selected; the mutual information-based MI [32] and the Fisher criterion for feature selection [31] follow. In the case of unsupervised approaches, Inf-FS$_U$ is first, followed by the clustering based approaches LS [26] and MCFS [27].

4.5 Challenge 4: Automatic Subset Selection

In this section, we test the process of selecting a subset of relevant features from the ranking provided by Inf-FS, explained in Section 3.6.

To this sake, we repeat all of the experiments with Inf-FS$_S$ and Inf-FS$_U$ on the 11 datasets examined so far, selecting as relevant features the ones indicated by the cluster which includes the first-ranked feature, and using them for the classification tasks. As comparative approach, we consider LASSO learned with hinge loss [52] and unhinged loss [53],

TABLE 3
The Feature Subset Selection Results Reported in Terms of Accuracy (%)

Dataset	LASSO (unhinged) [53]		LASSO (hinge) [52]		Inf-FS$_U$		Inf-FS$_S$	
	Subset	Best Prev. Perf.	Subset	Best Prev. Perf.	Subset	Best Prev. Perf.	Subset	Best Prev. Perf.
DEXTER	(10) 80.3%	(50) 82.9%	(2343) 79.9%	(10) 81.1%	(466) 83.8%	(200) 84.8%	(339) **92.8%**	(150) 92.9%
GISETTE	(2126) **95.3%**	(200) 90.3%	(2482) 85.9%	(200) 83.5%	(707) 87.7%	(200) 90.2%	(638) 94.1%	(200) 93.3
GINA	(478) **83.8%**	(200) 84.1%	(485) 80.4%	(200) 84.2%	(152) 76.2%	(200) 79.6%	(127) 83.4%	(200) 85.6%
MADELON	(233) 55.9%	(150) 56.9%	(396) 54.3%	(150) 53.9%	(48) **58.7%**	(10) 61.0%	(32) 57.1%	(200) 60.0
COLON	(22) 66.7%	(50) 85.5%	(1131) 84.4%	(200) 80.0%	(326) **91.1%**	(150) 92.7%	(174) **91.1%**	(100) 92.7%
LEUKEMIA	(18) 79.5%	(200) 97.1%	(1810) 93.8%	(150) 93.3%	(618) 94.7%	(10) 94.7%	(242) **95.2%**	(10) 94.8%
PROSTATE	(43) 87.0%	(100) 95.3%	(3168) 90.7%	(150) 93.7%	(1014) 93.0%	(100) 93.3%	(563) **94.7.6%**	(150) 96.6%
LYMPHOMA	(13) 56.7%	(150) 91.6%	(2105) 86.7%	(150) 75.8%	(674) 93.3%	(150) 93.3%	(395) **95.8%**	(200) 98.3%
LUNG	(49) 89.8%	(50) 96.6%	(5297) 96.2%	(150) 97.6%	(400) **99.9%**	(200) 99.8%	(361) **99.9%**	(200) 99.8%
VOC 2007	N/A	N/A	N/A	N/A	(1,883) 83.6%	(1024) 83.1%	(696) **83.5%**	(819) 83.8%
CalTech 101	N/A	N/A	N/A	N/A	(2250) **92.0%**	(1024) 92.1%	(942) 91.8%	(1024) 91.9%

The values enclosed in round brackets show the number of the features kept. In bold the best performance for the Subset *selection problem.*

since it is the only which allows to automatically select a precise number of features, that is, the ones which survive the shrinking process during the training stage. In particular, we individuate the best-performing LASSO by 5-fold cross-validating the regularization parameter over the training set of each benchmark, for both the hinged and unhinged versions. The results are reported in Table 3

For each pair $< dataset, method >$, we report four different quantities: in the *Subset* column we show in round brackets the number of selected features, and alongside the classification accuracy obtained with that number of features. In the *Best Prev. Perf.* column, we report in round brackets the number of features that provided the best performance obtained *in the previous experiments* (following on the right). In the table, bold scores indicate the highest classification performance among the scores obtained by the automatic selection of feature subset, not the highest absolute.

From the results, several observations can be drawn:

- The automatic selection of the number of features allow Inf-FS$_U$ and Inf-FS$_S$ to provide higher performances than LASSO on 9 out of 11 cases, with LASSO unhinged beating the Inf-FS framework on GISETTE and GINA;
- Tightly connected with the previous point, and worth noting, the Inf-FS framework selects definitely less features than the LASSO approaches (apart from the microarray datasets, where anyway LASSO unhinged is giving scarce performance). LASSO unhinged tends to keep features in a number which is highly variable; for example, it suggests a very large amount of features (2126 for GISETTE) or very few (the five microarray datasets); this seems to be correlated with the number of samples in the dataset, that, for the microarray datasets, is quite small. LASSO hinge appears to be more stable (but it gives the highest number of features).
- Inf-FS$_S$ requires for all of the datasets less features than Inf-FS$_U$ (operating with the automatic selection), showing that the class information enriches the discriminative power of the cues.
- Inf-FS performance with the automatic selection remain competitive in every scenario, while LASSO unhinged performs very poor on the small-sample, high dimensional case.

5 Conclusion

In this work we considered the feature selection problem under a brand-new perspective, i.e., as a regularization problem, where features are nodes in a weighted fully-connected graph, and a selection of l features is a path of length l through the nodes of the graph. Under this view, the proposed Inf-FS framework associates each feature to a score originating from pairwise functions (the weights of the edges) that measure relevance and non redundancy. This score has different explanations: under a power series of matrices view indicates the value that a feature can bring in a possibly infinite selection of features. Alternatively, under an absorbing Markov chain perspective, the score indicates how many times a feature would be associated to the other cues as complementary, before to end the process of selection. A precise subset of features can be provided, by examining the distribution of these scores.

Inf-FS can be customized by hand-crafting the pairwise functions, and here we presented two customizations, for unsupervised and supervised scenarios, respectively. Future work will be spent in designing an end-to-end system capable to infer the optimal pairwise functions.

Acknowledgments

This work was supported in part by the Engineering and Physical Sciences Research Council (EPSRC) under grant EP/N035305/1, and the project of the Italian Ministry of Education, Universities and Research (MIUR) “Dipartimenti di Eccellenza 2018-2022”.

References

[1] D. G. Lowe, “Object recognition from local scale-invariant features,” in *Proc. Int. Conf. Comput. Vis.*, 1999, pp. 1150–1157. [Online]. Available: http://dl.acm.org/citation.cfm?id=850924.851523

[2] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, “Speeded-up robust features (SURF),” *Comput. Vis. Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[3] S. L. Al-khafaji, J. Zhou, A. Zia, and A. W.-C. Liew, “Spectral-spatial scale invariant feature transform for hyperspectral images,” *IEEE Trans. Image Process.*, vol. 27, no. 2, pp. 837–850, Feb. 2018.

[4] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2005, pp. 886–893.

[5] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, 2003, Art. no. 1470.

[6] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Comput.*, vol. 14, no. 8, pp. 1771–1800, 2002.

[7] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.

[8] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell*, , vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[9] H. Zeng and Y.-M. Cheung, "Feature selection and kernel learning for local learning-based clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1532–1547, Aug. 2011.

[10] B. Efron *et al.*, "Least angle regression," *Ann. statist.*, vol. 32, no. 2, pp. 407–499, 2004.

[11] P. S. Bradley and O. L. Mangasarian, "Feature selection via concave minimization and support vector machines," in *Proc. 15th Int. Conf. Mach. Learn.*, 1998, pp. 82–90.

[12] U. Alon *et al.*, "Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays," in *Proc. Nat. Acad. Sci. U.S.A*, vol. 96, pp. 6745–6750, 1999.

[13] T. R. E. A. Golub, "Molecular classification of cancer: class discovery and class prediction by gene expression monitoring," *Science*, vol. 286, pp. 531–537, 1999.

[14] G. J. Gordon *et al.*, "Translation of microarray data into clinically relevant cancer diagnostic tests using gene expression ratios in lung cancer and mesothelioma," *Cancer Res.*, vol. 62, pp. 4963–4967, 2002.

[15] "Gene expression correlates of clinical prostate cancer behavior," *Cancer Cell*, vol. 1, no. 2, pp. 203–209, 2002. [Online]. Available: http://dx.doi.org/10.1016/s1535-6108(02)00030-2

[16] "GINA digit recognition database IJCNN," 2007.

[17] I. Guyon, J. Li, T. Mader, G. S. Pletscher, P. A. Pletscher, and M. Uhr, "Competitive baseline methods set new standards for the NIPS 2003 feature selection benchmark," *Pattern Recognit. Lett.*, vol. 28, no. 12, pp. 1438–1444, 2007.

[18] I. Guyon, S. Gunn, A. Ben-Hur, and G. Dror, "Result analysis of the nips 2003 feature selection challenge," in *Proc. 17th Int. Conf. Neural Inf. Process. Syst.*, 2004, pp. 545–552.

[19] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *IJCV*, vol. 111, no. 1, pp. 98–136, 2014.

[20] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, pp. 594–611, Apr. 2006.

[21] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[22] G. Roffo, S. Melzi, U. Castellani, and A. Vinciarelli, "Infinite latent feature selection: A probabilistic latent graph-based ranking approach," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1398–1406.

[23] G. Roffo, S. Melzi, and M. Cristani, "Infinite feature selection," in *Proc. Conf. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 4202–4210.

[24] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Comput. Elect. Eng.*, vol. 40, no. 1, pp. 16–28, 2014.

[25] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, 2003.

[26] X. He, D. Cai, and P. Niyogi, "Laplacian score for feature selection," in *Proc. Advances Neural Inf. Process. Syst.*, 2005, pp. 507–514.

[27] D. Cai, C. Zhang, and X. He, "Unsupervised feature selection for multi-cluster data," in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2010, pp. 333–342.

[28] H. Liu and H. Motoda, *Computational Methods of Feature Selection*. London, U.K.: Chapman and Hall, 2008.

[29] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, Aug. 2005.

[30] T. Suzuki, M. Sugiyama, T. Kanamori, and J. Sese, "Mutual information estimation reveals global associations between stimuli and biological processes," *BMC Bioinf.*, vol. 10, no. 1, 2009, Art. no. S52.

[31] Q. Gu, Z. Li, and J. Han, "Generalized fisher score for feature selection," *CoRR*, vol. abs/1202.3725, 2012.

[32] M. Zaffalon and M. Hutter, "Robust feature selection using distributions of mutual information," in *Proc. 18th Conf. Uncertainty Artif. Intell.*, 2002, pp. 577–584.

[33] J. Wang, J.-M. Wei, Z. Yang, and S.-Q. Wang, "Feature selection by maximizing independent classification information," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 4, pp. 828–841, Apr. 2017.

[34] D. Lin and X. Tang, "Conditional infomax learning: an integrated framework for feature extraction and fusion," in *Proc. Eur. Conf. Comput. Vis.*, 2006, pp. 68–82.

[35] R. Battiti, "Using mutual information for selecting features in supervised neural net learning," *IEEE Trans. Neural Netw.*, vol. 5, no. 4, pp. 537–550, Jul. 1994.

[36] A. Jakulin, "Machine learning based on attribute interactions," Ph.D. dissertation, Univerza v Ljubljani, Ljubljana, Slovenia, 2005.

[37] H. H. Yang and J. Moody, "Data visualization and feature selection: New algorithms for nongaussian data," in *Proc. Advances Neural Inf. Process. Syst.*, 2000, pp. 687–693.

[38] F. Fleuret, "Fast binary feature selection with conditional mutual information," *J. Mach. Learn. Res.*, vol. 5, pp. 1531–1555, 2004.

[39] G. Roffo and S. Melzi, "Features selection via eigenvector centrality," *5th International Workshop, NFMCP 2016, Held in Conjunction with ECML-PKDD*, Springer International Publishing, Oct 2016. doi: 10.1007/978-3-319-61461-8.

[40] G. Roffo and S. Melzi, "Ranking to Learn: Feature Ranking and Selection via Eigenvector centrality," *New Frontiers in Mining Complex Patterns, Lecture Notes in Computer Science*, Springer. Springer International Publishing, 2017, pp. 19–35. doi: 10.1007/978-3-319-61461-8_2

[41] J. Guo and W. Zhu, "Dependence guided unsupervised feature selection," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 2232–2239.

[42] L. Du and Y.-D. Shen, "Unsupervised feature selection with adaptive structure learning," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2015, pp. 209–218.

[43] J. Guo, Y. Quo, X. Kong, and R. He, "Unsupervised feature selection with ordinal locality," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2017, pp. 1213–1218.

[44] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Mach. Learn.*, vol. 46, no. 1–3, pp. 389–422, 2002. [Online]. Available: http://dx.doi.org/10.1023/A:1012487302797

[45] Y. Tang, Y.-Q. Zhang, and Z. Huang, "Development of two-stage SVM-RFE gene selection strategy for microarray expression data analysis," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 4, no. 3, pp. 365–381, Jul.–Sep. 2007.

[46] L. Yu, Y. Han, and M. E. Berens, "Stable gene selection from microarray data via sample weighting," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 9, no. 1, pp. 262–272, Jan.–Feb. 2012.

[47] M. Yousef, S. Jung, L. C. Showe, and M. K. Showe, "Recursive cluster elimination (RCE) for classification and feature selection from gene expression data," *BMC Bioinf.*, vol. 8, no. 1, 2007, Art. no. 144.

[48] Y. Yang, H. T. Shen, Z. Ma, Z. Huang, and X. Zhou, "L2,1-norm regularized discriminative feature selection for unsupervised learning," in *Proc. Conf. Int. Joint Conf. Artif. Intell.*, 2011, pp. 1589–1594.

[49] W. Chen, H. Ji, and Y. You, "An augmented lagrangian method for l1-regularized optimization problems with orthogonality constraints," *SIAM J. Sci. Comput.*, vol. 38, no. 4, pp. B570–B592, 2016.

[50] J. Weston, A. Elisseeff, B. Schölkopf, and M. Tipping, "Use of the zero-norm with linear models and kernel methods," *J. Mach. Learn. Res.*, vol. 3, pp. 1439–1461, 2003.

[51] H. Liu and H. Motoda, *Computational Methods of Feature Selection*. Boca Raton, FL, USA: CRC Press, 2007.

[52] S. A. Van de Geer *et al.*, "High-dimensional generalized linear models and the lasso," *Ann. Statist.*, vol. 36, no. 2, pp. 614–645, 2008.

[53] B. Van Rooyen, A. Menon, and R. C. Williamson, "Learning with symmetric label noise: The importance of being unhinged," in *Proc. Advances Neural Inf. Process. Syst.*, 2015, pp. 10–18.

[54] X. Yuan, P. Li, and T. Zhang, "Gradient hard thresholding pursuit for sparsity-constrained optimization," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 127–135.

[55] X.-T. Yuan, P. Li, and T. Zhang, "Gradient hard thresholding pursuit," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 6027–6069, 2017.

[56] S. Zhou, N. Xiu, and H.-D. Qi, "Global and quadratic convergence of newton hard-thresholding pursuit," 2019, *arXiv: 1901.02763*.

[57] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. New York, NY, USA: Wiley-Interscience, 2000.

[58] P. A. Estévez, M. Tesmer, C. A. Perez, and J. M. Zurada, "Normalized mutual information feature selection," *IEEE Trans. Neural Netw.*, vol. 20, no. 2, pp. 189–201, Feb. 2009.

[59] R. L. Graham, D. E. Knuth, and O. Patashnik, *Concrete Mathematics: A Foundation for Computer Science*. Boston, MA, USA: Addison-Wesley, 1994.

[60] E. Bergshoeff, "Ten physical applications of spectral zeta functions," *Classical Quantum Gravity*, vol. 13, no. 7, pp. 1–14, 1996.

[61] J. H. Hubbard and B. B. Hubbard, Eds., *Vector Calculus, Linear Algebra, and Differential Forms: A Unified Approach*, 2 ed., London, U.K.: Pearson, 2001.

[62] J. Powers and M. Sen, *Mathematical Methods in Engineering*. Cambridge, U.K.: Cambridge Univ. Press, 2015.

[63] J. G. Kemeny and J. L. Snell, *Markov chains*. New York, NY, USA: Springer, 1976.

[64] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 2012.

[65] D. Comaniciu, V. Ramesh, and P. Meer, "The variable bandwidth mean shift and data-driven scale selection," in *Proc. 8th IEEE Int. Conf. Comput. Vis.*, 2001, pp. 438–445.

[66] D. Dernoncourt, B. Hanczar, and J.-D. Zucker, "Analysis of feature selection stability on high dimension and small sample data," *Comput. Statist. Data Analy.*, vol. 71, pp. 681–693, 2014.

[67] L. Duan and L. Da Xu, "Business intelligence for enterprise systems: A survey," *IEEE Trans. Ind. Informat.*, vol. 8, no. 3, pp. 679–687, Aug. 2012.

[68] M. Pal and G. M. Foody, "Feature selection for classification of hyperspectral data by SVM," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 5, pp. 2297–2307, May 2010.

[69] A. Vinciarelli, M. Pantic, and H. Bourlard, "Social signal processing: Survey of an emerging domain," *Image Vis. Comput.*, vol. 27, no. 12, pp. 1743–1759, 2009.

[70] F. Scibelli *et al.*, "Depression speaks: Automatic discrimination between depressed and non-depressed speakers based on nonverbal speech features," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2018, pp. 6842–6846.

[71] V. Bolón-Canedo, N. Sánchez-Marono, A. Alonso-Betanzos, J. M. Benítez, and F. Herrera, "A review of microarray datasets and applied feature selection methods," *Inf. Sci.*, vol. 282, pp. 111–135, 2014.

[72] I. Guyon, "Design of experiments of the nips 2003 variable selection benchmark," *NIPS 2003 Workshop Feature Extraction Feature Selection*, vol. 253, 2003.

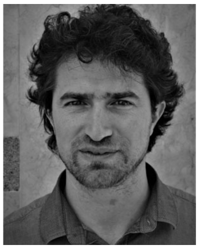
[73] Y. LeCun and C. Cortes, "MNIST handwritten digit database," 2010. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[74] M. Denil *et al.*,"Predicting parameters in deep learning," in *Proc. Advances Neural Inf. Process. Syst.*, 2013, pp. 2148–2156.

[75] R. Yu *et al.*, "NISP: Pruning networks using neuron importance score propagation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 9194–9203.

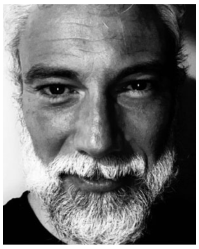
Giorgio Roffo received the European PhD degree in computer science from the University of Verona, Italy. He is currently is a research associate with the School of Computing Science, University of Glasgow. Previously, he was with the Italian Institute of Technology (IIT), Genoa, Italy. His primary research interests include the areas of machine learning, deep learning and computer vision. He contributed to the research field by publishing more than 10 articles in prestigious journals and conferences. He is in the technical program committee of leading conferences in computer vision and pattern recognition.

Simone Melzi (Member, IEEE) received the graduate degree in math summa cum laude from the University of Milan "La Statale", in 2013, and the PhD degree in computer science from the Università degli Studi di Verona, 2018. He is a postdoctoral researcher with Università degli Studi di Verona, Italy. He received the EG-Italy PhD thesis award (2018). His main research interests include geometry processing, shape matching, and 3D shape analysis. He has authored more than ten publications in leading journals and conferences.



Umberto Castellani received the Dottorato di Ricerca (PhD) degree in computer science from the University of Verona, in 2003, working on 3D data modelling and reconstruction. He is currently an associate professor with the Department of Computer Science, University of Verona. He held visiting research positions at Edinburgh University, U.K., Universite' Blaise Pascal, France, Michigan State University, Universite' D'Auvergne, France, Italian Institute of Technolgy (IIT), and University College London, U.K. His research interests include 3D geometry processing, statistical learning and medical image analysis. He is working on 3D shape processing from several acquisition systems for modelling, analysis and recognition. He has coauthored more than 130 papers published in leading conference proceedings and journals. He is member of the editorial board of the Pattern Recognition Journal and member of Program Commettee of several workshops and conferences. His research activity is mainly developed within European (EU) projects, national (MIUR) projects, and projects funded by private companies.

Alessandro Vinciarelli (Member, IEEE) is currently a full professor with the School of Computing Science, University of Glasgow and an associate academic with the Institute of Neuroscience and Psychology. His main research interests include social signal processing, the domain aimed at modeling analysis and synthesis of nonverbal behavior in social interactions. Overall, he has published more than 150 works, including one authored book, and 35 journal papers. He has been general chair of the IEEE International Conference on Social Computing in 2012 and of the ACM International Conference on Multimodal Interaction in 2017. He is or has been Principal Investigator of several national and international projects, including a Centre for Doctoral training (texttthttp://social-cdt.org), a European Network of Excellence (the SSPNet, *www.sspnet.eu*), and more than ten projects funded by the Swiss National Science Foundation end the UK Engineering and Physical Sciences Research Council. Last, but not least, Alessandro is co-founder of Klewel (*www.klewel.com*), a knowledge management company recognized with national and international awards, and scientific advisor of Neurodata Lab (*http://neurodatalab.com*).



Marco Cristani (Member, IEEE) is currently an associate professor (professore associato) with the Computer Science Department, University of Verona and an associate member with the National Research Council (CNR), External Collaborator at the Italian Institute of Technology (IIT). His main research interests include statistical pattern recognition and computer vision, mainly in deep learning and generative modeling, with application to social signal processing and fashion modeling. On these topics he has published more than 170 papers, including two edited volumes, 6 book chapters, 40 journal articles and 129 conference papers. He has organized 11 international workshops, cofounded a spin-off company, Humatics, dealing with e-commerce for fashion. He is or has been Principal Investigator of several national and international projects, including PRIN and H2020 projects. He is member of the editorial board of the Pattern Recognition and Pattern Recognition Letters journals. He is managing director of the Computer Science Park, a technology transfer centre with the University of Verona. He is member of ACM and IAPR.