Infinite Feature Selection

Giorgio Roffo
giorgio.roffo@univr.it

Simone Melzi
simone.melzi@univr.it

Marco Cristani
marco.cristani@univr.it

University of Verona, Department of Computer Science, Strada le Grazie 15, 37134, Verona, Italy

Abstract

Filter-based feature selection has become crucial in many classification settings, especially object recognition, recently faced with feature learning strategies that originate thousands of cues. In this paper, we propose a feature selection method exploiting the convergence properties of power series of matrices, and introducing the concept of infinite feature selection *(Inf-FS). Considering a selection of features as a path among feature distributions and letting these paths tend to an infinite number permits the investigation of the importance (relevance and redundancy) of a feature when injected into an arbitrary set of cues. Ranking the importance individuates candidate features, which turn out to be effective from a classification point of view, as proved by a thoroughly experimental section. The Inf-FS has been tested on thirteen diverse benchmarks, comparing against filters, embedded methods, and wrappers; in all the cases we achieve top performances, notably on the classification tasks of PASCAL VOC 2007-2012.*

1. Introduction

In many modern classification scenarios, the number of adopted features is usually very large, and it is continuously growing due to several causes, from technological reasons (Internet 2.0, Big Data) to methodological advancements (*e.g.*, deep learning). Obviously, the management of high-dimensional data requires a strong feature selection to individuate irrelevant and/or redundant features and avoid overfitting [17]. Feature selection techniques can be partitioned into three classes [17]: *wrappers*, which use classifiers to score a given subset of features; *embedded* methods, which inject the selection process into the learning of the classifier; *filter* methods, which analyze intrinsic properties of data, ignoring the classifier. Most of these methods can perform two operations, *ranking* and *subset selection*: in the former, the importance of each individual feature is evaluated, usually by neglecting potential interactions among the elements of the joint set [8]; in the latter, the final subset of features to be selected is provided. In some cases, these two operations are performed sequentially (first the ranking, then the selection) [20, 5, 15, 37, 25, 36]; in other cases, only the selection is carried out [16]. Generally, the subset selection is always supervised, while in the ranking case, methods can be supervised or not.

Generally, feature selection is *NP-hard* [17]; if there are n features in total, the goal is to select the optimal subset of $m \ll n$, to evaluate $\binom{n}{m}$ combinations; therefore, suboptimal search strategies are considered (see Sec.2). With the filters, features are first considered individually, ranked, and then a subset is extracted, some examples are MutInf [37], Relief-F [25], and SW Relief-F [36]. Conversely, with wrapper and embedded methods, subsets of features are sampled, evaluated, and finally kept as the final output, for instance, FSV [5, 15], SVM-RFE [20], Ens.SVM-RFE [36], and SW SVM-RFE [36].

In this paper, we propose a filter algorithm, which performs the ranking step in an unsupervised manner, followed by a simple cross-validation strategy for selecting the best m features. The most appealing characteristic of the approach is that it evaluates the importance of a given feature while considering all the possible subsets of features. Moreover, the score of each feature is influenced by all the other features of the set; this technique resembles the one for building path integrals [29], customized for the data clustering field [39] or the study of centralities on graphs [4]. The idea is to map the feature selection problem to an affinity graph, and then to consider a subset of features as a path connecting them. The cost of the path is given by the combination of pairwise relationships between the features, here modeled as a function of both the variance and correlation of the features, embedded in a cost matrix. By construction, the method allows to use convergence properties of the power series of matrices, and evaluate in an elegant fashion the relevance and redundancy of a feature with respect to all the other ones taken together. For this reason, we dub our approach *infinite feature selection* (Inf-FS).

The results are impressive: our approach is extensively tested on 13 benchmarks of cancer classification and

prediction on genetic data (*Colon* [32], *Lymphoma* [9], *Leukemia* [9], *Lung181* [13], DLBCL [31]), handwritten recognition (USPS [1, 6], GINA [2], *Gisette* [18]), generic feature selection (MADELON [18]), and more extensively, object recognition (Caltech 101-256 [24], PASCAL VOC 2007-2012 [10, 11]). We compare the proposed method on these datasets, against eight comparative approaches, under different conditions (number of features selected and number of training samples considered), overcoming all of them in terms of stability and classification accuracy, and setting the state of the art on 8 benchmarks, notably all the object recognition datasets. Additionally, Inf-FS also allows the investigation of the importance of different kinds of features, and in this study, the supremacy of deep-layer approaches for feature learning has been shown on the object recognition tasks.

The rest of the paper is organized as follows: Sec. 2 illustrates some related literature, mostly focusing on the comparative approaches we consider in this study. Sec. 3 details the Inf-FS approach, also giving a formal justification of its convergence properties. Extensive experiments are reported in Sec. 4, and, finally, in Sec. 5, conclusions are given, and future perspectives are envisaged.

2. Related Literature

Among the most used feature selection strategies, *Relief-F* [25] is an iterative, randomized, and supervised approach that estimates the quality of the features according to how well their values differentiate data samples that are near to each other; it does not discriminate among redundant features, and performance decreases with few data. Similar problems affect SVM-RFE [20], which is an embedded method that selects features in a sequential, backward elimination manner, ranking high a feature if it strongly separates the samples by means of a linear SVM.

Both these methods have been improved by a sample weighting policy, originating the *SW SVM-RFE* and *SW Relief-F* [36], which in practice give more weight to those samples that are close to the separating hyperplane. A bagging extension of SVM-RFE, *Ensemble SVM-RFE* [36], aggregates the results of several SVM-RFE selectors applied to randomized training data and has been empirically shown to be stronger than its original version. Other widely used filters are based on mutual information, dubbed *MutInf* here [37], which considers as a selection criterion the mutual information between the distribution of the values of a given feature and the membership to a particular class; the Fisher filter [16] computes a score for a feature as the ratio of interclass separation and intraclass variance. Even in the last two cases, features are evaluated independently, and the final feature selection occurs by aggregating the m top ranked ones.

Finally, for the wrapper method, we cite the *feature selection via concave minimization* (*FSV*) [5], where the feature selection process is injected *into* the training of an SVM by a linear programming technique.

The novelty of Inf-FS in terms of the state of the art is that it assigns a score of "importance" to each feature by taking into account *all the possible feature subsets* as paths on a graph, bypassing the combinatorial problem in a methodologically sound fashion. In this sense, the work resembles the extraction of centrality measures on a graph [4, 41], where the goal is to assign a score to each node of a graph, indicating the number of times that node is visited on whatever path of a given length. In the Inf-FS formulation, each feature is a node in the graph, a path is a selection of features, and the higher the centrality score, the most important (or most different) the feature. As a notable technical difference, in our case graphs are weighted, while in [4, 41] they are not.

3. Inf-FS method

Given a set of feature distributions $F = \{f^{(1)}, ..., f^{(n)}\}$ and $x \in \mathcal{R}$ representing a sample of the generic distribution f, we build an indirected fully-connected graph $G = (V, E)$; V is the set of vertices corresponding, one by one, to each feature distribution, while E codifies (weighted) edges, which model pairwise relations among feature distributions. Representing G as an adjacency matrix A, we can specify the nature of the weighted edges: each element a_{ij} of A, $1 \leq i, j \leq n$, represents a pairwise energy term. Energies have been represented as a weighted linear combination of two simple pairwise measures linking $f^{(i)}$ and $f^{(j)}$, defined as:

$$a_{ij} = \alpha\sigma_{ij} + (1 - \alpha)c_{ij}, \tag{1}$$

where α is a loading coefficient $\in [0, 1]$, $\sigma_{ij} = max\left(\sigma^{(i)}, \sigma^{(j)}\right)$, with $\sigma^{(i)}$ being the standard deviation over the samples $\{x\} \in f^{(i)}$, and the second term is $c_{ij} = 1 - \left|Spearman(f^{(i)}, f^{(j)})\right|$, with $Spearman$ indicating Spearman's rank correlation coefficient.

In practice, a_{ij} connects two feature distributions, accounting for the maximal feature dispersion and their correlation.

Note that the standard deviation is normalized by the maximum standard deviation over the set F and that $|Spearman(\cdot, \cdot)| \in [0, 1]$, so the two measures are comparable in terms of magnitude. The idea is that, suppose $\alpha = 0.5$, a high a_{ij} indicates at least one feature among $f^{(i)}$ and $f^{(j)}$ could be discriminant since it covers a large feature space, and $f^{(i)}$ and $f^{(j)}$ are not redundant [12].

After this pairwise analysis of features, we want to individuate the energy associated to sets larger than two feature distributions.

Let $\gamma = \{v_0 = i, v_1, ..., v_{l-1}, v_l = j\}$ denote a path of length l between vertices i and j, that is, features $f^{(i)}$

and $f^{(j)}$, through other features $v_1, ..., v_{l-1}$. For simplicity, suppose that the length l of the path is lower than the total number of features n, and the path has no cycles, so no features are visited more than once. In this setting, a path is simply a subset of the available features that come into play. We can then define the *energy* of γ as

$$\mathcal{E}_\gamma = \prod_{k=0}^{l-1} a_{v_k, v_{k+1}}, \tag{2}$$

where $\mathcal{E}_\gamma$ essentially accounts for the pairwise energies of all the features' pairs that compose the path, and it can be assumed as the joint energy of the subset of features.

Now we relax the assumption of the presence of cycles, and we define the set $\mathbb{P}^l_{i,j}$ as containing all the paths of length l between i and j; to account for the energy of all the paths of length l, we sum them as follows:

$$R_l(i,j) = \sum_{\gamma \in \mathbb{P}^l_{i,j}} \mathcal{E}_\gamma, \tag{3}$$

which, following standard matrix algebra, gives:

$$R_l(i,j) = A^l(i,j),$$

that is, the power iteration of A.

Much attention should be paid to R_l, which now contains cycles; in terms of feature selection, it is like if a single feature is considered more than once, possibly associated to itself (a self-cycle), or if two or more features are repeatedly used (e.g., the path $< 1, 2, 3, 1, 2, 3, 4 >$ connects feature 1 and 4 by a 3-variable cycle). Anyway, by extending the path length to infinity, the probability of being part of a cycle is uniform for all the features and is actually taken into account by the construction of R_l, so a sort of normalization comes into play.

Given this, we can evaluate the *single feature energy score* for the feature $f^{(i)}$ at a given path length l as

$$s_l(i) = \sum_{j \in V} R_l(i,j) = \sum_{j \in V} A^l(i,j), \tag{4}$$

In practice, Eq.4 models the role of the feature $f^{(i)}$ when considered in whatever selection of n features; the higher $s_l(i)$ is, the more energy is related to the i-th feature. Therefore, a first idea of feature selection strategy could be that of ordering the features decreasingly by s_l, taking the first m for obtaining an effective, nonredundant set. Unfortunately, the computation of s_l is expensive ($\mathcal{O}((l-1) \cdot n^3)$): as a matter of facts, 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 ($> 10K$); Inf-FS addresses this issue 1) by expanding the path length to infinity $l \to \infty$ and 2) using algebra notions to simplify the calculations in the infinite case.

3.1. Infinite sets of features

The passage to infinity implies that we have to calculate a new type of single feature score, that is,

$$s(i) = \sum_{l=1}^{\infty} s_l(i) = \sum_{l=1}^{\infty} \big(\sum_{j \in V} R_l(i,j) \big). \tag{5}$$

Let S be the geometric series of matrix A:

$$S = \sum_{l=1}^{\infty} A^l, \tag{6}$$

It is worth noting that S can be used to obtain $s(i)$ as

$$s(i) = \sum_{l=1}^{\infty} s_l(i) = [(\sum\nolimits_{l=1}^{\infty} A^l)\mathbf{e}]_i = [S\mathbf{e}]_i, \tag{7}$$

where $\mathbf{e}$ indicates a $1D$ array of ones. As it is easy to note, summing infinite A^l terms brings to divergence; in such a case, regularization is needed, in the form of generating functions [14], usually employed to assign a consistent value for the sum of a possibly divergent series. There are different forms of generating functions [3]. We define the generating function for the l-path as

$$\check{s}(i) = \sum_{l=1}^{\infty} r^l s_l(i) = \sum_{l=1}^{\infty} \sum_{j \in V} r^l R_l(i,j), \tag{8}$$

where r is a real-valued regularization factor, and r^l can be interpreted as the weight for paths of length l. Thus, for appropriate choices of r, we can ensure that the infinite sum converges.

From an algebraic point of view, $\check{s}(i)$ can be efficiently computed by using the convergence property of the geometric power series of a matrix [23]:

$$\check{S} = (\mathbf{I} - rA)^{-1} - \mathbf{I}, \tag{9}$$

Matrix $\check{S}$ encodes all the information about the energy of our set of features, the goodness of this measure is strongly related to the choice of parameters that define the underlying adjacency matrix A.

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

$$\check{s}(i) = [\check{S}\mathbf{e}]_i, \tag{10}$$

and by ranking in decreasing order the $\check{s}(i)$ energy scores, we obtain a rank for the feature to be selected. It is worth noting that so far, no label information has been employed. The ranking can be used to determine the number m of features to select, by adopting whatever classifier and feeding it with a subset of the ranked features, starting from the most energetic one downwards, and keeping the m that ensures

the highest classification score. This last operation resembles the works on graph centrality [4] (see for an example [41]), whose goal was to rank nodes in social networks that would be visited the most, along whatever path in the structure of the network. In our case, the top entries in the rank are those features more different w.r.t all the other ones, irrespective on the subsets of cues they lie. Procedure 1 reports the sketch of our algorithm.

Procedure 1 Infinite Feature Selection

Input: $F = \{f^{(1)}, ..., f^{(n)}\}$, α
Output: $\check{s}$ energy scores, for each feature
  Building the graph
  for $i = 1 : n$ **do**
    for $j = 1 : n$ **do**
      $\sigma_{ij} = max(std(f^{(i)}), std(f^{(j)}))$
      $c_{ij} = 1 - |Spearman(f^{(i)}, f^{(j)})|$
      $A(i,j) = \alpha\sigma_{ij} + (1-\alpha)c_{ij}$
    end for
  end for
  Letting paths tend to infinite
  $r = \frac{0.9}{\rho(A)}$
  $\check{S} = (\mathbf{I} - rA)^{-1} - \mathbf{I}$
  $\check{s} = \check{S}\,\mathbf{e}$
  return $\check{s}$

3.2. Convergence analysis

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) can be determined by relying on linear algebra [23]. Consider $\{\lambda_0, ..., \lambda_{n-1}\}$ eigenvalues of the matrix $\mathbf{A}$, drawing from linear algebra, we can define the spectral radius $\rho(\mathbf{A})$ as:

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

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

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

Furthermore, Gelfand's formula [28] states that for every matrix norm, we have:

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

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 $\mathbf{A}$ and $\mathbf{B}$, we have:

$$\rho(\mathbf{AB}) \leq \rho(\mathbf{A})\rho(\mathbf{B}).$$

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

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

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

$$\rho\Big((r\mathbf{I})\,\mathbf{A}\Big) \leq \rho(r\mathbf{I})\rho(\mathbf{A}) = r\rho(\mathbf{A}). \tag{11}$$

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

$$\begin{aligned} 0 < \rho(r\mathbf{A}) &= \rho\Big((r\mathbf{I})\,\mathbf{A}\Big) \leq \rho(r\mathbf{I})\rho(\mathbf{A}) \\ &= r\rho(\mathbf{A}) < \frac{1}{\rho(\mathbf{A})}\rho(\mathbf{A}) = 1 \end{aligned} \tag{12}$$

that implies $\rho(r\mathbf{A}) < 1$, and so:

$$\check{\mathbf{S}} = \sum_{l=1}^{\infty}(rA)^l = (\mathbf{I} - r\mathbf{A})^{-1} - \mathbf{I}$$

This choice of r allows us to have convergence in the sum that defines $\check{s}(i)$. Particularly, in the experiments, we use $r = \frac{0.9}{\rho(A)}$, leaving it fixed for all the experiments. For the computational complexity of Inf-FS, see the next section.

4. Experiments

The experimental section has three main goals. The first is to explore the strengths and weaknesses of Inf-FS, also considering eight comparative approaches: four filters, three embedded methods, and one wrapper (see Table 2). The Inf-FS overcomes all of them, although it ignores class membership information, being completely unsupervised. The second goal is to show how Inf-FS, when associated to simple classification models, allows the definition of top performances on benchmarks of cancer classification and prediction on genetic data (*Colon* [32], *Lymphoma* [9], *Leukemia* [9], *Lung181* [13], DLBCL [31]), handwritten recognition (USPS [1, 6], GINA [2], *Gisette* [18]), generic feature selection (MADELON [18]) and, more extensively, object recognition (Caltech 101-256 [24], PASCAL VOC 2007-2012 [10, 11]). Regarding the object recognition tasks, the third goal is to present a study that indicates which of the features commonly used in the recognition literature (bags of words and convolutional features) are the most effective, essentially confirming the supremacy of the deep learning approaches.

For setting the best parameters (the mixing α, the C of the linear SVM, and the number of features to consider on the object recognition datasets), we use only the training set

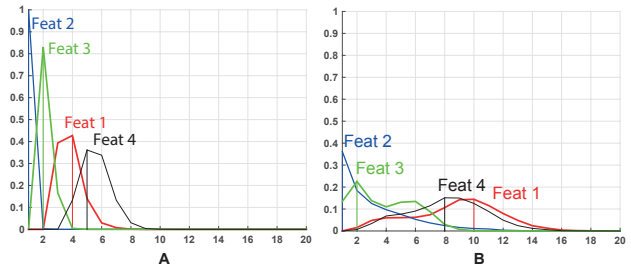

Figure 1. Ranking tendency of Inf-FS in the linear and periodic cases. The curve represents the density profile of a given feature (only the first four original features are reported) of being in a particular ranking position.

(or the validation sets in the PASCAL series), implementing a 5-fold cross validation. For a fair comparative evaluation, we adopt the same protocols used in the selected comparative approaches (partition of the dataset, cross-validation, and other settings). Other specific validation protocols are explained in the following subsections.

4.1. Synthetic experiment

In this experiment, we use synthetic data to gain insights into when Inf-FS can correctly rank the representative features, resembling the analysis done in [40]. The Inf-FS allows dealing with the set of initial features as if they constitute a weighted graph, with the weights modeling a similarity relation between features. Therefore, the design of the similarity relation suggests the scenarios where the approach has to be preferred.

In our case, we use Spearman rank correlation plus a variance score (see Eq. 1). Spearman individuates when features are connected with linear or nonlinear monotonic correlations, and in such cases, the approach is expected to work well. In case the features are connected in a more complicated manner (that is, via periodic relations), the approach does not work nicely. To show this, we first extract from the IRIS dataset 150 samples and 4 (independent) features. In a first case, 16 features are artificially generated as the linear convex combination of the 4 original ones. In the second case, the 16 features are generated by using a periodic function, that is, the linear convex combination of the *sin* of the features.

On these data, Inf-FS is expected to rank the four original features first, followed by the other ones. For the sake of generalization, we repeat the experiment 20 times, each one with diverse mixing coefficients. Results are shown in Fig.1. As expected, the Inf-FS works definitely better in the first case, as shown in Fig.1**A**, keeping the first 4 features in the top position, while in the second (see Fig.1**B**), it starts to produce very different orderings.

4.2. Datasets

Datasets are chosen for letting Inf-FS deal with diverse feature selection scenarios, as shown on Table 1. In the details, we consider the problems of dealing with few training samples and many features (*few train* in the table), unbalanced classes (*unbalanced*), or classes that severely overlap (*overlap*), or whose samples are noisy (*noise*) due to: a) complex scenes where the object to be classified is located (as in the VOC series) or b) many outliers (as in the genetic datasets, where samples are often *contaminated*, that is, artifacts are injected into the data during the creation of the samples). Lastly we consider the *shift* problem, where the samples used for the test are not congruent (coming from the same experimental conditions) with the training data.

Table 1 also reports the best classification performances so far (accuracy or average precision depending on the task), referring to the studies that produced them.

4.3. Comparative approaches

Table 2 lists the methods compared, whose details can be found in Sec. 2. Here we just note their *type*, that is, f = filters, w = wrappers, e = embedded methods, and their *class*, that is, s = supervised or u = unsupervised (using or not using the labels associated with the training samples in the ranking operation). Additionally, we report their computational complexity (if it is documented in the literature); finally, we report their timing when applied to a randomly generated dataset consisting of 20 classes, 10k samples, and 1k features (features follow a uniform distribution (range [0,1000])), on an Intel i7-4770 CPU 3.4GHz 64-bit, 16.0 GB of RAM, using MATLAB ver. 2015a. Note that only four of them have publicly available codes (that is, Relief-F [25] , FSV [5, 15], Fisher [16], and MutInf [37]), while in the other cases, we refer to the results reported in the literature.

The complexity of our approach is $\mathcal{O}(n^{2.37} + \frac{n^2}{2}T)$, the calculation of the matrix inversion for an $n \times n$ matrix requires $O(n^{2.37})$ [35], and the second term $O(\frac{n^2}{2}T)$ comes from the estimate of $a_{i,j}$ energies. This complexity allows our approach to obtain the timing that is the second best among the ones whose codes are publicly available.

4.4. Exp. 1: Varying the cardinality of the selected features

In the first experiment, we consider the protocol of [36], which starts with a pool of features characterizing the training data. These features are selected, generating different subsets of different cardinalities. The training data described by the selected features is then used to learn a linear SVM, subsequently employed to classify the test samples.

The dataset used in [36] and considered here is the *Colon*. The experiment serves to understand how well im-

Name	# samples	# classes	# feat.	*few train*	*unbal. (+/-)*	*overlap*	*noise*	*shift*	SoA
USPS [1, 6]	1.5K	2	241			X			97.4% [27]
GINA [2]	3153	2	970			X			99.7% [2]
Gisette [19]	13.5K	2	5K			X			99.9% [19]
Colon [32]	62	2	2K	X	(40/22)		X		89.6% [26]
Lymphoma [9]	45	2	4026	X					93.8%* [34]
Leukemia [9]	72	2	7129	X	(47/25)		X	X	97.2%* [36]
Lung181 [13]	181	2	12533	X	(31/150)		X	X	98.8%* [36]
DLBCL [31]	77	2	7129	X	(19/58)		X		93.3%* [30]
MADELON [19]	4.4K	2	500			X			98.0% [19]
Caltech-101 [24]	8K	102	n.s.	X					91.4%* [22]
Caltech-256 [24]	30K	257	n.s.	X		X	X		77.6%* [7]
VOC 2007 [10]	10K	20	n.s.		X		X		82.4%* [7]
VOC 2012 [11]	20K	20	n.s.		X		X		83.2%* [7]

Table 1. Panorama of the used datasets, together with the challenges for the feature selection scenario, and the state of the art so far. The abbreviation *n.s.* stands for *not specified* (for example, in the object recognition datasets, the features are not given in advance). We indicate with an asterisk each instance where our approach, together with a linear SVM, defines the new top performance.

Acronym	Type	Cl.	Compl.	Time (sec.)
SVM-RFE [20]	e	s	$\mathcal{O}(T^2 n log_2 n)$	N/A
Ens.SVM-RFE [36]	e	s	$\mathcal{O}(KT^2 n log_2 n)$	N/A
SW SVM-RFE [36]	e	s	$\mathcal{O}(T^2 n log_2 n)$	N/A
Relief-F [25]	f	s	$\mathcal{O}(iTnC)$	656.9
SW Relief-F [36]	f	s	$\mathcal{O}(iTnC)$	N/A
FSV [5, 15]	w	s	N/A	1414.6
Fisher [16]	f	s	$\sim \mathcal{O}(iCT)$	0.12
MutInf [37]	f	s	$\sim \mathcal{O}(n^2 T^2)$	8.61
Ours	f	u	$\mathcal{O}(n^{2.37}(1+T))$	4.05

Table 2. List of the feature selection approaches considered in the experiments, specified according to their *Type*, class (*Cl.*), complexity (*Compl.*), and *Time* spent on a standard feature selection task (see Sec. 4.3). As for the complexity, T is the number of samples, n is the number of initial features, K is a multiplicative constant, i is the number of iterations in the case of iterative algorithms, and C is the number of classes. The complexity of FSV cannot be specified since it is a wrapper (it depends on the chosen classifier).

portant features are ranked high by a feature selection algorithm. Table 3 presents the results in terms of AUC.

The Inf-FS outperforms all the competitors at all the features' cardinalities, being very close to the absolute state of the art. On all the other datasets, Table 4 lists the scores obtained by averaging the results of the different cardinalities of the features considered. Even in this case, the results show Inf-FS as overcoming the other competitors.

4.5. Exp. 2: CNN on object recognition datasets

This section starts with a set of tests on the object recognition datasets, that is, Caltech 101-256 and PASCAL VOC 2007-2012. The Caltech benchmarks have been taken into account due to their high number of object classes.

The second experiment considers as features the cues extracted with convolutional neural network (CNN) [38].

Colon					
	# Features				
Sel. Method	10	50	100	150	200
SVM-RFE	76.4	77.5	79.2	79.4	80.1
Ens. SVM-RFE	80.3	79.4	78.6	78.6	79.4
SW SVM-RFE	79.5	81.2	78.4	76.2	76.2
ReliefF	78.8	80.1	78.5	77.5	76.1
SW ReliefF	78.3	79.6	78.1	76.4	75.4
Fisher	84.2	86.2	87.1	86.0	86.9
MutInf	80.1	83.0	82.9	83.3	83.4
FSV	81.3	83.2	84.0	83.9	84.7
Ours	**86.4**	**89**	**89.4**	**89.3**	**89**

Table 3. Average accuracy results while varying the cardinality of the selected features.

Varying the # of the selected features - other datasets					
Dataset	FSV	Fisher	MutInf	ReliefF	**Ours**
GINA	84.2	87.1	77.7	87.7	**89.3**
USPS	91.2	88.6	92.1	92.0	**94.1**
Lymphoma	92.6	97.7	88.7	97.5	**97.9***
Leukemia	98.2	99.7	91.9	95.0	**100***
Lung181	99.7	99.7	97.0	96.8	**99.8***
DLBCL	92.5	97.7	88.7	97.5	**98.0***
MADELON	66.7	71.3	59.9	66.6	**74.6**
GISETTE	61.6	73.9	51.7	62.9	**87.3**

Table 4. Varying the cardinality of the selected features. AUC (%) on different datasets of SVM classification, averaging the performance obtained with the first 10, 50, 100, 150, and 200 features ordered by our Inf-FS algorithm. Each asterisk indicates a new top score (being an average of scores, the genuine top score for Lymphoma is 98% - 100 features and for DLBCL is 98.3% - 150 features).

The CNN features have been pre-trained on ILSVRC (we adopt the MatConvNet distribution [33]), using the 4,096-dimension activations of the penultimate layer as image features, L2-normalized afterwards. We do not perform fine

tuning, so the features are fixed, given a dataset. This will help future comparisons.

The classification results on the Caltech series have been produced by considering three random splits of training and testing data, averaging the results. For the PASCAL series, mean average precision (mAP) scores have been reported, while in the Caltech case, we show the average accuracies. Table 5 presents the results, where the percentages of the selected features are enclosed in parentheses. As for the comparative approaches, we evaluate only those whose codes are publicly available.

Object recognition by CNN features				
Methods	**Datasets**			
	VOC'07	**VOC'12**	**Cal.101**	**Cal.-256**
Relief-F	80.4 (81%)	82.7 (96%)	90.8 (81%)	79.8 (81%)
Fisher	80.7 (81%)	82.9 (87%)	90.9 (81%)	79.9 (81%)
MutInf	80.6 (88%)	82.8 (92%)	90.9 (81%)	79.9 (81%)
FSV	80.8 (86%)	81.6 (89%)	89.7 (81%)	79.6 (81%)
Ours	**83.5** (88%)	**84.0** (89%)	**91.8** (81%)	**81.5** (81%)

Table 5. Feature selection on the object recognition datasets. The numbers in parentheses are the percentages of features kept by the approach after the cross-validation phase.

As visible, the combination of our Inf-FS method and a simple linear SVM classifier gives the state of the art in all the datasets (see Table 1 for the current top scores). Notably, the top scores so far have been implemented by CNN features plus SVM, which can be considered the framework we adopt *without* the feature selection. As for the percentage of the selected features, Inf-FS is somewhat in line with the other comparative approaches.

4.6. Exp. 3: Varying the number of input samples

The availability of training samples for the feature selection operation is an important aspect to consider: actually, in some cases, it is difficult to deal with consistent quantities of data, especially in biomedical scenarios. For this sake, we consider the PASCAL VOC 2007 dataset (with plenty of data), and we evaluate our approach (and the comparative ones of the previous section) while diminishing the cardinality of the training + validation dataset, uniformly removing images from all the 20 classes, going from 5K samples to 600 circa. In all these cases, we keep 1K features for the final classification. Other than calculating the accuracy measures, we investigate how stables are the partial ranked lists produced, that is, how often the same subsets of features are selected with the same ordering. For this reason, we employ the stability index based on Jensen-Shannon Divergence D_{JS}, proposed by [21], with a [0,1] range, where 0 indicates completely random rankings and 1 means stable rankings. Interestingly, the index accounts for both the ability of having subsets of features a) with the same elements and b) ordered in the same way, where the differences at the top of the list are weighted more than those at the bottom. Table 6 presents interesting results since Inf-FS is the more

Stability analysis - PASCAL VOC 2007			
Method	**#Images**	**mAP**	$\mathbf{D_{JS}}$
Relief-F	5,011	80.4%	1.0
	2,505	80.3%	0.81
	1,252	78.2%	0.64
	626	74.4%	0.44
Fisher	5,011	80.7%	1.0
	2,505	80.3%	0.95
	1,252	78.2%	0.84
	626	74.6%	0.69
MutInf	5,011	80.6%	1.0
	2,505	80.3%	0.88
	1,252	78.2%	0.64
	626	74.5%	0.34
FSV	5,011	80.8%	1.0
	2,505	80.1%	0.90
	1,252	78.1%	0.87
	626	74.4%	0.86
Ours	5,011	83.5%	1.0
	2,505	81.9%	0.99
	1,252	79.8%	0.97
	626	76.5%	0.94

Table 6. Stability analysis: mAP scores by reducing the number of training images, and the D_{JS} index taking into account the first 1K ranked features.

stable even with 626 images, but at the same time, especially going from 5,011 to 2,505 images, it lowers more the final accuracy. This is probably because the pruned-away images could be those that the classifier uses to discriminate among the classes.

4.7. Exp. 4: Evaluating the mixing α

The coefficient α of Eq. 1 drives the algorithm in weighting the maximum variance of the features and their correlation. As previously stated, in all the experiments, we select α by cross-validation on the training set. In this section, we show how different values of α are generally effective for the classification. For this purpose, we examine all the datasets analyzed so far, fixing the percentage of the selected features to 80% of the starting set, and we keep the C value of the SVM that gave the best performance. We then vary the value of α in the interval [0,1] at a 0.1 step,

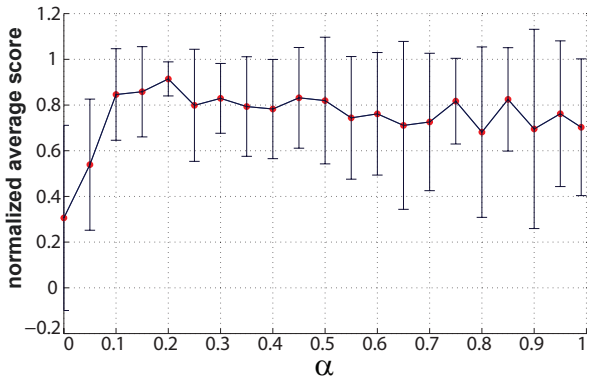

Figure 2. Evaluating the mixing α: normalized average score and related error bar (see Sec. 4.7)

keeping the classification accuracies/mAP obtained therein; given a dataset, all the classification scores with the different α values are normalized by the maximum performance. These normalized scores of all the experiments are then averaged, so having a *normalized averaged score* close to 1 means that with that α value, all the methods performed at their best. Fig. 2 illustrates the resulting curve, showing interesting and desirable characteristics. At $\alpha = 0$ (only correlation is accounted), the performance is low, rapidly increasing until $\alpha = 0.2$ where a peak is reached. After that, the curve is decreasing, even if close to 1, there are some oscillations. At $\alpha = 1$ (only maximum variance is considered), the approach works at 70% of its capabilities, on average. Analyzing the error bars (showing the 2.5 standard deviation intervals) is very informative, as it tells that the value of $\alpha = 0.2$ represents the best mix of the two feature characteristics.

4.8. Exp. 5: Augmenting the features

For our final study, we extend the kinds of features used for the object recognition datasets, including a 1,024-dimension BoW. The idea is to see if augmenting the descriptions of the images will improve the classification performance; at the same time, analyzing the kept features can give insights into the relevance of the features that come into play. Specifically, four word dictionaries of 256 entries have been calculated on a subset of 10% of the datasets VOC07/12 respectively, extracting dense PHOW features (SIFT have been extracted on 7-pixel squared cells with a 5-pixel step). Subsequently, 4-cell spatial histograms have been computed, ending with a 1,024-dimension representation for each image. Each histogram bin is thus a feature. BoWs have been concatenated to CNN features, resulting in a 5,120 feature set. As for the protocol, we have fixed the number of features to be selected at 85%, representing a valid compromise among the percentages chosen by the different approaches on the sole CNN (see Table 5). Table 7 shows the results.

CNN + BoW		
Datasets		
	VOC'07	**VOC'12**
Methods	(mAP)	(mAP)
Relief-F	81.6 ([76%,24%])	83.5 ([75%,25%])
Fisher	81.9 ([93%,7%])	83.9 ([95%,5%])
MutInf	80.7 ([97%,3%])	83.8 ([92%,8%])
FSV	81.1 ([98%,2%])	83.9 ([98%,2%])
Ours	**83.6*** ([91%,9%])	**84.1*** ([93%,7%])

Table 7. Feature selection on augmented feature descriptions. See the text.

It is evident that adding a further kind of cue is generally useful. With inf-FS the increase is minimal, probably because we are close to an intrinsic upper bound, given the features and the classifier. The numbers enclosed in square brackets (Table 7) show the percentages of the kept features, with CNN in the first position and BoW in the second position. In all the cases (except the relief-F method), CNN tends to be preferred to BoW, witnessing its expressivity. Moreover, the ordering of the features (not shown here) indicates that in almost all the cases, most of the CNN features (95% circa) are ranked ahead of the BoW ones.

5. Conclusions

In this paper we present the idea of considering feature selection as a regularization problem, where features are nodes in a graph, and a selection is a path through them. The application of our approach to all the 13 datasets against 8 competitors, at most, (employing simple linear SVM) contributes to top performances, notably setting the absolute state of the art on 8 benchmarks; The Inf-FS is also robust with a few sets of training data, performs effectively in ranking high the most relevant features, and has a very competitive complexity. This study also points to many future directions; focusing on the investigation of different relations among the features: for example, nonlinearities between the features can be encoded by theoretical information measures, instead of simple correlations. Finally, for the sake of repeatability, the source code will be posted online to provide the material needed to replicate our experiments.

References

[1] The benchmark data sets - USPS in semi-supervised learning book. 2006. 2, 4, 6

[2] GINA digit recognition database IJCNN. 2007. 2, 4, 6

[3] E. Bergshoeff. Ten physical applications of spectral zeta functions. *CQG*, 13(7), 1996. 3

[4] P. Bonacich. Power and centrality: A family of measures. *American journal of sociology*, pages 1170–1182, 1987. 1, 2, 4

[5] P. S. Bradley and O. L. Mangasarian. Feature selection via concave minimization and support vector machines. In *ICML*, pages 82–90. Morgan Kaufmann, 1998. 1, 2, 5, 6

[6] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, 2006. 2, 4, 6

[7] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *BMVC*, 2014. 6

[8] W. Duch, T. Wieczorek, J. Biesiada, and M. Blachnik. Comparison of feature ranking methods based on information entropy. In *IJCNN*, volume 2. IEEE, 2004. 1

[9] T. R. G. et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, 1999. 2, 4, 6

[10] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. 2, 4, 6

[11] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. 2, 4, 6

[12] J. H. Gennari, P. Langley, and D. Fisher. Models of incremental concept formation. *Artif. Intell.*, 40(1-3):11–61, 1989. 2

[13] G. J. Gordon, R. V. Jensen, L. li Hsiao, S. R. Gullans, J. E. Blumenstock, S. Ramaswamy, W. G. Richards, D. J. Sugarbaker, and R. Bueno. Translation of microarray data into clinically relevant cancer diagnostic tests using gene expression ratios in lung cancer and mesothelioma. *Cancer Res*, 62:4963–4967, 2002. 2, 4, 6

[14] R. L. Graham, D. E. Knuth, and O. Patashnik. *Concrete Mathematics: A Foundation for Computer Science*. Addison-Wesley, 1994. 3

[15] G. L. Grinblat, J. Izetta, and P. M. Granitto. Svm based feature selection: Why are we using the dual? In *IBERAMIA*, pages 413–422, 2010. 1, 5, 6

[16] Q. Gu, Z. Li, and J. Han. Generalized fisher score for feature selection. *CoRR*, abs/1202.3725, 2012. 1, 2, 5, 6

[17] I. Guyon. *Feature extraction: foundations and applications*, volume 207. Springer Science & Business Media, 2006. 1

[18] I. Guyon, S. Gunn, A. Ben-Hur, and G. Dror. Result analysis of the nips 2003 feature selection challenge. In *NIPS*, pages 545–552, 2004. 2, 4

[19] I. Guyon, J. Li, T. Mader, P. A. Pletscher, G. S. 0004, and M. Uhr. Competitive baseline methods set new standards for the NIPS 2003 feature selection benchmark. *PRL*, 28(12):1438–1444, 2007. 6

[20] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Mach. Learn.*, 46(1-3):389–422, 2002. 1, 2, 6

[21] R. Guzmán-Martínez and R. Alaiz-Rodríguez. Feature selection stability assessment based on the jensen-shannon divergence. *Machine Learning and Knowledge Discovery in Databases*, pages 597–612, 2011. 7

[22] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *CoRR*, abs/1406.4729, 2014. 6

[23] J. H. Hubbard and B. B. Hubbard, editors. *Vector Calculus, Linear Algebra, and Differential Forms: A Unified Approach (Edition 2)*. Pearson, 2001. 3, 4

[24] R. P. L. Fei-Fei; Fergus. One-shot learning of object categories. *IEEE TPAMI*, 28:594–611, 2006. 2, 4, 6

[25] H. Liu and H. Motoda, editors. *Computational Methods of Feature Selection*. Chapman & Hall, 2008. 1, 2, 5, 6

[26] P. Lovato, M. Bicego, M. Cristani, N. Jojic, and A. Perina. Feature selection using counting grids: Application to microarray data. LNCS, pages 629–637, 2012. 6

[27] S. Maji and J. Malik. Fast and accurate digit classification. *EECS*, 2009. 6

[28] J. Powers and M. Sen. *Mathematical Methods in Engineering*. Cambridge University Press, 2015. 4

[29] J. A. Rudnick and G. D. Gaspari. *Elements of the random walk: an introduction for advanced students and researchers*. Cambridge University Press, 2004. 1

[30] M. Seo and S. Oh. A novel divide-and-merge classification for high dimensional datasets. *Computational biology and chemistry*, 42:23–34, 2013. 6

[31] M. A. Shipp, K. N. Ross, P. Tamayo, and e. a. Weng. Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. *Nature medicine*, 8(1):68–74, 2002. 2, 4, 6

[32] U., Alon and N., Barkai and D.A., Notterman and K., Gish and S., Ybarra and D., Mack and A.J., Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. In *PNAS*, volume 96, pages 6745–6750. 1999. 2, 4, 6

[33] A. Vedaldi and K. Lenc. Matconvnet – convolutional neural networks for matlab. *CoRR*, abs/1412.4564, 2014. 6

[34] J. Wang, S. Zhou, Y. Yi, and J. Kong. An improved feature selection based on effective range for classification. *TSWJ*, 2014. 6

[35] K. Wu, C. Soci, P. P. Shum, and N. I. Zheludev. Computing matrix inversion with optical networks. *Opt. Express*, 22(1):295–304, 2014. 5

[36] L. Yu, Y. Han, and M. E. Berens. Stable gene selection from microarray data via sample weighting. *IEEE/ACM TCBB*, 9(1):262–272, 2012. 1, 2, 5, 6

[37] M. Zaffalon and M. Hutter. Robust feature selection using distributions of mutual information. In *UAI*, pages 577–584, 2002. 1, 2, 5, 6

[38] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. *CoRR*, abs/1311.2901, 2013. 6

[39] D. Zhao and X. Tang. Cyclizing clusters via zeta function of a graph. In *NIPS*, pages 1953–1960, 2008. 1

[40] P. Zhu, W. Zuo, L. Zhang, Q. Hu, and S. C. Shiu. Unsupervised feature selection by regularized self-representation. *Pattern Recognition*, 48(2):438 – 446, 2015. 5

[41] X. Zhu, A. B. Goldberg, J. Van Gael, and D. Andrzejewski. Improving diversity in ranking using absorbing random walks. In *HLT-NAACL*, pages 97–104, 2007. 2, 4