Infinite Latent Feature Selection: A Probabilistic Latent Graph-Based Ranking Approach

Giorgio Roffo
University of Glasgow
Giorgio.Roffo@Glasgow.ac.uk

Simone Melzi
University of Verona
Simone.Melzi@univr.it

Umberto Castellani
University of Verona
Umberto.Castellani@univr.it

Alessandro Vinciarelli
University of Glasgow
Alessandro.Vinciarelli@Glasgow.ac.uk

Abstract

Feature selection is playing an increasingly significant role with respect to many computer vision applications spanning from object recognition to visual object tracking. However, most of the recent solutions in feature selection are not robust across different and heterogeneous set of data. In this paper, we address this issue proposing a robust probabilistic latent graph-based feature selection algorithm that performs the ranking step while considering all the possible subsets of features, as paths on a graph, bypassing the combinatorial problem analytically. An appealing characteristic of the approach is that it aims to discover an abstraction behind low-level sensory data, that is, relevancy. Relevancy is modelled as a latent variable in a PLSA-inspired generative process that allows the investigation of the importance of a feature when injected into an arbitrary set of cues. The proposed method has been tested on ten diverse benchmarks, and compared against eleven state of the art feature selection methods. Results show that the proposed approach attains the highest performance levels across many different scenarios and difficulties, thereby confirming its strong robustness while setting a new state of the art in feature selection domain.

1. Introduction

Performance of machine learning methods is heavily dependent on the choice of features on which they are applied. Different features can entangle and hide the different explanatory factors of variation behind the data. Feature Selection (FS) aims at improving the performance of a prediction system, allowing faster and more cost-effective models, while providing a better understanding of the inherent regularities in data. In the recent *computer vision* literature there are many scenarios where FS is a crucial operation [5, 30, 10, 13, 24, 28]. From multiview face recognition [13] where FS is used to speed up the multiview face recognition process and to maintain the generalization performance, to object recognition [30], until real-time visual object tracking [28, 25] where FS dynamically identifies discriminative features that help in handling the appearance variability of the target by improving tracking performance.

In this paper, we propose a probabilistic latent graph-based feature selection algorithm that performs the ranking step by considering all the possible subsets of features exploiting the convergence properties of power series of matrices. We map the feature selection problem to an affinity graph (e.g., feature $\approx$ node), and then we consider a subset of features as a path connecting set of nodes. An appealing characteristic of the approach is that the importance of a given feature is modelled as a conditional probability of a latent variable and features, namely $P(z|f)$. Our approach aims to model an important hidden variable behind data, that is, *relevancy* in features. Raw values are observable while relevancy to a particular task is not (e.g., in classification), therefore, relevancy is modelled as an abstract latent variable. In particular, our approach consists of three main parts:

- **Pre-processing**: a quantization process is applied on raw feature distributions $\vec{x}_i$, mapping their values to a countable nominal smaller set of tokens. The pre-processing step assigns a descriptor f_i to each raw feature $\vec{x}_i$.
- **Graph-Weighting**: we build an undirected fully-connected graph, where nodes correspond, one by one, to each feature f_i, and each weighted edge among $f_i \rightsquigarrow f_j$ models the probability that features x_i and x_j are relevant. Weights are learnt automatically by a learning framework based on a variation of the probabilistic latent semantic analysis (PLSA) technique [21], which models the probability of each co-

occurrence in f_i, f_j as a mixture of conditionally independent multinomial distributions. Parameters are estimated using the Expectation Maximization (EM) algorithm.

- **Ranking**: the ranking step is done following the idea of the Infinite Feature Selection (Inf-FS) [30], that considers all the possible paths among nodes investigating the redundancy of any features when injected into arbitrary sets of cues.

The proposed method is compared against 11 state of the art feature selection methods selected from recent literature in the machine learning and pattern recognition domains, reporting results for a total of 576 unique tests (note, the source code is available at Matlab-Central). We selected 10 publicly available benchmarks of cancer classification and prediction on DNA microarray data (*Colon* [32], *Lymphoma* [14], *Leukemia* [14], *Lung* [15], Prostate [1]), handwritten character recognition (GINA [2]), text classification from the NIPS feature selection challenge (DEXTER [18]), and a movie reviews corpus for sentiment analysis (*POLARITY* [26]). More extensively, two object recognition datasets have been taken into account (PASCAL VOC 2007-2012 [11, 12]). Results show that the proposed approach represents the most robust algorithm, which achieves the highest level of performance across many different domains and challenging scenarios.

The rest of the paper is organized as follows: Sec. 2 illustrates the related literature, mostly focusing on the comparative approaches we consider in this study. Sec. 3 details the proposed approach, also giving a formal justification and interpretation based on absorbing Markov chain (Sec. 3.4). Extensive experiments are reported in Sec. 4, and, finally, in Sec. 5, conclusions are given, and future perspectives are envisaged.

2. Related Work

Since the mid-1990s, few domains used more than 20 features. The situation has changed considerably in the past few years and most papers explore domains with hundreds to tens of thousands of features. New approaches were proposed to address these challenging tasks involving many irrelevant and redundant variables and often comparably few training examples. Typically, FS techniques are partitioned into three classes [19]: *Filters*, *Wrappers* and *Embedded* methods. The proposed approach is a filter method, which analyzes intrinsic properties of data, ignoring the type of classifier. Conversely, wrappers use classifiers to score a given subset of features, and embedded methods inject the selection process directly into the learning process of the classification framework.

Among the most used filter-based strategies, *Relief-F* [23] 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. Another effective yet fast filter method is the *Fisher* method [17], which computes a score for a feature as the ratio of inter-class separation and intra-class variance, where features are evaluated independently. A Mutual Information based approach (*MI*) is proposed in [35]. MI 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. Even in the last case, features are evaluated independently, and the final feature selection occurs by aggregating the m top ranked ones. In unsupervised learning scenarios, a widely used method is the Laplacian Score (LS) [20], where the importance of a feature is evaluated by its power of locality preserving. In order to model the local geometric structure, this method constructs a nearest neighbor graph. LS algorithm seeks those features that respect this graph structure. The unsupervised feature selection for multi-cluster data is denoted MCFS in [8], which selects those features such that the multi-cluster structure of the data can be best preserved. [34] proposed a L2,1-norm regularized discriminative feature selection for unsupervised learning (UDFS) which selects the most discriminative feature subset from the whole feature set in batch mode. Feature selection and kernel learning for local learning-based clustering (LLCFS) [36] associates a weight to each feature and incorporates it into the built-in regularization of the LLC algorithm to take into account the relevance of each feature for the clustering. In the experiments, we also compare our approach against the unsupervised graph-based filter method dubbed Inf-FS [30]. 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. Another widely used FS method is SVM-RFE (RFE) [19], which is a wrapper 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. Finally, for the embedded methods, the *feature selection via concave minimization* (*FSV*) [7] is a popular FS strategy, where the selection process is injected *into* the training of an SVM by a linear programming technique. For further information, please see Tab. 2.

3. Our Approach

Given a training set X represented as a set of feature distributions $X = \{\vec{x}_1, ..., \vec{x}_n\}$, where each $m \times 1$ vector $\vec{x}_i$ is the distribution of the values assumed by the i^{th} feature with regards to the m samples, we build an undirected graph G, where nodes correspond to features and edges model relationships among pairs of nodes. Let the adjacency matrix A associated to G defining the nature of the weighted edges: each element a_{ij} of A, $1 \leq i, j \leq n$, models pairwise relationships between the features. Each weight represents the likelihood that features $\vec{x}_i$ and $\vec{x}_j$ are good candidates.

Weights can be associated to a binary function of the graph nodes:

$$a_{ij} = \varphi(\vec{x}_i, \vec{x}_j), \tag{1}$$

where $\varphi(\cdot,\cdot)$ is a real-valued potential function learned by the proposed approach in a PLSA-inspired framework. The learning framework models the probability of each co-occurrence in $\vec{x}_i, \vec{x}_j$ as a mixture of conditionally independent multinomial distributions, where parameters are learnt using the EM algorithm. Given the weighted graph G, the proposed approach analyses subsets of features as paths connecting them. The cost of each path is given by the joint probability of all the nodes belonging to it. The method exploits the convergence property of the power series of matrices as in [30], and evaluates in an elegant fashion the relevance of each feature with respect to all the other ones taken together. For this reason, we dub our approach *infinite latent feature selection* (ILFS).

3.1. Discriminative Quantization process

Since the amount of possible distinct values in $\vec{x}_i$ is huge, we map this large set of values to a countable smaller set, hereinafter referred to as set of *tokens*. Tokens are the words of our dictionary of features. Thus, each feature will be represented by a new low-dimensional vocabulary of meaningful tokens. The way used to assign each value to a specific token is based on a quantization process, we called *discriminative quantization* (DQ). The rationale behind the DQ process is to take into account how well a given feature is representative of a class before performing the many-to-few mapping.

Firstly, the Fisher criterion is used to compute a scoring vector $\Phi = [\cdot, ..., \cdot]$ which takes into account both means and standard deviations of the classes, for each sample and feature. In binary classification scenarios, this is given by

$$\Phi = \frac{1}{\mathcal{Z}}\left[\frac{(s-\mu_1)^2}{\sigma_1^2+\sigma_2^2}, \frac{(s-\mu_2)^2}{\sigma_1^2+\sigma_2^2}\right], \tag{2}$$

where s is a sample from the i^{th} feature $\vec{x}_i$, μ_k and σ_k denote the mean and standard deviation of class k, respectively. A normalization factor $\mathcal{Z}$ is introduced to ensure that the scores are a valid distribution over both classes. A natural generalization of these scores into a multi-class framework is given by

$$\Phi = \frac{1}{\mathcal{Z}}\left[\frac{(s-\mu_1)^2}{\sum_{k=1}^{K}\sigma_k^2}, ..., \frac{(s-\mu_K)^2}{\sum_{k=1}^{K}\sigma_k^2}\right], \forall_{k\in K} \tag{3}$$

where K is the number of classes, s is a single sample from the i^{th} feature. Therefore, considering all the samples, Φ results to be a $m \times K$ matrix.

Now, let us assume that the sample s belongs to class k. If $\vec{x}_i$ is a strong discriminant feature, s will *score high at* Φ_k. Then, we derive our priors π by extracting Φ scores for each feature according to the ground truth as follows:

$$\pi = diag(\Phi Y)$$

where Y is the 1-of-K representation of the ground truth. It is a particularly convenient representation where the class labels are represented by K-dimensional vectors in which one of the elements equals 1, and all remaining elements equal 0. As a result, $\pi \in [0,1]$ is a $1 \times m$ vector containing a score for each element of a particular feature i. It takes into account how well each element is represented by the feature i according to Eq.3.

Finally, quantization is performed. The first step is to divide the entire range of values $[0,1]$ into a series of $\mathcal{T}$ intervals (i.e., we use $\mathcal{T} = 6$ in this work: interval 1 corresponds to not-well-represented samples, and interval 6 is associated to well-represented samples). Secondly, we assign a token to values falling into each interval. Given the outcomes of the DQ process, we obtain a meaningful new representation of our training data X in the form of $F = \{f_1, ..., f_n\}$, where each feature is described by a vocabulary of few tokens. In other words, the derived feature representation f_i comes from x_i where each value is assigned to a token $\mathcal{T}$. According to this formulation, a strong discriminative feature will be intuitively associated to a descriptor f_i containing many relatively large tokens (e.g., $5, 6$) rather than small ones (e.g., $1, 2$).

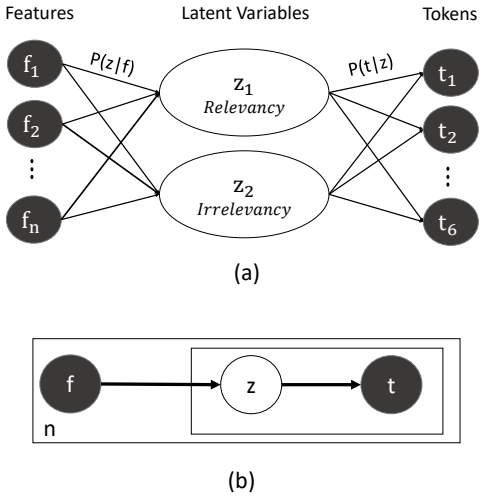

Figure 1. Illustration of the general structure of the model. (a) The intermediate layer of latent topics that links the features and the tokens. (b) The graphical model using plate representation.

3.2. From co-occurrences to graph weighting

Weighting the graph according to the nodes discriminatory power has a great influence on the quality of the ranking process. We designed a framework to automatically perform the graph weighting from training data, such that the

learnt parameters can be used to sort features according to their degrees of relevance or importance.

Our solution is based on a variation of the PLSA [21] technique, that considers co-occurrences of tokens and features, $\langle t, f \rangle$, to model the probability of each co-occurrence as a mixture of conditionally independent multinomial distributions.

In order to better understand the intuition behind the proposed model, we need to make some assumptions. We assume that a feature consists of only two topics representing the two main latent variables of any feature selection algorithms: *Relevancy* and *Irrelevancy*. Therefore, we introduce an unobserved class variable $\mathcal{Z} = \{z_1, z_2\}$ obtaining a latent variable model for co-occurrence tokens. As a result, there is a distribution $P(z|f)$ over the fixed number of topics for each feature f. Similarly, original PLSA model does not have the explicit specification of this distribution but it is indeed a multinomial distribution where $P(z|f)$ represents the probability that topic z appears in feature f. Fig. 1.(*a*) shows the general structure of the model, each feature can be represented as a mixture of concepts (Relevant/Irrelevant) weighted by the probability $P(z|f)$ and each token expresses a topic with probability $P(t|z)$. Fig. 1.(*b*) describes the generative process for each of the n features in the set by using plate representation. We can write the probability a token t appearing in feature f as follows:

$$P(t|f) = P(t|z_1)P(z_1|f) + P(t|z_2)P(z_2|f).$$

By replacing this for any feature in the set F we obtain,

$$P(f) = \prod_t \Big\{ P(t|z_1)P(z_1|f) + P(t|z_2)P(z_2|f) \Big\}.$$

The unknown parameters of this model are $P(t|z)$ and $P(z|f)$. As for PLSA, we derived the equation for computing these parameters by maximum likelihood. The log-likelihood function is given by

$$\mathcal{L} = \sum_f \sum_t Q(f,t) \log[P(t|f)]$$

where $Q(f,t)$ is the number of times token t appearing in feature f. The EM algorithm is used to compute optimal parameters. The E-step is given by

$$P(z|f,t) = \frac{P(z)P(f|z)P(t|z)}{P(z_1)P(f|z_1)P(t|z_1) + P(z_2)P(f|z_2)P(t|z_2)},$$

and the M-step is given by

$$P(t|z) = \frac{\sum_f Q(f,t)P(z|f,t)}{\sum_{f,t'} Q(f,t')P(z|f,t')},$$

$$P(f|z) = \frac{\sum_t Q(f,t)P(z|f,t)}{\sum_{f',t} Q(f',t)P(z|f',t)},$$

$$P(z) = \frac{\sum_{f,t} Q(f,t)P(z|f,t)}{\sum_{f,t} Q(f,t)}.$$

The responsibility for assigning the "condition of being relevant" to features lies to a great extent with the unobserved class variable Z. In particular, we initialize the model priors $P(t|z)$ in order to link z_1 to the abstract topic of *Relevancy*, and hence z_2 to *Irrelevancy*. By construction we limited the range of the tokens to values between 1 and 6 (see Sec.3.1), with 1 that behaves the same way as being the lowest rating for a sample of a particular feature, and 6 being the highest quality. As a result, a natural way to initialize these priors is to generate a pair of linearly spaced vectors assigning a higher probability $P(t'|Z = z_1)$ for those tokens t' which score higher, and consequently the opposite for $P(t'|Z = z_2)$.

Finally, the graph can be weighted by the estimated probability distribution $P(Z = z_1|f)$. According to Eq.1, each element a_ij of the adjacency matrix is the joint probability that the abstract topic of relevancy appears in feature f_i and f_j, namely:

$$a_{ij} = \varphi(\vec{x}_i, \vec{x}_j) = P(Z = z_1|f_i)P(Z = z_1|f_j), \qquad (4)$$

where mixing weights $P(Z = z_1|f_i)$ and $P(Z = z_1|f_j)$ are conditionally independent. Indeed, knowledge of whether $P(Z = z_1|f_i)$ occurs provides no information on the likelihood of $P(Z = z_1|f_j)$ occurring, and knowledge of whether $P(Z = z_1|f_j)$ occurs provides no information on the likelihood of $P(Z = z_1|f_i)$ occurring.

3.3. Probabilistic Infinite Feature Selection

Let $\gamma = \{v_0 = i, v_1, ..., v_{l-1}, v_l = j\}$ denote a path of length l between nodes i and j, that is, features $\vec{x}_i$ and $\vec{x}_j$, through other nodes $v_1, ..., v_{l-1}$. For simplicity, suppose that the length l of the path is lower than the total number of nodes n in the graph. In this setting, a path is simply a subset of the available features/nodes that come into play. Moreover, the network is characterized by walk structure [6], where nodes and edges can be visited multiple times.

We can then estimate the joint probability that γ is a good subset of features as

$$\mathcal{P}_\gamma = \prod_{k=0}^{l-1} a_{v_k, v_{k+1}}. \qquad (5)$$

Let us 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:

$$C_l(i,j) = \sum_{\gamma \in \mathbb{P}^l_{i,j}} \mathcal{P}_\gamma, \qquad (6)$$

which, following standard matrix algebra, gives:

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

that is, the adjacency matrix A elevated by l.

However, we want to consider all the possible paths of any length in the graph, which turns out to be the same as considering all the the possible subsets of features of any cardinality. Therefore, extending the path length to infinity implies that we have to calculate the geometric series of matrix A

$$\hat{C} = \sum_{l=1}^{\infty} A^l. \tag{7}$$

Summing infinite A^l terms brings divergence. Therefore, regularization is needed. Regularization is used to assign a consistent value for the sum of a possibly divergent series. Among the different forms of regularization [4, 16], we use a simple generating function for the l-path as

$$\check{C} = \sum_{l=1}^{\infty} r^l A^l, \tag{8}$$

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*, it is ensured that the infinite sum converges. From an algebraic point of view, $\check{C}$ can be efficiently computed by using the convergence property of the geometric power series of a matrix [22]:

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

Matrix $\check{C}$ encodes all the information about the goodness of our set of features. We can obtain final scores for each node simply by marginalizing this quantity:

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

where **e** indicates a 1D array of ones. Ranking in decreasing order the $\check{c}(i)$ scores 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 ILFS is to provide a score of importance for each feature as a function of the importance of its neighbors.

3.4. Markov chains and random walks

This section provides a probabilistic interpretation of the proposed algorithm based on Absorbing Random Walks. Here, we reformulate the problem in terms of Markov chains and random walks. The set of nodes in a Markov chain are called *states* and each move is called a *step*. Let T be the *matrix of transition probabilities*, or the *transition matrix* of the Markov chain. If the chain is currently in state v_i, then it moves to state v_j at the next step with a probability denoted by t_{ij}, and this probability does not depend upon which states the chain was in before the current state. The probabilities t_{ij} are called transition probabilities. The process can remain in the state it is in, and this occurs with probability t_{ii}. An absorbing Markov chain is a special Markov chain which has absorbing states, i.e., states which once reached cannot be transitioned out of (i.e., $t_{ii} = 1$). A Markov chain is absorbing if it has at least one absorbing state, and if from every state it is possible to go to an absorbing state in a finite number of steps. In an absorbing Markov chain, a state that is not absorbing is called transient. The transition matrix for any absorbing chain can be written in the *canonical* form

$$T = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ R & A \end{bmatrix}$$

where R is the rectangular submatrix giving transition probabilities from non-absorbing to absorbing states, A is the square submatrix giving these probabilities from non-absorbing to non-absorbing states, $\mathbf{I}$ is an identity matrix, and $\mathbf{0}$ is a rectangular matrix of zeros.

Note that R and $\mathbf{0}$ are not necessarily square. More precisely, if there are m absorbing states and n non-absorbing states, then R is $n \times m$, A is $n \times n$, $\mathbf{I}$ is $m \times m$, and $\mathbf{0}$ is $m \times n$. Iterated multiplication of the T matrix yields

$$T^2 = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ R & A \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ R & A \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ R + AR & A^2 \end{bmatrix}$$

$$T^3 = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ R + AR & A^2 \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ R & A \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ R + AR + A^2R & A^3 \end{bmatrix}$$

and hence by induction we obtain

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

The preceding example illustrates the general result that $A^l \to 0$ as $l \to \infty$. Thus

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

where the matrix

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

is called the *fundamental matrix* for the absorbing chain. Note that C, which is a square matrix with rows and columns corresponding to the non-absorbing states, is derived in the same way of Eq.9. $C(i, j)$ is the expected number of periods that the chain spends in the j^{th} non-absorbing state given that the chain began in the i^{th} non-absorbing state. Perhaps this interpretation comes from the specification of the matrix C as the infinite sum, since $A^l(i, j)$ is the probability that the process which began in the i^{th} non-absorbing state will occupy the j^{th} non-absorbing state in period l. However, $A^l(i, j)$ can also be understood as the expected proportion of period l spent in the j^{th} state. Summing over all time periods l, we thus obtain the total number of periods that the chain is expected to occupy the j^{th} state.

Dataset	Ref.	#Samples	#Classes	#Feat.	*few train*	*unbal. (+/-)*	*overlap*	*noise*	*sparse*
GINA	[2]	3153	2	970		(1,5K/1,6K)	X		
DEXTER	[18]	2600	2	20K		(1,3K/1,3K)	X		X
POLARITY	[26]	2K	2	3K		(1K/1K)			X
COLON	[32]	62	2	2K	X	(40/22)	*n.s.*	X	
LEUKEMIA	[14]	72	2	7129	X	(47/25)	*n.s.*	X	
PROSTATE	[1]	102	2	6033	X	(50/52)	*n.s.*		
LYMPHOMA	[14]	45	2	4026	X	(23/22)	*n.s.*		
LUNG	[15]	181	2	12533	X	(31/150)	*n.s.*	X	
VOC 2007	[11]	10K	20	4096		X	X	X	
VOC 2012	[12]	20K	20	4096		X	X	X	

Table 1. Datasets and the challenges for the feature selection scenario. The abbreviation *n.s.* stands for *not specified* (for example, in the DNA microarray datasets, any information on class overlap is given in advance).

4. Experiments and Results

This section has three main goals. The first goal is to evaluate the robustness of the proposed method, by choosing datasets spanning over a variety of domains and difficulties. For example, we consider the problems of dealing with few training samples and many features (few train in Tab. 1), sparse or dense dataset, 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 PASCAL VOC series) or b) many outliers (as in the genetic datasets, where samples are often contaminated, that is, artifacts are present into the data during the acquisition of the samples). The second goal is to analyze and empirically clarify how well important features are ranked high by the ILFS. We also include several comparative algorithms from recent literature, including filters, wrappers, and embedded methods. The last goal is to assess the reliability and validity of our research results. We present results obtained from more than 550 different tests, evaluating if the difference in performance is statistically significant by means of a set of Student's t-test and binomial cumulative distribution functions.

Comparative approaches and complexity

Tab. 2 lists the methods compared, where we note their *type* (*f* = filters, *w* = wrappers, *e* = embedded methods), and their *class* (*s* = supervised or *u* = unsupervised, i.e., 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). The complexity of our approach is $\mathcal{O}(n^{2.37} + in + T + C)$, the matrix inversion for a $n \times n$ matrix requires $O(n^{2.37})$ [33], and the second term $O(in+T+C)$ comes from the estimate of $P(z|f)$ through PLSA; hidden constants are the number of latent variables ($Z = 2$) and the number of tokens used ($\mathcal{T} = 6$). Finally, Tab. 2 reports the execution time of each method when applied to a randomly generated dataset consisting of 2 classes, $10k$ samples, and $5k$ features (features follow a uniform distribution - range [0,1000]), on an Intel i7 CPU 3.4GHz, 16.0 GB of RAM, using MATLAB 2016b.

ID	Acronym	Type	Cl.	Comp. Complexity	Exec.Time
1	CFS [19]	f	u	$\mathcal{O}(\frac{n^2}{2}T)$	2
2	Fisher [17]	f	s	$\mathcal{O}(Tn)$	1
3	FSV [7]	e	s	$\mathcal{O}(T^2n^2)$	2985
4	LLCFS [36]	f	u	N/A	2934
5	LS [20]	f	u	N/A	455
6	MCFS [8]	f	u	N/A	10
7	MI [35]	f	s	$\sim \mathcal{O}(n^2T^2)$	7
8	Relief-F [23]	f	s	$\mathcal{O}(iTnC)$	2024
9	RFE [19]	w	s	$\mathcal{O}(T^2nlog_2n)$	91799
10	UDFS [34]	f	u	N/A	1954
11	Inf-FS [30]	f	u	$\mathcal{O}(n^{2.37}(1+T))$	12
12	Ours	f	s	$\mathcal{O}(n^{2.37}+in+T+C)$	7

Table 2. Feature selection approaches considered in the experiments [29, 27]. The table reports their *Type*, class (*Cl.*), complexity (*Compl.*), and execution times in seconds (*Exec.Time*). 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 C is the number of classes.

4.1. Exp. #1: Deep Representation with pretraining

This section proposes a set of tests on the PASCAL VOC-2007 [11] and VOC 2012 [12] datasets. We want to assess the strengths and weaknesses of using the ILFS in an object recognition classification task. For this reason, we compare our approach against the 11 state-of-the-art FS methods reported in Tab. 2. This experiment considers as features the cues extracted with a deep convolutional neural networks (CNNs). We selected the pre-trained model called very deep ConvNets [31], which performed favorably to the state of the art for classification and detection in the ImageNet Large-Scale Visual Recognition Challenge 2014 (ILSVRC). We use the 4,096-dimension activations of the last layer as image descriptors (i.e., 4,096 features in total). According to the experimental protocol provided by the VOC challenge, a one-vs-rest SVM classifier for each class is trained (where cross-validation is used to find the best parameter C) and evaluated independently. The performance is measured as mean Average Precision (mAP) across all classes. This metric is used rather than the simple classification accuracy because some datasets (particularly the VOC series) were unbalanced in class cardinality.

The PASCAL Visual Object Classes (VOC)																								
	VOC 2007												VOC 2012											
	CFS	Fisher	FSV	LLCFS	LS	MCFS	MI	ReliefF	RFE	UDFS	Inf-FS	**Ours**	CFS	Fisher	FSV	LLCFS	LS	MCFS	MI	ReliefF	RFE	UDFS	Inf-FS	**Ours**
	90.72	**92.67**	91.57	91.32	91.43	91.00	92.46	90.30	91.44	91.98	91.37	91.75	96.83	96.97	97.20	97.70	97.32	97.30	**97.35**	96.54	96.95	96.84	96.11	97.05
	87.09	86.76	84.91	86.42	87.25	87.44	**87.79**	85.66	85.00	87.57	87.21	87.60	82.01	82.72	82.19	82.52	82.44	82.64	82.69	81.42	78.68	82.52	79.05	**82.83**
	89.72	90.17	89.51	89.11	89.74	90.23	88.75	89.20	88.61	89.12	89.25	**90.25**	89.75	90.21	89.84	89.91	89.80	90.07	**90.28**	89.19	88.56	89.81	88.44	89.44
	88.28	88.33	**88.83**	88.32	88.45	87.60	88.11	88.18	87.51	88.28	88.41	88.57	89.32	90.00	89.88	89.37	89.80	89.60	89.96	89.09	87.39	89.39	88.05	**90.20**
	56.45	56.06	56.27	54.44	55.53	54.83	55.80	54.51	50.35	57.84	54.63	56.18	60.02	60.99	60.61	60.45	60.18	60.81	**62.21**	57.93	50.91	61.31	56.18	61.47
	81.71	81.74	82.07	81.50	81.21	81.76	82.16	80.97	80.12	81.28	81.20	**83.02**	88.05	88.66	88.46	**89.55**	88.36	88.47	88.69	87.42	88.16	88.69	86.51	89.36
	86.97	87.32	**87.77**	87.28	87.09	87.13	87.47	87.93	85.52	87.71	87.47	87.23	81.42	**81.91**	81.62	81.31	81.26	81.67	81.77	80.30	73.98	81.02	78.80	81.74
	86.61	87.21	87.44	87.49	**88.06**	87.28	86.85	86.82	86.57	87.46	87.61	86.61	93.10	93.04	93.24	92.83	93.28	**93.43**	93.14	92.96	92.07	93.16	91.24	92.83
	67.05	67.19	63.50	67.25	67.53	67.14	67.35	64.74	59.34	66.93	**67.61**	66.96	71.04	**72.44**	70.46	71.40	72.29	71.70	71.60	69.72	59.31	72.03	67.42	71.89
	75.79	76.38	74.94	75.47	76.36	76.16	76.31	73.84	73.84	75.16	**76.89**	76.70	78.19	79.33	78.86	78.66	78.55	78.97	**79.64**	77.94	73.94	76.88	68.65	79.06
	73.85	75.81	74.95	**75.89**	75.10	75.55	75.41	73.12	68.97	74.53	75.16	75.07	76.04	76.55	75.40	75.73	75.97	76.55	76.43	73.35	68.45	76.50	71.19	**76.70**
	85.22	**87.47**	86.69	86.39	86.93	86.45	86.46	86.08	84.85	86.60	86.55	87.16	92.06	**92.31**	92.31	91.79	92.14	92.14	92.27	91.59	89.40	92.28	89.28	92.25
	87.40	87.74	87.78	87.43	87.64	87.79	87.91	86.93	86.81	87.16	87.37	**87.92**	88.09	**89.18**	88.61	88.29	89.00	87.93	88.97	87.21	86.19	87.97	82.46	88.59
	85.65	85.82	85.10	84.68	85.42	85.64	85.35	84.61	84.75	85.54	85.32	**85.87**	88.71	89.29	88.89	89.07	89.24	88.86	89.28	87.89	86.69	89.38	86.69	**89.59**
	92.37	92.58	91.27	92.46	92.28	92.46	**92.63**	92.39	89.70	92.20	92.15	92.22	94.24	94.37	94.04	94.21	**94.40**	94.31	94.37	94.02	92.75	94.24	91.73	93.65
	58.16	**61.33**	57.50	58.06	58.06	58.16	60.22	56.11	50.19	60.42	57.54	58.13	55.39	**56.72**	54.73	55.73	56.07	55.94	56.47	52.73	43.80	55.95	46.65	55.48
	81.13	81.13	80.33	82.38	83.10	80.94	80.88	77.99	79.51	79.94	**83.23**	81.88	81.19	82.04	80.65	80.78	81.37	81.45	**82.39**	79.72	78.97	81.77	76.39	81.37
	67.03	67.58	65.01	67.53	68.35	69.10	68.19	64.58	61.50	68.25	69.30	**70.87**	64.67	67.14	65.71	66.12	66.20	66.00	67.21	63.13	55.83	64.90	60.86	**68.11**
	92.33	91.50	92.60	92.00	92.36	92.90	92.49	91.66	91.32	**93.13**	92.08	92.50	94.85	94.38	**94.95**	94.23	94.35	94.30	94.25	93.71	94.37	94.92	93.12	94.22
	76.61	76.61	76.88	76.37	76.08	77.10	76.83	74.54	73.64	77.57	76.93	**77.62**	80.63	80.43	80.67	80.56	80.24	80.77	80.57	78.83	77.41	81.54	78.24	**81.80**
mAP:	80.52	**81.07***	80.25	80.59	80.90	80.83	**80.97***	79.46	77.98	80.93	80.86	**81.21***	82.28	**82.93***	82.42	82.48	82.61	82.65	**82.98***	81.23	78.19	82.56	78.86	**82.85***

Table 3. The image classification results achieved in terms of mean average precision (AP) scores while selecting the first $2,048$ (50%) features. In bold the top score of each class. We indicate with an asterisks the top three methods.

mAP is calculated according to the standard evaluation protocol which involves the use of the *PASCAL VOC Evaluation Server*. As for the Inf-FS, we set its parameters without any cross-validation (i.e., $\alpha = 0.2$). Tab. 3 serves to analyze how well important features are ranked high by several FS algorithms. The number of features used for both the experiments is set to: 50% of the total. The results are significant: our method achieved the best performance in terms of mean average precision (mAP) on the VOC-2007, followed by Fisher, MI. In the same way, results on VOC-12 shows that the ILFS is still one of the first three best approaches, namely: MI, Fisher, and ours. This set of FS methods achieved the best performance compared with the others, moreover, according to the overall performance over both VOC datasets the methods can be ranked as: *ILFS*, Fisher, and MI. However, it is not possible to infer which one of them performs better to a statistically significant extent (see Sec.4.3 for further details).

4.2. Exp.#2: Miscellaneous Datasets

In this section we provide results obtained on 8 different publicly available benchmarks provided without a particular definition of what the training, validation and testing set are. Therefore, the experimental protocol used in this section consists in splitting the dataset up to 2/3 for training and 1/3 for testing. In order to avoid any biases given for a particular favorable split, this procedure is repeated for 20 times and results are averaged over the trials. Accordingly, each method has been compared against all the others on the same splits for a fair comparison. Feature selection is applied only on the training set and features are selected, generating different subsets of different cardinality (i.e., 10, 50, 100, 150, and 200). As for the previous scenario, the classification is performed using a linear SVM, where a 5-fold cross validation on training data is used to set the best parameters. Results are reported in terms of mAP as for the previous experiment. Tab. 4 lists the mAP obtained by averaging the results of the different cardinality. As for the Inf-FS, we set its parameters without any cross-validation (i.e., $\alpha = 0.2$). Results show that our approach is very robust across all datasets. All the other methods show a high performance on some datasets and low on others. For example, MI is very close to a random performance on POLARITY and DEXTER, thereby indicating a weakness of the method when applied to sparse data (see Tab. 1). The ILFS is not affected by this problem, and it achieves the best significant performance on DEXTER ($\approx 20K$ features) and a high performance on POLARITY. Fisher, which performs well over all the datasets does not show the same ranking quality as ILFS. Tab. 4 also reports the overall average scores across the datasets, which clearly show that our approach outperforms all the competitors at all the features' cardinality. Min/Max values are reported in Table 4 to highlight the robustness of the ILFS to different datasets. In particular, on DNA Microarray data the overall minimum value reported by the ILFS is $+8.35\%$ over the second best (FSV). As for the other datasets, the ILFS still represents the top scoring method according to its overall average, minimum, and maximum scores.

4.3. Reliability and Validity

In order to assess whether the difference in performance is statistically significant, a set of Student's t-test have been applied to the results [3]. We use the statistical tests to determine if the accuracy given by the proposed approach is significantly different from the one of the other methods (whereas both the distribution of values were normal). The test for assessing whether the data come from normal distri-

	DNA Microarray data						Data from other sources			
Methods	COLON	LEUKEMIA	PROSTATE	LYMPHOMA	LUNG	**Average** [Min,Max]]	GINA	DEXTER	POLARITY	**Average** [Min,Max]
CFS	81.25 ± 0.08	96.27 ± 0.06	85.00 ± 0.08	84.00 ± 0.10	94.50 ± 0.17	88.20 [81.25,96.27]	81.91 ± 0.11	79.56 ± 0.06	86.99 ± 0.05	82.82 [79.56,86.99]
Fisher	87.83 ± 0.05	95.21 ± 0.006	93.55 ± 0.03	94.62 ± 0.05	97.75 ± 0.06	93.79 [87.83,97.75]	**89.36*** ± **0.03**	95.65 ± 0.06	82.61 ± 0.13	89.20 [82.61,95.65]
FSV	88.00 ± 0.05	91.57 ± 0.01	93.50 ± 0.02	89.38 ± 0.04	**98.83** ± **0.01**	92.25 [88.00,98.83]	81.73 ± 0.12	96.39 ± 0.01	86.12 ± 0.12	88.08 [81.73,96.39]
LLCFS	90.00 ± 0.05	**99.37** ± **0.02**	85.80 ± 0.09	84.12 ± 0.11	97.69 ± 0.04	91.39 [84.12,99.37]	81.91 ± 0.09	84.16 ± 0.10	**97.31** ± **0.02**	87.79 [81.91,97.31]
LS	91.58 ± 0.05	93.57 ± 0.006	82.00 ± 0.12	78.88 ± 0.16	97.81 ± 0.06	88.76 [78.88,97.81]	78.10 ± 0.08	85.25 ± 0.12	**97.77*** ± **0.02**	87.04 [78.10,97.77]
MCFS	90.92 ± 0.05	92.00 ± 0.02	76.75 ± 0.08	84.38 ± 0.09	96.53 ± 0.16	88.11 [76.75,96.53]	85.69 ± 0.07	87.80 ± 0.07	95.26 ± 0.03	89.58 [85.69,95.26]
MI	86.92 ± 0.05	93.36 ± 0.04	90.50 ± 0.04	94.00 ± 0.04	98.72 ± 0.02	92.70 [86.92,98.72]	88.85 ± 0.04	59.51 ± 0.04	56.19 ± 0.09	68.18 [56.19,88.85]
ReliefF	84.75 ± 0.07	93.07 ± 0.02	93.25 ± 0.04	91.75 ± 0.05	97.33 ± 0.03	92.03 [84.75,97.33]	88.86 ± 0.03	89.54 ± 0.12	95.82 ± 0.03	91.40 [88.86,95.82]
RFE	82.58 ± 0.09	86.43 ± 0.07	78.90 ± 0.10	77.50 ± 0.12	94.25 ± 0.17	84.53 [77.50,94.25]	83.05 ± 0.09	87.38 ± 0.09	94.20 ± 0.02	88.21 [83.05,94.20]
UDFS	88.00 ± 0.05	89.21 ± 0.07	84.25 ± 0.08	80.50 ± 0.12	96.36 ± 0.13	87.66 [80.50,96.36]	72.28 ± 0.11	80.40 ± 0.12	87.43 ± 0.08	80.03 [72.28,87.43]
Inf-FS	**96.10** ± **0.05**	**99.44** ± 0.008	92.10 ± 0.07	96.50 ± 0.06	97.36 ± 0.06	96.30 [92.10,99.44]	78.97 ± 0.04	81.95 ± 0.08	68.88 ± 0.09	76.60 [68.88,81.95]
Ours	**96.35*** ± **0.05**	**99.60*** ± **0.007**	**97.35*** ± **0.03**	**99.00*** ± **0.03**	**98.98*** ± **0.03**	**98.25*** [**96.35**,**99.60**]	**89.03** ± **0.03**	**97.81*** ± **0.01**	**97.76** ± **0.01**	**94.87*** [**89.03**,**97.81**]

Table 4. Performance of Feature Selection Methods. Average performance obtained with the first 10, 50, 100, 150, and 200 features. The final results are expressed as mean Average Precision (mAP) and their standard deviation. Furthermore, "*" indicates the top performance.

butions with unknown, but equal, variances is the *Lilliefors* test [9]. Each accuracy reported in Tab. 4 comes from the average of the accuracies obtained from a series of SVM classifications over 20 different splits of the data for 5 different subsets of features (i.e., a total of 100 different tests for each method). Thus, given the distribution of these accuracies for the proposed method d_p, and the ones of the i^{th} competitor d_{c_i}, a *two-sample t-test* has been applied obtaining a test decision for the *null hypothesis* H_0 that all the data come from independent random samples from normal distributions. As for the object recognition task (see Tab. 3), we consider as d_p the distribution of accuracies obtained over the 20 classes, and then we compare this distribution against the ones of all the other methods d_{c_i}. From each t-test we consider the probability (p-value) at which the null hypothesis H_0 can be rejected. Based on this result, we assess the validity of the reported results by the *binomial cumulative distribution* function [3, 9]. We consider $N = 10$ independent experiments (i.e., one for each dataset) with exactly two possible outcomes: success and failure. Success when the ILFS outperforms all the other methods with a certain probability to do it by chance p. From Tab. 4 and Tab. 3 we observe $k = 7$ successes where p is given by the exact p-value at which H_0 can be rejected. Since our approach is tested 10 times in the experiments and has p of probability of outperforming the competitors by chance, then the probability of ILFS outperforming more than k times by chance is $4.82 \cdot 10^{-3}$. In conclusion, our approach achieved top performance across many different datasets and difficulties.

5. Conclusion

In this paper we proposed a probabilistic feature selection algorithm that performs the ranking step by considering all the possible subsets of features bypassing the combinatorial problem. The most appealing characteristic of the ILFS is that it aims to model the features "relevancy" using PLSA-inspired process. The derived mixing weights $P(z|f)$ are used to weight a graph of features. The weighted graph, serves to perform the ranking step providing a score of importance for each feature as a function of the importance of its neighbors. Our approach overcomes all the methods in comparison in terms of robustness and ranking quality in a statistically significant extent, attaining the highest performance levels across all the challenging scenarios and difficulties. This study also points to many future directions. From a methodological perspective, the investigation of the absorbing Markov chains has every opportunity to reveal a criterion to perform the *subset selection* step automatically. Results of our work can possibly be improved by performing a validation over multiple $\mathcal{T}$ intervals. As for the applications, we hope that this work motivates researchers to take into account the use of FS as an integral part of future computer vision systems. Finally, for the sake of repeatability, the source code is available at `https://goo.gl/uTuZhc` to provide the material needed to replicate our experiments.

Acknowledgements

This work is supported in part by EPSRC under grants EP/N035305/1 and EP/M025055/1.

References

[1] Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell*, 1(2):203–209, 2002. 2, 6

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

[3] T. Anderson. Multivariate statistical analysis. *VVi11ey and Sons, New York, NY*, 1984. 7, 8

[4] E. Bergshoeff. Ten physical applications of spectral zeta functions. *CQG*, 13(7), 1996. 5

[5] J. Bins and B. A. Draper. Feature selection from huge feature sets. In *Conf. IEEE International Conference on Computer Vision*, volume 2, pages 159–165 vol.2, 2001. 1

[6] S. P. Borgatti and M. G. Everett. A Graph-theoretic perspective on centrality. *Social Networks*, 28(4):466–484, 2006. 4

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

[8] D. Cai, C. Zhang, and X. He. Unsupervised feature selection for multi-cluster data. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 333–342, 2010. 2, 6

[9] W. J. Conover and W. J. Conover. Practical nonparametric statistics. 1980. 8

[10] G. Dorko and C. Schmid. Selection of scale-invariant parts for object class recognition. In *Conf. IEEE International Conference on Computer Vision*, pages 634–639, 2003. 1

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

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

[13] Z.-G. Fan and B.-L. Lu. Fast recognition of multi-view faces with feature selection. In *Conf. IEEE International Conference on Computer Vision*, volume 1, pages 76–81, 2005. 1

[14] T. R. e. a. Golub. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, 1999. 2, 6

[15] 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, 6

[16] R. L. Graham, D. E. Knuth, and O. Patashnik. *Concrete Mathematics: A Foundation for Computer Science*. Addison-Wesley, 1994. 5

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

[18] 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. 2, 6

[19] 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. 2, 6

[20] X. He, D. Cai, and P. Niyogi. Laplacian score for feature selection. In *Advances in Neural Information Processing Systems 18*, 2005. 2, 6

[21] T. Hofmann. Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 289–296. Morgan Kaufmann Publishers Inc., 1999. 1, 4

[22] J. H. Hubbard and B. B. Hubbard, editors. *Vector Calculus, Linear Algebra, and Differential Forms: A Unified Approach (Edition 2)*. Pearson, 2001. 5

[23] H. Liu and H. Motoda. *Computational Methods of Feature Selection*. Chapman and Hall, 2008. 2, 6

[24] X. Liu and T. Yu. Gradient feature selection for online boosting. In *Conf. IEEE International Conference on Computer Vision*, pages 1–8, 2007. 1

[25] K. Matej and et Al. The visual object tracking VOT2016 challenge results. In *Conf. IEEE European Conference on Computer Vision, Workshops*, pages 777–823, 2016. 1

[26] B. Pang and L. Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the ACL*, 2004. 2, 6

[27] G. Roffo. Feature selection library (matlab toolbox). *arXiv preprint arXiv:1607.01327*, 2016. 6

[28] G. Roffo and S. Melzi. Online feature selection for visual tracking. In *Conf. The British Machine Vision Conference (BMVC)*, September 2016. 1

[29] G. Roffo and S. Melzi. *Ranking to Learn*, pages 19–35. Springer International Publishing, Cham, 2017. 6

[30] G. Roffo, S. Melzi, and M. Cristani. Infinite feature selection. In *Conf. IEEE International Conference on Computer Vision*, pages 4202–4210, 2015. 1, 2, 3, 6

[31] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 6

[32] U., Alon et Al. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. In *PNAS*, volume 96. 1999. 2, 6

[33] K. Wu, C. Soci, P. P. Shum, and N. I. Zheludev. Computing matrix inversion with optical networks. *Opt. Express*, 22(1):295–304, 2014. 6

[34] Y. Yang, H. T. Shen, Z. Ma, and et Al. L2,1-norm regularized discriminative feature selection for unsupervised learning. In *Conf. International Joint Conference on Artificial Intelligence*, pages 1589–1594, 2011. 2, 6

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

[36] H. Zeng and Y.-m. Cheung. Feature selection and kernel learning for local learning-based clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1532–1547, 2011. 2, 6