Features Selection via Eigenvector Centrality

Giorgio Roffo and Simone Melzi

Department of Computer Science, University of Verona, Italy
{giorgio.roffo,simone.melzi}@univr.it

Abstract. In an era where accumulating data is easy and storing it inexpensive, feature selection plays a central role in helping to reduce the high-dimensionality of huge amounts of otherwise meaningless data. In this paper, we propose a graph-based method for feature selection that ranks features by identifying the most important ones into arbitrary set of cues. Mapping the problem on an affinity graph - where features are the nodes - the solution is given by assessing the importance of nodes through some indicators of centrality, in particular, the *Eigenvector Centrality (EC)*. The gist of EC is to estimate the importance of a feature as a function of the importance of its neighbors. Ranking central nodes individuates candidate features, which turn out to be effective from a classification point of view, as proved by a thoroughly experimental section. Our approach has been tested on 7 diverse datasets from recent literature (e.g., biological data, object recognition, among others), and compared against filter, embedded, and wrappers methods. The results are remarkable in terms of accuracy, stability and low execution time.

Keywords: Feature Selection, Ranking, High Dimensionality, Data Mining

1 Introduction

As data collection technologies advance and computer power grows, a torrent of data is generated in almost every field computers are used [4]. Because the volume, velocity, variety and complexity of datasets is continuously increasing, pattern recognition methodologies have become indispensable in order to extract useful information from huge amounts of otherwise meaningless data.

Feature Selection (FS) is one of the long existing methods that deal with these problems [13]. Its objective is to select a minimal subset of those attributes that allow a problem to be clearly defined. By choosing a minimal subset of features, irrelevant and redundant features are removed according to some reasonable criteria so that the original task can be achieved equally well, if not better. FS techniques can be partitioned into three classes [13]: *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 [7]; 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) [6,11,16,22,31]; in other

cases, only the selection is carried out [12]. Usually, the subset selection is supervised, while in the ranking case, methods can be supervised or not. FS is *NP-hard* [13]; if there are n features in total, the goal is to select the optimal subset of $m \ll n$, by evaluating $\binom{n}{m}$ combinations; therefore, suboptimal search strategies are considered (see Section 2). With the filters, features are first considered individually, ranked, and then a subset is extracted, some examples are Mutual Information [31], Relief-F [22], and mRMR [24]. Conversely, with wrapper and embedded methods, subsets of features are sampled, evaluated, and finally kept as the final output, for instance, FSV [6,11] and SVM-RFE [16].

In this work, we propose a novel graph-based feature selection algorithm that ranks features according to a graph centrality measure (Eigenvector centrality [5]). The main idea behind the method is to map the problem to an affinity graph, and to model pairwise relationships among feature distributions by weighting the edges connecting them.

The novelty of the proposed method 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 other features mapped as nodes on the graph, bypassing the combinatorial problem in a methodologically sound fashion. Indeed, eigenvector centrality differs from other measurements (e.g., degree centrality) since a node - feature - receiving many links does not necessarily have a high eigenvector centrality. The reason is that not all nodes are equivalent, some are more relevant than others, and, reasonably, endorsements from important nodes count more (see Section 3.2). Noteworthy, another important contribution of this work is the scalability of the method. Indeed, centrality measurements can be implemented using the Map Reduce paradigm [18,21,30], which makes the algorithm prone to a possible distributed version [26].

Our approach is extensively tested on 7 benchmarks of cancer classification and prediction on genetic data (*Colon* [2], *Prostate* [10], *Leukemia* [10],*Lymphoma* [10]), handwritten recognition (GINA [1]), generic feature selection (MADELON [14]), and object recognition (PASCAL VOC 2007 [8]). We compare the proposed method on these datasets, against seven comparative approaches, under different conditions (number of features selected and number of training samples considered), overcoming all of them in terms of ranking stability and classification accuracy.

The paper is organized as follows. A brief overview of the related literature is given in Section 2, mostly focusing on the comparative approaches we consider in this work. Our feature selection algorithm is described in Section 3. Graph Construction and weighting are presented in Section 3.1 and Section 3.2 respectively, while the employed Eigenvector centrality is discussed in Section 3.3. Section 4 contains the experimental evaluations and results. Finally, conclusions are provided in Section 5.

2 Related Literature

Among the most used FS strategies, *Relief-F* [22] 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 (RFE) [16], 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.

An effective yet fast filter method is the *Fisher* method [12], it computes a score for a feature as the ratio of interclass separation and intraclass variance, where features are evaluated independently, and the final feature selection occurs by aggregating the m top ranked ones. Other widely used filters are based on mutual information, dubbed *MI* here [31], 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; Even in the last case, features are evaluated independently, and the final feature selection occurs by aggregating the m top ranked ones.

Selecting features in unsupervised learning scenarios is a much harder problem, due to the absence of class labels that would guide the search for relevant information. In this scenario, we compare our approach against the recent unsupervised graph-based filter dubbed Inf-FS [27]. 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. It assigns a score of "importance" to each feature by taking into account all the possible feature subsets as paths on a graph. Another unsupervised method is the Laplacian Score (LS) [17], 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.

Finally, for the wrapper method, we include the *feature selection via concave minimization* (*FSV*) [6], where the selection process is injected *into* the training of an SVM by a linear programming technique.

3 Proposed Method

3.1 Building the Graph

Given a set of features $X = \{x^{(1)}, ..., x^{(n)}\}$ we build an undirected graph $G = (V, E)$; where V is the set of vertices corresponding, one by one, to each variable x. E codifies (weighted) edges among features. Let the adjacency matrix A associated with G define the nature of the weighted edges: each element a_{ij} of A, $1 \leq i, j \leq n$, represents a pairwise potential term. Potentials can be represented as a binary function $\varphi(x^{(i)}, x^{(j)})$ of the nodes $x^{(k)}$ such as:

$$a_{ij} = \varphi(x^{(i)}, x^{(j)}). \tag{1}$$

The graph can be weighted according to different heuristics, therefore the function φ can be handcrafted or automatically learned from data.

3.2 φ-Design

The design of the φ function is a crucial operation. In this work, we weight the graph according to good reasonable criteria, related to class separation, so as to address the classification problem. In other words, we want to rank features according to how well

they discriminate between two classes. Hence, we draw upon best-practice in FS and propose an ensemble of two different measures capturing both relevance (supervised) and redundancy (unsupervised) proposing a kernelized-based adjacency matrix. Before continuing with the discussion, note that each feature distribution $x^{(i)}$ is normalized so as to sum to 1.

Firstly, we apply the Fisher criterion:

$$f_i = \frac{|\mu_{i,1} - \mu_{i,2}|^2}{\sigma_{i,1}^2 + \sigma_{i,2}^2},$$

where $\mu_{i,C}$ and $\sigma_{i,C}$ are the mean and standard deviation, respectively, assumed by the i-th feature when considering the samples of the *C*-th class. The higher f_i, the more discriminative the i-th feature.

Because we are given class labels, it is natural that we want to keep only the features that are related to or lead to these classes. Therefore, we use mutual information to obtain a good feature ranking that score high features highly predictive of the class.

$$m_i = \sum_{y \in Y} \sum_{z \in x^{(i)}} p(z,y) log\Big(\frac{p(z,y)}{p(z)p(y)}\Big),$$

where Y is the set of class labels, and $p(\cdot,\cdot)$ the joint probability distribution.

A kernel k is then obtained by the matrix product

$$k = (f \cdot m^\top),$$

where f and m are $n \times 1$ column vectors normalized in the range 0 to 1, and k results in a $n \times n$ matrix.

To boost the performance, we introduce a second feature-evaluation metric based on standard deviation [16] – capturing the amount of variation or dispersion of features from average – as follows:

$$\Sigma(i,j) = max\left(\sigma^{(i)}, \sigma^{(j)}\right),$$

where σ being the standard deviation over the samples of x, and Σ turns out to be a $n \times n$ matrix with values $\in$ [0,1].

Finally, the adjacency matrix A of the graph G is given by

$$A = \alpha k + (1-\alpha)\Sigma, \tag{2}$$

where α is a loading coefficient $\in [0,1]$. The generic entry a_{ij} accounts for how much discriminative are the feature i and j when they are jointly considered; at the same time, a_{ij} can be considered as a weight of the edge connecting the nodes i and j of a graph, where the i-th node models the i-th feature distribution.

3.3 Eigenvector Centrality

From a graph theory perspective identifying the most important nodes corresponds to individuate some indicators of centrality within a graph (e.g., the relative importance of

nodes). A first way used in graph theory is to study accessibility of nodes, see [9,25] for example. The idea is to compute A^l for some suitably large l (often the diameter of the graph), and then use the row sums of its entries as a measure of accessibility (i.e. $scores(i) = [A^l\mathbf{e}]_i$, where $\mathbf{e}$ is a vector with all entries equal to 1). The accessibility index of node i would thus be the sum of the entries in the i-th row of A^l, and this is the total number of paths of length l (allowing stopovers) from node i to all nodes in the graph. One problem with this method is that the integer l seems arbitrary. However, as we count longer and longer paths, this measure of accessibility converges to a index known as eigenvector centrality measure (EC) [5].

The basic idea behind the EC is to calculate v_0 the eigenvector of A associated to the largest eigenvalue. Its values are representative of how strongly each node is connected to the other nodes. Since the limit of A^l as l approaches a large positive number L converges to v_0,

$$\lim_{l \to L}[A^l\mathbf{e}] = v_0, \tag{3}$$

the EC index makes the estimation of indicators of centrality free of manual tuning over l, and computationally efficient.

Let us consider a vector, for example $\mathbf{e}$, that is *not* orthogonal to the principal vector v_0 of A. It is always possible to decompose $\mathbf{e}$ using the eigenvectors as basis with a coefficient $\beta_0 \neq 0$ for v_0. Hence:

$$\mathbf{e} = \beta_0 v_0 + \beta_1 v_1 + \ldots + \beta_n v_n, \quad (\beta_0 \neq 0). \tag{4}$$

Then

$$\begin{aligned} A\mathbf{e} = A(\beta_0 v_0 + \beta_1 v_1 + \ldots + \beta_n v_n) = \beta_0 A v_0 + \beta_1 A v_1 + \ldots + \beta_n A v_n = \\ = \beta_0 \lambda_0 v_0 + \beta_1 \lambda_1 v_1 + \ldots + \beta_n \lambda_n v_n. \end{aligned} \tag{5}$$

So in the same way:

$$\begin{aligned} A^l\mathbf{e} = A^l(\beta_0 v_0 + \beta_1 v_1 + \ldots + \beta_n v_n) = \beta_0 A^l v_0 + \beta_1 A^l v_1 + \ldots + \beta_n A^l v_n = \\ = \beta_0 \lambda_0^l v_0 + \beta_1 \lambda_1^l v_1 + \ldots + \beta_n \lambda_n^l v_n, \quad (\beta_0 \neq 0). \end{aligned} \tag{6}$$

Finally we divide by the constant $\lambda_0^l \neq 0$ (see Perron-Frobenius theorem [23]),

$$\frac{A^l\mathbf{e}}{\lambda_0^l} = \beta_0 v_0 + \frac{\lambda_1^l \beta_1 v_1}{\lambda_0^l} + \ldots + \frac{\lambda_n^l \beta_n v_n}{\lambda_0^l}, \quad (\beta_0 \neq 0). \tag{7}$$

The limit of $\frac{A^l\mathbf{e}}{\lambda_0^l}$ as l approaches infinity equals $\beta_0 v_0$ since $\lim_{l\to\infty} \frac{\lambda_1^l}{\lambda_0^l} = 0$, $\forall l > 0$. What we see here is that as we let l increase, the ratio of the components of $A^l\mathbf{e}$ converges to v_0. Therefore, marginalizing over the columns of A^l, with a sufficiently large l, corresponds to calculate the principal eigenvector of matrix A [5]. Figure 1 illustrates a toy example of three random planar graphs. Graphs are made of 700 nodes and they are weighted by the Euclidean distance between each pair of points. In the example, high scoring nodes are those ones farther from the mean (i.e., the distance is conceived as quantity to maximize), the peculiarity of the eigenvector centrality is that a node is important if it is linked to by other important nodes (higher scores).

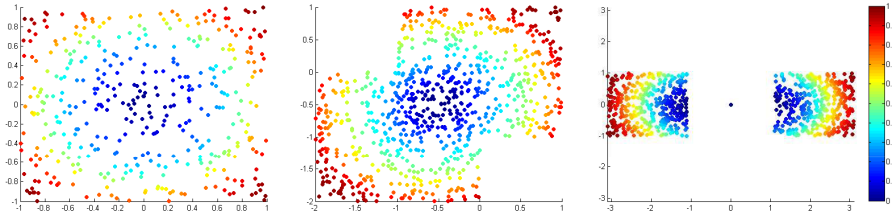

Fig. 1. Eigenvector centrality plots for three random planar graphs. On the left, a simple Gaussian distribution where central nodes are at the peripheral part of the distribution as expected. The central and right plots, some more complicated distributions, a node receiving many links does not necessarily have a high eigenvector centrality.

Name	# samples	# classes	# feat.	*few train*	*unbal. (+/-)*	*overlap*	*noise*	*shift*
GINA [1]	3153	2	970			X		
MADELON [15]	4.4K	2	500			X		
Colon [2]	62	2	2K	X	(40/22)		X	
Lymphoma [10]	45	2	4026	X	(23/22)			
Prostate [29]	102	2	6034	X	(50/52)			
Leukemia [10]	72	2	7129	X	(47/25)		X	X
VOC 2007 [8]	10K	20	n.s.		X		X	

Table 1. Panorama of the used datasets, together with the challenges for the FS 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).

To the aim of this work, the use of eigenvector centrality allows to individuate candidate features, which turn out to be effective from a classification point of view, since indicators of centrality characterize the *global* (as opposed to local) prominence of a feature in the graph. Summarizing, the gist of eigenvector centrality is to compute the centrality of a node as a function of the centralities of its neighbors.

4 Experiments and Results

4.1 Datasets and Comparative Approaches

Datasets are chosen for letting the proposed method deal with diverse FS 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, artefacts 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 2 lists the methods in comparison, whose details can be found in Sec. 2. Here we just note their *type*, that is, f = filters, w = wrappers, e = embedded methods, and

Acronym	Type	Cl.	Compl.
Fisher [12]	f	s	$\mathcal{O}(Tn)$
FSV [6,11]	w	s	N/A
Inf-FS [27]	f	u	$\mathcal{O}(n^{2.37}(1+T))$
MI [31]	f	s	$\sim \mathcal{O}(n^2T^2)$
LS [17]	f	u	N/A
Relief-F [22]	f	s	$\mathcal{O}(iTnC)$
RFE [16]	e	s	$\mathcal{O}(T^2nlog_2n)$
Ours	f	s	$\mathcal{O}(Tn+n^2)$

Table 2. List of the FS approaches considered in the experiments, specified according to their *Type*, class (*Cl.*), and complexity (*Compl.*). 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. N/A indicates that the computational complexity is not specified in the reference paper.

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). The computational complexity of our approach is $O(Tn + n^2)$. The term Tn is due to the computation of the mean values among the T samples of every feature (n). The n^2 concerns the construction of the matrix A. As for the computation of the leading eigenvector, it costs $O(m^2n)$, where m is a number much smaller than n that is selected within the algorithm [20]. In the case that the algorithm can not be executed on a single computer, we refer the reader to [18,21,26,30] for distributed algorithms.

4.2 Exp. 1: Deep Representation (CNN) with pre-training

This section proposes a set of tests on the PASCAL VOC-2007 [8] dataset. In object recognition VOC-2007 is a suitable tool for testing models, therefore, we use it as reference benchmark to assess the strengths and weaknesses of using our approach regarding the classification task. For this reason, we compare our approach against 8 state-of-the-art FS methods reported in Table 2. This experiment considers as features the cues extracted with a deep convolutional neural network architecture (CNN). We selected the pre-trained model called very deep ConvNets [28], which performs 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). The VOC-2007 edition contains about 10,000 images split into train, validation, and test sets, and labeled with twenty object classes. A one-vs-rest SVM classifier for each class is learnt (where cross-validation is used to find the best parameter C and α mixing coefficient in Eq. 2 on the training data) and evaluated independently and the performance is measured as mean Average Precision (mAP) across all classes.

Table 3 serves to analyze and empirically clarify how well important features are ranked high by several FS algorithms. The amount of features used for the two experiments is very low: $\approx$3% and $\approx$6% of the total. The results are significant: our method achieved the best performance in terms of mean average precision (mAP) followed by the unsupervised filter methods LS and Inf-FS. As for the methods in comparison, one

PASCAL VOC 2007																
	First 128/4096 Features Selected								First 256/4096 Features Selected							
	Fisher	FSV	Inf-FS	LS	MI	ReliefF	RFE	**Ours**	Fisher	FSV	Inf-FS	LS	MI	ReliefF	RFE	**Ours**
	52.43	87.90	88.96	**89.37**	12.84	57.20	86.42	88.09	82.65	90.22	**91.16**	90.94	73.51	81.67	88.17	90.79
	13.49	80.74	80.43	80.56	13.49	49.10	**82.14**	80.94	83.21	80.07	83.36	84.21	75.04	71.27	83.30	**84.72**
	85.46	86.77	87.04	86.96	80.91	75.42	83.16	**88.74**	89.14	86.15	88.88	**89.31**	85.48	83.54	86.12	89.15
	79.04	83.58	85.31	83.51	61.50	63.75	78.55	**86.90**	87.05	80.68	87.24	**87.84**	75.25	73.30	86.13	87.42
	46.61	39.80	44.83	**49.36**	35.39	18.33	46.24	47.37	52.54	49.00	52.65	49.44	48.94	35.67	47.28	**53.20**
	12.29	72.89	76.69	**76.98**	12.29	31.54	74.68	76.27	77.32	78.69	79.23	79.97	59.23	63.83	79.38	**80.57**
	82.09	78.61	85.78	85.82	63.58	74.95	83.94	**85.92**	85.86	84.01	86.74	**87.06**	85.27	82.76	85.61	86.56
	75.29	82.25	**83.34**	81.81	40.96	66.95	81.02	83.29	83.46	83.49	**85.61**	84.98	79.16	76.78	84.50	85.57
	54.81	52.37	58.62	60.07	16.95	29.07	59.84	**60.57**	63.14	62.54	63.93	64.23	63.20	48.19	62.16	**64.53**
	47.98	61.68	59.23	**65.50**	11.42	11.42	62.96	60.55	66.51	70.18	67.96	**71.54**	22.96	51.28	64.20	69.71
	49.68	63.50	67.69	63.86	12.62	12.62	67.05	**67.70**	68.42	69.27	**71.78**	71.01	65.77	52.24	71.43	70.95
	81.06	80.57	83.16	**83.21**	70.70	68.12	80.07	83.00	84.24	84.15	85.08	**85.20**	82.03	74.85	83.52	**85.20**
	74.91	**83.33**	81.23	81.75	14.13	63.06	81.55	82.79	**85.68**	83.13	85.28	85.41	71.36	75.53	83.47	85.28
	13.18	71.42	81.32	80.24	13.18	34.43	76.57	**82.20**	**84.29**	81.16	84.20	83.81	81.01	70.68	82.97	84.12
	91.33	90.03	89.10	89.33	91.08	88.85	89.03	91.27	91.95	89.99	90.65	90.64	91.77	90.38	90.64	**91.99**
	47.89	39.40	45.38	47.94	13.23	13.30	48.61	**49.05**	54.94	47.95	53.86	54.31	48.98	34.74	50.18	**55.88**
	10.87	68.82	73.35	**74.05**	10.87	10.87	66.86	73.80	73.43	75.84	79.01	**81.57**	10.87	11.73	75.47	78.85
	45.87	56.08	58.94	58.92	13.30	13.31	**62.06**	61.32	66.46	59.77	63.07	63.92	58.78	44.74	**66.68**	64.86
	63.51	88.52	91.42	**91.48**	58.62	73.32	88.46	91.30	84.05	90.61	**93.21**	93.16	81.33	82.93	90.24	92.31
	64.29	65.61	66.79	62.99	47.25	24.96	67.10	**67.30**	71.44	69.19	70.56	70.75	71.39	55.59	**73.17**	72.49
mAP	54.60	71.69	74.43	74.69	34.72	44.03	73.32	**75.42**	76.79	75.80	78.17	78.47	66.57	63.09	76.73	**78.71**

Table 3. Varying the cardinality of the selected features. The image classification results achieved in terms of average precision (AP) scores while selecting the first 128 (3%) and 256 (6%) features from the total $4,096$.

can observe the high variability in classification accuracy; indeed, results show that our method is robust to classes (i.e., by changing the testing class its performance is always comparable with the top scoring method).

4.3 Exp. 2: Testing on Microarray Databases

In application fields like biology is inconceivable to devise an analysis procedure which does not comprise a FS step. A clear example can be found in the analysis of expression microarray data, where the expression level of thousands of genes is simultaneously measured. Within this scenario, we tested the proposed approach on four well-known microarray benchmark datasets for two-class problems. Results are reported in Table 4. The testing protocol adopted in this experiment consists in splitting the dataset up to 2/3 for training and 1/3 for testing. In order to have a fair evaluation, the feature ranking has been calculated using only the training samples, and then applied to the testing samples. The classification is performed using a linear SVM. For setting the best parameters (C of the linear SVM, and α mixing coefficient) we used a 5-fold cross validation on the training data. This procedure is repeated several times and results are averaged over the trials. Results are reported in terms of the Receiver Operating Characteristic or ROC curves. A widely used measurement that summarizes the ROC curve is the Area Under the ROC Curve (AUC) [3] which is useful for comparing algorithms independently of

Microarray Databases														
	COLON							LEUKEMIA						
	# Features							# Features						
Method	10	50	100	150	200	**Average**	**Time**	10	50	100	150	200	**Average**	**Time**
Fisher	**91.00**	91.20	89.20	89.90	90.20	90.30	ϵ	99.32	99.62	99.70	99.62	99.62	99.57	ϵ
FSV	83.80	86.10	86.80	87.10	86.80	86.12	0.11	98.48	99.47	99.62	99.62	99.70	99.38	0.23
Inf-FS	74.50	86.10	89.90	89.10	90.20	85.96	0.91	99.17	99.92	99.62	**99.92**	**99.85**	99.70	5.49
LS	67.00	81.60	83.10	84.70	86.10	80.50	0.03	99.55	99.70	**99.77**	**99.92**	99.77	99.74	0.07
MI	90.80	**92.70**	91.30	91.20	91.40	91.28	0.31	92.73	99.47	99.54	99.62	99.55	98.18	057
ReliefF	71.60	86.80	89.50	90.10	90.10	85.62	0.52	96.67	99.62	99.32	99.55	99.55	98.94	1.09
RFE	87.20	84.90	86.00	86.50	87.60	86.44	0.08	54.39	89.24	95.45	95.45	97.57	86.42	0.02
Ours	89.40	92.00	**92.10**	**91.70**	**92.40**	**91.52**	0.45	**99.85**	**99.92**	**99.77**	**99.92**	**99.85**	**99.86**	1.50
	LYMPHOMA							**PROSTATE**						
	# Features							# Features						
Method	10	50	100	150	200	**Average**	**Time**	10	50	100	150	200	**Average**	**Time**
Fisher	**92.60**	99.20	98.80	98.80	**99.20**	97.72	0.01	95.00	95.00	94.94	95.64	95.90	95.30	0.02
FSV	89.00	96.20	97.19	98.20	97.60	95.64	0.18	96.19	95.67	95.64	95.71	95.26	95.69	0.63
Inf-FS	82.40	96.40	97.80	99.20	98.20	94.80	7.61	73.49	92.24	94.17	95.77	94.87	90.11	26.85
LS	58.00	87.60	93.00	97.60	96.20	86.48	0.04	64.97	88.62	93.62	**96.02**	**96.57**	87.96	0.24
MI	91.20	97.20	98.80	98.80	99.00	97.00	0.59	94.01	95.61	95.29	94.68	94.94	94.90	1.01
ReliefF	89.80	98.80	99.00	98.80	98.80	97.04	0.74	93.56	92.72	93.46	93.62	93.85	93.44	2.68
RFE	89.20	96.00	98.00	98.80	99.00	96.20	0.02	63.72	75.67	79.87	86.70	88.72	78.94	0.3
Ours	91.80	**99.40**	**99.20**	**99.60**	**99.20**	**97.84**	1.50	**96.32**	**96.28**	**96.28**	95.80	96.32	**96.20**	2.81

	Overall Performance on Microarray Database							
	Fisher	FSV	Inf-FS	LS	MI	ReliefF	RFE	**Ours**
10	**94.48**	91.87	82.39	72.38	92.18	87.91	73.63	94.34
50	96.26	94.36	93.67	89.38	96.24	94.49	86.45	**96.90**
100	95.66	94.82	95.37	92.37	96.23	95.32	89.83	**96.84**
150	95.99	95.16	96.00	94.56	96.07	95.52	91.86	**96.76**
200	96.23	94.84	95.78	94.66	95.97	95.57	93.22	**96.94**
Mean Time	0.01	0.29	10.22	0.10	0.63	1.26	0.04	1.57
Overall Avg.	95.72	94.21	92.64	88.67	95.34	93.76	87.00	**96.36**

Table 4. The tables show results obtained on the expression microarray scenario. Tests have been repeated 100 times, and the means of the computed AUCs are reported for each dataset. We indicate with ϵ each instance where the approach completed the task in less than 0.01 secs.

application. Hence, classification results for the datasets used show that the proposed approach produces superior results in all the cases. The overall performance (at the bottom of Table 4) indicates that our approach is more robust than the others, by changing the data it still produces high quality rankings. We assessed the stability of the selected features using the Kuncheva index [19]. This stability measure represents the similarity between the set of rankings generated over the different splits of the dataset. The similarity between sequences of size N can be seen as the number of elements n they have in common (i.e. the size of their intersection). The Kuncheva index takes values in [-1, 1], and the higher its value, the larger the number of commonly selected features in both sequences. The index is shown in Figure 2, comparing our approach and the other methods. The proposed method shows, in most of the cases, a high stability whereas the highest performance is achieved.

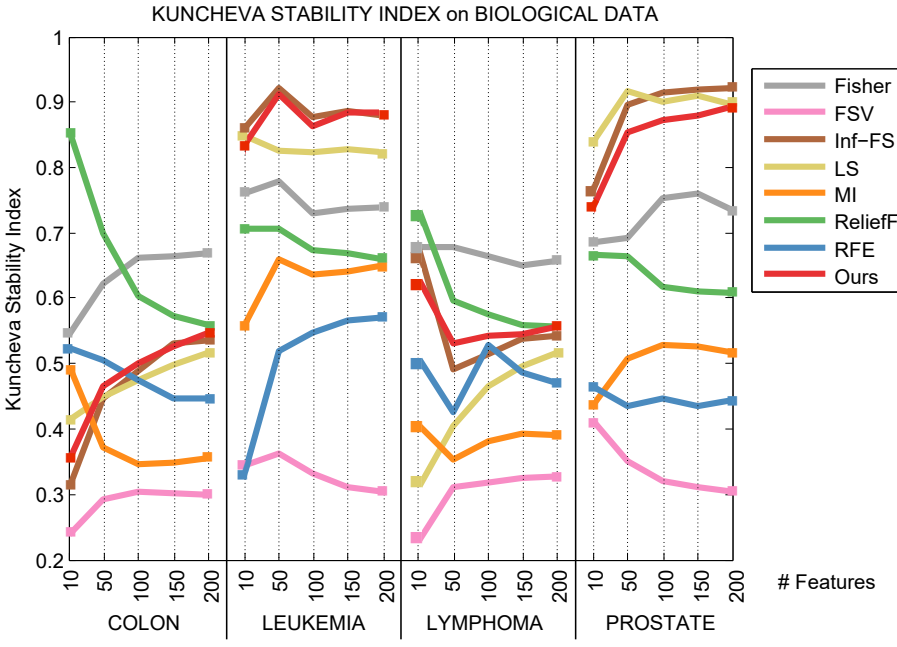

Fig. 2. The Kuncheva stability indices for each method in comparison are presented. The figure reports the stability while varying the cardinality of the selected features from 10 to 200 on different benchmarks.

4.4 Exp. 2: Other Benchmarks

GINA has sparse input variables consisting of 970 features. It is a balanced data set with 49.2% instances belonging to the positive class. Results obtained on GINA indicate that the proposed approach overcomes the methods in comparison, and select the most useful features from a data set with high-complexity and dimensionality. MADELON is an artificial dataset, which was part of the NIPS 2003 feature selection challenge. It represents a two-class classification problem with continuous input variables. The difficulty is that the problem is multivariate and highly non-linear. Results are reported in Table 5. This gives a proof about the classification performance of our approach that is attained on the test sets of GINA and MADELON.

FS techniques definitely represent an important class of preprocessing tools, by eliminating uninformative features and strongly reducing the dimension of the problem space, it allows to achieve high performance, useful for practical purposes in those domains where high speed is required.

5 Conclusion

In this paper we present the idea of solving feature selection via the Eigenvector centrality measure. We design a graph – where features are the nodes – weighted by a kernelized adjacency matrix, which draws upon best-practice in feature selection while assigning scores according to how well features discriminate between classes. The method estimates some indicators of centrality identifying the most important features within

FS Challenge Datasets															
	GINA - Handwritten Recognition							MADELON - Artificial Data						
	# Features							# Features						
Method	10	50	100	150	200	**Average**	**Time**	10	50	100	150	200	**Average**	**Time**
Fisher	90.0	89.8	89.4	90.2	**90.4**	89.9	0.05	61.7	61.9	63.0	62.3	64.0	62.5	0.02
FSV	82.4	81.9	83.7	82.0	83.6	82.7	138	61.1	59.9	60.6	61.0	61.0	60.7	732
Inf-FS	77.6	77.9	76.3	77.3	76.9	77.2	0.12	63.8	62.9	63.1	63.2	**64.9**	63.5	0.04
LS	82.3	82.2	82.4	83.4	83.2	82.7	1.30	63.7	62.8	62.9	63.3	64.7	63.4	8.13
MI	89.5	89.3	89.7	89.8	90.1	89.6	1.13	63.5	63.0	63.7	63.5	64.7	63.6	0.4
ReliefF	89.1	89.0	88.7	89.1	89.0	88.9	41	63.4	62.6	**63.8**	**65.4**	60.8	63.2	10.22
RFE	82.3	82.2	82.4	83.4	83.2	82.7	6.60	48.9	55.0	61.2	57.1	60.2	56.5	50163
Ours	**90.7**	**90.9**	**90.3**	**90.4**	89.5	**90.3**	1.56	**64.4**	**63.6**	**63.8**	63.7	63.3	**63.7**	0.57

Table 5. Varying the cardinality of the selected features. (ROC) AUC (%) on different datasets by SVM classification. Performance obtained with the first 10, 50, 100, 150, and 200 features.

the graph. The results are remarkable: the proposed method has been extensively tested on 7 different datasets selected from different scenarios (i.e., object recognition, handwritten recognition, biological data, and synthetic testing datasets), in all the cases we achieve top performances against 7 competitors selected from recent literature in feature selection. Our approach is also robust and stable on different splits of the training data, it performs effectively in ranking high the most relevant features, and it has a very competitive complexity. This study also points to many future directions; focusing on the investigation of different implementations for parallel computing for big data analysis or focusing on the investigation of different relations among the features. Finally, for the sake of repeatability, the source code will be posted on-line to provide the material needed to replicate our experiments.

References

1. GINA digit recognition database. IEEE Conf. International Joint Conference on Neural Networks (2007)
2. Alon, U. and Barkai, N. and Notterman, D.A. and Gish, K. and Ybarra, S. and Mack, D. and Levine, A.J.: Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. In: PNAS (1999)
3. Bamber, D.: The area above the ordinal dominance graph and the area below the receiver operating characteristic graph. Journal of Mathematical Psychology (1975)
4. Bólon-Canedo, V., Sánchez-Maroo, N., Alonso-Betanzos, A.: Recent advances and emerging challenges of feature selection in the context of big data. Knowledge-Based Systems (2015)
5. Bonacich, P.: Power and centrality: A family of measures. American Journal of Sociology (1987)
6. Bradley, P.S., Mangasarian, O.L.: Feature selection via concave minimization and support vector machines. In: Conf. International Conference on Machine Learning (ICML) (1998)
7. Duch, W., Wieczorek, T., Biesiada, J., Blachnik, M.: Comparison of feature ranking methods based on information entropy. In: IJCNN. vol. 2. IEEE (2004)
8. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results
9. Garrison, W.L.: Connectivity of the interstate highway system. Papers in Regional Science 6(1), 121–137 (1960)

10. Golub, T.R.: Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science (1999)
11. Grinblat, G.L., Izetta, J., Granitto, P.M.: SVM based feature selection: Why are we using the dual? In: Conf. Ibero-American Conference on AI (2010)
12. Gu, Q., Li, Z., Han, J.: Generalized fisher score for feature selection. Computing Research Repository (CoRR) (2012)
13. Guyon, I.: Feature extraction: foundations and applications, vol. 207. Springer Science & Business Media (2006)
14. Guyon, I., Gunn, S., Ben-Hur, A., Dror, G.: Result analysis of the nips 2003 feature selection challenge. In: NIPS. pp. 545–552 (2004)
15. Guyon, I., Li, J., Mader, T., Pletscher, P.A., 0004, G.S., Uhr, M.: Competitive baseline methods set new standards for the NIPS 2003 feature selection benchmark. PRL (2007)
16. Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. Machine Learning Journal (2002)
17. He, X., Cai, D., Niyogi, P.: Laplacian score for feature selection. In: Advances in Neural Information Processing Systems 18 (2005)
18. Kang, U., Papadimitriou, S., Sun, J., Watson, I.T.J., Tong, H., Watson, I.T.J.: Centralities in large networks: Algorithms and observations
19. Kuncheva, L.I.: A stability index for feature selection. In: Proceedings of the 25th Conference on Proceedings of the 25th IASTED International Multi-Conference: Artificial Intelligence and Applications. pp. 390–395. AIAP'07, ACTA Press, Anaheim, CA, USA (2007)
20. Lehoucq, R.B., Sorensen, D.C., Yang, C.: ARPACK users' guide: solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods, vol. 6. Siam (1998)
21. Lerman, K., Ghosh, R., Kang, J.H.: Centrality metric for dynamic networks. In: Proceedings of the Eighth Workshop on Mining and Learning with Graphs. pp. 70–77. MLG '10, ACM, New York, NY, USA (2010)
22. Liu, H., Motoda, H. (eds.): Computational Methods of Feature Selection. Chapman & Hall Data Mining and Knowledge Discovery Series (2008)
23. Meyer, C.D. (ed.): Matrix Analysis and Applied Linear Algebra. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA (2000)
24. Peng, H., Long, F., Ding, C.: Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) (2005)
25. Pitts, F.R.: A graph theoretic approach to historical geography. The Professional Geographer 17(5), 15–20 (1965)
26. Rawat, A., Saha, S., Ghrera, S.P.: Time efficient ranking system on map reduce framework. In: 2015 Third International Conference on Image Information Processing (ICIIP). pp. 496–501 (2015)
27. Roffo, G., Melzi, S., Cristani, M.: Infinite feature selection. In: IEEE Conf. International Conference on Computer Vision (ICCV) (2015)
28. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR abs/1409.1556 (2014)
29. Singh, D., Febbo, P.G., Ross, K., Jackson, D.G., Manola, J., Ladd, C., Tamayo, P., Renshaw, A.A., D'Amico, A.V., Richie, J.P., Lander, E.S., Loda, M., Kantoff, P.W., Golub, T.R., Sellers, W.R.: Gene expression correlates of clinical prostate cancer behavior. Cancer Cell 1 (2002)
30. Wu, D.D., Deng, X., Li, Y.: Safety and emergency systems engineering mapreduce based betweenness approximation engineering in large scale graph. Systems Engineering Procedia (2012)
31. Zaffalon, M., Hutter, M.: Robust feature selection using distributions of mutual information. In: Conf. International Conference on Uncertainty in Artificial Intelligence (UAI) (2002)