Pattern Recognition Letters 33 (2012) 1794–1804

Contents lists available at SciVerse ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Efficient feature selection filters for high-dimensional data

Artur J. Ferreira [a,c,*], Mário A.T. Figueiredo [b,c]

[a] Instituto Superior de Engenharia de Lisboa, Lisboa, Portugal
[b] Instituto Superior Técnico, Lisboa, Portugal
[c] Instituto de Telecomunicações, Lisboa, Portugal

ARTICLE INFO

Article history:
Received 12 December 2011
Available online 7 June 2012

Communicated by G. Borgefors

Keywords:
Feature selection
Filters
Dispersion measures
Similarity measures
High-dimensional data

ABSTRACT

Feature selection is a central problem in machine learning and pattern recognition. On large datasets (in terms of dimension and/or number of instances), using search-based or wrapper techniques can be computationally prohibitive. Moreover, many filter methods based on relevance/redundancy assessment also take a prohibitively long time on high-dimensional datasets.

In this paper, we propose efficient unsupervised and supervised feature selection/ranking filters for high-dimensional datasets. These methods use low-complexity relevance and redundancy criteria, applicable to supervised, semi-supervised, and unsupervised learning, being able to act as pre-processors for computationally intensive methods to focus their attention on smaller subsets of promising features. The experimental results, with up to 10^5 features, show the time efficiency of our methods, with lower generalization error than state-of-the-art techniques, while being dramatically simpler and faster.

1. Introduction

The need for *feature selection* (FS) often arises in machine learning and pattern recognition problems; FS may improve the accuracy of a classifier learnt from data, avoiding the well-known "curse of dimensionality", and may speed up the training process (Escolano et al., 2009; Guyon et al., 2006; Hastie et al., 2009; Liu and Motoda, 1998).

With high-dimensional data, typically many features are irrelevant and/or redundant for a given learning task, having harmful consequences in terms of performance and/or computational cost. Moreover, a large number of features requires a large amount of memory or storage space. In this context, FS techniques play an important role in reducing the number of features.

Furthermore, in high-dimensional datasets, search-based FS techniques, such as *floating search* (Pudil et al., 1994a,b) and *branch and bound* (BB) (Chen, 2003; Nakariyakul and Casasent, 2007; Somol et al., 2004) may easily become computationally too expensive. The same happens with "wrapper" FS approaches, since they involve repeatedly running some learning algorithm to evaluate feature subsets.

1.1. Learning on high-dimensional data

High-dimensional datasets are increasingly common in learning problems, in many different domains, such as *text categorization* (TC) (Joachims, 1998, 2001), genomics (see Baldi and Hatfield, 2002), econometrics, and computer vision.

In TC, texts may be represented by feature vectors known as *bags-of-words* (BoW), holding the frequency of terms/words (Joachims, 2001; Manning et al., 2008). Usually, we have a large vocabulary of d terms, thus each document is described by a high-dimensional vector. A collection of n documents is represented by the $n \times d$ *term-document* (TD) (Manning et al., 2008) matrix, holding the BoW representation of each document. Several FS techniques have been proposed for TC; see Forman (2003), Hyunsoo et al. (2005), Mohamed et al. (2006), Torkkola (2003), and references therein.

Gene expression microarray data contains the expression levels of a large number (d) of genes in a few (n) samples, and thus a large data matrix (Baldi and Hatfield, 2002). Like in TC, these matrices typically have $d \gg n$. In microarray gene expression data, the quest for learning to detect cancer and other diseases has a high scientific value and has stimulated the creation of several public domain datasets (see Table 1).

1.2. Our contributions

We propose filter-type (Guyon and Elisseeff, 2003) methods for FS, scalable for high-dimensional datasets, with the following key advantages: time-efficiency, since they are independent of the classifier; applicability to supervised, semi-supervised, or unsupervised learning for both binary and multi-class problems. The main disadvantages, as compared to wrapper approaches, are: sub-optimality, since the selected subset of features may not be the best for

* Corresponding author at: Instituto Superior de Engenharia de Lisboa, Lisboa, Portugal.
E-mail address: arturj@cc.isel.ipl.pt (A.J. Ferreira).

0167-8655/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.patrec.2012.05.019

any classifier; the sub-optimality of our efficient redundancy assessment process implies that, in some cases, the number of selected features could be further reduced without significantly penalizing the accuracy. Our methods can also act as a pre-screening stage, followed by a more expensive (possibly a wrapper) method, removing irrelevant and redundant features, commonly found in high-dimensional data (Yu and Liu, 2003; Yu et al., 2004; Peng et al., 2005).

The remaining sections are organized as follows. Section 2 reviews some successful supervised and unsupervised FS techniques. Section 3 details the proposed filters for FS. Section 4 reports the experimental evaluation of our methods against other techniques. Finally, Section 5 presents concluding remarks and hints for future work.

2. Feature selection for high-dimensional data

In this Section, we review some unsupervised and supervised FS techniques that have proven successful for machine learning problems, focusing on their application to high-dimensional datasets. There is a vast literature on FS; for a comprehensive coverage see Escolano et al. (2009), Guyon and Elisseeff (2003), Guyon et al. (2006), and Hastie et al. (2009); see also the special issue jmlr.csail.mit.edu/papers/special/feature03.html.

Let $D = \{(\mathbf{x}_1, c_1), \ldots, (\mathbf{x}_n, c_n)\}$ be a training set where $\mathbf{x}_j \in \mathbb{R}^d$, for $j = 1, \ldots, n$, denotes the feature vector of the jth training sample and c_j is its class label (absent in unsupervised learning). We collect all the feature vectors in a $n \times d$ matrix $\mathbf{X}$, where the jth row contains $\mathbf{x}_j$, the feature vector of the jth training sample, while the ith column, denoted $X_i \in \mathbb{R}^n$, contains the n samples of the ith feature. Finally, we denote as X_{ij} the jth sample of the ith feature.

FS techniques are classically grouped into two types of approaches: filters (Das, 1994; Guyon et al., 2006; Guyon and Elisseeff, 2003) and wrappers (Das, 1994; Kohavi and John, 1997). A wrapper assesses the adequacy of a subset of features based on the performance of some classifier operating with that subset of features and learnt from the training data. In contrast, filters assess the quality of a given subset of features using solely characteristics of that subset, without any learning algorithm. A third type of methods embeds the FS procedure into the learning algorithm, using the ability to ignore a subset of features. Although, filters are the simplest approach, thus expected to perform worse than the other types of methods, it is often the case that they are the only applicable option, on high-dimensional datasets, where wrappers and embedded methods can be too expensive. Below we briefly review these three types of methods.

2.1. Wrappers and embedded methods

2.1.1. Wrappers

Wrappers require some method to search the space of all possible subsets of features, assessing their quality by learning and evaluating a classifier with that feature subset. This is a combinatorial problem, with an objective function that is costly to compute, making wrappers inadequate for high-dimensional datasets. The *floating search* (Pudil et al., 1994a,b) and *branch and bound* (BB) (Chen, 2003; Nakariyakul and Casasent, 2007; Somol et al., 2004) algorithms are two examples of methods used to perform this search.

Liang et al. (2008) proposed an FS algorithm that addresses the problem of the high computational cost and overcomes some drawbacks of suboptimal schemes. Their method is able to find a good feature subset without resorting to exhaustive search.

Casado-Yusta (2009) applies the combinatorial algorithm known as *greedy randomized adaptive search procedure* (GRASP), proposed by Feo and Resende (1989). They concluded that tabu search and GRASP outperformed other techniques. Esseghir et al. (2010) proposed another GRASP-based FS hybrid filter-wrapper method, which is prohibitive for high-dimensional datasets. An improvement over the methods of Casado-Yusta (2009) and Esseghir et al. (2010) was proposed by Bermejo et al. (2011), making fewer wrapper evaluations.

Veenman and Bolck (2011) introduced a sparse model for high-dimensional data, aiming at selecting a small number of features. It uses a linear program, being competitive with state-of-the-art methods. Khushaba et al. (2011) proposed a method based on a differential evolution algorithm. Also recently, Jung et al. (2011) proposed a criterion for estimating the redundancy between feature sets, by the conditional mutual information between the selected and candidate features to each class variable.

2.1.2. Embedded approach

In the embedded approach, FS is intrinsic to the learning algorithm, which simultaneously learns the classifier and chooses a subset of features. These methods typically work by including in the objective function of the learning algorithm a sparsity-inducing regularizer or prior, which encourages the weights assigned to some features to become zero. Examples of such an approach are the *sparse multinomial logistic regression* (SMLR) algorithm proposed by Krishnapuram et al. (2005a) and the *sparse logistic regression* (SLogReg) method of Shevade and Keerthi (2003). Other related methods are the *Bayesian logistic regression* (BLogReg) of Cawley and Talbot (2006), a more adaptive version of SLogReg, and *joint classifier and feature optimization* (JCFO) (Krishnapuram et al., 2005b), which applies FS inside a kernel function.

2.2. Filters

In the past years, some computationally efficient filter FS methods tailored to high-dimensional data have been proposed. In this Subsection, we review some of these unsupervised and supervised methods. In Section 4, we compare our proposals against most of these methods.

2.2.1. Unsupervised

The *term-variance* (TV) criterion (Liu et al., 2005) sorts the features X_i based on their sample variance,

$$\mathrm{TV}_i = \mathrm{var}(X_i) = \frac{1}{n}\sum_{j=1}^{n}(X_{ij} - \overline{X}_i)^2, \tag{1}$$

where $\overline{X}_i$ is the sample mean of feature X_i, $\overline{X}_i = (1/n)\sum_{j=1}^{n} X_{ij}$. Although it is a scale-variant measure, variance has been found effective for TC problems with sparse data (Liu et al., 2005; Ferreira and Figueiredo, 2011), with many features having a high fraction of zeros; in other terms, their so-called "zero-norm", $\ell_0^{(i)}$ (the number of patterns where the feature is not zero), is small.

The *Laplacian score* (LS) (He et al., 2005) assumes that data from the same class is close to each other and the local geometric structure is crucial for discrimination. The relevance of a feature is given by its power of locally preserving it, that is, its LS. Experiments on different tasks show the efficiency of LS, as compared to TV and *Fisher's ratio* (FiR) (see SubSection 2.2.2).

The *extended* LS (LSE) combines LS with a distance-based entropy (Liu et al., 2009). The distance-based entropy replaces the k-means clustering of LS, showing higher stability and efficiency on high-dimensional datasets.

The LS and LSE methods are based on concepts from spectral FS, which identifies relevant features by measuring their capability of preserving sample similarity. Spectral FS provides a framework for supervised and unsupervised FS. Usually, spectral FS algorithms evaluate features individually, being unable to identify redundant features. Zhao et al. (2010b) addressed this limitation by proposing a new spectral FS algorithm to handle feature redundancy, adopting an embedded model.

Zhao and Liu (2007) exploit ideas and properties common to supervised and unsupervised FS algorithms, and proposes a unified framework for FS based on spectral graph theory. The proposed framework, named SPEC, generates families of algorithms for supervised and unsupervised FS.

2.2.2. Supervised

The *Relief* algorithm is based on a measure of relevance of each feature (Kira and Rendell, 1992). This relevance is proportional to how well its values distinguish between the instances of the same and different classes. The method randomly samples instances from the training set and updates the relevance of each feature based on the difference between the selected instance and the two nearest instances of the same and opposite classes. It scales well for high-dimensional datasets, but it does not remove redundant features. Considering s randomly sampled training instances it has overall complexity of $O(snd)$; usually, with $s > d$. The *ReliefF* algorithm (Kononenko, 1994) is an extension of *Relief*, which applies feature weighting and searches for several nearest neighbors. ReliefF can also be seen as a wrapper algorithm.

The *correlation-based filter selection* (CFS) method of Hall, 2000; Hall, 1998 is based on test theory (Ghiselli, 1964) concepts and relies on a set of heuristics to assess the adequacy of subsets of features. These heuristics take into account both the usefulness of individual features to predict the class label as well as their correlation. It copes with missing values on the data and has $O(nd^2)$ complexity.

The *fast correlation-based filter* (FCBF) of Yu and Liu (2003); Yu et al. (2004) follows a relevance-redundancy approach with two steps: compute the *symmetrical uncertainty* (SU), see (9) value of each feature and sorts these values into a list; remove redundant features from the list. Feature relevance alone is found insufficient for FS on high-dimensional data and a new framework decoupling relevance and redundancy analysis is introduced. The experimental results in Yu and Liu (2003); Yu et al. (2004) show the efficiency of the FCBF method, with complexity $O(nd\log(d))$, being faster than both ReliefF and CFS.

For binary problems ($c_i \in \{0,1\}$), the *Fisher ratio* (FiR) (Fisher, 1936), of the ith feature is defined as

$$\mathrm{FiR}_i = \frac{\left|\overline{X}_i^{(0)} - \overline{X}_i^{(1)}\right|}{\sqrt{\mathrm{var}(X_i)^{(0)} + \mathrm{var}(X_i)^{(1)}}}, \tag{2}$$

where $\overline{X}_i^{(0)}, \overline{X}_i^{(1)}, \mathrm{var}(X_i)^{(0)}$, and $\mathrm{var}(X_i)^{(1)}$, are the sample means and variances of feature i, for the patterns of each class. The FiR has been used successfully for microarray data (Furey et al., 2000).

The *minimum redundancy maximum relevancy* (mrMR) criterion (Peng et al., 2005) computes both the redundancy between features and the relevance of each feature. Redundancy is computed by the *mutual information* (MI) (Cover and Thomas, 1991) between pairs of features, whereas relevance is measured by the MI between each feature and the class labels. The mrMR method has also been applied successfully to microarray data (Ding and Peng, 2003).

Xing et al. (2001) proposed a hybrid filter-wrapper approach, able to successfully classify microarray data with 72 data points in a 7130-dimensional space. It applies a sequence of simple filters, followed by the Markov blanket filter (Koller and Sahami, 1996), to select particular feature subsets for each subset cardinality.

A game theoretic FS method, named *cooperative feature selection* (CoFS), was proposed and experimentally assessed by Sun et al. (2012). The method addresses the drawback of several information-theoretic FS methods, that rank poorly some features with strong discriminatory power as a group, being individually weak. For a recent survey on information-theoretic FS methods, see the work of Brown et al. (2012) and the many references therein.

3. Proposed filters

This Section describes the proposed FS filters. The first one exploits the idea that feature relevance is proportional to its dispersion. The second also includes a redundancy measure among relevant features.

The use of feature relevance/redundancy analysis is not novel for FS methods; we can find it, *e.g.*, in the CFS, mrMR, and FCBF methods. However, those previous approaches have the drawback of not scaling well for high-dimensional datasets. Our proposals fill in the gap of relevance/redundancy analysis for high-dimensional data, with two key contributions:

- assess relevance through dispersion, regardless of the class label;
- an efficient process to eliminate redundant features in high-dimensional datasets.

3.1. Dispersion measures

3.1.1. Existing measures

We now review some well-known dispersion measures that we use to compute relevance. One of the widely used dispersion measures is the variance (1), which is the relevance criterion of the TV method.

The *mean absolute difference* (MAD), defined as

$$\mathrm{MAD}_i = \frac{1}{n}\sum_{j=1}^{n}|X_{ij} - \overline{X}_i|, \tag{3}$$

computes the absolute difference from the mean value. The main difference between the variance and MAD measures is the absence of the square in the latter. The MAD, like the variance, is also scale-variant.

Another measure of dispersion applies the *arithmetic mean* (AM) and the *geometric mean* (GM). For a given (positive) feature X_i on n patterns, the AM and GM are given by

$$\mathrm{AM}_i = \overline{X}_i = \frac{1}{n}\sum_{j=1}^{n} X_{ij}, \quad \mathrm{GM}_i = \left(\prod_{j=1}^{n} X_{ij}\right)^{\frac{1}{n}}, \tag{4}$$

respectively; since $\mathrm{AM}_i \geqslant \mathrm{GM}_i$, with equality holding if and only if $X_{i1} = X_{i2} = \cdots = X_{in}$, then the ratio

$$\mathrm{R}_i = \frac{\mathrm{AM}_i}{\mathrm{GM}_i} \in [1, +\infty), \tag{5}$$

can be used as a dispersion measure. Higher dispersion implies a higher value of R_i, thus a more relevant feature. Conversely, when all the feature samples have (roughly) the same value, R_i is close to 1, indicating a low relevance feature.

3.1.2. Modified and proposed measures

We now describe dispersion measures that, to the best of our knowledge, have not been used for FS. If feature i has at least one zero occurrence, then $\mathrm{GM}_i = 0$, making R_i in (5) useless. We overcome this issue by applying the exponential function to each feature yielding

$$\mathrm{AMGM}_i = \frac{\frac{1}{n}\sum_{j=1}^{n}\exp(X_{ij})}{\left(\prod_{j=1}^{n}\exp(X_{ij})\right)^{\frac{1}{n}}} = \frac{1}{n\exp(\overline{X}_i)}\sum_{j=1}^{n}\exp(X_{ij}). \quad (6)$$

We also propose another dispersion measure, named *mean-median* (MM), given by

$$\mathrm{MM}_i = |\overline{X}_i - \mathrm{median}(X_i)|, \quad (7)$$

i.e., the absolute difference between the mean and median of X_i. Although this is not a classical dispersion measure, we have found it adequate for FS (see the experimental results in Section 4).

3.2. Similarity measures

3.2.1. Existing measures

We now address some similarity/redundancy measures that have been used for FS. Mitra et al. (2002) used the well-known sample *correlation coefficient* (CC)

$$\rho(X_i, X_j) = \frac{\frac{1}{n}\sum_{k=1}^{n}(X_{ik} - \overline{X}_i)(X_{jk} - \overline{X}_j)}{\sqrt{\mathrm{var}(X_i)\mathrm{var}(X_j)}}. \quad (8)$$

For uncorrelated variables $\rho(X_i, X_j) \simeq 0$, whereas for linearly dependent variables we have $\rho(X_i, X_j) = 1$.

Mitra et al. (2002) also proposed the *maximal information compression index* (MICI)

$$2\lambda(X_i, X_j) = \mathrm{var}(X_i) + \mathrm{var}(X_j) - \sqrt{(\mathrm{var}(X_i) + \mathrm{var}(X_j))^2 - 4\mathrm{var}(X_i)\mathrm{var}(X_j)(1 - \rho(X_i, X_j))}.$$

The FCBF method applies the *symmetrical uncertainty*

$$\mathrm{SU}(X_i, X_j) = 2\frac{\mathrm{H}(X_i) - \mathrm{H}(X_i|X_j)}{\mathrm{H}(X_i) + \mathrm{H}(X_j)}, \quad (9)$$

where $H(.)$ and $H(\cdot|\cdot)$ denote estimates[1] of the entropy and conditional entropy, respectively (Cover and Thomas, 1991).

3.2.2. Proposed measure

In order to quantify the redundancy between two features, say X_i and X_j, we propose to treat them as vectors and compute the *absolute cosine* (AC) of the relative angle between these vectors

$$|\cos(\theta_{X_iX_j})| = \left|\frac{\langle X_i, X_j\rangle}{\|X_i\|\|X_j\|}\right| = \frac{\sum_{k=1}^{n}X_{ik}X_{jk}}{\sqrt{\sum_{k=1}^{n}X_{ik}^2\sum_{k=1}^{n}X_{jk}^2}}, \quad (10)$$

where $\langle\cdot,\cdot\rangle$ denotes the inner product and $\|.\|$ the Euclidean norm. We have $0 \leqslant |\cos(\theta_{X_iX_j})| \leqslant 1$, with 0 meaning that the two features are orthogonal (maximally different) and 1 resulting from colinear features.

If we consider zero-mean features ($\overline{X}_i = \overline{X}_j = 0$), the CC of (8) satisfies

$$\rho(X_i, X_j) = \frac{< X_i, X_j >}{\|X_i\|\|X_j\|} = \cos(\theta_{X_iX_j}), \quad (11)$$

showing a connection between AC and CC. The AC measure uses the mean value of each feature; the *centering* procedure of removing the mean from each feature may influence the result of FS criteria.

For high-dimensional sparse data (*large d, small n*), the estimation of probabilities and entropies can be biased due to scarce data. As a consequence, information-theoretical FS methods perform better on dense than on sparse data (Ferreira and Figueiredo, 2011). This leads us to consider this geometric view of features, using angles to measure similarity.

3.3. Algorithm 1: Relevance-only filter

The first proposal applies one of the dispersion measures in order to assess the relevance of each feature. Given a pre-specified maximum number of features $m(\leqslant d)$, compute one of the dispersion measures described above (MAD, MM, TV and AMGM) of all the features, sort these values by decreasing order, and keep the top m (for simplicity, this is named Algorithm 1).

3.4. Algorithm 2: Filter based on relevance and redundancy

Although Algorithm 1 is very efficient, it is based solely on the individual feature relevances. However, it may happen that two (or more) relevant features are redundant and it is desirable to select only one of them. To achieve this goal, we propose to compute the redundancy between the most relevant features, that is, after sorting the d features by decreasing relevance, we compute *up to* $d-1$ pairwise similarities ($\ll d(d-1)/2$ possible comparisons) between consecutive features. In the end, we keep only features with high relevance and low (*i.e.*, below some threshold M_S) similarity among themselves.

It is not expected that redundant features are consecutive in the ranked sorted feature list. However, it is a waste of time to compute the redundancy between weakly relevant and irrelevant features, so we perform the redundancy check only among the top relevant features. Our method can be seen as a sub-optimal procedure for finding redundancy, since it allows removing the most redundant features from the subset of the most relevant features.

Algorithm 2 describes our relevance-redundancy FS procedure. In line 6, $S(\cdot,\cdot)$ refers to one of the feature similarity measures described in Section 3.2. Due to the redundancy elimination procedure, fewer than m features, say m', may be selected, depending on the value of parameter M_S; smaller values of M_S lead to smaller (and less redundant) feature subsets. We have found experimentally that on high-dimensional data, this redundancy check and elimination procedure usually stops after considering only a small percentage of the most relevant features.

Fig. 1 plots relevance and similarity of the consecutive m top-ranked features of the Brain-Tumor1 and Dexter datasets ($d = 5920$ and $d = 20000$ features, respectively; see Table 1). Relevance is computed by the MM measure (7) and similarity by the AC measure (10).

For both datasets, there are many pairs of consecutive features with similarity above 0.5 (some even close to 1), among the 1000 most relevant features. As referred by several authors (*e.g.*, Yu et al., 2004; Peng et al., 2005), we should remove this redundancy among the most relevant features. On the Brain-Tumor 1 dataset, the linear SVM classifier on the original features attains 8.0% error rate, whereas with the first 1000 features selected by Algorithms 1 and 2, we get 8.0% and 6.0%, respectively; on the Dexter dataset, we have 11.0%, 11.3%, and 9.0%, respectively.

Notice that Algorithms 1 and 2 can also work in supervised mode. In that case, the relevance and/or the redundancy measures should make use of the class labels (see Tables 3 and 4).

[1] Entropy estimate for a discrete feature can be easily obtained from its histogram. For real-valued features, some method for estimating entropy must be used (Beirlant et al., 1997).

Algorithm 2: Alg-relevance-redundancy unsupervised feature selection

Input $X : n \times d$ matrix, n patterns of a d-dimensional training set.
m ($\leqslant d$): maximum number of features to keep.
M_S: maximum allowed similarity between pairs of features.
Output FeatKeep: an $m'-$dimensional array (with $m' \leqslant m$) containing the indexes of the selected features.
$\widetilde{X} : n \times m'$ matrix, reduced dimensional training set, with features sorted by decreasing relevance.

1: Compute the relevance r_i of each feature X_i (columns of X), for $i \in \{1, \ldots, d\}$, using one of the dispersion measures (AMGM, MAD, MM, or IQR).
2: Sort the features by decreasing order of r_i. Let $i_1, i_2, \ldots, i_d$ be the resulting permutation of $\{1, \ldots, d\}$ (*i.e.*, $r_{i_1} \geqslant r_{i_2} \geqslant \ldots \geqslant r_{i_d}$).
3: FeatKeep[1] $= i_1$; {/* Keep the most relevant feature (the first). */}
4: prev = 1; next = 2; {/* The first (current) and next feature. */}
5: **for** $f = 2$ to d **do**
6: $s = S(X_{i_f}, X_{i_{\text{prev}}})$; {/* Compute similarity of pair i_f and i_{prev}. */}
7: **if** $s < M_S$ **then**
8: FeatKeep[next] $= i_f$; {/* Low similarity; keep feature i_f. */}
9: $\widetilde{X}_{\text{next}} = X_{i_f}$; {/* Store the feature values in $\widetilde{X}$. */}
10: prev = i_f; {/* prev holds the last selected and kept feature. */}
11: next = next + 1; {/* Move onto the next feature. */}
12: **end if**
13: **if** next = m **then**
14: break; {/* We have m features. Break loop. */}
15: **end if**
16: **end for**

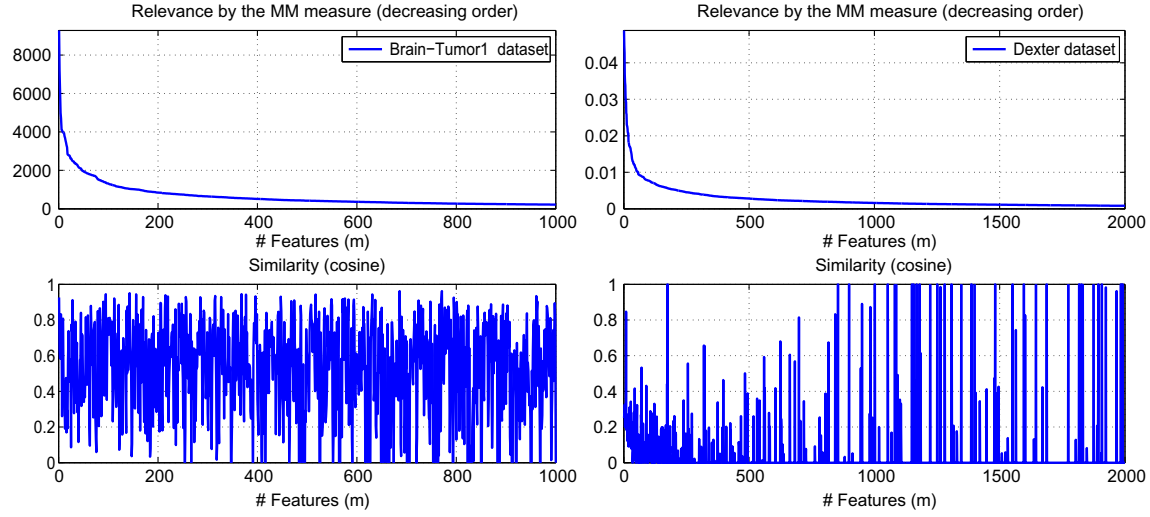

Fig. 1. Relevance and similarity of the top-rank $m = 1000$ and $m = 2000$ features, by the MM measure, of a single fold for the Brain-Tumor 1 ($d = 5920$) and the Dexter ($d = 20000$) datasets.

3.4.1. Complexity analysis

The dispersion measures considered have $O(nd)$ computational cost, while the sorting operation, using *quicksort*, has $O(d \log d)$ complexity. Thus, the overall cost of Algorithm 1 is

$$C_1 = \underbrace{O(nd)}_{Relevance} + \underbrace{O(d \log d)}_{Sort}. \tag{12}$$

Algorithm 2 adds to the complexity of Algorithm 1 the cost of computing the pairwise similarities. Thus, when selecting up to $m(\leqslant d)$ features, Algorithm 2 has a worst-case complexity of

$$C_2 = \underbrace{O(nd)}_{Relevance} + \underbrace{O(d \log d)}_{Sort} + \underbrace{O(nm)}_{Redundancy}. \tag{13}$$

3.5. Estimation of the number of features

In practice, many FS methods face the problem of selecting an adequate number (m) of features. This problem can be addressed by expensive methods such as cross-validation (CV), if the subsequent learning algorithm is available when FS is being performed. However, using CV defeats the purpose of using an unsupervised filter approach (independent of the learning algorithm and not

using training class labels). We propose an efficient criterion to be used in unsupervised filter mode, where the number of selected features depends on a *cumulative relevance* (CR) measure. Let $r_{i_1}, \ldots, r_{i_d}$ be the sorted relevance values and

$$c_v = \sum_{f=1}^{v} r_{i_f} \tag{14}$$

be the CR of the top v most relevant features. In Algorithm 1, we propose choosing the number of features as the lowest value m that satisfies

$$\frac{\sum_{f=1}^{m} r_{i_f}}{\sum_{i=1}^{d} r_i} = \frac{c_m}{c_d} \geqslant L, \tag{15}$$

where L is some threshold (*e.g.*, 0.95), leading to the choice of a fraction of the top-m ranked features. For Algorithm 2, we propose inserting another CR criterion that stops adding features when the sum of the relevance values of the already selected features exceeds some fraction (say 0.95) of the total relevance c_d.

4. Experimental evaluation

We have evaluated our methods on public-domain datasets for the task of supervised classification, using a 10-fold CV strategy. We use the linear *support vector machine* (SVM) (Schölkopf and Smola, 2002) from the PRTools[2] toolbox (Duin et al., 2007), with the 'c' parameter set to 1 (its default value) in all the experimentas. Changing the SVM kernel or tuning 'c' could yield better results; however, our main concern when assessing FS methods is to compare the performance of these methods with a common classifier and not necessarily with the best possible classifier. The filter and embedded FS methods considered for comparison are publicly available at featureselection.asu.edu. All experiments were performed on a laptop with 2.16 GHz CPU and 4 Gb of RAM.

Table 1 describes the datasets; we have datasets from the UCI repository[3] (Frank and Asuncion, 2010), text classification, face database, and bioinformatics datasets from the gene expression model selector (GEMS) project (Statnikov et al., 2005) [4] and from Arizona State University (ASU)[5] (Zhao et al., 2010a). Very recently, (Brown et al., 2012) proposed to measure the difficulty of a FS task by the ratio

$$R_{FS} = \frac{n}{a \times c}, \tag{16}$$

where a is the median arity of each feature, discretized with *equal-frequency binning* (EFB) (Witten and Frank, 2005) into 8 intervals; low values imply more difficult FS problems. Table 1 displays R_{FS} for each dataset.

Section 4.1 compares Algorithms 1 and 2, and the three different similarity measures in Algorithm 2 are evaluated in Section 4.2. A comparison with unsupervised and supervised FS techniques is provided in Section 4.3 and 4.4, respectively. Section 4.5 evaluates the running-time of our proposals against other methods, and Section 4.6 summarizes the results.

4.1. Comparison between Algorithms 1 and 2

Fig. 2 shows the 10-fold CV estimates of the test set error rates of linear SVM classifiers on the Hepatitis, Ionosphere, Prostate-Tumor, and Leukemia2 datasets, with features selected by: Algorithm 1 with TV, MAD, and MM; Algorithm 2, with the AC measure and $M_S = 0.7$. In each CV fold, the number of selected features m is obtained by setting different values of L in the range from 0.5 to 0.9. The dashed (blue) line is the baseline error (no FS).

For the lower dimensional datasets, Algorithm 2 with any of the three dispersion measures is advantageous with respect to Algorithm 1. On the Prostate-Tumor dataset, the MM measure attains better results than the other two measures, in both Algorithms 1 and 2, attaining test set error rates below the baseline. On the Leukemia2 dataset, the use of TV and MAD yields adequate results, reaching the baseline. With Algorithm 2, the MM measure and relevance-redundancy analysis achieve the best results, attaining error rates below the baseline.

Table 1
Datasets with c classes and n instances shown by increasing dimensionality d; in some cases $d \gg n$ and R_{FS} is small. None of these datasets contains multinomial variables.

Dataset	d	c	n	R_{FS}	Type of data
#1. Hepatitis	19	2	155	25.8	Biological
#2. Ionosphere	34	2	351	21.9	Radar
#3. Colon	2000	2	62	3.9	Microarray
#4. SRBCT	2309	4	83	2.6	Microarray
#5. AR10P	2400	10	130	1.6	Face
#6. PIE10P	2420	10	210	2.6	Face
#7. Lymphoma	4026	9	96	1.6	Microarray
#8. Leukemia1	5327	3	72	3.0	Microarray
#9. DLBCL	5470	2	77	4.8	Microarray
#10. TOX-171	5748	4	171	5.3	Microarray
#11. Brain-Tumor1	5920	5	90	2.3	Microarray
#12. Leukemia	7129	2	72	4.5	Microarray
#13. Example 1	9947	2	50	225.0	BoW
#14. ORL10P	10304	10	100	1.3	Face
#15. Brain-Tumor2	10367	4	90	1.6	Microarray
#16. Prostate-Tumor	10509	2	102	6.4	Microarray
#17. Leukemia2	11225	3	72	3.0	Microarray
#18. CLL-SUB-111	11340	3	111	4.6	Microarray
#19. 11-Tumors	12553	11	174	1.9	Microarray
#20. Lung-Cancer	12601	5	203	5.1	Microarray
#21. SMK-CAN-187	19993	2	187	11.7	Microarray
#22. Dexter	20000	2	2600	75.0	BoW
#23. GLI-85	22283	2	85	5.3	Microarray
#24. Dorothea	1000000	2	1950	288.0	Drug Discovery

4.2. Evaluation of similarity measures

Fig. 3 compares the use, in Algorithm 2, of the similarity measures mentioned in Section 3.2: AC, CC, and MICI, together with the AMGM relevance criterion, on the Hepatitis, Ionosphere, Leukemia1, and Example 1 datasets. The results of Algorithm 1 with AMGM are shown for comparison purposes. Notice that when using the MICI measure in Algorithm 2, the following feature is kept if the maximum similarity is below $M_S \times M_x$, where M_x is the maximum value of the MICI measure for that pair of features.

On the Hepatitis dataset, Algorithms 1 and 2 outperform the baseline, with any of the three similarity measures. On the Ionosphere dataset, neither algorithm attains better results than the baseline; the AMGM relevance is not adequate for this type of data. On the Leukemia1 dataset, the AC measure performs better than the other two measures. On the sparse BoW Example 1 dataset, the AC and CC measures attain the same results, better than MICI. On the Leukemia1 and Example 1 datasets, Algorithm 2 with the AC similarity measure reaches test errors below the baseline, using less than 2000 features. The AC measure is simpler and less computationally demanding than the other two measures, thus being preferable in some learning problems.

[2] http://www.prtools.org/prtools.html.
[3] archive.ics.uci.edu/ml/datasets.html.
[4] www.gems-system.org/.
[5] featureselection.asu.edu/datasets.php.

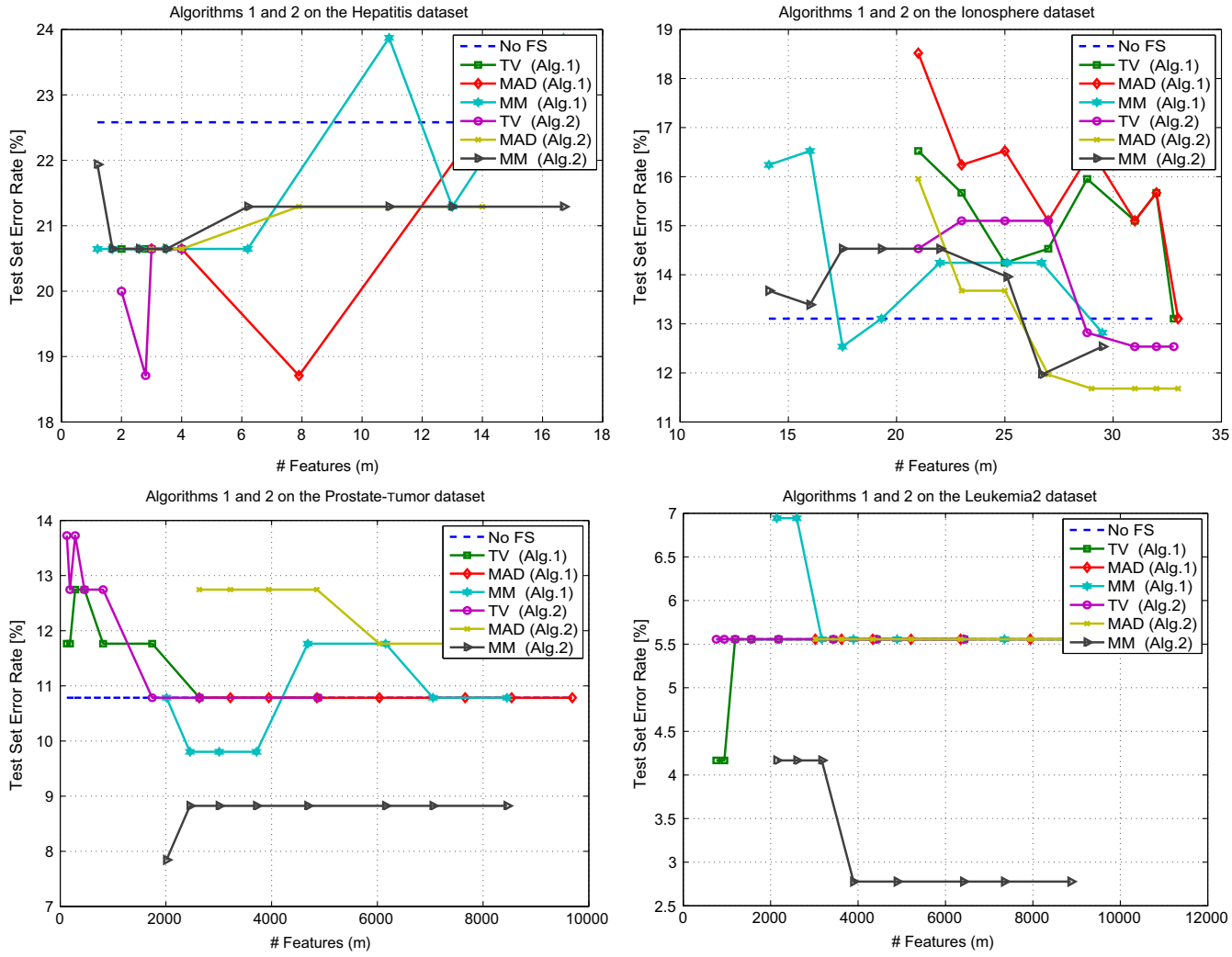

Fig. 2. Test set error rates (%) of 10-fold CV for the Hepatitis, Ionosphere, Prostate-Tumor, and Leukemia2 datasets, with a linear SVM classifier, as functions of the number of features for Alg1 and Alg2 (AC similarity with $M_S = 0.7$).

4.3. Unsupervised methods

We now compare Algorithm 2 against the unsupervised FS approaches LS and SPEC. Table 2 shows the 10-fold CV estimates of the test set error rates of linear SVM classifier on the features selected by Algorithm 2 using the MAD, MM, AMGM, and TV relevance, with AC similarity. In each of the 10-folds, the FS methods select the same number of features, computed by the CR criterion (15), for MAD relevance. On the Dexter and GLI-85 datasets, we display results for different values of L and M_S, to show their impact on the error. For LS and SPEC, the similarity matrix $\mathbf{W}$ in each fold is given by $\mathbf{W} = (1/2)\mathbf{X}_j\mathbf{X}_j^T$, where $\mathbf{X}$ is the training subset.

In most of these tests, all the FS methods are able to significantly reduce the number of features, with lower error than using the original set of features. The three dispersion measures proposed, applied by Algorithm 2, attain better results than the TV method. In many cases, Algorithm 2 selects subsets with less than the displayed number of features (m), due to the redundancy elimination procedure.

For the higher-dimensional datasets with sparse data, the LS and SPEC methods attain higher error rates than Algorithm 2. This suggests that LS and SPEC work better with lower-dimensional dense data. Our proposals performs consistently and stably across sparse and dense data. Among the four dispersion measures, MM attains the best results.

We assessed the statistical significance of our results using Friedman's test (Friedman, 1937; Friedman and March, 1940), as suggested by Demsar (2006) and Garcia et al. (2008). We obtained a p-value of $0.0231 < 0.05$, supporting the statistical significance of the favorable comparison of the proposed approach against other successful unsupervised FS techniques, on high-dimensional datasets. For further comparisons of the results on microarray datasets, with *recursive feature elimination* (RFE–SVM) (Guyon et al., 2002), the interested reader is referred to the works of Mundra and Rajapakse (2010), Mundra and Rajapakse (2007).

4.4. Supervised methods

We now compare Algorithm 2 against the supervised FS approaches ReliefF, CFS, FCBF, FiR, and mrMR. Table 3 shows the 10-fold CV estimates of test set error rates of linear SVM classifiers on features selected by Algorithm 2 using the unsupervised MM and the supervised FiR and MI relevance, with AC similarity using $M_S = 0.7$. In each CV fold, the FS methods select the same number of features m, computed by the CR criterion of (15), for the MM relevance measure. As done by Yu and Liu (2003), ReliefF uses $k = 5$ neighbors and $t = 30$ instances.

In our experiments, the CFS method showed an acceptable running time (a few minutes) for datasets with upto $d = 2000$ features. Beyond this point, its execution time becomes too high (hours),

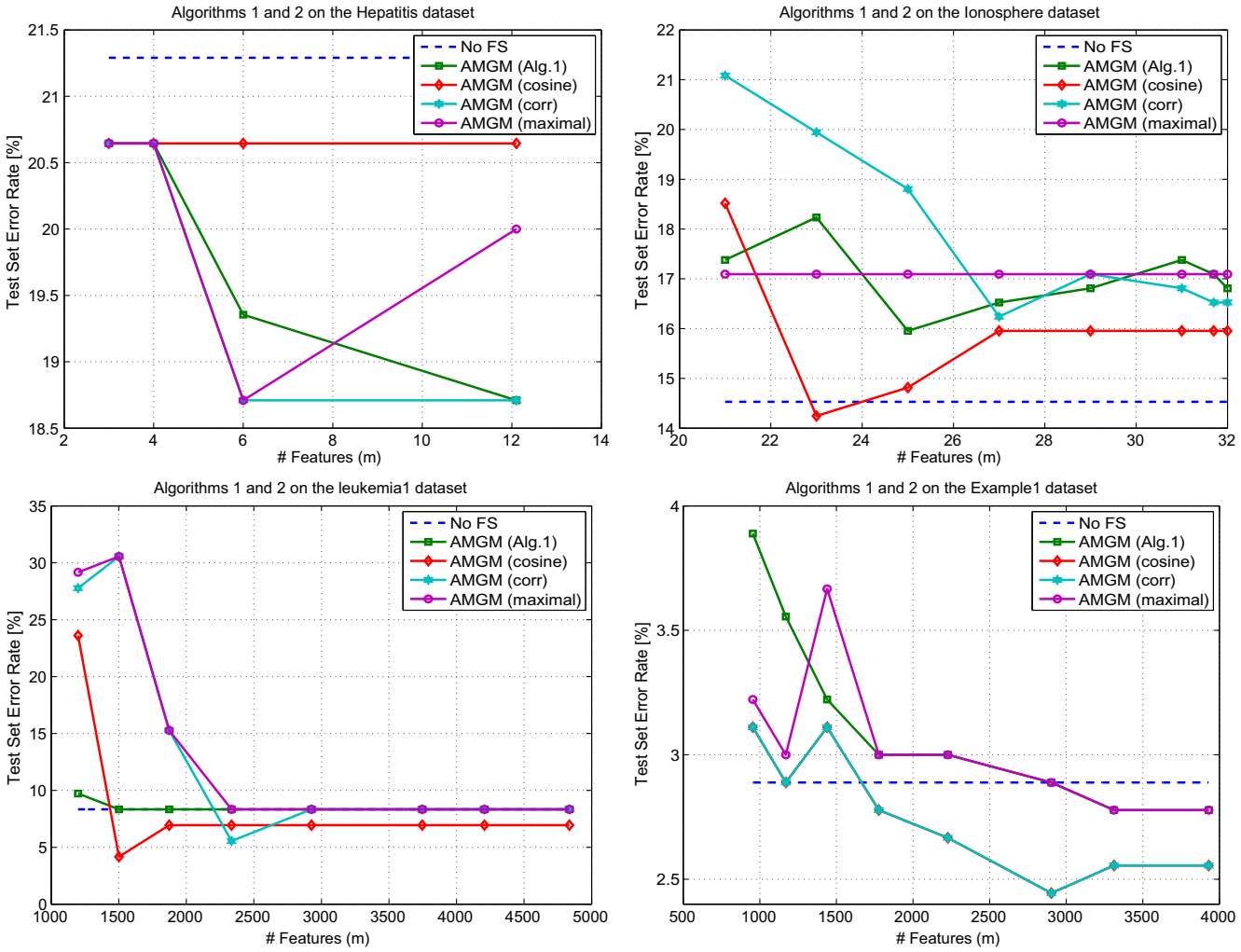

Fig. 3. Test set error rates (%) of 10-fold CV of linear SVM classifiers on the features selected by Alg2 ($M_S = 0.7$) for different similarity measures: AC-'cosine', CC-'corr', and MICI-'maximal') and Alg1, from the Hepatitis, Ionosphere, Leukemia1, and Example 1 datasets, with AMGM relevance.

due to its search mechanism. Consequently, we did not evaluate CFS beyond the AR10P dataset (symbol * in Table 3). Due to the large memory requirements of the Dorothea dataset, the ReliefF, FCBF, and mrMR methods from the ASU FS package were unable to run (they crashed due to insufficient memory resources); our methods and the FiR method from ASU were able to run.

The Friedman test on the average error (excluding the CFS method and the Dorothea dataset) has a p-value of $0.139 \times 10^{-6} < 0.05$, confirming statistical significance. Summarizing: (i) our Algorithm 2 attains adequate results, comparable to other techniques, performing consistently across different types of data; (ii) the execution time of some existing supervised FS methods becomes prohibitive on high-dimensional datasets, whereas our methods are still applicable; (iii) by using MI relevance in Algorithm 2, we make this criterion applicable to high-dimensional datasets, in contrast to the standard mrMR approach.

4.5. Running-time analysis

We now compare the running-time of Algorithm 2 (with MAD and FiR) with some filters and embedded approaches. Table 4 shows the *total* time taken by the FS methods to process the 10 CV folds and the test set error of the linear SVM classifiers. In each of the 10-folds, the FS methods select the same number of features. We have excluded the CFS and SMLR methods, since their running time is prohibitive for high-dimensional datasets.

Our methods are usually faster than the other unsupervised and supervised filter and embedded FS techniques, being closely followed by the LS method. Algorithm 2 usually outperforms the LS method.

Among the supervised FS filters, CFS is the slowest, being impractical for high-dimensional datasets. The FCBF is the fastest among the supervised filters. Our methods attain adequate results in terms of both time and error; for instance, in the Lung-Cancer dataset, Algorithm 2 is the fastest and achieves the lowest error.

4.6. Discussion

Our experimental results suggest the following remarks. The AMGM relevance, is adequate only for sparse data. The MAD and MM dispersion measures are good generic choices for sparse and dense data, with MM slightly preferable; these measures attain equal or better results than TV, with Algorithm 1 or 2. For Algorithm 1, we propose to use $L \in [0.7, 0.9]$; for Algorithm 2, $L \in [0.7, 0.99]$ and $M_S \in [0.7, 0.8]$ are adequate choices; the pair $(L, M_S) = (0.95, 0.8)$ is adequate for different datasets (see Tables 2 and 3).

Typically, Algorithm 2 yields better results than Algorithm 1, since it eliminates redundant features, but still scales well for

Table 2
Test set error rates (%) of linear SVM for a 10-fold CV for Alg2 (with MAD, MM, AMGM, and TV), LS and SPEC. We set the threshold in (15) as $L = 0.95$ and $M_S = 0.8$ (unless stated otherwise). In each fold, the FS methods select m features, computed by CR for the MAD relevance (best result in bold face, the second best is underlined).

Dataset	Alg2 (Unsupervised)				Unsupervised		Baseline
	MAD	MM	AMGM	TV	LS	SPEC	No FS
Hepatitis, $L = 0.99$	19.4	20.6	**18.7**	20.6	20.6	21.9	20.6
Ionosphere	15.1	**13.1**	15.1	**13.1**	16.2	14.2	13.7
Colon	21.0	**17.7**	19.4	**17.7**	21.0	22.6	19.4
SRBCT	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
AR10P	**0.8**	3.8	3.8	**0.8**	1.5	1.5	1.5
PIE10P	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
Lymphoma	**2.2**	**2.2**	**2.2**	**2.2**	**2.2**	**2.2**	**2.2**
Leukemia1	**4.2**	**4.2**	5.6	**4.2**	5.6	**4.2**	**4.2**
DLBCL	31.2	23.4	48.1	23.4	10.4	27.3	**6.5**
TOX-171	14.6	11.7	22.8	11.7	12.9	**1.8**	9.9
Brain-Tumor1	14.4	**12.2**	**12.2**	**12.2**	13.3	28.9	**12.2**
Leukemia	**2.8**	**2.8**	**2.8**	**2.8**	**2.8**	30.6	**2.8**
Example 1	2.7	2.7	**2.6**	2.7	3.3	22.9	2.7
ORL0P	2.0	2.0	4.0	5.0	**1.0**	**1.0**	**1.0**
Brain-Tumor2	38.0	38.0	**28.0**	38.0	40.0	40.0	34.0
Prostate-Tumor	**8.8**	10.8	10.8	10.8	13.7	10.8	10.8
Leukemia2	5.6	5.6	5.6	5.6	**4.2**	6.9	**4.2**
CLL-SUB-111	30.6	36.0	35.1	36.0	33.3	36.0	**29.7**
11-Tumors	12.1	13.8	14.9	13.8	23.0	30.5	**9.2**
Lung-Cancer	**4.9**	5.9	**4.9**	5.9	5.4	6.4	**4.9**
SMK-CAN-187	41.7	41.7	41.7	41.7	26.2	**25.7**	26.2
Dexter	5.3	5.2	5.2	5.2	5.2	39.3	**4.7**
Dexter, $L = 0.8,\ M_S = 0.7$	**4.8**	5.0	5.8	5.0	7.8	45.5	5.0
GLI-85	12.9	14.1	15.3	14.1	11.8	**9.4**	11.8
GLI-85, $L = 0.8,\ M_S = 0.7$	**12.9**	16.5	20.0	16.5	14.1	16.5	15.3
Dorothea	24.0	24.0	24.0	24.0	25.0	**22.0**	24.0

Table 3
Test set error rates (%) of linear SVM for a 10-fold CV for Alg2 (with MM, FiR, and MI relevance) and supervised filter methods. We use ($L = 0.99,\ M_S = 0.8$) and in each CV-fold the FS methods select $\hat{m}$ features, computed for the MM relevance (best result in bold face, the second best is underlined).

Dataset	Alg2			Supervised filters					Baseline
	MM	FiR	MI	RF	CFS	FCBF	FiR	mrMR	No FS
Hepatitis	**18.1**	18.7	**18.1**	21.3	19.4	20.0	20.0	20.6	18.7
Ionosphere	14.5	14.5	15.7	15.4	14.2	17.9	14.2	16.8	**13.7**
Colon	24.2	22.6	24.2	**19.4**	25.8	22.6	**19.4**	21.0	21.0
SRBCT	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	4.8	**0.0**	4.8	**0.0**
AR10P	2.3	1.5	**0.8**	**0.8**	6.2	13.1	**0.8**	56.2	**0.8**
PIE10P	**0.0**	**0.0**	0.5	**0.0**	*	1.0	**0.0**	24.8	**0.0**
Lymphoma	**2.2**	**2.2**	**2.2**	**2.2**	*	3.3	**2.2**	22.8	**2.2**
Leukemia1	5.6	**2.8**	6.9	6.9	*	5.6	4.2	9.7	5.6
DLBCL	16.9	24.7	36.4	10.4	*	**2.6**	7.8	16.9	5.2
TOX-171	15.8	21.1	11.1	**10.5**	*	15.2	**10.5**	36.8	15.8
Brain-Tumor1	13.3	12.2	13.3	11.1	*	18.9	11.1	25.6	**10.0**
Leukemia	**2.8**	12.5	**2.8**	**2.8**	*	4.2	4.2	8.3	**2.8**
Example 1	2.3	2.2	2.2	3.7	*	6.3	**2.1**	28.3	2.4
ORL0P	4.0	5.0	2.0	**1.0**	*	**1.0**	2.0	68.0	**1.0**
Brain-Tumor2	34.0	**22.0**	30.0	**22.0**	*	36.0	24.0	42.0	26.0
Prostate-Tumor	7.8	5.9	**4.9**	7.8	*	9.8	7.8	12.7	8.8
Leukemia2	6.9	5.6	5.6	5.6	*	**4.2**	5.6	8.3	6.9
CLL-SUB-111	35.1	30.6	39.6	36.0	*	38.7	36.0	**19.8**	36.0
11-Tumors	17.2	21.8	28.2	16.7	*	12.1	13.2	45.4	**9.8**
Lung-Cancer	5.9	6.4	4.9	4.9	*	6.4	5.4	11.8	5.9
SMK-CAN-187	41.7	40.6	53.5	24.6	*	33.2	**23.5**	33.2	24.1
Dexter	6.7	**6.0**	7.7	9.3	*	15.3	6.7	18.0	6.3
GLI-85	14.1	12.9	17.6	**11.8**	*	20.0	14.1	16.5	14.1
Dorothea	**25.0**	26.0	**25.0**	*	*	*	**25.0**	*	**25.0**

Table 4
For each dataset, numbered as in Table 1, the first row contains the test set error rates (%) of linear SVM for a 10-fold CV for Alg2 (with $M_S = 0.8$) and other FS methods selecting m features. The second row shows the *total* running time taken by each FS algorithm to select m features for the 10-folds. The best error rates and running times are in bold face; the second best are underlined.

#, m	Alg2		Filter						Embedded	Baseline
	MAD	FiR	LS	SPEC	RF	FCBF	FiR	mrMR	BLogReg	No FS
#3	24.2	30.6	21.0	24.2	24.2	25.8	24.2	**17.7**	21.0	21.0
m = 1000	**0.3**	7.4	0.4	1.5	9.0	147.0	7.1	19.8	2.2	–
#4	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	1.2	**0.0**	3.6	2.4	**0.0**
m = 1800	0.5	9.7	**0.4**	2.3	11.6	8.0	9.3	15.8	5.7	–
#7	**2.2**	**2.2**	**2.2**	3.3	**2.2**	3.3	3.3	25.0	8.7	**2.2**
m = 2000	**0.8**	29.5	**0.8**	9.0	21.7	25.2	29.2	24.7	143.7	–
#16	8.8	6.9	10.8	9.8	**5.9**	10.8	9.8	13.7	7.8	8.8
m = 4000	**1.8**	21.6	2.4	13.5	47.8	29.1	21.1	48.8	46.1	–
#20	**5.9**	6.4	6.4	8.4	6.9	7.4	6.9	11.3	7.4	6.4
m = 8000	**3.8**	54.9	7.5	47.5	95.5	208.8	54.1	88.5	207.4	–

high-dimensional datasets. The similarity measure AC is more efficient to compute and less dependent on the amount of data, as compared to previous measures.

Our methods are much faster than embedded methods and many filter methods (see Table 4) up to $d = 10^5$. The restriction of the redundancy analysis to pairs of the most relevant features is the reason behind the speed of Algorithm 2.

5. Conclusions

We have proposed two feature selection filters for high-dimensional datasets, with a relevance/redundancy approach, which can work in unsupervised or supervised mode. In the unsupervised case, relevance is based on simple statistical dispersion measures. The redundancy analysis is efficient, being computed solely between a few pairs of the most relevant features.

The experimental evaluation showed the efficiency of our methods. In some cases, our unsupervised methods attain lower generalization error than some supervised techniques, suggesting that for some medium and high-dimensional datasets, the use of the class labels does not necessarily help in choosing an adequate subset of features, since this choice is essentially based on redundancy analysis. Our filters are adequate for high-dimensional datasets, regarding the trade-off between accuracy, time, and memory efficiency. These methods yield reduced subsets of features that can be subsequently used by more sophisticated methods, such as embedded or wrapper techniques, lowering their cost and making them practical for high-dimensional data.

As ongoing work, we are developing criteria to compute both the maximum number of features and the maximum allowed similarity between features.

Acknowledgements

The authors thank the anonymous reviewers for their helpful and constructive comments and suggestions. This work was partially supported by the Polytechnic Institute of Lisbon under Grant SFRH/PROTEC/67605/2010 and by the FCT project PEst-OE/EEI/LA0008/2011.

References

Baldi, P., Hatfield, G., 2002. DNA Microarrays and Gene Expression. Cambridge University Press.
Beirlant, J., Dudewicz, E., Györfi, L., van der Meulen, E., 1997. Nonparametric entropy estimation: An overview. Internat. J. Math. Statist. Sci. 6, 17–39.
Bermejo, P., Gámez, J., Puerta, J., 2011. A GRASP algorithm for fast hybrid (filter-wrapper) feature subset selection in high-dimensional datasets. Pattern Recognition Lett. 32, 701–711.
Brown, G., Pocock, A., Zhao, M., Luján, M., 2012. Conditional likelihood maximisation: A unifying framework for information theoretic feature selection. J. Machine Learning Res. 13, 27–66.
Casado-Yusta, S., 2009. Adaptive branch and bound algorithm for selecting optimal features. Pattern Recognition Lett. 30 (5), 525–534.
Cawley, G., Talbot, N., 2006. Gene selection in cancer classification using sparse logistic regression with bayesian regularization. Bioinformatics 22, 2348–2355.
Chen, X., 2003. An improved branch and bound algorithm for feature selection. Pattern Recognition Lett. 24 (12), 1925–1933.
Cover, T., Thomas, J., 1991. Elements of Information Theory. John Wiley & Sons.
Das, S., 1994. Filters, wrappers and a boosting-based hybrid for feature selection. In: International Conference on Machine Learning – ICML, pp. 74–81.
Demsar, J., 2006. Statistical comparisons of classifiers over multiple data sets. J. Machine Learning Res. 7, 1–30.
Ding, C., Peng, H., 2003. Minimum redundancy feature selection from microarray gene expression data. In: Journal Bioinformatics and Computer Biology, pp. 523–529.
Duin, R., Juszczak, P., Paclik, P., Pekalska, E., Ridder, D., Tax, D., Verzakov, S., 2007. PRTools4.1, a Matlab Toolbox for Pattern Recognition. Tech. rep., Delft University of Technology.
Escolano, F., Suau, P., Bonev, B., 2009. Information Theory in Computer Vision and Pattern Recognition. Springer.
Esseghir, M., Liu, H., Motoda, H., Setiono, R., Zhao, Z., 2010. Effective Wrapper-Filter Hybridization Through GRASP Schemata. ACM Press, pp. 45–54.
Feo, T., Resende, M., 1989. A probabilistic heuristic for a computationally difficult set covering problem. Operat. Research Lett. 8 (2), 67–71.
Ferreira, A., Figueiredo, M., 2011. Unsupervised feature selection for sparse data. In: 19th European Symposium on Artificial Neural Networks-ESANN'2011, Bruges, Belgium, pp. 339–344.
Fisher, R., 1936. The use of multiple measurements in taxonomic problems. Annals Eugen. 7, 179–188.
Forman, G., 2003. An extensive empirical study of feature selection metrics for text classification. J. Machine Learning Res. 3, 1289–1305.
Frank, A., Asuncion, A., 2010. UCI machine learning repository. <http://archive.ics.uci.edu/ml>
Friedman, M., 1937. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. J. Amer. Statist. Assoc. 32 (200), 675–701.
Friedman, M., 1940. A comparison of alternative tests of significance for the problem of m rankings. Annals Math. Statist. 11 (1), 86–92, March.
Furey, T., Cristianini, N., Duffy, N., Bednarski, D., Schummer, M., Haussler, D., 2000. Support vector machine classification and validation of cancer tissue samples using microarray expression data. Bioinformatics 16 (10).
Garcia, S., Herrera, F., 2008. An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons. J. Machine Learning Res. 9, 2677–2694, Dec.
Ghiselli, E., 1964. Theory of Psychological Measurement. McGraw-Hill.
Guyon, I., Elisseeff, A., 2003. An introduction to variable and feature selection. J. Machine Learning Res. 3, 1157–1182.
Guyon, I., Gunn, S., Nikravesh, M., Zadeh, L. (Eds.), 2006. Feature Extraction, Foundations and Applications. Springer.
Guyon, I., Weston, J., Barnhill, S., Vapnik, V., 2002. Gene selection for cancer classification using support vector machines. Machine Learn. 46 (1–3), 389–422, March.
Hall, M., 1998. Correlation-based feature selection for machine learning. Ph.D. Thesis, Waikato University, Department of Computer Science, Hamilton, New Zealand.
Hall, M., 2000. Correlation-based feature selection for discrete and numeric class machine learning. In: ICML '00 Proceedings of the Seventeenth International Conference on Machine Learning. Morgan Kaufmann, pp. 359–366.
Hastie, T., Tibshirani, R., Friedman, J., 2009. The Elements of Statistical Learning. second ed.. Springer.

He, X., Cai, D., Niyogi, P., 2005. In: Laplacian score for feature selection. In: Weiss, Y., Scholkopf, B., Platt, J. (Eds.). Advances in Neural Information Processing Systems, 18. MIT Press, Cambridge, MA.

Hyunsoo, K., Howland, P., Park, H., 2005. Dimension reduction in text classification with support vector machines. J. Machine Learning Res. 6, 37–53.

Joachims, T., 1998. Text categorization with suport vector machines: Learning with many relevant features. In: ECML '98: Proc. of the 10th European Conference on Machine Learning. Springer-Verlag, London, UK, pp. 137–142.

Joachims, T., 2001. Learning to Classify Text Using Support Vector Machines. Kluwer Academic Publishers.

Jung, C.-S., Seo, H., Kang, H.-G., 2011. Estimating redundancy information of selected features in multi-dimensional pattern classification. Pattern Recognition Lett. 32 (4), 590–596.

Khushaba, R., Al-Ani, A., Al-Jumaily, A., 2011. Feature subset selection using differential evolution and a statistical repair mechanism. Expert Syst. Appl. 38 (9), 11515–11526.

Kira, K., Rendell, L., 1992. The feature selection problem: Traditional methods and a new algorithm. In: Association for the Advancement of Artificial Intelligence. AAAI Press and MIT Press, Cambridge, MA, USA, pp. 129–134.

Kohavi, R., John, G., 1997. Wrappers for feature subset selection. Artificial Intelligence 97 (1), 273–324.

Koller, D., Sahami, M., 1996. Toward optimal feature selection. Internat. Conf. Machine Learn. 1996 (May), 284–292.

Kononenko, I., 1994. Estimating attributes: Analysis and extensions of RELIEF. In: Proc. of the European Conference on Machine Learning. Springer, Verlag, pp. 171–182.

Krishnapuram, B., Carin, L., Figueiredo, M., Hartemink, A., 2005a. Learning sparse bayesian classifiers: Multi-class formulation, fast algorithms, and generalization bounds. IEEE Trans. Pattern Anal. Machine Intell. 27 (6), 957–968, June.

Krishnapuram, B., Carin, L., Figueiredo, M., Hartemink, A., 2005b. Sparse multinomial logistic regression: Fast algorithms and generalization bounds. IEEE Trans. Pattern Anal. Machine Intell. 27 (6), 957–968, June.

Liang, J., Yang, S., Winstanley, A., 2008. Invariant optimal feature selection: A distance discriminant and feature ranking based solution. Pattern Recognition 41, 1429–1439.

Liu, H., Motoda, H., 1998. Feature Selection for Knowledge Discovery and Data Mining. Kluwer Academic Publishers.

Liu, L., Kang, J., Yu, J., Wang, Z., 2005. A comparative study on unsupervised feature selection methods for text clustering. In: IEEE International Conference on Natural Language Processing and Knowledge Engineering, pp. 597–601.

Liu, R., Yang, N., Ding, X., Ma, L., nov. 2009. An unsupervised feature selection algorithm: Laplacian score combined with distance-based entropy measure. In: Third International Symposium on Intelligent Information Technology Application, 2009. IITA 2009, vol. 3. pp. 65–68.

Manning, C., Raghavan, P., Schütze, H., 2008. Introduction to Information Retrieval. Cambridge University Press.

Mitra, P., Murthy, C., Pal, S., 2002. Unsupervised feature selection using feature similarity. IEEE Trans. Pattern Anal. Machine Intell. 24, 301–312.

Mohamed, E., El-Beltagy, S., El-Gamal, S., 2006. A feature reduction technique for improved web page clustering. In: Innovations in Information Technology, Dubai, pp. 1–5.

Mundra, P., Rajapakse, J., 2007. SVM-RFE with relevancy and redundancy criteria for gene selection. Pattern Recognition Bioinform. 4774, 242–252.

Mundra, P., Rajapakse, J., 2010. SVM-RFE with MRMR filter for gene selection. IEEE Trans. NanoBiosci. 9 (1), 31–37.

Nakariyakul, S., Casasent, D., 2007. Adaptive branch and bound algorithm for selecting optimal features. Pattern Recognition Lett. 28 (12), 1415–1427.

Peng, H., Long, F., Ding, C., 2005. Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. IEEE Trans. Pattern Anal Machine Intell. 27 (8), 1226–1238.

Pudil, P., Ferri, F., Novovicova, J., Kittler, J., 1994a. Floating search methods for feature selection with nonmonotonic criterion functions. In: Proceedings of the Twelveth International Conference on Pattern Recognition, IAPR, pp. 279–283.

Pudil, P., Novovicova, J., Kittler, J., 1994b. Floating search methods in feature selection. Pattern Recognition Lett. 15 (11), 1119–1125.

Schölkopf, B., Smola, A., 2002. Learning with Kernels. MIT Press.

Shevade, S., Keerthi, S., 2003. A simple and efficient algorithm for gene selection using sparse logistic regression. Bioinformatics 19 (17), 2246–2253.

Somol, P., Pudil, P., Kittler, J., 2004. Fast branch & bound algorithms for optimal feature selection. Trans. Pattern Anal. Machine Intell. 26 (7), 900–912.

Statnikov, A., Tsamardinos, I., Dosbayev, Y., Aliferis, C.F., 2005. GEMS: A system for automated cancer diagnosis and biomarker discovery from microarray gene expression data. Internat. J. Medical Informat. 74 (7-8), 491–503. Aug.

Sun, X., Liu, Y., Li, J., Zhu, J., Chen, H., Liu, X., February 2012. Feature evaluation and selection with cooperative game theory. Pattern Recognition. <http://dx.doi.org/10.1016/j.patcog.2012.02.001>.

Torkkola, K., 2003. Discriminative features for text document classification. Pattern Anal. Appl. 6 (4), 301–308.

Veenman, C., Bolck, A., 2011. A sparse nearest mean classifier for high dimensional multi-class problems. Pattern Recognition Lett. 32 (6), 854–859.

Witten, I., Frank, E., 2005. Data Mining: Practical Machine Learning Tools and Techniques. second ed.. Elsevier, Morgan Kauffmann.

Xing, E., Jordan, M., Karp, R., 2001. Feature selection for high-dimensional genomic microarray data. In: Proceedings of the Eighteenth International Conference on Machine Learning. Morgan Kaufmann, pp. 601–608.

Yu, L., Liu, H., 2003. Feature selection for high-dimensional data: A fast correlation-based filter solution. In: Proceedings of International Conference on Machine Learning – ICML'03, pp. 856–863.

Yu, L., Liu, H., 2004. Efficient feature selection via analysis of relevance and redundancy. J. Machine Learning Res. 5, 1205–1224, Dec.

Zhao, Z., Liu, H., 2007. Spectral feature selection for supervised and unsupervised learning. In: Proceedings of the 24th international conference on Machine learning. ICML '07. ACM, New York, NY, USA, pp. 1151–1157.

Zhao, Z., Morstatter, F., Sharma, S., Alelyani, S., Anand, A., Liu, H., 2010a. Advancing feature selection research – asu feature selection repository. Tech. rep., Computer Science & Engineering, Arizona State University.

Zhao, Z., Wang, L., Liu, H., 2010b. Efficient spectral feature selection with minimum redundancy. In: Twenty-Fourth AAAI Conference on Artificial Intelligence.