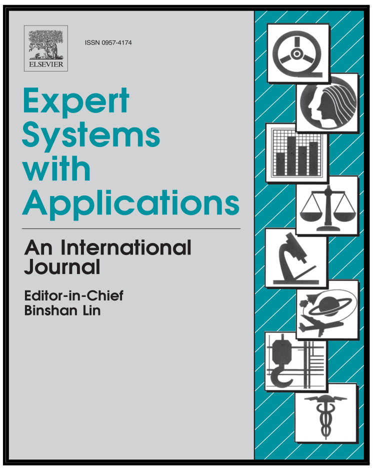
Accepted Manuscript



Improved multiclass feature selection via list combination

Javier Izetta, Pablo F. Verdes, Pablo M. Granitto

PII: S0957-4174(17)30467-0
DOI: 10.1016/j.eswa.2017.06.043
Reference: ESWA 11414

To appear in: *Expert Systems With Applications*

Received date: 8 March 2017
Revised date: 30 June 2017
Accepted date: 30 June 2017

Highlights

- We introduce new SVM-RFE feature selection methods for multiclass problems
- We use binary decomposition followed by strategies to combine lists of features
- We discuss statistical approaches and voting theory methods
- One-vs-One methods give better results than One-vs-All methods
- The new K-First method is the more effective in selecting relevant features

Improved multiclass feature selection via list combination

Javier Izetta[a], Pablo F. Verdes[a], Pablo M. Granitto[a,*]

[a]*CIFASIS, French Argentine International Center for Information and Systems Sciences, UNR–CONICET, Bv. 27 de Febrero 210 Bis, 2000 Rosario, Argentina*

Abstract

Feature selection is a crucial machine learning technique aimed at reducing the dimensionality of the input space. By discarding useless or redundant variables, not only it improves model performance but also facilitates its interpretability. The well-known Support Vector Machines–Recursive Feature Elimination (SVM-RFE) algorithm provides good performance with moderate computational efforts, in particular for wide datasets. When using SVM-RFE on a multiclass classification problem, the usual strategy is to decompose it into a series of binary ones, and to generate an importance statistics for each feature on each binary problem. These importances are then averaged over the set of binary problems to synthesize a single value for feature ranking. In some cases, however, this procedure can lead to poor selection. In this paper we discuss six new strategies, based on list combination, designed to yield improved selections starting from the importances given by the binary problems. We evaluate them on artificial and real-world datasets, using both One–Vs–One (OVO) and One–Vs–All (OVA) strategies. Our results suggest that the OVO decomposition is most effective for feature selection on multiclass problems. We also find that in most situations the new K-First strategy can find better subsets of features than the traditional weight average approach.

Keywords:

*Corresponding author

Email addresses: `izetta@cifasis-conicet.gov.ar` (Javier Izetta), `verdes@cifasis-conicet.gov.ar` (Pablo F. Verdes), `granitto@cifasis-conicet.gov.ar` (Pablo M. Granitto)

Feature Selection, Multiclass problems, Support Vector Machine

1. Introduction

Many important problems in Machine Learning, as well as in in-silico Chemistry (Raies & Bajic, 2016), Biology, "high-throughput" technologies (Golub et al., 1999; Leek et al., 2010) or text processing (Forman, 2003; Uysal, 2016), share the property of involving much more features than measured samples are available (Guyon & Elisseeff, 2003). The datasets associated to these problems are, unsurprisingly, called "wide". Usually, most of these variables carry a relatively low importance for the problem at hand. Furthermore, in some cases they interfere with the learning process instead of helping it, a scenario usually referred to as "curse of dimensionality".

Feature selection is an important pre–processing technique of Machine Learning aimed at coping with this curse (Kohavi & John, 1997). Its main goal is to find a small subset of the measured variables that improve, or at least do not degrade, the performance of the modeling method applied to the dataset. But feature selection methods do not only avoid the curse of dimensionality: they also allow for a considerable reduction in model complexity, an easier visualization and, in particular, a better interpretation of the data under analysis and the developed models (Liu et al., 2005).

Several methods have been introduced in recent years, from general ones like Wrappers (Kohavi & John, 1997) and filters (Kira & Rendell, 1992) to very specific ones developed for SVM (Weston et al., 2000; Nguyen & De la Torre, 2010) and RVM (Mohsenzadeh et al., 2013, 2016) classifiers. Amongst other methods in the field (Hua et al., 2009), the well-known Recursive Feature Elimination (RFE) algorithm provides good performance with moderate computational efforts (Guyon et al., 2002) on wide datasets. The original and most popular version of this method uses a linear Support Vector Machine (SVM) (Vapnik, 2013) to select the candidate features to be eliminated. According to the SVM–RFE algorithm, the importance of an input variable i is directly correlated with the corresponding component (w_i) of the vector defining the separating hyperplane ($\mathbf{w}$). The method is widely used in Bioinformatics (Guyon et al., 2002; Statnikov et al., 2005). Alternative RFE methods using other classifiers have also been introduced in the literature (Granitto et al., 2006; You et al., 2014).

Typical feature selection algorithms are designed for binary classification problems, as the original version of RFE. Multiclass problems have received much less attention because of their increased difficulty. Also, because some classifiers involved in the selection process are designed to solve binary problems. Most methods available for feature selection on multiclass problems are simple extensions of base methods. For example, RFE can be associated to a multiclass classifier like Random Forest (Breiman, 2001; Granitto et al., 2006).

Although SVM was originally developed to deal only with binary problems, it was extended to directly solve multiclass problems in different manners (Weston & Watkins, 1999; Crammer & Singer, 2001; Hsu & Lin, 2002), but with a modest success attributed mainly to the increased complexity of the solutions. On the other hand, in the last years several methods were developed to solve a multiclass problem using an appropriate combination of binary classifiers (Allwein et al., 2000; Hsu & Lin, 2002). The most usually followed strategy for multiclass SVM is known as "One–vs–One" (OVO). According to this approach, a classification problem with c classes is replaced with $M = c(c-1)/2$ reduced binary ones, each one of them consisting of discriminating a pair of classes. In order to classify a new example, it is passed through all binary classifiers and the most voted class is selected. Another useful strategy is "One–vs–All" (OVA). In this second case, a problem with c classes is replaced with $M = c$ reduced binary problems, each one of them consisting of discriminating a single class from all remaining ones.

Therefore, the most usual approach to implement a multiclass SVM–RFE method is to directly apply the RFE algorithm over an OVO or OVA multiclass SVM (Ramaswamy et al., 2001; Duan et al., 2007; Zhou & Tuck, 2007). The pioneering work of Ramaswany et al. (Ramaswamy et al., 2001) proposed the OVA solution, but also compared results with the OVO strategy. Duan et al. (Duan et al., 2007) and Zhou et al. (Zhou & Tuck, 2007) developed slight variations of the method, always considering both OVA and OVO implementations. Zhou et al. (Zhou & Tuck, 2007) also considered solutions to the RFE problem using a direct multiclass implementation.

Interestingly, the solutions to the multiclass SVM–RFE problem that we have just described involve an important decision about the feature selection process which is usually neglected: they rank features by simply averaging components over the binary problems. For an input variable i they use $< |w_{ij}| >_j$, the mean importance over all binary problems j, as the corresponding importance. As we discuss in the next section, this strategy can lead to sub–optimal selections in many cases. Once the original multiclass problem has been divided into multiple binary ones, the feature selection problem can be treated in a similar way. Then, a possible solution is to cast the multiclass feature selection problem as the problem of selecting candidate features from multiple lists (Jurman et al., 2008), each list corresponding to a different binary sub-problem.

Similar solutions have been studied in related fields. In Bioinformatics, for example, Haury et al. (Haury et al., 2011) discussed the combination of multiple lists of genes from bootstraps of the same gene-expression dataset. Zhou and Dickerson (Zhou & Dickerson, 2014) and Zhou and Wang (Zhou & Wang, 2016) proposed the use of class–dependent features (different features for each binary problem) for biomarker discovery. Dittman et al. (Dittman et al., 2013) showed that combining multiple lists in binary classification problems can improve the feature selection results. In a short work in text categorization, Neumayer et al. (Neumayer et al., 2011) suggested that the combination of rankings generated by diverse methods can improve the results of using a single method. Kanth and Saraswathi (Kanth & Saraswathi, 2015) used class–dependent features for speech emotion recognition, but using independent features for each class, not a final unique list.

In this work we discuss in depth the use of combination of multiple lists in feature selection for multiclass classification problems. We first introduce a simple mathematical framework for multiple lists. Using this framework, we propose diverse strategies to produce improved selection of feature subsets with SVM-RFE. Also, we use some specifically–designed artificial datasets and real–world examples to evaluate them extensively, using both the OVO and OVA strategies.

The rest of this article is organized as follows: in Section 2, we describe the feature selection methods introduced in this work. In Section 3 we evaluate these methods on diverse datasets and experimental setups. Finally, we draw our conclusions in Section 4.

2. List combination methods for SVM–RFE

The RFE selection method is a recursive process that ranks variables according to a given importance measure. At each iteration of the algorithm, the importance of each feature is calculated and the less relevant one is removed —in order to speed up the process, not one but a group of low relevance features is usually removed. Recursion is needed because the relative importance of each feature can change substantially when evaluated over a different subset of features during the stepwise elimination process, in particular for highly correlated features. The inverse order in which features are eliminated is used to create a final ranking. Then, the feature selection process itself is reduced to take the first n features from this ranking.

In the original binary version of SVM–RFE (Guyon et al., 2002), the projection of $\mathbf{w}$ (the normal vector to SVM's decision hyperplane) in the direction of feature i, w_i, is used as the importance measure. The method was efficiently extended to multiclass problems, employing the well-known OVO or OVA strategies to decompose the multiclass problem into a series of related binary ones (Ramaswamy et al., 2001; Duan et al., 2007; Zhou & Tuck, 2007). In both cases a set of M related binary problems is generated, each one solved by a vector $\mathbf{w_j}$. For each binary problem j, the importance of feature i is given by the corresponding component, w_{ij}.

In order to obtain a unique importance for each feature in this setup, the simplest solution is to average the absolute value of the components $|w_{ij}|$ over all related binary problems. We will call this method "Average" in the following. The Average solution is implemented, to the best of our knowledge, in all available RFE software packages, including the most popular amongst researchers (MATLAB, R and PYTHON platforms).

However, the only real advantage of the Average strategy is its simplicity. Two main drawbacks of this approach should be taken into consideration but are usually ignored:

1. The first issue can be called the *flattening* problem. Consider, for example, a feature e which is able to separate class j from all remaining classes, but is uninformative in other cases. Component w_{ej} will be large, but components w_{ek} with $k \neq j$ will be small, giving a low value for $< w_{ej} >_j$. Consider now another feature d which can give a modest help in separating any class from the others, obtaining always moderate values of w_{dj}, and therefore giving a medium value for $< w_{dj} >_j$. The Average strategy will clearly rank the latter over the former, but in most scenarios it will be desirable to keep the first variable over the second.
2. The second issue with the Average solution refers to *relative scales*. The length of vector $\mathbf{w_j}$ is different for each binary problem, as it depends on the margin of the solution, which can change considerably for classes that are relatively close or far away in feature space. Averaging components of vectors of different lengths can lead to the selection of sub–optimal subsets.

New strategies for feature selection able to overcome these drawbacks are

Ranking	List 1	List 2	...	List M
1	f_2	f_3	...	f_1
2	f_1	f_7	...	f_3
3	f_5	f_2	...	f_6
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
p	f_8	f_4	...	f_7

Table 1: List of ranked features for each binary problem.

needed. Here we propose to cast the problem as a selection of candidate features from multiple ranking lists (Jurman et al., 2008). We start by decomposing the multiclass problem into a set of M related binary problems (through the OVA or OVO strategies). The problem involves a set of p features, $F = \{f_1, f_2, \dots f_p\}$. SVM–RFE produces a ranking (an ordered list) for each individual problem using the components w_{ij}. An example is shown in Table 1. This set of lists can be arranged in a matrix (Table 2) where each row shows the position of each feature in the ranking produced for the binary problem shown on each column. We can now define a matrix of relative ranking positions as:

$$r_{i,j} = \frac{p - pos_{i,j}}{p} = 1 - \frac{pos_{i,j}}{p},$$

where $r_{i,j}$ is the relative ranking of feature f_i in the list corresponding to binary problem j, $pos_{i,j}$ is the position of the same feature in the corresponding ranking (Table 2), and p is the total number of features in the problem. Notice that the values of $r_{i,j}$ belong to the unit interval $[0, 1]$ and depend linearly on the ranking position (a value of $1 - 1/p$ must be interpreted as the first position in the ranking).

Important features should reflect in high values of $r_{i,j}$ for some i, j, meaning they are relevant to at least some of the binary classification problems. Two main strategies can be used to select those relevant features from this matrix and are discussed in the following.

2.1. Methods based on relative ranking statistics

The first strategy consists of measuring an appropriate statistic for each feature over all binary problems, and then using it to elaborate a final ranking of features. We selected the following four methods:

Features	List 1	List 2	...	List M
f_1	2	5	...	1
f_2	1	4	...	7
f_3	4	1	...	2
$\vdots$	$\vdots$	$pos_{i,j}$	$\ddots$	$\vdots$
f_p	$pos_{p,1}$	...	...	$pos_{p,M}$

Table 2: Matrix showing the position of each feature in the ranking of each binary problem. Rows correspond to features and columns to binary problems.

2.1.1. Average-SD

In this method, feature ranking is given by the average value of the relative position over all binary problems:

$$R_i = < r_{i,j} >_j,$$

where R_i is the ranking of feature f_i in the final ordered list, used to select features in the multiclass problem. Ties are broken by the standard deviation (SD) of the relative position (higher is better). We show in the next section that features with higher SD are preferable over lower SD ones, because a larger SD means that the feature has some better–than–average rankings.

Average-SD can be considered as the base strategy for multiple lists. It can overcome the *relative scales* problem on averaging weights, but is not expected to solve the *flattening* problem.

2.1.2. Best Ranking

In this second approach we rank every feature according to the best relative ranking that it reaches over the set of binary problems:

$$R_i = \max(r_{i,j})_j.$$

Ties are broken by the mean value of the relative position over all problems. A similar method has been used to select the winning class in multiple classifier systems (Ho et al., 1994). This strategy can be viewed as an extreme case, considering for each feature just one of the multiple rankings it receives and disregarding the rest. On the other hand, it is most aggressive in dealing with the flattening problem.

2.1.3. 3Q-SD

The third method orders features according to the 3rd quartile of the distribution of relative rankings:

$$R_i = 3Q(r_{i,j})_j,$$

where the 3Q function returns the 3rd quartile of its argument. As in Average-SD, ties are broken by the SD. This approach is intermediate between the two previous ones, searching for features that reach a high relative position, but also considering the full relative rankings distribution.

2.1.4. K-First

This method is adapted from a strategy to select relevant documents in information retrieval (Nuray & Can, 2006). The idea is to only consider features located in the top k positions of each individual list. We re-scale the relative rankings with a linear mapping reaching 0 for the $k + 1$ feature, and then take the average of this new relative importance:

$$r'_{i,j} = \max(1 - \frac{pos_{i,j}}{k}, 0)$$

$$R_i =< r'_{i,j} >_j,$$

where $r'_{i,j}$ is the re-scaled relative weight for feature f_i and k is the number of features to be considered from each list ($k < p$). As in the Best Ranking method, ties are broken by the mean value of the original relative ranking, $< r_{i,j} >_j$. We discuss the set of parameter values k in the next section. This strategy is aimed at searching for features which are highly relevant for some of the problems, but is not limited to searching for the most relevant features —as the Best Ranking method is. It can potentially overcome both drawbacks of Average: relative scaling and flattening.

2.2. Methods based on voting theory

The second general strategy is related to voting theory (Saari, 2001; Young, 1988). In this setup we consider each binary problem as a voter, producing a ranking over a set of p candidates. Multiple methods were developed over the years to solve the problem of combining elector preferences to find winner candidates —the most useful of them are known collectively as "Condorcet Methods". We focused on two popular procedures as selection methods for relevant features over multiple lists:

2.2.1. Condorcet

The most basic Condorcet method is known as Copeland's method, or simply as Condorcet method (we will use the latter name in this work). It confronts each pair of features on every list (all binary problems), and then counts the number of wins minus the number of defeats for each feature (Young, 1988). A feature wins over another if it is ranked higher in the considered list. The global difference between wins and loses is used to rank features in the multiclass problem. Ties are broken by average relative rankings.

2.2.2. Schulze

This method, introduced by Schulze informally in 1997 and published later (Schulze, 2011), represents an improvement over previous Condorcet methods. It begins by counting wins and loses over each pair of features and all lists, storing these numbers in a pairwise preference matrix. Then a graph is constructed, with features as nodes and values in the matrix as weights. Finally, using a variant of the FloydWarshall algorithm, the strongest paths over the graph are selected for each pair of features, and their strengths are used to compare features. The strength of a path is defined as that of its weakest link (i.e., lower value in the matrix of preferences). A path between two nodes is valid if there is a sequence of strictly decreasing weights connecting them (Schulze, 2011). Features with more wins upon strength comparison are ranked first. The method is expected to perform better than basic Condorcet, but the computational load involved is significant.

3. Evaluation on artificial datasets

We first consider artificial classification problems in order to evaluate specific aspects of the new methods and to be able to compare their capabilities in a controlled manner.

3.1. Experimental setup

As in previous works (Granitto et al., 2006), we strive to use an appropriate computational setup for feature selection. We perform $n = 100$ times a random split ($75\% - 25\%$) of each dataset in training and testing sets (the former are used to select features and train the classifiers, while the latter for model accuracy estimation). The testing sets are completely external to the feature selection process, thus providing unbiased estimates of classification errors for different number of features. The results of the n replicated experiments are then aggregated to yield mean error rate estimations and their corresponding SD.

SVM–RFE was implemented using the OVO strategy unless specified otherwise. In both cases, we created the corresponding binary problems and produced a ranking of features for each of them. To create a ranking we used the standard SVM–RFE (linear kernel), as described by their authors (Guyon et al., 2002), eliminating 10% of the features at each iteration until there were less than 20 features left, when we slowed the procedure to eliminate 1 feature at each iteration. The fixed set of lists of ranked features were combined using the methods described before, producing a final list of features for each method under evaluation. Finally, for each method we fitted a multiclass SVM for a varying number of features, from 2 to p, using only the training data, and measured the classification error using the testing set. The C parameter was estimated in all cases using 5-fold cross validation of the training set.

3.2. Artificial datasets

We created three different multiclass datasets that provide diverse challenges to our methods. In all cases, each class is sampled from a Gaussian distribution with diagonal covariance matrix. For each dataset we can identify, by construction, a group of relevant features that can discriminate amongst classes and another group of irrelevant features containing Gaussian noise. All noisy features have the same mean (0) and SD (1) for all classes. Each dataset is composed of 3000 points evenly distributed among classes. The number of noisy features is fixed at 500.

In the first dataset, called Artificial-1, there is a group of 5 features that is relevant for each class, i.e., class-specific features. The set of 5 features together shift the class center away from other classes. All relevant features have the same importance for the problem. The SD of the Gaussian distributions corresponding to relevant features are always set to 0.5.

A different situation arises when there are sets of features which are relevant to some of the classes (more than one) but not for all of them. We created a second classification problem, Artificial-2, to evaluate this challenge. The dataset has 8 classes and 25 relevant features, all sampled from Gaussian distributions with a SD of 0.5. The first 5 features are relevant for the first 3 classes of the problem. The following 5 are relevant for classes 4

Dataset	Classes	Relevant features	Noisy features
Artificial-1-3C	3	15	500
Artificial-1-4C	4	20	500
Artificial-1-5C	5	25	500
Artificial-1-8C	8	40	500
Artificial-1-16C	16	80	500
Artificial-2-8C	8	25	500
Artificial-3-3C	3	15	500
Artificial-3-8C	8	40	500
Artificial-3-16C	16	80	500

Table 3: Details of the artificial datasets used in this work.

and 5 only, and are less relevant than the first 5. The rest of the features are relevant for a single class, 5 for each of the remaining 3 classes. These last features are less relevant than the first 10 features.

Finally, we created a third problem, called Artificial-3, where all relevant features are equally useful for all classes at the same time. As in Artificial-1, there are 5 features for each class, all sampled from Gaussian distributions with a SD of 0.5.

In all problems there is an overlap among classes, giving a nonzero Bayes error. We created five datasets for Artificial-1, with an increasing number of classes, and 3 datasets for Artificial-3 in the same way. Table 3 collects technical details of the datasets.

3.3. Methodological setup

3.3.1. K-First

The K-First method is the only approach involving a parameter that needs to be set, k. The value of this parameter regulates the number of variables that receive a relative ranking. A very low value would make the method similar to Best Ranking, while a high one would turn the method into 3Q-SD (furthermore, $k = p$ would convert the method into Average-SD).

We evaluated several values of k (increasing fractions of p) over all artificial datasets considered. Figure 1 shows the corresponding error curves as a function of the number of features selected by the method, for some representative problems. Error curves for all other artificial problems are similar to the reported ones (we include more figures in the Additional Material section). The vertical dotted lines show the correct number of relevant features for the problem, i.e. where the minimum of the curve should ideally be located. When possible, we show with a gray horizontal line the chance error level for the corresponding problem. The top row shows typical results for the Artificial-3 problem. In this case there are no class-specific features and, as a consequence, the results are almost independent of k. The bottom row shows results for class-dependent problems, Artificial-1 and 2. In this case the results clearly depend on k. We found that a value of 10% of p gives consistently good results in all artificial cases considered here, therefore we will use this value for the rest of the paper.

3.3.2. Average-SD and 3Q-SD

As noted before, these methods use the SD of the relative rankings as a breaking tie criterion, considering larger values of SD as better than smaller ones. This is based on the assumption that a large SD is associated with high rankings for some of the binary problems, and that such behavior is able to highlight class-dependent features over flat ones. In order to confirm this, we compared for both methods over a set of artificial problems the use of maximum versus minimum SD to break ties. Figure 2 shows the corresponding results for some representative cases. They are similar in all other cases (some of which are shown in the Additional Material section). As this figure shows, using maximum values always leads to equal or better performance than using minimum ones.

3.3.3. OVA–SVM vs. OVO–SVM

We applied both OVA and OVO strategies, combined with our feature selection methods, to all artificial datasets. We compared all results and found that the OVO strategy yields equal or superior performance in all cases. In Figure 3 we show some representative examples of this comparison, using the *Artificial-1-8C* and *Artificial-2-8C* datasets. On the left column we show OVA results, while the OVO case is depicted on the right column. We use the same scale for the corresponding panels. We also included the Bayes error for both datasets as dotted horizontal lines, and the true number of relevant features as a dotted vertical lines. More datasets are included in the Additional Material section.

It is interesting to note that the two methods more directly aimed at finding class-relevant features (K-First and Best Ranking) are the ones showing

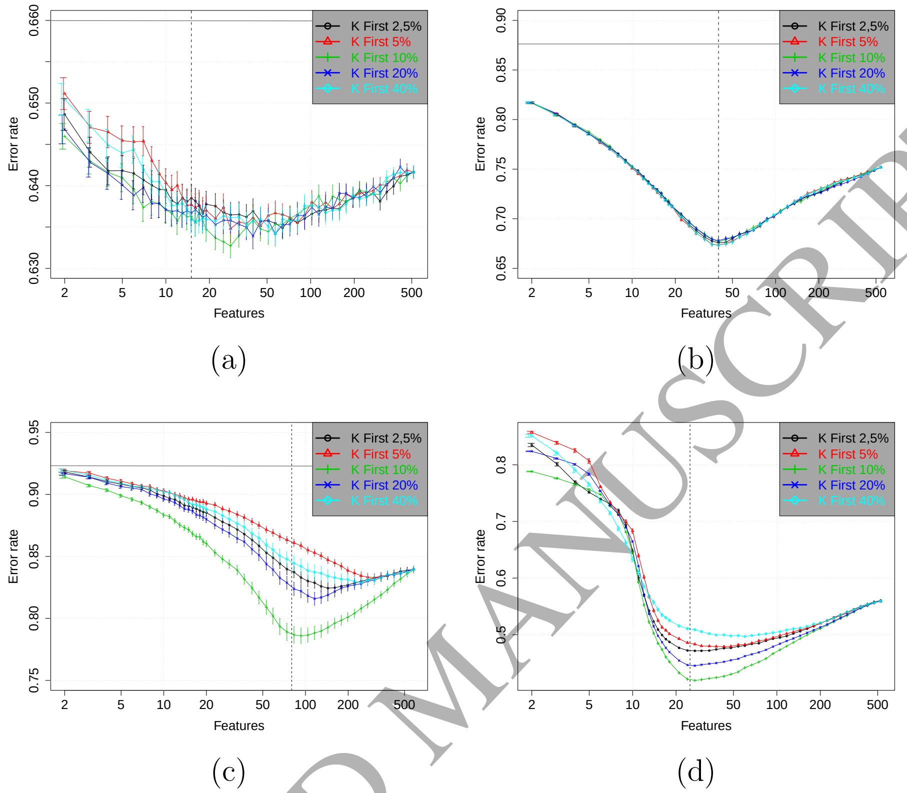
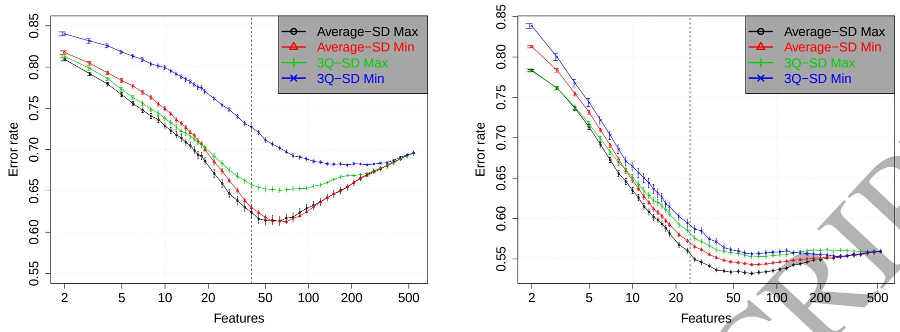


Figure 1: Evaluation of different values of k for the K-First method. Each line shows average error rates as a function of the number of features selected by the corresponding method, with 1 SD error bars. (a) Artificial-3-3C (b) Artificial-3-8C (c) Artificial-1-16C (d) Artificial-2-8C (chance error 0.875).

the bigger gains under the OVO strategy. Probably the OVO strategy can filter some of the noisy features more efficiently than OVA, as it considers significantly more lists of features ($M = c(c-1)/2$ vs. $M = c$). After this comparison we will only use OVO-SVM to evaluate our new methods.

3.4. Evaluation of the Methods on Artificial Datasets

Figure 4 shows the results for 4 versions of the Artificial-1 problem and 2 of the Artificial-3 problem. The remaining dataset from Artificial-1 is shown on Panel (c) of Figure 3. Results for Artificial-2 are shown on Panel (d) of the same figure. Additional datasets are included in the Additional Material section. Overall, artificial problems show that the K-First method is the



(a) (b)

Figure 2: Comparison of tie breaking using maximum or minimum SD for Average-SD and 3Q-SD. Details are similar to Figure 1. Chance error of 0.875 for both panels. (a) Artificial-1-8C. (b) Artificial-2-8C.

most efficient one in finding subsets of features with low classification error, followed closely by the Best-Ranking method. The Schulze method shows good performance in several datasets. The other 3 methods show similar results, though not as good as the first group.

On the Artificial-3 datasets (all relevant features are useful for all classes) the differences among methods are clearly smaller than on the other 2 problems (with class-dependent features). Differences in performance increase with the number of classes for the Artificial-1 dataset.

Comparing the two methods based on voting theory, the low performance of Condorcet compared with Schulze is notorious. Taking a closer look at the method, we noticed that Condorcet produces a lot of ties in the rankings, which are broken using average positions. This produces a bias towards features with good global average values instead of features highly relevant for a few lists.

Another interesting analysis that can be made with artificial datasets is the position occupied by the truly relevant features on the rankings produced by the different methods, as we know in advance which features are noisy and which ones are informative. A perfect method should rank all relevant features first, with all noisy features following.

For each artificial problem, we analyzed the distribution of rankings given by each selection method to the set of relevant features and to the set of noisy features. We then computed some descriptive statistics of those two

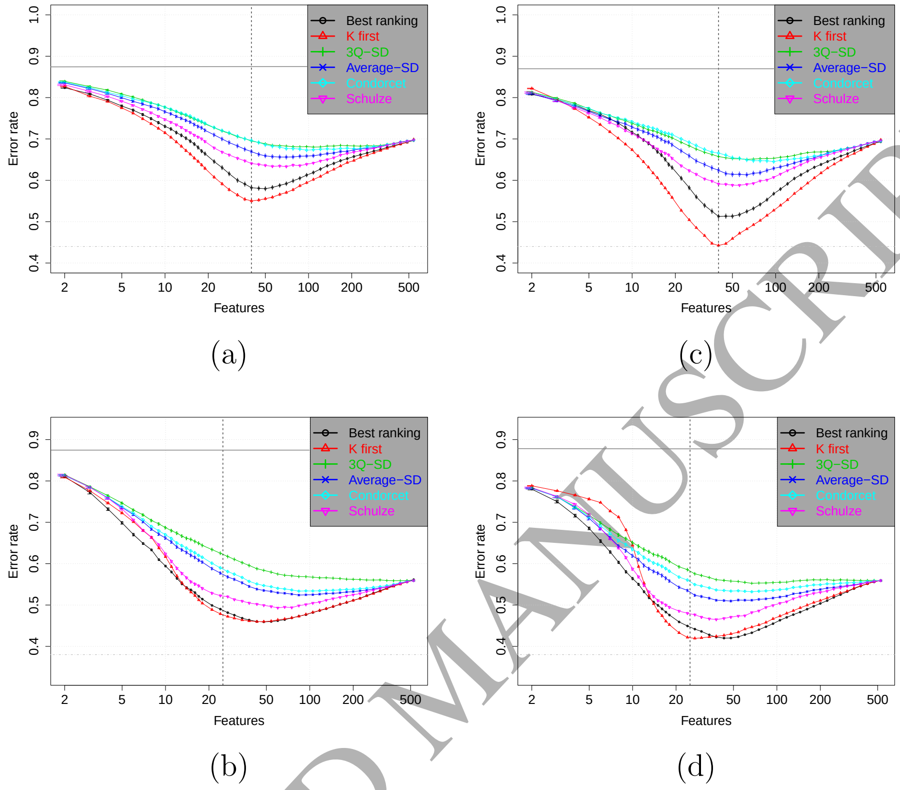


Figure 3: OVA-SVM vs. OVO-SVM on two artificial problems. Details are similar to Figure 1. (a) RFE-OVA-SVM on Artificial-1-8C. (b) RFE-OVA-SVM on Artificial-2-8C. (c) RFE-OVO-SVM on Artificial-1-8C (d) RFE-OVO-SVM on Artificial-2-8C.

distributions (Best, 1st. quartile, Mean, 3rd. quartile and Worst). In Table 4 we show these statistics on dataset Artificial-1-8C, which is representative of the results obtained on the other versions of this problem. All six methods rank relevant features at the first positions and noisy features at the last ones, but there are important differences. Looking at the Mean and 3rd. Quartile of the distributions, it is clear that K-First, Best Ranking and Schulze, in that order, are the most accurate ones in ranking most of the features according to their global relevance. These results confirm that the low error rates on the figures discussed before are directly related to a better feature selection by those methods.

In Table 5 we show the corresponding statistics for the Artificial-2-8C

Relevant	3Q-SD	Av-SD	Best Rank.	K-First	Condorcet	Schulze
Best	1	1	1	1	1	1
1st Q.	14	14	12	11	17	13
Mean	70	65	33	22	79	48
3rd Q.	98	86	41	31	110	61
Worst	436	476	357	495	474	410
Noisy	**3Q-SD**	**Av-SD**	**Best Rank.**	**K-First**	**Condorcet**	**Schulze**
Best	1	1	1	3	1	1
1st Q.	160	161	165	165	159	163
Mean	287	287	290	290	286	288
3rd Q.	415	415	415	415	415	415
Worst	540	540	540	540	540	540

Table 4: Statistics of the rankings given to relevant and noisy features by the diverse methods considered in this study for the Artificial-1-8C dataset. Values are rounded when needed.

dataset. This is the most interesting problem, as it contains subsets of relevant features with diverse levels of relevance. In the table we separated the relevant features into 3 subsets. As it can be observed in the table, K-First is the most effective strategy in separating the subsets of relevant features, and not only relevant from noisy features.

Finally, in Table 6 we show statistics for the Artificial-3-8C dataset — it is representative of all versions of this problem. As discussed before, all methods are almost equivalent on this dataset.

A last comment is in order about the Best Ranking method. As it can be seen in the tables, it can give high rankings to noisy features more easily than the K-First method, as it bases the final ranking on a single value for each feature.

4. Evaluation on real–world datasets

We used 14 real–world datasets to evaluate our new methods. Details and origins of the datasets are collected on Table 7. We selected datasets from three different domains. The first 4 datasets collect mass spectrometry measurements of food products. All recorded peaks are present in the datasets. Some of the products under analysis can present class-specific features, reflecting particularities of some products, such as origin or manufacturing

1 to 5	3Q-SD	Av-SD	Best Rank.	K First	Condorcet	Schulze
Best	1	1	1	1	1	1
1st Q.	2	2	2	2	2	2
Mean	22	5	9	3	15	4
3rd Q.	28	5	17	4	18	5
Worst	481	62	45	7	85	18
6 to 10	3Q-SD	Av-SD	Best Rank.	K First	Condorcet	Schulze
Best	1	1	1	1	1	1
1st Q.	17	7	6	7	13	6
Mean	80	24	13	8	50	9
3rd Q.	158	30	19	9	58	11
Worst	523	241	44	14	453	70
11 to 25	3Q-SD	Av-SD	Best Rank.	K First	Condorcet	Schulze
Best	1	2	1	8	1	4
1st Q.	35	19	12	14	29	15
Mean	161	87	22	18	101	65
3rd Q.	265	116	30	22	185	70
Worst	524	500	57	29	524	501
Noisy	3Q-SD	Av-SD	Best Rank.	K First	Condorcet	Schulze
Best	1	3	3	10	1	8
1st Q.	142	147	151	151	142	148
Mean	269	273	275	275	274	274
3rd Q.	399	400	400	400	399	400
Worst	525	525	525	525	525	525

Table 5: Statistics of the rankings given to relevant and noisy features by the diverse methods considered in this study for the Artificial-2-8C dataset. The relevant features are divided into three subsets, and ordered according to their relevance by construction. Values are rounded when needed.

Relevant	3Q-SD	Av-SD	Best Rank.	K First	Condorcet	Schulze
Best	1	1	1	1	1	1
1st Q.	11	11	11	11	11	11
Mean	27	29	41	26	28	29
3rd Q.	33	37	53	33	33	37
Worst	440	431	368	250	342	252
Noisy	3Q-SD	Av-SD	Best Rank.	K First	Condorcet	Schulze
Best	5	2	2	8	3	7
1st Q.	165	165	164	165	165	165
Mean	290	290	290	290	290	290
3rd Q.	415	415	415	415	415	415
Worst	540	540	540	540	540	540

Table 6: Statistics of the rankings given to relevant and noisy features by the diverse methods considered in this study for the Artificial-3-8C dataset. Values are rounded when needed.

method. The following 6 datasets come from the UCI repository. These are more traditional datasets, with more samples than features and multiple classes, involving typical pattern recognition problems. Finally, we selected 4 gene expression datasets from human tissues. These datasets were filtered by curators to obtain circa 1000 genes with high signal–to–noise ratio in each case.

In order to compare our results against previous methods we implemented 3 versions of MSVM-RFE, as described by Zhou & Tuck (2007). The first method uses the multiclass SVM developed by Crammer & Singer (2001). We will denote it "Zhou C & S". The second one uses the method of Weston & Watkins (1999), in the following denoted by "Zhou W & W". Finally, we implemented MSVM-RFE with the OVO decomposition of the traditional binary SVM. We will refer to this method as "Zhou OVO". Notice that it is equivalent to the Average Weights methodology, which is implemented by default in most available Machine Learning packages, as previously explained.

On Figures 5 and 6 we show the results for eight datasets, while the remaining cases are shown in the Additional Material section. In general, differences in results for real world data are less notorious than for the Artificial problems. For UCI and Mass-Spectrometry datasets, K-First is in general the method showing the best results in finding small subsets with reduced classification error, followed by Best Ranking and 3rd Quartile. In some problems, like Apples and Libra, differences are more notorious. On the gene expression datasets all methods show small differences, but in general the variants of MSVM-RFE exhibit better results than on the other domains. These datasets have been filtered by curators and as a consequence all features are informative. We believe that this improves the performance of averaging methods over methods that search for class-specific features.

Error curves show the complete behavior of the methods as a function of the number of features, but occasionally diverse methods are more efficient in selecting a high number or just a few features. To produce a more concrete comparison, we measured for two fixed numbers of selected features (10 and 20) the proportion of runs on which method A shows a smaller error than method B, for each of the three domains under evaluation. The full resulting matrices are shown in the Additional Material section. From these matrices we computed a ranking for each method, counting the number of other methods that it excels. We show the corresponding results in Table 8. They confirm the information extracted from the error curves: on the UCI and Mass-Spectrometry domains K-first shows the best results, but Best Ranking and 3rd Quartile also have high rankings. On the gene expression domains the best results come from one of the MSVM-RFE methods.

5. Computational burden

We evaluated the burden of the 6 new methods as a function of the number of features and samples using the Artificial-1 dataset. In panel (a) of Figure 7 we show how the running time scales with the number of features in the problems, using a log-scale for times. We include the 3 versions of the method by Zhou et al. as a comparison. It is clear from this figure that all but the Schulze method scale almost linearly with the number of features, being Condorcet and 3rd Quartile the slowest methods. Schulze is cubic in the number of features, as it involves a variant of the FloydWarshall algorithm to find shortest paths in a graph. On panel (b) of the same figure we show the dependence on the number of samples in the dataset. All new methods, including Schulze, scale almost linearly with the number of samples. The two variants of Zhou's method using direct multiclass SVMs show power–law scaling with the number of samples.

6. Conclusions

In this work we discussed in depth the use of combinations of lists of features (instead of averaging individual importances) in SVM–RFE for feature selection on multiclass problems. Using an appropriate mathematical framework we introduced 6 different methods to produce the final ranking of features starting from a set of ranked features list produced by each binary problem. We evaluated them in a series of artificial and real world datasets.

Our first conclusion is that the OVO strategy should be preferred over OVA for multiclass feature selection. Probably the higher number of binary problems in OVO helps in filtering out some noisy features that receive high rankings from just one or only a few binary problems, a similar beneficial effect to the use of ensembles in general.

Our second conclusion is that, overall, the K-First method is the most consistent one in selecting subsets of relevant features that lead to smaller classification errors. The idea is well–known in the document retrieval literature, only considers the top k values of each list, and adapts efficiently to feature selection. We showed with several artificial and real–world datasets that this new method is superior to the typical weights averaging that is implemented by default in all current Machine Learning libraries.

Finally, two other methods also showed good results but present some drawbacks. The Best Ranking strategy is simple and efficient, but can lose performance on some problems, such as Artificial-2. Also, the use of a single value to characterize the behavior of a feature can give high rankings to noisy features by chance. The Schulze strategy, based on voting theory, shows a very good performance on some artificial datasets but does not compare well on real–world ones, and is by far the most complex and time–consuming strategy out of the six methods under evaluation.

Overall, the new methods were designed for problems with class-specific features, which is where they show their best performance. As they employ the OVO strategy, they are also resistant to noisy features. Filtered domains, with lots of low-relevance features and little noise like our gene expression datasets, seem to represent a more challenging domain for our new methods.

Work in progress includes a more extensive evaluation and the use of a penalty term to help discard correlated features.

References

Allwein, E. L., Schapire, R. E., & Singer, Y. (2000). Reducing multiclass to binary: A unifying approach for margin classifiers. *Journal of machine learning research*, *1*, 113–141.

Breiman, L. (2001). Random forests. *Machine learning*, *45*, 5–32.

Cappellin, L., Soukoulis, C., Aprea, E., Granitto, P., Dallabetta, N., Costa, F., Viola, R., Märk, T. D., Gasperi, F., & Biasioli, F. (2012). Ptr-tof-ms and data mining methods: a new tool for fruit metabolomics. *Metabolomics*, *8*, 761–770.

Crammer, K., & Singer, Y. (2001). On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of machine learning research*, *2*, 265–292.

Del Pulgar, J. S., Soukoulis, C., Carrapiso, A., Cappellin, L., Granitto, P., Aprea, E., Romano, A., Gasperi, F., & Biasioli, F. (2013). Effect of the pig rearing system on the final volatile profile of iberian dry-cured ham as detected by ptr-tof-ms. *Meat science*, *93*, 420–428.

Dittman, D. J., Khoshgoftaar, T. M., Wald, R., & Napolitano, A. (2013). Classification performance of rank aggregation techniques for ensemble gene selection. In *FLAIRS Conference*.

Duan, K.-B., Rajapakse, J. C., & Nguyen, M. N. (2007). One-versus-one and one-versus-all multiclass svm-rfe for gene selection in cancer classification. In *European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics* (pp. 47–56). Springer.

Fabris, A., Biasioli, F., Granitto, P. M., Aprea, E., Cappellin, L., Schuhfried, E., Soukoulis, C., Märk, T. D., Gasperi, F., & Endrizzi, I. (2010). Ptr-tof-ms and data-mining methods for rapid characterisation of agro-industrial samples: influence of milk storage conditions on the volatile compounds profile of trentingrana cheese. *Journal of mass spectrometry*, *45*, 1065–1074.

Forman, G. (2003). An extensive empirical study of feature selection metrics for text classification. *Journal of machine learning research*, *3*, 1289–1305.

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A. et al. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *science*, *286*, 531–537.

Granitto, P. M., Furlanello, C., Biasioli, F., & Gasperi, F. (2006). Recursive feature elimination with random forest for ptr-ms analysis of agroindustrial products. *Chemometrics and Intelligent Laboratory Systems*, *83*, 83–90.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, *3*, 1157–1182.

Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine learning*, *46*, 389–422.

Haury, A.-C., Gestraud, P., & Vert, J.-P. (2011). The influence of feature selection methods on accuracy, stability and interpretability of molecular signatures. *PloS one*, *6*, e28210.

Ho, T. K., Hull, J. J., & Srihari, S. N. (1994). Decision combination in multiple classifier systems. *IEEE transactions on pattern analysis and machine intelligence*, *16*, 66–75.

Hsu, C.-W., & Lin, C.-J. (2002). A comparison of methods for multiclass support vector machines. *IEEE transactions on Neural Networks*, *13*, 415–425.

Hua, J., Tembe, W. D., & Dougherty, E. R. (2009). Performance of feature-selection methods in the classification of high-dimension data. *Pattern Recognition*, *42*, 409–424.

Jurman, G., Merler, S., Barla, A., Paoli, S., Galea, A., & Furlanello, C. (2008). Algebraic stability indicators for ranked lists in molecular profiling. *Bioinformatics*, *24*, 258–264.

Kanth, N. R., & Saraswathi, S. (2015). Efficient speech emotion recognition using binary support vector machines & multiclass svm. In *Computational Intelligence and Computing Research (ICCIC), 2015 IEEE International Conference on* (pp. 1–6). IEEE.

Kira, K., & Rendell, L. A. (1992). The feature selection problem: Traditional methods and a new algorithm. In *AAAI* (pp. 129–134). volume 2.

Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. *Artificial intelligence*, *97*, 273–324.

Leek, J. T., Scharpf, R. B., Bravo, H. C., Simcha, D., Langmead, B., Johnson, W. E., Geman, D., Baggerly, K., & Irizarry, R. A. (2010). Tackling the widespread and critical impact of batch effects in high-throughput data. *Nature Reviews Genetics*, *11*, 733–739.

Lichman, M. (2013). UCI machine learning repository.

Liu, H., Dougherty, E. R., Dy, J. G., Torkkola, K., Tuv, E., Peng, H., Ding, C., Long, F., Berens, M., Parsons, L. et al. (2005). Evolving feature selection. *IEEE Intelligent systems*, *20*, 64–76.

Mohsenzadeh, Y., Sheikhzadeh, H., & Nazari, S. (2016). Incremental relevance sample-feature machine: A fast marginal likelihood maximization approach for joint feature selection and classification. *Pattern Recognition*, *60*, 835–848.

Mohsenzadeh, Y., Sheikhzadeh, H., Reza, A. M., Bathaee, N., & Kalayeh, M. M. (2013). The relevance sample-feature machine: A sparse bayesian learning approach to joint feature-sample selection. *IEEE transactions on cybernetics*, *43*, 2241–2254.

Monti, S., Tamayo, P., Mesirov, J. P., & Golub, T. R. (2003). Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data. *Machine Learning*, *52*, 91–118.

Neumayer, R., Mayer, R., & Nørvåg, K. (2011). Combination of feature selection methods for text categorisation. In *European Conference on Information Retrieval* (pp. 763–766). Springer.

Nguyen, M. H., & De la Torre, F. (2010). Optimal feature selection for support vector machines. *Pattern recognition*, *43*, 584–591.

Nuray, R., & Can, F. (2006). Automatic ranking of information retrieval systems using data fusion. *Information processing & management*, *42*, 595–614.

Raies, A. B., & Bajic, V. B. (2016). In silico toxicology: computational methods for the prediction of chemical toxicity. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, *6*, 147–172.

Ramaswamy, S., Tamayo, P., Rifkin, R., Mukherjee, S., Yeang, C.-H., Angelo, M., Ladd, C., Reich, M., Latulippe, E., Mesirov, J. P. et al. (2001). Multiclass cancer diagnosis using tumor gene expression signatures. *Proceedings of the National Academy of Sciences*, *98*, 15149–15154.

Saari, D. (2001). *Chaotic elections!: A mathematician looks at voting*. American Mathematical Soc.

Schulze, M. (2011). A new monotonic, clone-independent, reversal symmetric, and condorcet-consistent single-winner election method. *Social Choice and Welfare*, *36*, 267–303.

Statnikov, A., Aliferis, C. F., Tsamardinos, I., Hardin, D., & Levy, S. (2005). A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis. *Bioinformatics*, *21*, 631–643.

Uysal, A. K. (2016). An improved global feature selection scheme for text classification. *Expert systems with Applications*, *43*, 82–92.

Vapnik, V. (2013). *The nature of statistical learning theory*. Springer science & business media.

Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T., & Vapnik, V. (2000). Feature selection for svms. In *Proceedings of the 13th International Conference on Neural Information Processing Systems* (pp. 647–653). MIT Press.

Weston, J., & Watkins, C. (1999). Support vector machines for multi-class pattern recognition. In *ESANN* (pp. 219–224). volume 99.

You, W., Yang, Z., & Ji, G. (2014). Feature selection for high-dimensional multi-category data using pls-based local recursive feature elimination. *Expert Systems with Applications*, *41*, 1463–1475.

Young, H. P. (1988). Condorcet's theory of voting. *The American Political Science Review*, (pp. 1231–1244).

Zhou, N., & Wang, L. (2016). Processing bio-medical data with class-dependent feature selection. In *Advances in Neural Networks* (pp. 303–310). Springer.

Zhou, W., & Dickerson, J. A. (2014). A novel class dependent feature selection method for cancer biomarker discovery. *Computers in biology and medicine*, *47*, 66–75.

Zhou, X., & Tuck, D. P. (2007). Msvm-rfe: extensions of svm-rfe for multiclass gene selection on dna microarray data. *Bioinformatics*, *23*, 1106–1114.

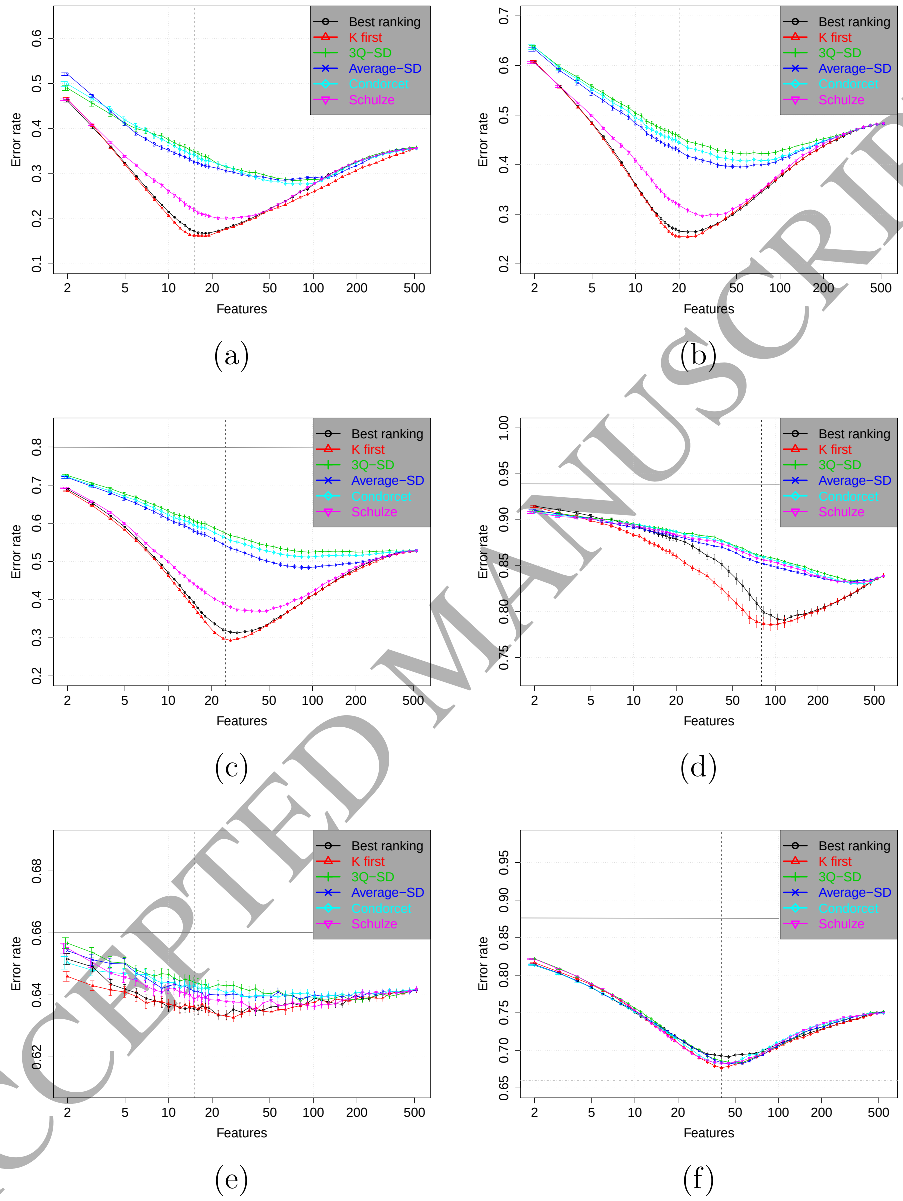


Figure 4: Error curves for the six methods on some artificial datasets. Details are similar to Figure 1. (a) Artificial-1-3C (chance error 0.666) (b) Artificial-1-4C (chance error 0.75) (c) Artificial-1-5C (d) Artificial-1-16C (e) Artificial-3-3C (f) Artificial-3-8C

Dataset	F	S	C	Description
Apple	714	150	15	Mass spectrometry measurements over 15 varieties of Apple clones (Cappellin et al., 2012)
Cheese	117	72	8	Mass spectrometry measurements over 8 varieties of Italian cheese (Fabris et al., 2010)
Ham	1338	382	11	Mass spectrometry measurements over 11 varieties of Iberian Hams (Del Pulgar et al., 2013)
Strawberry	232	233	9	Mass spectrometry measurements over 9 varieties of strawberries (Granitto et al., 2006)
Multi-F	649	2000	10	Features of handwritten numerals extracted from a collection of Dutch utility maps (Lichman, 2013)
Libras	90	360	15	Diverse hand movements from the Brazilian hands language (Lichman, 2013)
Robot1	90	88	4	Robot Execution Failures Data Set, from UCI. Failures in approach to grasp position (Lichman, 2013)
Robot3	90	47	4	Same as Robot1. Position of part after a transfer failure
Robot4	90	117	3	Same as Robot1. Failures in approach to ungrasp position
Robot5	90	164	5	Same as Robot1. Failures in motion with part
Leukemia	985	248	6	Gene expression of Bone marrow samples with 6 subtypes of Leukemia (Monti et al., 2003)
Lung	1000	197	4	Gene expression of lung tissues with 4 cancer types (Monti et al., 2003)
CNS	989	42	5	Gene expression of 5 tumor types of the central nervous system (Monti et al., 2003)
Novartis	1000	103	4	Gene expression of tissue samples from 4 distinct cancer types (Monti et al., 2003)

Table 7: Details on the 14 real–world datasets used in this work. Columns show the number of features (F), samples (S) and classes (C).

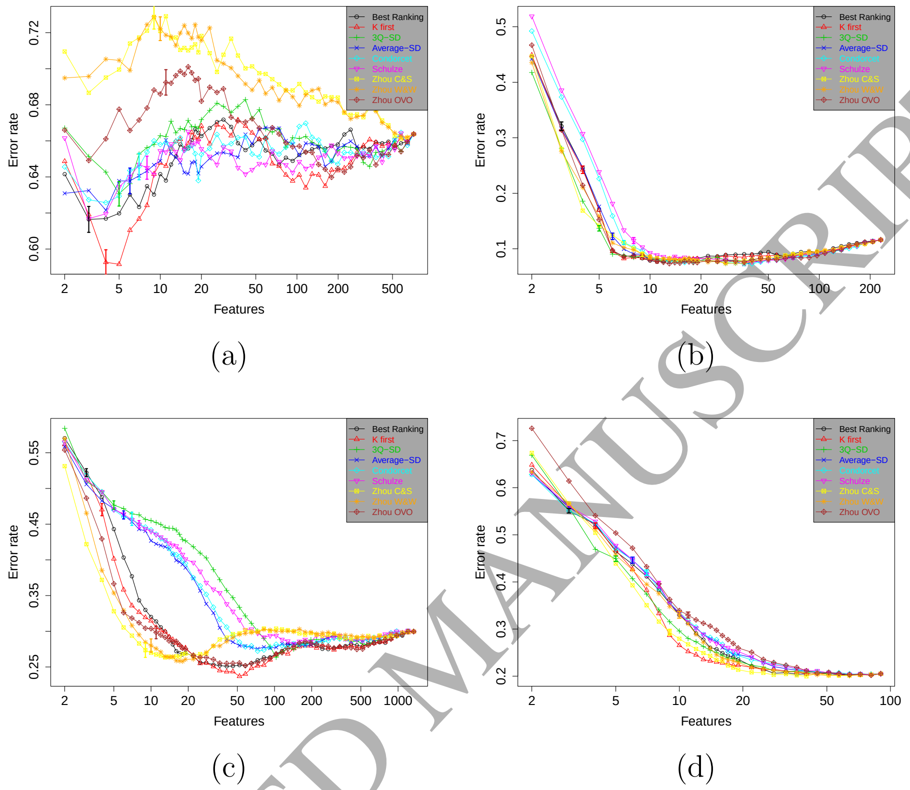


Figure 5: Results on some real world datasets. Details are similar to Figure 1. (a) Apples. (b) Strawberry. (c) Ham. (d) Libras.

Group	3Q-SD	Av-SD	Best	K-First	Cond.	Schulze	Z. C&S	Z. W&W	Z. OVO
UCI-10 Feat	7	5	5	**8**	3	2	0	2	5
MS -10 Feat	0	3	7	**8**	2	1	5	4	6
GE -10 Feat	1	3	6	4	2	0	7	5	**8**
UCI-20 Feat	5	2	6	**8**	2	0	3	7	3
MS -20 Feat	0	2	7	**8**	3	1	4	5	6
GE -20 Feat	1	6	5	4	2	0	**8**	3	7

Table 8: Rankings of methods (higher is better) counting the number of times that one method outperforms another on each domain and number of selected features.

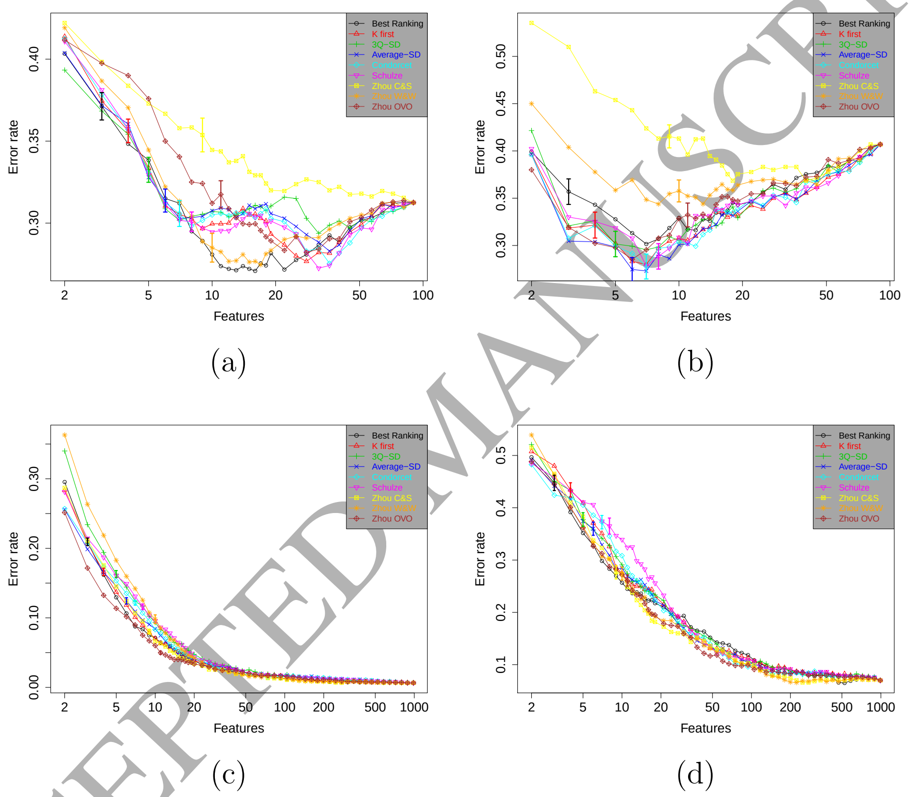
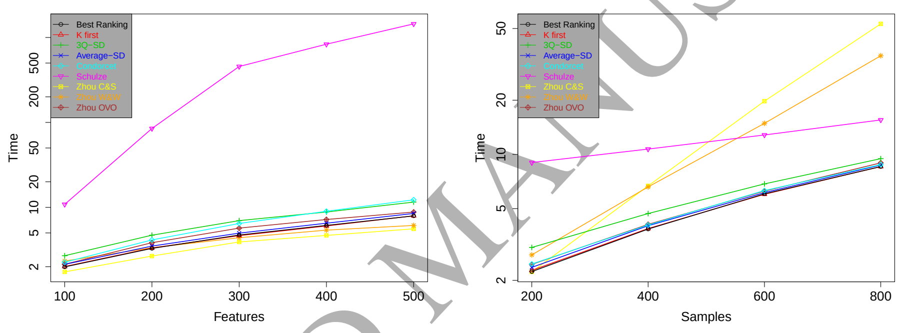


Figure 6: Results on some real world datasets. Details are similar to Figure 1. (a) Robot1. (b) Robot3. (c) Leukemia. (d) CNS.



(a) (b)

Figure 7: Comparison of running times for all methods evaluated in this work as a function of (a) the number of features and (b) the number of samples. Times (in seconds) are in log–scale.