Kernel methods for heterogeneous feature selection

Jérôme Paul[a,*], Roberto D'Ambrosio[a], Pierre Dupont[a]

[a] *Université catholique de Louvain – ICTEAM/Machine Learning Group, Place Sainte Barbe 2 bte L5.02.01, B-1348 Louvain-la-Neuve, Belgium*

Abstract

This paper introduces two feature selection methods to deal with heterogeneous data that include continuous and categorical variables. We propose to plug a dedicated kernel that handles both kinds of variables into a Recursive Feature Elimination procedure using either a non-linear SVM or Multiple Kernel Learning. These methods are shown to offer state-of-the-art performances on a variety of high-dimensional classification tasks.

Keywords: heterogeneous feature selection, kernel methods, mixed data, multiple kernel learning, support vector machine, recursive feature elimination

1. Introduction

Feature selection is an important preprocessing step in machine learning and data mining as increasingly more data are available and problems with hundreds or thousands of features have become common. Those high dimensional data appear in many areas such as gene expression array analysis, text processing of internet documents, economic forecasting, etc. Feature selection allows domain experts to interpret a decision model by reducing the number of variables to analyze. It also reduces training and classification times as well as measurement and storage requirements.

To the best of our knowledge, little effort has been dedicated to develop feature selection methods tailored for datasets with both categorical and

*Corresponding author

Email addresses: jerome.paul@uclouvain.be (Jérôme Paul), roberto.dambrosio@uclouvain.be (Roberto D'Ambrosio), pierre.dupont@uclouvain.be (Pierre Dupont)

[1] http://www.ucl.ac.be/mlg/

numerical values. Such heterogeneous data are found in several applications. For instance, in the medical domain, high dimensional continuous feature sets (e.g. gene expression data) are typically considered along with a few clinical features. These features can be continuous (e.g. blood pressure) or categorical (e.g. sex, smoker vs non-smoker). To highlight important variables, a naive approach would transform heterogeneous data into either fully continuous or categorical variables before applying any standard feature selection algorithm. To get a continuous dataset, categorical variables can be encoded as numerical values. The specific choice of such numerical values is however arbitrary. It introduces an artificial order between the feature values and can lead to largely different distance measures between instances [1].

A standard approach relies on a multivariate numerical encoding, such as the disjunctive encoding, to represent categorical variables. For instance, a feature having 3 categories as possible values could be encoded by considering 3 new features instead: $(1, 0, 0)$, $(0, 1, 0)$ and $(0, 0, 1)$. However, they need specific approaches, such as group lasso [2], to correctly handle feature selection at the granularity of the original features.

The discretization of continuous features is a common alternative to represent categorical and numerical features in a similar space. Such approach comes at the price of making the selection highly sensitive to the specific discretization [1].

A natural alternative would consider tree ensemble methods such as Random Forests (RF), since they can be grown from both types of variables and these methods perform an embedded selection. RF were however shown to bias the selection towards variables with many values [3]. The cForest method has been introduced to correct this bias [3] but its computational time is drastically increased and becomes prohibitive when dealing with thousands of features[2].

In this paper we propose two kernel based methods for feature selection. They are conceptually similar to disjunctive encoding while keeping original features throughout the whole selection process. In both approaches, the selection is performed by the Recursive Feature Elimination (RFE) [4] mechanism that iteratively ranks variables according to their importances. We propose to extract those feature importances from two different kernel meth-

[2]In each node of each tree of the forest, a conditional independence permutation test needs to be performed to select the best variable instead of a simple Gini evaluation.

ods : the Support Vector Machine (SVM) and the Multiple Kernel Learning (MKL), with a dedicated heterogeneous kernel. We use the clinical kernel [5], that handles both kinds of features in classification tasks.

The remainder of this document is organized as follows. Section 2 describes the two proposed methods. Section 3 briefly presents competing approaches we compare to in our experiments. The experimental setting is presented in Section 4. Results are discussed in Section 5. Finally, Section 6 concludes this work.

2. Material and Methods

This section presents the different building blocks that compose our two heterogeneous feature selection methods. Recursive Feature Elimination (RFE), the main feature selection mechanism, is presented in Section 2.1. It internally uses a global variable ranking for both continuous and categorical features. This ranking is extracted from two kernel methods (Support Vector Machine and Multiple Kernel Learning) that use a dedicated heterogeneous kernel called the *clinical kernel* (Section 2.2). Section 2.3 details how to obtain a feature ranking from a non-linear SVM. Finally, Section 2.4 sketches Multiple Kernel Learning, which offers an alternative way to rank variables with the clinical kernel.

2.1. Recursive feature elimination

RFE [4] is an embedded backward elimination strategy that iteratively builds a feature ranking by removing the least important features in a classification model at each step. Following [6], a fixed proportion of 20 % of features is dropped at each iteration. The benefit of such a fixed proportion is that the actual number of features removed at each step gradually decreases till be rounded to 1, allowing a finer ranking for the most important features. This iterative process is pursued till all variables are ranked. The number of iterations automatically depends on the total number p of features to be ranked while following this strategy. RFE is most commonly used in combination with a linear SVM from which feature weights are extracted. However, it can be used with any classification model from which individual feature importance can be deduced. A general pseudo-code for RFE is given in Algorithm 1.

$R \leftarrow$ empty ranking
$F \leftarrow$ set of all features
while F *is not empty* **do**
  train a classifier m using F
  extract variable importances from m
  remove the 20% least important features from F
  put those features on top of R
end
return R

Algorithm 1: Recursive Feature Elimination

2.2. Clinical kernel

The so-called clinical kernel proposed in [5] was shown to outperform a linear kernel for classifying heterogeneous data. It averages univariate subkernels [7] defined for each feature.

$$k(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{p} \sum_{f=1}^{p} k_f(x_{if}, x_{jf}) \tag{1}$$

$$k_f(a, b) = \begin{cases} I(a = b) & \text{if } f \text{ is categorical} \\ \frac{(max_f - min_f) - |a-b|}{max_f - min_f} & \text{if } f \text{ is continuous} \end{cases} \tag{2}$$

where $\mathbf{x}_i$ is a data point in p dimensions, x_{if} is the value of $\mathbf{x}_i$ for feature f, I is the indicator function, a and b are scalars and max_f and min_f are the maximum and minimum values observed for feature f. One can note that summing kernels simply amounts to concatenating variables in the kernel induced space.

Given two data points, the subkernel values lie between 0, when the feature values are farthest apart, and 1 when they are identical, similarly to the gaussian kernel. The clinical kernel is basically an unweighted average of overlap kernels [8] for categorical features and triangular kernels [9, 10] for continuous features. The overlap kernel can also be seen as a rescaled l_1-norm on a disjunctive encoding of the categorical variables. The clinical kernel assumes the same importance to each original variable. We show here the benefit of adapting this kernel for heterogeneous feature selection.

2.3. Feature importance from non-linear Support Vector Machines

The Support Vector Machine (SVM) [11] is a well-known algorithm that is widely used to solve classification problems. It looks for the largest margin

hyperplane that distinguishes between samples of different classes. In the case of a linear SVM, one can measure the feature importances by looking at their respective weights in the hyperplane. When dealing with a non-linear SVM, we can instead look at the variation in margin size $\frac{1}{\|\mathbf{w}\|}$. Since the larger the margin, the lower the generalization error (at least in terms of bound), a feature that does not decrease much the margin size is not deemed important for generalization purposes. So, in order to measure feature importances with a non-linear SVM, one can look at the influence on the margin of removing a particular feature [12].

The margin is inversely proportional to

$$W^2(\alpha) = \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{w}\|^2 \tag{3}$$

where α_i and α_j are the dual variables of a SVM, y_i and y_j the labels of $\mathbf{x}_i$ and $\mathbf{x}_j$, out of n training examples, and k a kernel. Therefore, the importance of a particular feature f can be approximated without re-estimating α by the following formula:

$$J_{SVM}(f) = |W^2(\alpha) - W^2_{(-f)}(\alpha)| \tag{4}$$

$$W^2_{(-f)}(\alpha) = \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i^{-f}, \mathbf{x}_j^{-f}) \tag{5}$$

where $\mathbf{x}_i^{-f}$ is the i-th training example without considering the feature f. In Equation (5), the α's are kept identical to those in Equation (3). This is a computationally efficient approximation originally proposed in [12]. The feature importance is thus evaluated with respect to the separating hyperplane in the current feature space and hence the current decision function.

Updating $k(\mathbf{x}_i, \mathbf{x}_j)$ to $k(\mathbf{x}_i^{-f}, \mathbf{x}_j^{-f})$ is pretty efficient and straightforward with the clinical kernel (Section 2.2). There is no need to recompute the sum of all subkernels but one only has to remove k_f (Equation (2)) and normalize accordingly. Removing one such sub-kernel is equivalent to removing features in the projected space, which is similar to what is done with a linear kernel.

In this work, we propose to combine the J_{SVM} feature importance (Equation (4)) with the RFE mechanism in order to provide a full ranking of the features. This method will be referred to as RFE^{SVM}.

2.4. Feature importance from Multiple Kernel Learning

MKL [13] learns an appropriate linear combination of M basis kernels, each one possibly associated to a specific input variable, as well as a discriminant function. The resulting kernel is a weighted combination of different input kernels.

$$k(\mathbf{x}_i, \mathbf{x}_j) = \sum_{m=1}^{M} \mu_m k_m(\mathbf{x}_i, \mathbf{x}_j) \qquad \text{s.t. } \mu_m \geq 0 \tag{6}$$

Summing kernels is equivalent to concatenating the respective feature maps $\psi_1, \ldots, \psi_m$ induced by those kernels. The associated decision function $f(\mathbf{x})$ is a generalized linear model in the induced space:

$$f(\mathbf{x}) = \sum_{m=1}^{M} \sqrt{\mu_m} \mathbf{w}_m^T \psi_m(\mathbf{x}) + b \tag{7}$$

where μ_m, $\mathbf{w}_m$ and ψ_m are respectively the kernel weight, feature weight and explicit feature map corresponding to the m-th kernel, and b a bias term. Those parameters are estimated by minimizing the following objective

$$\underset{\mathbf{w}, b, \boldsymbol{\mu} \geq 0}{\operatorname{argmin}} C \sum_{i=1}^{n} \ell(f(\mathbf{x}_i), y_i) + \frac{1}{2} \sum_{m=1}^{M} \|\mathbf{w}_m\|_2^2 \text{ such that } \|\boldsymbol{\mu}\|_2^2 \leq 1 \tag{8}$$

where $C > 0$ and ℓ denotes the hinge loss $\ell(f(\mathbf{x}), y) = \max\{0, 1 - yf(\mathbf{x})\}$. We note that the kernel weight vector $\boldsymbol{\mu}$ is l_2-regularized in contrast to MKL approaches using sparsity inducing norms [14]. Indeed, non-sparse MKL has been shown to be more effective on various computational biology problems [15]. It is also more convenient in our context since we interpret $|\mu_m|$ as a feature importance measure and look for a full ranking of all features.

In this work, we adapt the clinical kernel (Equation (2)) with MKL to learn a non-uniform combination of the basis kernels, each one associated to a single feature. As we can see in Equation (7), μ_f reflects the influence of kernel k_f in the decision function [13]. μ_f can thus be seen as the importance $J_{MKL}(f)$ of feature f.

The combination of RFE with this feature importance extracted from MKL will be referred to as RFE^{MKL}. It specifically uses the kernel weights $|\mu_f|$ as feature importance value to eliminate at each iteration a prescribed fraction of the least relevant features.

3. Competing approaches

This section presents the three competing methods we compare to in the experiments: Random Forest [16] and two variants of Hybrid Feature Selection [1].

The Random Forest (RF) algorithm builds an ensemble of T decision trees. Each one is grown on a bootstrap sample of the dataset. The subset of data points that are used to build a particular tree forms its bag. The remaining set of points is its out-of-bag. To compute variable importances, Breiman [16] proposes a permutation test. It uses the out-of-bag samples to estimate how much the predictive performances of the RF decrease when permuting a particular variable. The bigger the drop in accuracy, the higher the variable importance. In order to obtain good and stable feature selection from RF, a large ensemble of 10,000 trees ($RF10000$) is considered according to the analysis in [17].

An alternative method performs a greedy forward selection aggregating separate rankings for each type of variables into a global ranking [1]. The authors report improved results over those of the method proposed in [18], which is based on neighborhood relationships between heterogeneous samples. Out of a total of p variables, categorical and continuous features are first ranked independently. Mutual information (MI) was originally proposed for those rankings but a reliable estimate of MI is difficult to obtain whenever fewer samples than dimensions are available. Instead we use the p-values of a t-test to rank continuous features and of a Fisher exact test for categorical ones. The two feature rankings are then combined into a global ranking by iteratively adding the first categorical or continuous variable that maximizes the predictive performance of a Naive Bayes or a 5-NN classifier (consistently with the choices made in [1]). The NN classifier uses the Heterogeneous Euclidian-Overlap Metric [19] between pairs of instances as follows:

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{f=1}^{p} d_f(x_{if}, x_{jf})^2} \tag{9}$$

$$d_f(a,b) = \begin{cases} I(a \neq b) & \text{if } f \text{ is categorical} \\ \frac{|a-b|}{max_f - min_f} & \text{if } f \text{ is continuous} \end{cases} \tag{10}$$

$$= 1 - k_f(a,b) \tag{11}$$

This metric is closely related to the clinical kernel (Equation (2)). For each feature, d_f takes value 0 for identical points and value 1 for points that are farthest apart in that dimension. We refer to these approaches as HFS^{NB} and HFS^{5NN} in the sequel.

4. Experiments

In order to compare the five feature selection methods, we report predictive performances of classifiers built on selected variables as well as quality measures on those feature sets. A statistical analysis is also performed to assess if there are significant differences between the performances of the various methods. This section presents the experimental protocol, the various evaluation metrics and the datasets that we use in our experiments.

4.1. Experimental protocol

When a sufficient amount of data is available, 10-fold cross validation (10-CV) provides a reliable estimate of model performances [20]. However, it may lead to inaccurate estimates on small-sized datasets, due to a higher variability in the different folds. We thus make use of a resampling strategy consisting of 200 random splits of the data into training (90%) and test (10%). Such a protocol has the same training/test proportions as 10-CV but benefits from a larger number of tests. It also keeps the training size sufficiently large so as to report performances close enough to those of a model estimated on the whole available data.

For each data partition, the training set is used to rank features and build predictive models using different numbers of features. The ranking is recorded and predictive performances are measured while classifying the test set. Average predictive performances are reported over all test folds and the stability of various signature sizes is computed from the 200 feature rankings. The average number of selected categorical features is also computed for each signature size. This number does not reflect a specific performance value of the feature selection methods but rather gives some insight into how they deal with the selection of heterogeneous variables.

Whenever a SVM is trained with the clinical kernel, the regularization parameter is fixed to a predefined value estimated from preliminary experiments on independent datasets. Such a value is set to 0.1 for the feature selection itself and to 10 when learning a final classifier on the selected features.

4.2. Performance metrics

Predictive performances are reported here in terms of balanced classification rate (BCR), which is the average between sensitivity and specificity. These metrics are particularly popular in the medical domain and BCR, unlike AUC, easily generalizes to multi-class with unbalanced priors. For binary classification, it is defined as follows :

$$BCR = \frac{1}{2}\left(\frac{TP}{P} + \frac{TN}{N}\right) \tag{12}$$

where TP (resp. TN) is the number of true positives (resp. negatives) and P (resp. N) the number of positive (resp. negative) samples in the dataset.

Selection stability is assessed here through the Kuncheva's index (KI) [21] which measures to which extent K sets of s selected features share common elements.

$$KI(\{S_1, ..., S_K\}) = \frac{2}{K(K-1)} \sum_{i=1}^{K-1} \sum_{j=i+1}^{K} \frac{|S_i \cap S_j| - \frac{s^2}{p}}{s - \frac{s^2}{p}} \tag{13}$$

where p is the total number of features and $\frac{s^2}{p}$ is a correction for the random chance that 2 feature sets S_i and S_j share common features. KI takes values in $(-1, 1]$. A value of 0 indicates random selection. The larger KI, the larger the number of commonly selected features.

In order to globally compare the five feature selection methods, a Friedman statistical test [22] is performed across all datasets and all feature set sizes. A low p-value indicates that there is indeed a difference between the various algorithm performances. In that case, a Nemenyi post-hoc test [22] is performed to find out which methods perform significantly differently than others.

4.3. Datasets

We report results on 7 binary classification datasets briefly described in Table 1 in terms of number of features and class priors. The Arrhythmia [23] dataset aims at distinguishing between the presence or absence of cardiac arrhythmia from features extracted from electrocardiograms. The Bands [23] dataset tackles the problem of band (grooves) detection on cylinders engraved by rotogravure printing. It consists of physical measurements and technical printing specifications. The task associated to the Heart [23] dataset is to

Name	Continuous features	Categorical features	Class priors
Arrhythmia [23]	198	64	245/185
Bands [23]	20	14	312/228
Heart [23]	6	7	164/139
Hepatitis [23]	6	13	32/123
Housing [24]	15	2	215/291
Rheumagene [25]	100	3	28/21
van't Veer [26]	4353	2	44/33

Table 1: Datasets overview

detect the presence of a heart disease in the patient. Variables come from clinical measurements. The Hepatitis [23] dataset is about predicting survival to hepatitis from clinical variables. The goal of the Housing [24] dataset is to evaluate the median value of owner-occupied homes from local statistics. The two classes are defined by a cutoff at $20,000. The Rheumagene [25] dataset aims at diagnosing arthritis at a very early stage of the disease. Genomic variables are provided along with 3 clinical variables. Finally, the van't Veer [26] dataset tackles a breast cancer prognosis problem. This very high dimensional dataset consists of genomic features from microarray analysis and seven clinical variables, two of them being categorical.

5. Results and discussion

We compare here RFE^{MKL} and RFE^{SVM} to HFS^{NB}, HFS^{5NN} and RF of 10,000 trees on 7 real-life datasets resulting in more than 7,000 experiments. These methods essentially provide a ranking of the features, without defining specific feature weights[3]. Predictive performances can then be assessed on a common basis for all techniques by selecting all features up to a prescribed rank and estimating a classifier restricted to those features. We use here a non-linear SVM with the clinical kernel reduced to the selected features as final classifier. Other final classifiers such as RF, Naive Bayes or 5-NN offer similar predictive performances and are not reported here.

We compare first all selection techniques across all feature set sizes and

[3]Feature weights are used at each RFE iteration but those weights need not be comparable globally across iterations.

datasets to give a general view of the performances. Choosing a specific number of features is indeed often left to the final user who, for instance, might favor the greater interpretability of a reduced feature set at the price of some predictive performance decrease. Our second analysis focuses on a fixed number of features offering a good trade-off between predictive performances and sparsity.

Figure 1 reports the statistical analysis across all datasets and all feature set sizes using a Friedman test, followed by a Nemenyi post-hoc test. Figures 2, 3, 4 and 5 report more detailed results. They show the predictive performance, the stability of feature selection and the average number of selected categorical features on each signature size of each dataset.

The Friedman test [22] can be seen as a non-parametric equivalent to the repeated-measures ANOVA. It tests whether the methods significantly differ based on their average ranks. In our experiments, it shows significant differences of the predictive performances of the 5 feature selection methods across all datasets and all feature set sizes (p-value $< 10^{-6}$). According to the Nemenyi post-hoc test, (see Figure 1, left), RFE^{MKL} is best ranked (i.e. it has the lowest mean rank) and performs significantly better than HFS^{5NN} and RFE^{SVM} which appear at the end of the ranking. Our data does not show significant differences between the predictive performances of RFE^{MKL}, $RF10000$ and HFS^{NB}. A Friedman test on the feature selection stability also shows highly significant differences (p-value $< 10^{-29}$) between the 5 feature selection approaches. According to a Nemenyi post-hoc test (see Figure 1, right), our RFE approaches are at the bottom of the ranking. RFE^{MKL} is however not significantly less stable than HFS^{NB} and $RF10000$. In addition, the two HFS approaches may have the natural advantage that they are based on filter methods that are more stable than embedded methods [27]. Moreover, the RFs had to be run with a very large number of trees (10,000) to provide a stable feature selection [17]. This leads to increased computational times and heavier models, especially on datasets with a higher number of instances. On the Arrhythmia and Bands datasets, the 200 resamplings require 1.5 more CPU time with RF10000 (single-core implementation in the *randomForest* R-package [28]) than with the RFE methods (in the Shogun [29] implementation of MKL and SVM). On the Housing

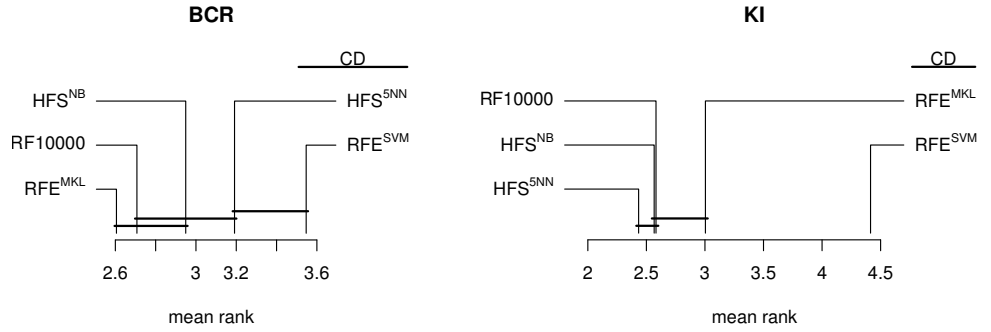

Figure 1: Nemenyi critical difference diagrams [22] : comparison of the predictive performances (BCR) and stability (KI) of the five algorithms over all signature sizes of all datasets. Horizontal black lines group together methods whose mean ranks do not differ significantly. CD represents the rank difference needed to have a 95% confidence that method performances are significantly different.

dataset, the RF implementation is 5 times slower than the RFE methods[4]

The top left graph of Figure 2 shows predictive performances of the five methods on the Arrhythmia dataset. We can see that RFE^{MKL} and $RF10000$ perform best since they avoid to select categorical features which happen to be noisy on this dataset (Figure 2, left). The bottom right plot of Figure 4 reports the average number of categorical features among selected features for the Rheumagene dataset. It shows that all but RFE^{SVM} and HFS^{5NN} select two categorical variables first, leading to already good predictive performances with very few selected variables (top right graph of Figure 4). The third categorical variable is actually never selected since it happens to convey very few information to predict the class label[5]. On the van't Veer dataset, the HFS approaches tend to keep selecting the two categorical variables even when the feature selection is very aggressive (Figure 5, bottom). They show a peak in predictive performances when 5 features are kept (Figure 5, left). However, the best predictive performance (Figure 5, left) is obtained with RFE^{MKL} which selects one of the two categorical variables. It also corresponds to a very good feature selection stability, as shown

[4]Specifically, CPU times were measured on a 2.60 Ghz machine with 8GB Ram memory. On this dataset, RFE^{MKL}, RFE^{SVM}, and $RF10000$ took respectively 23 min, 26 min and 114 min to be run.

[5]Out of 49 samples (28 negative, 21 positive), this variable takes value '0' 46 times and '1' only 3 times.

in the right graph of Figure 5. Finally, on the three high dimensional datasets (Arrhythmia, Rheumagene and van't Veer), RFE^{SVM} is significantly less stable.

We further analyze below the various feature selection methods for a fixed number of selected features. One could indeed be interested in selecting a feature set as small as possible with only a marginal decrease in predictive performances. For each dataset, we choose the smaller feature set size such that the BCR of RFE^{MKL} lies in the 95% confidence interval of the best RFE^{MKL} predictive performance. Those signature sizes are highlighted in Figures 2–5 by vertical dashed lines. A Friedman test on those predictive performances finds significant differences (p-value of 0.008). A Nemenyi post-hoc test (Figure 6, left) shows that the two best ranked methods, RF10000 and RFE^{MKL}, perform significantly better than RFE^{SVM} in terms of BCR. Feature selection stabilities also significantly differ according to a Friedman test (p-value of 0.02). Figure 6 illustrates that the ranking among the five methods is the same for stability and BCR. Those results on a fixed number of features show that the RFE^{MKL} and RF10000 are the two best performing methods without significant differences between them, but at a larger computational cost for the latter.

6. Conclusion and perspectives

We introduce two heterogeneous feature selection techniques that can deal with continuous and categorical features. They combine Recursive Feature Elimination with variable importances extracted from MKL (RFE^{MKL}) or a non-linear SVM (RFE^{SVM}). These methods use a dedicated kernel combining continuous and categorical variables. Experiments show that RFE^{MKL} produces state-of-the-art predictive performances and is as good as competing methods in terms of feature selection stability. It offers results similar to Random Forests with smaller computational times. RFE^{SVM} performs worse than RFE^{MKL}. It also seems less efficient in terms of prediction and stability than competing approaches, even though not significantly different from all competitors.

The two kernel based methods proposed here are among the few existing selection methods that specifically tackle heterogeneous features. Yet, we plan in our future work to improve their stability possibly by resorting to an ensemble procedure [6].

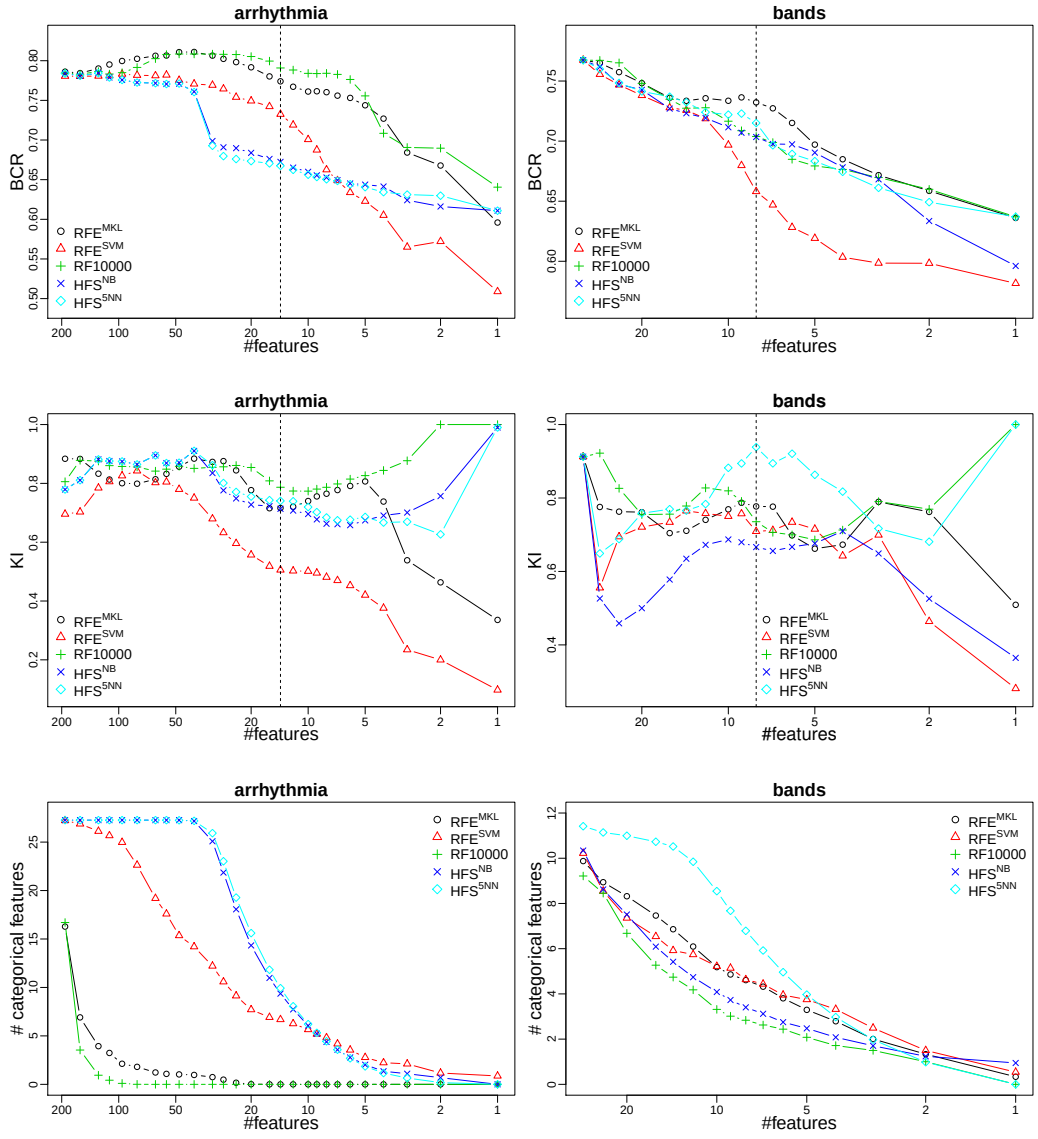

Figure 2: Predictive performances (BCR), feature selection stability (KI) and number of selected categorical features for each signature size of the Arrhythmia and Bands datasets. The dashline defines the minimal number of features to select without loosing much in predictive performances (see text).

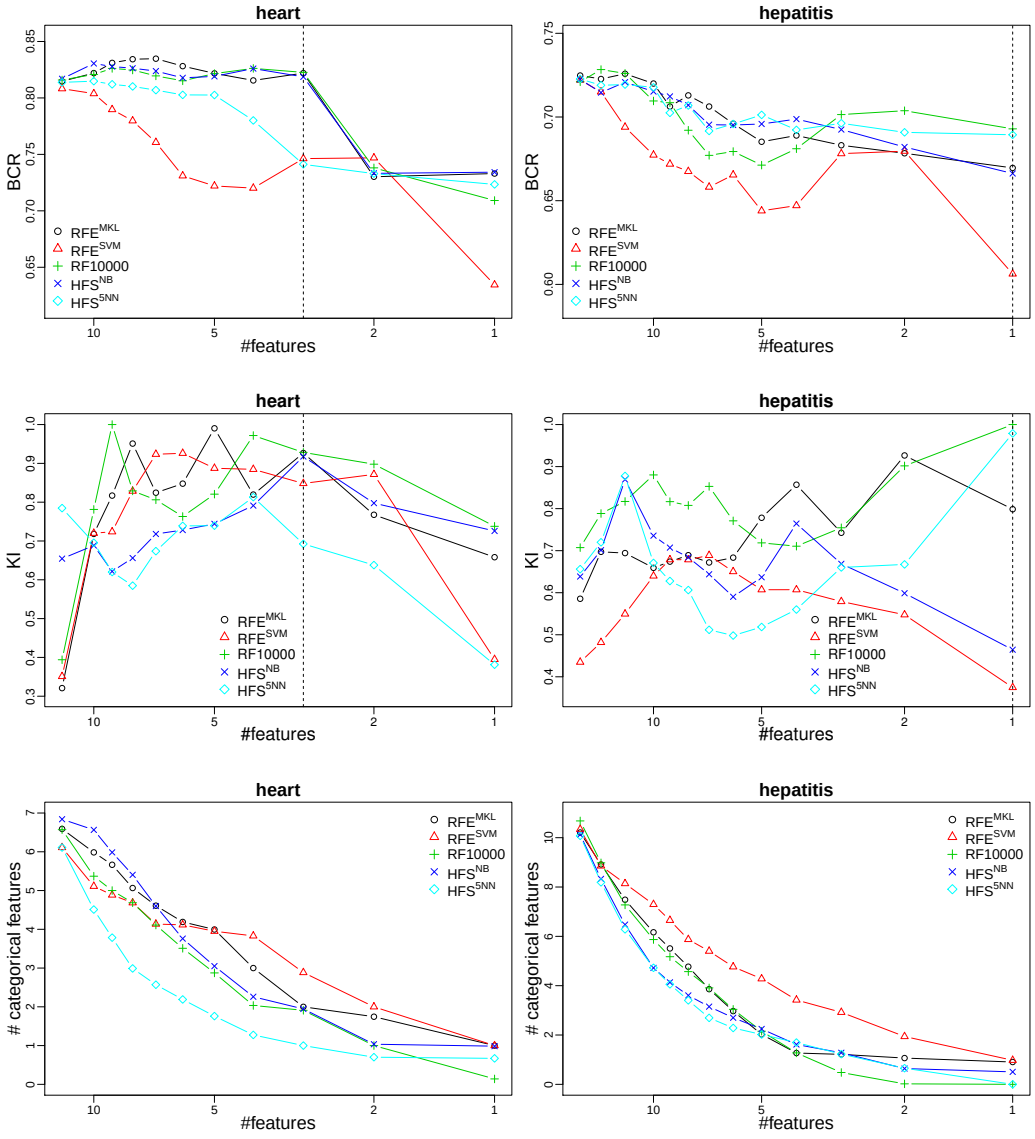

Figure 3: Predictive performances (BCR), feature selection stability (KI) and number of selected categorical features for each signature size of the Heart and Hepatitis datasets. The dashline defines the minimal number of features to select without loosing much in predictive performances (see text).

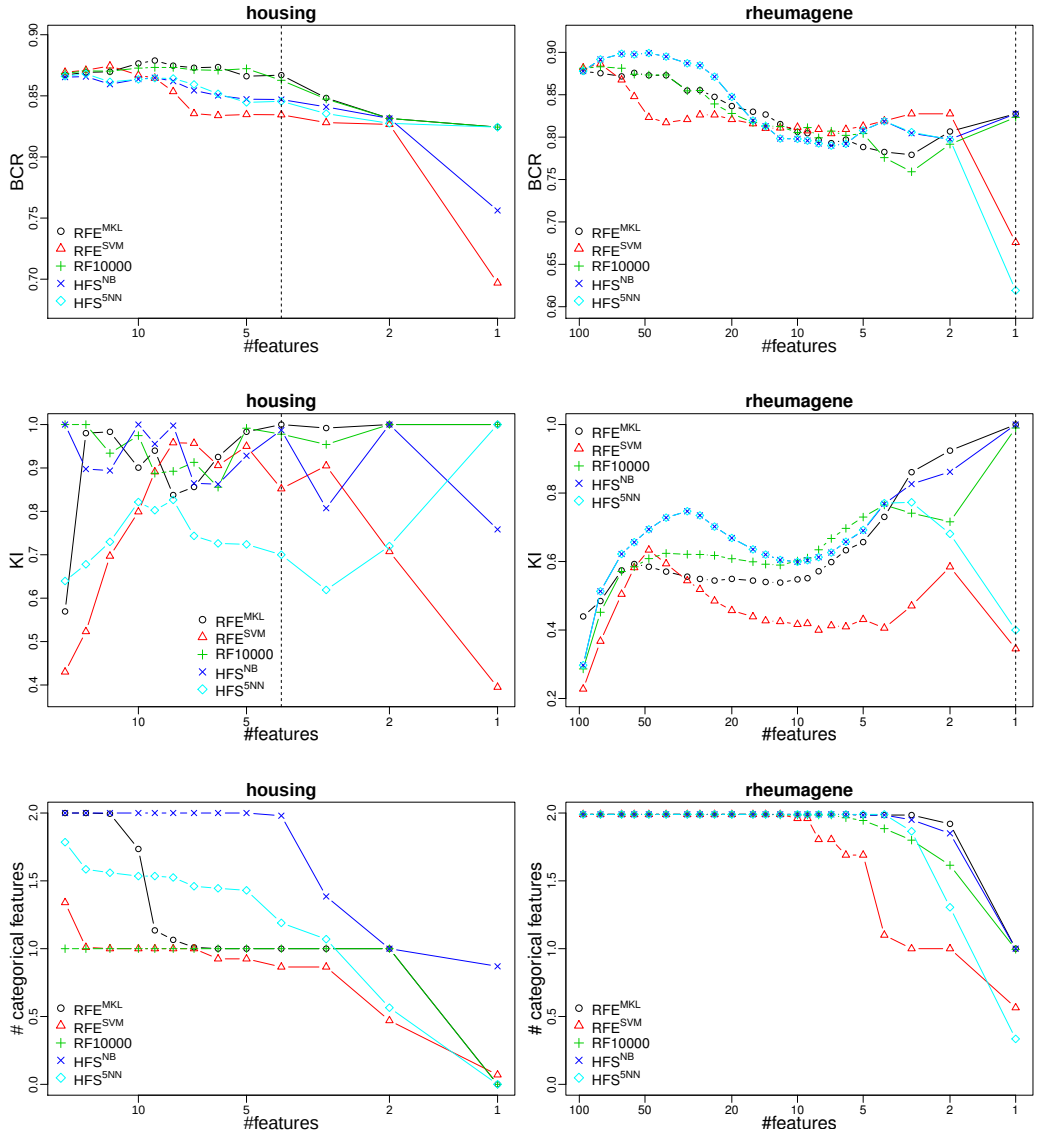

Figure 4: Predictive performances (BCR), feature selection stability (KI) and number of selected categorical features for each signature size of the Housing and Rheumagene datasets. The dashline defines the minimal number of features to select without loosing much in predictive performances (see text).

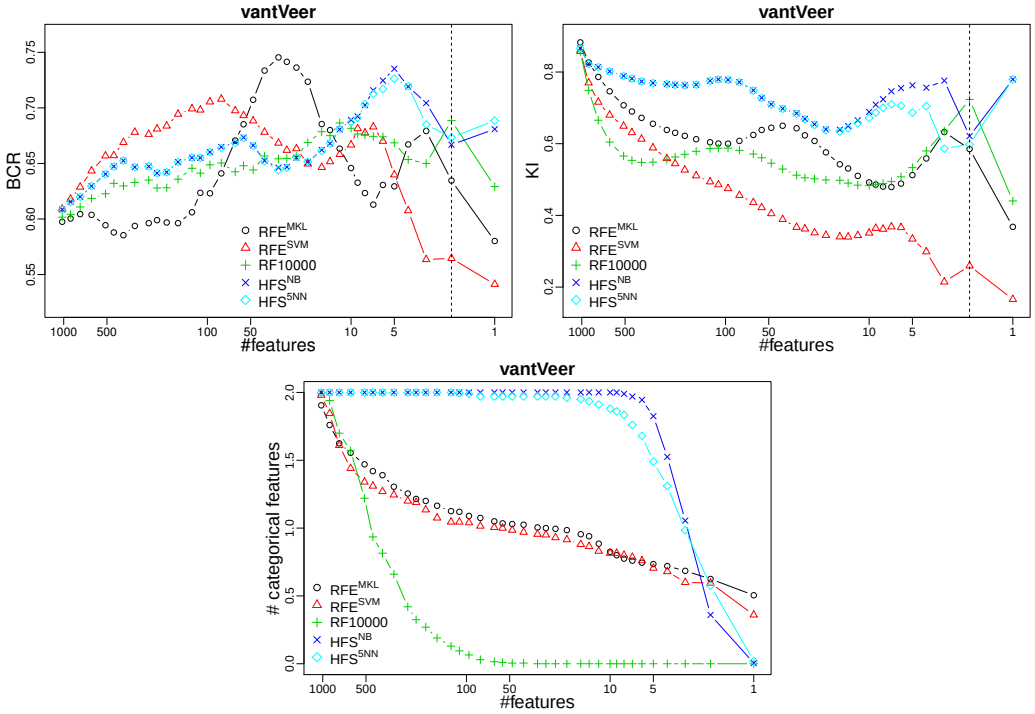

Figure 5: Predictive performances (BCR), feature selection stability (KI) and number of selected categorical features for each signature size of the van't Veer dataset. The dash-line defines the minimal number of features to select without loosing much in predictive performances (see text).

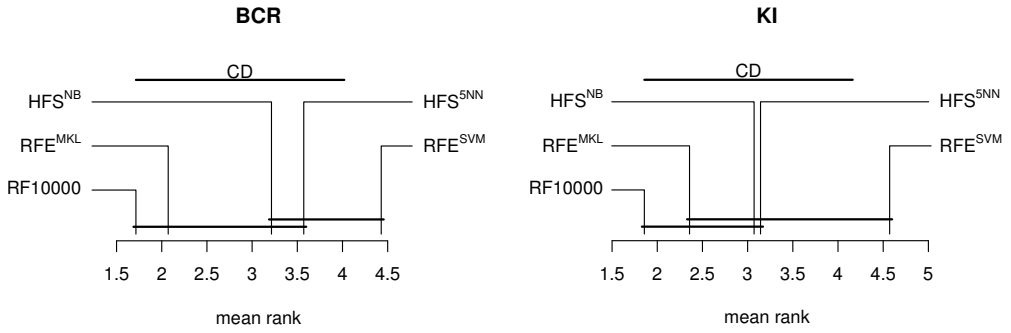

Figure 6: Nemenyi critical difference diagrams [22] : comparison of the predictive performances (BCR) and stability (KI) of the five algorithms for one small signature size in each dataset. Horizontal black lines group together methods whose mean ranks do not differ significantly. CD represents the rank difference needed to have a 95% confidence that methods performances are significantly different.

We observed that the proposed methods run faster than the competing approaches on various datasets. Those differences would be worth to reassess in a further study relying on parallel implementations.

Acknowledgements

Computational resources have been provided by the supercomputing facilities of the Université catholique de Louvain (CISM/UCL) and the Consortium des Equipements de Calcul Intensif en Fédération Wallonie Bruxelles (CECI) funded by the Fonds de la Recherche Scientifique de Belgique (FRS-FNRS).

References

[1] G. Doquire, M. Verleysen, An hybrid approach to feature selection for mixed categorical and continuous data., in: International Conference on Knowledge Discovery and Information Retrieval (KDIR 2011), 2011, pp. 394–401.

[2] M. Yuan, Y. Lin, Model selection and estimation in regression with grouped variables, Journal of the Royal Statistical Society: Series B (Statistical Methodology) 68 (1) (2006) 49–67. doi:10.1111/j.1467-9868.2005.00532.x.
URL http://dx.doi.org/10.1111/j.1467-9868.2005.00532.x

[3] C. Strobl, A.-L. Boulesteix, A. Zeileis, T. Hothorn, Bias in random forest variable importance measures: Illustrations, sources and a solution, BMC Bioinformatics 8 (1) (2007) 25. doi:10.1186/1471-2105-8-25.
URL http://www.biomedcentral.com/1471-2105/8/25

[4] I. Guyon, J. Weston, S. Barnhill, V. Vapnik, Gene selection for cancer classification using support vector machines, Machine Learning 46 (1-3) (2002) 389–422. doi:10.1023/A:1012487302797.

[5] A. Daemen, B. De Moor, Development of a kernel function for clinical data, in: Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE, 2009, pp. 5913–5917. doi:10.1109/IEMBS.2009.5334847.

[6] T. Abeel, T. Helleputte, Y. Van de Peer, P. Dupont, Y. Saeys, Robust biomarker identification for cancer diagnosis with ensemble feature selection methods, Bioinformatics 26 (3) (2010) 392–398. doi:10.1093/bioinformatics/btp630.
URL http://bioinformatics.oxfordjournals.org/content/26/3/392

[7] J. Shawe-Taylor, N. Cristianini, Kernel Methods for Pattern Analysis, Cambridge University Press, New York, NY, USA, 2004.

[8] B. Vanschoenwinkel, B. Manderick, Appropriate kernel functions for support vector machine learning with sequences of symbolic data, in: J. Winkler, M. Niranjan, N. Lawrence (Eds.), Deterministic and Statistical Methods in Machine Learning, Vol. 3635 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2005, pp. 256–280. doi:10.1007/11559887_16.

[9] M. G. Genton, Classes of kernels for machine learning: A statistics perspective, J. Mach. Learn. Res. 2 (2002) 299–312.
URL http://dl.acm.org/citation.cfm?id=944790.944815

[10] C. Berg, J. P. R. Christensen, P. Ressel, Harmonic analysis on semigroups, Springer-Verlag, 1984.

[11] B. Boser, I. Guyon, V. Vapnik, A training algorithm for optimal margin classifiers, in: Proceedings of the fifth annual workshop on Computational learning theory, COLT '92, ACM, New York, NY, USA, 1992, pp. 144–152. doi:10.1145/130385.130401.

[12] I. Guyon, Feature extraction: foundations and applications, Vol. 207, Springer, 2006.

[13] G. Lanckriet, T. De Bie, N. Cristianini, M. Jordan, W. Noble, A statistical framework for genomic data fusion, Bioinformatics 20 (16) (2004) 2626–2635. doi:10.1093/bioinformatics/bth294.
URL http://bioinformatics.oxfordjournals.org/content/20/16/2626

[14] F. Bach, G. Lanckriet, M. Jordan, Multiple kernel learning, conic duality, and the smo algorithm, in: Proc. of the 21st International Conference on Machine Learning, 2004, pp. 41–48.

[15] M. Kloft, U. Brefeld, P. Laskov, K.-R. Müller, A. Zien, S. Sonnenburg, Efficient and accurate lp-norm multiple kernel learning, in: Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, A. Culotta (Eds.), Advances in Neural Information Processing Systems 22, Curran Associates, Inc., 2009, pp. 997–1005.

[16] L. Breiman, Random Forests, Machine Learning 45 (1) (2001) 5–32. doi:10.1023/A:1010933404324.

[17] J. Paul, M. Verleysen, P. Dupont, The stability of feature selection and class prediction from ensemble tree classifiers, in: ESANN 2012, 20th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, 2012, pp. 263–268.

[18] Q. Hu, J. Liu, D. Yu, Mixed feature selection based on granulation and approximation, Knowledge-Based Systems 21 (4) (2008) 294 – 304. doi:http://dx.doi.org/10.1016/j.knosys.2007.07.001.
URL http://www.sciencedirect.com/science/article/pii/S0950705107000755

[19] R. Wilson, T. Martinez, Improved Heterogeneous Distance Functions, Journal of Artificial Intelligence Research 6 (1997) 1–34.
URL http://www.jair.org/media/346/live-346-1610-jair.pdf

[20] R. Kohavi, A study of cross-validation and bootstrap for accuracy estimation and model selection, in: Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2, IJCAI'95, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1995, pp. 1137–1143.
URL http://ijcai.org/Past\%20Proceedings/IJCAI-95-VOL2/PDF/016.pdf

[21] L. Kuncheva, A stability index for feature selection, in: AIAP'07: Proceedings of the 25th conference on Proceedings of the 25th IASTED International Multi-Conference, ACTA Press, Anaheim, CA, USA, 2007, pp. 390–395.

[22] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.
URL http://jmlr.org/papers/volume7/demsar06a/demsar06a.pdf

[23] A. Frank, A. Asuncion, UCI ML repository (2010).
URL http://archive.ics.uci.edu/ml

[24] F. Leisch, E. Dimitriadou, mlbench: Machine Learning Benchmark Problems, R package version 2.1-1 (2010).

[25] I. Focant, D. Hernandez-Lobato, J. Ducreux, P. Durez, A. Toukap, D. Elewaut, F. Houssiau, P. Dupont, B. Lauwerys, Feasibility of a molecular diagnosis of arthritis based on the identification of specific transcriptomic profiles in knee synovial biopsies, Arthritis & Rheumatism 63 (2011) 751.

[26] L. van 't Veer, H. Dai, M. van de Vijver, Y. He, A. Hart, M. Mao, H. Peterse, K. van der Kooy, M. Marton, A. Witteveen, G. Schreiber, R. Kerkhoven, C. Roberts, P. Linsley, R. Bernards, S. Friend, Gene expression profiling predicts clinical outcome of breast cancer., Nature 415 (6871) (2002) 530–536. doi:10.1038/415530a.

[27] A.-C. Haury, P. Gestraud, J.-P. Vert, The influence of feature selection methods on accuracy, stability and interpretability of molecular signatures, PLoS ONE 6 (12) (2011) e28210. doi:10.1371/journal.pone.0028210.

[28] A. Liaw, M. Wiener, Classification and regression by randomforest, R News 2 (3) (2002) 18–22.
URL http://CRAN.R-project.org/doc/Rnews/

[29] S. Sonnenburg, G. Rätsch, S. Henschel, C. Widmer, J. Behr, A. Zien, F. de Bona, A. Binder, C. Gehl, V. Franc, The shogun machine learning toolbox, Journal of Machine Learning Research 11 (2010) 1799–1802.
URL http://jmlr.org/papers/volume11/sonnenburg10a/sonnenburg10a.pdf