Enhancing Classification Accuracy Through Feature Selection Methods

Everton R. Reis[1], **Paulo A. L. de Castro**[2], **Jaime S. Sichman**[1*]

[1]Laboratory of Intelligent Techniques – EP/USP
Av. Prof. Luciano Gualberto, 158 – trav. 3
05508-010 – São Paulo – SP – Brazil

[2]Division of Computational Science – IEC/ITA
Praça Marechal Eduardo Gomes, 50
12228-900 – São José dos Campos – SP – Brazil

everton.rreis@gmail.com, pauloac@ita.br, jaime.sichman@poli.usp.br

Abstract. *Feature selection is an important step to reduce data dimensionality and hopefully improve classification accuracy. In this paper we implemented a ”visual” wrapper+Sequential Forward Search (SFS) procedure usign R/Netlogo on two datasets, Breast Cancer Diagnosis from UCI repository (WBCD) and IBOVESPA daily data from 2001 to 2015, retrieved from YAHOO Finance with twenty eight technical indicators added to the data. In the IBOVESPA dataset we also used a wrapper+Sequential Backward Search (SBS) procedure and a hybrid method, combining a filter with SBS for five different filters. All of them used a Support Vector Machine (SVM) with a (Gaussian) Radial Basis Function kernel (RBF) and grid-search optimization.*

1. Introduction

Feature selection is an important step in machine learning and data mining, especially when dealing with high dimensional space of features and noisy data. Given a large set of features, it is applied to select a subset of features aiming to clean the data from noisy, irrelevant or misleading features. Guyon and Elisseeff [Guyon and Elisseeff 2003] also pointed out others benefits such as: facilitating data visualization and data understanding, reducing the measurement and storage requirements, reducing training and utilization times and defying the curse of dimensionality to improve prediction performance.

Algorithms for feature subset selection can be divided into three general approaches: filters, wrappers and embedded. Filters do not consider the predictor, it is a preprocessing step that filters out irrelevant attributes using general characteristics of the training set to select or exclude features. In the wrapper approach, the induction algorithm used to evaluate the subset of features is considered as a black box, the algorithm conducts a search for a good subset using the induction algorithm itself and scores the subset according to its predictive power, based on some metric. In embedded methods, the induction algorithm is not used as a black box, it differs from wrappers in the way feature selection and learning interact. Learning is not separated from the feature selection process, it is part of it [Guyon and Elisseeff 2003] [Kohavi and John 1997] [Blum and Langley 1997].

*Jaime Sichman is partially financed by CNPq, grant 303950/2013-7.

Blum and Langley [Blum and Langley 1997] characterize any feature-selection method in four basic components that determine the nature of the heuristic search process:

- *Starting point in the feature space:*The subset can start empty (i), with all features (ii), or with a random subset of features (iii). In the first case, the features are added successively to the subset, in the second, they are removed, and on the third, they can be added/removed or reproduced by a certain procedure;
- *Search Procedure:* An exhaustive search is the ideal search procedure, because it evaluates all the possible subset of features. However, it may become prohibitive, because there are 2^N possible combinations for N features. To avoid this constraint, other search procedures have been developed, but they can fall into suboptimal subset of features;
- *Evaluation Function:* is the function used to evaluate the available subsets of attributes. The function can be based on information theory, for a filter example, or by an evaluation metric obtained from the induction algorithm on the training set / evaluation set, in the case of wrapper and embedded methods;
- *Stop criterion*: It is a condition that stops the search, when it is fulfilled. It can stop adding or removing features when the metric chosen does not improve, keep revising the subset while the accuracy does not degrade, stop when the number of pre-defined features are achieved and so on.

In this paper, we are concerned with the search procedure, in particular to discover how to improve classification accuracy using feature selection. We adopt a wrapper with SVM as induction algorithm. As stop criterion, we utilized the stagnation of accuracy.

The remaining of the text is organized as follows. Section 2 describes feature selection methods, while in section 3 we describe the experimental setting. We present and analyze then the obtained results in section 4, and finally show our conclusions and further work in section 5.

2. Feature Selection Methods

Figure 1 illustrates the wrapper approach that we used for feature subset selection. Initially, the training set is given as an input to a ”black box”, which is composed by a SVM evaluator and a feature selection search strategy. First, we used the sequential forward search (SFS) as our search strategy, and secondly a hybrid method and the SBS.

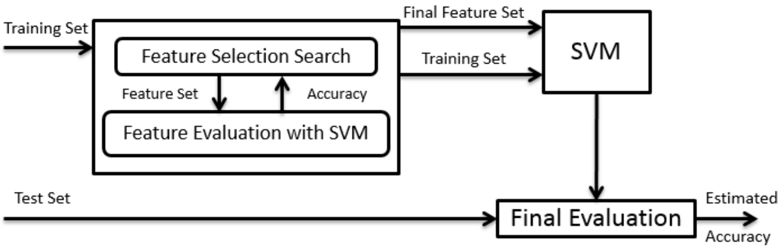


Figure 1. Wrapper model, adapted from [Kohavi and John 1997].

The sequential forward search is an heuristic method that can be generalized by the Algorithm 1 [Shroff and Maheta 2015].

Algorithm 1 Sequential Forward Search.

```
procedure SFS
    Optimal subset ← empty
    for $i = 1$ to number of features do
        Evaluate feature $i$ with some method
    Optimal subset ← best feature evaluated at the for loop
    repeat
        Add a feature, one by one, to a sub-optimal subset        ▷ Composed by a non-optimal feature and the best feature evaluated at step 4
        Evaluate Performance of each sub-optimal subset
        Select the best sub-optimal subset and set as Optimal Subset
    until stop criteria reached
    return Optimal subset
```

We implemented this procedure using R [R Core Team 2015] and Netlogo [Wilensky 1999], enabling to visualize the search procedure. Using this implementation, both agents (gray robots) and features (white labels) are located in a simulated environment. Agents have the shape of robots, and randomly walk through the environment looking for a feature, as shown in Figure 2(a); these are orderly sampled in the environment, forming a circle[1]. If there is already another agent in the discovered feature, the agent looks then for another unoccupied feature.

The system monitors the position of agents and features, and when all agents are located on features, these latter are sent to be sequentially evaluated by the wrapper method. The method returns the classification accuracy of each one of the features, and these values are sent to an agent's internal variable. The system sends then a message to agents come back to the base, carrying the name of their discovered feature, leaving a trail (gray dots) that represents the quality of its feature; the quality is based on the accuracy's value, that is scaled between 0 and 1. The darker is the gray, better is the feature, as shown in Figure 2(b).

When all agents have returned to the base, the system sets the best feature to another agent's variable. The agents then come back to their previous feature location, collecting their own trail and carrying the name of the best feature settled at the base. When they reach their previous feature, the best feature is added to the current feature subset, combining its previous feature with the best evaluated one. The result is shown in Figure 2(c). This implementation simulates the sequential forward search and enables the visualization of the SFS procedure, as a didactic approach.

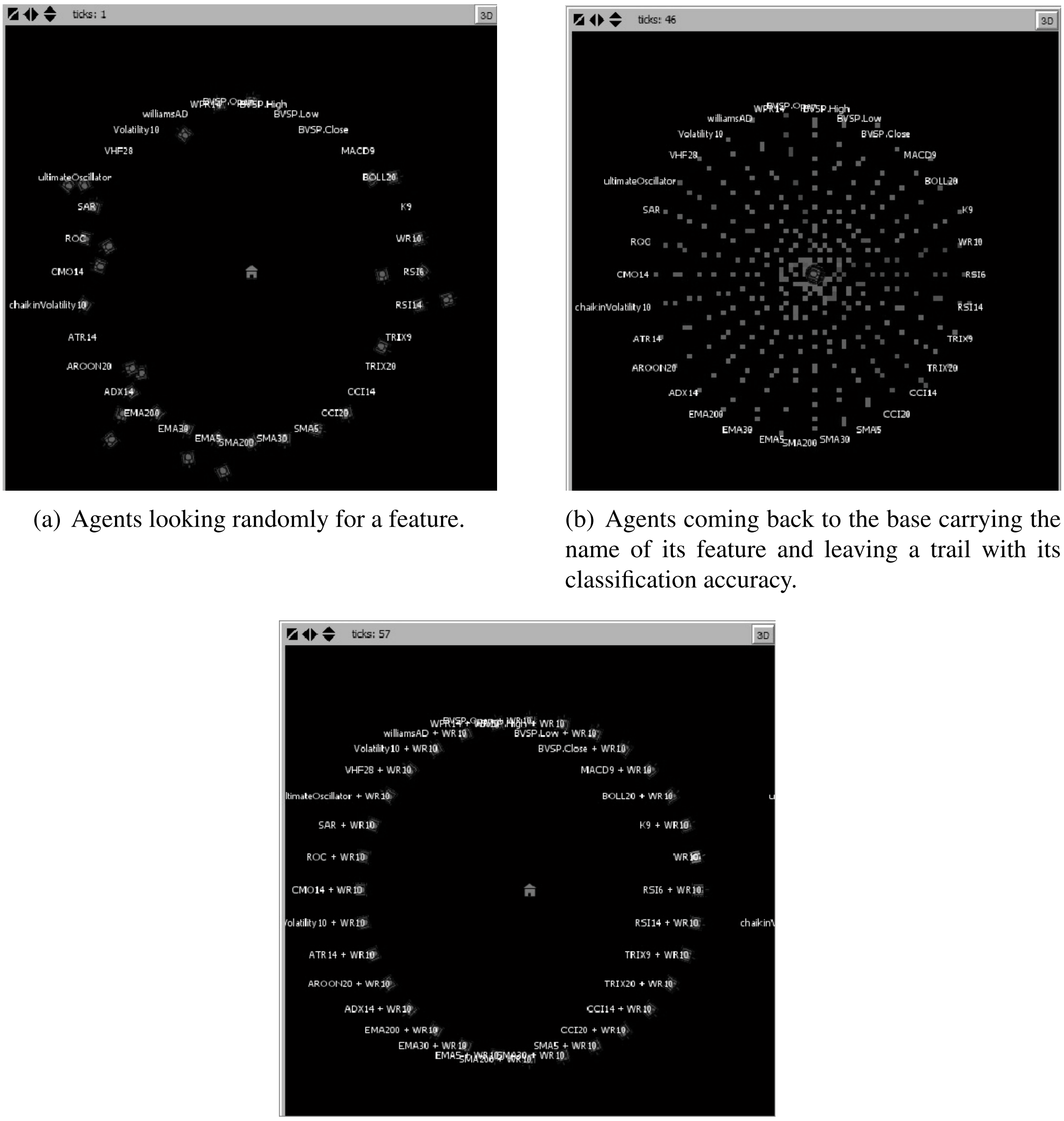
These steps are repeated until the stop criterion is fitted, i.e., until the current accuracy of subset t becomes equal or lower than the accuracy of subset $t - 1$.

Based on the results that we obtained from the stock market prediction experiments, we proposed a second feature selection method, which is a hybrid procedure. In Figure 3, it is possible to see how the method work. First we apply a filter, and we perform a feature pre-selection. Secondly, we apply again the wrapper approach, adopting a sequential backward search (SBS) as search strategy and a SVM with grid search opti-

[1] This is a aesthetic choice, the features could also be randomly sampled.

mization as the induction algorithm. The SBS algorithm is similar to the SFS, but instead of starting with an empty subset, it starts with all features and at each step it eliminates one of the features until the stop criterion is reached. We also used the SBS algorithm with no pre-selection step to compare the results.

In the wrapper, features are evaluated by a 10-fold cross validation. The hybrid procedure was used in [Lee 2009] with the F-Score filter. In this study we will use Information Gain (IG), Symmetrical Uncertainty (SU), ReliefF, Cfs (correlation based feature selection) and OneR, in the hybrid procedure. As far as we know, we are the first ones to test these different methods for stock trend prediction.



(a) Agents looking randomly for a feature.

(b) Agents coming back to the base carrying the name of its feature and leaving a trail with its classification accuracy.

(c) Feature subset updated by the best previous feature.

Figure 2. SFS implementation using R and Netlogo.

3. Experimental Description

3.1. Datasets

For the experiments, we used two datasets. The first one is the Breast Cancer Wisconsin (Original) Data Set, available at the UCI machine learning repository. The dataset contains 699 samples, but removing the missing values it remains with 683 samples. There are also nine features, represented as an integer between 1 and 10. This dataset has been chosen to evaluate the proposed method, the implementation with R/Netlogo, and to compare the results with those obtained by Akay [Akay 2009]. The features of the dataset are: clump thickness (F_1), uniformity of cell size (F_2), uniformity of cell shape (F_3), marginal adhesion (F_4), single epithelial cell size (F_5) bare nuclei (F_6), bland chromatin (F_7), normal nucleoli (F_8) and mitoses (F_9). Four hundred and forty four samples of the dataset belong to benign class, and the rest are of malignant class.

The second dataset is composed by daily data from IBOVESPA (Brazilian stock index). The data was retrieved from YAHOO Finance, utilizing the 'quantmod' package for R [Ryan 2015]. The period ranges from '2001-01-01' to '2015-12-31'. The data came with 6 features, that are: BVSP.Open, BVSP.High, BVSP.Low, BVSP.Close, BVSP.Volume and BVSP.Adjusted. The BVSP.Adjusted was removed because it is identical to the BVSP.Close[2], the BVSP.Volume was also removed because there was a lot of missing values, remaining 4 features that were lagged by one day (to not use today's closing price to predict the actual day). Another 28 features were added to the data, that are technical indicators, and they were chosen based on some of the technical indicators used in [Ni et al. 2011].

When using technical indicators to stock market prediction we have a lot of options to create our subsets. The problem is to define which indicators work better together and with which parameters (e.g. We can use different periods of time for RSI calculation), so feature selection plays an important role in this problem, supporting the analyst to choose the best subset of indicators that will enhance classification accuracy and consequently, its stock market prediction.

The features and their parameters are described at Table 1. The features were also scaled between $[0, 1]$ by Eq. (1), where X' is the rescaled value, X is the original value, X_{min} is the lowest value in the sample and X_{max} is the biggest value in the sample:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \qquad (1)$$

To create the labels for classification we used the ZigZag indicator, that work as a filter for noise and pattern identification [Raftopoulos 2003]. It ignores movements that are less than a specific threshold[3].

As we used moving averages of periods equal to 200, the dataset was reduced. The final dataset is then composed of 3524 observations that ranges from '2001-10-23' to '2015-12-22', and comprises 32 features (28 technical indicators and Open, High, Low,

[2] As IBOV is an index, there are not splits nor dividends.

[3] In this work, we used 6%.

Close prices). The labels are $\{1\}$ for bullish observations (prices are rising) and $\{0\}$ for bearish observations (prices are decreasing).

Table 1. Technical indicators and parameters from TTR package (documentation available at [Ulrich 2015]).

Feature	Description	Technical Trading Rules
MACD9	Moving average Convergence/Divergence	*MACD (maType = 'EMA')*
BOLL20	Bollinger Bands	*BBands (n = 20)*
K9	Stochastic	*stock (default)*
WR10	William's Overbought/Oversold Index	*WPR (n = 10)*
RSI6	Relative Strength Index	*RSI (n = 6)*
RSI14	Relative Strength Index	*RSI (default)*
TRIX9	Triple Smoothed Exponential Oscillator	*TRIX (n = 9)*
TRIX20	Triple Smoothed Exponential Oscillator	*TRIX (default)*
CCI14	Commodity Channel Index	*CCI (n = 14)*
CCI20	Commodity Channel Index	*CCI (default)*
SMA5	Simple Moving Average	*SMA (n = 5)*
SMA30	Simple Moving Average	*SMA (n = 30)*
SMA200	Simple Moving Average	*SMA (n = 200)*
EMA5	Exponencial Moving Average	*EMA (n = 5)*
EMA30	Exponencial Moving Average	*EMA (n = 30)*
EMA200	Exponencial Moving Average	*EMA (n = 200)*
ADX14	Welles Wilder's Directional Movement Index	*ADX (maType = 'EMA')*
AROON20	Aroon	*aroon (default)*
ATR14	Average True Range	*ATR (default)*
chaikinvolatility10	Chaikin Volatility	*chaikinVolatility (chaikinVolatility (default))*
CMO14	Chande Momentum Oscillator	*CMO (default)*
ROC	Rate of Change/Momentum	*ROC (default)*
SAR	Parabolic Stop-and-Reverse	*SAR (default)*
ultimateOscillator	The Ultimate Oscillator	*ultimateOscillator (default)*
VHF28	Vertical Horizontal Filter	*VHD (default)*
Volatility10	Volatility	*volatility (default)*
WilliamsAD	Williams Accumulation/Distribution	*williamsAD (default)*
WPR14	William's %R	*WPR (default)*

3.2. SVM parameters and Dataset Evaluation

It is essential that kernel parameters are properly set in order to improve SVM classification accuracy. We choose a SVM with a (Gaussian) Radial Basis Function Kernel (RBF), like Akay [Akay 2009], Lee [Lee 2009] and Ni *et al.* [Ni et al. 2011] did. The parameters to be optimized are the cost (C) and gamma (γ). The cost (C) is analogous to a regularization coefficient that controls the trade-off between minimizing training errors and controlling model complexity. Gamma (γ) is the parameter that represents the

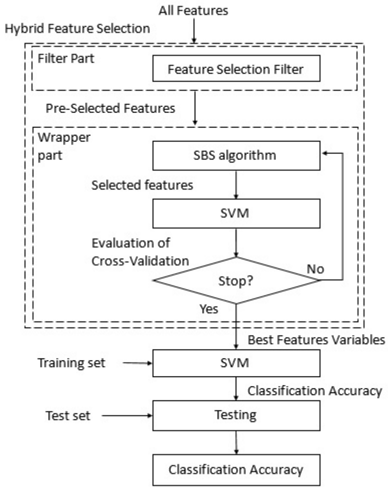


Figure 3. A Hybrid feature selection method, adapted from [Lee 2009].

precision of the Gaussian distribution (it is the inverse of variance) [Bishop 2006]. The search approach for optimization was also the grid search, where pairs of (C,γ) are tried out and the one with the best cross-validation accuracy is chosen. The grid space of C was $log_2 C\{-5, -3, ..., 15\}$ and for γ was $log_2\gamma\{-15, -13, ..., 3\}$. We utilized the *caret* package [Kuhn 2016] for optimization with parallel processing of four clusters (created by the `doParallel` package [Analytics and Weston 2015], using the `makeCluster` function). The computer used is an Intel(R) Core(TM) i5-4440 CPU @ 3.10GHz 3.10GHz, with 8,00 GB of RAM and the OS is a Windows 7 64-bits.

For the ”breast cancer” dataset, on each pair (C,γ) a 10-fold cross-validation was performed and the best parameter pair was chosen and tested on the validation set. We used the same splits percentages used by Akay [Akay 2009], that were 50%/50%, 70%/30% and 80%/20%. Figure 5(a) illustrates the process.

For IBOVESPA dataset, the data was divided at 70% for cross-validation and tuning parameters, and 30% for test. To select the best pair (C,γ), it was used a 10-fold time series cross-validation. For reproducibility, we used the parameters `clusterSetRNGStream(cl = cl,iseed = 2)` when using parallel processing. The folds were created using the function `createTimeSlices` with the following parameters:

- `initialWindow = 488`: the amount of data used for training was approximately two years;
- `horizon = 244`: the amount of data used for validation was approximately one year;
- `fixedWindow = TRUE`: each training sample starts at a different time, that is given by the skip value (at fold $k + 1$ is added the value of skip);
- `skip = 191`: the amount of data skipped at each fold.

These parameters were used to create the folds for cross-validation. Figure 4 illustrates how these folds were created. This procedure is needed to avoid bias, since when using time-series cross-validation we use earlier subsets to predict later ones. After selecting the best pair, the SVM was trained with 70% of the dataset using optimized

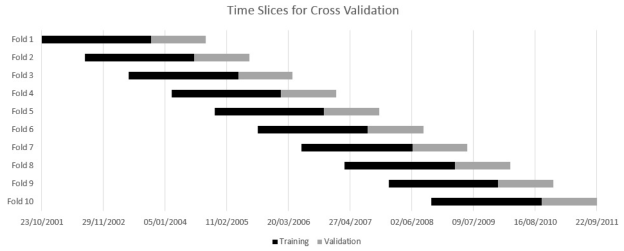
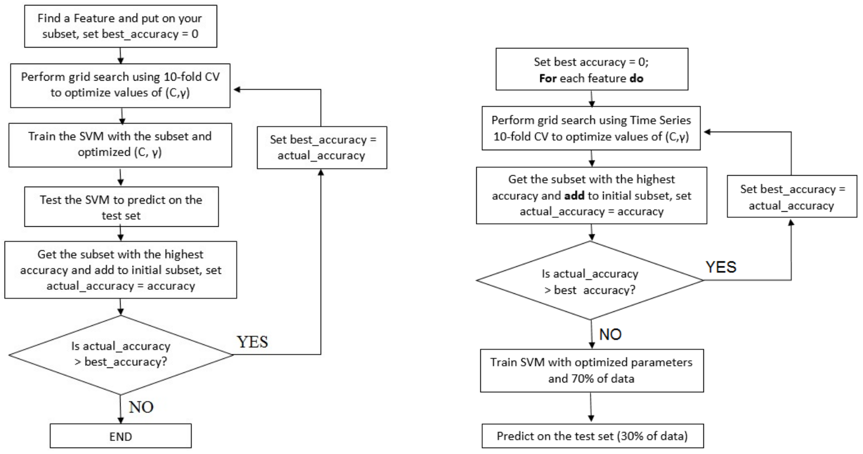


Figure 4. 70% of IBOVESPA dataset sliced at 10-fold for cross-validation.

parameters, and finally tested with the other 30% of the data. Figure 5(b) illustrates the whole process.


Find a Feature and put on your subset, set best_accuracy = 0
Perform grid search using 10-fold CV to optimize values of (C,γ)
Train the SVM with the subset and optimized (C, γ)
Set best_accuracy = actual_accuracy
Test the SVM to predict on the test set
Get the subset with the highest accuracy and add to initial subset, set actual_accuracy = accuracy
Is actual_accuracy > best_accuracy?
YES
NO
END
Set best accuracy = 0; For each feature do
Perform grid search using Time Series 10-fold CV to optimize values of (C,γ)
Get the subset with the highest accuracy and add to initial subset, set actual_accuracy = accuracy
Set best_accuracy = actual_accuracy
Is actual_accuracy > best accuracy?
YES
NO
Train SVM with optimized parameters and 70% of data
Predict on the test set (30% of data)

(a) SVM-based model using grid search with 10-fold CV to optimize model parameters (Used in the WBCD dataset).

(b) SVM-based model using grid search with 10-fold time-series CV to optimize model parameters (Used in the IBOVESPA dataset).

Figure 5. Processes used to evaluate the datasets.

The filters described at Section 2 were calculated using, `set.seed(1)` for reproducibility, and the parameters of ReliefF filter were `neighbours.count = 5` and `sample.size = 10`.

4. Obtained Results

4.1. Breast Cancer Dataset

Table 2 presents our result for the Breast cancer dataset. Table 3 compares classification accuracy among our method and others classifiers from literature.

For the 50-50% split, the obtained values of (C, γ) were $(2^{-15}, 2^{13})$, for the 70-30% split we obtained $(2^{-1}, 2^{-1})$ and for the 80-20% split the results were $(2, 2^{-3})$. Our results are quite the same as the ones obtained by Akay [Akay 2009]; however, he has achieved this accuracy using a 80-20% split and 5 features $(F_6 + F_3 + F_2 + F_1 + F_7)$, while we just used 2 features $(F_1 + F_7)$ with a split of 70-30%. Therefore, we have got a higher dimensionality reduction for the same accuracy.

Table 2. Classification accuracies for each model and different test subsets for the Breast Cancer Dataset.

Model	Classification Accuracy		
	50-50% training-test partition	70-30% training-test partition	80-20% training-test partition
1	99.12%	-	-
2	-	99.51%	99.27%
$1 = F_1 + F_7 + F_9$			
$2 = F_1 + F_7$			

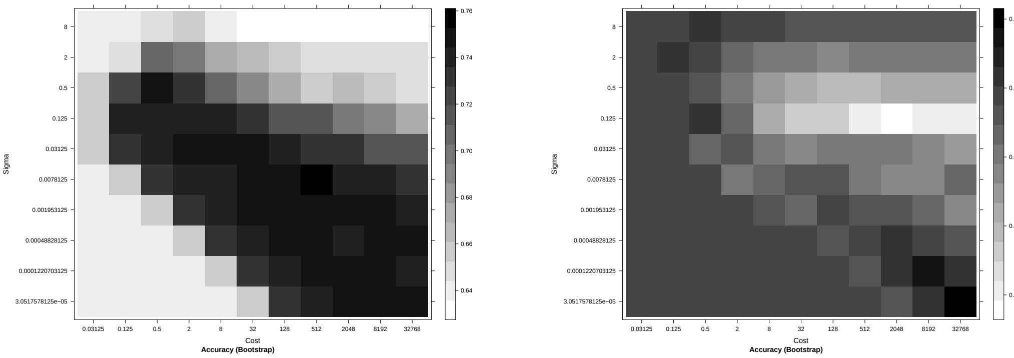
Table 3. Classification accuracies from literature, adapted from [Akay 2009] for the Breast cancer dataset.

Author (year)	Method	Classification accuracy (%)
Quinlan (1996)	C4.5	94.74
Hamiton et al. (1996)	RIAC	95.00
Ster and Dobnikar (1996)	LDA	96.80
Nauck and Kruse (1999)	NEFCLASS	95.06
Pena-Reyes and Sipper (1999)	Fuzzy-GA1	97.36
Setiono (2000)	Neuro-rule 2a	98.10
Goodman et al. (2002)	Optimized-LVQ	96.70
Goodman et al. (2002)	Big LVQ	96.80
Goodman et al. (2002)	AIRS	97.20
Albrecht et al. (2002)	LSA with perceptron algorithm	98.80
Abonyi and Szeifert (2003)	Supervised fuzzy clustering	95.57
Polat and Gunes (2007)	LS-SVM	98.53
Akay (2009)	$F - score$ + SVM	99.51
This study (2016)	SVM + wrapper_SFS	99.51

4.2. IBOVESPA dataset

For the second dataset (IBOVESPA), the results are presented in Table 4. It is possible to see that the hybrid methods OneR_SBS and Cfs_SBS were the methods that most reduced data dimensionality (just 5 features in each one of them), and OneR_SBS had the best accuracy in the test set (70,04%). The use of sequential backward search with OneR improved accuracy, at the test set, from 59,26% to 70,04%, an improvement of 18,19%, and reduced features from 12 to 5, a reduction of 58,33%. Looking at the filters, Cfs got the highest accuracy in the test set (69,94%) using 9 features. The wrappers did not performed as well as the hybrid methods. The hybrid approach reduced features from all filters (IG - 13/10, SU - 15/11, ReliefF - 22/7, Cfs 9/5, OneR - 12/5), improving the accuracy's value (test set) in 2 out of 5 cases.

The values for the grid-search optimization of parameters (C, γ) on Cfs_SBS and OneR_SBS are presented at Figure 6(a) and Figure 6(b) respectively, as an example. The black square in the figures represents the best combination of (C, γ) for accuracy's value.



(a) Cfs_SBS method.

(b) OneR_SBS method.

Figure 6. Values of Grid-Search optimization for RBF kernel (C, γ) and IBOVESPA dataset.

5. Conclusions and Further Work

We presented in this work three different methods for feature selection: wrappers, filters and a hybrid method. The wrappers were wrapper + Sequential Forward Search (SFS) and wrapper + Sequential Backward Search (SBS). The filters were Information Gain (IG), Symmetrical Uncertainty (SU), ReliefF (RF), Correlation Based feature selection (Cfs), and OneR. The hybrid methods were a combination of each filter with a SBS. The wrapper_SFS was applied at WBCD and IBOVESPA dataset, and implemented using R/Netlogo to create a visualization of the searching process. The other methods were applied just at IBOVESPA dataset.

Comparing to Akay's study [Akay 2009], we achieved the same accuracy, but with a smaller number of features (2 vs 5) in the Breast Cancer dataset (WBCD). In the IBOVESPA dataset, all methods were compared. The Cfs_SBS and OneR_SBS were the methods that most reduced features (just 5 were used) and still had a good performance on the test set $(68,72\%$ and $70,04\%)$, for applications that demands fast processing for decision-making, like High Frequency Trading, it can be a good option. The one that achieved the best accuracy's value in the test set was the OneR_SBS (70,04%), using the hybrid method it reduced features significantly (58,33%) and also improved accuracy (18,19%). A faster and also good method were the filters: Symmetrical Uncertainty (69,57% with 15 features) and Cfs (69,94% with 9 features). The hybrid method reduced features when used with all filters, increased accuracy in two cases, and had a small decreasing in three of them. It is a good trade-off between wrapper and filters, as expected, because it reduces processing time and final number of features, increasing accuracy in some cases and having just a small decrease in others, but reducing features in all of them.

In future work, we intend to test different parameters for the technical indicators, try other features, like fundamental indicators, indexes, exchange rates, commodities, and also different induction algorithms, like neural network, random forest, an ensemble of them, and so on.

Table 4. Classification accuracies for each model and different test subsets for the IBOVESPA Dataset.

Method	Features Selected	Training/Validation (Accuracy)	Test set (Accuracy)	Cost (C)	gamma (γ)
SVM	All features	$70,16 \pm 9,04\%$	69,57%	2^{7}	2^{-15}
SVM + wrapper_SFS	RSI14 + WR10 + ROC + AROON20 + CCI20 + CMO14 + BOLL20	$75,82 \pm 5,57\%$	66,82%	2^{15}	2^{-9}
SVM + wrapper_SBS	BVSP.High + BVSP.Low + BVSP.Close + MACD9 + BOLL20 + K9 + WR10 + RSI6 + RSI14 + TRIX9 + TRIX20 + CCI14 + CCI20 + SMA5 + SMA30 + EMA5 + EMA30 + EMA200 + AROON20 + CMO14 + ROC + SAR + williamsAD + WPR14	$72,34 \pm 7,82\%$	69,19%	2^{7}	2^{-15}
SVM + Information Gain ($t = 0.05$)	EMA200 + WR10 + WPR14 + RSI6 + SMA200 + CCI14 + ultimateOscillator + BOLL20 + RSI14 + CCI20 + ROC + AROON20 + CMO14	$72,87 \pm 6,57\%$	68,62%	2^{-1}	2^{-7}
SVM + Symmetrical uncertainty ($t = 0.05$)	WR10 + WPR14 + RSI6 + CCI14 + EMA200 + BOLL20 + CCI20 + SMA200 + ultimateOscillator + RSI14 + AROON20 + ROC + CMO14 + K9 + SMA30	$72,17 \pm 7,51\%$	69,57%	2^{5}	2^{-11}
SVM + ReliefF ($t = 0.01$)	AROON20 + chaikinVolatility10 + TRIX20 + ADX14 + VHF28 + ultimateOscillator + SMA30 + EMA30 + Volatility10 + BVSP.Close + EMA5 + ATR14 + SMA5 + BVSP.High + BVSP.Low + williamsAD + SMA200 + BVSP.Open + SAR + ROC + MACD9 + TRIX9	$66,11 \pm 13,17\%$	66,92%	2^{7}	2^{-13}
SVM + Cfs	WR10 + RSI6 + CCI14 + SMA30 + SMA200 + EMA200 + AROON20 + ROC + WPR14	$72,50 \pm 7,50\%$	69,94%	2^{3}	2^{-11}
SVM + OneR ($t = 3.5$)	ADX14 + Volatility10 + SMA30 + VHF28 + AROON20 + TRIX9 + BVSP.High + BVSP.Low + EMA5 + SMA5 + BVSP.Open + SAR	$65,04 \pm 10,86\%$	59,26%	2^{-1}	2^{-1}
SVM + IG_SBS	WR10 + WPR14 + RSI6 + CCI14 + BOLL20 + RSI14 + CCI20 + ROC + AROON20 + CMO14	$75,57 \pm 5,94\%$	67,77%	2^{5}	2^{-7}
SVM + SU_SBS	WR10 + WPR14 + RSI6 + CCI14 + BOLL20 + ultimateOscillator + RSI14 + AROON20 + ROC + CMO14 + K9	$74,88 \pm 6,87\%$	69,09%	2^{13}	2^{-15}
SVM + ReliefF_SBS	AROON20 + chaikinVolatility10 + TRIX20 + SMA5 + SAR + ROC + MACD9	$72,01 \pm 7,91\%$	69,75%	2^{15}	2^{-15}
SVM + Cfs_SBS	WR10 + RSI6 + AROON20 + ROC + WPR14	$75,29 \pm 6,07\%$	68,72%	2^{9}	2^{-7}
SVM + OneR_SBS	AROON20 + TRIX9 + BVSP.Low + SMA5 + SAR	$69,39 \pm 8,71\%$	70,04%	2^{15}	2^{-15}

References

[Akay 2009] Akay, M. F. (2009). Support vector machines combined with feature selection for breast cancer diagnosis. *Expert systems with applications*, 36(2):3240–3247.

[Analytics and Weston 2015] Analytics, R. and Weston, S. (2015). *doParallel: Foreach Parallel Adaptor for the 'parallel' Package*. R package version 1.0.10.

[Bishop 2006] Bishop, C. M. (2006). *Pattern Recognition*.

[Blum and Langley 1997] Blum, A. L. and Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artificial intelligence*, 97(1):245–271.

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

[Kohavi and John 1997] Kohavi, R. and John, G. H. (1997). Wrappers for feature subset selection. *Artificial intelligence*, 97(1):273–324.

[Kuhn 2016] Kuhn, M. (2016). *caret: Classification and Regression Training*. R package version 6.0-64.

[Lee 2009] Lee, M.-C. (2009). Using support vector machine with a hybrid feature selection method to the stock trend prediction. *Expert Systems with Applications*, 36(8):10896–10904.

[Ni et al. 2011] Ni, L.-P., Ni, Z.-W., and Gao, Y.-Z. (2011). Stock trend prediction based on fractal feature selection and support vector machine. *Expert Systems with Applications*, 38(5):5569–5576.

[R Core Team 2015] R Core Team (2015). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

[Raftopoulos 2003] Raftopoulos, S. (2003). The zigzag trend indicator. *TECHNICAL ANALYSIS OF STOCKS AND COMMODITIES-MAGAZINE EDITION-*, 21(11):26–33.

[Ryan 2015] Ryan, J. A. (2015). *quantmod: Quantitative Financial Modelling Framework*. R package version 0.4-5.

[Shroff and Maheta 2015] Shroff, K. P. and Maheta, H. H. (2015). A comparative study of various feature selection techniques in high-dimensional data set to improve classification accuracy. In *Computer Communication and Informatics (ICCCI), 2015 International Conference on*, pages 1–6. IEEE.

[Ulrich 2015] Ulrich, J. (2015). *TTR: Technical Trading Rules*. R package version 0.23-0.

[Wilensky 1999] Wilensky, U. (1999). *NetLogo*. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston,IL.