Filter- versus Wrapper-based Feature Selection For Credit Scoring

Petr Somol[1], Bart Baesens[2], Pavel Pudil[1],
Jan Vanthienen[2]
[1]Institute of Information Theory and Automation
Pod vodárenskou věží 4, 182 08 Prague 8, Czech Republic
{Somol;Pudil}@utia.cas.cz
[2]K.U.Leuven, Department of Applied Economic Sciences,
Naamsestraat 69, B-3000 Leuven, Belgium
{Bart.Baesens; Jan.Vanthienen}@econ.kuleuven.ac.be

Abstract

We address the problem of credit scoring as a classification and feature subset selection problem. Based on the current framework of sophisticated feature selection methods, we identify features that contain the most relevant information to distinguish good loan payers from bad loan payers. The feature selection methods are validated on several real world datasets with different types of classifiers. We show the advantages following from using the sub-space approach to classification. We discuss many practical issues related to the applicability of feature selection methods. We show and discuss some difficulties that use to be insufficiently emphasised in standard feature selection literature.

Keywords: credit scoring, feature selection, subset search, probabilistic dependence measures

1 Introduction

One of the key decisions financial institutions have to make is to decide whether or not to grant a loan to a customer. This decision basically boils down to a binary classification problem which aims at distinguishing good payers from bad payers. Until recently, this distinction was made using a judgmental approach by merely inspecting the application form details of the applicant. The credit expert then decided upon the credit-worthiness of the applicant using all possible relevant information which describes his socio-demographic status, economic conditions and intentions. The advent of data storage technology has facilitated financial institutions to store all information regarding the characteristics and repayment behaviour of credit applicants electronically. This has motivated the need to automate the credit granting decision by using statistical or machine learning algorithms.

Numerous methods have been proposed in the literature to develop credit scoring models [1, 2]. These models include traditional statistical methods (e.g. logistic regression [15]), nonparametric statistical models (e.g. k-nearest neighbour [8] and classification trees [3]) and neural network models [4]. Most of these studies primarily focus at developing classification models with high predictive accuracy. However, besides classification accuracy, it is also very important to have parsimonious models which only consider a limited number of features to make the credit granting decision. This feature selection idea is in the data mining literature often embodied by the principle of Occam's razor which essentially advocates the use of simple, low parametrised models. By using only a few features to decide upon credit approval, the scorecard builder will gain more insight into the model and better understand its working.

In this paper, we intend to investigate the potential of feature selection methods for credit scoring. We will study the possible problems that can emerge when using probabilistic dependence measures (Filter approach) instead of classification rates of concrete classifiers (Wrapper approach) for feature selection [11]. The experiments will be conducted using 4 real-life credit scoring datasets.

This paper is organised as follows. Section 2 describes the adopted feature selection methodology consisting of two recent subset search methods. The empirical evaluation, consisting of the feature selection methods setup, the dataset characteristics and the results, are described in section 3. Section 4 concludes the paper.

2 Adopted Feature Selection Methodology

In this paper, we discuss the possibilities of applying feature selection methods to credit scoring problems to reduce problem dimensionality, i.e. to identify the most important features and possibly to improve the performance of employed classifiers.

The feature selection problem is more or less a special case of a much broader problem of subset selection. Suppose a large set of (D) items is given from which we need to find a small (d) subset being optimal in a certain sense. In statistical subset selection, being optimal usually means being most suitable for classification or data approximation. Although it may not be apparent, the subset selection problem may become prohibitive because of its computational complexity. It is not possible to rank the items in our set simply according to their individual properties and select only the best. The item properties may depend strongly on each other and a subset of individually "bad" features may prove to be rather "good" because of positive interaction effects. Because of this uncertainty, the only apparent way of searching for optimal subsets is – simply to evaluate all the 2^D possible item combinations. However, testing all subsets is a combinatorial problem that requires an exponential amount of computational time. Because of this limitation a broad range of sub-optimal feature selection methods has been developed over the last decades.

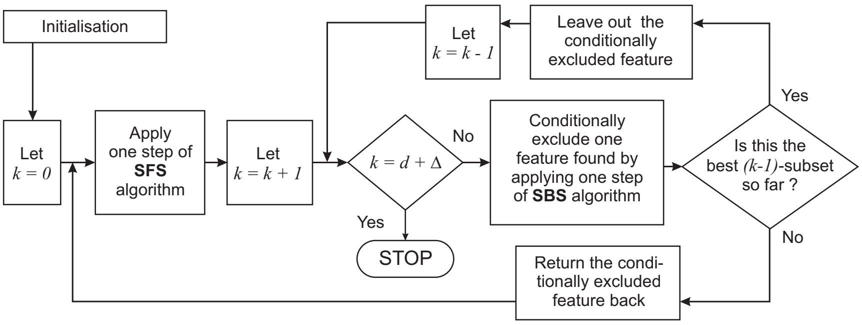
Initialisation
Let $k = 0$
Apply one step of **SFS** algorithm
Let $k = k + 1$
$k = d + \Delta$
No
Yes
STOP
Let $k = k - 1$
Leave out the conditionally excluded feature
Conditionally exclude one feature found by applying one step of **SBS** algorithm
Is this the best *(k-1)*-subset so far ?
Yes
No
Return the conditionally excluded feature back

Figure 1: Simplified diagram of the "Forward Floating Search" feature selection algorithm.

Based on our recent experience we have adopted a methodology for applying a combination of modern feature selection methods to achieve the best possible results in a reasonable time. At the first stage, we apply the so-called *floating search* methods [12] because of their proven performance and relatively high computational speed. At the second stage, we try to improve some of the most promising results using more computationally demanding, but more flexible and efficient *oscillating search* methods [14]. Let us summarise the basic principles of these methods first.

2.1 Floating Search Methods

In contrast to the conventional sequential search methods, the *Floating Search* algorithms attempt to improve the feature subset after every step by means of backtracking. Consequently, the resulting dimensionality in respective intermediate stages of the algorithm is not changing monotonically but is actually "floating" up and down. Two different algorithms have been defined according to the dominant direction of the search: forward (SFFS) and backward (SBFS) floating search. A simplified diagram of the SFFS method can be seen on Figure 1. The **SFS** abbreviation denotes the simple *Sequential Forward Selection* algorithm that is used here as a sub-procedure. SFS adds one feature to the current set of features that improves the overal criterion value the most. The functionality of **SBS** as *Sequential Backward Selection* is obvious. For details on both SFS and SBS see e.g. [5]. The k variable denotes the current subset size, d is a user set constant depicting the target subset size and Δ denotes an additional parameter allowing the algorithm to continue for a short time after reaching the target subset size d – to take use of the "floating" principle and thus possibly improve the solution for the target subset size d. It should be noted that restricting the algorithm to stop at $d + \Delta$ does not have any other reason than to spare computational time. The recommended way of usage (the way we use it here) is letting the algorithm go through all possible subset sizes and examine all results at the end.

The effectiveness of the floating search has been demonstrated on different problems [12, 6, 9]. The subsequent extension, the *Adaptive Floating Search* [13], is able to focus on the

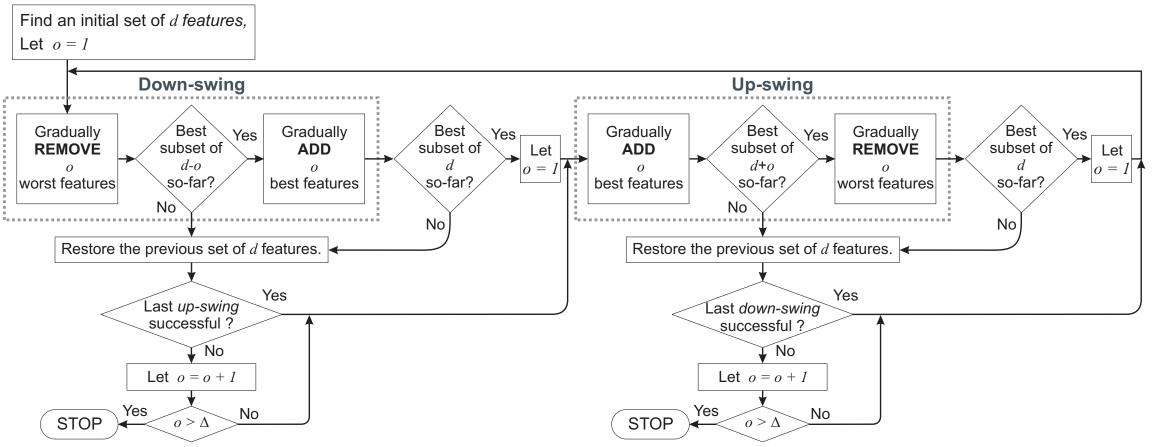
Find an initial set of d features,
Let o = 1
Down-swing
Gradually REMOVE o worst features
Best subset of d-o so-far?
Yes
No
Gradually ADD o best features
Best subset of d so-far?
Yes
No
Let o = 1
Restore the previous set of d features.
Last up-swing successful ?
Yes
No
Let o = o + 1
o > Δ
Yes
No
STOP
Up-swing
Gradually ADD o best features
Best subset of d+o so-far?
Yes
No
Gradually REMOVE o worst features
Best subset of d so-far?
Yes
No
Let o = 1
Restore the previous set of d features.
Last down-swing successful ?
Yes
No
Let o = o + 1
o > Δ
Yes
No
STOP

Figure 2: Simplified diagram of the "Oscillating Search" feature selection algorithm.

desired dimensionality and find a better solution but only at the cost of significantly increased computational time. For a detailed description of the floating search principle see [12].

2.2 Oscillating Search Methods

Unlike other methods, the *Oscillating Search* (OS) [14] repeatedly modifies the current subset of d features. This is achieved by alternating the so-called *down-* and *up-swings*. The *down-swing* removes o "bad" features from the current set to obtain a new set of $d - o$ features at first, then adds o "good" ones to obtain a renewed set of d features. A "good" feature may be defined as "a feature that being added increases the criterion value the most". The meaning of "bad" may be defined correspondingly (we do not define precisely what "good" and "bad" means here as it is not necessary for understanding the overal algorithm principle). The *up-swing* is simply a counterpart to the *down-swing*. Two successive opposite swings form an *oscillation cycle*. Using this notion, the oscillating search consists of repeating oscillation cycles. The value of o is denoted the *oscillation cycle depth* and should be set to 1 initially. If the last oscillation cycle has not improved the working subset of d features, the algorithm increases the oscillation cycle depth by setting $o = o+1$. Whenever any swing results in finding a better subset of d features, the depth value o is reset to 1. The algorithm stops after the value of o has exceeded a user-specified limit Δ (see Figure 2). The meaning of **ADD** and **REMOVE** can be specified in different ways to get different versions of the OS algorithm. In this paper we substitute "ADDing o features" by repeating successively the SFS step o times and "REMOVEing o features" by o successive SBS steps.

The OS algorithm is principally a variant of the gradient method. Its most important property when compared to the Floating Search is the ability to run in a randomised mode. This may (in some cases) overcome the problem of practically all sequential feature selection methods – the inclination to get stuck in local extremes. Another advantage of the oscillating search method is its ability to run not only as a stand alone search method, but also as a tuning tool capable of improving results obtained in another way, e.g. by using the Floating Search. For a detailed algorithm description see [14].

3 Empirical Evaluation

3.1 Feature Selection Methods Setup

To obtain results representative enough for drawing conclusions, we investigated different *classifier / feature selection criteria / subset search algorithm* combinations. We set up the following components:

- The *SFFS* subset search algorithm (see section 2.1). This is the *forward* version of the floating search algorithm set to get through all subset sizes (from 1 to the total number of features).
- The *OS* subset search algorithm (see section 2.2). We use the simplest and fastest *sequential* version with oscillation depth limit set to 2. The algorithm is called repeatedly for each subset size starting from random initial feature subsets until 5 consecutive runs have not led to result improvement.
- The *Gaussian* classifier (simple Bayesian classifier assuming the data distribution to be normal) to be used both as a classifier and a subset evaluation criterion. For details on classifiers see e.g.[5, 7].
- The *1–Nearest Neighbour* classifier to be used both as a classifier and a subset evaluation criterion.
- The *Bhattacharyya* probabilistic distance measure to be used as a subset evaluation criterion. For details on all probabilistic measures mentioned here see e.g.[5, 7].
- The *Divergence* probabilistic distance measure to be used as a subset evaluation criterion.
- The *Patrick-Fischer* probabilistic distance measure to be used as a subset evaluation criterion.

Our selection of algorithms and classifiers is affected by the expected computational complexity of the problem. As the number of features gets higher (as high as 76 in one of the datasets tested), more complex algorithms may require weeks of computational time. We believe the algorithms we use constitute a reasonable compromise suitable for our purposes. The actual computations on our PentiumIII-800MHz machines took altogether about a month, though. The results obtained from testing the different combinations of the listed components are described in section 3.3.

3.2 Datasets and Experimental Setup

Table 1 displays the characteristics of the datasets that have been be used to evaluate the different feature selection methods. The *Bene1* and *Bene2* datasets were obtained from two major financial institutions in the Benelux (Belgium, The Netherlands and Luxembourg). For both datasets, a bad customer was defined as someone who has missed three consecutive months of payments [16]. The *German credit* and *Australian credit* datasets are publicly available at the UCI repository (`http://kdd.ics.uci.edu/`).

	Dataset Size	Original Features		
		Total	Continuous	Nominal
Austr	690	14	6	8
Germ	1000	20	7	13
Bene1	3123	28	14	14
Bene2	7190	28	18	10

Table 1: Characteristics of the original datasets (containing nominal features).

The methods and functions we intend to employ and evaluate are currently defined for continuous data only. For this reason, we had to transform the datasets to eliminate the nominal features while preserving the information contained in them. We have adopted the simplest way to do that – we replaced each nominal feature by a number of binary features representing every possible nominal value that the original feature can take.

It should be noted that such a transformation has negative consequences. First, the overall dimensionality of the problem increases with unchanged number of data samples. Second, the new binary features are highly correlated. However, it will be shown that despite these drawbacks the use of feature selection methods leads to reasonable results.

	Dataset Size			Transformed Features
	Total	in Class 1	in Class 2	
Austr	690	307	383	38
Germ	1000	300	700	59
Bene1	3123	2082	1041	55
Bene2	7190	5033	2158	76

Table 2: Characteristics of the transformed datasets (continuous features only).

As so often with real world problems, we face the 'curse of dimensionality' problem here. The number of samples in datasets is relatively low in relation to the dimensionality. Nevertheless, the number of samples in all cases is higher than ca. 10 times the dimensionality. According to different studies (see e.g. [10]) and our experience this is enough to achieve our goals.

3.3 Results

The results are presented in five graphs per dataset (see Figures 3,4,5,6). The two main graphs show the best achieved classification rates for all subset sizes while the three small graphs show the maximal probabilistic distance measure values found. We do not include full listings and description of selected feature subsets because of the extent of the complete list.

It can be noted that some values are missing – this is either because of unreasonably high computational complexity of certain cases or because of numerical problems. Occasional numerical problems follow apparently from sparseness of the feature space as well as from the use of correlated binary features we have created during the dataset transformation. Numerical problems can be observed especially for subsets obtained by the Filter approach and tested using the *Gaussian* classifier. Also, according to the definition all *Bhattacharyya*, *Divergence* and *Patrick-Fischer* graphs should be monotonically ascending with increasing subset size what is not the case for larger subsets with all the tested datasets.

	Austr	**Germ**	**Bene1**	**Bene2**
Best Gaussian Classification Rate	0.926	0.838	0.739	0.753
Corresponding Subset Size	10 (14)	22	14	23
Best FS Procedure (Gaussian rate as criterion)	OS rand. (SFFS) Wrapper	OS rand. Wrapper	OS rand. Wrapper	OS rand. Wrapper
Best 1–NN Classification Rate	0.913	0.784	0.727	0.710
Corresponding Subset Size	13	19	25	11
Best FS Procedure (1–NN rate as criterion)	SFFS Wrapper	OS rand. Wrapper	SFFS Wrapper	OS rand. Wrapper
Full Set Gaussian Classification Rate	num.err.	num.err.	num.err.	num.err.
Full Set 1–NN Classification Rate	0.670	0.557	0.599	0.648
Full Set Size	38	59	55	76

Table 3: Best achieved results. Comparison of sub-set and full set classification results.

The graphs confirm that only a fraction of features is sufficient to achieve the maximum classification rate. Moreover, in all cases the full set of features yields worse results than a selected subset or is not usable at all due to numerical problems (see Table 3). In terms of the original feature set this means that many continuous features can be omitted. In case of the nominal features some of their values carry more important information than others. It can also be seen that the number of used features can be further significantly reduced at a cost of only slight decrease of the classifier performance. In case of all tested datasets, it is possible to select only about 5–10 features to achieve classification rates almost as good as the best ones.

The results show an overall superiority of Wrappers over Filters. The best results have been always obtained using the Wrapper approach. A very interesting observation is that there is a weaker than expected connection between Filter based subset preference and final classification rate with all datasets and practically all measure / classifier combinations.

The results also illustrate how differently the principally different classifiers can behave. Although practically all the highest classification rates were achieved using the *Gaussian* classifier, the 1-*NN* proved to be much less vulnerable to dimensionality or numerical problems. This can be clearly seen e.g. on Figure 5 for feature subsets of more than ca. 20 features as well as on Figures 3 and 4 for subsets of high number of features.

3.3.1 'Austr' Results

The two classifiers perform comparably well on this dataset. For subset sizes close to the full dimension the performance deteriorates rapidly. The Filter results are very bad with the *gaussian* classifier. The explanation is offered in section 4. Only with the *1-NN* classifier the Filter results show some dependence of classification rate on the measure values (notably in case of *Patrick-Fischer*). The results are inferior to those obtained by the Wrapper approach though.

3.3.2 'Germ' Results

The *gaussian* classifier outperforms the *1-NN* classifier in this case except for subsets of more than ca. 50 features. Almost no dependence between classification rate and probabilistic distance measure values can be observed here. The subsets selected using the Filter approach perform poorly with both classifiers. There seems to be almost no relation between the Filter and *gaussian* classifier subset performance. Only the Patrick-Fischer measure seems to perform slightly better than the two others in conjunction with the 1–NN classifier.

For the 'Germ' dataset, we include a complete description of the original and best selected features (see Table 4).

3.3.3 'Bene1' Results

The *gaussian* classifier yields generally better results, however it is more vulnerable to numerical problems with sparse datasets than the *1-NN* classifier (here for feature set sizes > ca. 30). The partially chaotic behaviour of Filter results clearly shows the problems of weak principal connection between the search and classification processes as stated above. The additional graphs of distance measure values show the problem of difference in meaning of particular distance measures. Although due to monotonicity of the probabilistic measures we cannot simply identify the best subset size according to measure value, the graphs demonstrate certain effects. First, the Bhattacharyya shows rapid increase for subsets of ca. 10 features, what is different from the Divergence (starts to rise at ca. 25–30) and Patrick-Fischer (ca. 30–35). Second, the graphs help to identify the point from where the numerical problems begin. The Divergence and Patrick-Fischer graphs decline for subset sizes of above ca. 35 features, what indicates numerical problems as the monotonicity is breached. This point of decline corresponds with what can be seen at the upper two graphs – the SFFS with *1-nn* collapsed there and the SFFS with the *gaussian* classifier reached an unchanging state.

3.3.4 'Bene2' Results

For all datasets, the OS algorithm proves to be able to outperform the SFFS. However, with this dataset and this configuration, the OS not only showed to be computationally expensive, but also failed to yield reasonable results for subsets of more than ca. 35 features. A dependence between the Bhattacharrya and Divergence values and corresponding *gaussian* classifier performance for subsets from 1 to ca. 20 can be observed. In contrary, for bigger subsets the dependence between the probabilistic measures and classification rate becomes practically random. However, the Filters yield better performing subsets than in case of the other datasets. Similarly to the 'Bene1' dataset, there is a point (subset size ca. 58) from which on the numerical errors become overriding.

4 Conclusions

For the four credit scoring datasets, we have achieved slightly better classification results than published before. Moreover, we have identified subsets of features (and particular values in case of nominal features) that are sufficient to accomplish the classification task.

Besides the known fact that dimensionality reduction may lead not only to computational (and data acquisition) cost savings but also to performance and accuracy gain, we have demonstrated other less obvious factors that may affect the classification process. In many cases the process of dimensionality reduction proved to be necessary to enable classification at all due to numerical problems with larger feature subsets.

If numerical problems should emerge, then the Filter approach seems to be likely more vulnerable. This can be explained simply by the fact that unlike the Wrapper approach (which employs only the classifier related computations) the Filter approach employs both the classifier related and probabilistic distance related computations.

The obvious conclusion regarding the usability of probabilistic distance measures is that they should be used to evaluate feature subsets only if there is no other choice. We have demonstrated that different measures may lead to considerably different results depending on data and on what further processing is to take place (classifier choice). The explanation may be in the fact that the underlying probability distributions of datasets do not fulfill the assumptions implicitly hidden behind the use of probabilistic measures as the feature selection criterion. There is no way of stating a general recommendation which measure is generally the most preferable. The best recommendation we can give is that dimensionality reduction should take place as part of classification tasks and that it should be principally connected to the functioning of the classifier (i.e. Wrappers are preferable to Filters).

The results also show a problem of practically all sequential subset search algorithms – the inclination to follow similar deterministic paths through the feature space that often end up in similar local extremes. The sequentiality as a property is thus a restricting concept in a sense. The randomisation employed in the OS algorithm initialisation phase partly overcomes this problem and gives a chance (but does not guarantee) to find different and possibly better extremes in the feature space. The results shown here illustrate the advantage clearly as all the best classification rates have been obtained using the simplest OS version with randomisation in contrary to the sophisticated but deterministic SFFS.

4.1 Acknowledgements

The work has been supported by grants of the Czech Ministry of Education CEZ:J18/98:31600001, ME567, and the Grant Agency of the Czech Republic No. 402/01/0981, No. 402/02/1271. We also thank Helmut Mayer from the University of Salzburg for letting us use his classifier code.

References

[1] B. Baesens, R. Setiono, C. Mues, and J. Vanthienen. Using Neural Network Rule Extraction and Decision Tables for Credit-Risk Evaluation. *Management Science*, 49(3):312–329, 2003.

[2] B. Baesens, T. Van Gestel, S. Viaene, M. Stepanova, J. Suykens, and J. Vanthienen. Benchmarking State of the Art Classification Algorithms for Credit Scoring. *Journal of the Operational Research Society*, forthcoming, 2003.

[3] R.H. David, D.B. Edelman, and A.J. Gammerman. Machine learning algorithms for credit-card applications. *IMA Journal of Mathematics Applied In Business and Industry*, 4:43–51, 1992.

[4] V.S. Desai, J.N. Crook, and G.A. Overstreet Jr. A comparison of neural networks and linear scoring models in the credit union environment. *European Journal of Operational Research*, 95(1):24–37, 1996.

[5] P.A. Devijver and J. Kittler. *Pattern Recognition: A Statistical Approach*. Prentice-Hall, 1982.

[6] F.J. Ferri, P. Pudil, M. Hatef, and J. Kittler, Comparative study of techniques for large-scale feature selection, in: E.S. Gelsema and L.N. Kanal (eds.) *Pattern Recognition in Practice IV*, (Elsevier Science B.V., 1994) pp. 403–413.

[7] K. Fukunaga, *Introduction to Statistical Pattern Recognition: 2nd edition.* (Academic Press, Inc. 1990).

[8] W.E. Henley and D.J. Hand. Construction of a k-nearest neighbour credit-scoring system. *IMA Journal of Mathematics Applied In Business and Industry*, 8:305–321, 1997.

[9] A.K. Jain and D. Zongker, Feature selection: Evaluation, application and small sample performance, *IEEE Transactions on Pattern Analysis and Machine Inteligence* **19** (1997) 153–158.

[10] L. Kanal and B. Chandrasekaran. On dimensionality and sample size in statistical pattern classification. Pattern Recognition, 3:225–234, 1971.

[11] R. Kohavi and G.H. John. Wrappers for Feature Subset Selection. *Artificial Intelligence*, 97(1-2): 273–324, 1997.

[12] P. Pudil, J. Novovičová and J. Kittler, Floating search methods in feature selection, *Pattern Recognition Letters* **15** (1994) 1119–1125.

[13] P. Somol, P. Pudil, J. Novovičová and P. Paclík, Adaptive floating search methods in feature selection, *Pattern Recognition Letters* **20**, 11/13 (1999) 1157–1163.

[14] P. Somol and P. Pudil, Oscillating search algorithms for feature selection, *Proceedings of the 15th International Conference on Pattern Recognition* (IEEE Computer Society, Los Alamitos, 2000) 406–409.

[15] A. Steenackers and M.J. Goovaerts. A credit scoring model for personal loans. *Insurance: Mathematics and Economics*, 8:31–34, 1989.

[16] L.C. Thomas. A survey of credit and behavioural scoring: forecasting financial risk of lending to customers. *International Journal of Forecasting*, 16:149–172, 2000.

Nr	Name	Type	Explanation
1	Checking account	nominal	1: $<$ 0 DM; 2: $\geq$ 0 and $<$ 200 DM; 3: $\geq$ 200 DM/salary assignments for at least one year; 4: no checking account
2	Term	continuous	
3	Credit history	nominal	0: no credits taken/all credits paid back duly; 1: all credits at this bank paid back duly; 2: existing credits paid back duly till now; 3: delay in paying off in the past; 4: critical account/ other credits (not at this bank)
4	Purpose	nominal	0: car (new); 1: car (old); 2: furniture/equipment; 3: radio/television 4: domestic appliances; 5: repairs; 6: education; 7: vacation; 8: retraining 9: business; 10: others
5	Credit amount	continuous	
6	Savings account	nominal	1: $<$ 100 DM; 2: $\geq$ 100 DM and $<$ 500 DM; 3: $\geq$ 500 and $<$ 1000 DM; 4: $\geq$ 1000 DM; 5: unknown/no savings account
7	Present employment since	nominal	1: unemployed; 2: $<$ 1 year; 3: $\geq$ 1 year and $<$ 4 years; 4: $\geq$ 4 and $<$ 7 years; 5: $\geq$ 7 years
8	Installment rate (% of disposable income)	continuous	
9	Personal status and sex	nominal	1: male,divorced/separated; 2: female, divorced/separated/married; 3: male, single; 4: male, married/ widowed; 5: female,single
10	Other parties	nominal	1: none; 2: co-applicant; 3: guarantor
11	Present residence since	continuous	
12	Property	nominal	1: real estate; 2: if not 1: building society savings agreement/life insurance; 3: if not 1/2: car or other; 4: unknown/no property
13	Age	continuous	
14	Other installment plans	nominal	1: bank; 2: stores; 3: none
15	Housing	nominal	1: rent; 2: own; 3: for free
16	Number of existing credits at this bank	continuous	
17	Job	nominal	1: unemployed/unskilled-non-resident; 2: unskilled-resident; 3: skilled employee/ official; 4: management/self employed/ highly qualified employee/officer
18	Number of dependents	continuous	
19	Telephone	nominal	1: none; 2: yes, registered under the customer name
20	Foreign worker	nominal	1: yes; 2: no

Table 4: Feature description for the German credit dataset. The features and values that have been selected as important to achieve the best classification rate in this experiment (83.8%) are underlined.

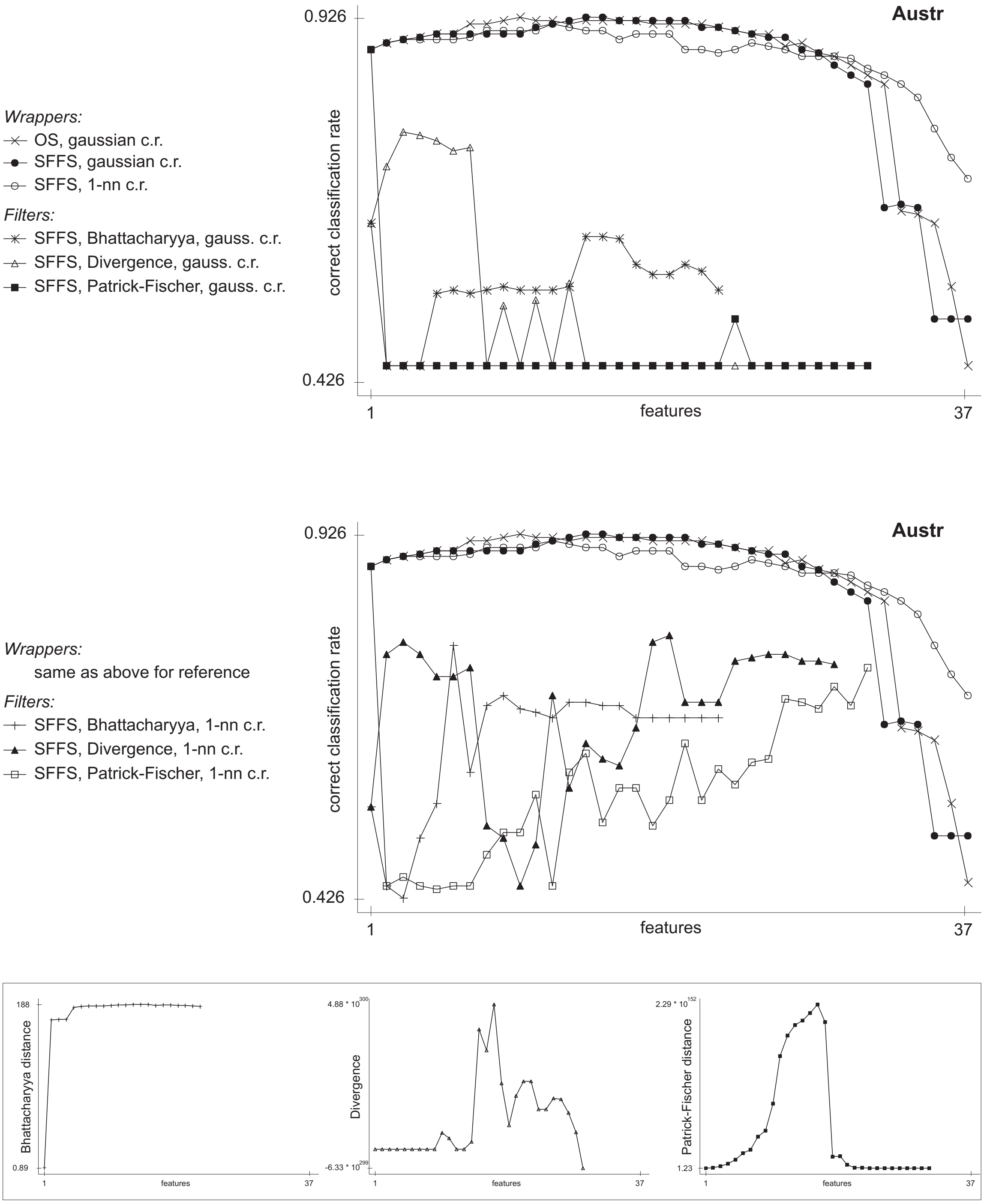

Figure 3: Results on the **'Austr'** dataset. Note especially the overall difference in Wrapper and Filter performance.

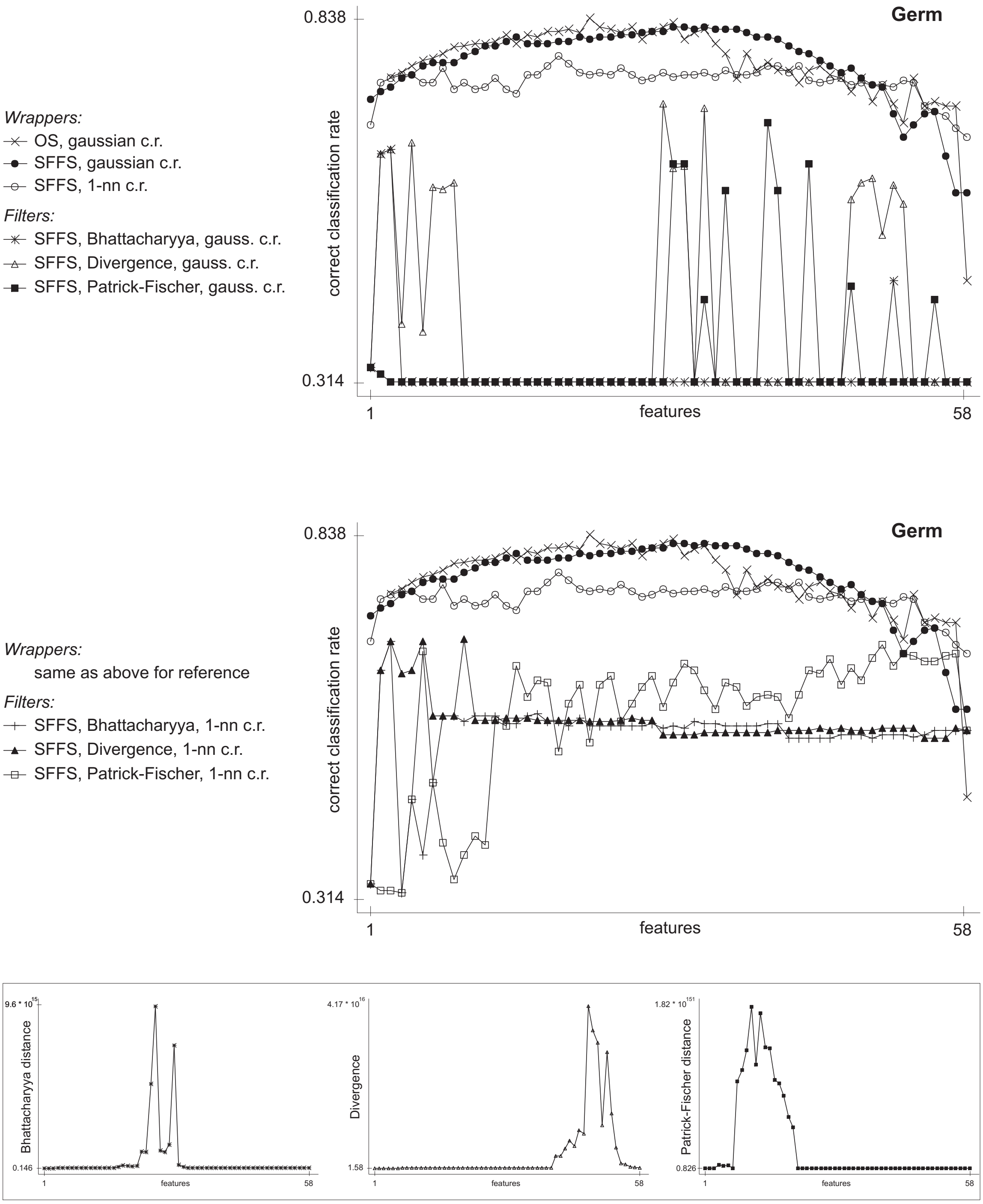

Figure 4: Results on the **'Germ'** dataset.

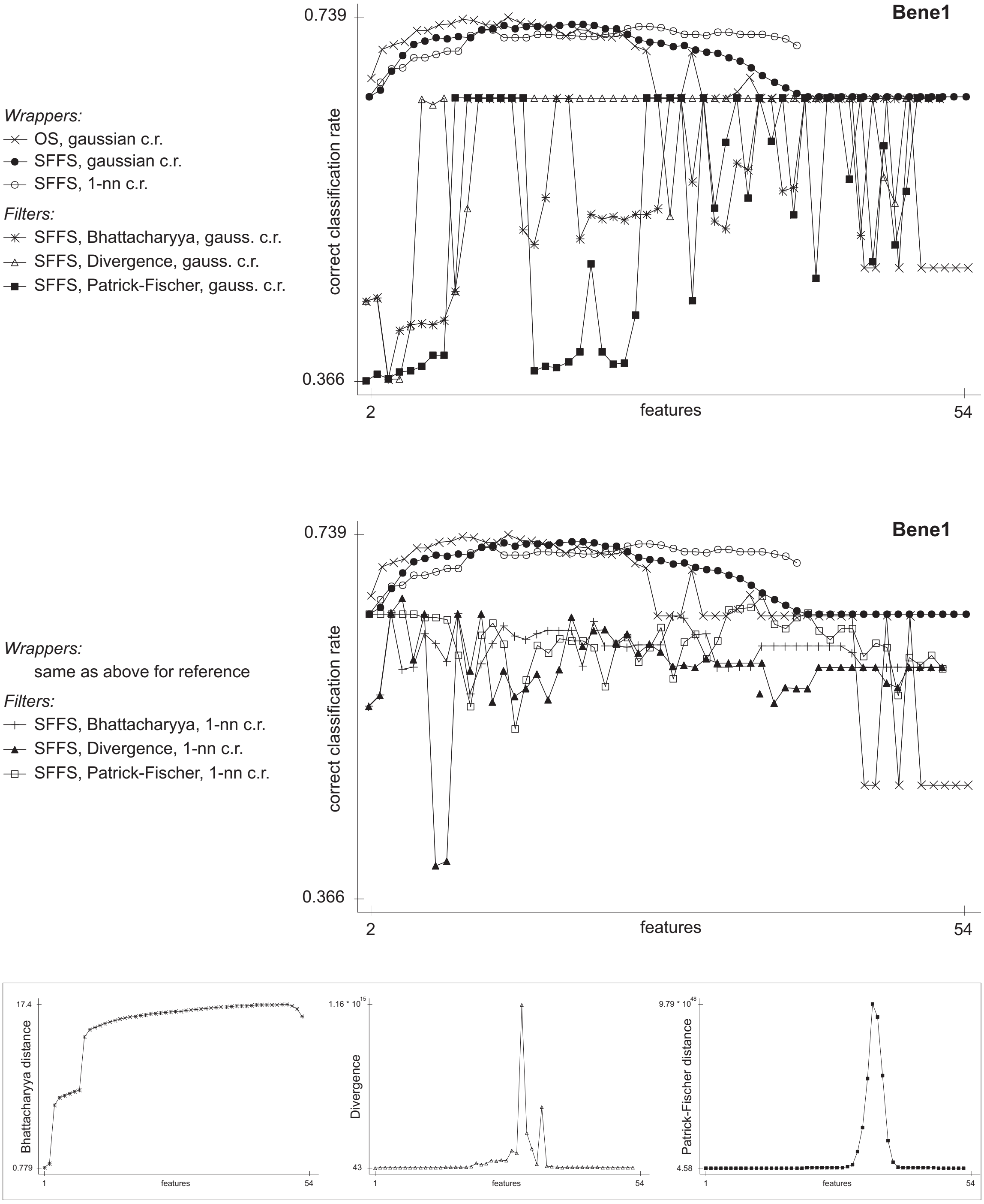

Figure 5: Results on the **'Bene1'** dataset.

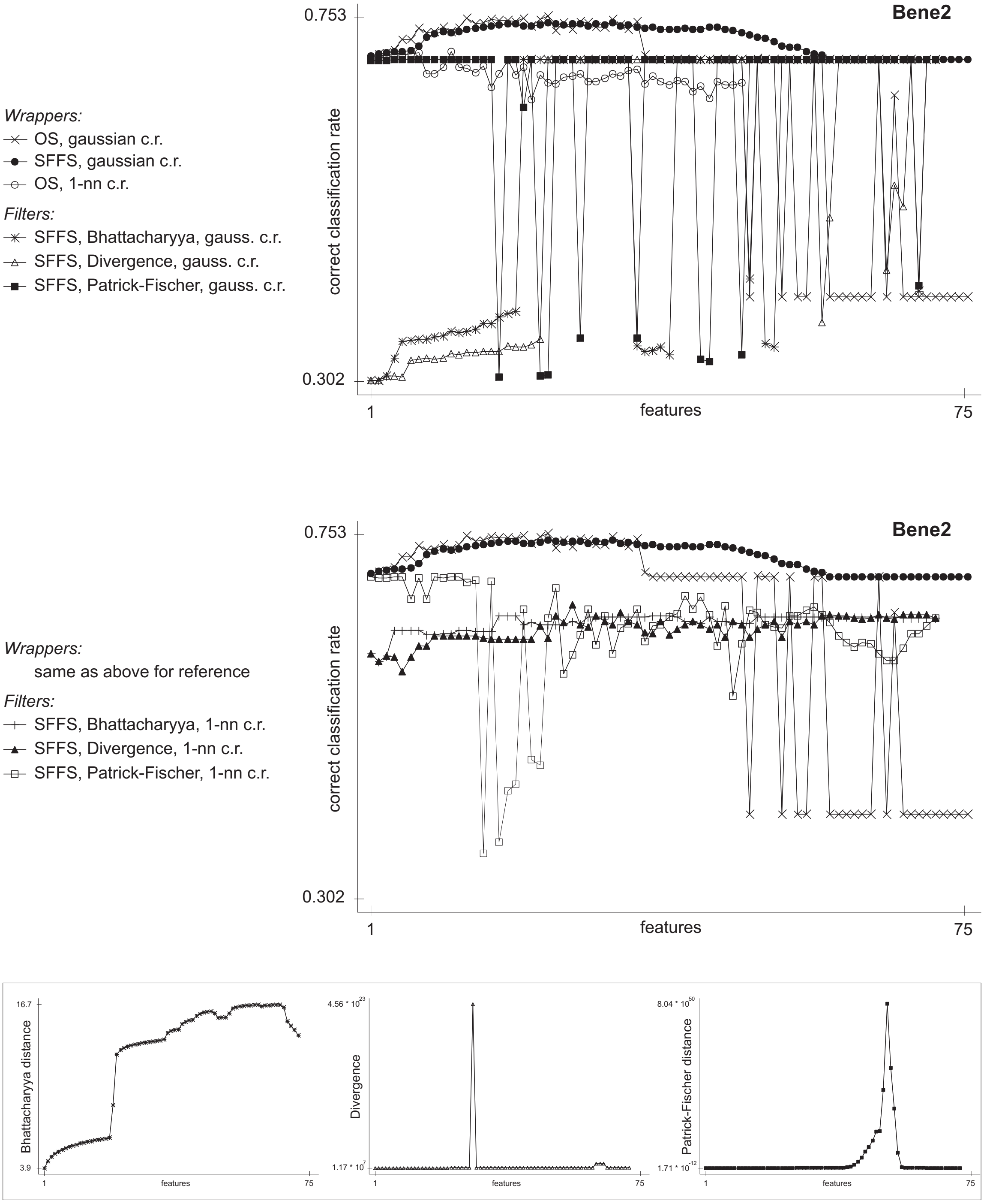

Figure 6: Results on the **'Bene2'** dataset.