On the use of data filtering techniques for credit risk prediction with instance-based models

V. García a, A.I. Marqués b, J.S. Sánchez a,*

a Institute of New Imaging Technologies, Department of Computer Languages and Systems, Universitat Jaume I, Av. Sos Baynat s/n, 12071 Castelló de la Plana, Spain
b Department of Business Administration and Marketing, Universitat Jaume I, Av. Sos Baynat s/n, 12071 Castelló de la Plana, Spain

A R T I C L E   I N F O

Keywords:
Finance
Credit risk
Instance selection
Outlier
Filtering
Editing
Nearest neighbour rule

A B S T R A C T

Many techniques have been proposed for credit risk prediction, from statistical models to artificial intelligence methods. However, very few research efforts have been devoted to deal with the presence of noise and outliers in the training set, which may strongly affect the performance of the prediction model. Accordingly, the aim of the present paper is to systematically investigate whether the application of filtering algorithms leads to an increase in accuracy of instance-based classifiers in the context of credit risk assessment. The experimental results with 20 different algorithms and 8 credit databases show that the filtered sets perform significantly better than the non-preprocessed training sets when using the nearest neighbour decision rule. The experiments also allow to identify which techniques are most robust and accurate when confronted with noisy credit data.

1. Introduction

Modern-day finance is a broad field of business activity that comprises hard decision-making problems related to risk management. Financial risk refers to the uncertainties associated with any form of financing, including credit risk, business risk, investment risk, market risk, and operational risk (Horcher, 2005). The focus of this paper is on credit risk, which denotes the probability that a loan to a borrower will not be repaid. In this context, the credit risk prediction models intend to estimate whether a new credit applicant will default (bad applicant) on a loan or not (good applicant). The development of reliable prediction models is of major importance because the use of inadequate credit risk assessment tools constitutes one of the main reasons of enterprise bankruptcy.

The development of financial prediction methods is a complex process, which involves data collection and preprocessing, model design, validation and implementation. With regard to model design, numerous strategies have been proposed, including statistical methods, computational intelligence techniques and operations research methodologies (Abdou & Pointon, 2011; Ince & Akta, 2009; Khandani, Kim, & Lo, 2010; Khashman, 2010; Wozabal & Hochreiter, 2012). In all cases, however, the quality of the data represents a critical point to further obtain accurate predictions. This mainly depends on the adequacy of the data in terms of the number of examples, the relevance of the attributes (independent variables) used in the analysis and the presence of outliers in the data set, among other issues. As a result, data preprocessing becomes a crucial step in real-world classification and prediction problems, as is the case of credit risk assessment.

Whilst some characteristics of data have largely been studied in the credit management literature, others have received relatively little attention so far. For instance, a lot of research effort has been devoted to the problem of attribute relevance by using new and existing feature selection algorithms (Chen & Li, 2010; Liu & Schumann, 2005; Piramuthu, 1999; Shukai, Chaudhari, & Dash, 2010; Wang, Hedari, Wang, & Ma, 2012). In contrast, despite its apparent influence on the performance of the prediction models, very few studies have addressed the problem of credit data with noise and outliers. Kotsiantis, Kanellopoulos, and Tampakas (2006) presented a survey of data preprocessing techniques for financial prediction, including discretization, feature selection and instance selection. Tsai and Chou (2011) used a genetic algorithm to perform feature selection and data filtering for bankruptcy prediction. Tsai and Cheng (2012) explored the performance of artificial neural networks, decision trees, logistic regression and support vector machines after removing different amounts of outliers from credit data sets.

Outliers have traditionally been defined as observations that appear to be inconsistent with the rest of the data. Nowadays, this term is being employed to cover a broad range of circumstances, including noisy and atypical data, new unidentified objects that do not belong to any of the predefined classes, and also mislabelled instances. Since the presence of outliers adversely affect the performance of any classification or prediction model, it is necessary to filter the data in order for eliminating the instances that disturb
the generalization process. Identifying outliers becomes especially important with instance-based learning methods, as for example is the case of the k nearest neighbours decision rule.

Although instance selection has proved to be effective in many data mining and knowledge discovery applications, it has not been fully explored in the domain of credit management. This paper faces the problem of outlier removal in the context of credit risk prediction with the k nearest neighbours model. To this end, several instance selection algorithms of different nature will be used to filter out outliers from the training set, thus trying to obtain a cleaner representation of the credit data. It has to be pointed out that the aim of this work is not to search for the most effective instance selection method, but to test whether the application of these preprocessing techniques produces some increase in the performance of credit risk prediction models.

2. Instance-based models

From a practical point of view, the credit risk prediction problem can be deemed as a binary classification problem where a new input sample (the credit applicant) must be categorized into one of the predefined classes based on a number of observed variables or attributes related to that sample. The input of the classifier consists of a variety of information that describes socio-demographic characteristics and economic conditions of the applicant, and then the classifier has to produce the output in terms of the applicant’s creditworthiness.

Formally, the credit risk prediction problem can be described as follows. Given a set of applicants \( T = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\} \), where each sample \( x_i \) is characterized by \( m \) attributes, \( x_{i1}, x_{i2}, \ldots, x_{im} \), and \( y_i \) denotes the class (good/bad applicant), then the credit risk classification problem consists of constructing a model \( \delta \) to predict the value \( y \) for a new input applicant \( x \), that is, \( \delta(x) = y \).

A large number of prediction models compute a distance between the input applicant \( x \) and the stored samples in \( T \) (called training set) when generalizing. These classification models are referred to as instance-based learning algorithms, where one of the most straightforward examples is the \( k \)-NN decision rule (Dasarathy, 1991).

The \( k \)-NN prediction model is a standard non-parametric technique used for probability density function estimation and classification. In brief, this classifier consists of assigning a new input sample to the class most frequently represented among the \( k \) closest instances in the training set, according to a suitable dissimilarity measure (e.g., Euclidean distance, Manhattan distance). A particular case is when \( k = 1 \), in which an input sample is decided to belong to the class indicated by its closest neighbour.

The characteristics of the \( k \)-NN classifier need the entire training set stored in computer memory, what may cause large time and memory requirements. On the other hand, this technique is extremely sensitive to the presence of noisy, atypical and erroneously labeled examples in the training set. Nevertheless, it has been found to perform better than other more sophisticated methods when dealing with arbitrarily complex problems, and it is a fairly intuitive procedure that could be easily implemented and explained to analysts.

3. Instance selection

In machine learning and data mining, the problem of instance selection is primarily related to instance deletion as irrelevant and harmful instances are removed from the original training set while retaining only critical instances. In general, the aim of the instance selection techniques is to produce a representative and relevant subset of the training set with similar or even higher generalization performance of the prediction model. An important advantage of instance selection against instance generation or abstraction refers to the fact that it chooses relevant examples without generating new artificial data, which do not make sense in many real-world applications. For instance, the incorporation of artificially generated samples into a credit data set may distort some socio-economic conditions of the problem in hand.

Let \( T \) be a training set and let \( R \subseteq T \) be the reference set, that is, the subset of examples selected by some instance selection algorithm. Usually, the reference set will be much smaller than the original training set (\( |R| < |T| \)). The class label \( y \) of a new input case \( x \) will be then estimated by a given prediction function \( \delta \) using the reference set \( R \) instead of \( T \). The general framework for the application of an instance selection algorithm is as follows: some instance selection procedure chooses an appropriate reference set \( R \) from the training set \( T \), then a learning algorithm is applied to build the prediction model \( \delta \) and finally, this is evaluated using an independent test set \( S \).

Traditionally, instance selection methods have been divided into two groups: editing and condensing. The goal of editing (or filtering) algorithms is to remove noisy, atypical and mislabelled instances and clean the possible overlapping between regions from different classes, thereby producing smoother decision boundaries and improving generalization. On the other hand, condensing (or thinning) algorithms aims at discarding superfluous or redundant instances that will not significantly affect the performance of the prediction model. Apart from these two general families of techniques, we can find hybrid methods that search for a small subset of the training set by simultaneously removing both outliers and superfluous instances.

According to their location in the input space, samples can be categorized into two main types: border and internal samples. Border samples are close to the decision boundaries, while internal samples are within the region of a class. Under this taxonomy, another factor that distinguishes instance selection techniques is whether they pursue to retain border points or internal points (Wilson & Martinez, 2000). The filtering algorithms seek to remove border samples because the internal examples are considered as the most representative of each class. Conversely, the condensing algorithms discard internal points under the idea that they do not affect the decision boundaries as much as the border samples and therefore, they can be removed with relatively little effect on the prediction process.

Yet another distinction among the instance selection algorithms can be made in terms of the direction of search for a subset of instances (Garcia, Derrac, Cano, & Herrera, 2012; Wilson & Martinez, 2000). An incremental search begins with an empty subset \( R \) and adds the training instances in \( T \) that fulfill some criterion; in this case, the order of presentation of instances can result very important. The decremental search starts putting all training instances into the reference set \( R \), and then searches for instances to be removed from \( R \); the order of presentation is also important, but unlike the incremental process, all training examples are available for examination at any time. Another way to apply an instance selection process is in batch mode, which involves deciding if each instance in \( T \) meets the removal criterion before discarding any of them; then, all instances that satisfy such a criterion are removed from \( R \) at once. Finally, a mixed search begins with an initially preselected subset \( R \), obtained either by random selection or by an incremental or decremental process, and then iteratively allows for additions or removals of instances that fulfill some criterion.

It has to be pointed out that only filtering techniques will be here taken into account because the focus of the present study is mainly on improving the performance of credit risk prediction models rather than on reducing the data set size. Some of the benefits of using filters are: (i) they make a prediction problem easier
because of smoothing the decision boundaries, (ii) they can sub-
stantially increase the class separability of the data in the input
space, (iii) they can improve the generalization performance of
the learning algorithm, and (iv) they reduce the computational
burden of the prediction model because as a by-product, the train-
ing set size is also reduced.

Fig. 1 illustrates the result of applying a filter to an exemplar
data set. As can be observed, the overlapping between both classes
is eliminated, the class separability is increased and the decision
boundaries are smoother, thus the prediction problem using the
filtered set (Fig. 1(b)) will result much easier than using the (unfil-
tered) original set (Fig. 1(a)).

From the vast amount of methods proposed in the literature, a
total of 20 filtering algorithms available in the KEEL software tool
(Alcalá-Fdez, Fernández, Luengo, Derrac, & García, 2011) will be
analysed in this paper. Table 1 summarizes the methods here used
and indicates the direction of search for each algorithm: D (decre-
mental), B (batch) or M (mixed). Note that none of these algo-
rithms works in an incremental mode since this is basically used
by the condensing approach.

4. Experimental set-up

The aim of the experiments is to evaluate the performance of an
instance-based credit risk prediction model when training data are
preprocessed to remove outliers and noisy examples. More specif-
ically, we are interested in finding out whether the 1-NN decision
rule applied on the filtered sets leads to better performance than
that applied on the original training sets. After this, we also try
to determine which filtering technique yields the best results.

All the experiments have been carried out on eight real-world
credit data sets, whose main characteristics are reported in Table
2. The Australian, German and Japanese data sets are from the
UCI Machine Learning Database Repository (http://archive.ics.u-
ci.edu/ml/). The Iranian data set comes from a modification to a
corporate client database of a small private bank in Iran (Sabzevari,
Soleymani, & Noorbakhsh, 2007). The Polish data set contains
bankruptcy information of 120 companies recorded over a two-
year period (Pietruszkiewicz, 2008). The Thomas data set, which
comes with the book by Thomas, Edelman, and Crook (2002),
describes applicants for a credit product. The UCSD data sets

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Acronym</th>
<th>Search</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>All-KNN editing</td>
<td>AllKNN</td>
<td>B</td>
<td>(Tomek, 1976)</td>
</tr>
<tr>
<td>Edited nearest neighbour</td>
<td>ENN</td>
<td>D</td>
<td>(Wilson, 1972)</td>
</tr>
<tr>
<td>Estimating class probabilities and threshold</td>
<td>EENN</td>
<td>D</td>
<td>(Vázquez et al., 2005)</td>
</tr>
<tr>
<td>Edited normalized radial basis function</td>
<td>ERBF</td>
<td>D</td>
<td>(Jankowski &amp; Grochowski, 2004)</td>
</tr>
<tr>
<td>Modified edited nearest neighbour</td>
<td>MENN</td>
<td>D</td>
<td>(Hattori &amp; Takahashi, 2000)</td>
</tr>
<tr>
<td>Model class selection</td>
<td>MCS</td>
<td>B</td>
<td>(Brodley, 1995)</td>
</tr>
<tr>
<td>Multiedit</td>
<td>MEdit</td>
<td>D</td>
<td>(Devijver, 1986)</td>
</tr>
<tr>
<td>Nearest centroid neighbours editing</td>
<td>NCN</td>
<td>D</td>
<td>(Sánchez et al., 2003)</td>
</tr>
<tr>
<td>Patterns by ordered projections</td>
<td>POP</td>
<td>B</td>
<td>(Riquelme et al., 2003)</td>
</tr>
<tr>
<td>Relative neighbourhood graph editing</td>
<td>RNG</td>
<td>D</td>
<td>(Sánchez et al., 1997)</td>
</tr>
<tr>
<td>CHC evolutionary algorithm</td>
<td>CHC</td>
<td>M</td>
<td>(Cano et al., 2003)</td>
</tr>
<tr>
<td>Generational genetic algorithm</td>
<td>GGA</td>
<td>M</td>
<td>(Cano et al., 2003)</td>
</tr>
<tr>
<td>Population based incremental learning</td>
<td>PBIL</td>
<td>M</td>
<td>(Cano et al., 2003)</td>
</tr>
<tr>
<td>Steady-state genetic algorithm</td>
<td>SGA</td>
<td>M</td>
<td>(Cano et al., 2003)</td>
</tr>
<tr>
<td>Automatic noise reduction</td>
<td>ANR</td>
<td>B</td>
<td>(Zeng &amp; Martinez, 2003)</td>
</tr>
<tr>
<td>Classification filter</td>
<td>ClassF</td>
<td>D</td>
<td>(Gamberger et al., 1999)</td>
</tr>
<tr>
<td>Cross-validated committee filter</td>
<td>CVF</td>
<td>B</td>
<td>(Verbaeten &amp; Assche, 2003)</td>
</tr>
<tr>
<td>Ensemble of filters</td>
<td>EnsF</td>
<td>B</td>
<td>(Brodley &amp; Friedl, 1999)</td>
</tr>
<tr>
<td>Iterative partitioning filter</td>
<td>IPF</td>
<td>B</td>
<td>(Khoshgoftaar &amp; Rebours, 2007)</td>
</tr>
<tr>
<td>Saturation filter</td>
<td>SatF</td>
<td>D</td>
<td>(Gamberger et al., 2000)</td>
</tr>
</tbody>
</table>
correspond to two subsets with samples randomly chosen from the database used in the 2007 Data Mining Contest organized by the University of California San Diego and Fair Isaac Corporation.

The standard way to assess credit risk prediction systems is to use a holdout sample since large sets of past applicants are usually available. However, there are situations in which data are too limited to build an accurate model and therefore, other strategies have to be used in order to obtain a good estimate of the predictive performance. The most common way around this corresponds to cross-validation (Thomas et al., 2002). Accordingly, a fivefold cross-validation has been adopted for the present experiments: each original data set has been randomly divided into five stratified parts of equal (or approximately equal) size. For each fold, four blocks have been pooled as the training data, and the remaining part has been employed as an independent test set. Ten repetitions have been run for each trial. The training data, and the remaining part has been employed as an independent

<table>
<thead>
<tr>
<th>Data set</th>
<th>#Attributes</th>
<th>#Good</th>
<th>#Bad</th>
</tr>
</thead>
<tbody>
<tr>
<td>Australian</td>
<td>14</td>
<td>307</td>
<td>383</td>
</tr>
<tr>
<td>German</td>
<td>24</td>
<td>700</td>
<td>300</td>
</tr>
<tr>
<td>Iranian</td>
<td>27</td>
<td>950</td>
<td>50</td>
</tr>
<tr>
<td>Japanese</td>
<td>15</td>
<td>296</td>
<td>357</td>
</tr>
<tr>
<td>Polish</td>
<td>30</td>
<td>128</td>
<td>112</td>
</tr>
<tr>
<td>Thomas</td>
<td>12</td>
<td>902</td>
<td>323</td>
</tr>
<tr>
<td>UCSD2500</td>
<td>38</td>
<td>1250</td>
<td>1250</td>
</tr>
<tr>
<td>UCSD5000</td>
<td>38</td>
<td>2500</td>
<td>2500</td>
</tr>
</tbody>
</table>

Type-I error (or miss) is the rate of bad applicants being categorized as good. When this happens, the misclassified bad applicants will become default. Therefore, if the credit granting policy of a financial institution is too generous, this will be exposed to high credit risk. Type-II error (or false-alarm) defines the rate of good applicants being predicted as bad. When this happens, the misclassified good applicants are refused and therefore, the financial institution has opportunity cost caused by the loss of good customers. As stated by Caouette, Altman, Narayanan, and Nimmo (2008), the misclassification costs associated with type-I errors are typically much higher than those associated with type-II errors and therefore, they are also more critical for determining higher credit risk for banks and financial institutions.

4.2. Statistical significance tests

A common way to compare two or more classifiers over various data sets is the Student’s paired t-test, which checks whether the average difference in their performance over the data sets is significantly different from zero. However, this appears to be conceptually inappropriate and statistically unsafe because parametric tests are based on a variety of assumptions (normality, large number of data sets, homogeneity of variance) that are often violated due to the nature of the problems (Demšar, 2006). In general, the non-parametric tests (e.g., Wilcoxon and Friedman tests) should be preferred over the parametric ones, especially in multi-problem analysis, because they do not assume normal distributions or homogeneity of variance (Demšar, 2006; Garcia, Fernández, Luengo, & Herrera, 2010).

The Wilcoxon signed-ranks test ranks the differences in performances of two algorithms for each data set, ignoring the signs, and compares the ranks for the positive and the negative differences. Let \(d_i\) be the difference between the performance scores of the two algorithms on \(i\)th out of \(N\) data sets. The differences are ranked according to their absolute values; in case of ties, average ranks are assigned (for example, if two differences are tied when assigning ranks 2 and 3, we can assign 2.5 to both differences). Let \(R^-\) be the sum of ranks for the data sets on which the first algorithm outperforms the second, and \(R^+\) the sum of ranks for the opposite. Ranks of \(d_i = 0\) are split evenly among the sums; if there is an odd number of them, one is ignored:

\[
R^- = \sum_{d_i < 0} \text{rank}(d_i) + \frac{1}{2} \sum_{d_i = 0} \text{rank}(d_i)
\]

\[
R^+ = \sum_{d_i > 0} \text{rank}(d_i) + \frac{1}{2} \sum_{d_i = 0} \text{rank}(d_i)
\]

(3)

Let \(Z\) be the smaller of the sums, \(Z = \min (R^+, R^-)\). If \(Z\) is less than or equal to the value of the distribution of Wilcoxon for \(N\) degrees of freedom, the null-hypothesis that both algorithms perform equally well can be rejected. However, in the case of multiple hypothesis testing, there exist other statistical methods more powerful than the Wilcoxon signed-ranks test (Demšar, 2006). Two of the most typical procedures for testing the differences between multiple prediction models are ANOVA and the non-parametric Friedman test.

The Friedman test is based on the average ranked performances of a collection of techniques on each data set separately. Under the null-hypothesis being tested, which states that all the algorithms are equivalent and the observed differences are merely random, the Friedman statistic can be computed as follows:

\[
\chi^2_F = \frac{12N}{K(K+1)} \left( \sum_{j=1}^{K} R_j^2 - \frac{K(K+1)^2}{4} \right)
\]

(4)
where \( N \) denotes the number of data sets, \( K \) is the total number of algorithms and \( R_i \) is the average ranks of algorithms. The \( 
abla^2_f \) is distributed according to the Chi-square distribution with \((K - 1)\) degrees of freedom, when \( N \) (number of data sets) and \( K \) (number of algorithms) are big enough. However, it has been demonstrated that the Friedman statistic produces an undesirably conservative effect. In order to overcome the conservativeness, Iman and Davenport (Iman & Davenport, 1980) proposed a better statistic distributed according to the \( F \)-distribution with \((K - 1) \) and \((K - 1)(N - 1)\) degrees of freedom:

\[
F_F = \frac{(N - 1)\nabla^2_f}{N(K - 1) - \nabla^2_f}
\]

If the null-hypothesis of the Friedman test is rejected, we can then proceed with a post hoc test in order to find the particular pairwise comparisons that produce significant differences. A post hoc test compares a control algorithm opposite to the remainder techniques, making possible to define a collection of hypothesis around the control method. The Nemenyi post hoc test, which is analogous to the Tukey test for ANOVA, states that the performances of two or more algorithms are significantly different if their average ranks are at least as great as their critical difference (CD) with a given level of significance (Demšar, 2006):

\[
CD = q_{a,n,K}\sqrt{\frac{K(K + 1)}{6N}}
\]

where \( q_{a,n,K} \) is a critical value based on the studentised range statistic divided by \( \sqrt{2} \).

5. Results

In this section, the experimental results are discussed. First, we compare the performance of the 1-NN decision rule using filtered sets with that obtained when no filtering has been applied. Next, we try to determine which method performs the best and evaluate for statistically significant differences among the different filtering algorithms presented in Table 1.

5.1. Do the filtered sets perform better than the original training sets?

Table 4 shows the accuracy rates achieved with the selected reference sets. For each database, the best result is highlighted in boldface. As can be seen, the credit risk prediction with the non-preprocessed sets (the original training sets) have consistently performed worse than the classification models using the filtered sets. In most cases, differences between the original set and the edited sets are high and relevant, what is of great value in financial applications because it may lead to significant cost savings for the credit institution. For example, the increase in accuracy is of 5.80% for the Australian database, 7.30% for the German database, 2.90% for the Iranian database, and 6.90% for the Japanese database.

Table 5 reports the set size reduction rates for each filtering method on each database. Although this is not the main objective of the present study, it is still interesting to analyse the size reduction produced by each algorithm because it is usual to find real databases with millions of records in banking and financial applications. Thus one can observe that the evolutionary-based filtering methods (CHC, GGA, PBIL and SGA) have achieved more than 90% of set size reduction rate, what means significant saving in time computing and storage requirements when compared to the unfiltered data set. Conversely, the ensemble-based filters, such as CVC, EnsF and IPF, have contributed to low size reduction rates with less than 10% in most cases: also the POP algorithm, which is a filter based on feature projection, has obtained reduction rates similar to those of the ensemble-based techniques.

To visualize the performance of a data filtering algorithm we can use a scatterplot of the size reduction rate versus the accuracy rate of 1-NN with the selected reference set. Since the accuracy rates are very different for the different data sets, using the average accuracy across the data sets will be inadequate. Instead we calculate ranks for the methods. For each data set, the best method receives rank 1, and the worst receives rank 21. As there are two criteria –accuracy rate and size reduction rate– each method will receive two ranks. Let \( r_i(i) \) and \( r_j(i) \) be the accuracy and the size reduction ranks for method \( i \), respectively evaluated on data set \( j \). Fig. 2 displays all the methods (including the unfiltered original set) in the space spanned by the average rank on size reduction rate \( r_j(i) = \frac{1}{7} \sum_j r_j(i) \) on the \( x \)-axis and the average rank on accuracy \( r_j(i) = \frac{1}{7} \sum_j r_j(i) \) on the \( y \)-axis. In such a way, points close to the origin \((0,0)\) of the plot correspond to the best methods in terms of a reasonably balanced trade-off between accuracy and size reduction.

One can observe in Fig. 2 that the 1-NN rule applied to the unfiltered (original) set lies the furthest from the origin of the plot, showing that it has achieved both the worst accuracy and the
On the other hand, as already seen in Table 5, the evolutionary filter is still far from the best methods in terms of reduction percentage. The technique that has yielded the highest accuracy corresponds to the RNG filter, but it is between accuracy and size reduction. The technique that has achieved accuracy rates significantly higher than the many other compared algorithms. The CHC evolutionary filter, which has produced the highest size reduction rates, is also among the best algorithms in terms of accuracy, performing significantly better than four other filtering methods.

Table 5
Set size reduction rates of the filtering algorithms.

<table>
<thead>
<tr>
<th></th>
<th>Australian</th>
<th>German</th>
<th>Iranian</th>
<th>Japanese</th>
<th>Polish</th>
<th>Thomas</th>
<th>UCSD2500</th>
<th>UCSD5000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>AllKNN</td>
<td>26.27</td>
<td>42.58</td>
<td>9.72</td>
<td>24.94</td>
<td>40.21</td>
<td>43.00</td>
<td>35.37</td>
<td>32.28</td>
</tr>
<tr>
<td>ENNTh</td>
<td>34.78</td>
<td>59.28</td>
<td>15.48</td>
<td>34.41</td>
<td>56.88</td>
<td>63.16</td>
<td>48.82</td>
<td>45.20</td>
</tr>
<tr>
<td>ERBF</td>
<td>14.02</td>
<td>29.25</td>
<td>5.00</td>
<td>13.49</td>
<td>37.40</td>
<td>26.37</td>
<td>47.70</td>
<td>29.10</td>
</tr>
<tr>
<td>MENN</td>
<td>35.00</td>
<td>61.53</td>
<td>15.60</td>
<td>34.67</td>
<td>57.40</td>
<td>63.92</td>
<td>50.13</td>
<td>46.26</td>
</tr>
<tr>
<td>MCS</td>
<td>10.87</td>
<td>17.13</td>
<td>3.78</td>
<td>10.65</td>
<td>11.77</td>
<td>17.96</td>
<td>12.29</td>
<td>10.26</td>
</tr>
<tr>
<td>MEdit</td>
<td>20.47</td>
<td>40.15</td>
<td>7.53</td>
<td>20.69</td>
<td>37.60</td>
<td>42.35</td>
<td>32.20</td>
<td>30.21</td>
</tr>
<tr>
<td>NCN</td>
<td>18.70</td>
<td>29.28</td>
<td>6.22</td>
<td>18.05</td>
<td>22.08</td>
<td>32.00</td>
<td>22.36</td>
<td>19.80</td>
</tr>
<tr>
<td>POP</td>
<td>5.65</td>
<td>54.28</td>
<td>29.45</td>
<td>4.90</td>
<td>0.00</td>
<td>3.53</td>
<td>0.12</td>
<td>0.31</td>
</tr>
<tr>
<td>RNG</td>
<td>16.34</td>
<td>29.35</td>
<td>5.55</td>
<td>16.13</td>
<td>23.75</td>
<td>28.86</td>
<td>22.56</td>
<td>20.52</td>
</tr>
<tr>
<td>GGA</td>
<td>96.63</td>
<td>93.90</td>
<td>99.03</td>
<td>97.47</td>
<td>92.29</td>
<td>95.37</td>
<td>91.66</td>
<td>90.44</td>
</tr>
<tr>
<td>PBIL</td>
<td>97.46</td>
<td>94.65</td>
<td>99.63</td>
<td>97.32</td>
<td>92.92</td>
<td>96.14</td>
<td>91.61</td>
<td>88.97</td>
</tr>
<tr>
<td>SGA</td>
<td>96.67</td>
<td>94.00</td>
<td>99.65</td>
<td>97.28</td>
<td>91.04</td>
<td>97.41</td>
<td>94.22</td>
<td>93.31</td>
</tr>
<tr>
<td>ANN</td>
<td>11.56</td>
<td>24.10</td>
<td>4.90</td>
<td>12.41</td>
<td>30.83</td>
<td>17.08</td>
<td>27.15</td>
<td>32.48</td>
</tr>
<tr>
<td>ClassF</td>
<td>19.60</td>
<td>31.35</td>
<td>7.60</td>
<td>19.23</td>
<td>22.81</td>
<td>34.41</td>
<td>20.63</td>
<td>18.14</td>
</tr>
<tr>
<td>CVC</td>
<td>2.46</td>
<td>2.97</td>
<td>2.08</td>
<td>3.22</td>
<td>0.83</td>
<td>16.06</td>
<td>1.25</td>
<td>1.30</td>
</tr>
<tr>
<td>EnsF</td>
<td>6.92</td>
<td>11.45</td>
<td>3.60</td>
<td>8.16</td>
<td>11.88</td>
<td>16.47</td>
<td>7.74</td>
<td>6.28</td>
</tr>
<tr>
<td>IPP</td>
<td>2.97</td>
<td>4.47</td>
<td>2.20</td>
<td>3.56</td>
<td>1.77</td>
<td>19.29</td>
<td>1.48</td>
<td>1.47</td>
</tr>
<tr>
<td>SatF</td>
<td>11.74</td>
<td>1.68</td>
<td>4.38</td>
<td>14.37</td>
<td>14.48</td>
<td>10.02</td>
<td>0.75</td>
<td>0.38</td>
</tr>
</tbody>
</table>

It is worth pointing out that the original training set has not been significantly better than any other algorithm. At a significance level of 0.05, 9 out of the 20 filters here tested have been significantly superior to the original set, while 8 techniques have performed significantly better than the unfiltered set for a significance level of 0.05. On the other hand, it appears that the best filtering methods correspond to RNG, EnsF, ENN and NCN, which have achieved accuracy rates significantly higher than those of the other compared algorithms. The CHC evolutionary filter, which has produced the highest size reduction rates, is also among the best algorithms in terms of accuracy, performing significantly better than four other filtering methods.

5.2. Which filtering method performs the best?

Analysing the results of the Wilcoxon signed-ranks test reported in Table 6, it is possible to draw the following remarks:

- The RNG method, which corresponds to the best filter in terms of accuracy rates, is significantly superior to AllKNN, POP, GGA, PBIL, SGA and SatF at a significance level of 0.05.
- The CHC method appears to be the best performing evolutionary-based filtering technique, being significantly superior to GGA and SGA at a significance level of 0.05, and better than PBIL and SGA at a significance level of 0.10.
- Among the methods based on Wilson’s editing, it seems that ENN and NCN are the best performing algorithms. At significance level of 0.05, ENN is significantly better than MCS, POP, PBIL and SatF, whereas NCN is significantly better than POP, PBIL and SatF.
- The POP algorithm is the worst filter: 12 out of the 20 methods are significantly better than POP at a significance level of 0.05.

Next, an additional statistical test will be carried out in order to support our preliminary conclusions and produce evidence of the pros and cons related to the filters here compared. Thus, as a way to further corroborate that there exist statistically significant differences in the accuracy rates, we have also computed the Iman-Davenport statistic (Eq. 5), yielding the value $F = 2.055$. The critical values for the F-distribution with $(20 – 1 = 19)$ and $(20 – 1) = 133$ degrees of freedom at confidence levels of 0.05 and 0.10 have been $F(19,133)_{0.05} = 1.665$ and $F(19,133)_{0.10} = 1.487$, respectively. Thus the null-hypothesis that all filters here...
explored perform equally well can be rejected and consequently, we can now apply a Nemenyi post hoc test in order to find out the set of techniques that are significantly worse than the control method (the algorithm with the best Friedman's rank).

The results of the Nemenyi test are depicted by significance diagrams (Lessmann, Baesens, Mues, & Pietsch, 2008), plotting the Friedman average ranks and the critical difference tail (calculated using Eq. 6). The diagram in Fig. 3 plots filtering algorithms versus average rankings, whereby all methods are sorted according to their ranks. The line segment to the right of each algorithm represents the critical difference \( \text{CD} = 10.483 \) for \( \alpha = 0.05 \) and \( \text{CD} = 9.818 \) for \( \alpha = 0.10 \). The vertical dotted line indicates the end of the best performing method. Therefore, all algorithms right to this line mean to perform significantly worse than the best filtering technique.

Fig. 3 shows that the only algorithms that have been significantly worse than the best performing method (RNG) correspond to POP for a level of significance of 0.05 and POP and SatF at a significance level of 0.10. This result is in agreement with the conclusions drawn from the Wilcoxon test in the sense that the POP algorithm is the worst filter for the credit databases used in these experiments. On the other hand, the fact that there are no other methods right to the vertical dotted line in Fig. 3 suggests that any of the remaining techniques could be appropriate for accurately filtering the training set. Therefore, when testing a number of filters for a real-world credit risk prediction problem, apart from the accuracy rate, one should also take care of other factors, such as the set size reduction rate or the type-I and type-II errors.

Here we will analyse the type-I and type-II errors because, in many banking and financial applications where the misclassification cost associated to each type of error may be different, it is especially important to evaluate the error on each class separately. As already stated in Section 4.1, the type-I errors are usually more critical for the financial institution since they typically lead to higher credit risks. Taking this feature into account, Tables 7 and 8 report the type-I and type-II errors derived from the filtering algorithms, respectively. For each database, the lowest error rates are highlighted in boldface.

Table 6
Summary of the Wilcoxon statistic for the filtering methods.

<table>
<thead>
<tr>
<th>(1) Original</th>
<th>(2) AllKNN</th>
<th>(3) ENN</th>
<th>(4) ENNTh</th>
<th>(5) ERBF</th>
<th>(6) MENN</th>
<th>(7) MCS</th>
<th>(8) MEdit</th>
<th>(9) NCN</th>
<th>(10) POP</th>
<th>(11) RNG</th>
<th>(12) CHC</th>
<th>(13) GGA</th>
<th>(14) PBIL</th>
<th>(15) SGA</th>
<th>(16) ANR</th>
<th>(17) ClassF</th>
<th>(18) CVC</th>
<th>(19) EnsF</th>
<th>(20) IPF</th>
<th>(21) SatF</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) Original</td>
<td>-</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
</tr>
<tr>
<td>(2) AllKNN</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(3) ENN</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(4) ENNTh</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(5) ERBF</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(6) MENN</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(7) MCS</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(8) MEdit</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(9) NCN</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(10) POP</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(11) RNG</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(12) CHC</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(13) GGA</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(14) PBIL</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(15) SGA</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(16) ANR</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(17) ClassF</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(18) CVC</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(19) EnsF</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(20) IPF</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(21) SatF</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>( \alpha = 0.10 )</td>
<td>0</td>
<td>6</td>
<td>5</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>0</td>
<td>8</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>7</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>( \alpha = 0.05 )</td>
<td>0</td>
<td>1</td>
<td>5</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>7</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>3</td>
<td>0</td>
</tr>
</tbody>
</table>

Fig. 3. Nemenyi’s significance diagrams for the filtering algorithms.
Despite the asymmetric misclassification costs, it is commonly accepted that a reliable prediction model should procure a well balanced trade-off between type-I and type-II error rates. Correspondingly, in a similar way as previously done for the accuracy and size reduction rates, the performance of the filtering algorithms can be visualized by means of a scatterplot of the type-II error versus type-I error plot for the filtering methods using average ranks.

The plot in Fig. 4 reinforces the comments made from the results in Tables 7 and 8. As can be observed, the POP algorithm is the most effective method, followed by the CHC and RNG algorithms. Despite the asymmetric misclassification costs, it is commonly accepted that a reliable prediction model should procure a well balanced trade-off between type-I and type-II error rates. Correspondingly, in a similar way as previously done for the accuracy and size reduction rates, the performance of the filtering algorithms can be visualized by means of a scatterplot of the type-II error versus type-I error plot for the filtering methods using average ranks.

The plot in Fig. 4 reinforces the comments made from the results in Tables 7 and 8. As can be observed, the POP algorithm is the most effective method, followed by the CHC and RNG algorithms. Despite the asymmetric misclassification costs, it is commonly accepted that a reliable prediction model should procure a well balanced trade-off between type-I and type-II error rates. Correspondingly, in a similar way as previously done for the accuracy and size reduction rates, the performance of the filtering algorithms can be visualized by means of a scatterplot of the type-II error versus type-I error plot for the filtering methods using average ranks.

The plot in Fig. 4 reinforces the comments made from the results in Tables 7 and 8. As can be observed, the POP algorithm is the most effective method, followed by the CHC and RNG algorithms. Despite the asymmetric misclassification costs, it is commonly accepted that a reliable prediction model should procure a well balanced trade-off between type-I and type-II error rates. Correspondingly, in a similar way as previously done for the accuracy and size reduction rates, the performance of the filtering algorithms can be visualized by means of a scatterplot of the type-II error versus type-I error plot for the filtering methods using average ranks.

The plot in Fig. 4 reinforces the comments made from the results in Tables 7 and 8. As can be observed, the POP algorithm is the most effective method, followed by the CHC and RNG algorithms. Despite the asymmetric misclassification costs, it is commonly accepted that a reliable prediction model should procure a well balanced trade-off between type-I and type-II error rates. Correspondingly, in a similar way as previously done for the accuracy and size reduction rates, the performance of the filtering algorithms can be visualized by means of a scatterplot of the type-II error versus type-I error plot for the filtering methods using average ranks.

The plot in Fig. 4 reinforces the comments made from the results in Tables 7 and 8. As can be observed, the POP algorithm is the most effective method, followed by the CHC and RNG algorithms. Despite the asymmetric misclassification costs, it is commonly accepted that a reliable prediction model should procure a well balanced trade-off between type-I and type-II error rates. Correspondingly, in a similar way as previously done for the accuracy and size reduction rates, the performance of the filtering algorithms can be visualized by means of a scatterplot of the type-II error versus type-I error plot for the filtering methods using average ranks.
among the best methods in terms of average rank on type-I error, but has achieved the highest average rank on type-II error. The opposite behaviour is shown by ALIKNN, ERBF, MENN, MEDit and ENNTh, which have attained the highest average ranks on type-I error and the lowest average ranks on type-II error. The ENN, NCN and CHC methods appear to yield the most balanced trade-off between both ranks, suggesting that these might be the most suitable techniques to filter out noisy data in a credit risk prediction domain.

6. Conclusions

This paper has presented a comprehensive study of noise filtering techniques when used for credit risk assessment with an instance-based prediction model. In particular, the performance of 20 algorithms (plus no filtering) on 8 real-world credit databases has been compared by means of the accuracy, the type-I error, the type-II error and the set size reduction rate. Statistical significance differences among the methods have been evaluated with the Wilcoxon signed-ranks and Friedman statistics, as well as the Nemenyi post hoc test.

From the experiments carried out, some conclusions can be drawn. First, it has empirically been demonstrated that the use of filters leads to significant improvement in performance and impressive saving in storage resources when compared to the nearest neighbour prediction model with no filtering. Second, the RNG algorithm appears to be the best filtering method in terms of overall accuracy, but it is only significantly superior to POP and SatF when using the Nemenyi post hoc test. This suggests that most filters can become useful for cleaning a credit data set and therefore, other factors have also to be taken into account in order for choosing the most suitable filtering method.

When measuring the type-I and type-II errors, we have found that the ENN, NCN and CHC algorithms yield the most balanced trade-off between both rates. It has also been observed that the class imbalance problem plays a key role in the results of the different filtering methods. In this sense, for instance, many algorithms have achieved a type-I error of 100% and a type-II error of 0% on the strongly imbalanced Iranian database, what is certainly of no practical value. Although the type-I error is typically judged as the most critical for banks and financial institutions, a reliable credit risk prediction model should still obtain a relatively low type-II error rate in order not to lose potentially good customers. Therefore, the POP, IPF, CVC and SatF algorithms, despite achieving the lowest type-I error rates in average, seem not to be appropriate filtering methods because of their high type-II error rates.

Future work will extend this study by using filters in conjunction with condensing algorithms in order to check whether they can achieve even better results. Another avenue for further research refers to analyse the combined problem of noise and class imbalance in credit risk prediction. Also, this work could be extended by using other classification models rather than instance-based learners.

Acknowledgment

This work has partially been supported by the Spanish Ministry of Education and Science under Grant TIN2009–14205, the Fundació Caixa Castelló-Bancaixa under Grant P1–1B2009–04, and the Generalitat Valenciana under Grant PROMETEO/2010/028.

References


