Meriem DJENNAS, Abdesslam BENDIABDELLAH et Mustapha DJENNAS

Readiness Assessment of A Business Intelligence System: Empirical Evidence From African Countries

Meriem DJENNAS
Faculty of economics, University of Amiens, France, meriemdjennas@gmail.com

Abdesslam BENDIABDELLAH
Faculty of economics, University of Tlemcen, Algeria, mbenbouziane@yahoo.fr

Mustapha DJENNAS
Faculty of economics, University of Tlemcen, Algeria, djennasm@yahoo.fr

Abstract:
Business intelligence warning system is of crucial importance in knowledge management strategies with the increasing competition and environment complexity. Although a variety of methods have been applied in this field, there are still some problems remained. The accurate and sensitive prediction in presence of unequal misclassification costs is an important one. Learning vector quantization (LVQ) is a powerful tool to solve early warning distress problem as a classification task. In this paper, a cost-sensitive version of LVQ is proposed which incorporates the cost information in the model. Experiments on a real data sets show the proposed approach is effective to improve the predictive capability in cost-sensitive situation.

1. INTRODUCTION

Nowadays, Economic Intelligence is widely considered as strong tool to take competitive advantage and to exploit opportunities in order to develop better methods for the identification of relevant sources of information and its manipulation to provide what the user needs for decision making. In broad terms, economic intelligence consist in the prevention and avoidance of all situations that can disrupt the life of companies or States and their success depends on their ability to anticipate, imagine new schemes, adapt norms and structures to the changes and to build networks of competences and cooperation in the worldwide economic intelligence systems. To achieve this set of goals, business applications utilize more specific learning processes and technologies to try and make better sense of the environment potentially enormous variability. Among these learning processes we can talk about Artificial Intelligence (AI) which is widely used for complex problem-solving and decision-support techniques in real-time business applications.

Hence, in the fiercely competitive and dynamic environment scenario, decision-making has become fairly complex and latency is inherent in many processes. In addition, the amount of data to be analyzed has increased substantially. AI technologies help organizations reduce latency in making business decisions and enhance competitiveness.

In this paper, we are going to present an approach to build a distress system to predict early warning signals in an intelligence economic system able to deal with the increasing competition and complexity of the macro-environment. Although a variety of methods have been applied in this kind of studies, there are still some problems remained. The accurate and sensitive prediction in presence of unequal misclassification costs is an important one. Learning vector quantization (LVQ) is a powerful tool to solve early warning distress prediction problem as a classification task. In this paper,
a cost-sensitive version of LVQ is proposed which incorporates the cost information in the model. Thus, our main objective is to find whether the proposed approach is effective to improve the predictive capability in cost-sensitive situation for an economic intelligence system or there are more robust AI tools to be used than the proposed one.

2. HISTORY OF BUSINESS INTELLIGENCE

In the 1970s and 1980s analytical software packages started showing up in the marketplace. However, lack of computing power, poor user friendliness, and cumbersome and manual integration with the transaction systems providing the data kept business intelligence (BI) tools from widespread usage. The release of spreadsheet software like Excel in the 1980s opened up for end users creating their own data models for business analysis. Spreadsheets are still widely used in this area today, and probably will be for many years to come. For a period in the 1980s and early 1990s, so-called executive information systems (EIS) grew in popularity with the promise that they would put key information on the desktops of executives. However, one of the biggest problems with EIS systems was that it took a lot of manual work to convert and load data from the data sources, as well as to maintain customized versions of the user screens. Major efforts in maintaining the EIS made many implementations short-lived (Rasmussen et al., 2003).

In the 1990s and the new millennium, with the widespread usage of SQL (standard query language) databases; datawarehouse technologies; extraction, transformation, and loading (ETL) tools; as well as new and powerful end-user analytical software, the stage is set for fast growth in usage of BI tools in the next decade (See Figure 1).

Furthermore, most of the BI software vendors have now released webbased versions of their solutions. Companies can now easily and at a low cost give users access to large amounts of corporate data and sophisticated analytical tools. By providing access to the Internet or an intranet connection, a person can investigate and analyze data from home, when traveling, or from any other location at which they may happen to be.

Figure 1. Evolution from static reports to business intelligence
Consequently, the business intelligence process is the matter of many branches of knowledge in management sciences as shown in figure. In this paper, we have tackled the data mining and optimization tools by introducing the neural network modeling in a business intelligence system. As a major branch of the neural network toolbox, we have chosen the Self-Organizing Maps (SOM) as a learning approach (Vercellis, C., 2009).

3. SELF-ORGANIZING AND COST-SENSITIVE LEARNING VECTOR QUANTIZATION NETWORKS METHODOLOGY

Self-organizing in networks is one of the most fascinating topics in the neural network field. Such networks can learn to detect regularities and correlations in their input and adapt their future responses to that input accordingly. The neurons of competitive networks learn to recognize groups of similar input vectors. Self-organizing maps learn to recognize groups of similar input vectors in such a way that neurons physically near each other in the neuron layer respond to similar input vectors.

Learning vector quantization (LVQ) is a method for training competitive layers in a supervised manner. A competitive layer automatically learns to classify input vectors. However, the classes that the competitive layer finds are dependent only on the distance between input vectors. If two input vectors are very similar, the competitive layer probably will put them in the same class. There is no mechanism in a strictly competitive layer design to say whether or not any two input vectors are in the same class or different classes.

LVQ networks, on the other hand, learn to classify input vectors into target classes chosen by the user (Kohonen, 1987).

The cost-sensitive LVQ resembles the basic batch LVQ except that the misclassification costs are utilized as weights guiding the prototype learning so that more attention is paid to the class associated with higher cost. During the training process, an input vector $x$ is projected to the best-matching unit (BMU), i.e., the winner with the closest prototype according to the distance measurement $d$. 
BMU\( (x) = \arg \min_{1 \leq i \leq m} d(x, m_i) \)  \hspace{1cm} (1)

The projection of input \( x_i \) \((1 \leq i \leq m)\) is defined by an indicative function \( h_{ip} \) whose value is 1 if \( m_p \) is the BMU of \( x_i \), and 0 otherwise:

\[
h_{ip} = \begin{cases} 1 & \text{if } m_p = \text{BMU}(x_i) \\ 0 & \text{otherwise} \end{cases}
\]  \hspace{1cm} (2)

Regarding the BMU, a Voronoi set \( V_i \) is generated for each neuron and composed of the observations projected to the neuron. In the Voronoi set, an element is positive if its class label agrees with the map neuron, and negative otherwise. Positive examples move the prototype towards the input while negative examples move the prototype away from them. Intuitively, the positive examples of relatively higher cost should impose more impact on the prototypes so that they are harder to be misclassified. The denotative function \( S_{ip} \) takes \( C_{\text{label}(x_i)} \) which is the misclassification cost associated with the class of observation \( x_i \), as the value in case of positive example, and -1 otherwise.

\[
S_{ip} = \begin{cases} C_{\text{label}(x_i)} & \text{if } \text{label } (m_p) = \text{label } (x_i) \\ -1 & \text{otherwise} \end{cases}
\]  \hspace{1cm} (3)

The indicative and denotative functions are then used in the prototype update, which combines the contribution of positive examples and suppression of negative examples to each neuron in a batch round. Let \( m_p(t) \) be the prototype vector of the \( p \)th unit at epoch \( t \). The update rule of cost-sensitive LVQ is formulated as follows (If the denominator is 0 or negative for some \( m_p \) no updating is done):

\[
m_p(t + 1) = \frac{\sum_{i=1}^{n} h_{ip} x_i}{\sum_{i=1}^{n} h_{ip}}
\]  \hspace{1cm} (4)

As a special case of SOM, the LVQ algorithms benefit from a trained map by a preceding SOM in the initialization (Kohonen, 1987). In one round, one instance \( x_i \) is input and the distance between \( x_i \) and prototypes is calculated, consequently the input is projected to the BMU according to Equation (1). After all the inputs are processed, the neurons are assigned by the majority of class labels in Voronoi set for acquiring the labeled map. In other words, if there are more good examples than bad examples, the unit is labeled as good and conversely. Then the values of indicative and denotative functions are calculated for each pair of input and neuron regarding Equation (2) and (3). Afterwards, the prototypes are updated according to Equation (4). This training process is repeated iteratively until the maximum number of iteration is reached or the amount of variation of prototypes between two consecutive iterations is less than a specified threshold. In summary, the algorithm is performed as follows:

1. For \( p = 1, ..., m \), initialize the map with prototypes \( m_p \);
2. For \( i = 1, ..., n \), input instance \( x_i \) to the map and project it to the BMU;
3. For \( p = 1, ..., m \), assign the class to \( m_p(t) \) by majority labeling principle;
4. For \( i = 1, ..., n, p = 1, ..., m \), calculate \( h_{ip} \) and \( s_{ip} \);
5. For \( p = 1, ..., m \), calculate the new prototype \( m_p(t + 1) \) for the next epoch;
6. Repeat from Step 2 a few iterations until the termination condition is satisfied.
4. EMPIRICAL SIMULATIONS

The proposed cost-sensitive LVQ algorithms are implemented based on som toolbox (Stork, Elad, 2003) in Matlab. We mainly concern about the effectiveness of the proposed algorithms on the tradeoff between two kinds of errors and the improvement on the total misclassification error rather than on the comparison with competing classification models.

4.1. Data Sets

In this paper, the capability of cost-sensitive LVQ is validated by a data set representing most important information for a business intelligence system. The data set contains information statements of 53 countries from Africa from January 1994 to March 2010 (16 years and 3 months). In order to diversify the number of examples in the data set, a Monte Carlo simulation is conducted to create weekly data. After simulation, the data set contain 780 examples. 10 years are used as the learning data set and 6 years as a testing and prediction set. The learning data set contain 400 bad examples and 380 good examples. The decision about good or bad examples is referenced to the variables thresholds according to the largest economy in the selected countries (South Africa). As described in Table 1, each country is characterized by a set of 43 variables besides the independent class. The problem is to predict whether a country is under a weak situation of it business intelligence system over a given period (one year).

Table 1. Neural network Input and output variables

<table>
<thead>
<tr>
<th>X_i</th>
<th>Variable description</th>
<th>X_i</th>
<th>Variable description</th>
<th>X_i</th>
<th>Variable description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X1</td>
<td>Annual real GDP growth</td>
<td>X16</td>
<td>Foreign Direct Investment inflows</td>
<td>X31</td>
<td>Access to electricity</td>
</tr>
<tr>
<td>X2</td>
<td>Demand Composition and growth rate</td>
<td>X17</td>
<td>Foreign Direct Investment outflows</td>
<td>X32</td>
<td>Water supply coverage (%)</td>
</tr>
<tr>
<td>X3</td>
<td>Total revenue and grants</td>
<td>X18</td>
<td>FDI inflows/GFCF</td>
<td>X33</td>
<td>Estimated adult literacy rate</td>
</tr>
<tr>
<td>X4</td>
<td>Total expenditure and net lending</td>
<td>X19</td>
<td>Inward FDI Potential Index</td>
<td>X34</td>
<td>Estimated youth literacy rate</td>
</tr>
<tr>
<td>X5</td>
<td>Inflation</td>
<td>X20</td>
<td>ODA net total, All donors</td>
<td>X35</td>
<td>Public expenditure on education</td>
</tr>
<tr>
<td>X6</td>
<td>Exchange Rate</td>
<td>X21</td>
<td>ODA net total, DAC countries</td>
<td>X36</td>
<td>High School Enrolment ratio:</td>
</tr>
<tr>
<td>X74</td>
<td>Broad Money</td>
<td>X22</td>
<td>ODA net total, Multilateral</td>
<td>X37</td>
<td>Unemployment rate</td>
</tr>
<tr>
<td>X8</td>
<td>Reserves</td>
<td>X23</td>
<td>Debt outstanding, at year end</td>
<td>X38</td>
<td>Inactivity rate</td>
</tr>
<tr>
<td>X9</td>
<td>Trade balance (USD million)</td>
<td>X24</td>
<td>Total debt outstanding (as % of GDP)</td>
<td>X39</td>
<td>Worker remittances (USD million)</td>
</tr>
<tr>
<td>X10</td>
<td>Current account balance (USD million)</td>
<td>X25</td>
<td>Debt service (as % of Exports of goods and)</td>
<td>X40</td>
<td>Corruption Perception Index (CPI)</td>
</tr>
</tbody>
</table>
The variables values are transformed with a logarithm calculation. The new values are then normalized in order to transform the maximum and the minimum value to 1 and 0 respectively.

\[
y = \begin{cases} \log(x + 1) & \text{if } x > 0 \\ -\log(1 - x) & \text{otherwise} \end{cases}
\]

(5)

\[
y = \frac{x - \min(x)}{\max(x) - \min(x)}
\]

(6)

4.2. Evaluation Criteria of the Cost-Sensitive Learning Vector

In the real life, the classification problems are commonly encountered. In general, most classifiers assume that the misclassification costs (false/bad and false/good cost) are the same. This assumption is not true. For example, in customer relationship management, the cost of mailing to non-buyers is less than the cost of not mailing to the buyers (Elkan, 2001). In this case, cost is not necessarily monetary, for examples, it can be a waste of time.

A misclassified good example is denoted by \( G_b \), and a misclassified bad example is denoted by \( B_g \). \( C_{ij} \) is the cost of predicting an example belonging to class \( i \) when in fact it belongs to class \( j \). The cost matrix or the confusion matrix is defined as follow:

<table>
<thead>
<tr>
<th>Predicted class</th>
<th>Actual class</th>
<th>Bad</th>
<th>Good</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bad</td>
<td>Bb</td>
<td>Bg</td>
<td>B</td>
<td></td>
</tr>
<tr>
<td>Good</td>
<td>Gb</td>
<td>Gg</td>
<td>G</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>b</td>
<td>g</td>
<td>T</td>
<td></td>
</tr>
</tbody>
</table>

Table 2. Cost (confusion) matrix

All the examples in the data set can be classified into class \( i \). Mathematically, we can define the \((i, j)\) entry in the cost matrix \( C \) the cost of predicting class \( i \) when the true class is \( j \). If \( i = j \), then the prediction is correct.
Otherwise, the prediction is wrong. The optimal prediction for an example \( x \) is the class \( i \) with the minimum expected cost by using the Bayes risk criterion (Chen, Marques, 2009):

\[
L(x, i) = \arg \min_i \left( \sum_{j \in \{b, g\}} P(j|x) \cdot C(i, j) \right)
\]

(7)

where \( P(j|x) \) is the posterior probability of classifying an example \( x \) as class \( j \). We assume that there is no cost for correct classifications, so the cost matrix can be described by the cost ratio:

\[
CR = \frac{C_{gb}}{C_{bg}}
\]

(8)

The purpose of Cost-Sensitive Learning is to build a model with minimum misclassification costs (total cost):

\[
T \cdot C = (C_{gb}) \cdot FB + (C_{bg}) \cdot FG
\]

(9)

Where: \( FB \) and \( FG \) are the number of false bad and false good examples respectively.

The most used assessment criteria for the predictive capability are:

- Type I error rate (fraction of good examples classified wrongly to bad classes): \( \frac{Gb}{G} \)
- Type II error rate (fraction of bad examples classified wrongly to good classes): \( \frac{Bg}{B} \)
- Overall error rate (percent of examples classified incorrectly): \( \frac{Gb+Bg}{Total \ examples} \).

The overall error treats two kinds of errors, namely type I error and type II error equivalently. Accordingly, the complementary rates denote the percent of observations classified correctly.

4.3. Experimental Results

The experiments are performed in the following steps:

1. The entire data set is divided randomly into 16 folds for cross-validation, in which 10 folds are used for model training, and the remaining is used for testing the generalization capability of the built model.
2. In each trial, the cost-sensitive LVQ algorithm is applied to the training data set.
3. For validation, each sample of the test data set is input to the resultant map and the predicted class is the label of the BMU.
4. After the experiment is repeated 10 times, the confusion matrix is calculated by comparing the real class to the predicted class for the entire data. Then the evaluation criteria are obtained from the confusion matrix.

The simulations produced 16 major classifications. The corresponding confusion matrix is summarized in Table 3. It can be concluded that the cost-sensitive LVQ is able to improve the predictive capability on the class with higher cost without great degradation on the other class. Since the misclassification cost on 'bad' category is higher, the classifier achieving lower type II error is preferred in practice.

The performance tendency can be detected in Figure 3, in which the left graph shows the error rates with respect to the varying cost ratios, and the right graph shows the cost ratio evolution.
### Table 3. Cost matrix

<table>
<thead>
<tr>
<th>Predicted class</th>
<th>Bad</th>
<th>Good</th>
<th>Total</th>
<th>Type I/II/overall error</th>
<th>Bad</th>
<th>Good</th>
<th>Total</th>
<th>Type I/II/overall error</th>
<th>Bad</th>
<th>Good</th>
<th>Total</th>
<th>Type I/II/overall error</th>
<th>Bad</th>
<th>Good</th>
<th>Total</th>
<th>Type I/II/overall error</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bad</strong></td>
<td>112</td>
<td>345</td>
<td>457</td>
<td>0.89164087</td>
<td>102</td>
<td>335</td>
<td>437</td>
<td>0.86880466</td>
<td>89</td>
<td>299</td>
<td>388</td>
<td>0.79336735</td>
<td>103</td>
<td>275</td>
<td>378</td>
<td>0.73880597</td>
</tr>
<tr>
<td><strong>Good</strong></td>
<td>288</td>
<td>35</td>
<td>323</td>
<td>0.75492241</td>
<td>298</td>
<td>45</td>
<td>343</td>
<td>0.76659039</td>
<td>311</td>
<td>81</td>
<td>392</td>
<td>0.77061836</td>
<td>297</td>
<td>105</td>
<td>402</td>
<td>0.72751323</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>400</td>
<td>380</td>
<td>780</td>
<td>0.81153846</td>
<td>400</td>
<td>380</td>
<td>780</td>
<td>0.81153846</td>
<td>400</td>
<td>380</td>
<td>780</td>
<td>0.78205128</td>
<td>400</td>
<td>380</td>
<td>780</td>
<td>0.73333333</td>
</tr>
<tr>
<td><strong>Cost matrix</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.83478261</td>
<td></td>
<td></td>
<td></td>
<td>0.88955224</td>
<td></td>
<td></td>
<td>1.04013378</td>
<td></td>
<td>1.08</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Bad</strong></td>
<td>112</td>
<td>251</td>
<td>363</td>
<td>0.69064748</td>
<td>125</td>
<td>231</td>
<td>356</td>
<td>0.64858491</td>
<td>114</td>
<td>233</td>
<td>347</td>
<td>0.66050808</td>
<td>132</td>
<td>180</td>
<td>312</td>
<td>0.57264957</td>
</tr>
<tr>
<td><strong>Good</strong></td>
<td>288</td>
<td>129</td>
<td>417</td>
<td>0.69146006</td>
<td>275</td>
<td>149</td>
<td>424</td>
<td>0.6488764</td>
<td>286</td>
<td>147</td>
<td>433</td>
<td>0.67146974</td>
<td>268</td>
<td>200</td>
<td>468</td>
<td>0.57692308</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>400</td>
<td>380</td>
<td>780</td>
<td>0.69102564</td>
<td>400</td>
<td>380</td>
<td>780</td>
<td>0.69102564</td>
<td>400</td>
<td>380</td>
<td>780</td>
<td>0.66538462</td>
<td>400</td>
<td>380</td>
<td>780</td>
<td>0.57435897</td>
</tr>
<tr>
<td><strong>Cost matrix</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.14741036</td>
<td></td>
<td></td>
<td></td>
<td>1.19047619</td>
<td></td>
<td></td>
<td>1.22746781</td>
<td></td>
<td>1.48888889</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Bad</strong></td>
<td>145</td>
<td>165</td>
<td>310</td>
<td>0.54255319</td>
<td>178</td>
<td>143</td>
<td>321</td>
<td>0.48366013</td>
<td>204</td>
<td>122</td>
<td>326</td>
<td>0.43171806</td>
<td>285</td>
<td>70</td>
<td>355</td>
<td>0.27088824</td>
</tr>
<tr>
<td><strong>Good</strong></td>
<td>255</td>
<td>215</td>
<td>470</td>
<td>0.53225806</td>
<td>222</td>
<td>237</td>
<td>459</td>
<td>0.44548287</td>
<td>196</td>
<td>258</td>
<td>454</td>
<td>0.37423313</td>
<td>115</td>
<td>310</td>
<td>425</td>
<td>0.1971831</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>400</td>
<td>380</td>
<td>780</td>
<td>0.53846154</td>
<td>400</td>
<td>380</td>
<td>780</td>
<td>0.46794872</td>
<td>400</td>
<td>380</td>
<td>780</td>
<td>0.40769231</td>
<td>400</td>
<td>380</td>
<td>780</td>
<td>0.23717949</td>
</tr>
<tr>
<td><strong>Cost matrix</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.5454555</td>
<td></td>
<td></td>
<td></td>
<td>1.55244755</td>
<td></td>
<td></td>
<td>1.60655738</td>
<td></td>
<td>1.64285714</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Bad</strong></td>
<td>301</td>
<td>55</td>
<td>356</td>
<td>0.23349057</td>
<td>345</td>
<td>28</td>
<td>373</td>
<td>0.13351354</td>
<td>369</td>
<td>15</td>
<td>384</td>
<td>0.07828283</td>
<td>385</td>
<td>7</td>
<td>392</td>
<td>0.03865979</td>
</tr>
<tr>
<td><strong>Good</strong></td>
<td>99</td>
<td>325</td>
<td>424</td>
<td>0.15449438</td>
<td>55</td>
<td>352</td>
<td>407</td>
<td>0.07056702</td>
<td>31</td>
<td>365</td>
<td>396</td>
<td>0.0390625</td>
<td>15</td>
<td>373</td>
<td>388</td>
<td>0.01785714</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>400</td>
<td>380</td>
<td>780</td>
<td>0.1974359</td>
<td>400</td>
<td>380</td>
<td>780</td>
<td>0.10641026</td>
<td>400</td>
<td>380</td>
<td>780</td>
<td>0.05897436</td>
<td>400</td>
<td>380</td>
<td>780</td>
<td>0.02820513</td>
</tr>
<tr>
<td><strong>Cost matrix</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.8</td>
<td></td>
<td></td>
<td></td>
<td>1.96428571</td>
<td></td>
<td></td>
<td>2.06666667</td>
<td></td>
<td>2.14285714</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
It is observed that the cost-sensitive learning is a good solution to the class imbalance problem by assigning different costs to different classes. Hence the proposed algorithm can be employed for distress signals prediction.

In a second stage of simulations, we have established a neural network for detecting early warning signals in a business intelligence system. Traditionally, the performance assessment of a warning tool is based on two measures which can be defined from the following matrix:

Table 4. Contingency Table

<table>
<thead>
<tr>
<th></th>
<th>Distress</th>
<th>Healthy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Signal</td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>No Signal</td>
<td>C</td>
<td>D</td>
</tr>
</tbody>
</table>

Let A represents the number of true signals released when a crisis is indeed taking place and B is the number of false or noise signals when no crisis is on stake. C is the number of false silences (no-signal) and D is the number of true silences. The table indicates if a signal (or a no-signal) occurs during one year (or 12 months).

We begin by assessing the quality of our system; we thus calculate conditional probabilities based upon the cell counts in the contingency table. We calculate the percentage of time over which the indicator released a signal when there was a crisis. In this case we are looking only at the crisis column of the contingency table to compute the probability that a signal was released. This probability is given by $A/(A+C)$. A high probability is associated with a good quality of the model. We also need to know how noisy the signal is. In particular, if no crisis occurs over the forecast horizon, we have to determine how often the indicator released a signal. Looking at the no crisis column of the contingency table, the ratio $B/(B+D)$ is calculated. A lower probability is a signal of a good model. Let the noise-to-signal ratio represent a measure of the background noise relative to the signal strength. The ratio is usually measured by the following equation:

$$NSR = \frac{B/(B+D)}{A/(A+C)}$$ (10)

Figure 3. Simulation results

![Simulation results](image)
The smaller the NSR is, the better the indicator is for signaling a currency crisis. Table 2 presents the performance results:

<table>
<thead>
<tr>
<th>Predicted class</th>
<th>Distress</th>
<th>Healthy</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Signal</td>
<td>72</td>
<td>4</td>
<td>76</td>
</tr>
<tr>
<td>Non signal</td>
<td>0</td>
<td>704</td>
<td>704</td>
</tr>
<tr>
<td>Total</td>
<td>72</td>
<td>708</td>
<td>780</td>
</tr>
<tr>
<td>A/A+C</td>
<td>100%</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>B/B+D</td>
<td>---</td>
<td>0,56%</td>
<td></td>
</tr>
<tr>
<td>NSR</td>
<td></td>
<td>0,0056</td>
<td></td>
</tr>
</tbody>
</table>

The performance obtained using neural networks is good for our forecasting horizon since the NSR approaches zero. This very small NSR is associated with significant coverage, i.e. 100% of distress signals. This implies that the proposed learning approach is a very promising.

Also, the most important crises in the considered period (from January 1994 to March 2010) are successfully captured, 72 distress signals identified as follow: South Africa global economic crisis in late 2008, the third and fourth quarters of 2008 (24 weeks). The South African electrical crisis in 2007, 8 months (32 weeks). Zimbabwe living standards deterioration in late 2008, 4 months (16 weeks). However, the model also released 4 false signals while there was no distress.

It can be concluded that the proposed approach performs well. The main reason is that the Self-organizing neural networks produce some intrinsic functions with different scales, which simplifies the problem. Furthermore, different functions with different scales include different information and, therefore, the neural network is able to extract more knowledge, thereby increasing its generalization ability.

5. CONCLUSIONS AND FUTURE REMARKS

Early warning signals in a Business intelligence system have been considered an important topic in many management domains to evaluate the risk associated in decisions concerning a state or a nation. Due to the presence of unequal misclassification costs in practical applications, the cost-sensitive classification is of particular importance to distress prediction. This paper presents a simple, yet reasonably effective modification of LVQ by integrating the cost information into the model. The results reveal the proposed algorithm is a good supporting tool in improving classification performance when the misclassification costs are unequal.

To sum up, this work provides the following main contribution: we proposed an estimate for the probabilities of distress occurrence in a business intelligence system, as well as an alternative to deal with the inherent nonlinearities of this problem. The main drawback of our work is the inability of this modeling approach to offer an economic explanation of the distress signal and to detect potential
economic indicators that are responsible for crisis. This limit comes from the fact that we have used only one endogenous indicator in our empirical study (healthy or distress situation).

6. REFERENCES

- Zadrozny, B., Elkan, C.: Learning and Making Decisions When Costs and Probabilities are Both Unknown. In: 7th International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, pp. 204-213 (2001)