Predicting Bankruptcies with the Self-Organizing Map

Kimmo Kiviluoto

Helsinki University of Technology
Laboratory of Computer and Information Science
P.O. Box 2200
FIN-02015 HUT, Espoo, Finland
kimmo.kiviluoto@@hut.fi

The Self-Organizing Map is used for analysis of financial statements, focusing on bankruptcy prediction. The phenomenon of going bankrupt is analyzed qualitatively, and companies are also classified into healthy and bankrupt-prone ones. In the qualitative analysis, the Self-Organizing Map is used in a supervised manner: both input and output vectors are represented in the weight vector of each unit, and during training, the whole weight vector is updated, but the best-matching unit search is based on the input vector part only. In the quantitative analysis, three classifiers that utilize the Self-Organizing Map are compared to Linear Discriminant Analysis and Learning Vector Quantization. A modification of the Learning Vector Quantization algorithm to accommodate the Neyman-Pearson classification criterion is also presented.

1 Introduction

Assessing the probability of bankruptcy of an enterprise is one of the key issues in a credit granting decision. Besides analyzing the strategy, personnel etc. of the firm, the financiers usually perform an analysis of the financial statements. One standard approach has been to use a mathematical model based on Linear Discriminant Analysis [1,2], but a wide variety of other statistical techniques have also been proposed. Recently, models utilizing neural networks have been introduced and compared with the "traditional" techniques – see e.g. [5,14,12,7,16,18,6,3,15,17].

The importance of the problem has made it something of a benchmark test for different models. Usually, in these tests the problem has been reduced to a classification of companies into healthy and non-healthy ones. There are two characteristics common to many of the reported studies: they are based on fairly small data sets, and the proportion of the bankrupt firms is much higher in the data than in the total population from which the sample is selected. This makes the results somewhat difficult to interpret. With small data sets, especially when the results are not cross-validated, the differences in classifier performance cannot be clearly distinguished from statistical noise; with biased sample, one may also get an over-optimistic view of the classifier performance on the total population. In the present study, we have tried to avoid these problems by using a very large sample consisting of nearly 5 000 financial statements, in which the ratio of healthy and bankrupt-prone firms is the same as in the base population.

2 The tools

The study consists of two parts: qualitative analysis and classification. Both parts utilize the Self-Organizing Map (SOM) [10]. In qualitative analysis, the SOM is used to form a "non-linear regression" from the input space into a plane; this makes it possible to visually examine the differences between firms that go bankrupt and those that do not. The idea is similar to that used in [6,13,3,9]. In classification, SOM is used as a vector quantizer, and also to determine the basis function centers of a Radial Basis Function (RBF) network (for a review of RBF networks, see e.g. [4]).

2.1 Qualitative Analysis

The input vectors $\mathbf{x}(t)$, $t = 1, \ldots, N$ consist of the financial indicators that are derived from the financial statements, and of binary indicator variables that correspond to whether the company went bankrupt within a certain time interval after giving that particular financial statement. As a preprocessing technique, the original values of the indicators are nonlinearly transformed using histogram equalization for each component separately.

Below, we shall denote those input vector components that contain the financial indicators by $\mathbf{x}^f(t)$, and those components that contain the bankruptcy indicator variables by $\mathbf{x}^b(t)$; thus, $\mathbf{x}(t) = [\mathbf{x}^{fT}(t)\ \mathbf{x}^{bT}(t)]^T$, superscript T denoting the vector transpose. The SOM model vectors $\mathbf{m}_i$ are correspondingly divided into two parts: $\mathbf{m}_i^f$ and $\mathbf{m}_i^b$.

The $\mathbf{x}^b(t)$ part of an input vector is not used in searching the winner unit, so that the winner index $c(t)$ is given by

$$c(t) = \underset{i}{\operatorname{argmin}} ||\mathbf{x}^f(t) - \mathbf{m}_i^f(t)|| \quad (1)$$

However, after the winner has been found, all the components of the whole model vectors are updated using the SOM rule

$$\mathbf{m}_j(t+1) := \mathbf{m}_j(t) + h_{c(t),j}[\mathbf{x}(t) - \mathbf{m}_j(t)] \quad (2)$$

where $h_{c(t),j}$ is a Gaussian neighborhood function, defined as in [10].

2.2 Classification

The classification of companies into healthy and non-healthy ones is done in two different ways: minimizing the total number of misclassifications, and using the Neyman-Pearson classification criterion [11]. With the latter technique, the type I error (classifying a bankrupt company erroneously as a healthy company) is fixed to a suitable value, and within this constraint the type II error (classifying a healthy company erroneously as a bankrupt company) is minimized. In practice, a classifier that is based on the Neyman-Pearson criterion would be the preferred one: type I error is much more costly than type II error, but because the proportion of non-bankrupt companies is higher, a classifier that minimizes the total number of misclassifications would pay more attention on minimizing the type II errors.

The classifiers used for minimizing the total number of misclassifications are Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), k-Nearest-Neighbour Classifier (kNN), and Learning Vector Quantization (LVQ). With the Neyman-Pearson criterion, the following classifiers are used: Linear Discriminant Analysis (LDA), Learning Vector Quantization (LVQ), Self-Organizing Map (SOM), and SOM-based Radial Basis Function Network (RBF-SOM).

The parameters needed for each classifier are determined using five-fold cross-validation, in which the sample is first divided into five different sets. The sets are otherwise random, but all the financial statements from any particular company are required to belong to the same set. Then, the model is trained using four of these sets, and its performance is evaluated on the fifth. This is repeated five times so that each set in turn is used as the validation set. The performance for the given parameters is finally obtained as the average performance on the validation sets.

The SOM-based classifiers are briefly discussed below, as is also the modification of the LVQ for the Neyman-Pearson criterion.

2.2.1 *SOM Classifiers*

The SOM is used for classification in two different ways, which we shall call SOM-1 and SOM-2; in addition to these, there is also a third classifier, RBF-SOM, that is partly based on the SOM.

With SOM-1 and SOM-2 classifiers, all the financial statements that are mapped to the same neuron are assigned the same class label, ie. the map consists of "bankruptcy units" and "non-bankruptcy units". The difference between these two models is in the labeling method of map units.

SOM-1 is based on a simple voting scheme: for each unit i, $P(\text{bankruptcy}|i)$ is estimated to be the proportion of the financial statements mapped to the i^{th} unit that were given by companies that had gone bankrupt. This conditional probability in turn is used as the classification criterion. If there are no input vectors mapped to some unit, the unit assumes the label of the class with higher *a priori* probability, which in this case is the class of healthy companies.

SOM-2 classifier utilizes a strategy similar to that used in qualitative analysis (see section 2.1). The estimate of conditional probability of bankruptcy given the unit, $P(\text{bankruptcy}|i)$, is based on a bankruptcy indicator component m_i^b of the model vector of that unit. These indicator components are not used in the winner unit search, but they are updated with the other components, as in equations (1) and (2). In effect, SOM-2 may be considered as a smoothed version of SOM-1; the smoothing is here carried out by the neighborhood function, but some other type of smoothing kernel could be used as well.

The third model that utilizes the SOM, here referred to as the RBF-SOM, is a standard RBF network (see [4]), in which the basis function locations are determined using the SOM. The training takes place in two steps:

(i) A SOM is trained using the vectors $\mathbf{x}^f(t)$ as inputs. For each map unit i is associated one Gaussian basis function $\phi_i(\mathbf{x}^f)$ that is centered on the model vector $\mathbf{m}_i^f$ of the map unit. The width β_i of the i^{th} Gaussian is found using a procedure proposed in [8]: set

$$\beta_i = \rho \min_k ||\mathbf{m}_i^f - \mathbf{m}_k^f|| \tag{3}$$

where the optimal value of parameter ρ – common to all basis functions – is searched using cross-validation.

(ii) The weights $\mathbf{W}$ connecting the basis functions to the output unit are chosen to minimize the error

$$\mathcal{E} = \frac{1}{2} \sum_t ||\mathbf{W}\boldsymbol{\phi}[\mathbf{x}^f(t)] - \mathbf{x}^b(t)||^2 \tag{4}$$

where vector $\boldsymbol{\phi}[\mathbf{x}^f(t)]$ consists of the individual basis functions $\phi_i(\mathbf{x}^f)$.

The minimization of the error $\mathcal{E}$ is easily accomplished by finding its gradient with respect to the weights, equating the gradient to zero, and solving the resulting group of linear equations.

2.2.2 Modifying LVQ for the Neyman-Pearson Criterion

The LVQ is usually used to minimize the total number of misclassifications; however, it can also accommodate the Neyman-Pearson criterion, as shown below. The original LVQ learning rule [10] is

$$\mathbf{m}_c^f(t+1) := \begin{cases} \mathbf{m}_c^f(t) + \alpha(t)[\mathbf{x}^f(t) - \mathbf{m}_c^f(t)] & \text{when } \mathbf{x}^f(t) \text{ and } \mathbf{m}_c^f(t) \text{ belong to the same class,} \\ \mathbf{m}_c^f(t) - \alpha(t)[\mathbf{x}^f(t) - \mathbf{m}_c^f(t)] & \text{otherwise} \end{cases} \tag{5}$$

Here $\mathbf{x}(t)$ is an input vector, $\mathbf{m}(t)$ is an LVQ prototype vector and $\alpha(t)$ is the learning rate parameter. Changing (5) to

$$\mathbf{m}_c^f(t+1) := \begin{cases} \mathbf{m}_c^f(t) + \alpha(t)\beta[\mathbf{x}^f(t) - \mathbf{m}_c^f(t)] & \text{when } \mathbf{x}^f(t) \text{ and } \mathbf{m}_c^f(t) \text{ belong to the same class,} \\ \mathbf{m}_c^f(t) - \alpha(t)(1-\beta)[\mathbf{x}^f(t) - \mathbf{m}_c^f(t)] & \text{otherwise} \end{cases} \tag{6}$$

the desired level of type I error results with the asymmetry parameter β, $0 < \beta < 1$ suitably chosen – exactly what is needed for a classifier employing the Neyman-Pearson criterion.

The motivation for the modified algorithm is similar to that of the original algorithm in [10]. Whereas the original algorithm (5) approximates the decision boundaries where

$$p(\mathbf{x}|\mathcal{C}_1)P(\mathcal{C}_1) - p(\mathbf{x}|\mathcal{C}_2)P(\mathcal{C}_2) = 0 \tag{7}$$

$\mathcal{C}_1$ and $\mathcal{C}_2$ being the classes, the modified version (6) approximates the boundaries

$$\beta p(\mathbf{x}|\mathcal{C}_1)P(\mathcal{C}_1) - (1-\beta)p(\mathbf{x}|\mathcal{C}_2)P(\mathcal{C}_2) = 0 \tag{8}$$

But this yields a classification rule that can be written as the likelihood ratio test for a Neyman-Pearson classifier:

$$\frac{p(\mathbf{x}|\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_2)} \underset{\mathcal{C}_2}{\overset{\mathcal{C}_1}{\gtrless}} \frac{(1-\beta)P(\mathcal{C}_2)}{\beta P(\mathcal{C}_1)} \equiv \lambda \tag{9}$$

Here, the classification threshold λ would be chosen so that the desired type I error rate is achieved. In practice, this is accomplished by finding a suitable value for β using cross-validation.

Let us note that with the Neyman-Pearson-LVQ, also the prototype vector initialization should be slightly modified. This speeds up the convergence of the algorithm and can also increase the classification accuracy by reducing the risk of the prototype vectors getting stuck in local minima. Ordinarily, the LVQ prototype vectors are initialized so that they are located within their respective classes, which is usually done using a kNN to check the prototype vector class labels. With the Neyman-Pearson-LVQ, an appropriately weighted kNN should be used instead.

3 Material

The material used in the present study represents a certain segment of Kera Ltd.'s customer companies. The segment consists of small and medium-sized Finnish industrial enterprises, from which the sample has been selected using the line of business, age, and size as the pruning criteria. It was also required that the history and state of the enterprise is known well enough: if there was no data available for a longer period than two years before the bankruptcy, or if the last known financial statements were very poor, so that the company had a high risk of going bankrupt within the next few years, the company was excluded from the sample. However, in excess to these criteria, no data was rejected because it was "atypical", or looked like an outlier.

The total number of financial statements used is 4 898; these have been given by 1 137 companies, of which 304 have gone bankrupt. The lengths of the known financial statement histories vary from 1 to 14 years, and their mode is 5 years. The financial indicators used here are operating margin, net income before depreciation and extraordinary items, net income before depreciation and extraordinary items of the previous year, and equity ratio.

4 Results and discussion

In the qualitative analysis, the SOM turned out to be a very valuable tool. Its main strength is, that the visual exploration of the financial indicator - bankruptcy risk space becomes possible. The capability of the SOM to locate similar input vectors to nearby units helps to find the common properties of companies mapped to some particular region of the map. Consequently, an unknown company can be easily and quite reliably characterized on the basis of its location on the map.

The U-matrix representation and Sammon projection (for an introduction to these techniques of visualizing the SOM, see [10]) of the map, displayed in figure 1, suggest that there are no clearly separated clusters in the input space but that the input vectors (to be precise, the $\mathbf{x}^f$-part of them, i.e. the financial indicators) come from a single cluster instead. The Sammon projection also shows that the input space can be rather well approximated with a two-dimensional SOM shaped like the one used here: no significant folding, to approximate an intrinsically higher-dimensional input space, or stretching, to approximate an input space of different shape, of the map is visible.

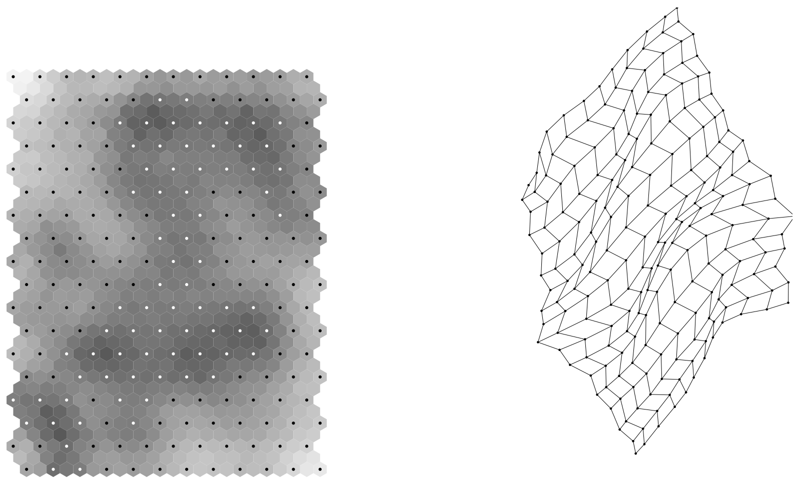

Fig. 1. U-matrix representation (left) and Sammon projection of the trained SOM

In figure 2, the relative values of the financial indicators on each unit of the map are displayed, together with the numbers of healthy and non-healthy companies that are mapped to the units. The two "natural dimensions" of the map may be roughly characterized as the profitability (vertical direction) and solidity (horizontal direction) of the company. Perhaps the most interesting, however, is the figure 3 which shows how the bankruptcy risk is related to the relative values of the financial indicators. The proportion of the failing companies is highest in the upper left corner of the map, and the shorter the time to bankruptcy, the stronger the tendency of the companies to have their financial statements located in this area.

The classification results trying to minimize the total number of misclassifications in are displayed in table 1; the results using the Neyman-Pearson criterion are in table 2. The results shown here are the average performances in the validation sets, using five-fold cross-validation.

In classification, it is clear that a classifier based on the Neyman-Pearson criterion is much more useful than one minimizing the total number of misclas-

Table 1
Classification results using some standard classifiers when minimizing the total number of misclassifications (per cent), based on financial statements given 2 ... 0 years before bankruptcy

Classifier	total error	error I	error II
LVQ	8,6	65,2	2,7
*k*NN (k=15)	8,5	75,2	1,5
LDA	10,5	47,1	6,6
QDA	11,1	55,9	6,5

Table 2
Classification results using Neyman-Pearson criterion with two different error I values (per cent), based on financial statements given 2 ... 0 years before bankruptcy

Classifier	error I target	total error	st.dev.	error I	st.dev.	error II	st.dev.
LDA	25	15,7	(1,0)	25,7	(5,4)	14,6	(1,5)
	30	14,1	(1,0)	29,5	(5,1)	12,5	(1,4)
LVQ	25	15,9	(0,8)	25,7	(5,4)	14,9	(1,6)
	30	14,3	(1,0)	30,3	(4,5)	12,5	(1,5)
RBF-SOM	25	15,8	(0,8)	26,4	(6,1)	14,7	(1,3)
	30	13,5	(1,0)	30,5	(6,4)	11,7	(1,6)
SOM-1	25	18,9	(2,8)	24,7	(6,3)	18,4	(3,4)
	30	18,2	(1,6)	30,8	(9,2)	16,9	(2,4)
SOM-2	25	16,6	(1,3)	25,4	(6,7)	15,7	(2,0)
	30	14,8	(0,4)	30,4	(6,8)	13,2	(0,9)

sifications. One trick that is often used in practice to make the latter classifier type work is to artificially balance the difference in the *a priori* probabilities of the classes. However, to do this by throwing away some of the financial statements given by the healthy companies would be wasting data, and such an artificial balancing would also result in an incorrect classification threshold which would have to be corrected somehow – and for many types of classifiers, e.g. LVQ, this would be a difficult if not impossible task.

Using the Neyman-Pearson criterion, most classifiers performed roughly at the same level, with the exception of the brute-force-approach SOM-1 that was clearly outperformed by other classifiers. The SOM-2 worked much better: the smoothing by the Gaussian neighborhood function seems to make it surprisingly insensitive to the number of units, as shown in figure 4, where the performance of the SOM-1 and SOM-2 classifiers as a function of the map size is depicted.

SOM with RBF performed slightly better than any other classifier considered here, although its generalization properties appear to be highly sensitive to the value of the kernel width parameter ρ. However, its performance was not significantly different from that of LDA and LVQ, and LDA has an additional advantage of simplicity – the only parameter that needs to be chosen is the classification threshold. An interesting alternative to these classifiers might be a weighted kNN for Neyman-Pearson classification task; at least the performance of the kNN classifier in minimizing the total number of misclassifications was promising.

An important next step in this study would be using information from several consecutive years. It seems that the information used here is not enough in all cases; for instance, a fast growing company with rather low solidity has quite different prospects than another company which also has low solidity but no growth. The state of a company during a few consecutive years might also reveal important factors concerning the ability of the management to adapt to the changing environment.

Acknowledgements

This project has been financially supported by Kera Ltd., which is gratefully acknowledged. Special thanks are due to Mr. Pentti Bergius whose application field expertise, guidance and support has been of great value throughout the project. Thanks are also due to the two anonymous reviewers for helping to improve the readability of this paper.

References

[1] E. I. Altman. Financial ratios, discriminant analysis, and the prediction of corporate bankruptcy. *The Journal of Finance*, (4):589–609, September 1968.

[2] E. I. Altman. *A complete guide to predicting, avoiding, and dealing with bankruptcy.* John Wiley & Sons, Inc., 1983.

[3] B. Back, G. Oosterom, K. Sere, and M. van Wezel. A comparative study of neural networks in bankruptcy prediction. In *Multiple Paradigms for Artificial Intelligence (STeP94).* Finnish Artificial Intelligence Society, 1994.

[4] C. M. Bishop. *Neural Networks in Pattern Recognition.* Oxford University Press, Oxford, 1995.

[5] D. Cadden. Neural networks and the mathematics of chaos – an investigation of these methodologies as accurate predictors of corporate bankruptcy. In *Proceedings of the First International Conference on Artificial Intelligence on Wall Street*, pages 52–57, Los Alamitos, CA, USA, October 1991. IEEE, IEEE Comput. Soc Press.

[6] B. M. del Brío and C. Serrano-Cinca. Self-organizing neural networks for the analysis and representation of data: Some financial cases. *Neural Computing & Applications*, 1:193–206, 1993.

[7] D. Fletcher and E. Goss. Forecasting with neural networks: an application using bankruptcy data. *Information and Management*, 24(3):159–167, March 1993.

[8] A. Hämäläinen. *Self-organizing map and reduced kernel density estimation.* PhD thesis, Rolf Nevanlinna Instituutti, Helsingin yliopisto, September 1995.

[9] S. Kaski and T. Kohonen. Exploratory data analysis by the self-organizing map: Structures of welfare and poverty in the world. In A.-P. Refenes, Y. Abu-Mostafa, J. Moody, and A. Weigend, editors, *Neural Networks in Financial Engineering. Proceedings of the Third International Conference on Neural Networks in the Capital Markets*, Singapore, 1996. World Scientific.

[10] T. Kohonen. *Self-Organizing Maps.* Springer Series in Information Sciences 30. Springer, Berlin Heidelberg New York, 1995.

[11] J. Neyman and E. S. Pearson. On the problem of the most efficient tests of statistical hypotheses. *Phi. Trans. Roy. Soc. London*, 231:289–337, 1933.

[12] W. Raghupathi, L. Schkade, and B. Raju. A neural network application for bankruptcy prediction. In V. Milutinovic and B. Shriver, editors, *Proceedings of the Twenty-Fourth Annual Hawaii International Conference on System Sciences*, volume 4, pages 147–155, Los Alamitos, CA, USA, Jan 1991. IEEE, IEEE Comput. Soc. Press.

[13] C. Serrano-Cinca. Beyond Z-analysis: Self-organizing neural networks for financial diagnosis. Discussion Papers in Acconting and Management Science 94-92, University of Southampton, Dec. 1994.

[14] K. Tam and M. Kiang. Managerial applications of neural networks: the case of bank failure predictions. *Management Science*, 38(7):926–947, July 1992.

[15] J. Tsukuda and S.-I. Baba. Predicting Japanese corporate bankruptcy in terms of financial data using neural networks. *Computers & Industrial Engineering*, 27:445–448, September 1994.

[16] G. Udo. Neural network performance on the bankruptcy classification problem. *Computers and Industrial Engineering*, 25(1-4):377–380, 1993.

[17] R. Wilson and R. Sharda. Bankruptcy prediction using neural networks. *Decision Support Systems*, 11(5):545–557, June 1994.

[18] Youngohc Yoon, G. Swales, Jr., and T. Margavio. A comparison of discriminant analysis versus artificial neural networks. *Journal of the Operational Research Society*, 44(1):51–60, January 1993.

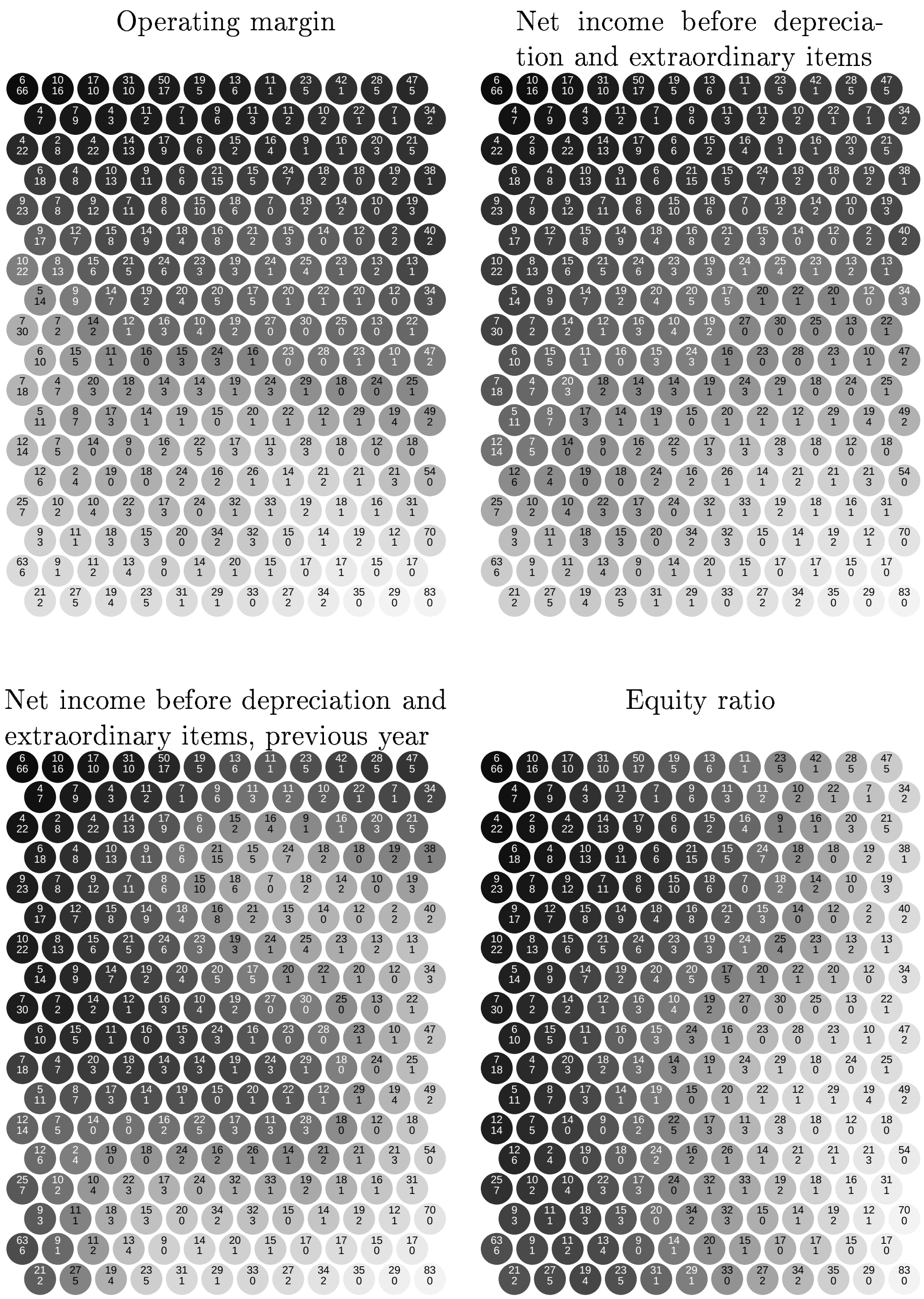

Fig. 2. The relative values of the financial indicators – the lighter the color, the better the relative value. The number of the healthy (upper figure) and bankruptcy (lower figure) companies that have been mapped to each map unit is also shown; here a company has been considered as a healthy one, if it has not gone bankrupt within five years.

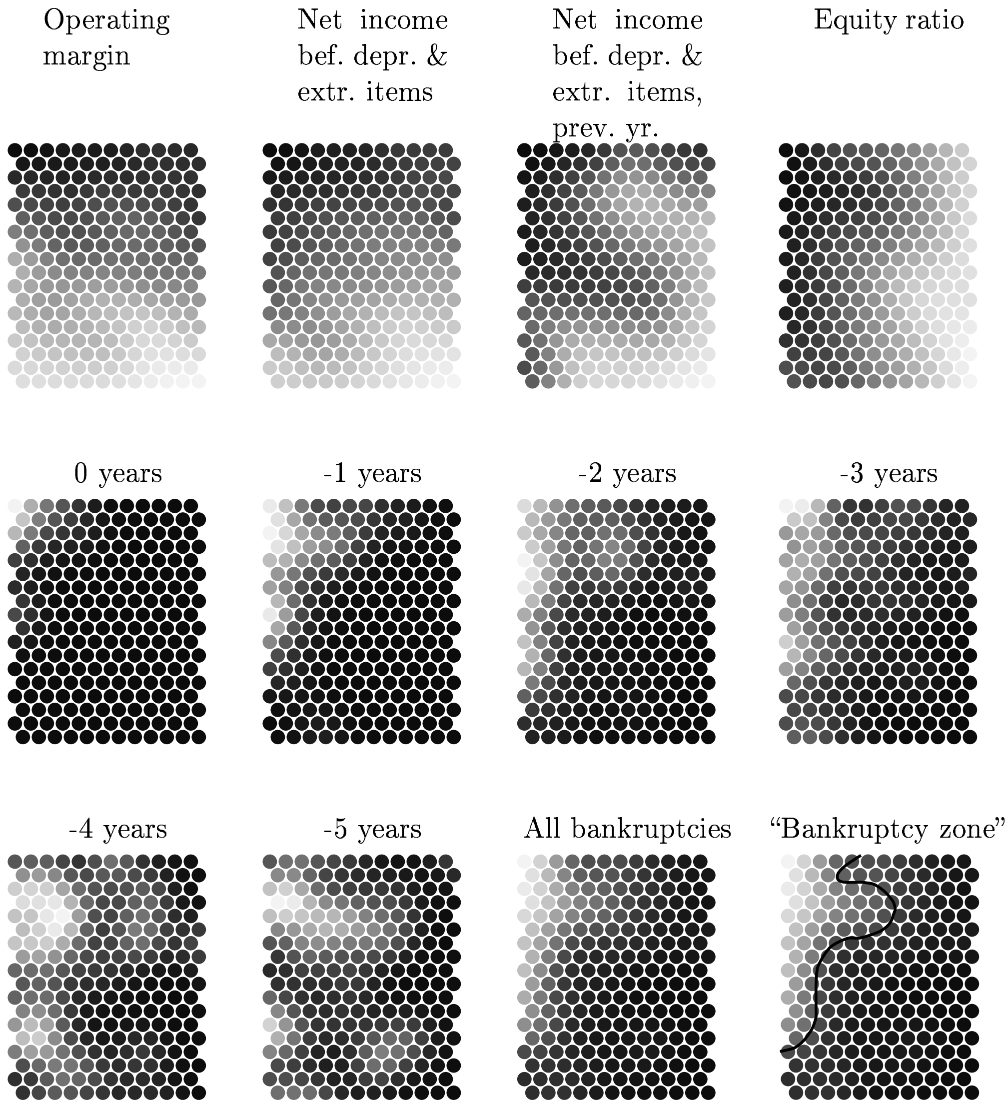

Fig. 3. Financial indicators vs. bankruptcies, depicted 5 … 0 years before the bankruptcy. On the upmost row, light color corresponds to good values of the financial indicators; on the two lower rows, light color corresponds to higher proportions of bankruptcy companies. In the rightmost panel of the bottom row, a "bankruptcy zone" is drawn: more than one third of the companies that are projected on the left side of the line have gone bankrupt within five years.

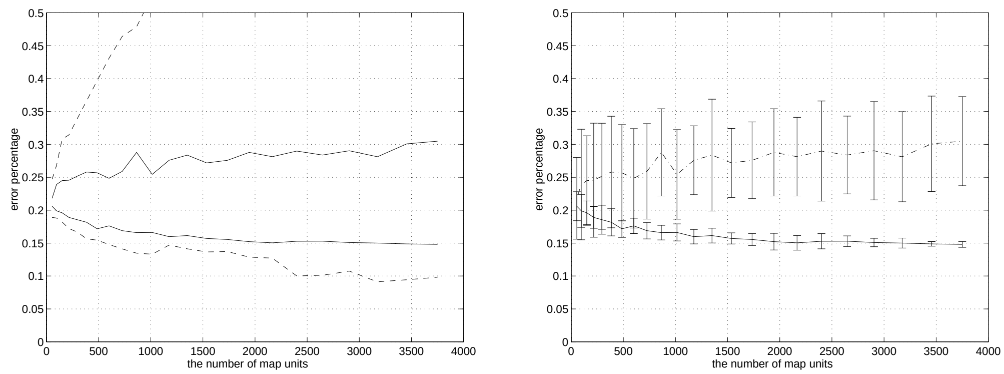

Fig. 4. On the left: the classification accuracies of the SOM-1 (dashed line) and SOM-2 (solid line) classifiers vs. the number of map units – the lower lines represent the percentage of all misclassifications, the upper lines type I misclassifications (classifying a bankrupt company erroneously as a healthy company). On the right: SOM-2 classifier performance standard deviations, using 5-fold cross-validation. Again, the lower line represents the percentage of all misclassifications, the upper line type I misclassifications.