Comparing 2D and 3D Self-Organizing Maps in Financial Data Visualization

Kimmo Kiviluoto

Laboratory of Computer and Information Science
Helsinki University of Technology
P.O. Box 2200, FIN-02015 HUT, Finland
Email:Kimmo.Kiviluoto@hut.fi

Keywords: self-organizing map, data visualization

Abstract

The self-organizing map is used to visualize financial statement data. The effect of increasing the map dimension from two to three is first demonstrated in a three-dimensional toy data example; then maps of both dimensionalities are used to visually explore financial data. It turns out that there are cases where two-dimensional map suggests that the data has separate clusters sharing some common property, but three-dimensional map only finds a single cluster. This is most likely a result of the two-dimensional map folding itself into the input space that has an intrinsic dimension higher than two, which produces artifacts appearing as separate clusters.

1 Introduction

The Self-Organizing Map (SOM) [8] has been used in a large number of diverse applications. In many of these, the focus has been on visualizing high-dimensional data. This is a task that suits the SOM well. The basic SOM algorithm finds a mapping from a high-dimensional input space onto a low-dimensional map, and because the mapping is constrained so that it tends to preserve the topological relations between the data vectors, it can be used as a basis for a number of different visualization techniques.

Usually, the SOM units have been arranged into a two-dimensional lattice. This is despite the fact that in some cases a higher-dimensional lattice has better theoretical justification, as shortly discussed in the next section. One obvious practical reason for the popularity of the two-dimensional SOM is that it is straightforward to print out and view; another, related issue is that most existing software implementations of SOM support only one- and two-dimensional lattices. Yet another, slightly more involved reason might be that the border effect [8] of the SOM becomes more pronounced when the lattice dimension increases.

2 SOM and data dimensionality

Let us assume that our data lives in some M-dimensional data manifold $\mathcal{D}$ that is embedded in an N-dimensional input space: $\mathcal{D} \subset \mathbb{R}^N$ and $N \geq M$ (for a rigorous definition of the data manifold, see [3]). Our data then has an intrinsic dimensionality M.

In practice there is always some noise present in the observations, which blurs the data manifold so that it starts looking N-dimensional at a length scale comparable to the standard deviation of the noise. However, it can still be approximated well by a M-dimensional manifold. In fact, the models used to describe or analyze the data need not – and usually should not – have higher dimensionality than M: there is no point in trying to model the noise or in building a model that is more complicated than is necessary. On the other hand, it is sometimes advantageous to use models of lower dimensionality than M to capture only the few most important aspects of the data manifold.

As an example, a data manifold that is an M-dimensional Gaussian could be easily modeled using the classical Principal Component Analysis (PCA). The M first eigenvalues of the data covariance matrix that correspond to the signal subspace are larger than the remaining $N-M$ eigenvalues that correspond to the noise subspace, which can be used to estimate M. The data can then be modeled using the coordinate system spanned by the M first eigenvectors, even though in some applications, such as data compression, fewer than M first eigenvectors would be used.

Linear techniques such as PCA do not work properly when the data manifold is nonlinear or non-Gaussian, however; in such cases, one usually needs to resort to iterative techniques such as the SOM. It seems intuitively reasonable to suggest that the SOM could be used to model any data manifold that is constructed by a smooth diffeomorphism from an M-dimensional unit cube into $\mathbb{R}^N$, given that the SOM has an M-dimensional lattice. The SOM unit coordinates on the map lattice would then give a discrete approximation of the natural coordinates (as defined in [3]) of the data manifold, although the SOM magnification factor [8] which is always less than one would introduce some distortion in this approximation.

When the dimensionality of the SOM lattice is lower than that of the data manifold, the situation becomes more problematic. The map tries to fill the data manifold by folding itself in a manner that is analogous to a one-dimensional Peano curve filling the two-dimensional unit square. This phenomenon has been dubbed "automatic selection of feature dimensions" by Kohonen [8], and has been rigorously analyzed by Ritter and Schulten [9]. As a result of folding, the resolution of the mapping from the data manifold onto the SOM lattice improves, but only at the cost of introducing discontinuities in the mapping [5].

In certain applications it would be useful to find only the few most important aspects of the data manifold, which can be regarded as the non-linear counterpart to finding the first few principal components. One possible way to achieve this is to control the stiffness of the map by choosing wide enough neighborhood function, so that the map will not fold significantly [4, 2, 5]. This way the discontinuities are avoided (or reduced) but some of the resolution is sacrificed, so again there is a tradeoff.

In principle, the preferred solution would thus be using a SOM that has the same dimensionality as the data manifold. On the other hand, increasing the dimensionality introduces certain drawbacks, as will be discussed later – at least visualization of maps with dimensionality higher than three is problematic. However, sometimes already increasing the dimensionality from two to three may improve the results significantly. This is demonstrated in the next section, both with toy data and real financial data.

3 Simulations with 2D and 3D SOMs

In the first example, toy data set is used to demonstrate how the folding of the SOM shows with two common visualization methods: the component plane display and the U-matrix. The data set consists of 5 000 points that are uniformly distributed inside a box that has side lengths 16, 9 and 4. Both two- and three-dimensional map ,lattices with equal number of units were trained using the SOM Toolbox [1] in batch mode and with a Gaussian neighborhood function.

The U-matrices and component planes of the maps are shown in figures 1 and 2. The four layers of the third dimension of the 3D-SOM are here displayed side by side, and, for ease of interpretation, the U-matrix has been calculated for each layer separately instead of also across the layers. In the middle of the map, the U-matrix of the 2D SOM is more uneven than that of the 3D SOM, and with the "bubble" neighborhood function, the difference would become still more distinctive. On the other hand, the stronger border effect of the 3D SOM is clearly visible.

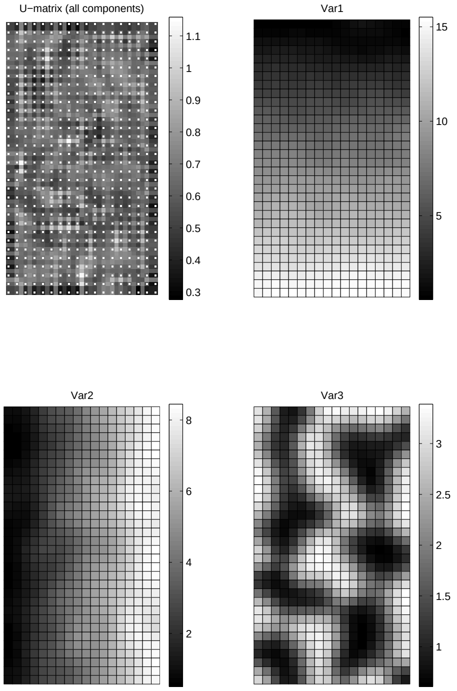

Figure 1: The U-matrix and component planes of the 2D SOM trained with toy data.

The first two component planes of the two maps look quite similar, but the third shows the difference: the 2D SOM folds into the data manifold forming periodic stripes or speckles, whereas the 3D SOM is able to capture the shape of the data manifold correctly. Taking a closer look, one may note smaller but otherwise similar patterns also on the second component plane. The folding may thus make simple data look rather complicated when visualized using a SOM of too low dimensionality!

In this toy example the input space is only three-dimensional, so it is possible to display the SOM unit weight vectors in the input space. This is done in figures 3 and 4. The folding of the lower-dimensional SOM is here easily visualized.

In the second example, financial statement data from small and medium-sized Finnish enterprises is visualized using again both 2D and 3D SOMs. The data

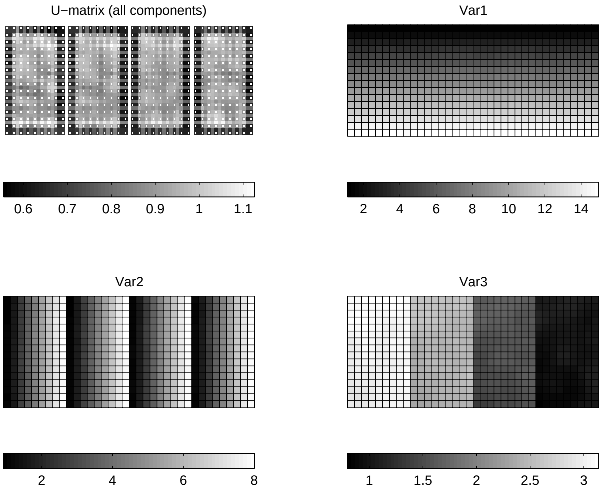

Figure 2: The U-matrix and component planes of the 3D SOM trained with toy data.

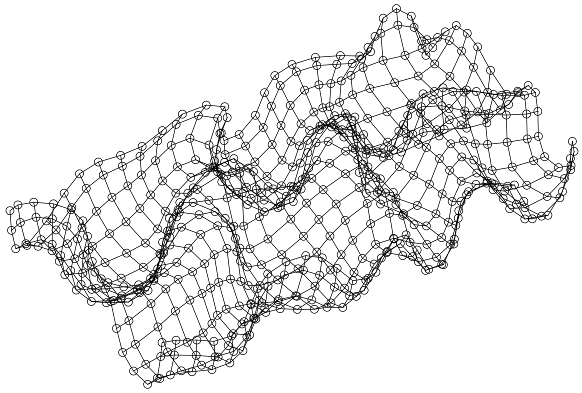

Figure 3: The weight vectors of the 2D SOM trained with toy data, shown in the input space.

set consists of 30 595 financial statements from 7 028 enterprises of which 1 244 eventually failed. Here we are mostly interested in the characterization of the differences between the financial statements of failing and non-failing companies, so we are trying to find those areas of the SOM where most bankruptcies occur. We use both single-year data that consists of seven different financial indicators and two-year data that consists of the coordinates of the enterprise on the single-year map during two consecutive years; the methodology used is described in more detail in [7].

The single-year maps are displayed in figures 5 and 6, and two-year maps in figures 7 and 8; the dark color corresponds to a high risk of bankruptcy. In both single- and two-year cases, the three-dimensional SOM shows much more clearly the boundary of the "bankruptcy area". In the single-year case, the 2D SOM shows a rather hazy boundary, whereas 3D SOM shows a single bankruptcy cluster with rather sharp boundaries. In the two-year case, the 2D SOM shows two separate bankruptcy clusters, which is most likely an artifact caused by the folding of the map: the 3D SOM still shows only a single cluster.

4 Discussion

The simulations suggest that in such cases when the intrinsic dimension of the data may be higher than the dimensionality of the SOM lattice, the results of visualizing the data must be interpreted with great caution. The folding of the map can make the "minor components" of the data look much more complicated than they actually are; for instance, it may be possible that one particular part of the input space is represented in several non-neighboring areas of the map.

These problems can sometimes be alleviated by using a higher-dimensional map. There is a price to pay, however. Adding extra dimensions quickly increases the map size, and although the SOM is rather insensitive to the number of map units (in a classification application, this is demonstrated in [6]), at least the computational cost increases linearly with the number of map units. Also the border effect becomes stronger with increasing the map dimensionality and should be countered either "explicitly" as proposed by Kohonen [8] or "implicitly" by increasing the map size. Finally, visualizing the SOM becomes tricky when the dimensionality of the map is more than three.

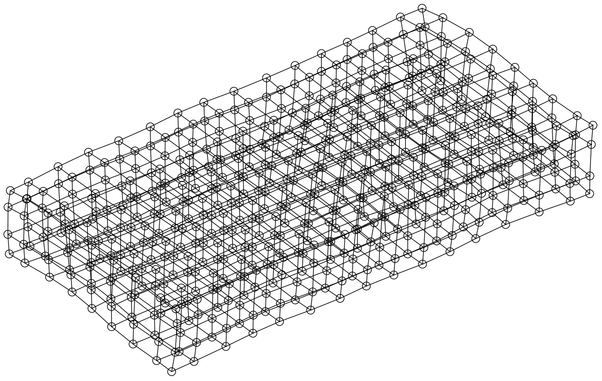

Figure 4: The weight vectors of the 3D SOM trained with toy data, shown in the input space.

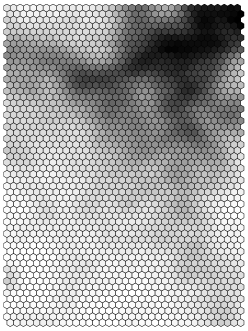

Figure 5: The 2D bankruptcy map, trained with the single-year financial statement data.

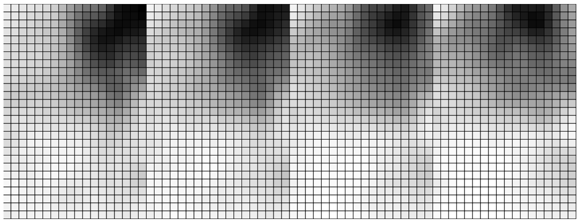

Figure 6: The 3D bankruptcy map, trained with the single-year financial statement data.

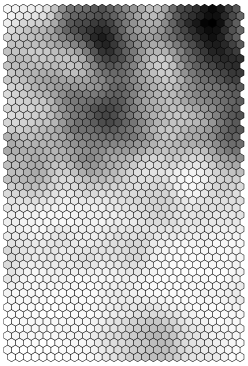

Figure 7: The 2D bankruptcy map, trained with the two-year trajectory data.

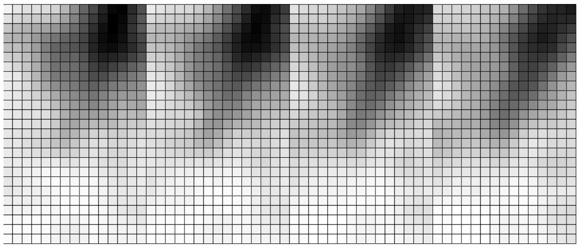

Figure 8: The 3D bankruptcy map, trained with the two-year trajectory data.

References

[1] E. Alhoniemi, J. Himberg, K. Kiviluoto, and J. Vesanto. SOM Toolbox. Published via WWW, Sept. 1997. Available at http://www.cis.hut.fi /projects/somtoolbox/.

[2] R. Der, G. Balzuweit, and M. Herrmann. Constructing principal manifolds in sparse data sets by self-organizing maps with self-regulating neighborhood widths. In *Proceedings of the International Conference on Neural Networks (ICNN'96)*, volume 1, pages 480–483, Piscataway, New Jersey, USA, June 1996. IEEE Neural Networks Council.

[3] R. Hecht-Nielsen. Replicator neural networks for universal optimal source coding. *Science*, 269(5232):1860–1863, Sept. 1995.

[4] M. Herrmann. Self-organizing feature maps with self-organizing neighborhood widths. In *Proceedings of the International Conference on Neural Networks (ICNN'95)*, volume 6, pages 2998–3003, Piscataway, New Jersey, USA, Nov. 1995. IEEE Neural Networks Council.

[5] K. Kiviluoto. Topology preservation in self-organizing maps. In *Proceedings of the International Conference on Neural Networks (ICNN'96)*, volume 1, pages 294–299, Piscataway, New Jersey, USA, June 1996. IEEE Neural Networks Council.

[6] K. Kiviluoto. Predicting bankruptcies with the self-organizing map. *Neurocomputing*, 1998. To appear.

[7] K. Kiviluoto and P. Bergius. Two-level self-organizing maps for analysis of financial statements. In *Proceedings of the 1998 IEEE International Joint Conference on Neural Networks (IJCNN'98)*, volume 1, pages 189–192, Piscataway, New Jersey, USA, May 1998. IEEE Neural Networks Council.

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

[9] H. Ritter and K. Schulten. Convergence properties of Kohonen's topology conserving maps: Fluctuations, stability, and dimension selection. *Biological Cybernetics*, 60(1):59–71, Nov. 1988.