Publication 5

Hunting for Correlations in Data Using the Self−Organizing Map

Juha Vesanto and Jussi Ahola

In *Proceeding of the International ICSC Congress on Computational Intelligence Methods and Applications (CIMA'99)*, ICSC
Academic Press, pp. 279−285, 1999.

Hunting for Correlations in Data Using the Self-Organizing Map

Juha Vesanto and Jussi Ahola
Helsinki University of Technology
Laboratory of Computer and Information Science
P.O. Box 5400, FIN-02015 HUT, Finland
email: {Juha.Vesanto, Jussi.Ahola}@hut.fi

Abstract

The Self-Organizing Map (SOM) is an efficient tool for visualization of multidimensional numerical data. One of the tasks it is used for is correlation hunting. In this paper we present a simple method to enhance correlation hunting in the case of a large number of variables. Different variations of the method - component plane reorganization - are evaluated on a complex test data. The purpose is to somewhat validate the use of SOM in correlation hunting and to evaluate the strengths and weaknesses of different reorganization procedures. A case with a real world data is also presented to show the usefulness of the method.

1 Introduction

Data mining is an emerging area of new research efforts, responding to the presence of large databases in commerce, industry, and research. It is also a title for a large number of widely divergent methods ranging from relational learning to neural networks and is, as such, part of a larger framework, Knowledge Discovery in Databases (KDD) [3]. Data mining is an interactive process requiring that the intuition and background knowledge of humans are coupled with the computational efficiency of modern computer technology. For this reason visualization is a very important part of data mining.

The Self-Organizing Map (SOM) [6] is a neural network algorithm based on unsupervised learning. The SOM implements an ordered dimensionality-reducing mapping of the training data. It has several beneficial features which make it a useful method in data mining. The map follows the probability density function of the data, is readily explainable, simple and - perhaps most importantly - easy to visualize. The SOM has proven to be a valuable tool in data mining and KDD [2, 5, 8].

In this paper one aspect of the use of SOM in data mining is inspected: correlation hunting. The term "correlation" does not encompass only linear correlations, but also nonlinear and local or partial correlations between variables. The term "hunting" is used because visualization is used as the primary tool, and the final assessment of a connection between variables is done by the human rather than by the algorithm.

Correlations are hunted from the component planes visualization of the SOM: similar patterns in identical positions indicate correlation between the respective components (variables). This kind of pattern matching is something that the human eye is very good at. However, when the number of variables is large (a few dozen or more), the pattern matching becomes a rather tedious process. To aid it, the component planes can be organized so that possibly correlated components are close to each other. In the subsequent sections several variations of this basic scheme are considered and tested on a complex test data set. A case of a real-world data set is also presented to demonstrate the usefulness of the method.

2 Methods

2.1 Self-Organizing Map

The SOM is formed of neurons located on a regular low-dimensional grid (usually 1D or 2D). Higher dimensional grids are also possible, but they are not generally used since their visualization is problematic. The lattice of the grid can be either hexagonal or rectangular. Each neuron i of the SOM is an n-dimensional prototype vector $\mathbf{m}_i = [m_{i1}, \ldots, m_{in}]$, where n is equal to the dimension of the input space. On each training step, a data sample $\mathbf{x}$ is selected and the unit $\mathbf{m}_c$ closest to it (the best-matching unit, BMU) is located from the map. The prototype vectors of the BMU and its neighbors on the grid are moved towards the sample vector:

$$\mathbf{m}_i = \mathbf{m}_i + \alpha(t)h_{ci}(t)(\mathbf{x} - \mathbf{m}_i),$$

where $\alpha(t)$ is learning rate and $h_{ci}(t)$ is a neighborhood kernel centered on the winner unit c. Both learning rate and neighborhood kernel radius decrease monotonically with time. During the iterative training, the SOM behaves like a flexible net that folds onto the "cloud" formed by input data.

2.2 Component planes

The fact that the prototype vectors are organized on a low-dimensional grid makes their visual inspection easy. One way of doing this is component plane representaion. Each component plane of the SOM consists of the values of the same component in each prototype vector. Thus, they can be thought of as a "sliced" version of the map. The component planes are typically visualized by giving each map neuron a color according to the relative value of the respective component in that neuron. By plotting all component planes and comparing them with each other, correlations between variables can be seen.

The order of the component planes is usually that of the order of the variables in the vector. However, when the number of components is large, the sheer number of component planes makes it difficult to discern which planes resemble each other (Figure 4(a)). The task can be made easier if the component planes are reorganized so that the correlating ones are presented near each other (Figure 4(b)). To do this, the component planes, or some representation of them, is projected on a plane. The projection could be done using, e.g., Sammon's mapping [7] or another SOM.

If the SOM is used, the projection of each component plane is the location of its BMU. To prevent several component planes from being assigned to the same place a simple procedure is applied: for any unit with multiple component planes, the worst matching of them is moved to its next-best-matching unit. This is repeated until no two component planes are in the same location. The use of a SOM for the componet plane projection gives an important benefit thatw the placements of the component planes can be shown on a regular grid. However, as these placements are limited due to the grid, they may not be as "correct" as in Sammon's mapping or PCA. One of the goals of this paper is to investigate whether this is a serious problem.

The choice of the component plane representation allows several variations to this basic procedure. There are two possibilities for the initial representation of the component planes:

- the component plane itself is transformed into a vector and normalized to unit length to ignore different scalings of components (CP)
- the average difference in component values (distance) of each unit with respect to its neighbors is calculated and processed as above (DTN). The motivation of this is that large cooccuring changes (positive or negative) in two components imply that they are connected.

The vectors thus gained can either be used as such, or they can be further processed by calculating the covariance matrix of the representation vectors, taking the absolute value, and using these as data ($|\mathrm{cov(CP)}|$, $|\mathrm{cov(DTN)}|$). By taking the absolute value components having strong inverse correlation are also projected near each other, which is desirable.

3 Experiments

The variations of the proposed method obviously highlight different features and therefore produce different results. The purpose of the experiments was to compare them with each other and to determine their sensitivity to noise in the input data.

The data set used in the tests contained 1000 data vectors with 17 components. The first three components (x, y, z) were independent variables with uniform distribution between $[0, 1]$. The other variables were different functions of one, two, or all of them. In some variables the first 500 samples had a different dependence than the last 500. The dependent variables are presented in Figure 1. Some noise was also added to the data (signal to noise ratio (SNR) of ∞, 10, 2 and 1).

Three map sizes were used: $6 \times 4, 18 \times 12$, and 36×24 units. Two different representations of the component planes was used: $|\mathrm{cov(CP)}|$ and $|\mathrm{cov(DTN)}|$, introduced inthe previous section. These representations were then projected on a plane using three projection algorithms: Principal Component Analysis (PCA) [1], Sammon's mapping and the SOM.

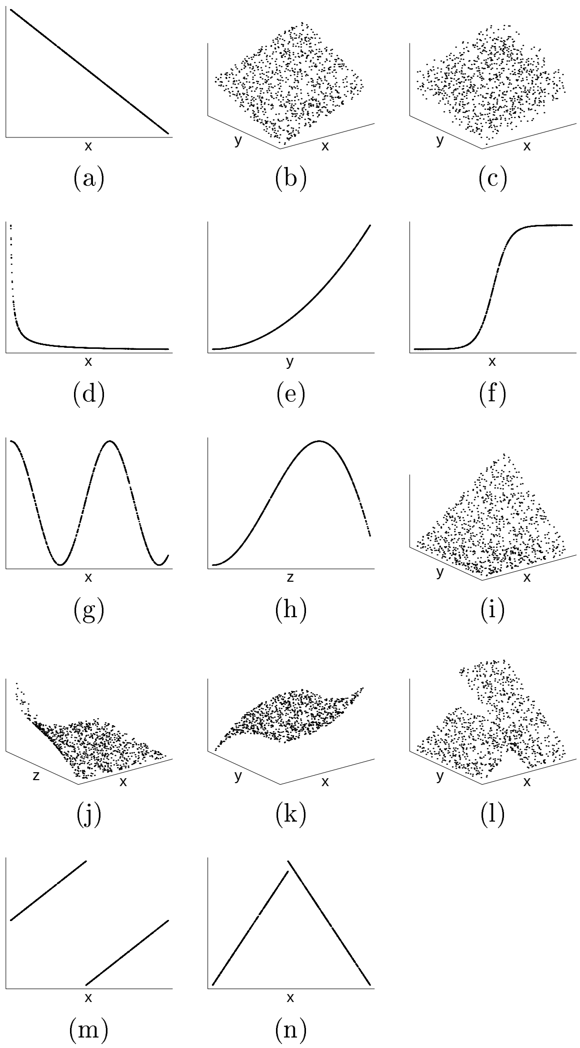


Figure 1: The dependent variables as a function of the primary variables (x, y, z): $-2x$ (a), $x + y$ (b), $x+y+z$ (as a function of x and y only)(c), $(x+0.01)^{-1}$ (d), $4y^2$ (e), $tanh(10x-5)$ (f), $cos(10x)$ (g), $zsin(3x)$ (h), xy (i), $z/(x + 0.1)$ (j), $(x - y)^3$ (k), $[x; y]$ (l), $[x; x - 1]$ (m), $[x; 1 - x]$ (n).

For reference also a random placement of components and placement using values of the component planes as such (CP) for the representation were considered.

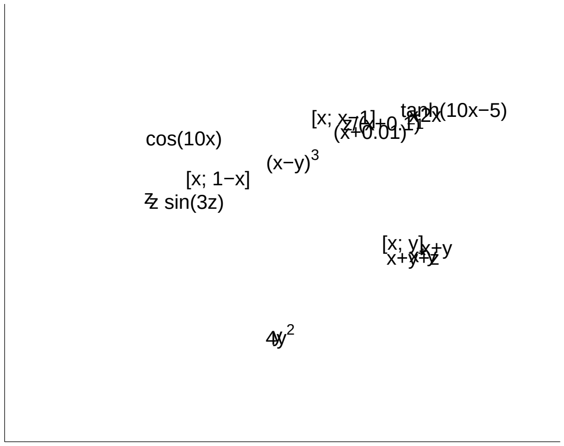
Reaching the objective of positioning components with a connection near each other was tested with a heuristic test: for each of the 14 derived variables, the distance between the placement of the dependent variable and each of the three primary variables was calculated. Based on relations between the three distances it was determined whether the placement was "correct", "acceptable" or "wrong" (ranking values 2, 1 and 0 respectively). In a "correct" placement the distance to the variables of which the component depends on must be smaller (by a certain amount) than to the other primary variables. In a "wrong" placement, the distances are in opposite order. For example, in Figure 2 (c) in the upper left corner variable $4y^2$ clearly is much closer to y than x or z, so its ranking would be "correct". However, even though the variable $[x; x - 1]$ in the lower right corner is closer to x than y or z, the fact that the distance between the variable and x is not small enough compared to distance between the variable and z gives it ranking "acceptable". On the other hand, since the variable $cos(10x)$ up in the middle of the figure is closer to z and y than x, its ranking would be "poor". The ranking values were averaged over ten test runs. This test shows us how the methods could indicate dependencies between variables. Note, however, that the test results should not be taken as the absolute truth. The test method is very heuristic, and furthermore using distances as the indicator of correctness is not exactly proper since our primary interest is on visual impression.

4 Results

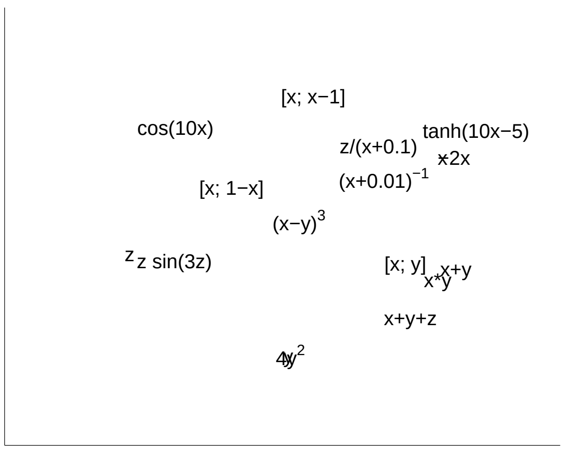
The average rankings using different approaches are shown in Figure 3. When calculating the effect of a certain parameter, the rankings were averaged over all other parameters (e.g. when the effect of the map size was studied, the results were averaged over noise, representation type and projection algorithm). Furthermore, random placement and CP representation were excluded in the averaging, since were only included in the tests for reference. As seen from Figure 3(d), the average for random positioning is about 0.4. Thus, results higher than 0.6 can be considered adequate, and results over 1 good.

From Figure 3(a) it can be seen that while noise degraded the results, the average rankings were still pretty good for a number of variables even with SNR of 1.

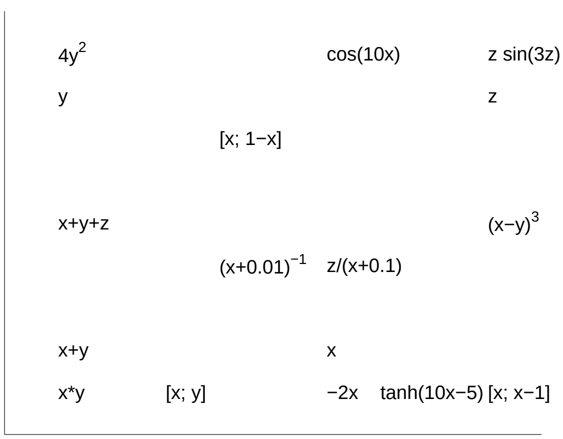
The effects of different map sizes can be seen from Figure 3(b). The results show two tendencies: some variables fare best with smaller maps but the rest, especially non-monotonic functions $[x, 1 - x]$ and $cos(10x)$ have best results with the middle sized map. Presumably the small map averages too much, and the biggest map is effected too much by the noise. Corresponding tendencies can be seen from Figure 3(c): for the non-monotonic variables $|\mathrm{COV}(\mathrm{CP})|$



(a) PCA



(b) Sammon



(c) SOM

Figure 2: Examples of typical projection results for different projection algorithms. In PCA and Sammon's mapping the variables can move freely which may cause overlapping. The SOM, on the other hand, enables the direct visualization of the component planes themselves, since they will not overlap.

fails, but $|\mathrm{COV}(\mathrm{DTN})|$ still works.

The choice of the projection algorithm (SOM, PCA or Sammon) had very little importance (Figure 3(d)). This is nice because it shows that the SOM can be used as the second-level projection algorithm. This is convenient because of the straightforward visualization of the SOM.

5 Discussion

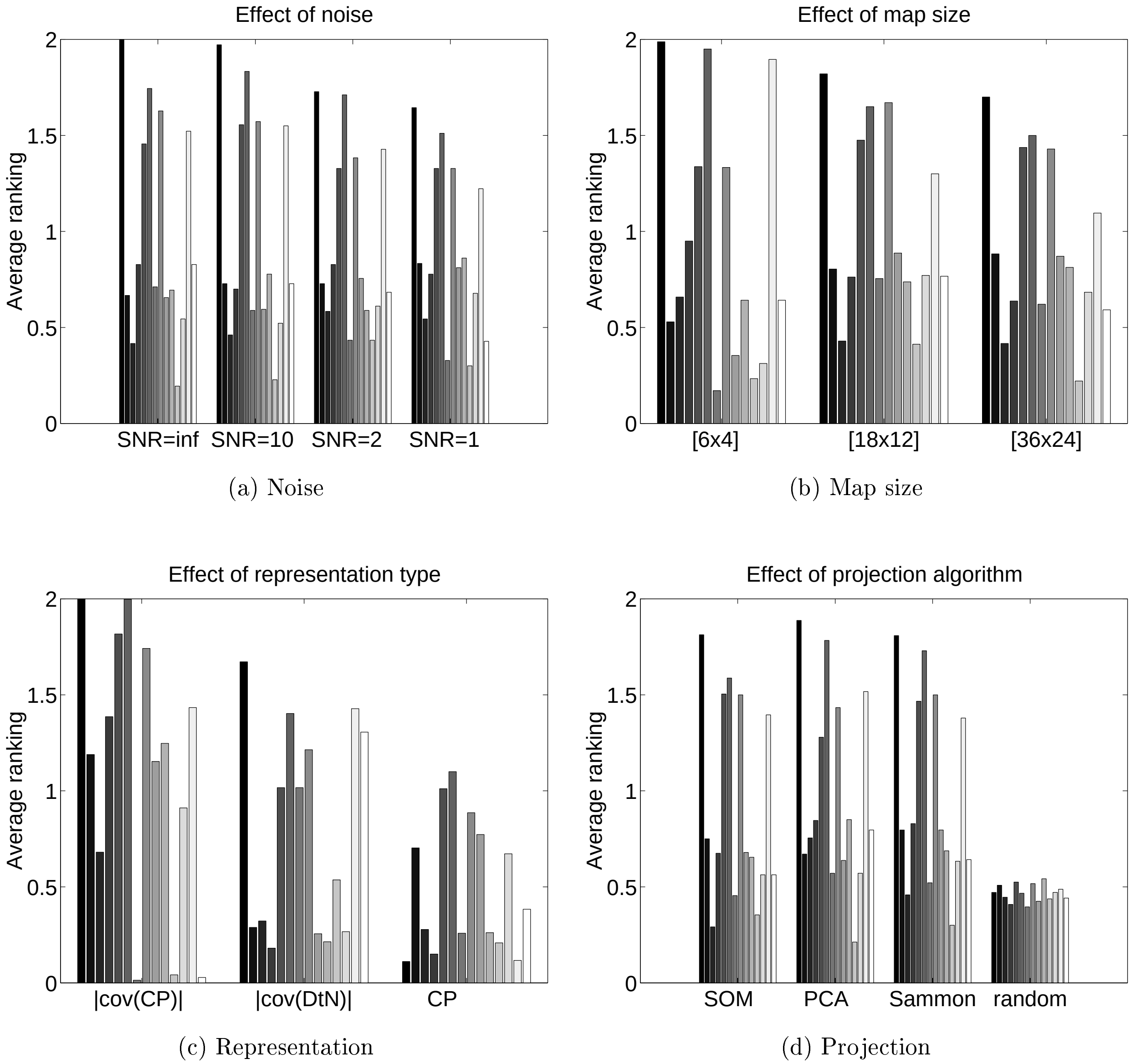
When compared with a simple correlation analysis by calculating the covariance matrix from the data (results not shown in this paper), the proposed method had both advantages and disadvantages. In some cases, when the covariance matrix correctly indicated dependency between variables, the proposed method fared poorly. However, the covariance matrix is much more sensitive to noise, and fares poorly in non-monotonic cases, e.g., $cos(10x)$. Therefore, the proposed method should be used in addition to normal correlation analysis.

Visual examination of the maps brought up some issues that our heuristic evaluation method missed. One such is that we ignored correlations between derived variables: for example variables $(x + 0.01)^{-1}$ and $z/(x+0.1)$ seemed to be very close to each other in the projections.

Another way of looking for dependencies or correlations from the data would be to use scatter plot for each pair of variables in a "Grand Tour". Since, in the proposed method each variable need only be visualized once, the number of plots is reduced from $n(n-1)/2$ to n. From this kind of visualization it is easy to select interesting component combinations to be investigated further. Furthermore, using color to link the map and scatter plots of component pairs together makes it possible to identify partial correlations with clusters on the map [4].

6 Case: Hot strip mill data

The proposed method was also tested for a real world data. The data set used was collected from Rautaruukki's hot strip mill in Raahe. It included information about chemical content, casting machines, re-heating furnaces, process conditions, and end product quality of about 4000 hot rolled micro-alloyed strips. The data was analysed in order to find out dependencies between the variables.



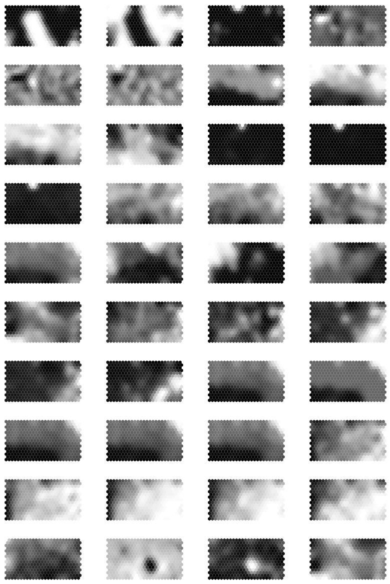
(a) Noise

(b) Map size

(c) Representation

(d) Projection

Figure 3: The results of the experiments. Each bar chart shows the results averaged over all other parameters, except the one under observation. Each bar shows the average result for one variable (the order is the same as in Figure 1). Note that average ranking of about 0.4 corresponds to the result of random positioning, as seen from bottom right figure. Results higher than 0.6 can be considered adequate, and results over 1 good. For example, in the upper left figure in the first group of bars (where there was no noise in signals) the leftmost (black) bar represents variable $-2x$ averaged over map size, representation type (excluding CP) and projection algorithm (excluding random placement). It can be seen that its ranking value is about 2, so the placement of the variable was good, regardless of the other parameters used. On the other hand, in the same group the third (light gray) bar from the left, representing variable $[x; y]$, shows ranking value of only about 0.2. So, the placement of the variable was poor throughout.



(a)

(b)

Figure 4: Correlations between components can be searched from the component planes visualization in Figure (a). The task is easier if the planes are reorganized so that component planes which seem to be correlated are placed near each other, as in Figure (b).

In the study component plane reorganization was used together with traditional correlation analysis. Both of the representation types ($|\mathrm{cov}(\mathrm{CP})|$, $|\mathrm{cov}(\mathrm{DtN})|$) were used. Only the SOM was used as a second level projection method because of its quick and easy visualization. In Figure 4 the component planes are reorganized illustrating the benefit of using the method.

As a result several global linear correlations, most of which were already known or otherwise uninteresting, were found. However, some of them, in addition to a few local correlations turned out to have some valuable information. No nonlinear correlations could be found, but that was probably more due to the nature of the data than incapability of the methods.

Summa summarum, component plane reorganization proved to be a very useful method in finding both global and local correlations. Its biggest contribution for the analysis was the detection of the local correlations since traditional correlation analysis could not find them. Furthermore, in this case they were the most interesting ones, since in further analysis they revealed some detailed information about, e.g., the problems with different products under different process conditions.

Acknowledgments

This work has been carried out within EU financed Brite/Euram project "Application of Neural Network Based Models for Optimisation of the Rolling Process" (NEUROLL). Especially, Rautaruukki is gratefully acknowledged for providing the authors with the process data.

References

[1] C. M. Bishop. *Neural Networks for Pattern recognition.* Oxford University Press, 1995.

[2] G. Deboeck and T. Kohonen. *Visual explorations in Finance using Self-Organizing Maps.* Springer-Verlag, London, 1998.

[3] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors. *Advances in Knowledge Discovery and Data Mining.* AAAI Press / The MIT Press, California, 1996.

[4] J. Himberg. Enhancing the som based data visualization by linking different data projections. In

Proceedings of International Symposium on Intelligent Data Engineering and Learning (IDEAL), Hong Kong, October 1998. to appear.

[5] S. Kaski. *Data Exploration Using Self-Organizing Maps*. PhD thesis, Helsinki University of Technology, 1997.

[6] T. Kohonen. *Self-Organizing Maps*. Springer, Berlin, Heidelberg, 1995.

[7] J. W. Sammon, Jr. A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, C-18(5):401–409, May 1969.

[8] J. Vesanto. Data mining techniques based on the self-organizing map. Master's thesis, Helsinki University of Technology, June 1997. http://www.cis.hut.fi/~juuso/dippa/.