Abstract

The most important aspects for creating good SOM maps are the selection of the size and shape of the SOM, the scaling of the input variables, the selection of the neighborhood function and the learning rate, and the initialization of the model vectors. In addition, we discuss automatic procedures for color-coding of SOM maps. The data entries corresponding to, for example, different countries can be automatically colored so that similar data attain a similar color.

14.1 The SOM Array

In order to make a good visual display, the arrangement of the neurons in the array ought to be hexagonal. The array should be oblong, because the distribution of the data samples is usually oblong, and the net of the model vectors should be fitted to it. Square arrays usually do not orient themselves well enough with the distribution.

The number of neurons in the array is a rather delicate figure that depends on the application and the amount of available data. For instance, if you have only one set of indicators per entry then the number of neurons might be selected as somewhat smaller than the number of entries. If, on the other hand, you have sample vectors that are stochastic variables that fall into more or less fuzzy clusters and you are interested in showing the cluster structure on the SOM, then the number of neurons would be better as a multiple of the number of clusters.

Sometimes the number of available samples is limited due to the high costs or difficulties in acquiring them. In such a case the statistical accuracy of representation may become a problem; you cannot fit many nodes to a single sample. If you are studying clustering by the SOM, the number of neurons should not be higher than, say, a fraction of the order of 10% of the total number of samples. So there are many different cases, and it is essential to understand the nature of the problem and the data before trying to display it with SOM.
14.2 Scaling the Input Variables

In some cases it may perhaps be surprising that you have different indicators with different weights, but nonetheless you may have to equalize the scales of the different indicators. Equal scaling, scaling of the variance of each indicator to unity, is in general a good first choice in preliminary clustering tests, and effective even in the final maps. However, sometimes, especially when concentrating on particular indicators, it may turn out to be useful to try different scales for various indicators. Variable scaling is closely connected with the selection of the most important input variables, as mentioned elsewhere in this book.

14.3 Initialization of the Algorithm

A very special property of the SOM algorithm is that the initial values for the model vectors can be selected as arbitrary random vectors. The algorithm is able to scale them properly and sort them into a smooth order in the long run. Selection of random initial values, however, is not at all the most reasonable or effective way to proceed in practice. The self-organizing process works orders of magnitude faster, and the final results are much more stable, if you set a preliminary rough order for the model vector values before starting the algorithm. For instance, the SOM_PAK software has a preprogrammed provision for automatic determination of the two largest principal components of the data samples, upon which a regular grid of points can be spanned along the hyperplane defined by the sample data. These values should be taken for the initial values of the model vector grid; this allows much narrower neighborhood functions and a smaller learning-rate factor.

14.4 Selection of the Neighborhood Function and Learning Rate

In extensive tests it has turned out that the SOM algorithm tolerates very different choices for the neighborhood function and the learning-rate factor. However, what seems to be most essential is that

- the neighborhood function should be wider in the beginning of the learning process, and the width should decrease with time so that at the end of the process only the immediate neighbors of the winner are updated;

- there should be enough training steps in the self-organizing process.

While the Gaussian neighborhood function introduced in Chapter 11 may be a little difficult to use, it is quite possible to resort to the following simpler choice for neighborhood function that also works fairly well, especially if the model vectors are initialized as recommended above.

The simpler SOM equations are

\[ m_i(t+1) = m_i(t) + \alpha(t) \left[ x(t) - m_i(t) \right] \text{ if } i \text{ belongs to } N_c(t), \]

\[ m_i(t+1) = m_i(t) \text{ if } i \text{ does not belong to } N_c(t). \]