

Data Visualization Using the DPSOM

Chao Shao, Houkuan Huang

School of Computer and Information Technology, Beijing Jiaotong University, Beijing, 100044, China

Abstract

Due to the topology preserving nature, the Self-Organizing Map (SOM) is effective for visualizing the high-dimensional data. However, for visualization the SOM requires some color-coding scheme such as U-matrix to imprint the distances between neighboring neurons on the regular lattice. Even so, the distribution and structure of the data may often appear in a distorted and unnatural form. In order for the map to visualize the data faithfully and naturally, the dissimilarity or even the distance quantity should be preserved on the map, along with the topology. To do this, the positions of the neurons on the map should be adjustable. This paper proposes a novel variant of SOM—Distance-Preserving SOM (DPSOM), which can adaptively adjust the positions of the neurons on the map according to the corresponding distances in the data space. What's the most important, our DPSOM algorithm can automatically avoid the excess contraction of the neurons to one point without any additional parameter, which makes our DPSOM algorithm more advantageous over those position-adjustable SOMs such as the Adaptive Coordinate (AC) algorithm, the Double SOM (DSOM) algorithm, and the Position-Adjustable SOM (PASOM) algorithm. Finally, our algorithm is verified by experimental results very well.

Keywords: Self-Organizing Map (SOM), Multidimensional Scaling (MDS), Himberg's contraction model, the stochastic gradient descent method, position-adjustable SOM, Position-Adjustable SOM (PASOM), Distance-Preserving SOM (DPSOM)

1. Introduction

The Self-Organizing Map (SOM), which was first introduced in 1982 by Kohonen[1], is a singular-layer neural network model based on competitive learning. As the combination of vector quantization and non-linear projection[2], the SOM can map the high-dimensional data onto a low-dimensional regular lattice, i.e. a fixed grid of neurons, while preserving the topological relations of the data as faithfully as possible, which makes it a popular clustering, visualization and abstraction tool.

Due to the topology preserving nature, the SOM is effective for visualizing the high-dimensional data. However, due to the fixed grid of neurons, the inter-neuron distances are visible indirectly, which makes it necessary to use some color-coding scheme such as the U-matrix[2] to imprint the distances between neighboring neurons on the regular lattice. Even so, the distribution and structure of the data may often appear in a distorted and unnatural form[3].

In order for the map to visualize the distribution and structure of the data faithfully and naturally, the dissimilarity or even the distance quantity should be preserved on the map, along with the topology[3]. To do this, the positions of the neurons on the map should be adjustable during or after the learning process. Taking the concision of the model and the amount of computing time into consideration, we prefer adjusting the positions of the neurons synchronously during the learning process. We name such variants of SOM as position-adjustable SOMs[4], such as the Grouping Neuron (GN) algorithm[5], the Adaptive Coordinate (AC) algorithm[6], the Double SOM (DSOM) algorithm[7], and the Position-Adjustable SOM (PASOM) algorithm[4]. Being add-ins to the standard Kohonen's SOM, the robustness of these methods is assured[6]. In addition, the adjustment rule is relatively simple. However, these methods fall into the category of Himberg's contraction model[8][4], in which the neurons are contracted according to their similarities, and which is inevitably confronted with the problem of the excess contraction to one point[4]. To avoid this problem, these models have to use variant parameters, which are quite difficult to control. In this paper, we propose a novel variant of SOM—Distance-Preserving SOM (DPSOM), which can adaptively adjust the positions of the neurons on the map according to the corresponding distances in the data space. What's the most important, our algorithm can automatically avoid the excess contraction of the neurons to one point without any additional parameter, which makes our DPSOM algorithm more advantageous over those position-adjustable SOMs.

This paper is organized as follows: In Section 2, we recall the standard Kohonen's SOM. In Section 3, we propose our DPSOM algorithm. Finally, experimental results, the analysis and conclusions are given in Section 4, 5 and 6 respectively.

2. The Standard Kohonen's SOM

The standard Kohonen's SOM consists of a singular-layer of neurons located on a low-dimensional regular lattice. Each neuron k is represented by an n -dimensional weight vector $w_k = \{w_{k1}, \dots, w_{kn}\}$, where n is the dimensionality of the data. On each learning step, a data sample x is selected randomly, and then the winner neuron (BMU, the best-matching unit) v is found according to the following rule:

$$\|w_v - x\| = \min_k \|w_k - x\| \quad (1)$$

After that, a neighborhood learning is adopted, i.e. the weight vectors of the BMU and its neighbors defined by a neighborhood kernel are updated according to the following rule:

$$w_k(t+1) = w_k(t) + \eta(t)h_{vk}(t)(x - w_k(t)) \quad (2)$$

Where $\eta(t)$ is the learning rate and $h_{vk}(t)$ is the neighborhood kernel centered on the BMU v which usually takes the following form:

$$h_{vk}(t) = e^{-\frac{\|v-k\|^2}{2\sigma(t)^2}} \quad (3)$$

Where $\sigma(t)$ represents the neighborhood kernel radius. To assure the convergence, both the learning rate $\eta(t)$ and the neighborhood kernel radius $\sigma(t)$ decrease monotonically with time.

Because not only the BMU but also its neighbors are updated in the same direction, the weight vectors of neighboring neurons resemble each other[2]. Consequently, the BMUs of similar data samples are close to each other within the lattice, which is so-called SOM's topological ordering or topological preserving. For the sake of visual comparison and easy implementation, the neurons are fixed onto a regular lattice. However, the inter-neuron distances are indirectly visible, and thus the distribution and structure of the data may often be distorted and unnatural.

3. The DPSOM Algorithm

In order for the map to visualize the data faithfully and naturally, the dissimilarity or even the distance quantity should be preserved on the map, along with the topology[3]. Therefore, we should combine the distance-preserving mapping such as Multidimensional Scaling (MDS) with the SOM so that the distance quantity can be preserved on the map.

MDS is a point-to-point mapping, which maps the high-dimensional data into a low-dimensional space, while preserving the inter-point distances as close as possible. Thus, the objective function of MDS can be defined as follows:

$$E = \sum_i \sum_{j < i} E_{ij} = \frac{1}{2} \sum_i \sum_{j < i} (d_{ij}^* - d_{ij})^2 \quad (4)$$

Where d_{ij}^* is the distance between the i -th and j -th data points in the data space, and d_{ij} is the corresponding distance in the low-dimensional space. We can minimize E by variant methods such as the steepest gradient descent method and the stochastic gradient descent method.

In the SOM, we represent the position vector of neuron i on the map by $X_i = \{x_{i1}, \dots, x_{im}\}$, then we

have $d_{ij} = \sqrt{\sum_{k=1}^m (x_{ik} - x_{jk})^2}$. On each learning step,

after the BMU i related to the given data sample x is found, we can decrease the value of the objective function related to the BMU i , i.e. $E_i = \sum_{j \neq i} E_{ij}$, by

adjusting the positions of all the other neurons on the map according to the stochastic gradient descent method:

$$\begin{aligned} x_{jk} &= x_{jk} - \alpha(t) \frac{\partial E_i}{\partial x_{jk}} \\ &= x_{jk} - \alpha(t) \frac{\partial E_{ij}}{\partial x_{jk}} \\ &= x_{jk} - \alpha(t)(d_{ij} - d_{ij}^*) \frac{x_{jk} - x_{ik}}{d_{ij}}, \quad \forall j \neq i \end{aligned} \quad (5)$$

Where $\alpha(t)$ is the learning rate which decreases monotonically with time. We can also rewrite Equation (5) as follows:

$$x_{jk} = x_{jk} + \alpha(t) \frac{(d_{ij} - d_{ij}^*)}{d_{ij}} (x_{ik} - x_{jk}), \quad \forall j \neq i \quad (6)$$

Equation (6) means that X_j is adjusted based on the relative difference between d_{ij} and d_{ij}^* . Note that Equation (6) is similar to the adjustment rules of those position-adjustable SOMs above, and $\frac{(d_{ij} - d_{ij}^*)}{d_{ij}}$ is

the counterpart of the similarity measures in Himberg's contraction model and those position-adjustable SOMs. However, unlike the similarity measures, $\frac{(d_{ij} - d_{ij}^*)}{d_{ij}}$ is not always positive,

sometimes negative, which means that the neurons are not always contracted, and which can automatically avoid the problem of the excess contraction of the neurons to one point.

So, our DPSOM algorithm can be described as follows: on each learning step, a data sample x is selected randomly, then the BMU v is found according to Equation (1), and then the positions of all the neurons but the BMU v are adjusted according to Equation (6) after the weight vectors of the BMU v and its neighbors are updated according to Equation (2). In the DPSOM algorithm, we can specify that $\alpha(t) = \eta(t)$ simply.

We can see that our DPSOM algorithm is also a type of position-adjustable SOMs, so it has the advantages of other position-adjustable SOMs too, such as the robustness and the simplicity[4].

4. Experimental Results

In this section, we will apply the standard Kohonen's SOM, PASOM and DPSOM algorithms on the following two datasets respectively:

The **butterfly** dataset: a 2-D dataset, which has 53 data points (seen in Fig. 1).

The **IRIS** dataset: a well-known four-dimensional dataset, which has 150 data points and is divided into three groups equally and two of them are overlapping.

In the experiments, we specify that the neighborhood kernel is a Gauss function whose form is given in Equation (3), and the parameters are defined as follows: the number of neurons $N = rows \times cols$, where $rows$ and $cols$ represent the number of rows and columns of neurons respectively, the values of which are specified in the brackets; $\eta(t) = 0.9 \cdot e^{-\frac{t}{100}}$; $\sigma(t) = \max(rows, cols) \cdot e^{-\frac{t}{100}}$, where $\max(rows, cols)$ is the larger one between the number of rows and columns of neurons, i.e. $rows$ and $cols$; the maximal number of iterations is 2000, and the learning process stops when the square sum of the difference of the weight vectors between two successive steps is smaller than a threshold $\varepsilon = 0.00001$ and the BMU of each data point is stable. The results of these three algorithms on the butterfly dataset are given in Fig. 2-6, and the results of these three algorithms on the IRIS dataset are given in Fig. 7-12. In addition, in the PASOM algorithm, besides the number of neurons, there are another two parameters, i.e. cf and $threshold$ (seen in [4]), which are specified in the brackets too.

5. Analysis of Experimental Results

From these results, we can see that the data can be visualized more faithfully and naturally by the DPSOM algorithm (seen in Fig. 5-6 on the butterfly dataset and Fig. 10-12 on the IRIS dataset respectively)

than the standard Kohonen's SOM (seen in Fig. 2 on the butterfly dataset and Fig. 7 on the IRIS dataset respectively). The reason is that our DPSOM algorithm can also dynamically optimize the objective function (3) of MDS during the learning process, and thus preserve the inter-neuron distances on the map as close as possible.

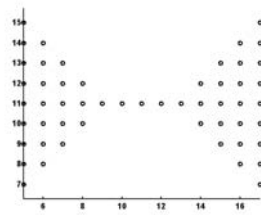


Fig. 1: The butterfly dataset

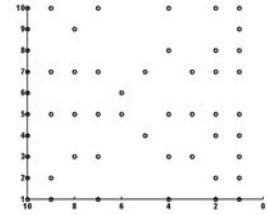


Fig. 2: The standard Kohonen's SOM (10x10)

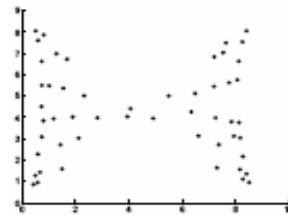


Fig. 3: PASOM (10x10, $cf=0.10$, $threshold=200[4]$)

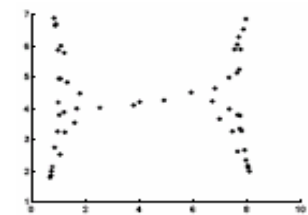


Fig. 4: PASOM (10x10, $cf=0.25$, $threshold=200[4]$)

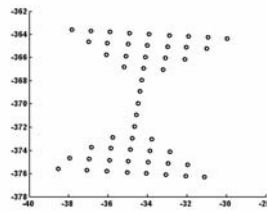


Fig. 5: DPSOM (10x10)

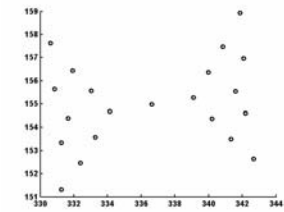


Fig. 6: DPSOM (5x5)

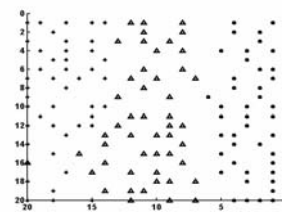


Fig. 7: The standard Kohonen's SOM (20x20)

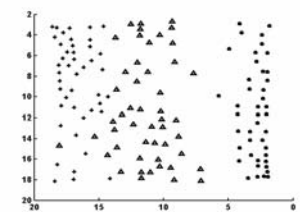


Fig. 8: PASOM (20x20, $cf=0.01$, $threshold=150$)

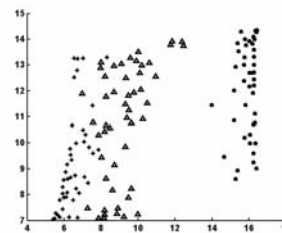


Fig. 9: PASOM (20x20, $cf=0.05$, $threshold=150$)

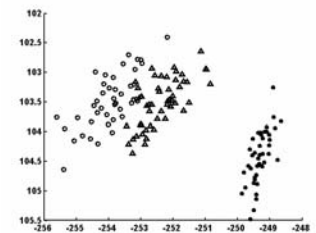


Fig. 10: DPSOM (20x20)

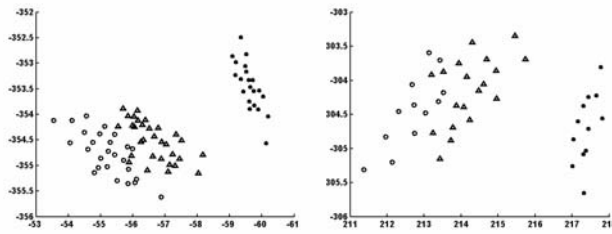


Fig. 11: DPSOM (10x10)

Fig. 12: DPSOM (7x7)

In addition, unlike the PASOM algorithm which need proper additional parameters, i.e. cf and $threshold$ [4], to obtain relatively good results (seen in Fig. 3-4 on the butterfly dataset and Fig. 8-9 on the IRIS dataset respectively), the data can be visualized very well by the DPSOM algorithm without any additional parameters (seen in Fig. 5-6 on the butterfly dataset and Fig. 10-12 on the IRIS dataset respectively). The reason is that our DPSOM algorithm precisely adjusts the positions of the neurons based on the corresponding distances in the data space in the way similar to MDS, however, the PASOM algorithm uses Himberg's contraction model which contracts the neurons to the weighted center of the neurons gradually, and thus needs proper additional parameters to avoid the excess contraction of the neurons to one point, and these additional parameters influence the quality of the result maps greatly. Therefore, the DPSOM algorithm has better controllability than the PASOM algorithm. In addition, for the same reason, the results of the DPSOM algorithm are more precise than those of the PASOM algorithm.

6. Conclusions

In this paper, we propose a novel variant of SOM--Distance-Preserving SOM (DPSOM), which can adaptively adjust the positions of the neurons on the map according to the corresponding distances in the data space in the way similar to MDS, so the distance quantity can be preserved on the map very well. In addition, as a type of position-adjustable SOMs, our DPSOM algorithm has the advantages of other position-adjustable SOMs too, such as the robustness and the simplicity. What's the most important, our DPSOM algorithm can automatically avoid the problem of the excess contraction of the neurons to one point without any additional parameter that other position-adjustable SOMs need, which makes our DPSOM algorithm much more advantageous over other position-adjustable SOMs, that is, our DPSOM algorithm has better controllability than other position-adjustable SOMs. Finally, our DPSOM algorithm is proved to work very well by experimental

results.

However, due to its intrinsic similarity to MDS, like other position-adjustable SOMs, our DPSOM algorithm does not work very well on such datasets as the "Swiss roll" dataset, which need the manifold learning methods.

7. References

- [1] T. Kohonen, "Self-Organized Formation of Topologically Correct Feature Maps", *Biol. Cybern.*, vol. 43, pp. 59-69, 1982.
- [2] J. Vesanto, "SOM-Based Data Visualization Methods", *Intelligent Data Analysis*, Vol. 3, pp. 111-126, 1999.
- [3] H. Yin, "ViSOM - a Novel Method for Multivariate Data Projection and Structure", Visualization. *IEEE trans. Neural networks*, vol. 13, pp. 237 - 243, 2002.
- [4] C. Shao, H. Huang, "Improvement of Data Visualization Based on SOM", *Proc. of ISNN2004*, pp. 707-712, 2004.
- [5] M. Rubio, V. Gimnez, "New Methods for Self-Organising Map Visual Analysis", *Neural Comput & Applic*, vol. 12, pp. 142-152, 2003.
- [6] D. Merkl, A. Rauber, "Alternative Ways for Cluster Visualization in Self-Organizing Maps", *Proc. of WSOM'97*, pp. 106-111, 1997.
- [7] D. Wang, H. Ressom, M. Musavi, and C. Domnisoru, "Double Self-Organizing Maps to Cluster Gene Expression Data", *Proc. of ESANN'02*, pp. 45-50, 2002.
- [8] J. Himberg, "A SOM Based Cluster Visualization and Its Application for False Coloring", *Proc. of IJCNN'00*, pp. 587-592, 2000.