Cogn Neurodyn (2009) 3:9–15
DOI 10.1007/s11571-008-9060-2

RESEARCH ARTICLE

Ordering process of self-organizing maps improved by asymmetric neighborhood function

Takaaki Aoki · Kaiichiro Ota · Koji Kurata · Toshio Aoyagi

Received: 8 April 2008 / Revised: 19 August 2008 / Accepted: 19 August 2008 / Published online: 24 September 2008

Abstract The Self-organizing map (SOM) is an unsupervised learning method based on the neural computation, which has found wide applications. However, the learning process sometime takes multi-stable states, within which the map is trapped to an undesirable disordered state including topological defects on the map. These topological defects critically aggravate the performance of the SOM. In order to overcome this problem, we propose to introduce an asymmetric neighborhood function for the SOM algorithm. Compared with the conventional symmetric one, the asymmetric neighborhood function accelerates the ordering process even in the presence of the defect. However, this asymmetry tends to generate a distorted map. This can be suppressed by an improved method of the asymmetric neighborhood function. In the case of one-dimensional SOM, it is found that the required steps for perfect ordering is numerically shown to be reduced from $O(N^3)$ to $O(N^2)$. We also discuss the ordering process of a twisted state in two-dimensional SOM, which can not be rectified by the ordinary symmetric neighborhood function.

Keywords Self-organizing map · Asymmetric neighborhood function · Fast ordering

T. Aoki (✉) · K. Ota · T. Aoyagi
Graduate School of Informatics, Kyoto University,
Kyoto 606-8501, Japan
e-mail: aoki@acs.i.kyoto-u.ac.jp

K. Kurata
Faculty of Engineering, University of the Ryukyus,
Nishihara, Okinawa 903-0213, Japan

T. Aoyagi
CREST, JST, Kyoto 606-8501, Japan

Introduction

The self-organizing map (SOM) is an unsupervised learning method of a type of nonlinear principal component analysis (Kohonen 1982). Historically, it was proposed as a simplified neural network model having some essential properties to reproduce topographic representations observed in the brain (Hubel and Wisel 1962, 1974; von der Malsburg 1973; Takeuchi and Amari 1979). The SOM algorithm can be used to construct an ordered mapping from input stimulus data onto two-dimensional array of neurons according to the topological relationships between various characters of the stimulus. This implies that the SOM algorithm is capable of extracting the essential information from complicated data. From the viewpoint of applied information processing, the SOM algorithm can be regarded as a generalized, nonlinear type of principal component analysis and has proven valuable in the fields of visualization, compression and data mining. On the basis of the biological simple learning rule, this algorithm behaves as an unsupervised learning method and provides a robust performance without a delicate tuning of learning conditions.

However, there is a serious problem of multi-stability or meta-stability in the learning process (Erwin et al. 1992; Geszti et al. 1990; Der et al. 1997). When the learning process is trapped to these states, the map seems to be converged to the final state practically. However, some of theses states are undesirable for the solution of the learning procedure, in which typically the map has topological defects as shown in Fig. 1a. In the figure, there is a two-dimensional array of the SOM map that represents a two-dimensional input space. In the case that the input data is taken uniformly from square space, this two-dimensional array should be arranged in the square space, as the result of the learning process. However, in some situation, the

array becomes to be twisted with a topological defect at the center during the learning process, as in Fig. 1a. When the array is twisted, it will require a sheer number of learning steps to rectify the topological defect, because this topological defect is a global conflicting point which is difficult to remove by local modulations of the reference vectors of units. Thus, the existence of the topological defect critically aggravates the performance of the SOM algorithm.

To avoid the emergence of the topological defect, several conventional and empirical methods have been used (Kohonen 2001). However, it is more favorable that the SOM algorithm works well without tuning any model parameters, even when the topological defect emerged. Thus, let us consider a simple method which enables the effective ordering procedure of SOM in the presence of the topological defect. Therefore we propose an asymmetric neighborhood function which effectively removes the topological defect (Aoki and Aoyagi 2007). In the process of removing the topological defect, the conflicting point must be moved out toward the boundary of the arrays and vanished. Therefore, the efficiency of the ordering process critically depends on the process of the movement of the conflicting point. As shown below, with the original symmetric neighborhood, the movement of the defect is similar to a random walk stochastic process, whose efficiency is worse. By introducing the asymmetry of the neighborhood function, the movement behaves like a drift, which enables the faster ordering. For this reason, in this article we investigate the effect of an asymmetric neighborhood function on the performance of the SOM algorithm for the case of one- and two-dimensional SOMs.

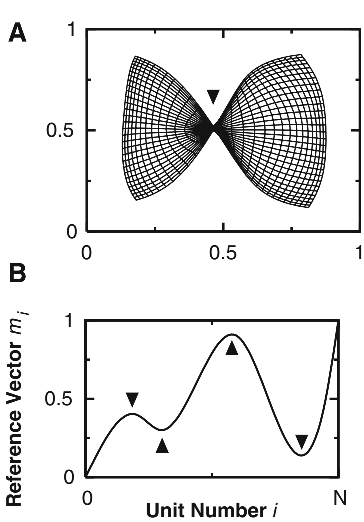

Fig. 1 (**a**) An example of a topological defect in a two-dimensional array of SOM with a uniform rectangular input space. The triangle point indicates the conflicting point in the feature map. (**b**) Another example of topological defect in a one-dimensional array with scalar input data. The triangle points also indicate the conflicting points

Methods

SOM

The SOM constructs a mapping from the input data space to the array of nodes, we call the 'feature map'. To each node i, a parametric 'reference vector' $\mathbf{m}_i$ is assigned. Through SOM learning, these reference vectors are rearranged according to the following iterative procedure. An input vector $\mathbf{x}(t)$ is presented at each time step t, and the best matching unit whose reference vector is closest to the given input vector $\mathbf{x}(t)$ is chosen. The best matching unit c, called the 'winner' is given by $c = \arg\min_i \|\mathrm{x}(t) - \mathbf{M}_i\|$. In other words, the data $\mathbf{x}(t)$ in the input data space is mapped on to the node c associated with the reference vector $\mathbf{m}_i$ closest to $\mathbf{x}(t)$. In SOM learning, the update rule for reference vectors is given by

$$\begin{aligned} &\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + \alpha \cdot h(r_{ic})[\mathbf{x}(t) - \mathbf{m}_i(t)], \\ &\quad r_{ic} \equiv \|\mathbf{r}_{ic}\| \equiv \|\mathbf{r}_i - \mathbf{r}_c\| \end{aligned} \tag{1}$$

where α, the learning rate, is some small constant. The function $h(r)$ is called the 'neighborhood function', in which r_{ic} is the distance from the position $\mathbf{r}_c$ of the winner node c to the position $\mathbf{r}_i$ of a node i on the array of units. A widely used neighborhood function is the Gaussian function defined by, $h(r_{ic}) = \exp\left(-\frac{r_{ic}^2}{2\sigma^2}\right)$. We expect an ordered mapping after iterating the above procedure a sufficient number of times.

Asymmetric neighborhood function

Let us define an asymmetry parameter β ($\beta \geq 1$), representing the degree of asymmetry and the unit vector $\mathbf{k}$ indicating the direction of asymmetry. If a unit i is located on the positive direction with $\mathbf{k}$, then the component parallel to $\mathbf{k}$ of the distance from the winner to the unit is scaled by $1/\beta$. If a unit i is located on the negative direction with $\mathbf{k}$, the parallel component of the distance is scaled by β. Hence, the asymmetric function $h_\beta(r)$, transformed from its symmetric counterpart $h(r)$, is described by

$$\begin{aligned} &h_\beta(r_{ic}) = 2\left(\frac{1}{\beta} + \beta\right)^{-1} \cdot h(\tilde{r}_{ic}), \\ &\tilde{r}_{ic} = \begin{cases} \sqrt{\left(\frac{r_\parallel}{\beta}\right)^2 + \|\mathbf{r}_\perp\|^2} & (\mathbf{r}_{ic} \cdot \mathbf{k} \geq 0) \\ \sqrt{(\beta r_\parallel)^2 + \|\mathbf{r}_\perp\|^2} & (\mathbf{r}_{ic} \cdot \mathbf{k} < 0) \end{cases}, \end{aligned} \tag{2}$$

where $\tilde{r}_{ic}$ is the scaled distance from the winner. $r_\parallel$ is the projected component of $\mathbf{r}_{ic}$, and $\mathbf{r}_\perp$ are the remaining components perpendicular to $\mathbf{k}$, respectively. Note that the overall area of the neighborhood function, $\int_{-\infty}^{\infty} h(\mathbf{r})\mathrm{d}\mathbf{r}$, is preserved under this transformation, in order to single out the effect of asymmetry in the

following comparison between a ordinal symmetric neighborhood function and the proposed asymmetric one. This condition of the preservation results in a addition of the scale factor, $2(\beta^{-1}+\beta)^{-1}$. It should be commented that the factor is valid even in the case of the two- or higher-dimensional array, because the neighborhood function is rescaled along the one direction **k** through the asymmetric transformation. In the special case of the asymmetry parameter $\beta = 1$, $h_\beta(r)$ is equal to the original symmetric function $h(r)$. Figure 2b displays an example of asymmetric Gaussian neighborhood functions in the two-dimensional array of SOM.

Next, we introduce an improved algorithm for the asymmetric neighborhood function. The asymmetry of the neighborhood function causes to distort the feature map, in which the density of units does not represent the probability of the input data. Therefore, two novel procedures will be introduced. The first procedure is an inversion operation on the direction of the asymmetric neighborhood function. As illustrated in Fig. 2c, the direction of the asymmetry is turned in the opposite direction after every time interval T, which is expected to average out the distortion in the feature map. It is noted that the interval T should be set to a larger value than the typical ordering time for the asymmetric neighborhood function. The second procedure is an operation that decreases the degree of asymmetry of the neighborhood function. When $\beta = 1$, the neighborhood function equals the original symmetric function. With this operation, β is decreased to 1 with each time step, as illustrated in Fig. 2c. In our numerical simulations, we adopt a linear decreasing function.

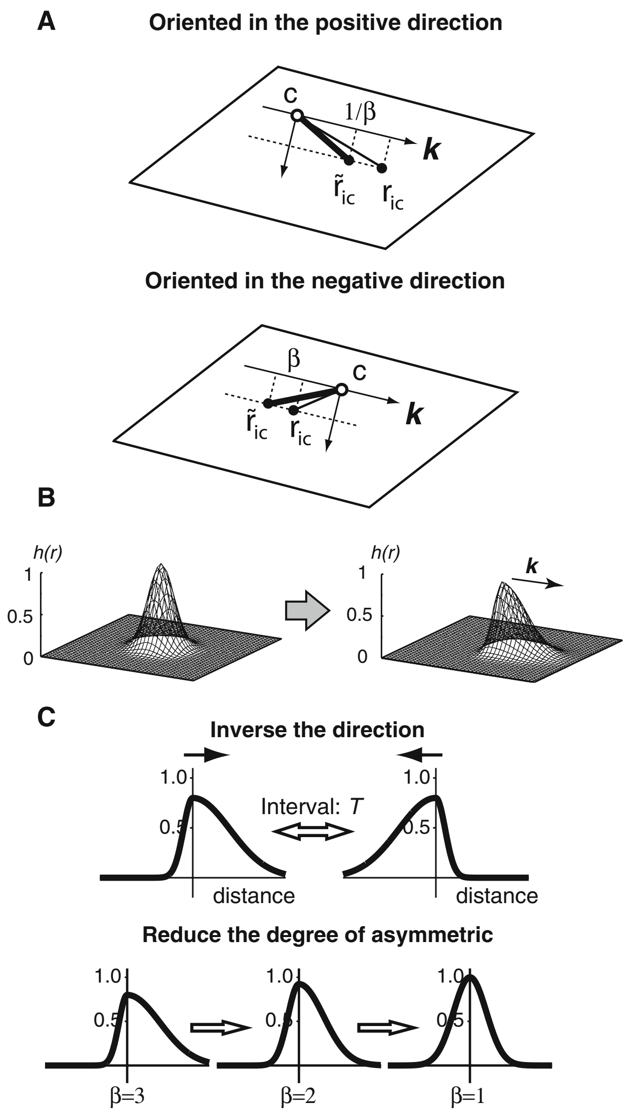

Fig. 2 (**a**) Method of generating an asymmetric neighborhood function by scaling the distance r_{ic} asymmetrically. The degree of asymmetry is parameterized by β. The distance of the node on the positive direction with asymmetric unit vector **k**, is scaled by $1/\beta$. The distance on the negative direction is scaled by β. Therefore, the asymmetric function is described by $h_\beta(r_{ic}) = \frac{2}{\beta+1/\beta} h(\tilde{r}_{ic})$ where $\tilde{r}_{ic}$ is the scaled distance of node i from the winner c. (**b**) An example of an asymmetric Gaussian function. (**c**) An illustration of the improved algorithm for asymmetric neighborhood function

Numerical simulations

In the following sections, we tested learning procedures of SOM with sample data to examine the performance of the ordering process. We investigate mainly two cases: one-dimensional array of SOM with one-dimensional input space and two-dimensional array of SOM with two-dimensional input space. The reason why we consider these simple cases is that their simplicity enable us to reveal the essential properties of the ordering process with the asymmetric neighborhood function that we introduce. The basic result in these simple cases is fundamental to an understanding of the effect of the asymmetric neighborhood function in more generic situations.

The sample data is generated from a random variable of a uniform distribution. In the case of one-dimensional SOM, the distribution is uniform in the ranges of [0, 1]. Here we use Gaussian function for the original symmetric neighborhood function. The model parameters in SOM learning was used as follows: The total number of units $N = 1{,}000$, the learning rate $\alpha = 0.05$ (constant), and the neighborhood radius $\sigma = 50$. The asymmetry parameter $\beta = 1.5$ and asymmetric direction **k** is set to the positive direction in the array. The interval period T of flipping the asymmetric direction is 3000. In the case of two-dimensional SOM (2D $\rightarrow$ 2D map), the input data taken uniformly from square space, $[0, 1] \times [0, 1]$. The model parameters are same as in one-dimensional SOM, excepted that the total number of units $N = 900$ (30×30) and $\sigma = 4$. The asymmetric direction **k** is taken in the direction (1, 0), which can be determined arbitrary. In the following

numerical simulations, we also confirmed the result holds with other model parameters and other form of neighborhood functions.

Topological order and distortion of the feature map

For the aim to examine the ordering process of SOM, lets us consider two measures which characterize the property of the feature map. One is the 'topological order' η for quantifying the order of reference vectors in the SOM array. The units of the SOM array should be arranged according to its reference vector m_i. In the presence of the topological defect, most of the units satisfy the local ordering. However, the topological defect violates the global ordering and the feature map is divided into fragments of ordering domains within which the units satisfy the order-condition. Therefore, the topological order η can be defined as the ratio of the maximum domain size to the total number of units N, given by

$$\eta \equiv \frac{\max_l N_l}{N}, \tag{3}$$

where N_l is the size of domain l. In the case of one-dimensional SOM, the order-condition for the reference vector of units is defined as, $m_{i-1} \leq m_i \leq m_{i+1}$, or $m_{i-1} \geq m_i \geq m_{i+1}$. In the case of two-dimensional SOM as referred in the previous section, the order-condition is also defined explicitly with the vector product $\mathbf{a}_{(i,i)} \equiv (\mathbf{m}_{(i+1,i)} - \mathbf{m}_{(i,i)}) \times (\mathbf{m}_{(i,i+1)} - \mathbf{m}_{(i,i)})$. Within the ordering domain, the vector products $\mathbf{a}_{(i,i)}$ of units have same sign, because the reference vectors are arranged in the sample space ordered by the position of the unit.

The other measure we consider is the 'distortion' χ, which measures the distortion of the feature map. The asymmetry of the neighborhood function tends to distort the distribution of reference vectors, which is quite different from the correct probability density of input vectors. For example, when the probability density of input vectors is uniform, a non-uniform distribution of reference vectors is formed with an asymmetric neighborhood function. Hence, for measuring the non-uniformity in the distribution of reference vectors, let us define the distortion χ. χ is a coefficient of variation of the size-distribution of unit Voronoi tessellation cells, and is given by

$$\chi = \frac{\sqrt{\mathrm{Var}(\Delta_i)}}{\mathrm{E}(\Delta_i)}, \tag{4}$$

where Δ_i is the size of Voronoi cell of unit i. To eliminate the boundary effect of the SOM algorithm, the Voronoi cells on the edges of the array are excluded. When the reference vectors are distributed uniformly, the distortion χ converges to 0. It should be noted that the evaluation of Voronoi cell in two-dimensional SOM is time-consuming, and we approximate the size of Voronoi cell by the size of the vector product $\|\mathbf{a}_{(i,i)}\|$. If the feature map is uniformly formed, the approximate value also converges to 0.

Results

One-dimensional case

In this section, we investigate the ordering process of SOM learning in the presence of a topological defect in symmetric, asymmetric, and improved asymmetric cases of the neighborhood function. For this purpose, we use the initial condition that a single topological defect appears at the center of the array. Because the density of input vectors is uniform, the desirable feature map is a linear arrangement of SOM nodes.

Figure 3a shows a typical time development of the reference vectors m_i. In the case of the symmetric neighborhood function, a single defect remains around the center of the array even after 10,000 steps. In contrast, in the case of the asymmetric one, this defect moves out to the right so that the reference vectors are ordered within 3,000 steps. This phenomenon can also be confirmed in Fig. 3b, which shows the time dependence of the topological order η. In the case of the asymmetric neighborhood function, η rapidly converges to 1 (completely ordered state) within 3,000 steps, whereas the process of eliminating the last defect takes a large amount of time ($\sim$18,000 step) for the symmetric one.

On the other hand, one problem arises in the feature map obtained with the asymmetric neighborhood function. After 10,000 steps, the distribution of the reference vectors in the feature map develops an unusual bias (Fig. 3a). Figure 3c shows the time dependence of the distortion χ during learning. In the case of the symmetric neighborhood function, χ eventually converges to almost 0. This result indicates that the feature map obtained with the symmetric one has an almost uniform size distribution of Voronoi cells. In contrast, in the case of the asymmetric one, χ converges to a finite value ($\neq$0). Although the asymmetric neighborhood function accelerates the ordering process of SOM learning, the resultant map becomes distorted which is unusable for the applications. Therefore, the improved asymmetric method will be introduced, as mentioned in Method. Using this improved algorithm, χ converges to almost 0 as same as the symmetric one (Fig. 3c). Furthermore, as shown in Fig. 3b, the improved algorithm preserves the faster order learning. Therefore, by utilizing the improved algorithm of asymmetric neighborhood function, we confer the full benefit of both the fast order learning and the undistorted feature map.

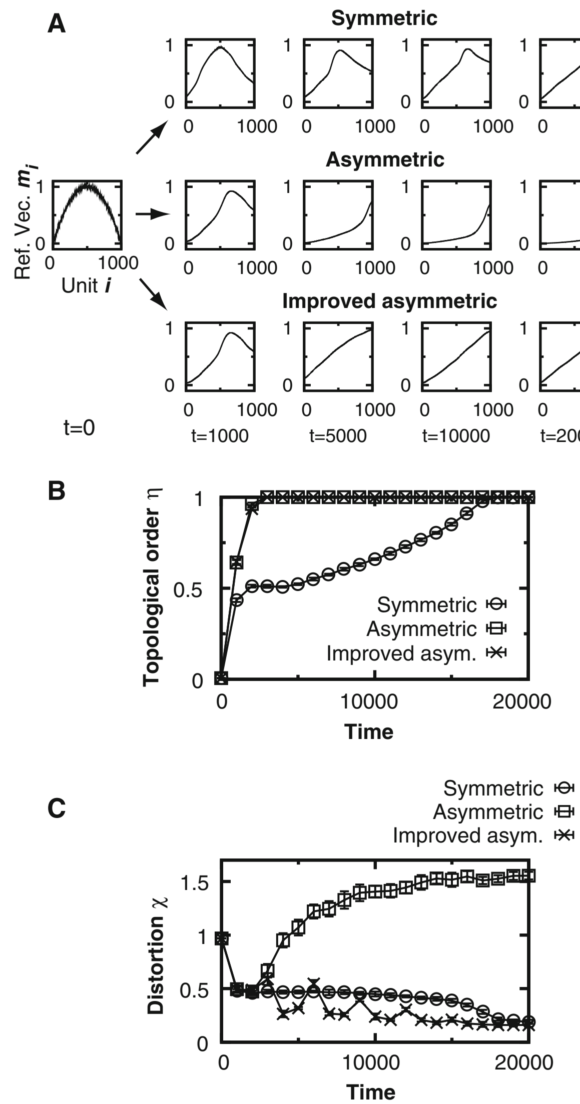

Fig. 3 The asymmetric neighborhood function enhances the ordering process of SOM. (**a**) A typical time development of the reference vectors m_i in cases of symmetric, asymmetric, and improved asymmetric neighborhood functions. (**b**) The time dependence of the topological order η. The standard deviations are denoted by the error bars, which cannot be seen because they are smaller than the size of the graphed symbol. (**c**) The time dependence of the distortion χ

To quantify the performance of the ordering process, let us define the 'ordering time' as the time at which η reaches to 1. Figure 4a shows the ordering time as a function of the total number of units N for both improved asymmetric and symmetric cases of the neighborhood function. It is found that the ordering time scales roughly as N^3 and N^2 for symmetric and improved asymmetric neighborhood functions, respectively. For detailed discussion about the reduction of the ordering time, refer to the Aoki and Aoyagi (2007). Figure 4b shows the dependency of the ordering time on the width of neighborhood function, which indicates that ordering time is proportional to $(N/\sigma)^k$ with $k = 2/3$ for asymmetric/symmetric neighborhood function. This result implies that if one use annealing method for the width combined with the asymmetric method, the method will be more effective.

Two-dimensional case

In this section, we investigate the effect of asymmetric neighborhood function for two-dimensional SOM (2D → 2D map). Figure 5 shows that a similar fast ordering process can be realized with an asymmetric neighborhood function in two-dimensional SOM. The initial state has a global topological defect, in which the map is twisted at the center. In this situation, the conventional symmetric neighborhood function has trouble in correcting the twisted map. Because of the local stability, this topological defect is never corrected even with a huge learning iteration. The asymmetric neighborhood function also is effective to overcome such a topological defect, like the case of one-dimensional SOM. However, the same problem of 'map distortion' occurs. Therefore, by using the improved asymmetric neighborhood function, the feature map converges to the completely ordered map in much less time without any distortion.

The width of neighborhood function is an important parameter which affects the ordering process in rectification of the twisted state. Therefore, we discuss a dependency of

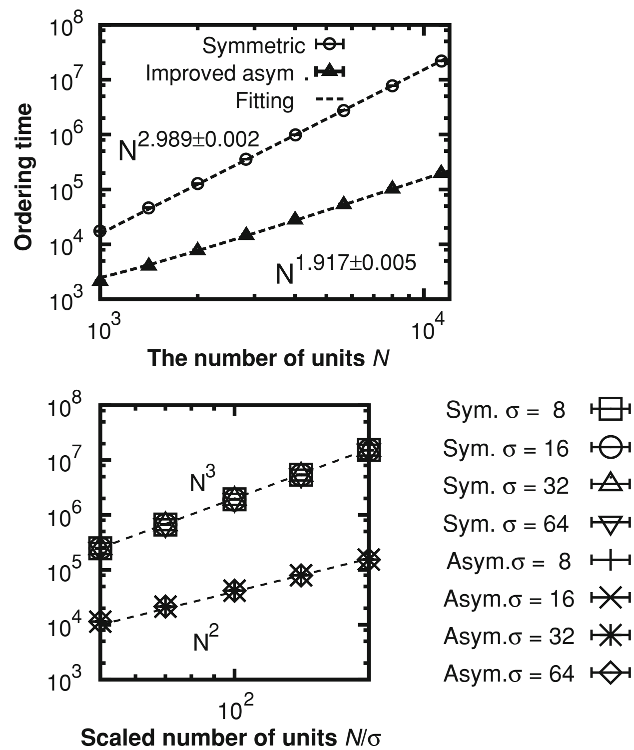

Fig. 4 Ordering time as a function of the total number of units N. The fitting function is described by $Const. \cdot N^{\gamma}$

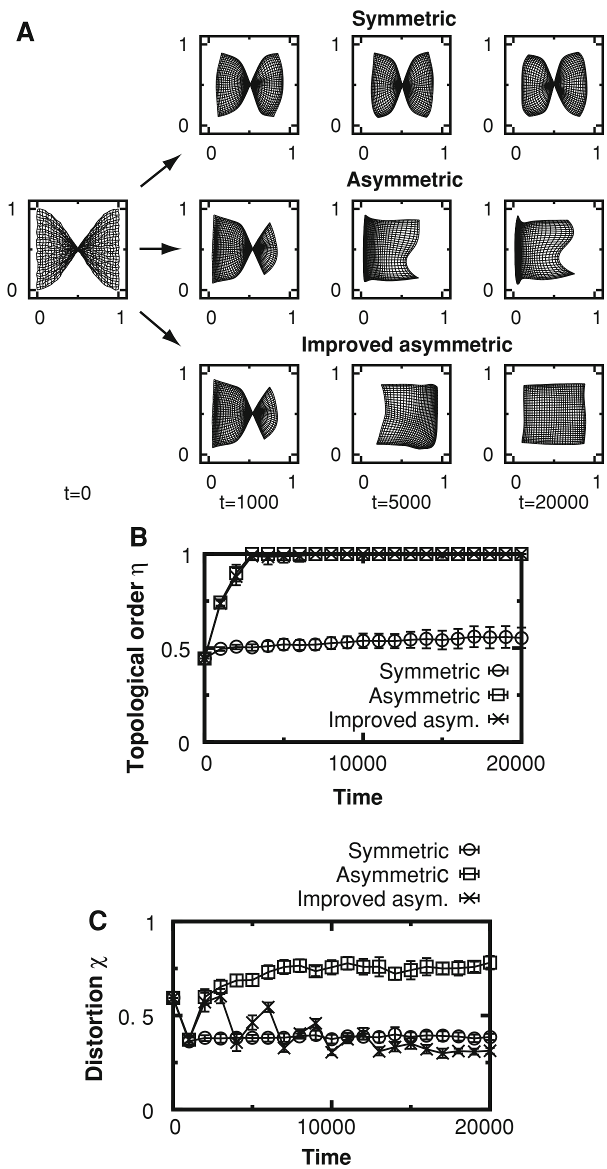

Fig. 5 (**a**) A typical time development of reference vectors in two-dimensional array of SOM for the cases of symmetric, asymmetric and improved asymmetric neighborhood functions. (**b**) The time dependence of the topological order η. (**c**) The time dependence of the distortion χ

the width σ. Figure 6a shows ordering time which is required learning steps until the topological order η becomes to 1 (perfect order), as a function of σ. As shown in the figure, in the case of symmetric neighborhood function, there is a value for σ at which the required step to rectify the twisted state exceeds to a practical range of steps in numerical simulations. In contrast, in the case of asymmetric neighborhood function, the ordering time does not diverge.

In the previous simulations, we have considered a simple situation that a single defect exists around the center of the feature map as an initial condition in order to investigate the ordering process with the topological defect.

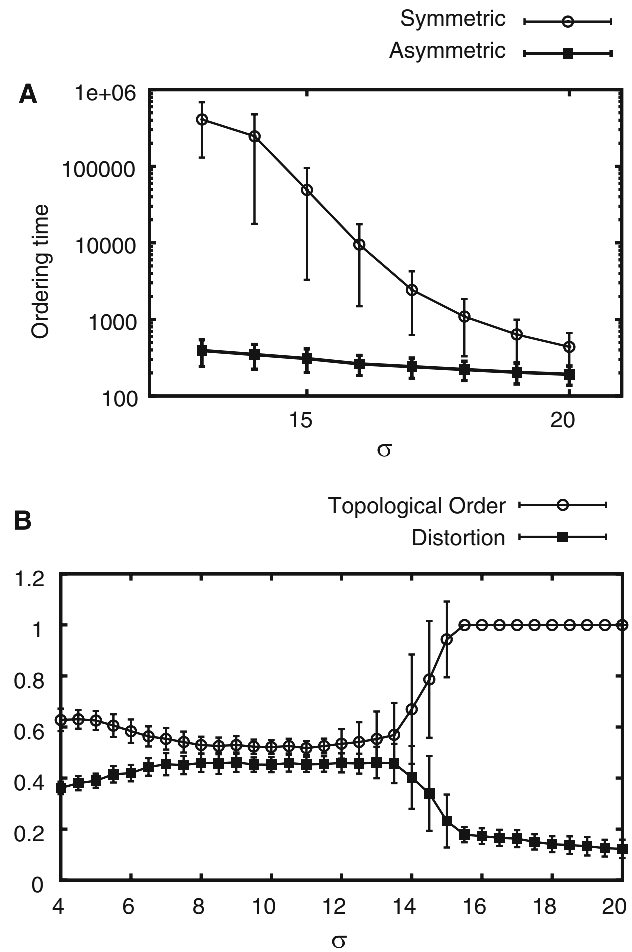

Fig. 6 Dependency on the width σ of neighborhood function for rectification of the twisted state in two-dimensional array of SOM. The units are 30×30. The input data and the initial reference are same in Fig. 5. (**a**) Ordering time as a function of the width σ in the cases of the symmetric and asymmetric neighborhood function. In the case of ordinal symmetric neighborhood function, as the width σ becomes smaller, the ordering time increases remarkably, and quickly exceeds a limit of learning steps for practical usage. (**b**) Topological order η and distortion χ obtained after 10^5 trials. For small σ, topological order η does not converge to 1 (ordered state)

However, when the initial reference vectors are set randomly, the total number of topological defects appearing in the map is not generally equal to one. Therefore, it is necessary to consider the statistical distribution of the ordering time, because the total number of the topological defects and the convergence process depend generally on the initial conditions. Figure 7 shows the distribution of the ordering time, when the initial reference vectors are randomly selected from the uniform distribution $[0, 1] \times [0, 1]$. In the case of symmetric neighborhood function, the learning process could not converged to the ordered state for some part of trials, even after 5.0×10^5 steps. For these trials, the learning processes are trapped in undesirable meta-stable states in which the array of the map has the

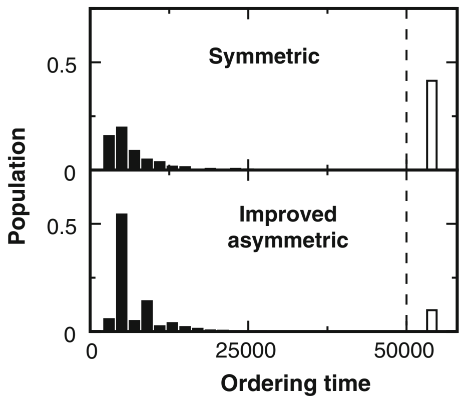

Fig. 7 Distribution of ordering times when the initial reference vectors are generated randomly. The white bin at the right in the graph indicates a population of failed trails which could not converge to the perfect ordering state within 50,000 steps

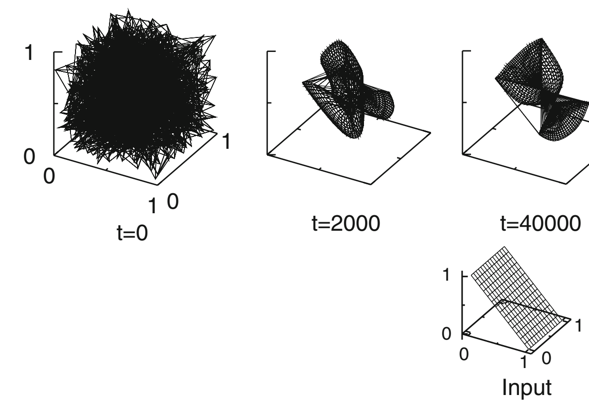

Fig. 8 A typical example of twisted state for a high-dimensional input space. When a input space has a high-dimensional input space, a twist of the network of reference vectors is rare. It should be noted, however, that the effective dimension of the input data can be low depending on the distribution of data embedded in the input space, and it causes that the twisted state can be formed during the learning period. In the figure, the input data are selected from a uniform distribution on a sub-plane in 3D input space with a small noise

topological defects. Therefore, although the fast ordering process is observed in some successful cases (lucky initial conditions), the formed map with symmetric one is highly dependent on the initial conditions. In contrast, for the improved asymmetric neighborhood function, the distribution of the ordering time has a single sharp peak and the successful feature map is constructed stably without any tuning of the initial condition.

Discussion

In this article, we discussed the learning process of the self-organized map, especially in the presence of a topological defect. Interestingly, even in the presence of the defect, we found that the asymmetry of the neighborhood function enables the system to accelerate the learning process. Compared with the conventional symmetric one, the convergence time of the learning process can be roughly reduced from $O(N^3)$ to $O(N^2)$ in one-dimensional SOM(N is the total number of units). Furthermore, this acceleration with the asymmetric neighborhood function is also effective in the case of two-dimensional SOM (2D → 2D map). In contrast, the conventional symmetric one can not rectify the twisted feature map even with a sheer of iteration steps due to its local stability. These results suggest that the proposed method can be effective for a more general case of SOM, which is the subject of future study.

In practical usage of SOM, the input space of data has a high-dimensional space. In this situation, some types of topological defects, such as a twisted state shown in this article, would be rare due to a mismatch of dimension between the input space and the array of SOM. It should be noted, however, that the distribution of input data embedded in the high-dimensional space, can have a intrinsic structure of a lower-dimensional subspace, which we want to estimate. In this case, it causes that the network of reference vectors can be twisted on the low-dimensional subspace, as shown in Fig. 8.

Acknowledgments This work was supported by KAKENHI (20033012, 18047014) from MEXT and Grant-in-Aid for JSPS Fellows.

References

Aoki T, Aoyagi T (2007) Self-organizing maps with asymmetric neighborhood function. Neural Comput 19(9):2515–2535
Der R, Herrmann M, Villmann T (1997) Time behavior of topological ordering in self-organizing feature mapping. Biol Cybern 77(6):419–427
Erwin E, Obermayer K, Schulten K (1992) Self-organizing maps—stationary states, metastability and convergence rate. Biol Cybern 67(1):35–45
Geszti T, Csabai I, Cazokó F, Szakács T, Serneels R, Vattay G (1990) Dynamics of the Kohonen map. In: Statistical mechanics of neural networks: proceedings of the XIth Sitges conference. Springer-Verlag, New York, pp 341–349
Hubel DH, Wiesel TN (1962) Receptive fields, binocular interaction and functional architecture in cats visual cortex. J Physiol (Lond) 160(1):106–154
Hubel DH, Wiesel TN (1974) Sequence regularity and geometry of orientation columns in monkey striate cortex. J Comp Neurol 158(3):267–294
Kohonen T (1982) Self-organized formation of topologically correct feature maps. Biol Cybern 43(1):59–69
Kohonen T (2001) Self-organizing maps, 3rd edn. Springer-Verlag, New York
Takeuchi A, Amari S (1979) Formation of topographic maps and columnar microstructures in nerve fields. Biol Cybern 35(2):63–72
von der Malsburg C (1973) Self-organization of orientation sensitive cells in striate cortex. Kybernetik 14(2):85–100