[9] D. Cheng and H. Qi, "Controllability and observability of Boolean control networks," *Automatica*, vol. 45, no. 7, pp. 1659–1667, Jul. 2009.
[10] Y. Zhao, H. Qi, and D. Cheng, "Input-state incidence matrix of Boolean control networks and its applications," *Syst. Control Lett.*, vol. 59, no. 12, pp. 767–774, Dec. 2010.
[11] F. Li, J. Sun, and Q.-D. Wu, "Observability of Boolean control networks with state time delays," *IEEE Trans. Neural Netw.*, vol. 22, no. 6, pp. 948–954, Jun. 2011.
[12] G. Ambika and R. E. Amritkar, "Anticipatory synchronization with variable time delay and reset," *Phys. Rev. E*, vol. 79, no. 5, pp. 056206-1–056206-11, May 2009.
[13] Y. Kuramoto, *Chemical Oscillation, Waves, and Turbulence*. Berlin, Germany: Springer-Verlag, 1984.
[14] E. Mosekilde, Y. Maistrenko, and D. Postnov, *Chaotic Synchronization: Applications to Living Systems*. Singapore: World Scientific, 2002.
[15] M. Steriade, E. G. Jones, and R. R. Llinas, *Thalamic Oscillations and Signaling*. New York: Wiley, 1990.
[16] J. Liang, Z. Wang, Y. Liu, and X. Liu, "Robust synchronization of an array of coupled stochastic discrete-time delayed neural networks," *IEEE Trans. Neural Netw.*, vol. 19, no. 11, pp. 1910–1921, Nov. 2008.
[17] J. Lu and D. W. C. Ho, "Globally exponential synchronization and synchronizability for general dynamical networks," *IEEE Trans. Syst., Man, Cybern. Part B: Cybern.*, vol. 40, no. 2, pp. 350–361, Apr. 2010.
[18] B. Shen, Z. Wang, and X. Liu, "Bounded H_∞ synchronization and state estimation for discrete time-varying stochastic complex networks over a finite horizon," *IEEE Trans. Neural Netw.*, vol. 22, no. 1, pp. 145–157, Jan. 2011.
[19] L. G. Morelli and D. H. Zanette, "Synchronization of stochastically coupled cellular automata," *Phys. Rev. E*, vol. 58, no. 1, pp. R8–R11, Jul. 1998.
[20] F. Bagnoli and R. Rechtman, "Synchronization and maximum Lyapunov exponents of cellular automata," *Phys. Rev. E*, vol. 59, no. 2, pp. R1307–R1310, Feb. 1999.
[21] L. G. Morelli and D. H. Zanette, "Synchronization of Kauffman networks," *Phys. Rev. E*, vol. 63, no. 3, pp. 036204-1–036204-10, Mar. 2001.
[22] M.-C. Ho, Y.-C. Hung, and I.-M. Jiang, "Stochastic coupling of two random Boolean networks," *Phys. Lett. A*, vol. 344, no. 1, pp. 36–42, Aug. 2005.
[23] C. Zhou, L. Zemanová, G. Zamora-López, C. C. Hilgetag, and J. Kurths, "Structure–function relationship in complex brain networks expressed by hierarchical synchronization," *New J. Phys.*, vol. 9, no. 6, p. 178, Jun. 2007.
[24] J. Garcia-Ojalvo, M. B. Elowitz, and S. H. Strogatz, "Modeling a synthetic multicellular clock: Repressilators coupled by quorum sensing," *Proc. Nat. Acad. Sci. United States Amer.*, vol. 101, no. 30, pp. 10955–10960, Jul. 2004.
[25] Y.-C. Hung, M.-C. Ho, J.-S. Lih, and I.-M. Jiang, "Chaos synchronization of two stochastically coupled random Boolean networks," *Phys. Lett. A*, vol. 356, no. 1, pp. 35–43, Jul. 2006.
[26] J. L. Guisado, F. Jiménez-Morales, and J. M. Guerra, "Cellular automaton model for the simulation of laser dynamics," *Phys. Rev. E*, vol. 67, no. 6, p. 066708, Jun. 2003.
[27] A. Veliz-Cuba and B. Stigler, "Boolean models can explain bistability in the lac operon," *J. Comput. Biol.*, vol. 18, no. 6, pp. 783–794, Jun. 2011.
[28] Y. Hong and X. Xu, "Solvability and control design for dynamic synchronization of Boolean networks," in *Proc. 29th Chin. Control Conf.*, Beijing, China, Jul. 2010, pp. 805–810.
[29] D. Cheng and H. Qi, "A linear representation of dynamics of Boolean networks," *IEEE Trans. Autom. Control*, vol. 55, no. 10, pp. 2251–2258, Oct. 2010.
[30] D. Cheng, "Input-state approach to Boolean networks," *IEEE Trans. Neural Netw.*, vol. 20, no. 3, pp. 512–521, Mar. 2009.
[31] D. Cheng, H. Qi, and Z. Li, *Analysis and Control of Boolean Networks: A Semi-Tensor Product Approach*. London, U.K.: Springer-Verlag, 2011.
[32] D. Cheng, H. Qi, and Z. Li, "Model construction of Boolean network via observed data," *IEEE Trans. Neural Netw.*, vol. 22, no. 4, pp. 525–536, Apr. 2011.
[33] C. Farrow, J. Heidel, J. Maloney, and J. Rogers, "Scalar equations for synchronous Boolean networks with biological applications," *IEEE Trans. Neural Netw.*, vol. 15, no. 2, pp. 348–354, Mar. 2004.

Data-Driven Cluster Reinforcement and Visualization in Sparsely-Matched Self-Organizing Maps

Narine Manukyan, Margaret J. Eppstein, and Donna M. Rizzo

***Abstract*— A self-organizing map (SOM) is a self-organized projection of high-dimensional data onto a typically 2-dimensional (2-D) feature map, wherein vector similarity is implicitly translated into topological closeness in the 2-D projection. However, when there are more neurons than input patterns, it can be challenging to interpret the results, due to diffuse cluster boundaries and limitations of current methods for displaying interneuron distances. In this brief, we introduce a new cluster reinforcement (CR) phase for sparsely-matched SOMs. The CR phase amplifies within-cluster similarity in an unsupervised, data-driven manner. Discontinuities in the resulting map correspond to between-cluster distances and are stored in a boundary (B) matrix. We describe a new hierarchical visualization of cluster boundaries displayed directly on feature maps, which requires no further clustering beyond what was implicitly accomplished during self-organization in SOM training. We use a synthetic benchmark problem and previously published microbial community profile data to demonstrate the benefits of the proposed methods.**

***Index Terms*— Boundary matrix (B-matrix), cluster reinforcement, cluster visualization, self-organizing map (SOM), unified distance matrix (U-matrix).**

I. Introduction

Finding patterns in vast multidimensional data sets can be difficult and time-consuming, especially when there are nonlinear relationships between multiple dimensions in the data. The biologically inspired self-organizing map (SOM) algorithm proposed by Kohonen [1], [2] can be used to organize high-dimensional data and enable users to identify both linear and nonlinear relationships between vectors in the data. Briefly, an SOM comprises a grid of so-called "neurons" (weight vectors). Input pattern vectors are compared to these neurons one at a time in random order; for each input vector, the closest neuron is identified (commonly called the "best matching unit," or BMU). Neuron values are then updated (trained) to be more similar to each input pattern, in such a way that updates are larger for neurons topologically closer to the BMU. This process is repeated for a number of iterations and the size of the "neighborhood" around the winning neuron shrinks with successive iterations. In this way, the neuron weights self-organize (phase I) and then converge (phase II), so that topologically close neurons have more similar weights than more distant neurons. Since its introduction twenty years

Manuscript received June 24, 2011; accepted March 3, 2012. Date of publication April 4, 2012; date of current version May 2, 2012. This work was supported in part by the U.S. Department of Transportation through the University of Vermont Transportation Research Center.

N. Manukyan and M. J. Eppstein are with the Department of Computer Science, University of Vermont, Burlington, VT 05401 USA (e-mail: Narine.Manukyan@uvm.edu; Maggie.Eppstein@uvm.edu).

D. M. Rizzo is with the School of Engineering, University of Vermont, Burlington, VT 05401 USA (e-mail: Donna.Rizzo@uvm.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2012.2190768

ago, the SOM algorithm has proven useful in a variety of application domains. However, extracting meaningful information about relationships between input patterns from trained SOMs can be challenging.

In densely-matched SOMs (i.e., where most neurons are the BMUs of multiple input patterns), hierarchical agglomerative clustering algorithms can exploit the information embedded in the topological organization of the BMUs during clustering of the input data (e.g., [3], [4]). However, using more neurons than inputs can be desirable in some applications because this preserves more subtle information about relationships between the individual input patterns (e.g., [5], [6]). In sparsely-matched SOMs, unique BMUs are trained to closely match individual input patterns, while the intervening nonmatched neurons are trained to interpolate between the topologically nearest BMUs. The resulting diffuse boundaries between BMUs complicate the determination of clustering patterns that are implicitly embedded in the SOM topology.

Interneuron distances have traditionally been used for SOM visualization (e.g., [7]–[9]). For example, the "unified distance matrix" (U-matrix) [10], [11] is often used to visualize the SOM output (e.g., [5], [6], [12], [13]). However, U-matrix visualization can be insufficient for identifying clusters in sparsely-matched SOMs that have diffuse cluster boundaries.

We propose: 1) a new cluster reinforcement (CR) phase, to be applied after the self-organization and convergence phases of the SOM, that advances cluster separation in sparsely-matched SOMs by strengthening cluster boundaries in a data-driven manner and 2) a new boundary (B) matrix visualization technique for displaying the resulting sharpened boundaries directly onto feature maps. Together, these two advances permit hierarchical visualization of the self-organized clusters already trained into the SOM topology, without performing any additional clustering *per se*.

This brief is organized as follows. In Section II, we describe the proposed CR phase. In Section III, we explain how to visualize the reinforced inter-cluster distances using the proposed B-matrix. In Section IV, we demonstrate the new approaches on the classic animal benchmark problem [14], and show why agglomerative image segmentation [15] is not sufficient. Finally, in Section V, we use previously published microbial community profile data [5], [16] to demonstrate how the proposed methods facilitate automatic identification and visualization of clusters in real-world high-dimensional data with complex relationships.

II. CR Phase

After the self-organization and convergence phases of training a sparsely-matched SOM, the resulting map forms a relatively smooth multidimensional interpolation between separated BMUs. This gradual change in neuron values across the map can make it difficult to clearly visualize cluster boundaries and to accurately characterize inter-cluster distances. Thus, we propose an additional CR phase. This cluster sharpening proceeds in an unsupervised data-driven manner, without prior knowledge of the domain or of the number of clusters to be identified. Unlike the first two SOM phases, in which iterative asynchronous updates are performed within topological neighborhoods surrounding each BMU in the grid, the CR phase iteratively performs synchronous updates within neighborhoods determined by distances between input vectors and all neurons in the map. The CR phase reduces within-cluster differences by updating all neurons to more closely match the input vectors they are already most similar to, and thus sharpens boundaries between clusters. Discontinuities in the resulting map can then be interpreted as between-cluster distances, facilitating subsequent identification and visualization of boundaries between existing self-organized clusters, without necessarily doing any clustering *per se*.

Algorithm 1 Pseudo-code for the CR phase, to be run as a post-processing phase after the SOM

```
   FUNCTION C = CR(X, W, σ0, t_max, [p])
   Input: X, W, σ0, t_max, [p]
   Output: C
1  C = W;
2  σ = σ0;
3  for t = 1 to t_max do
4      for j = 0 to N − 1 do
5          Δc_j = 0;
6      for i = 0 to L − 1 do
7          for j = 0 to N − 1 do
8              h = exp(−‖X_i − C_j‖² / (2σ²)) σ;
9              ΔC_j = ΔC_j + h(X_i − C_j). × [p];
10     for j = 0 to N − 1 do
11         C_j = C_j + ΔC_j;
12     σ = σ0 exp(−t / t_max);
```

Pseudo-code for the CR phase is shown in Algorithm 1, where the inputs are X (the list of L input vectors that are individually denoted as X_i, each with M features), W (the SOM weight map previously trained on X, comprising N neurons each of length M), σ_0 (a user-defined control parameter), and, optionally, p (a vector of M importance weights corresponding to the different features in each input vector). The output map C is initialized from the trained SOM input map W and then trained for a prespecified number of timesteps $t_{\max}$. For each training iteration (lines 3–12), we loop through all L input vectors (line 6) and all N neurons (line 7), calculate the neighborhood update weight h for the neuron (line 8), and calculate and accumulate neuron vector updates in ΔC_j (line 9). (The use of optional feature importance weights p in line 9 of the CR update parallels the weighted SOM update described in [5]; the operator ".×" indicates element-by-element multiplication.) After the update vectors for all N neurons and all L input patterns have been accumulated, we apply these updates synchronously (lines 10–11). The use of synchronous updates ensures consistent performance and makes randomization of the applied input patterns unnecessary.

A distinguishing feature of CR updates, relative to SOM updates, is that the strength of neuron membership in a CR update neighborhood (h, line 8) is based on the *Euclidean distance between the values* of the input vector and the neuron ($||X_i - C_j||$), in contrast to the standard SOM neighborhood membership, which is based on the *topological distance in the grid* between the neuron and the BMU of the input vector. Because the input to the CR phase is the SOM-trained map, there is no need for further topological organization. Rather, the CR phase updates neurons to be more similar to input vectors they are already similar to, with individual features updated by different amounts based on the feature-specific differences between the input vector and the neuron's weight vector ($X_i - C_j$) (line 9). The result is that the map is altered away from one that smoothly interpolates between BMUs into one with stepwise discontinuities between the self-organized clusters of BMUs.

In the standard SOM, separate decreasing control parameters are required for the neighborhood size and the learning rate. However, in the CR phase, the exponentially decreasing parameter σ (line 12) performs double duty as the standard deviation of the Gaussian neighborhood kernel (i.e., determining the width of the Gaussian function) and as the learning rate (i.e., determining the height of the Gaussian function, which scales the maximum size of the update), since both of these are based on the same measure of distances between vector values. Thus, the only user-specified control parameter for the CR phase is σ_0. In general, we have found that $t_{max} = 150$ gives robust results (although for relatively higher σ_0 this many iterations may not be required). If the goal is to sharpen the boundaries between existing adjacent self-organized clusters, without performing any additional clustering, one should use as low a value of σ_0 as possible to achieve sharp boundaries. A good heuristic is to initialize σ_0 to one tenth of the maximum Euclidean distance between neurons in W; if cluster boundaries remain too diffuse [e.g., as visualized later in Fig. 4(c)], one can slowly increase the value of σ_0 until the boundaries sharpen [e.g., as visualized later in Fig. 4(f)]. Alternatively, one can tune σ_0 to higher values to perform agglomerative clustering. It is possible this may prove to be an effective means of clustering SOMs, even if densely-matched, although this has yet to be explored and is not the approach we are advocating in this brief. The time complexity of each iteration of the CR phase is $O(LMN)$, the same as for the SOM; however, fewer iterations are required for the CR phase.

III. Cluster Boundary Visualization

The magnitudes of interneuron discontinuities in the cluster-reinforced map C can be interpreted as degrees-of-separation between adjacent clusters of similar BMUs in the SOM. In the following, we assume the topology of the N neurons in C is arranged in a square ($n \times n$, where $n = \sqrt{(N)}$) grid (the approach is easily generalized to rectangular or hexagonal grids). We then compute distances between adjacent neurons

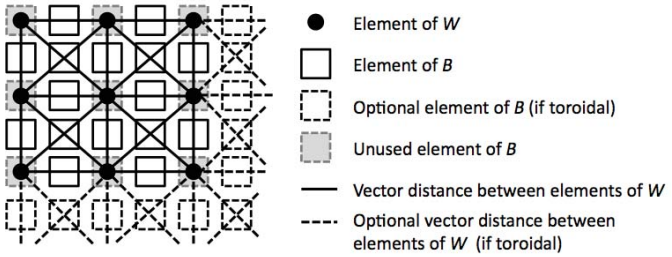


Fig. 1. Correspondence between the elements of B (squares) and W (black dots).

and store them in a B-matrix of size $2n \times 2n$, as follows:

$$B_{i,j}(W) = \begin{cases} \text{NaN} & \text{even}(i), \text{even}(j) \\ \text{Dist}\left(W_{k,l}, W_{v,l}\right) & \text{odd}(i), \text{even}(j) \\ \text{Dist}\left(W_{k,l}, W_{k,q}\right) & \text{even}(i), \text{odd}(j) \\ \text{mean}\begin{pmatrix} \text{Dist}\left(W_{k,l}, W_{v,q}\right) \\ \text{Dist}\left(W_{v,l}, W_{k,q}\right) \end{pmatrix} & \text{odd}(i), \text{odd}(j) \end{cases} \quad (1)$$

where

$$k = \text{floor}(i/2) \quad \text{and} \quad l = \text{floor}(j/2)$$
$$v = \text{mod}(k+1, n) \quad \text{and} \quad q = \text{mod}(l+1, n)$$
$$\text{even}(i) = \begin{cases} \text{True}, & \text{if } i \text{ is an even number} \\ \text{False}, & \text{otherwise} \end{cases}$$
$$\text{odd}(i) = \neg\text{even}(i).$$

Equation (1) assumes that matrix indexing runs from 0 to $2n - 1$, floor(x) rounds down to the nearest integer, and mod (x, y) is x modulo y. B is defined here to be toroidal, although it can be easily modified to a $2n - 1 \times 2n - 1$ nontoroidal matrix, if desired. Any vector distance metric Dist may be used; we employ Euclidean distances.

The elements of the B-matrix with two even indices (Fig. 1, shaded squares) correspond to the $n \times n$ neurons in W (Fig. 1, black dots) and remain unused, indicated in (1) by not a number (NaN). The remaining elements of B (Fig. 1, open squares), referred to as B-values, contain distances between adjacent elements of W: those shown on horizontal lines contain the distances between horizontally adjacent elements of W; those on vertical lines contain the distances between vertically adjacent elements of W; those shown on crossing diagonal lines contain the mean of the distances between diagonally adjacent elements of W.

It is important to note the similarities and differences between the $2n \times 2n$ B-matrix and the $n \times n$ U-matrix. Neither B nor U are clustering methods *per se*; they simply record interneuron distances. Each element of U is calculated as the average distance from the corresponding neuron in W to its eight topologically nearest neighbors in a rectangular SOM grid [10]. This averaging can sometimes obscure individual interneuron distances and make cluster boundaries difficult to identify. Exactly the same number of interneuron distances must be computed for the B-matrix as for the U-matrix. However, because the B-matrix is larger, more of these are stored separately, and thus the B-matrix retains more information about interneuron distances than the U-matrix of averaged

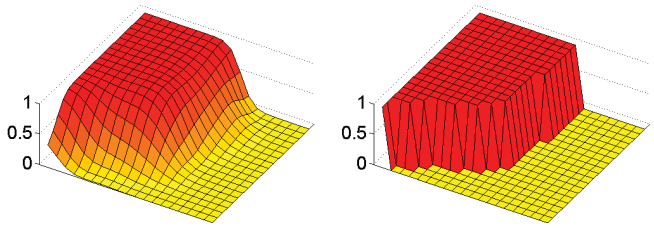
distances. We note that some SOM visualization packages already permit the display of nonaveraged interneuron distances (e.g., [17]), but without prior cluster sharpening these can be of limited usefulness in determining cluster boundaries in sparsely-matched SOMs. To our knowledge, such distances have not been displayed using grid lines of varying thicknesses on component planes, as we suggest below.

It is difficult to simultaneously visualize the U-matrix and a component plane of W on the same graph, since these matrices are the same size. Although one can overlay a contour plot of the U-matrix on a heat map of a component plane, it can be difficult to identify cluster boundaries from such a contour plot [as illustrated later in Fig. 4(d)]. In contrast, because the B-matrix is $2n \times 2n$ and the $N = n^2$ elements of B that directly correspond to the neurons of W remained unused, one can overlay a heat map of a component plane of W with a simultaneous display of the B-values shown as grid lines drawn between the neuron component values of W. By optionally limiting the display of grid lines for B-values above a tunable user-specified minimum value θ, and making the widths of the displayed grid lines proportional to the B-values, one can hierarchically visualize the different levels of clustering already embedded in the SOM, without actually performing any clustering. This is in contrast to agglomerative clustering methods, including image segmentation algorithms, that require computationally intensive merging of adjacent neurons for user-specified levels of clustering. The visualization techniques discussed in this section are illustrated and compared in the following sections.

IV. Benchmark Application: Animal Data Set

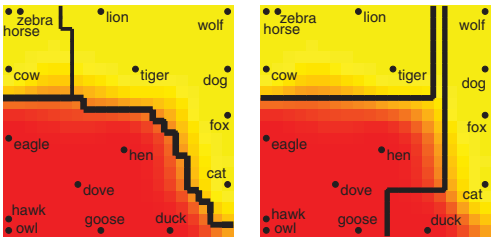
In this section, we demonstrate the CR phase and B-matrix visualization on the classic SOM benchmark problem in which 16 species of animals self-organize based on the similarity of 13 binary features [14]. After SOM training using a 20×20 nontoroidal rectangular grid, sparsely-matched neuron values for each component plane form a continuous surface that smoothly interpolate between the BMUs, as illustrated in Fig. 2(a) for the component plane "has feathers." After the CR phase, using $\sigma_0 = 0.15$ (about one fifth of the maximum Euclidean distance in W of 0.75), the boundary between birds (high plateau) and mammals (low plateau) forms a sharp discontinuity Fig. 2(b). We have elected to display an SOM for the animal benchmark that was trained without first normalizing the input vectors, to make the component planes easier to interpret. However, the results were not qualitatively different when we normalized first.

In Fig. 3(a), the thicknesses of the grid lines are proportional to B-values representing interneuron distances in the cluster-reinforced C map (showing only those distances above $\theta = 1.75$), superimposed on top of a heat map for the same "has feathers" component plane in the trained SOM, with the locations of the animal BMUs indicated. The thickest grid lines in Fig. 3(a) separate the two large natural clusters comprising the birds and the mammals (with a mean Silhouette value s [18] of 0.44 for these two clusters). The thinner grid line in Fig. 3(a) indicates there is a slightly less well-defined third subcluster of mammals comprising the ungulates



(a) (b)

Fig. 2. Results on animal data. (a) "Has feathers" component plane after SOM training. (b) Same component plane after the CR phase. BMUs for birds are located in the region of the high plateau, whereas those for mammals are located in the low plateau.



(a) (b)

Fig. 3. Results on animal data. (a) Thicknesses of the grid lines correspond to B-values of interneuron distances in the C matrix ($\theta = 1.75$), superimposed on the "has feathers" component plane of W for the animals benchmark. (b) Grid lines indicate the boundaries of a three-cluster image segmentation, using statistical region merging [15].

($mean(s) = 0.43$ for these three clusters). For comparison, we show the results of applying a state-of-the-art image segmentation algorithm, statistical region merging [15], that uses a user-defined control parameter to specify the level of clustering. A three-cluster segmentation (Fig. 3(b), $mean(s) = 0.27$) shows that the mammals have been classified into two disjoint clusters, and the duck has been erroneously grouped with one of the mammal groups ($s_{\text{duck}} = -0.46$). With a two-cluster segmentation the rightmost cluster boundary shown in Fig. 3(b) is eliminated, thus lumping the birds with half of the mammals (resulting in four negative Silhouette values).

The reason that image segmentation does not reliably work to cluster BMUs in sparsely-matched trained SOMs is because these approaches assume (correctly, for actual images) that the image *is* the data, and clustering is performed by repeatedly averaging similar (at most 3-D) values of adjacent pixels of the image. However, a trained SOM is different from an image in that neurons typically have many more than three dimensions, and different neurons encode different amounts of information from the actual input data X. The higher the dimensions (M) in the input neurons, the more information that is likely to be lost when values of adjacent neurons are averaged, because dissimilar dimensions are averaged to the same degree as similar dimensions. And since the actual input data X are not utilized during the image segmentation process, the information in the map becomes further and further removed from the input data values as segmentation proceeds. In contrast, the CR phase updates all neurons to become closer to the actual input data X and updates each dimension by different amounts, depending on their similarity to the input data. Consequently, the

information encoded in C becomes closer and closer to the input data as the CR phase proceeds. After the CR phase has converted the continuously interpolated W map into the piecewise discontinuous C map that respects all of the input data X, one can perform hierarchical visualization of the discontinuities with the B-matrix, as illustrated in the next section.

It is worth noting that image segmentation of C produces much better clustering than image segmentation of W; however, it still does not necessarily achieve the optimal clusters for a given level of clustering, requires additional computation for agglomeration of neurons, and requires the user to prespecify the degree of clustering.

V. Example Application: Clustering Microbial Data Around a Landfill

In order to explore how microbial communities may act as indicators for the gradient of contamination in groundwater, Pearce *et al.* [5] rigorously compared several clustering approaches of microbial community profile data from 22 monitoring wells around the leaking Schuyler Falls Landfill in Clinton, NY [Fig. 4(a)]. The authors preprocessed the 209 measured microbial variables using both parametric and nonparametric (using a Spearman's rank correlation matrix) principal component analysis (PCA) to reduce the feature set to the top 21 principal components (PCs), which together explained 100% of the variance. They then compared hierarchical, K-means, SOM, and weighted SOM clustering approaches on these data, and assessed the validity of the resulting clusters using an F-statistic [5] based on a nonparametric MANOVA [19]. In their detailed analysis, the nonparametric PCA followed by the weighted SOM was shown to be the most appropriate clustering method for this data set (which violates parametric assumptions of independence, normality, and equal variance), and provided the best overall match to the independent expert classifications given in [16] that were based on detailed hydrochemistry data. Specifically, in the nonparametric SOM, the $L = 22$ vectors of $M = 21$ normalized Spearman rank PC values were used to train a 20×20 nontoroidal rectangular SOM, where updates for each component plane were weighted by the percent variance explained by the corresponding PCs. Results of this sparsely-matched SOM were visualized using a heat map of the similarly weighted nontoroidal U-matrix, superimposed with visually-approximated, hand-drawn cluster boundary lines. In the remainder of this section, we show how the CR phase and B-matrix automate the detection and visualization of the cluster hierarchy already embedded in this trained SOM. The reader is referred to [5] to see how the clusters resulting from this SOM compare to those of the other clustering methods.

In Fig. 4(b) we show a heat map of the U-matrix computed from the trained W matrix (this is the same data as shown in [5, Fig. 3(a)], but without the hand-drawn cluster boundaries), and in Fig. 4(d) we superimpose a contour map of this U-matrix over a heat map of the first component plane of W (corresponding to the first PC). It is difficult to make out which wells cluster together from either of these visualizations, especially near the edges of the map. We applied the proposed CR phase (using $\sigma_0 = 0.29$, approximately one eighth of the maximum interneuron Euclidean distance in W of 2.26) to the trained weight matrix W from [5]. To be consistent with [5], we used a weighted Euclidean distance metric and weighted CR updates, where the importance weight vector p in Algorithm 1 was set to the percent variance explained by each of the 21 PCs, and a nontoroidal grid. A heat map of the B-matrix computed from the resulting cluster-reinforced C map is shown in Fig. 4(f), where one can readily identify clusters at various levels of separation, based on the intensity of the color. For comparison, we show the B-matrix computed directly from W in Fig. 4(c) showing that prior to the CR phase the boundaries between clusters are quite diffuse. After the CR phase, the U-matrix visualization is also improved [Fig. 4(e)], although clusters are still not as clearly defined as with the B-matrix [Fig. 4(f)]. In Fig. 4(b)–(f), the numbered dots indicate the locations of the BMUs in W, for each of the PC input vectors for the 22 numbered wells in Fig. 4(a).

Viewing all of the interneuron distances stored in the B-matrix as a heat map [Fig. 4(f)] shows the varying degrees-of-separation in the natural clustering hierarchy embedded in the SOM. To interactively display only specific levels in this cluster hierarchy, one could brighten or darken the associated colormap. Alternatively, hierarchical visualization of clusters can be shown on top of the heat maps of component planes, as previously suggested, by superimposing grid lines with thickness proportional to B-values above a user-specified minimum display threshold θ. We illustrate this approach on the first component plane of W, for minimum display thresholds θ of 0.92, 0.88, and 0.60, respectively, showing two, three, and eight clusters of wells [Fig. 5(a)–(c)]. The cluster membership for the two-cluster visualization [Fig. 5(a)] corresponds to wells identified as contaminated and noncontaminated, and shows that wells near the fringe of the contamination plume are more similar to contaminated wells than to uncontaminated wells [see Fig. 4(a)]. The cluster membership for the three-cluster visualization [Fig. 5(b)] corresponds to the most contaminated wells, wells near the fringe of the contamination plume, and uncontaminated wells [see Fig. 4(a)]. These two- and three-cluster memberships are identical to the best clusterings found in [5], and are consistent with the hydrochemistry-determined classification given in [16]. Both the nonparametric F-statistic (from [5]) and the Silhouette values (where the normalized input data are weighted by p prior to computing these cluster validation metrics) indicate that this data are more appropriately clustered into three groups ($F = 10.47$, $mean(s) = 0.43$, all $s > 0$) than into to two groups ($F = 8.08$, $mean(s) = 0.27$, $\min(s) = -0.15$). As the display threshold θ is lowered further, the fringe cluster rapidly breaks apart into individual wells [Fig. 5(c)], reflecting the fact that these wells have the most heterogeneity in microbial profiles [16] and reinforcing the conclusion in [5] (based on the F-statistic for several clustering methods) that more than three clusters are probably not meaningful for this data. Although one could use a cluster validation metric, such as the F-statistic or mean Silhouette value, to determine where to optimally set θ, this example illustrates

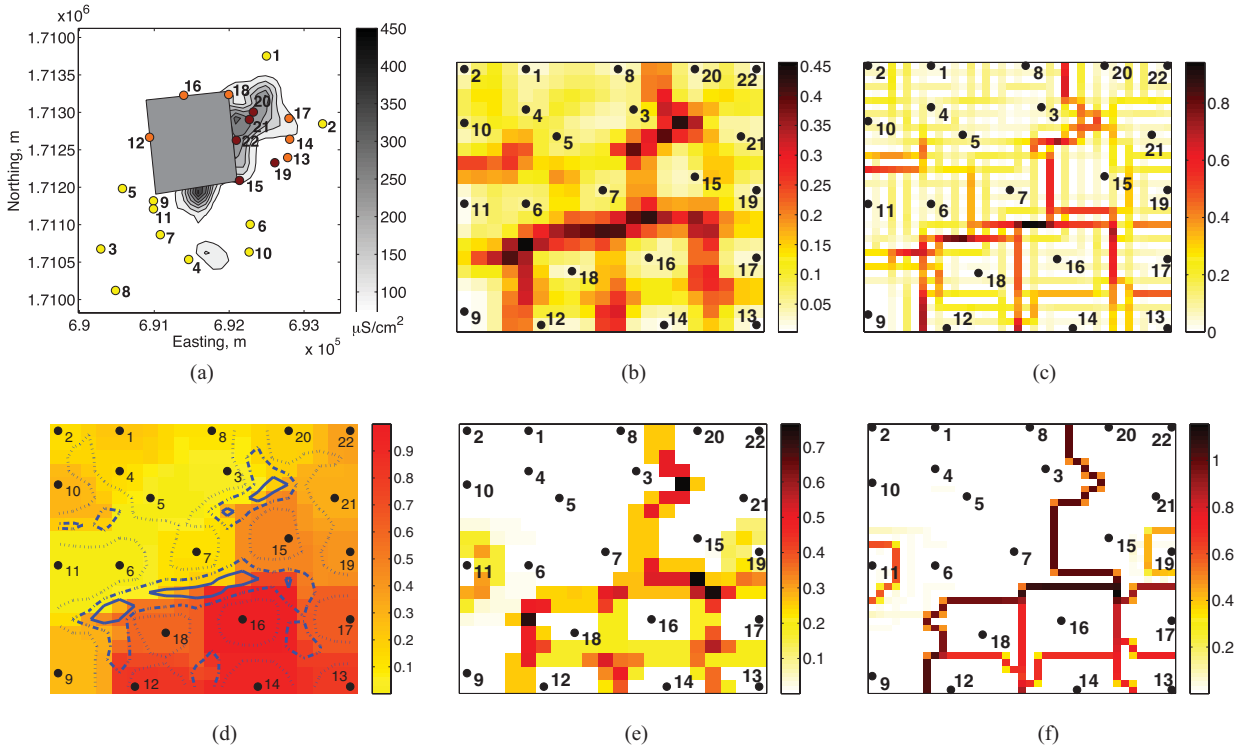


Fig. 4. Results on landfill data. (a) Locations of the numbered wells (colored dots) relative to the landfill (gray square) and plume of contamination (contour plot) estimated from conductivity measurements, collected from surface electromagnetic surveys (EM-34) and interpolated using the method of ordinary kriging. The wells are color-coded to show the three clusters of Fig. 5(b), where dark red indicates the most contaminated wells, orange indicates fringe wells, and yellow indicates uncontaminated wells. (b) Heat map of the U-matrix computed from W. (c) Heat map of the B-matrix computed from W. (d) First component plane of W overlain with a three-level contour plot of the U-matrix computed from W (contours at U-values of 0.34, 0.23, and 0.12). (e) Heat map of the U-matrix computed from C. (f) Heat map of the B-matrix computed from C.

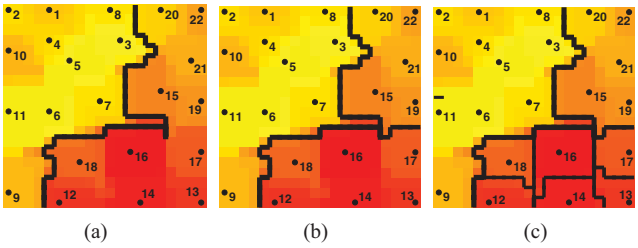


Fig. 5. Results on landfill data. (a)–(c) Heat maps of the first component plane of W. The thicknesses of the grid lines correspond to B-values computed from the interneuron distances in C (as in Fig. 4). B-values are only displayed above minimum distance threshold levels of (a) $\theta = 0.92$, (b) $\theta = 0.88$, and (c) $\theta = 0.60$.

that interactive visualization of B-values above different thresholds can be informative even at levels of nonoptimal clustering. Visualizing the remaining 20 component planes of W overlain with the B-matrix as grid lines reveals additional sources of within- and between-cluster heterogeneity, although space does not permit us to show them here.

VI. Conclusion

SOMs have proven to be useful tools for clustering and visualization of high-dimensional data. Nevertheless, it was often challenging to identify clusters in sparsely-matched feature maps, where neuron interpolation between BMUs can result in diffuse cluster boundaries. In this brief, we introduced an additional CR phase, to be run after the SOM self-organizing and convergence phases, for sharpening boundaries between existing self-organized clusters of BMUs. This cluster sharpening proceeds in an unsupervised data-driven manner, without prior knowledge of the domain or of the number of clusters to be identified. By iteratively performing synchronous updates within neighborhoods based on distances between input vectors and all neurons in the map, the CR phase reduces within-cluster differences, and thus sharpens boundaries between clusters. Discontinuities in the resulting map can then be interpreted as between-cluster distances using the proposed B-matrix. The B-matrix can be directly displayed on the heat maps of component planes using grid lines with thicknesses corresponding to the distances between adjacent clusters in the SOM. By thresholding the lower bound of the displayed lines, one obtains hierarchical control of the visual level of cluster resolution, without having to further cluster the data beyond what was already accomplished by the self-organization during SOM training. The proposed methods were demonstrated using the classic 13-D binary-valued animal SOM benchmark problem and a 21-D real-valued microbial profile data set. MATLAB code for the proposed methods is provided online [20].

Acknowledgment

The authors would like to thank A. Pearce and P. Mouser for graciously sharing the microbial profile data.

REFERENCES

[1] T. Kohonen, "The self-organizing map," *Proc. IEEE*, vol. 78, no. 9, pp. 1464–1480, Sep. 1990.

[2] T. Kohonen, *Self-Organizing Maps*, 3rd ed. New York: Springer-Verlag, 2001.

[3] K. Tasdemir and E. Merényi, "Exploiting data topology in visualization and clustering of self-organizing maps," *IEEE Trans. Neural Netw.*, vol. 20, no. 4, pp. 549–562, Apr. 2009.

[4] K. Tasdemir, P. Milenov, and B. Tapsall, "Topology-based hierarchical clustering of self-organizing maps," *IEEE Trans. Neural Netw.*, vol. 22, no. 3, pp. 474–485, Mar. 2011.

[5] A. Pearce, D. Rizzo, and P. Mouser, "Subsurface characterization of groundwater contaminated by landfill leachate using microbial community profile data and a nonparametric decision-making process," *Water Resour. Res.*, vol. 47, no. 6, pp. W06511-1–W06511-11, 2011.

[6] L. Besaw, D. Rizzo, M. Kline, K. Underwood, J. Doris, L. Morrissey, and K. Pelletier, "Stream classification using hierarchical artificial neural networks: A fluvial hazard management tool," *J. Hydrol.*, vol. 373, nos. 1–2, pp. 34–43, 2009.

[7] M. Kraaijveld, J. Mao, and A. Jain, "A nonlinear projection method based on Kohonen's topology preserving maps," *IEEE Trans. Neural Netw.*, vol. 6, no. 3, pp. 548–559, May 1995.

[8] D. Merkl and A. Rauber, "Alternative ways for cluster visualization in self-organizing maps," in *Proc. Workshop Self-Organizing Maps*, 1997, pp. 1–7.

[9] J. Moehrmann, A. Burkovski, E. Baranovskiy, G. Heinze, A. Rapoport, and G. Heidemann, "A discussion on visual interactive data exploration using self-organizing maps," in *Proc. Adv. Self-Organ. Maps*, 2011, pp. 178–187.

[10] A. Ultsch, "Self-organizing neural networks for visualization and classification," in *Proc. Conf. Soc. Inf. Classificat.*, 1993, pp. 307–313.

[11] A. Ultsch, *U*-Matrix: A Tool to Visualize Clusters in High Dimensional Data*. Marburg, Germany: Fachbereich Mathematik Informatik, 2003.

[12] D. Brown, I. Craw, and J. Lewthwaite, "A SOM based approach to skin detection with application in real time systems," in *Proc. British Mach. Vis. Conf.*, vol. 2. 2001, pp. 491–500.

[13] J. Wiggins, S. Peltier, S. Ashinoff, S. Weng, M. Carrasco, R. Welsh, C. Lord, and C. Monk, "Using a self-organizing map algorithm to detect age-related changes in functional connectivity during rest in autism spectrum disorders," *Brain Res.*, vol. 1380, pp. 187–197, Mar. 2011.

[14] H. Ritter and T. Kohonen, "Self-organizing semantic maps," *Biol. Cybern.*, vol. 61, no. 4, pp. 241–254, 1989.

[15] R. Nock and F. Nielsen, "Statistical region merging," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 11, pp. 1452–1458, Nov. 2004.

[16] P. Mouser, D. Rizzo, G. Druschel, S. Morales, P. O'Grady, N. Hayden, and L. Stevens, "Enhanced detection of groundwater contamination from a leaking waste disposal site by microbial community profiles," *Water Resour. Res.*, vol. 46, no. W12506, pp. 1–12, 2010, DOI: 10.1029/2010WR009459.

[17] J. Vesanto, J. Himberg, E. Alhoniemi, and J. Parhankangas, "Self-organizing map in MATLAB: The SOM toolbox," in *Proc.* MATLAB *DSP Conf.*, vol. 1999. 1999, pp. 35–40.

[18] L. Kaufman and P. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*, vol. 39. New York: Wiley, 1990.

[19] M. Anderson, "A new method for non-parametric multivariate analysis of variance," *Austral Ecol.*, vol. 26, no. 1, pp. 32–46, Feb. 2001.

[20] *Cluster Reinforcement (CR) Phase* [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/35538