Methodology for Emulating Self Organizing Maps for Visualization of Large Datasets

Macario O. Cordel II and Arnulfo P. Azcarraga
College of Computer Studies
**Corresponding Author: macario.cordel@dlsu.edu.ph*

Abstract: The self-organizing map (SOM) methodology does vector quantization and clustering on the dataset, and then projects these clusters in a lower dimensional space, such as 2D map, by positioning similar clusters in locations that are spatially closer in the lower dimension space. This makes the SOM methodology an effective tool for data visualization. However, in a world where mined information from big data have to be available immediately, SOM becomes an unattractive tool because of its space and time complexity. In this paper, we propose an alternative visualization methodology for large datasets with clustering information without the speed and memory constraints inherent to SOM. To demonstrate the efficiency and the vast potential of the proposed scheme as a fast visualization tool, the methodology is used to cluster and project the 3,823 image samples of handwritten digits of the Optical Recognition of Handwritten Digits dataset.

Key Words: Data visualization; self-organizing map; multidimensional scaling; two-level clustering; fast data analysis; positions of clusters

1. Introduction

One of the enablers of Big data is the intuitive presentation of information such as data visualization (ITU Telecommunication Standardization Bureau, 2013). Visualization provides intuitive display of unstructured information e.g. emails, text messages, audio as well as video streams. These types of unstructured data continuously grow requiring visualization tools to have more efficient running performance. One of these visualization tools is the Self-Organizing Map (SOM).

SOM represents data using nodes as points in the two-dimensional (or three-dimensional) vector space. These SOM nodes have weight vectors which are updated per iteration depending on the input vector from the data set. Generally, the weight vectors are updated as follows.

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) + G(t)\boldsymbol{\alpha}_i(t) \,\|\, \mathbf{x}(t) - \mathbf{w}_i(t) \,\| \qquad \text{(Eq. 1)}$$

where t represents the iteration number, $\mathbf{w}_i$ represents the weight vector of the ith node, $\mathbf{x}$ is the input vector chosen randomly from the training set, $\boldsymbol{\alpha}_i(t)$ is the learning rate of the adaptation process, $G(t)$ is a window function which is typically a Gaussian window or a rectangular window, and $\|\mathbf{x}(t) - \mathbf{w}_i(t)\|$ is the Euclidean distance between $\mathbf{x}(t)$ and $\mathbf{w}_i(t)$. The intuitive display of the data's relative distance, distribution and clusters make SOM an attractive tool for data visualization. However, for large dataset, Eq. (1) has to be performed several times, increasing SOM's complexity.

For N' number of SOM nodes with M weights per node, the computational requirement per iteration per node is **O**(N'^2×M) for distance computation, **O**(N'^2×Mlog$M$) for winning node selection, and **O**(N'^2×M) for weight update computation using a Gaussian window. That is, for large amount of samples N, the complexity becomes **O**(N×N'^2×M).

An alternative simpler data visualization tool, called the Multidimensional Scaling (MDS), makes use of singular-value decomposition for data mapping to remove the need for iteration which is highly based on the number of samples. The MDS reveals the structure of a data set, typically high dimensional data, by transforming the pairwise dissimilarities of each element (in the data set) into distances in low dimensional vector space (Torgeson, 1958), (Cox & Cox, 2001), (Bartholomew, Steele, Moustaki, & Galbraith, 2008). Recent works (Shang, Wheeler, Zhang, & Fromherz, 2004), (Cheung & So, 2005), (So & Chan, 2009), (Costa, Patwari, & Hero, 2006) on wireless sensor nodes (WSN) make use of MDS on node localization problem where only the nodes' receive signal information are known. Despite of its applicability to complex problems, e.g. in marketing and wireless networks, it lacks clustering and distribution information which make it ineffective data visualization tool.

In this work, we present an alternative data visualization methodology to overcome the complexity issue of the SOM in large number of samples and the limited information provided by the MDS as a projection tool. This proposed scheme is discussed in section 2 in details. To demonstrate its vast potential as a visualization tool, an experiment is performed using the Optical Recognition of Handwritten Digits dataset (Bache & Lichman, 2013). Results and analysis of which are presented in section 3. Finally, conclusion and future works are provided in section 4.

2. LARGE DATA VISUALIZATION METHODOLOGY

Consider a large database of M-dimensional data with N samples whose attribute vector is denoted by $\phi_i^{(M)}$, where i = 1, 2, …, N. The relative Euclidean distance measurement between two data entries i and j of the given data set is given by

$$D=[d_{ij}]=|||\phi_i^{(M)}-\phi_j^{(M)}|| \qquad \text{(Eq. 2)}$$

where ||•|| denotes the Frobenius norm. Applying classical MDS for large value of M requires M×M memories e.g. 10^{10} for $N = 10^5$ data. Applying SOM, similarly, is impractical. The task is to provide mapping of N high-dimensional in $R^{(M)}$ onto a low-dimensional vector space, e.g. $R^{(2)}$ while providing the clustering information and data distribution.

The proposed scheme is designed to emulate SOM by providing data proximity and clustering information. It is mainly divided into three phases: (1) data summarization into prototypes, (2) clustering of prototypes and (3) data mapping, as shown in Figure 1. The first phase aims to decrease the number of data samples, N, into smaller number of *prototypes*, N', by performing k-means on the large dataset. Since N' equals the number of prototypes, then N' equals the number of clusters in this application of k-means. The second phase performs prototype clustering to introduce this information in data mapping. For supervised learning, the number of clusters, called the small k, is usually set to be equal to the number of actual classes in the data. To distinguish k of phase 1 k-means from k of phase 2 k-means, the former is called big K (which is equal to N') while the latter is called small k. Finally, phase 3 performs the projection of the clustered prototypes onto a lower dimensional space, e.g. 2-dimensional (2D) or 3-dimensional (3D) space, for visualization.

2.1 Phase 1: The first level k-means for vector quantization

One of the requirements of effective data visualization is to provide compact representation of a dataset. This is the objective of the first phase

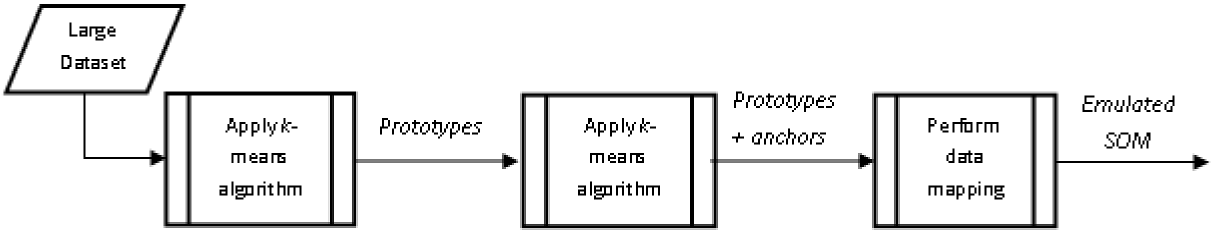

Fig 1. Block diagram of the proposed methodology.

which performs vector quantization to decrease the dataset size by quantizing similar data to their respective representative attribute vectors, called in this work as the *prototypes*. This phase can essentially be accomplished thru Expectation-Minimization (EM) algorithms and vector quantization methods. In the simple implementation of proposed methodology, we use k-means, also known as Lloyd's algorithm (Lloyd, 1982), to convert the large dataset with N samples into a smaller set of N' prototypes such that $N' << N$. These prototypes are nothing but the centroids of the N' clusters that are formed by ordinary k-means, setting the number of clusters to N'.

To aid in the discussion, the clusters formed in the first application of k-means are called "big-K" clusters and the centroids of these big-K clusters formed, as mentioned, are called the prototypes. As such, the value of N is "big-K", denoted by the capital K, and this corresponds to the size of a SOM if the SOM methodology were used. To illustrate, if the SOM would have been a 20×20 map, then K in this approach would be set to 400.

In summarizing the data, the distribution of the original dataset must be reflected by the distribution of the prototypes. We attempt to achieve this by choosing randomly the initial values of the prototypes from the large dataset such that for some sufficiently large K, the initial distribution of K prototypes reflects the distribution of the original dataset (Dinov, Christou, & Gould, 2009). Discussion on whether the distribution of the sample prototypes after first application of k-means algorithm reflect the distribution of the actual large dataset or not, is beyond the scope of this paper. The value of big-K, however, in terms of emulating the SOM methodology, is nothing but the number of nodes in a SOM.

2.2 Phase 2: The second level of k-means for prototype clustering

Phase 2 involves a second application of the k-means algorithm. This time, the input is no longer the large original dataset, but just the smaller set represented by the K prototypes. In this work, the clusters of K prototypes formed in this level are called the "small-k", denoted as k, clusters and the k centroids are referred to as the *centroids*, to distinguish them from the K prototypes from phase 1. The second level k-means of phase 2 performs the same initialization process for anchors but does not aim to reflect the dataset distribution, rather to provide clustering information of the actual datasets via the K prototypes.

To recapitulate, the number of anchors is less than the number of prototypes, which is in turn much less than the number of original samples in the dataset. We have $N \gg K > k$.

2.3 Phase 3: Anchor projection mapping via Multidimensional scaling

Phase 3 transforms the high-dimensional prototypes into 2D representation for visualization via Multidimensional scaling (MDS). Let $\mathbf{\Phi}_K \in \mathrm{R}^{(M)}$ be the set of K prototype vectors. We consider the two-dimensional vector space, R^2, for data mapping as we try to project K prototypes into $\mathbf{X} \in \mathrm{R}^2$ via MDS. Furthermore, let $\mathbf{D}$ be the pairwise distance of $\mathbf{\Phi}_K$, applying Equations (3), (4) and (5) on $\mathbf{D}$ gives the following expressions for the location of data $\mathbf{X} = (x, y)$ on the Cartesian plane.

$$\mathbf{B}_{K\times K} = -0.5\mathbf{J}_{K\times K}\mathbf{D}^2_{K\times K}\mathbf{J}_{K\times K} \qquad \text{(Eq. 3)}$$

$$\mathbf{B}_{K\times K} = -0.5\mathbf{U}_{K\times K}\mathbf{S}_{K\times K}\mathbf{U}^T_{K\times K} \qquad \text{(Eq. 4)}$$

$$\mathbf{X}_{K\times 2} = \mathbf{U}_{K\times 2}\mathbf{S}^{1/2}_{2\times 2} \qquad \text{(Eq. 5)}$$

where $\mathbf{J} = \mathbf{I} - n^{-1}\mathbf{1}\mathbf{1}^T$ and $\mathbf{D}^2 = [\mathrm{d}^2_{ij}]$. The K Cartesian coordinates of $\mathbf{X}$ are the corresponding coordinates of K prototypes. For large dataset of size N, the distance matrix requires $N\times(N-1)$ memories. Thus, mapping the prototypes to 2D via MDS would only be feasible after the application of the first level k-means, i.e. after data summarization into prototypes.

3. EXPERIMENT RESULTS AND ANALYSIS

The proposed methodology for large dataset visualization was evaluated using the training dataset from the Optical Recognition of Handwritten Digits dataset (Bache & Lichman, 2013). The dataset contains 3,823, handwritten digits samples which were taken from 32×32 bitmaps of handwritten digits images and downsampled into 8×8 images. Thus, each sample has 64 attributes with integer values from 0 to 16 and label from 0 to 9. Because it will require large amount of memory to map the whole database on a Cartesian plane, the large amount of samples were first compressed into manageable amount of data via the first level k-means (phase 1). That is, the 3,823 handwritten samples were compressed into 400 (K = 400) prototypes, corresponding for example to a 20×20 SOM.

Table 1. Distribution of each handwritten prototypes to different clusters

	Prototype digit 0	Prototype digit 1	Prototype digit 2	Prototype digit 3	Prototype label 4	Prototype label 5	Prototype label 6	Prototype label 7	Prototype label 8	Prototype label 9	Total prototypes per cluster
Cluster 0 ●	38	0	0	0	0	0	1	0	0	0	39
Cluster 1 ●	0	10	0	1	3	0	0	3	0	11	28
Cluster 2 ●	0	1	33	0	0	0	0	0	0	0	34
Cluster 3 ●	0	0	1	41	0	0	0	0	0	1	43
Cluster 4 ●	0	0	0	0	36	0	0	0	0	0	36
Cluster 5 ●	0	0	0	0	0	36	0	0	0	0	36
Cluster 6 ●	0	0	0	0	1	0	30	0	0	0	31
Cluster 7 ○	0	0	2	0	5	0	0	40	1	3	51
Cluster 8 ●	0	22	0	0	2	0	0	0	31	0	55
Cluster 9 ○	0	1	1	6	0	7	0	0	0	32	47
Total prototypes per digit	38	34	37	48	47	43	31	43	32	47	

The resulting prototypes were then clustered into 10 groups, corresponding to the ten decimal digits via the second level k-means (phase 2). This will hopefully provide clustering information when the prototypes are projected in 2D. Table 1 shows the 10 clusters formed after applying the second level k-means algorithm on the 400 prototypes. The first column corresponds to the clusters and their respective legend (which will be used in the 2D mapping) while the second to tenth columns correspond to the number of prototypes per cluster. For example, there are 38 prototype digits 0 and one prototype digit 6 in cluster 0. Similarly, there are 10 prototype digits 1, one prototype digit 3, three prototypes digit 4 and 11 prototypes digit 9 in cluster 1.

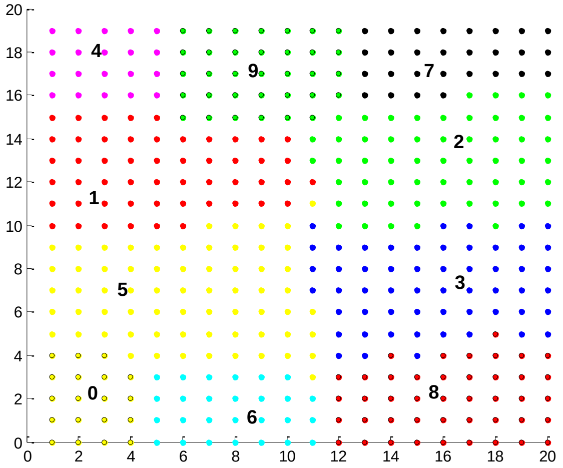

Fig 2. SOM solution as applied to the handwritten digits dataset. The colors and numbers are the clusters and label of each group, respectively.

The last step in performs MDS for data projection onto the 2D plane. This phase aims to emulate the SOM display as shown in Figure 2. The SOM plot provides both clustering information and relative distance information between clusters. For example, the handwritten digits with similar strokes are positioned adjacent to each other, e.g. clusters 8 and 3, and 9 and 4.

For the emulated SOM display, shown in Figure 3, clusters with similar strokes are positioned relatively near each other as well, such as the clusters “3”, “9”, “2” and “0” whose upper portion of the prototypes have similar strokes. Furthermore, clusters “0” and “6” which have similar curvy strokes on the left portion of the prototypes are located side by side in the 2D map. In contrast, cluster “6” is relatively far

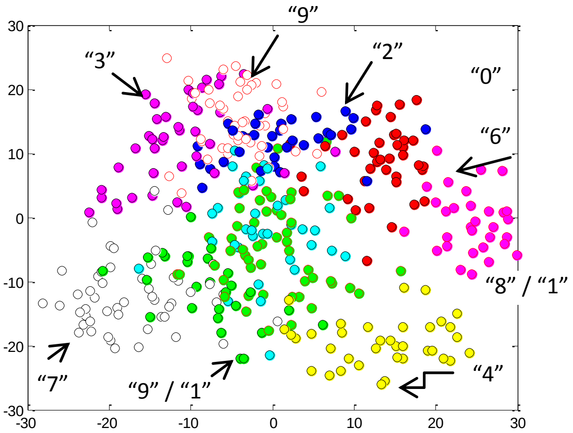

Fig 3. Emulated SOM’s relative data distance and cluster information. The arrows pointing a cluster indicate the majority of prototypes in that cluster

from clusters “2”, “9” and “3”, which despite of their proximity to “0”, clusters “2”, “9” and “3” have minimal resemblance with “6”.

Clusters with significantly the same number of prototypes, shown as cluster “8”/“1” and “9”/“1”, detailed portions of the plot are provided in Figures 4 and 5 to determine which prototypes are actually near each other. Figure 4 shows that prototypes “1” and “4” are actually near each other. These prototypes have the same vertical strokes at the right portion of the digit. Similarly, Figure 5 shows that prototypes “7” and “9”, both with diagonal downward stroke, are also near each other.

4. CONCLUSIONS

We presented an alternative scheme which emulates SOM as a visualization tool for large dataset. The proposed approach summarizes the large dataset first via k-means algorithm. To include clustering information in the visualization, second k-means is applied to the summarized data, called the prototypes. These prototypes are then projected to the 2D map via the application of MDS. Similar to SOM, the emulated SOM was able to provide cluster information and similarity distance of the prototypes as shown in Figures 2 - 5. With N being the number of samples and M being the number of attributes, the proposed scheme complexity for large N is O(NM) as compared to SOM which has O(N^2M). Future considerations include addressing the limitation of MDS on higher dimensional data (curse of dimensionality).

6. REFERENCES

Azcarraga, A., Hsieh, M.-H., Pan, S.-L., & Setiono, R. (2008). Improved SOM Labeling Methodology for Data Mining Applications. *Soft Computing for Knowledge Discovery and Data Mining*, 45-75.

Bache, K., & Lichman, M. (2013). UCI Machine Learning Repository. University of California, Irvine, School of Information and Computer Sciences. Retrieved from http://archive.ics.uci.edu/ml

Bartholomew, D. J., Steele, F., Moustaki, I., & Galbraith, J. I. (2008). Multidimensional Scaling. In *Analysis of Multivariate Social Science Data* (pp. 55-81). Chapman and Hall/CRC.

Cheung, K. W., & So, H. C. (2005). A multidimensional scaling framework for mobile location using time-of-arrival measurements. *Signal Processing, IEEE Transactions on, 53*(2), 460-470.

Costa, J. A., Patwari, N., & Hero, A. I. (2006). Distributed weighted-multidimensional scaling for node localization in sensor networks. *ACM Transactions on Sensor Networks (TOSN), 2*(1), 39-64.

Cox, T. F., & Cox, M. A. (2001). *Multidimensional Scaling (2nd Ed.).* London: Chapman and Hall/CRC.

Demmel, J., Dumitriu, I., & Holtz, O. (2007). Fast linear algebra is stable. In *Numerische Mathematik* (pp. 59-91). Springer.

Dinov, I. D., Christou, N., & Gould, R. (2009). Law of Large Numbers: the Theory, Applications and Technology-based Education. *Journal of Statistics Education*.

ITU Telecommunication Standardization Bureau. (2013). *Big Data: Big today, normal tomorrow.* Technical Report, Geneva.

Lloyd, S. P. (1982). Least square quantization in PCM. *IEEE Transactions on Information Theory.*

Shang, Y., Wheeler, R., Zhang, Y., & Fromherz, M. (2004). Localization from Connectivity in Sensor Networks. *Parallel and Distributed Systems, IEEE Transactions on, 15*(11), 961-974.

So, H. C., & Chan, F. K. (2009). Efficient weighted multidimensional scaling for wireless sensor network localization. *Signal Processing, IEEE Transactions on, 57*(11), 4548-4553.

Torgeson, W. S. (1958). *Theory and Methods of Multidimensional Scaling*. New York: John Wiley & Sons.

Young, F. W. (2013). *Multidimensional Scaling: History, Theory and Applications.* Psychology Press.