

Distance Metrics for Time-Series Data with Concentric Multi-Sphere Self Organizing Maps

Tom Gedeon, Lachlan Paget, and Dingyun Zhu

Research School of Computer Science, College of Engineering and Computer Science,
The Australian National University, Canberra, ACT 0200, Australia
{tom.gedeon,dingyun.zhu}@anu.edu.au, lachlan.paget@gmail.com

Abstract. Self-Organizing Maps have been shown to be a powerful unsupervised learning tool in the analysis of complex high dimensional data. SOMs are capable of performing topological mapping, clustering and dimensionality reduction in order to effectively visualize and understand data and it is desirable to apply these techniques to time-series data. In this project a novel approach to time-series learning using Concentric Multi-Sphere SOMs has been expanded and generalized into a unified framework in order to thoroughly test the learning capabilities. It is found that Quantization and Topological Error are not suitable to test the learning performance of the algorithms and it is suggested that future work focus on developing new error measures and learning algorithms.

Keywords: Spherical SOM, time series, topological error, quantization error, concentric multi-sphere SOM (CSM-SOM).

1 Introduction

Self-Organizing Maps (SOMs) are primarily used for clustering, classification, sampling and visualizing high dimensional data [1]. This technique has been widely applied in many ways, for instance clustering high-frequency financial data. The conventional neighborhood arrangements are planar SOM made of two-dimensional rectangular or hexagonal lattices. However, the planar SOM has a disadvantage which is the “border effect” [2]. During training, the neurons compete with others. The weight of the winning neuron and its neighbor are updated. Ideally, all the units have the same chance to be updated. However, in the planar map, the units at the border of the map have fewer neighbors than the inside units. At the end of training, the map may not form expected similar regions of the data space, since there are many units with unequal chances of being modified during training [3]. Therefore, many spherical SOMs were proposed in order to solve that problem.

The motivation of this research is to provide a method that users can use an arbitrary number of spheres, and to observe the results of clustering data as well as the quality of SOMs on multiple spheres. In this paper, we consider the Spherical SOM introduced in [4] as the base for our Concentric Multi-Spherical SOM (CMS-SOM) topology.

2 Spherical SOM (SSOM)

Traditional Kohonen Self-Organizing Maps feature a non-uniform neighborhood structure in which neurons on the periphery of the map have fewer neighbors than those in the interior. This topological asymmetry is known as the border effect and results in edge neurons suffering from a reduced probability of adaption and consequently their weight vectors “collapse into the center of the input space” [5]. A discussion of the various approaches to avoid this problem can be found in [6] which includes the solution of joining the left, right and top, bottom map edges to effectively create a toroidal map. The unfortunate consequence of toroidal maps is that they are difficult to visualize effectively in three-dimensions.

Spherical Self-Organizing Maps (SSOM) were first introduced by Ritter [7], and have the ability to eliminate the border effect [8] and can also be readily and easily visualized in 3D. SSOM, like traditional SOM have been shown to be effective tools of data analysis and visualisation in a diverse range of fields including medical [9] and financial [10]. Several approaches have been proposed for the construction of the of the spherical feature map. Sangole and Leontitsis [4] proposed a method using recursive subdivision of the Icosahedron to produce an approximately uniform distribution of neurons on a sphere. The size of the data structure used to store the adjacency information scales poorly with the sphere size (number of recursive subdivisions) but the lookup time is improved as a result [6]. Pre-processing of the adjacency matrix can be used as a work-around but this implementation still suffers from the inability to create spheres with an arbitrary number of neurons. The number of neurons N is shown below as an exponential function of the number of recursive subdivisions R of the Icosahedron: $N = 2 + 10(4^R)$.

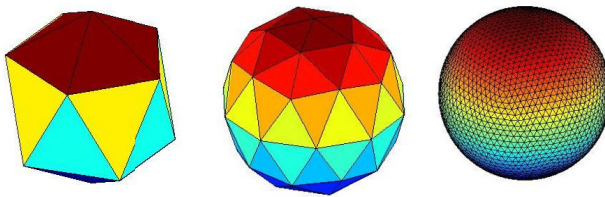


Fig. 1. Icosahedron subdivided 0, 1, 4 times

An alternative approach to the generation of spheres which can produce arbitrary number of neurons is to place neurons equispaced on a helix that sits on the surface of a sphere [3]. The identification and requirement of uniform neuron separations however becomes significantly more complicated than in the previous case. A third approach was shown by [6] where the Icosahedron method was used again but with a reduced time and space complexity for lookup within and creation of the neighborhood adjacency data structure.

2.1 Error Measures

Two distinct but complementary error measures are introduced here to quantify the quality of a SOM. Error measures for use in SOM are discussed more completely by [23] and [24].

The Quantisation Error (QE) represents the average difference between each pattern vector x_j and its best matching unit m_c and is calculated as follows.

$$QE = 1/n \sum_{j=1}^n \|m_c - x_j\|$$

This measure does not however take into account the quality of the topological mapping for which a second error measure, Topological Error (TE) is used.

$$TE = 1/n \sum_{j=1}^n \varphi(x_j, m_1, \dots, m_p)$$

where $\varphi(x_j, m_1, \dots, m_p) = \begin{cases} 1, & \text{if 1st closest neighbor maps to 2nd closest} \\ 0, & \text{otherwise} \end{cases}$

3 Concentric Multi-Sphere SOMs and Connection Metrics

A key component of our approach is that the multiple spheres are concentric. This does not in general mean that inner spheres are smaller than outer spheres as would be the case for physical spheres. Rather, we consider that each neuron on a sphere has a nominal equivalence with “the same” neuron on each of its adjacent spheres. Notionally, each sphere in sequence is considered to be similar to next sphere which is at the next quantum of time distant. We can consider “space” and “time” to be topologically equivalent, which allows us to represent time via our concentric spheres. Our approach allows a number of possible connection metrics.

3.1 Self Metric

The ‘Self’ metric defines a sphere connection scheme where each neuron on each sphere is additionally connected to the equivalent neuron on some or all of the other spheres. Each inter-sphere connection is of length 1 regardless of which sphere maps to which. It will be seen that it is effective to conceptualize these spheres as sitting in a sequential order. Alternatively the arrangement could be thought of as multiple concentric spheres.

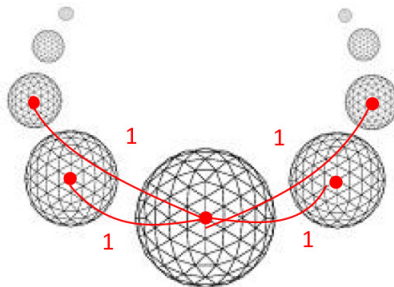


Fig. 2. Self Metric: 4 connected spheres with connection distances 1

In this metric the number of spheres connected is a parameter. For example in figure 2, four spheres are connected with two on each side. Alternatively this parameter could be specified as a fraction of the remaining spheres and rounded down to the nearest even number spheres. In this way a neuron can connect to as little as 2 spheres or potentially all spheres.

3.2 Distance Metric

On a single sphere it has often been decided to limit the maximum initial neighborhood size to half the sphere or one hemisphere [4]. The ‘Distance’ metric proposes that a virtual distance can be imagined to separate spheres. It should be noted that any neurons connected directly will always have a separation distance of 1 in terms of the SOM algorithm however the use of a virtual distance between spheres can be used to provide a more gradual tapering of the inter-sphere neighborhoods. This scheme is visualized in figure 3.

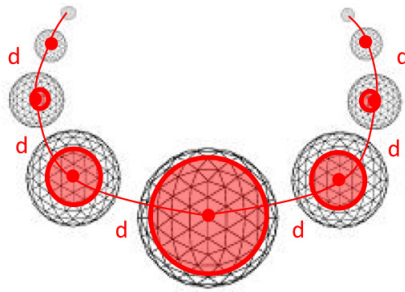


Fig. 3. Distance Metric: Connections of length d . Red circles indicate the neighborhoods of the center node.

The virtual distance d between spheres can be set such that the distance from a given node to the edge of the hemisphere should be the same as the distance travelled around half of the spheres. This is equivalent to $d = 2R/N_{sph}$. Here R is the distance (number of neurons) to the edge of the hemisphere and N_{sph} is the number of spheres. This metric is proposed to give a smooth decay in inter-sphere neighborhood connectivity.

3.3 Integer Metric

This is a simplified form of the distance metric which uses a constant value for d , and was our first notion for connecting spheres – with a ‘time’ distance of 1 equal to a ‘space’ distance of one. This metric can have a problem of ‘wrapping around’ if the spheres are in a cycle or not decaying fast enough if the spheres are merely linearly organised.

3.4 Data Order and Sphere Targeting

Our objective is to compare a number of distance metrics and error measures on our concentric multi-sphere SOMs (CMS-SOMs). With multiple spheres and a data set we can identify an number of plausible training regimes.

Data order refers to the presentation order of the input data set and can be randomised or preserved. For a time-series data set, we would expect better results if the data order is preserved if we learn useful time-based features in our model.

Sphere targeting refers to the restriction that the best matching neuron must be selected from a specific sphere. In this case during training the sphere targeted is incremented in order as the patterns are presented. The other option for targeting is ‘any’ where all spheres are candidates to select a winning neuron. For time-series data, we would expect better results from ‘preserved, specific’ if consistent time-based features were extracted. On the other hand, ‘random, any’ would provide little evidence for good learning of time-based features but would demonstrate the general power of our approach.

4 Data Set – ECSH

The Easy Calm Stressful Hard (ECSH) dataset contains eye gaze and physiological data collected while the subject is reading some paragraphs of text which are from these four categories. This dataset is from a larger project [11-12] on computational models for stress.

The ECSH dataset has 7 input fields being the x,y co-ordinates of the eye gaze, left and right pupil diameters, electrocardiogram (ECG), galvanic skin response (GSR) and instantaneous blood pressure (BP).

Biopac ECG100C, Biopac GSR100C and Finapres Finger Cuff systems were used to take ECG, GSR and blood pressure recordings at a sampling rate of 1000 Hz. Eye gaze and pupil dilation signals were obtained using a Seeing Machines FaceLAB system with a pair of infrared cameras at 60 Hz.

The data was synchronized and the ECSH dataset consists of 3,461 pattern vectors corresponding to a sampling rate of 60Hz. The data is classified into the four categories being the nature of the text being read, and is in 4 contiguous blocks of patterns.

5 Experiment and Results

In this experiment each inter-sphere connection metric is used to train a range of numbers of spheres of different sizes and the Quantisation and Topological Errors are calculated for each combination of the data order and sphere targeting parameters.

Table 1. Shows the different cases of sphere numbers and sizes

No. spheres	Sphere structure	Neurons / sphere	Total neurons
1	$ICOSA_4$	2562	2562
4	$ICOSA_3$	642	2568
15	$ICOSA_2$	162	2430
61	$ICOSA_1$	42	2562
214	$ICOSA_0$	12	2568

Here $ICOSA_N$ refers to an icosahedron with N recursive subdivisions. We have chosen the structure and number of spheres so that the total number of neurons is as similar as possible in each case. Simulations were run for 50 epochs with the neighbourhood radius parameter R being 0.5. Error values are averaged over 10 runs.

5.1 Quantisation Error

The best result for QE on each sphere configuration in each metric is in bold. The best result for each sphere configuration (omitting self metrics) is also shaded in grey.

Table 2. QE – Integer metric

Data Order	Targeting	1 sphere	4 spheres	15 spheres	61 spheres	214 spheres
preserved	any	272.19	252.94	529.68	231.69	303.88
preserved	specific	563.82	292.80	624.78	1542.46	721.03
random	any	176.57	168.70	283.12	126.86	166.95
random	specific	204.12	183.40	721.81	3548.49	925.05

Random-any is clearly the best performing algorithm with lowest error for the 214 sphere case. (The best results for 61 spheres is random-any using self-metric with 4 connected spheres.) Random, specific performs quite poorly in general. In both cases with targeting of specific large errors are seen for the 61 sphere case..

Table 3. QE – Distance metric

Data Order	Targeting	1 sphere	4 spheres	15 spheres	61 spheres	214 spheres
preserved	any	188.18	274.19	412.80	673.87	626.99
preserved	specific	132.94	301.95	924.14	823.21	879.35
random	any	148.43	157.40	547.53	724.62	625.38
random	specific	179.99	245.15	895.46	8703.96	908.92

Here preserved, any and random any perform similarly indicating the data order is having little effect on learning. Disappointingly preserved, specific performs poorly again except in the case of a single sphere which is somewhat irrelevant as targeting is not very meaningful for a single sphere.

It is clear that Quantisation error has no way to measure time-based learning as it simply measures how closely the neurons map to the input vectors. It is proposed that random, any performs so well for the following two reasons. Firstly the random data presentation order may be beneficial to avoid any ‘forgetting’ problems if the data has some non-random structure. For example if the data is a constant value continually for most of the data set and only deviates in value for small subsection then randomly indexed may be beneficial to ensure the feature map trains on the different values at a more uniform rate. The second reason is that the specific targeting limits the feature

maps exposure to each data point as only a subset of the map is available for competition and adaptation. This will probably reduce QE or at least slow its decrease over time whereas ‘any’ or not targeting facilitates a faster training and more complete exposure to the input data.

This hypothesis is also supported by the fact that there seems to be somewhat of a general decrease in QE as the amount of inter-sphere connectivity increases, i.e. as we transition from the Self metrics to the Distance metrics.

5.2 Topological Error

Here significantly low TE values are encountered for smaller number of spheres in all cases but performance degrades with an increasing number of spheres. Neither of the good results for preserved-specific on 1 sphere and the random-specific result are a clear signal of successful learning of time-based features.

Table 4. TE – Integer metric

Data Order	Targeting	1 sphere	4 spheres	15 spheres	61 spheres	214 spheres
preserved	any	0.078	0.049	0.237	0.231	0.405
preserved	specific	0.067	0.035	0.341	0.612	0.989
random	any	0.078	0.044	0.226	0.245	0.416
random	specific	0.078	0.024	0.192	0.691	0.991

Table 5. TE – Distance metric

Data Order	Targeting	1 sphere	4 spheres	15 spheres	61 spheres	214 spheres
preserved	any	0.073	0.210	0.228	0.158	0.098
preserved	specific	0.096	0.130	0.227	0.279	0.553
random	any	0.077	0.226	0.214	0.135	0.096
random	specific	0.094	0.094	0.064	0.061	0.564

TE seems to perform quite well over a large range of cases with little of the severe degradation in performance seen previously.

As with QE there seems to be some level of improved performance for greater inter-sphere connectivity. It is unclear whether TE is a suitable measure to quantify time learning due to the uncertainty whether the learning algorithms presented here are strongly capable of learning time-series characteristics. We will investigate further.

6 Conclusion and Future Work

We have used our concentric multi-sphere SOM structure to show that neither the Quantization error nor the Topological error measures are well suited to demonstrating

the quality of learning of time-based features on an SOM structure designed for that goal. We have introduced a scheme of DataOrder X Targeting to demonstrate this. Of the four cases, random-any clearly demonstrates the (non-time-based) power of our algorithm, preserved-specific would show good time-based feature learning well aligned to our SOM structure, preserved-any would show time-based feature learning less well aligned to our SOM structure (time-based features are learnt, but would imply some alternative topology could achieve better results), while finally random-specific is a general control and has no particular useful meaning, so a good result here implies both a better topology and distance metric / learning algorithm should be used.

Acknowledgements. The authors would like to express their appreciation to all the volunteers who participated in the data collection and user study.

References

1. Bação, F., Lobo, V.: Introduction to Kohonen's Self-Organizing Maps, Universidade Nova De Lisboa, Portugal (2010), <http://edugi.uji.es/Bacao/SOM%20Tutorial.pdf>
2. Kohonen, T.: Self-Organizing Maps. Springer (1995)
3. Nishio, H., Altaf-Ui-Amin, M.D., Kurokawa, K., Kanaya, S.: Spherical SOM and Arrangement of Neurons Using Helix on Sphere. *IPSI Digital Courier* 2, 133–137 (2006)
4. Leontitsis, A., Sangole, A.P.: Estimating an optimal neighborhood size in the spherical self-organizing feature map. *Int. Journal of Computational Intelligence* 2, 94–98 (2005)
5. Sarle, W.S.: Neural Network FAQ Part 1 (2002), <ftp://ftp.sas.com/pub/neural/FAQ.html>
6. Wu, Y., Takatsuka, M.: Spherical self-organizing map using efficient indexed geodesic data structure. *Neural Networks* 19, 900–910 (2006)
7. Ritter, H.: Self-organizing maps on non-euclidean spaces. *Kohonen Maps*, 97–108 (1999)
8. Sangole, A., Knopf, G.K.: Visualization of randomly ordered numeric data sets using spherical self-organizing feature maps. *Computers & Graphics* 27, 963–976 (2003)
9. Sugii, Y., Satoh, H., Yu, D., Matsuura, Y., Tokutaka, H., Seno, M.: Spherical self-organizing map as a helpful tool to identify category-specific cell surface markers. *Biochemical and Biophysical Research Communications* 376 (2008)
10. Blazejewski, A., Coggins, R.: Application of self-organizing maps to clustering of high-frequency financial data. In: 2nd Workshop on Australasian Information Security, Data Mining and Web Intelligence, and Software Internationalisation, vol. 32, pp. 85–90 (2004)
11. Sharma, N., Gedeon, T.D.: Objective measures, sensors and computational techniques for stress recognition and classification: A survey. *Computer Methods and Programs in Biomedicine* 108(3), 1287–1301 (2012), doi:10.1016/j.cmpb.2012.07.003
12. Sharma, N., Gedeon, T.: Computational Models of Stress in Reading Using Physiological and Physical Sensor Data. In: Pei, J., Tseng, V.S., Cao, L., Motoda, H., Xu, G. (eds.) *PAKDD 2013, Part I. LNCS (LNAI)*, vol. 7818, pp. 111–122. Springer, Heidelberg (2013)