ESANN 2011 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2011, i6doc.com publ., ISBN 978-2-87419-044-5. Available from http://www.i6doc.com/en/livre/?GCOI=28001100817300.

SO-VAT: Self-Organizing Visual Assessment of cluster Tendency for large data sets

Enrique Pelayo and Carlos Orrite and David Buldain

Aragon Institute for Engineering Research, University of Zaragoza - SPAIN

Abstract. A new method, Self-Organizing Visual Assessment of cluster Tendency (SO-VAT), is given for visually assessing the cluster tendency in large data sets. It is based on training a SOM with the input samples, and then calculating the VAT image from a selected group of the generated neurons, selection that is done according to a certain density of activation. Tests with synthetic and real examples demonstrate that the new SO-VAT algorithm results in clearer images and shorter computing time than applying directly the VAT procedure to the whole input-data set.

1 Introduction

Cluster analysis or clustering is the assignment of a set of data samples into subsets (called clusters) in such a form that data in the same cluster are similar in some sense. One of the major problems in cluster analysis is the determination of the number of clusters in unlabelled data. Recently, spectral-VAT algorithm [1] has been proposed combining spectral analysis and [2] to automatically determinate the number of clusters. However, this approach involves the eigen-decomposition of an nxn similarity matrix, which is clearly intractable for a large number (n) of samples. The Self-Organizing Map (SOM) is a type of artificial neural network trained by unsupervised learning to produce a low-dimensional discrete representation of the training data distribution, called a map [3], usually configured as a two dimensional grid of neurons. It is a powerful tool in data mining, as it is capable of projecting high-dimensional data onto a neuron grid with good topological preservation between both spaces.

In this article, we present a new algorithm, SO-VAT (Self-Organizing Visual Assessment of cluster Tendency), to deal with large data sets. The new algorithm models the data using a SOM, then selects a group of the neuron prototypes according to their density of activation, and then applies the VAT algorithm to the selected prototypes. Our algorithm takes advantage of the use of the SOM, reducing significantly the computation time when applying the VAT algorithm.

2 The SO-VAT algorithm

Figure 1 shows the steps followed in the SO-VAT algorithm. First, the SOM is trained with the data to generate a number of map neurons that model the input data distribution. In the second step, for each neuron prototype is calculated a value, which we call density. This value is proportional to the number of data samples belonging to the data-space region (Voronoi region) modelled by the neuron prototype. The Density Matrix for the map prototypes is calculated, so we can index neurons by increasing density. After that, a removing-neuron phase has place: neurons are eliminated by order from minimum density to maximum density. Each neuron elimination produces an increment of the map-representation error in the data. The distribution of error increments for all neuron eliminations, except the ultimate one, is analyzed to decide an index of neuron for stopping this process. Finally, we apply the VAT algorithm to the remaining neuron prototypes.

The VAT algorithm is based on the principle that cluster structure in an unlabeled data set may be revealed by an image of some reordering of the rows and columns of the dissimilarity matrix, resulting blocks in the ordered image that correspond to clusters in the data. For example, three blocks in VAT images of first example in Figure 2(b) and Fig:2(c), represents three clusters. Our experiments demonstrate that VAT images are similar in both methods, Original VAT and SO-VAT, but with shorter computation time with this one.

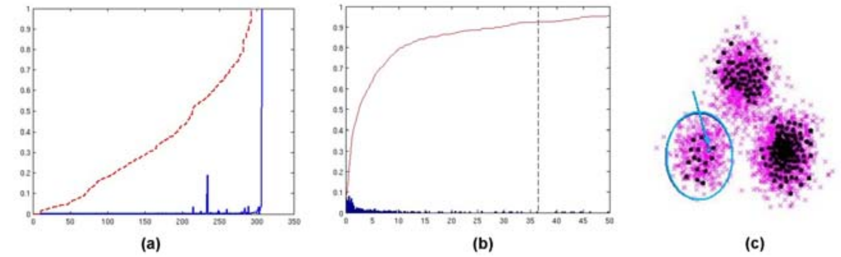

Fig. 1: The SO-VAT algorithm: (a) Ordering of neurons by density (dashed line) and the resulting increments of total quantization error, Qe_{tot}, when each neuron is removed (blue bars), (b) Probability density function of Qe_{tot} increments, (c) Original data vectors (magenta) and neuron prototypes (black).

2.1 Construction of the density matrix

An important characteristic of the SOM is that input-space regions with a number of data samples are represented with a proportional number of neurons, thus high-density data regions have more detailed prototype-representation than sparse-data regions. Therefore, removing neurons with low density of activation is a good way to get a cleaner prototype-representation for clustering.

The simplest way to define a density matrix on the SOM is the Hit Histogram, which visualises density by counting how many input samples are assigned to each neuron prototype. This solution lacks of taking into account the prototype inter-distances. A more accurate representation of the density of activation is the P-Matrix [4]. As calculating an exact Voronoi volume is a difficult task, they use approximated volumes of hyper-spheres of certain radius, whose centers are the neuron prototypes.

In this work, to simplify calculation we follow a different approach: we suppose each neuron prototype and its adjacent prototypes in the map grid, as if

ESANN 2011 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence
and Machine Learning. Bruges (Belgium), 27-29 April 2011, i6doc.com publ., ISBN 978-2-87419-044-5.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100817300.

they were positioned locally in a plane in the input space. Instead of calculating a volume, we estimate an area defined as a circle of radius $u_m/2$, where u_m is the mean distance between the prototype m and the prototypes of adjacent neurons in the map grid. We define the density of each neuron D_m, as the total number of samples assigned to it, divided by its corresponding area.

2.2 Selection of highest-density neurons

After ordering the map neurons by their densities, we proceed to the elimination of the less relevant neurons: those neurons with low density-values are removed from the map. The limiting k-th neuron-elimination, must be high enough for elimination of less relevant neurons, but low enough to avoid the elimination of neurons that represent clusters in the data. In Figure 1(a), the red dashed line represents the densities of the neurons (normalized by the maximum value), ordered in the horizontal axis by their increasing densities.

The second step is to evaluate the successive increments of the total quantization error in the surviving map with the data set, as neurons are eliminated from lower to higher densities. When evaluating the map with the data, each data vector is assigned to a neuron in the map called the best matching unit, BMU, which presents the minimum Euclidean distance between the data vector and its prototype. This distance is called Quantization error (Qe). The total quantization error of the map, Qe_{tot}, is obtained by summing up these errors for all the data vectors. After removing a neuron, their assigned data vectors must be reassigned to the remaining neurons. As the removed neuron was the BMU for that data vectors, this data reassignment produces an increment in the total quantization error, ΔQe_{tot}. The ΔQe_{tot} values (normalized by the maximum value) are represented as blue bars in the same Figure of densities 1(a). We see that there exists a lot of neurons whose elimination produce low ΔQe_{tot}, but there are two neurons (near 230 and 299) whose eliminations produce a very high increments (the elimination of the ultimate neuron of the map is not represented). In Figure 1(c) we can see an example of high increment of Qe_{tot}: let us suppose that, in certain point of the removing phase, the prototype pointed with the blue arrow is the last one that remains in the cluster inside the oval. As this prototype corresponds to the BMU for the data samples inside the oval region, when it is removed, those data samples are assigned to new BMUs with prototypes farther from the oval area, and the consequence is that total quantization error increases considerably. This visualization of ΔQe_{tot}, shows us that there are at least three important clusters in the data (two peaks and the ultimate neuron).

In order to select the limiting neuron k, we represent in Figure 1(b) the normalized histogram (equivalent to the probability density function) of the variable ΔQe_{tot}, zoomed in the lower zone. The increasing red line represents the cumulative distribution function. This distribution shows that almost all the neuron eliminations present low ΔQe_{tot}, which means that, when these neurons are removed, their assigned data vectors are reassigned to close neurons-prototypes, so the remaining neurons preserve a representation of the data that can be considered still good enough. The dashed black vertical line represents the mean value of ΔQe_{tot} and two times the standard deviation from the mean takes a value of 766 (it has not been represented for better visualization). This last value will be used as a threshold (ΔQe_{max}) to obtain the index k. Index k corresponds to the first neuron-elimination whose ΔQe_{tot} is equal or over this threshold. With this ΔQe_{max}, and following the Chebyshev's inequality, we can assure that at least we are dealing with the 75 % of the neuron-eliminations that produce a low ΔQe_{tot}. In order to preserve in the map-representation as much as possible of the data clusters, we will maintain in the map all the neurons with indexes over the limiting neuron k (surely representing data clusters), and a fraction of the neurons under k to avoid that low probability data-clusters being represented by only a few neurons.

3 Experimental results

To test the SO-VAT algorithm, we have used three synthetic 2-D large datasets and a high-dimensional real case (they are large enough to show the SO-VAT possibilities). We measure the computation time and compare it against the original VAT method. Image histogram of both matrices is also displayed. Clearly defined clusters show separated peaks in the corresponding VAT image histogram. On the other hand, when cluster limits are fuzzy, the histogram tends to be plain. All computations have been done using MATLAB and the SOM Toolbox ([5]) on a PC with 4024 MB RAM and 1.6Ghz Intel chip.

The first example consists on N = 3000 observations from a mixture of three 2-D normal distributions. The Original Data (magenta), and the computed SOM (black) are shown in Fig. 2(a). Figures 2(b) and (c) represents the VAT images after applying the VAT algorithm to the whole dataset (b) or to the SOM selected prototypes (c). We can appreciate that images for both dissimilarity matrices are quite similar; perhaps those for SOM prototypes are a bit clearer. However, SOM dissimilarity matrix needs a lower computation time (see Table 1). Image histograms (magenta = Original Data, black = SO-VAT) also reflect that the VAT image generated with our method is also slightly better than the one obtained with original algorithm applied to the input data.

The second example gives a scatter plot of a data set of N = 500 points that can be partitioned into four clusters (three small clusters with 20 samples and a larger cluster with 440). This example aims to test what is the response of the SO-VAT method with unbalanced clusters. The VAT images show the existence of a cluster with large cardinality and three additional clusters with much smaller ones. Even thought our method is based on removing neuron prototypes from the SOM map, it is effective in representing the four clusters. The third example is a bidimensional complex shape composed of two half-moon-like patterns. The size of this data set is N=3000, with 1500 points for each cluster. The resulting SO-VAT image presents a better quality (peaks in its histogram are clearly identified), and time saving is considerable (5.8 sec vs. 400 sec. with the original algorithm).

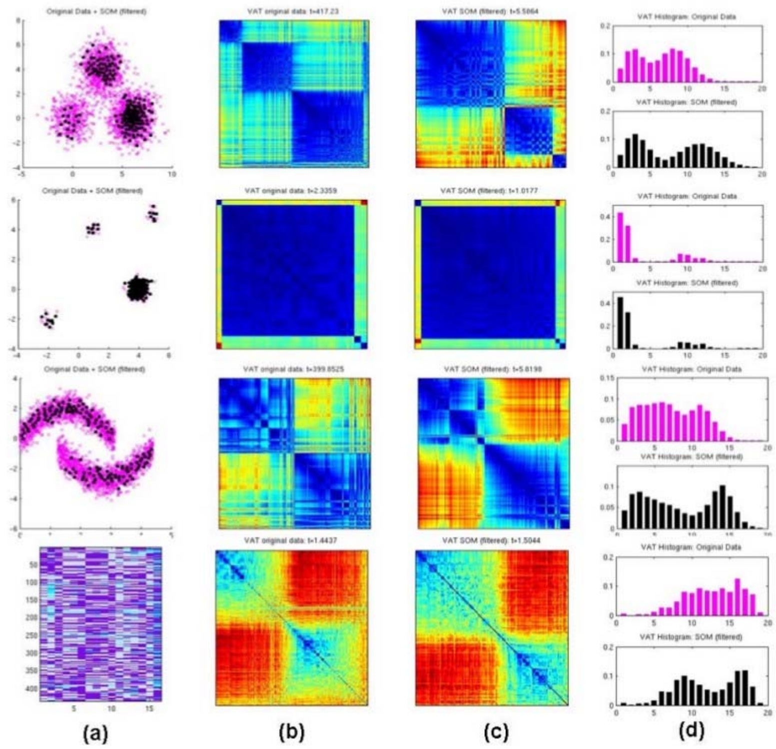

Fig. 2: Different examples based on MoG and complex data sets.

The final example involves a real data set consisting of the 1984 records of the 435 United States Representatives on 16 key votes [6] (Last row of Fig. 2(a) shows this matrix). Votes were numerically encoded so that the voting record of each Congressman is represented using an object data vector in $\Re^{16}$. The VAT image obtained with the traditional method, last row of Fig. 2(b), gives evidence that there are two well defined clusters and a smaller subset of data (corresponding to the last 15% or so of the rows and columns) with no clear cluster structure. The SO-VAT removes this last group, showing clearly the two clusters. This gives an idea of the capabilities of our algorithm to diminish the effect of outliers and long-tail samples in high-dimensional data sets.

4 Conclusions

In this study, we have proposed the Self-Organizing Visual Assessment of cluster Tendency (SO-VAT), a procedure to visually show cluster tendency in large data sets. SO-VAT works in four steps: (1) SOM prototypes are trained with the input data set. (2) Calculate the Density Matrix and order the prototypes according

Example	Clusters	Samples @ dim	t_{VAT}	t_{SO_VAT}
1	3	3000 @ 2	417.23	5.59
2	4	500 @ 2	2.34	1.02
3	2	3000 @ 2	399.85	5.82
4	2	435 @ 16	1.44	1.5

Table 1: Computation time (in seconds) for the mentioned examples.

to the increasing value of their densities. (3) Neurons with larger densities are selected based in the analysis of the distribution of increments of total Map Quantization-Error when neurons are removed following this ordering. (4) The VAT algorithm is applied to the these remaining neuron prototypes.

We have tested the behaviour of SO-VAT by applying it to four data sets. The synthetic data sets demonstrated that the new algorithm results in a drastically reduced computing time when compared to the original VAT algorithm.

On the other hand, the neuron prototypes obtained by the SO-VAT model accurately the full data set, excluding the undesirable effects of outliers and long-tail samples in the input data. Therefore, the new algorithm produces a better cluster visualization than the VAT. This is clearly seen in the high dimensional voting data set and, additionally, it shows that SO-VAT algorithm works perfectly with high-dimensional data. It is important to mention that our aim is to get clusters clearly visible in the VAT images, but not the size of these blocks. Future work has to be done to obtain block sizes closer to the cluster sizes, and to find an appropriate selection method of the cut-off threshold, that is a not automatically determined by our algorithm.

Acknowledgements

This work is partially supported by Spanish Grant TIN2010-20177 (MICINN) and FEDER and the University of Zaragoza under grant 171-161.

References

[1] L. Wang, X. Geng, J. Bezdek, C. Leckie, and K. Ramamohanarao, Enhanced Visual Analysis for Cluster Tendecy Assessment and Data Partitioning, *IEEE Trans. on Knowledge and Data Engineering*, 22:1401-1414, 2010.

[2] J.C. Bezdek, R.J. Hathaway and J.M. Huband, Visual Assessment of Clustering Tendency for Rectangular Dissimilarity Matrices, *IEEE Trans. on Fuzzy Systems*, 15:890-903, 2007.

[3] T. Kohonen. *Self-Organizing Maps*, Springer-Verlag, New York, 1997.

[4] A. Ultsch, Maps for the Visualization of High-dimensional Data Spaces. In *proceedings of the Workshop on Self-Organizing Maps* (WSOM 2003), pages 225-230, Kyushu Institute of Technology, Kitakyushu, (Japan), 2003.

[5] J. Vesanto, J. Himberg, E. Alhoniemi, J. Parkankangas, SOM Toolbox for Matlab 5. Technical report A57. CIS laboratory, Helsinki Univ. of Technology, Finland, 2000.

[6] J. Schlimmer, Concept acquisition through representational adjustment. T Doctoral dissertation, Dep. of Information and Computer Science, Univ. of California, Irvine, 1987.