

Self Organizing Star (SOS) for health monitoring

E. Côme¹, M. Cottrell¹, M. Verleysen² and J. Lacaille³

1- Université Paris I, SAMOS-MATISSE, UMR CNRS 8174
90 rue de Tolbiac, F-75634 Paris Cedex 13 - France

2- Université Catholique de Louvain, Machine Learning Group
Place du levant 3, 1348 Louvain-La-Neuve - Belgium

3- Snecma, Rond-Point René Ravaud-Réau,
77550 Moissy-Cramayel CEDEX - France

Abstract. Self Organizing Maps (SOM) have been successfully applied in a lot of real world hard problems since their apparition. In this paper we present new topologie for SOM based on a planar graph. The design of a specific graph to encode prior information on the dataset topology is the central question addressed in this paper. In this context, star-shaped graphs are advocated for health monitoring applications, leading to a new kind of SOM that we denote by Self Organizing Star (SOS). Experiments using aircraft engine measurements show that SOS lead to meaningful and natural dataset representation.

1 Introduction

SOM are well-known data mining tools which are frequently used to project high-dimensional data into a low-dimensional discrete space for visualization and exploratory purpose. In the context of dimensionality reduction, SOM can be thought as a kind of nonlinear but discrete PCA. SOM are classically defined by two key elements: a set of prototype vectors $\{\mathbf{m}_1, \dots, \mathbf{m}_K\}$, which are d -dimensional vectors, d being the dimensionality of the dataset $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ and a lattice or grid which defines a neighborhood relation between the prototypes. This lattice plays the role of the embedding space. Each prototype is associated with a fixed position in this lattice. Self-organization means that after convergence prototypes which are close in the lattice (with a small distance between them) will also be close in the data space. New data can be projected after learning of the SOM by affecting them to the location on the map of the best matching unit (BMU): the closest prototype vector to the datapoint.

In this paper we investigate the opportunity of using other lattices different from regular rectangular or hexagonal grids defined thanks to undirected graphs, which will have the additional interesting property of being planar. By using such graphs all the interesting properties of SOM are kept, in particular easy data visualization and data exploration with the additional freedom of designing lattices more suited to the expected dataset topology. We also show how such graph can be designed in the context of a health monitoring application. The classical lattices for SOM are rectangular or hexagonal grids. Extensions of these classical topologies have already been investigated by several authors. Toroidal or spherical maps [1, 2], growing grid with free topology [3], and tree structured

SOM (to speed up computational time required for BMU search) [4, 5], among others. We follow here a path more similar to [6] which focuses on the use of graphs to define SOM topologies. Our solution differs from [6] by the nature of the graph being used for defining the SOM topology. We promote in particular the star topology, which leads to the method proposed here and called Self Organizing Star (SOS). Such a solution will mix ideas from star plot [7] with the self-organization properties of Kohonen algorithm. The number of branches of the star is supposed to be known in this paper, while the SOS prototypes still have to be placed at a relevant position in the state space. Therefore SOS is not used for exploratory analysis which is a usual purpose of SOM but for visualization taking into account prior information on the dataset topology (the number of branches).

The paper is organized as follow. In Section 2 the basics of SOM will be reminded and the question of how the neighborhood relationship induced by a lattice influences the results of the algorithm will be addressed. The use of undirected graphs to define new lattices will also be investigated. In Section 3, the SOS solution proposed to deal with health monitoring problems will be advocated and described. Finally, some experimental results on real and simulated datasets will be given using such a topology; the paper ends with some concluding remarks.

2 Self-Organizing Maps Background

In its classical presentation [8, 9], the SOM algorithm is an iterative algorithm, which iterates the two following steps over training patterns \mathbf{x}_j for computing the set of prototypes $\mathbf{m}_i, i \in \{1, \dots, K\}$ which define the map:

- *Competitive step*, this step aims at finding the BMU for sample \mathbf{x}_j : $c = \arg \min_{i \in \{1, \dots, K\}} \|\mathbf{x}_j - \mathbf{m}_i\|$.
- *Cooperative step*, this step aims at moving the BMU and close prototypes (on the map), closer to the training pattern: $\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + \alpha(t)h_{ci}(t)[\mathbf{x}_j - \mathbf{m}_i(t)]$, with t the time step, $\alpha(t)$ the learning rate of the algorithm and $h_{ci}(t)$ the neighborhood function between prototypes c and i at time t .

Several neighborhood functions are commonly used such as $h_{ci}(t) = \mathbf{1}_{(d_{ci} < \sigma_t)}$ or $h_{ci}(t) = \exp(-d_{ci}^2/2\sigma_t^2)$. All of them depend on a radius σ_t which is classically decreasing during the learning process. These neighborhood function also depends on d_{ci} : the distance between neuron c and i , which is determined by the map topology. During the cooperative step the coordinates of the best matching prototype c are updated in order to move it closer to the training pattern. Other prototypes \mathbf{m}_i are also moved towards the training pattern according to their distance d_{ci} from the BMU defined by the lattice. Closer prototypes from the BMU will be more affected than the others. The two steps are iterated over the dataset during several epochs until convergence. Thanks to the cooperative steps, self-organization is reached at the end of the algorithm.

One can see that the distance between prototypes in the map plays a key role in the self-organization property of the algorithm; modifications of this distance will have an impact on the results of the algorithm. Therefore, the lattice structure can be a way to incorporate prior information concerning the dataset topology into the dimensionality reduction process. In the context of classical SOM, one assumes that the dataset topology can effectively be represented by a grid or a straight line but other hypotheses can be interesting and can advantageously be investigated. The key element that implicitly or explicitly defines the lattice is the distance between prototype on the map. It has been already noted that graph theory can be used to define such distance [6, 4]. In this case, map prototypes are the nodes of a graph, and distances between them are defined as the minimum number of edges needed to reach one node starting from the other, that is the so called shortest path distance. We therefore propose to modify the SOM algorithm only by taking as input an adjacency matrix which specifies the graph topology that the user desires. All undirected graphs can theoretically be used, but a specific class is of interest: the class of planar graph, because such graphs can easily be represented in a 2-dimensional setting allowing us to supply the SOM with visualization tools, as seen in the next section.

3 Star topology

In health monitoring applications, there exists a particular state which corresponds to the normal operation of the component under monitoring. The aim of health monitoring is to look at departures from this normal state, because faults lead to deviation of state variables from this normal state. Furthermore, we may hope that different faults are characterized by different shapes of drifts. The classical lattices of SOM do not take into account this prior information

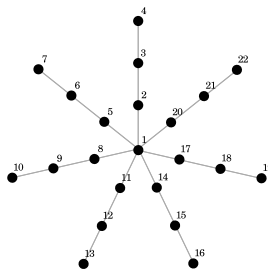


Fig. 1: Planar graph representation of the star topology with 7 rays and 3 prototypes by rays

on the dataset topology: there is no specific prototypes which takes a central place on the map. Moreover the users do not easily understand the meaning of the map since the normal states can for example be represented by a prototype located on an edge of the map, instead of a central position that would be more

natural. To prevent this difficulty, we propose to use a star-shaped graph as shown in Figure 1. This graph has a clear natural center from which different arms or rays grow. Such graph can easily be understood by users if different rays correspond to different faults and distances to the center describe the seriousness of the problem. These graphs can be characterized by their number of rays and by the length of the rays. SOM using such topologies will be called Self-Organizing Stars (SOS).

4 Results

4.1 Toy dataset

The first experiment aims at demonstrating the ability of such topology to represent a star-shaped dataset. For this purpose a toy dataset was generated in a 3-dimensional space. One hundred data points have been generated using a normal distribution with mean $(0, 0, 0)$ and standard deviation $2.I$, where I denotes the identity matrix of size 3×3 . Three other sets of one hundred data points have been drawn, each corresponding to a noisy version of the segment $[0, 15]$ along one of the three vectors of the \mathbb{R}^3 canonical basis. These sets were obtained by first sampling the location along the vectors using a uniform distribution between 0 and 15 and by adding isotropic noise with standard deviation equal to 1. A realization of such a dataset is presented in Figure 2 (a) with the

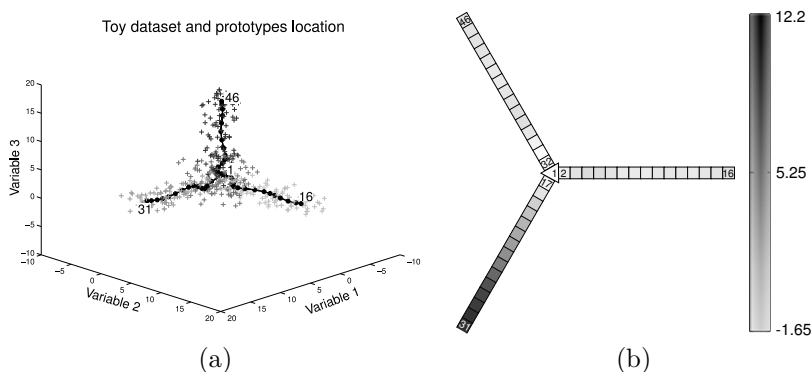


Fig. 2: (a) toy dataset in \mathbb{R}^3 and SOS prototypes positions after learning depicted by big black dots, (b) Variable 1 component plane; colors are proportional to the first value of prototype vectors. Codebook index $\{1, 16, 31, 46\}$ are given for both figures.

SOS prototypes positions after learning. These results were obtained using a SOS with 3 rays of 15 prototypes each. The training was performed using the classical inverse linear learning rate scheme of SOM algorithms and a decreasing radius between 5 and 1; the number of epochs of the algorithm was fixed to 5 and the initialization was performed using random assignments in the data range for all

prototypes. Figure 2 (b) gives a representation of map results in the form of a component plane. We can also see on this figure that one ray corresponds to the part of the dataset simulated along the first canonical vector. This two representations clearly show that such dataset can be effectively described by a SOS; moreover using the planar representation of the star graph, such a SOM can be effectively represented in a 2 dimensional way.

4.2 Health monitoring dataset

The second dataset comes from an health monitoring application which concerns aircraft engines. The data are multi-dimensional measurements on the engines (9 variables describing engine states and 7 variables describing the environment). The dataset was simulated using a thermo-dynamical model of the engine; six different sets of points were simulated, each one starting from a normal engine and gradually moving to a defective state. Such sets will be denoted as failure trajectory. Five sets concern progressive failures of different engine macro-components : fan, booster, high pressure core, high pressure turbine, low pressure turbine; the last one was generated with progressive degradation of all macro-components together. Random environmental conditions (external temperature, altitude, Mach, ...) were also simulated for all samples and finally realistic noise was added to all variables. Each one of the six sets contains 2000 points, leading to a dataset of 12000 multi-dimensional measurements. Raw engine measurements are inappropriate for direct use; they are indeed influenced by external conditions. We therefore first processed the data by a Linear Model (LM), to eliminate the effect of measured environmental conditions (external temperature, altitude, Mach, rotation speed, air levy, engine variable geometries positions) on the other 9 variables (temperatures and pressures on different locations and fuel consumption). Finally, the LM residuals are filtered using a simple moving average (15 samples) in order to reduce noise. The filtered residuals are then used as inputs to a SOS for the easy visualization of fault degradations. The map was trained using the filtered residuals of the 9 variables describing engine sates and an inverse linear scheme for the learning rate, 5 epochs and a neighborhood radius decreased from 5 to 1. The topology of the map was fixed manually: it contains 6 rays (one for each trajectory) of 20 prototypes each. Figure 3 (left) presents the projection of two trajectories on the map. As expected each fault corresponds to a ray; similar results were obtained for the other faults. Once learned, the map is therefore a very interesting tool for health monitoring as it enables the diagnosis of specific sub-component failures and the estimation of the severity of failure. Comparing these results with those obtained with a classical square map (see Figure 3 right), they are more easy to interpret.

5 Conclusion

The proposed SOM extension (SOS) based on a star shaped planar graph gives interesting results in the health monitoring application investigated in this paper. This approach is interesting when prior information on the dataset topology is

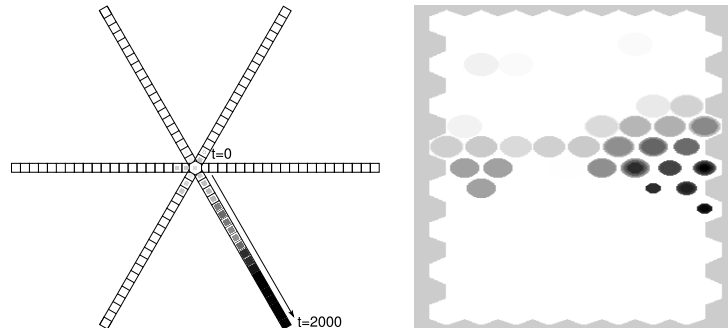


Fig. 3: Star-shaped map (left) and square map (right) for turbo fan engine health monitoring; projection of one failure trajectory on the maps. The trajectories contain 2000 measurements and these measurements are ordered according to the gravity of the failure. Color (from white to black) indicates the position of the measurements in the trajectory (from normal state to defective state).

available as in health-monitoring applications. However, several questions are still open, first including the choice of the number of rays and their lengths which is left up to now to the user. Classical model selection tools, approaches similar to Fritzke's GNG or GCS [3] with growing branches, can probably help for such a task.

References

- [1] Y. Wu and M. Takatsuka. Spherical self-organizing map using efficient indexed geodesic data structure. *Neural Networks*, 19:900–910, July 2006.
- [2] G. Andreu, A. Crespo, and J. M. Valiente. Selecting the toroidal self-organizing feature maps (tsomf) best organized to object recognition. In *Proceedings of International Conference on Artificial Neural Networks*, volume 1327 of *Lecture Notes in Computer Science*, pages 1341–1346, June 1997.
- [3] B. Fritzke. Growing cell structures – a self-organizing network for unsupervised and supervised learning. *Neural Networks*, 7:1441–1460, 1994.
- [4] J. Pakkanen, J. Iivarinen, and E. Oja. The evolving tree - analysis and applications. *IEEE Transactions on Neural Networks*, 17:591–603, 2006.
- [5] M.M. Campos and G.A. Carpenter. S-tree: self organizing tree for data clustering and online vector quantization. *Neural Networks*, 14:505–525, 2001.
- [6] A. Barsi. Neural self-organization using graphs. In *Proceedings of the Third International Conference Machine Learning and Data Mining in Pattern Recognition, Leipzig (Germany)*, volume 2734 of *Lecture Notes in Artificial Intelligence*, pages 343–352. Springer, July 2003.
- [7] J. Chambers, W. Cleveland, B. Kleiner, and P. Tukey. *Graphical Methods for Data Analysis*. Wadsworth, 1983.
- [8] T. Kohonen. *Self-Organizing Maps*. Information sciences. Springer, 1995.
- [9] M. Cottrell, Fort J.-C., and Pagès G. Theoretical aspects of the SOM algorithm. *Neurocomputing*, 21:119–138, 1998.