ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2.
Available from http://www.i6doc.com/en/.

Self-organizing maps in manifolds with complex topologies: An application to the planning of closed path for indoor UAV patrols

Hervé Frezza-Buet *

Université de Lorraine, CentraleSupélec, CNRS, LORIA, F-57000 Metz, France

Abstract. In this paper, the ability of 1D-SOMs to address the Euclidian Travelling Salesperson problem is extended to more irregular topologies, in order to compute short closed paths covering an indoor environment. In such environments, wall constraints makes the topology of the area to be visited by a patroller very irregular. An application to indoor unmanned aerial vehicule (UAV) security patrols is considered.

1 Introduction

Among the variety of vector quantization methods, the self-organizing feature maps (SOM) introduced by Kohonen [1] has the core property of forcing the space where the data lives to be projected onto an arbitrary topology, choosen by the user. Indeed, the SOMs summarize the data as a *graph* of prototypes linked by non-oriented edges. The number of vertices as well as the edges define the arbitrary topology of the SOM. The self-organization is such that similar prototypes are, as far as possible, neighbours on the graph. As the SOM topology may not reflect the topology of the manifold containing the data, the way a SOM finds a compromise between the actual topology of the data manifold and the arbitrary topology of the graph is the real strength of this algorithm.

The vast majority of SOMs are made of grid-shaped graphs. This is very convenient for large dataset visualization over the surface of a screen, as done for documents in the WEBSOM approach [2]. However, the present paper rather focuses on 1D SOMs where the graph edges connect the vertices as a loop. The SOM organizes then the prototypes such as close prototypes are placed on adjacent vertices in the loop. Visiting sequentially the prototypes in the loop order leads to following a closed path in the input space. This has motivated the use of 1D-SOM as an approximation of the Euclidian Travelling Salesperson Problem (TSP) [3], even if a similar idea with elastic nets has been proposed previously in [4]. In these approaches, the cities are points of a 2D Euclidian space, and the travelling salesperson is allowed to move from any city to any other in a straight line. In this context, adaptations of SOMs have been proposed, including adjustments of the number of prototypes. A deeper introduction to the problem as well as improvements of SOM for solving the Euclidian TSP can be found in [5]. After convergence, the SOM loop approximates the shortest tour that visits *all* the cities.

*This work is supported by the European Interreg Grande Région/Région Grand-Est project GRONE.

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2.
Available from http://www.i6doc.com/en/.

Aiming to visit all the cities justifies the SOM adjustments made in these works. Nevertheless, when used in practical context, an approximation of the tour, i.e. a closed path passing nearby all the cities rather than exatly at all the cities, may be enough, as argued in [6]. Sticking to the classical SOM formulation works in that case. In [6], "cities" are indeed loci of interest over the sea. The tour is made by a patroller who navigates without a strict requirement of passing accurately at each exact locus position. Let us stress that in this sea navigation case, the Euclidian TSP assumption is fulfilled since no obstacles over the see are considered, i.e. two loci can always be linked by a navigable straight line. The same assumption as been made when applying this idea to umanned aerial vehicules (UAVs) that need to patrol in a wide outdoor area [7], with a supplementary Dubins constaint added to the path.

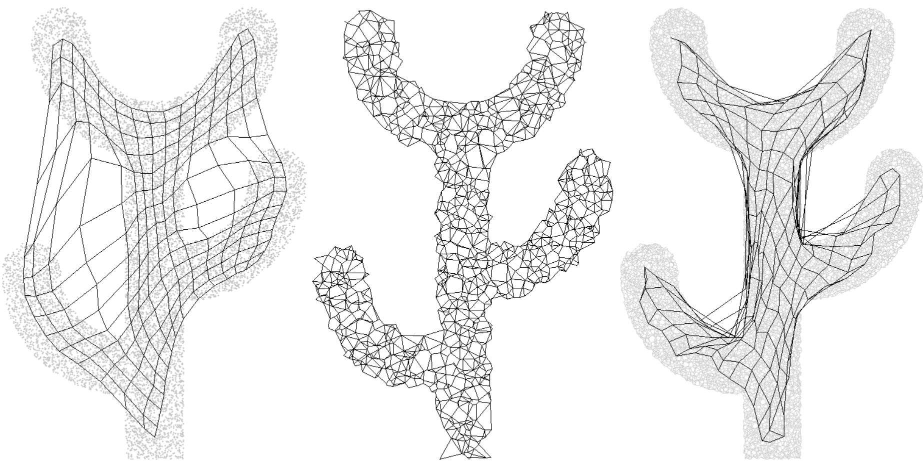

Fig. 1: On the left, a 15×15 SOM that maps the 2D cactus distribution. The gray dots are the input samples. On the center, the support graph G^{sup} with 1249 vertices and 2723 edges is plotted, for the sake of illustration of CHL. On the right, exactly the same process as the one on the left is run, with a SOM living in a graph-induced topology. The support graph actually used has 16737 vertices and 38334 edges. It is depicted in gray.

Our approach is motivated by similar applications, i.e. allowing a UAV to patrol in order to visit all the places of a region. The difference is that the area to visit is made of obstacles, as walls in an indoor environment. Our goal is then to tackle this problem with 1D-SOM as well, even if we cannot rely on the Euclidian property. Indeed, in an indoor environment, many straight lines between two random loci cross at least one wall. To overcome this topographical constraint, the present paper proposes to rewrite the SOM algorithm with a metric where the distance between two loci is the length of the shortest path allowing to travel from one locus to the other. Incoming section 2 presents an approximation of geodesics in complex topological spaces, that is used in section 3 to rewrite the

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

SOM algorithm. The application to UAV patrols with 1D SOMs in that context is adressed in section 4.

2 Arbitrary topologies from graphs

Let $\mathcal{X}$ be an Euclidian space with a distance d. Let us consider the samples $\xi \in \mathcal{X}$ as randomly chosen from a manifold $\mathcal{M} \subset \mathcal{X}$. The set $\mathcal{M}$ looks like a Euclidian space locally, but not at the global scale, sine it can have holes, different dimensions, etc. Only manifolds with a single connected component are considered here. The shortest path from one point in $\mathcal{M}$ to another one is a geodesic. Computing geodesics is not easy, especially when the manifold is only known through a finite set of samples taken from it. If the $\mathcal{M}$ is a surface and $\mathcal{X} = \mathbb{R}^3$, a triangularization of the surface can be computed so that geodesics, crossing the triangles' sides, can be approximated. Such algorithms have been proposed in computer graphics [8]. In this case, the triangulation is a graph that approximates the surface $\mathcal{M}$. In more general manifolds and higher dimensions, the Competitive Hebbian Learning (CHL) algorithm [9] allows to build such a graph from samples, at least approximately. Once the vertices are samples from $\mathcal{M}$, CHL consist in sampling a lot of supplementary points in $\mathcal{M}$ and, for each one, connect the two closest vertices, according to the Euclidian distance in $\mathcal{X}$. However, in as $\mathcal{M}$ may not be locally a surface, the graph may not be a triangulation and interpolations used in [8] cannot be applied.

In this paper, we propose to approximate geodesics as follows. First, we set up the vertices $(s_1, \cdots, s_V)$ of a so called "support" graph denoted by G^{sup} by taking V random samples from $\mathcal{M}$. Then, we use CHL to add the edges of G^{sup}. The cost of an edge $[s_1, s_2]$ is $d\,(s_1, s_2)$. Let us approximate the geodesic between two points $(\xi_1, \xi_2) \in \mathcal{X}^2$ as a multiline. The first segment is $[\xi_1, s_{\xi_1}]$, where $s_\xi = \text{argmin}_{s \in G^{\text{sup}}}\, d\,(\xi, s)$, the last segment is $[s_{\xi_2}, \xi_2]$ and the intermediate segments are the edges of the shortest path from s_{ξ_1} to s_{ξ_2}, computed by $A^\star$. The "graph-induced" distance $d^{\text{sup}}\,(\xi_1, \xi_2)$ is defined as the length of that multiline.

Although such a geodesic and *graph-induced* metrics can be easily defined for any manifold in any dimension, it is a rough approximation of the actual geodesics of $\mathcal{M}$. In the next section, we use it in a SOM algorithm in order to see if the SOM properties are preserved.

3 Extension of the SOM updating rule

A SOM is a graph of prototypes, usually a grid. Let us denote a vertex by p and the prototype hosted by this vertex w_p. The SOM algorithm consists basically in repeating the two following steps:

- Get a new sample ξ, and determine the vertex $p^\star = \text{argmin}_p\, d\,(\xi, w_p)$.
- $\forall p,\ w_p \leftarrow (1 - \lambda) w_p + \lambda \xi$ with $\lambda = \alpha h\,(p, p^\star)$.

with $\alpha \in]0, 1[$ and $h\,(p, p^\star)$ a positive function such as $h\,(p^\star, p^\star) = 1$ and $h\,(p, p^\star)$ decreases as the distance *on the SOM graph* between positions p and $p^\star$

increases. The decrease is low at the beginning, and it fastens as the algorithm runs [1].

In addition to the SOM graph, another graph G^{sup} is involved here, such as the distance used in the first step of the SOM algorithm racalled above is d^{sup}. The second step consists of moving the prototype along the segment $[w_p, \xi]$, with a fraction λ of the segment's length. Relying on G^{sup}, this step is replaced here by moving along the approximated geodesic defined in section 2, with a fraction λ of its length as well.

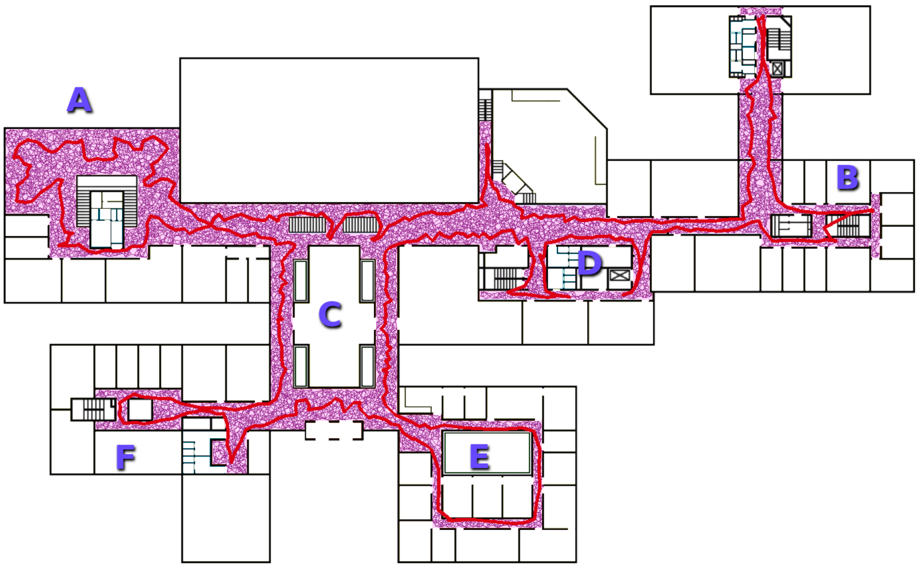

Fig. 2: A 1D self-orgnizing map, in red, approximating a TSP in the first floor's corridors (the pink region) of our building, thanks to a graph-induced topology. The SOM has 500 vertices. The support graph has 16406 vertices and 38419 edges. It is depicted in dark purple.

The comparison between the Euclidian and the graph-induced SOM proposed here, on a distribution inspired by the cactus used in [1], shows that the SOM is confined in the cactus manifold in our approach (see figure 1). This will be exploited in the definition of indoor patrols in the next section.

4 Non Euclidian TSP with 1D-self-organizing maps

As mentioned previously, our application context consists of building up a closed path in an indoor environment, for a security patrol made by a UAV. In this context, the UAV is supposed to move quite slowly, so there are no real constraints on the path curvature, as opposed to outdoor path planning [7] for faster UAV. The path is thus only required to cover all the area, to be as short as possible, and to form a close loop. In figure 2, the ξ inputs are 2D points taken randomly

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2.
Available from http://www.i6doc.com/en/.

in the corridors of the building. A support graph is built-up thanks to the CHL algorithm and a 1D SOM of 500 vertices is applied, using the graph-induced topology. The result is a closed path that actually covers all the corridor area. The path follows the corridor cycles, as in regions A, B, E, F. The surround of the area C is visited by 4 distinct portions of the path for each side, which is better that cycling around C and then visit the rest of the area. This should have been done at area D as well, where the result is sub-optimal there. In wider areas, the path follows meanders, as at A and at the bottom of C, in order to cover the area. The experiment shows that, in spite of few local sub-optimalities (at D and B), the properties that 1D-SOMs exhibit in TSP approximations is still preserved in more complex graph-induced topologies.

One critical part of the computation is the decay of the width of h, that has to be wide at the beginning of learning and has to get narrow as the 1D SOM learns. For the sake of computing efficiency, our h kernel is not a Gaussian function as in usual SOMs, but rather a linearly decreasing ramp saturated to zero (see equation 1).

$$h\left(p, p^{\star}\right) = \max\left(1 - \frac{\nu\left(p, p^{\star}\right)}{\rho}, 0\right),\ \rho > 1 \tag{1}$$

where $\nu\left(p, p^{\star}\right)$ is the number of edges in the closest path from p to $p^{\star}$ in the SOM graph. As it is a finite support function, only a part of the prototypes are actually updated when a new sample is presented, expecially when h gets narrow (i.e. ρ decreases). This saves a lot of computation time. Moreover, the computation involves a huge amount of Dijkstra and $A^{\star}$ algorithms executions on quite a wide graph G^{sup}. This is kept tractable thanks to `vq3` [10], an optimized C++ implementation of vector quantification that we provide to the community. Computational issued are not discussed further here.

5 Conclusion and perspectives

The first results presented in this paper show that approximating the topology of the data samples by a support graph, obtained by a CHL process, allows for applying SOMs with the real topology of the manifold where the data live. This enables the use the "TSP effect" of 1D SOMs for path planning in structured environments. The proposed method, thanks to CHL, is not specific to the dimension of the input space. To illustrate this, figure 3 shows an extension to 3D spaces on an artificial distribution, even if the application addressed in this paper is 2D. The result for indoor patrol path planning presented in figure 2 is promising, but very sensitive to the decay of the winner-take-most kernel (h) width. Let us stress that we have only applied a basic SOM algorithm, while former TSP approaches use refinements to adjust the number of prototypes as the SOM expands. Such refinements will be investigated for graph-induced 1D SOMs in incoming work, in order to offer a ready-to-use planner for indoor patrols, as the one by UAV that motivates our work.

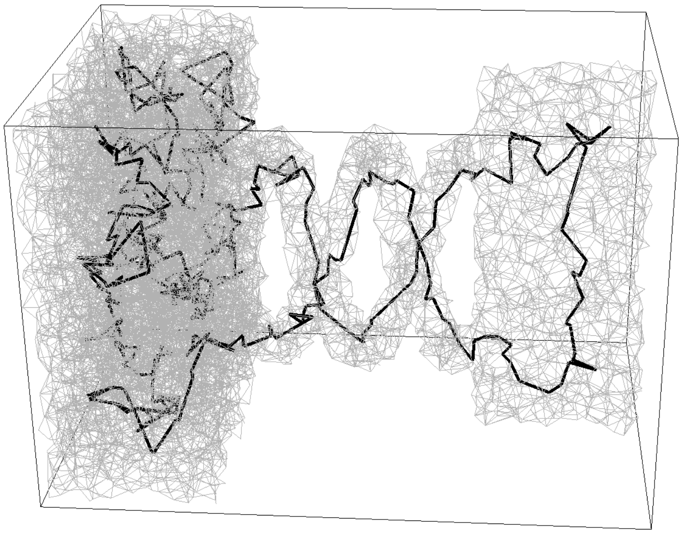

Fig. 3: A 1D self-orgnizing map in a 3D distribution. The 3D distribution is a full 3D box on the left, a 2D rectangular surface on the right, both being linked by two interlaced (but separate) spring shapes. The SOM (in thick black) has 500 vertices, the support graph has 10000 vertices and 30355 edges.

References

[1] Teuvo Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43(1):59–69, 1982.

[2] Samuel Kaski, Timo Honkela, Krista Lagus, and Teuvo Kohonen. Websom – self-organizing maps of document collections. *Neurocomputing*, 21(1):101 – 117, 1998.

[3] Bernard Angéniol, Gaël de La Croix Vaubois, and Jean-Yves Le Texier. Self-organizing feature maps and the travelling salesman problem. *Neural Networks*, 1(4):289 – 293, 1988.

[4] Richard Durbin and David Willshaw. An analogue approach to the travelling salesman problem using an elastic net method. *Nature*, (326):689—691, 1987.

[5] N. Aras, B.J. Oommen, and İ.K. Altınel. The kohonen network incorporating explicit statistics and its application to the travelling salesman problem. *Neural Networks*, 12(9):1273 – 1284, 1999.

[6] Victor Lobo. One dimensional self-organizing maps to optimize marine patrol activities. In *Oceans 2005 - Europe*, volume 1, pages 569 – 572, 2005.

[7] J. Faigl and P. Váňa. Unsupervised learning for surveillance planning with team of aerial vehicles. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 4340–4347, 2017.

[8] Vitaly Surazhsky, Tatiana Surazhsky, Danil Kirsanov, Steven J. Gortler, and Hugues Hoppe. Fast exact and approximate geodesics on meshes. *ACM Transactions on Graphics*, 24(3):553–560, 2005.

[9] T. M. Martinez and K. J. Schulten. Topology Representing Networks. *Neural Networks*, 7(3):507–522, 1994.

[10] Hervé Frezza-Buet. The vq3 library, 2018. `https://github.com/HerveFrezza-Buet/vq3`.