ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2012, i6doc.com publ., ISBN 978-2-87419-049-0. Available from http://www.i6doc.com/en/livre/?GCOI=28001100967420.

Extraction of Betti numbers based on a generative model

Maxime Maillot[1], Michaël Aupetit[1] and Gerard Govaert[2]

1- CEA, LIST, Information, Models and Machine Learning Laboratory (LIMA)
F-91191 Gif-sur-Yvette, France

2- University of Technology of Compiegne - Heudiasyc
BP 20529 F 60205 Compiègne, France

Abstract. Analysis of multidimensional data is challenging. Topological invariants can be used to summarize essential features of such data sets. In this work, we propose to compute the Betti numbers from a generative model based on a simplicial complex learnt from the data. We compare it to the Witness Complex, a geometric technique based on nearest neighbors. Our results on different data distributions with known topology show that Betti numbers are well recovered with our method.

1 Introduction

Exponential growth of sensors and databases leads to more and more multidimensional data to process. The analyst needs new exploration tools to get more relevant summaries of these data. Unsupervised learning methods usually extract some relevant variables or combinations of variables using dimension reduction techniques, or summarize the data instances straight into the multidimensional space using clustering techniques [1].

Here we focus on a topological summary of the data. Interesting topological invariants are preserved through homotopy, a very large class of nonlinear transformations which includes homeomorphism, similarities and isometries as nested cases. Thus these invariants are more robust than geometrical or statistical properties. So they are more likely to survive the processing chain from the set of sensors observing the physical phenomenon under study, to the set of instances and variables finally used to describe this phenomenon. In this work, we focus on the extraction of such topological invariants from the data set.

2 State of the art

We assume that data are drawn from a set of manifolds in $\mathbb{R}^D$(the population) following some probability density function (pdf) corrupted with noise. Gaussian Mixture Models (GMM)[1] are generative models which attempt to estimate the population pdf from the data sample with a linear combination of Gaussian pdf. The GMM can also be used for a clustering purpose, where each component accounts for some part of the population. This latter point of view assumes that the population is a set of generative points (the mean vector of each Gaussian component), *i.e.* 0-dimensional manifolds corrupted with Gaussian noise (the Gaussian pdf of each component). From a topological point of view, this

ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2012, i6doc.com publ., ISBN 978-2-87419-049-0. Available from http://www.i6doc.com/en/livre/?GCOI=28001100967420.

model is very simple as it encodes the data as a set of disjoint point-sources. Some attempts have been made to extract more topological information from the data. Generative models in the spirit of Self-Organizing Maps [2], like the Generative Principal Manifolds [3] or the Generative Topographic Maps [4] have been proposed but they impose a priori the manifolds to be connected and one or two dimensional. In our previous work [5, 6], we proposed the Generative Gaussian Graph similar to the Topology Representing Network [7], to learn the connectedness of the population in a statistical learning framework. We defined a weighted Delaunay subgraph of some prototypes located with a GMM, and use convolution of this graph with a Gaussian pdf as the basis of a generative mixture model. Vertices and edges of the graph are the components of the mixture. Proportions of the components are tuned to maximize the likelihood. Edges with low proportions are pruned from the graph so that remain only edges and vertices which explain the data.

In the field of Computational Topology, new tools have been proposed to extract more subtle topological information from the data than the connectedness. The Betti numbers are such information which count the number of unconnected component in each dimension. For instance, a sphere has one connected component, no hole and one cavity, its Betti numbers are (1,0,1,0,0,...). More details about these topological notions can be found in [8]. Betti numbers provide a topological signature of manifolds which allows classifying them according to some characteristics of their topology. The basis to compute Betti numbers is to model the data with a simplicial complex. A simplical complex C is a collection of simplices S such that for any two simplices S_i and S_j in C, their intersection is either empty or in C. A k-simplex is a set of $(k+1)$-vertices which can be embedded in $\mathbb{R}^{D \geq k}$ as the convex hull of $k+1$ points in independent position : a 0-simplex is a point; a 1-simplex is a line segment; a 2-simplex is a triangle with its interior; a 3-simplex is a tetrahedron with its interior...

The Witness Complex (WitC) have been proposed in [DeSilva04]. It extends the Topology Representing Network (TRN) from Delaunay subgraphs to Delaunay simplicial complexes. In WitC, a set of vertices (0-simplex) is defined subsampling the data or using some vector quantization technique. For each data, the two nearest vertices to it are connected with an edge (1-simplex). A data which leads to the creation of a simplex is called its ”witness”. At that point, the algorithm is identical to the TRN. Then for each data, the three nearest vertices to it define a triangle (a 2-simplex) except if one of its edges (1-face of the 2-simplex) has no witness. And so on, for each data, the k nearest vertices to it define a $(k-1)$-simplex of the Witness Complex except if one of its $(k-2)$-faces has no witness. The obtained Witness Complex is a simplicial complex whose topology is intended to catch the one of the population underlying the data. This technique is essentially a geometrical one and it has some limits among which : it is sensitive to noise; no statistical criterion is optimized; the witness complex is not self-consistent as points densely drawn from its embedding may not give rise to it although based on the very same set of vertices.

In the sequel, we propose to extend the Generative Gaussian Graph to a generative simplicial complex model and we compare it to the Witness Complex while extracting Betti numbers from some sampled manifolds.

3 The Generative Simplicial Complex

In this paper, we assume that the data are vectors of $\mathbb{R}^D$ drawn from a collection of manifolds corrupted with a centered gaussian noise, whose variance σ^2 is unknown. We assume that this collection of manifolds can be approximated with a Delaunay simplicial complex of some vertices located in $\mathbb{R}^D$. We define the Generative Simplicial Complex (GSC) as such a model, and use the EM method [9] to tune its parameters and maximize its likelihood.

3.1 The generative simplex

A gaussian simplex is the elementary component of the Generative Simplicial Complex. It is a probability density function. Let S^d be a simplex of dimension d with $d+1$ vertices $\underline{w}$ in $\mathbb{R}^D$, $|S^d|$ its volume,g a D-dimension gaussian distribution, $\sigma > 0$ the standard deviation, and p the propability distribution induced by the gaussian simplex associated with S^d. Then

$$p(x|S^d, \sigma^2) = \frac{1}{|S|} \int_{S^d} g(x|v, \sigma^2) dv \quad \text{with} \quad g(x|\mu, \sigma^2) = \frac{1}{2\pi^{D/2}\sigma^D} e^{-\frac{1}{2}\frac{||(x-\mu)||^2}{\sigma^2}}$$

We use a quasi Monte-Carlo method to evaluate this integral [10].

3.2 From the Generative Simplex to the GSC

A Generative Simplicial Complex is a mixture of generative simplices. Let S_i^d be the i-th simplex of dimension d, π_i^d its proportion in the mixture model, $p(x|S_i^d, \sigma^2)$ its probability density function, D the maximum dimension of a simplex in the model, n_d the number of simplices of dimension d, σ the standard deviation (the same for every simplex), and $D_t \leq D$ the current maximum dimension in the iterative algorithm.

$$p(x) = \sum_{d=0}^{D_t} \sum_{i=1}^{n_d} \pi_i^d p(x|S_i^d, \sigma^2)$$

4 The learning process

Given a data set, a generative simplicial complex can be learnt from this data, to approximate the underlying manifold.

1. **Initialization :** This step gives us a set of prototypes in $\mathbb{R}^D$ that we will use as the vertices $\underline{w}$ of the GSC. We use a classical Gaussian Mixture Model where the prototypes will be the center of the gaussian distributions optimized thanks to the Expectation-Maximization [9] algorithm. The selection of the number of prototypes is made using the BIC criterion [11].

ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2012, i6doc.com publ., ISBN 978-2-87419-049-0. Available from http://www.i6doc.com/en/livre/?GCOI=28001100967420.

2. **Building the initial complex :** We build the Delaunay simplicial complex of the prototypes in dimension D with the prototypes. We sort the simplices by increasing dimension. These simplices are the components of the GSC model. In the next step we want to prune this GSC, to get another one that would better fit the data with respect to the likelihood.

3. **Iterative building of the GSC :** We start with the vertices and the edges belonging to the GSC, we have a partial GSC of dimension 1 ($D_t = 1$). Step-by-step, we will add the triangles ($D_t = 2$), then the tetrahedra ($D_t = 3$) and so on until $D_t = D$. At step t, we have a D_t-dimensional GGSC. During each step, we maximize the likelihood, using EM, only to optimize the weights π. They are initialized as $\pi_i^d = 1/\sum_{j=0}^{D_t} n_j$.

   After convergence, if a weight π_i^d is below a certain threshold s, the simplex S_i^d is automatically removed from the GGSC *ie* $\pi_i^d = 0$. If a simplex has a weight $\pi_i^d \geq s$, its facets of lower dimension are removed from the GGSC.

4. **Getting the Betti numbers :** Now that we have a pruned GSC, we use Plex [12] to get its Betti numbers.

5 Experiments

5.1 Checking the validity of the model

These first experiments shows that learning with a GSC is possible on elementary simplices. It also gives us a threshold to prune the GSC during the iteration step 3 in section 4. The data are generated using a GSC model with vertices $\underline{w}$ located at coordinates $(1, 0, ..., 0), (0, 1, 0, ..., 0), ...$ in $\mathbb{R}^D$. A simplex of dimension d will compete with simplices of dimension $d-1$. We want to be sure that the model can learn the intrinsic dimension of the data, which is among the basic topological information we want to learn. For example, three segments forming a triangle are competing with the interior of the triangle and vice-versa. We set the vertices of the GSC on the ones used to generate the data, and set the noise to $\sigma = 0.05$, in the data and in the model. We also set the threshold at a tenth of the initial weight values.

In each case, the algorithm is able to make a distinction between a simplex of dimension d and its border of intrinsic dimension $d - 1$.

5.2 Learning the Betti numbers on a random simplicial complex

We generate data from a random simplicial complex, drawn from the elementary simplex of dimension 5, with a noise $\sigma = 0.05$. We get non trivial topology with such a complex. We use the same vertices for the GSC model to learn as for the data the GSC generates. The algorithm is capable of dealing with different intrinsic dimensions. We only compare the Betti numbers of the generated random complex to the ones learnt from the GGSC. The following table gives the results. It is possible to learn the Betti numbers from data generated by a simplicial complex.

	$\sigma = 0.005$	$\sigma = 0.01$	$\sigma = 0.05$
GGSC	87.00%	83.00%	95.00%

Fig. 1: Betti numbers learning with different noises

5.3 Learning the Betti numbers of a sphere and a ball

Two data sets are used in this experiment : a sphere, of radius 1 corrupted with three different noise variances, and a ball of the same radius, also corrupted with the same noise. Betti numbers of the sphere are (1,0,1,0,...), Betti numbers of the ball are (1,0,0,0,...). Data are generated in a three dimension space. 30 vertices are used for the learning (30 was the number found after running GMM with BIC on the data).

	WitC	GGSC 0.005	GGSC 0.01	GGSC 0.05
$\sigma = 0.005$	0.00%	83.00%	0.00%	0.00%
$\sigma = 0.01$	0.00%	90.00%	98.00%	0.00%
$\sigma = 0.05$	1.00%	95.00%	89.00%	87.00%

Fig. 2: Success rate of extracting Betti numbers of a sphere made of 2000 points corrupted with noise σ with WitC and GSC with different σ values

	WitC	GGSC 0.005	GGSC 0.01	GGSC 0.05
$\sigma = 0.005$	50.00%	100.00%	86.00%	0.00%
$\sigma = 0.01$	56.00%	89.00%	97.00%	0.00%
$\sigma = 0.05$	1.00%	0.00%	0.00%	87.00%

Fig. 3: Success rate of extracting Betti numbers of a ball made of 4000 points corrupted with noise σ with WitC and GSC with different σ values

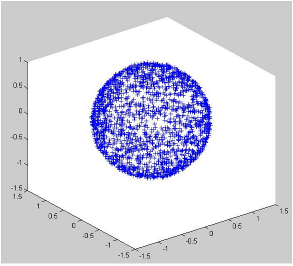

Fig. 4: Data used for learning the Betti numbers of a sphere

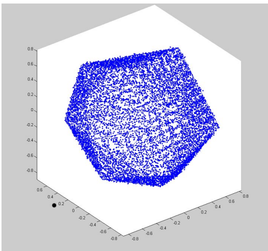

Fig. 5: Generative model learnt from a distribution of a sphere

The GSC model can have very good performances, but it really depends on the variance parameter. Such a parameter does not exist in WitC. WitC bad results on the sphere can be explained by its weakness towards noise. The sphere problem is the most difficult because a 2-dimension manifold is corrupted by a 3-dimension noise, which makes it look like a ball.

6 Conclusion

The GGSC is the first generative model capable of extracting the Betti numbers of data drawn from a manifold, corrupted with noise. If the second Betti number or upper is non-zero, it means there are cycles or holes in the data distribution: projection in lower dimension may contain tears or false-neighbourhood. GSC has a dependancy on the variance parameter, that we intend to learn automatically. It takes about three minutes to compute a GSC in the case of the sphere; computing the initial probability densities with quasi Monte-Carlo and the Delaunay Complex are the two longest steps. We compared the GSC to the WitC, but we plan to compute the persistance of homology [8] to be more accurate.

References

[1] Geoffrey McLachlan and David Peel. In Wiley, editor, *Finite Mixture Models*, 2000.

[2] Teuvo Kohonen. Self-organization and associative memory formation. 1988.

[3] Robert Tibshirani. Principal curves revisited. In Springer, editor, *Statistics and Computing vol. 2*, pages 183–190, 1992.

[4] Christopher Bishop, Markus Svensén, and Christopher Williams. The generative topographic mapping. In *Neural Computation*, pages 215–234, 1998.

[5] Michaël Aupetit. Learning topology with the generative gaussian graph and the em algorithm. In MIT Press, editor, *Advances in Neural Information Processing Systems*, pages 83–90, 2006.

[6] Pierre Gaillard, Michaël Aupetit, and Gérard Govaert. Learning topology of a labeled data set with the supervised generative gaussian graph. In *Neurocomputing*, pages 1283–1299, 2008.

[7] Thomas Martinetz and Klaus Schulten. Topology representing networks. In Elsevier London, editor, *Neural Networks vol. 7*, pages 507–522, 1994.

[8] Gunnar Carlsson. Topology and data. pages 255–308, 2009.

[9] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, SERIES B*, pages 1–38, 1977.

[10] William Morokoff and Russel Caflisch. Quasi-monte carlo integration. In *Journal of Computational Physics*, pages 218–230, 1995.

[11] Gideon Schwarz. Estimating the dimension of a model. In *The Annals of Statistics vol. 6*, pages 461–464, 1978.

[12] Vin de Silva. Plex - a matlab library for studying simplicial homology - *comptop.stanford.edu/programs/plex/plexintro.pdf*.