Learning the topology of a data set

Michaël Aupetit – Researcher Engineer (CEA)
Pierre Gaillard – Ph.D. Student
Gerard Govaert – Professor (University of Technology of Compiègne)
Given a set of $M$ data in $\mathbb{R}^D$, the estimation of the density allow solving various problems:

**classification, clustering, regression**
A question without answer…

The generative models cannot answer this question: Which is the « shape » of this data set?
An subjective answer

The expected answer is:

1 point and 1 curve not connected to each other

The problem: what is the topology of the principal manifolds
Why learning topology: (semi)-supervised applications

Estimate the complexity of the classification task
[Lallich02, Aupetit05Neurocomputing]

Add a topological a priori to design a classifier
[Belkin05Nips]

Add topological features to statistical features

Classify through the connected components or the intrinsic dimension.
[Belkin]
Why learning topology: unsupervised applications

Clusters defined by the connected components

Data exploration (e.g. shortest path)

Robotic (Optimal path, inverse kinematic)

[Zeller, Schulten -IEEE ISIC1996]
Generative manifold learning

Gaussian Mixture

MPPCA [Bishop]

GTM [Bishop]

Revisited Principal Curves [Hastie, Stuetzle]

Problems: fixed or incomplete topology
All the previous work about topology learning has been grounded on the result of Edelsbrunner and Shah (1997) which proved that given a manifold and a set of \( N \) prototypes nearby \( M \), it exists a subgraph* of the Delaunay graph of the prototypes which has the same topology as \( M \).

* more exactly a subcomplex of the Delaunay complex
All the previous work about topology learning has been grounded on the result of Edelsbrunner and Shah (1997) which proved that given a manifold \( M \) and a set of \( N \) prototypes nearby \( M \), it exists a subgraph of the Delaunay graph of the prototypes which has the same topology as \( M \).
All the previous work about topology learning has been grounded on the result of Edelsbrunner and Shah (1997) which proved that given a manifold and a set of N prototypes nearby M, it exists a subgraph of the Delaunay graph of the prototypes which has the same topology as M.
All the previous work about topology learning has been grounded on the result of Edelsbrunner and Shah (1997) which proved that given a manifold $M$ and a set of $N$ prototypes nearby $M$, it exists a subgraph of the Delaunay graph of the prototypes which has the same topology as $M$. 
All the previous work about topology learning has been grounded on the result of Edelsbrunner and Shah (1997) which proved that given a manifold $M$ and a set of $N$ prototypes nearby $M$, it exists a subgraph of the Delaunay graph of the prototypes which has the same topology as $M$. 

$M_1$ $M_2$
Computational Topology

All the previous work about topology learning has been grounded on the result of Edelsbrunner and Shah (1997) which proved that given a manifold and a set of $N$ prototypes nearby $M$, it exists a subgraph of the Delaunay graph of the prototypes which has the same topology as $M$.

Extractible topology $O(DN^3)$
Application: known manifold

Topology of molecules
[Edelsbrunner1994]
Approximation : manifold known through a data set
Topology Representing Network

- Topology Representing Network [Martinetz, Schulten 1994]

Connect the 1st and 2nd NN of each data
Topology Representing Network

- Topology Representing Network [Martinetz, Schulten 1994]
Topology Representing Network

- Topology Representing Network [Martinetz, Schulten 1994]
Topology Representing Network

- Topology Representing Network [Martinetz, Schulten 1994]
Topology Representing Network

- Topology Representing Network [Martinetz, Schulten 1994]
Topology Representing Network

- Topology Representing Network [Martinetz, Schulten 1994]
Topology Representing Network
Good points:

1. \( O(DNM) \)
2. If there are enough prototypes and if they are well located then resulting graph is « good » in practice.

Some drawbacks from the machine learning point of view
Topography Representing Network: some drawbacks

- Noise sensitivity
Topology Representing Network: some drawbacks

- **Not self-consistent [Hastie]**
Topography Representing Network: some drawbacks

• **No quality measure**
  – How to measure the quality of the TRN if $D > 3$?
  – How to compare two models?

For all these reasons, we propose a generative model
General assumptions on data generation

Unknown principal manifolds
Unknown principal manifolds …from which are drawn data with a unknown pdf
General assumptions on data generation

Unknown principal manifolds

...from which are drawn data with an unknown pdf

...corrupted with some unknown noise leading to the observation
General assumptions on data generation

The goal is to learn from the observed data, the principal manifolds such that their topological features can be extracted.
The manifold is close to the DG of some prototypes

\[ p(x) = \sum_{j \in J} \]
3 assumptions…1 generative model

The manifold is close to the DG of some prototypes

we associate to each component a weighted uniform distribution

\[ p(x) = \sum_{j \in J} p(j) \]
3 assumptions…1 generative model

The manifold is close to the DG of some prototypes

we convolve the components by an isotropic Gaussian noise

we associate to each component a weighted uniform distribution

\[ p(x) = \sum_{j \in J} p(j) \, p(x | j, \sigma) \]

\[ \sum_{p(j)} p(j) = 1 \]

\[ p(j) \geq 0 \]
How to define a generative model based on points and segments?

\[ p^0(x|A,\sigma) = (2\pi\sigma^2)^{\frac{n}{2}} \exp\left(-\frac{(x - A)^2}{2\sigma^2}\right) \]

\[ p^1(x[AB],\sigma) = \int_{[AB]} p(x|v,\sigma) dv \]

can be expressed in terms of « erf »
Hola!
Proposed approach : 3 steps

1. Initialization

Location of the prototypes with a « classical » isotropic GM

...and then building of the Delaunay Graph
Initialize the generative model
(equiprobability of the components)
Number of prototypes

\[
\text{min BIC} \sim - \text{Likelihood} + \text{Complexity of the model}
\]

![Graph showing the minimum BIC vs. number of prototypes. The graph has a blue line with a peak at around 40 prototypes. The y-axis is labeled with values starting from 3.84 to 3.93 and the x-axis ranges from 35 to 70.]}
Proposed approach : 3 steps

2. Learning

\[ p(x) = \sum_{j \in J} p(j) \cdot p(x \mid j, \sigma) \]

update the variance of the Gaussian noise, the weights of the components, and the location of the prototypes with the EM algorithm

in order to maximize the Likelihood of the model w.r.t the N observed data:

\[ L(\pi, \sigma; x, DG) = \prod_{i=1}^{N} p(x_i; DG, \pi, \sigma) \]
EM updates

\[
p(x, c; \pi, \beta, w, \sigma, DG) = \sum_{d} \sum_{j} \pi_{j}^{d} \beta_{c_{j}}^{d} g(x|(d, j); \sigma)
\]

\[
L(\pi, \beta, w, \sigma, DG) = \prod_{i=1}^{M} p(x_i, c_i; \pi, \beta, w, \sigma, DG)
\]

\[
\pi_{j}^{d[\text{new}]} = \frac{1}{M} \sum_{i=1}^{M} p((d, j)|x_i, c_i)
\]

\[
\sigma^{2[\text{new}]} = \frac{1}{DM} \sum_{i=1}^{M} \left[ \sum_{j=1}^{N_0} \frac{N_1}{p((0, j)|x_i, c_i)(x_i - w_j)^2} + \sum_{j=1}^{N_1} \frac{(2\pi\sigma^2)^{-D/2} \exp\left(-\frac{(x_i - q_j)^2}{2\sigma^2}\right)(I_1[(x_i - q_j)^2 + \sigma^2] + I_2)}{I_j \cdot g(x_i|(1, j); \sigma)} \right]
\]

\[
\beta_{c_{j}}^{d[\text{new}]} = \frac{\sum_{i=1; c_i = c}^{M} p(d, j|x_i, c_i)}{\sum_{i=1}^{M} p(d, j|x_i, c_i)}
\]

(9)

\[
I_1 = \sigma \sqrt{\frac{\pi}{2}} (\text{erf}\left(\frac{Q_i^{j}}{\sigma\sqrt{2}}\right) - \text{erf}\left(\frac{Q_j^{i}-L_j}{\sigma\sqrt{2}}\right))
\]

\[
I_2 = \sigma^2 \left( (Q_j^{i}-L_j) \exp\left(-\frac{(Q_j^{i}-L_j)^2}{2\sigma^2}\right) - Q_j^{i} \exp\left(-\frac{(Q_j^{i})^2}{2\sigma^2}\right) \right)
\]
EM updates

\[ w^{[\text{new}]}_k = \frac{\sum_{i=1}^{M} \left[ p(0,k|x_i)x_i + \sum_{j \in W_k} p(1,j|x_i) \frac{g_0(x_i|q_j^i;\sigma)}{L_j \cdot g_1^j(x_i;\sigma)} (-E_2 w_{b_j} + E_3 x_i) \right]}{\sum_{i=1}^{M} \left[ p(0,k|x_i) + \sum_{j \in W_k} p(1,j|x_i) E_1 \right]} \]

\[ E_1 = \frac{\sigma^2}{L_j} \left[ e^{-\frac{(Q_j)^2}{2\sigma^2}} (Q_j - 2L_j) - e^{-\frac{(Q_j-L_j)^2}{2\sigma^2}} (Q_j - L_j) \right] + \frac{1}{L_j} ((L_j - Q_j)^2 + \sigma) I_1 \]

\[ E_2 = \frac{\sigma^2}{L_j^2} \left[ e^{-\frac{(Q_j-L_j)^2}{2\sigma^2}} Q_j - e^{-\frac{(Q_j)^2}{2\sigma^2}} (Q_j - L_j) \right] - \frac{1}{L_j^2} (Q_j^2 - L_j Q_j + \sigma^2) I_1 \]

\[ E_3 = \frac{1}{L_j} \left[ e^{-\frac{(Q_j-L_j)^2}{\sigma^2}} - e^{-\frac{Q_j^2}{\sigma^2}} + (Q_j - L_j) I_1 \right] \]

\[ Q_j^i = \frac{(x_i - w_{a_j} | w_{b_j} - w_{a_j})}{L_j} \]

\[ q_j^i = w_{a_j} + (w_{b_j} - w_{a_j}) \frac{Q_j^i}{L_j} \]
Proposed approach: 3 steps

3. After the learning

Some components have a (quasi-) null probability (weights):
They do not explain the data and can be pruned from the initial graph
Threshold setting

\[ \varepsilon = 0.0022 \]

\[ \sim \text{ « Cattell Scree Test »} \]
Toy Experiment

\[ \sigma_{\text{noise}} = 0.05 \]

(a) GGG: \( \sigma^* = 0.06 \) 
(d) CHL: \( T = 0 \) 
(g) CHL: \( T^* = 60 \)

Seuillage sur le nombre de witness
Toy Experiment

\[ \sigma_{noise} = 0.15 \]

(b) GGG: \( \sigma^* = 0.17 \)  
(e) CHL: \( T = 0 \)  
(h) CHL: \( T^* = 65 \)
Toy Experiment

\[ \sigma_{noise} = 0.2 \]

(c) GGG: \( \sigma^* = 0.21 \)   (f) CHL: \( T = 0 \)   (i) CHL: \( T^* = 58 \)
Other applications
Comments

• There is « no free lunch »
  – Time Complexity $O(DN^3)$ (initial Delaunay graph)
  – Slow convergence (EM)
  – Local optima
Key Points

- **Statistical learning of the topology of a data set**
  - Assumption:
    - Initial Delaunay graph is rich enough to contain a sub-graph having the same topology as the principal manifolds
  - Based on a statistical criterion (the likelihood) available in any dimension

- **“Generalized” Gaussian Mixture**
  - Can be seen as a generalization of the “Gaussian mixture” (no edges)
  - Can be seen as a finite mixture (number of “Gaussian-segment”) of an infinite mixture (Gaussian-segment)

- This preliminary work is an attempt to bridge the gap between Statistical Learning Theory and Computational Topology
Open questions

• **Validity of the assumption**
  « good » penalized-likelihood = « good » topology

• Theorem of «universal approximation» of manifold ?
Related works

• Publications NIPS 2005 (unsupervised) and ESANN 2007 (supervised: analysis of the iris and oil flow data sets)

• Workshop submission at NIPS on this topic in collaboration with F. Chazal (INRIA Futurs), D. Cohen-Steiner (INRIA Sophia), S. Canu and G. Gasso (INSA Rouen)
Thanks
\[
p(x, c; \pi, \beta, w, \sigma, DG) = \sum_{d=0}^{1} \sum_{j=1}^{N_d} \pi^d_j \beta^d_{cj} g(x|(d,j); \sigma)
\]
\[
L(\pi, \beta, w, \sigma, DG) = \prod_{i=1}^{M} p(x_i, c_i; \pi, \beta, w, \sigma, DG)
\]
\[
\pi^d_{j[new]} = \frac{1}{M} \sum_{i=1}^{M} p((d,j)|x_i, c_i)
\]
\[
\sigma^2_{new} = \frac{1}{DM} \sum_{i=1}^{M} \left[ \sum_{j=1}^{N_0} p((0,j)|x_i, c_i)(x_i - w_j)^2 
+ \sum_{j=1}^{N_1} p((1,j)|x_i, c_i) \frac{(2\pi\sigma^2)^{-D/2} \exp(-\frac{(x_i - q^j_i)^2}{2\sigma^2})}{L_j \cdot g(x_i|(1,j); \sigma)} \right] 
\]
\[
\beta^d_{cj[new]} = \frac{\sum_{i=1}^{M} p(d,j|x_i, c_i)}{\sum_{i=1}^{M} p(d,j|x_i, c_i)} 
\]
\[
I_1 = \sigma \sqrt{\frac{\pi}{2}} (\text{erf}(\frac{Q_j^i}{\sigma \sqrt{2}}) - \text{erf}(\frac{Q_j^i - L_j}{\sigma \sqrt{2}}))
\]
\[
I_2 = \sigma^2 \left( (Q_j^i - L_j) \exp\left(-\frac{(Q_j^i - L_j)^2}{2\sigma^2}\right) - Q_j^i \exp\left(-\frac{(Q_j^i)^2}{2\sigma^2}\right) \right)
\]
Topology

Number of holes (Betti), connectedness…

Homeomorphism: topological equivalence

\[
d_i = 0 \\
d_i = 1 \\
d_i = 2
\]