Manifold Learning for Complex Visual Analytics: Benefits from and to Neural Architectures

Stephane Marchand-Maillet
Viper group
University of Geneva
Switzerland

Edgar Roman-Rangel, Ke Sun (Viper)
A. Agocs, D. Dardanis, R. Forster, J.-M. Le Goff, X. Ouvrard (CERN)
Outline

• Visual Analytics and Manifold Learning

• Deriving manifold
  – Learning strategies
  – Spacetime
  – Information geometry

• Make Manifold Learning inductive with Neural Architectures

• Application potential: Visualising Neuroscience data
• Choice of features

• Preservation of local information
  – MDS: preserve exact neighborhood
  – t-SNE: preserve neighborhood distribution

• At the heart of visualisation (and *Visual Analytics*)

Preserving local information

Given \( \{x_i\}_{i=1...N} \in \mathbb{R}^D \) find \( \{y_i\}_{i=1...N} \in \mathbb{R}^d \) \( (d < D) \)

- **Distance-based**

\[
\begin{align*}
    d_{ij} &= \|x_i - x_j\| \\
    \delta_{ij} &= \|y_i - y_j\| \Rightarrow \min_y \sum_i \sum_j w_{ij} (d_{ij} - \delta_{ij})^2
\end{align*}
\]

- **Stochastic neighbourhood**

\[
\begin{align*}
    p_{j|i} &= \frac{h\left(\|x_i - x_j\|^2\right)}{\sum_{j \neq i} h\left(\|x_i - x_j\|^2\right)} \\
    q_{j|i} &= \frac{h\left(\|y_i - y_j\|^2\right)}{\sum_{j \neq i} h\left(\|y_i - y_j\|^2\right)}
\end{align*}
\]

\[
E(y) = -\sum_{i=1}^{N} \sum_{j:j \neq i} q_{j|i}(y) \log p_{j|i}
\]

Stochastic Unfolding (SU)

Extension to spacetime

• Use relativistic pseudo-metric tensor for including a “time” (negative) dimension

\[
c(x, y)^2 = \sum_{\text{space}} (x_i - y_i)^2 - \sum_{\text{time}} (x_i - y_i)^2
\]

Provides more power for representation

Visualising Spacetime
A geometric view of Machine Learning

Information Geometry allows us to consider statistical machine learning as geometric operations (e.g., projections) over statistical manifolds.

Embarking Neural Architectures

... as feature extractors

- We use the representation derived internally by Deep Learning architectures as input dimensions

\[ c_5 \text{ in VGGNet} \]

Embarking Neural Architectures

... as mappers

• Manifold Learning techniques are transductive
  – No absolute mapper learnt

→ We use Neural Architectures to make them inductive

A Visual Analytics platform for Big Data 
Case of Neuroscience

A. Agocs, D. Dardanis, R. Forster, J.-M. Le Goff, X. Ouvrard
CERN
The macaque case

**Linear graph model of the network of cortical interactions**

- Derived a second linear graph of the visio-tactile network
- Projection of the derived graph back to the original network
- Characterize the nodes which are responsible for the information transmission

\[ g_0: \text{directed graph of brain area interconnectivity}^* \]
\[ (42 \text{ vertices = areas, 601 edges = interactions}) \]

\[ g_2: \text{directed graph of cortical interactions}^* \]
\[ (9869 \text{ vertices = IPT flows, 166219 edges = common interactions}) \]

*Data/slide: L. Négyessy, A. Fülöp/Wigner Institute, Budapest

\[ g_2 \text{ is too large for visual perception} \rightarrow \text{Communities} \]
\[ 172 \text{ clusters 10668 edges} \]
Macaque brain network data: optimal for navigation
$g_2$ (with Quotient graph)
Conclusion / Outlook

• **Visual Analytics** both inherits from and complements **Machine Learning**

• Neural Architectures are flexible tools to learn non-linear processes

• Their integration in Learning processes can be diverse

• The parallel with **understanding neurological processes** may still have a lot to offer