Nonlinear Dimensionality Reduction

Chapter 5.3 + Appendixes D and E

Antti Sorjamaa and Yoan Miche

Time Series Prediction and ChemoInformatics Group
Adaptive Informatics Research Centre
Helsinki University of Technology
Outline

- Vector Quantization
- Graph Building

Chapter 5.3

- Locally Linear Embedding (LLE)
- Laplacian Eigenmaps (LE)
- Isotop
Vector Quantization

- Aim is to decrease the amount of data
- Many methods use VQ as preprocessing method
- Used in many fields
  - Data analysis and compression
  - Clustering and classification
  - Telecommunications
Vector Quantization

- Aim is to decrease the amount of data
Vector Quantization (2)

Quantization distortion

\[ E_{VQ} = \frac{1}{N} \sum_{i=1}^{N} \| y(i) - \text{dec}(\text{cod}(y(i))) \|^2 \]

- **Coding** denotes a function for finding the Best Matching Unit (BMU) of the data point.
- **Decoding** denotes a function for replacing the data point with the BMU.
Vector Quantization (3)

- **K-means**

  \[
  c(j) \leftarrow \frac{1}{V_j} \sum_{y(i) \in V_j} y(i)
  \]

- **Stochastic Gradient Descent**

  \[
  \xi(j) \leftarrow \xi(j) - \alpha(\xi(j) - x_i)
  \]
Vector Quantization (4)

- **Static** $\rightarrow$ fixed number of prototypes
  - Classical techniques
    - $K$-means, LBG
  - Competitive Learning
    - Winner takes all (Stochastic Gradient Descent)
    - Winner takes most (SOM and Neural Gas)

- **Incremental** $\rightarrow$ increasing

- **Dynamic** $\rightarrow$ increasing and decreasing
Graph Building

Yoan presents...
Isotop

Close to SOM algorithm except

- Vector Quantization optional
- No predefined latent space
- Data points are not used in the learning phase, but instead a graph of them
- No online version
- More "data-driven" methodology
Isotop Algorithm

- Optional Vector Quantization
- Build graph with pairwise distances
- Initialize all nodes to zero
- Calculate parameters with respect to $q$
- For each node
  1. Generate random point around the node
  2. Select closest node
  3. Update coordinates of all nodes according to neighborhood
- Increase $q$ and goto step 4 unless converged
Isotop Algorithm (2)

1) Initialize all nodes to zero
   – Place Gaussian kernel of unit variance on each node, $N(x(i), I)$

n Calculate parameters with respect to $q$
   – Learning rate $\alpha$
   – Neighborhood width $\lambda$
For each node from 1 to $N$

- Generate random point $r$ from the distribution of the node $N(x(i), I) \rightarrow r$
- Calculate the nearest node
- Update all nodes

$x(i) \leftarrow x(i) - \alpha \nu_{\lambda}(i, j)(r - x(i))$

$\nu_{\lambda}(i, j) = \exp \left( - \frac{1}{2} \frac{\delta^2_y(i, j)}{\frac{\lambda^2}{2} \mu^2(v_h, v_j) \in E(\delta_y(h, j))} \right)$
Notes of Convergence

- "Nodes won’t collapse"
  - Gaussian centers make boundary nodes expand the graph
  - Neighborhood size must be kept reasonable

- "Nodes won’t disperse infinitely"
  - Update rule generates an attractive force
  - Neighborhood size larger than zero
Questions?

Antti.Sorjamaa@hut.fi

http://www.cis.hut.fi/projects/tsp