Neighbor Embedding XOM for Dimension Reduction and Visualization

Kerstin Bunte, Barbara Hammer, Thomas Villmann, Michael Biehl, Axel Wismüller

Intelligent Systems Group
Mathematics and Computing Science
University of Groningen
The Netherlands

April 28, 2010

Introduction

Dimension Reduction

- Curse of dimensionality
- Feature extraction
- Reduce redundancies
- Visualization

Introduction

Dimension Reduction

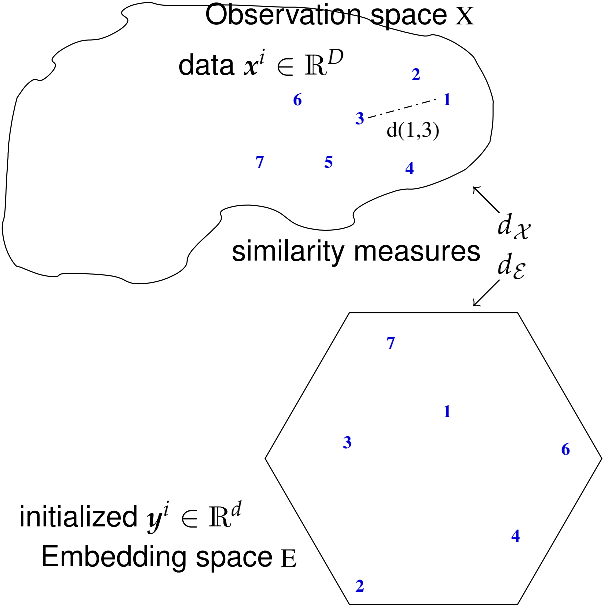
- Curse of dimensionality
- Feature extraction
- Reduce redundancies
- Visualization

Motivation

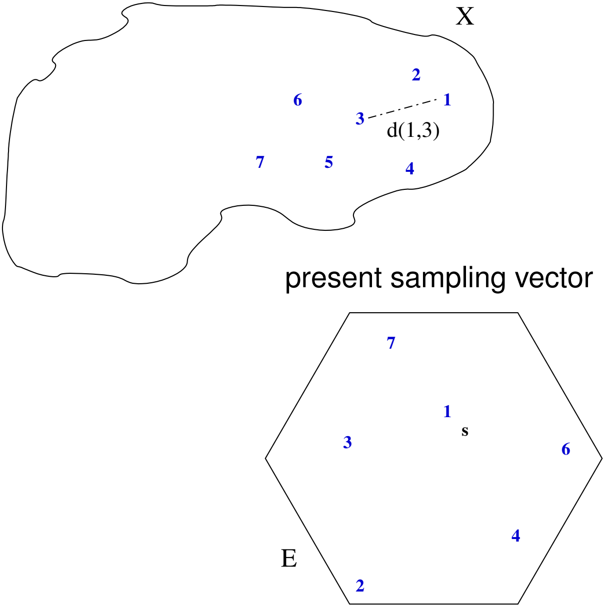
- Most methods are quadratic with number of points
- Conceptual links:

fast online learning - direct divergence optimization

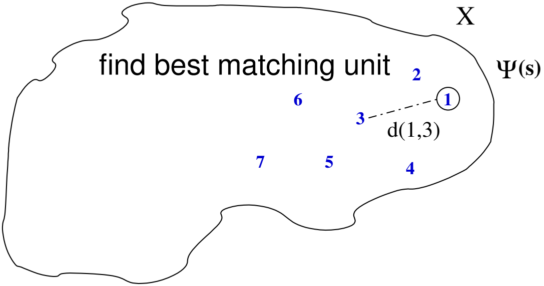
Exploratory Observation Morphogenesis (XOM)



Exploratory Observation Morphogenesis (XOM)

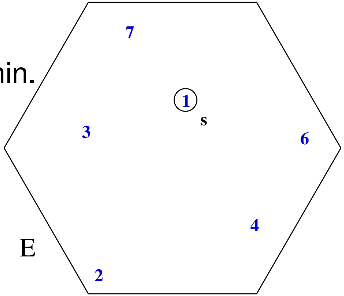


Exploratory Observation Morphogenesis (XOM)

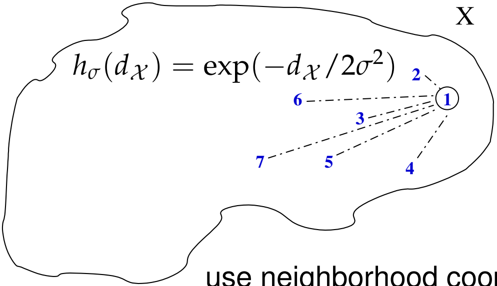


$\Psi(s) = x^i$ where

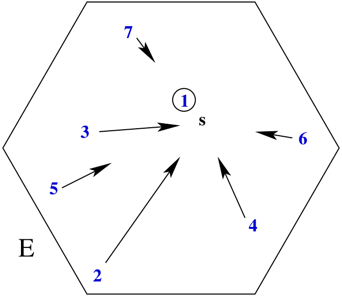
$d_{\mathcal{E}}(s, y^i)$ is min.



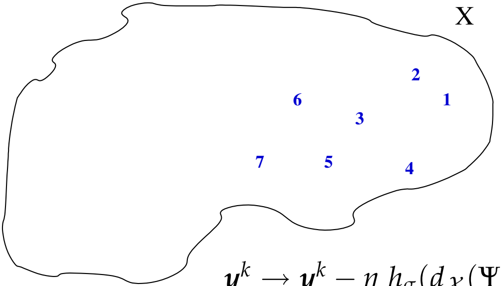
Exploratory Observation Morphogenesis (XOM)



use neighborhood cooperation . . .

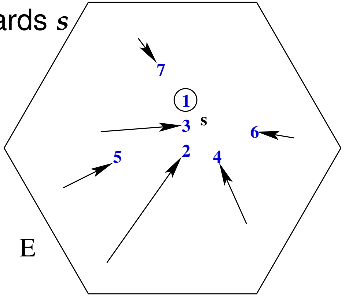


Exploratory Observation Morphogenesis (XOM)

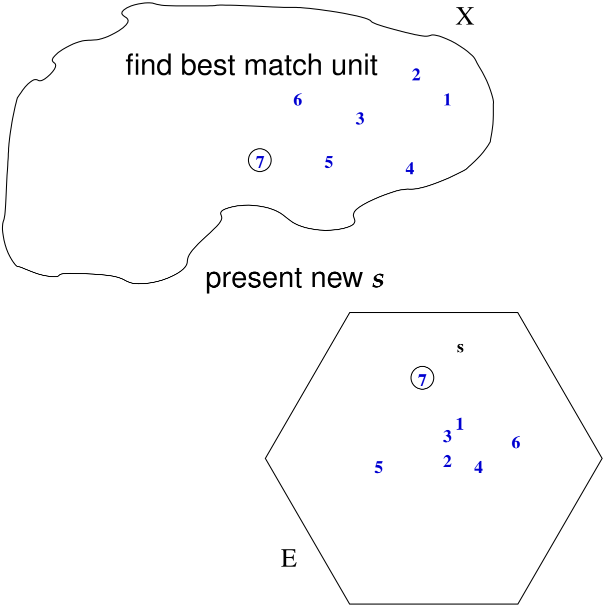


$$\boldsymbol{y}^k \to \boldsymbol{y}^k - \eta\, h_\sigma(d_{\mathcal{X}}(\Psi(\boldsymbol{s}), \boldsymbol{x}^k)) \frac{\partial d_{\mathcal{E}}(\boldsymbol{s}, \boldsymbol{y}^k)}{\partial \boldsymbol{y}^k}$$

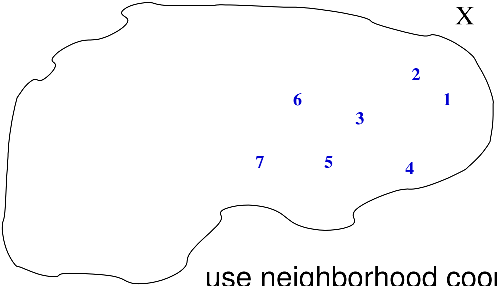
… to update $\boldsymbol{y}^k$ towards $\boldsymbol{s}$



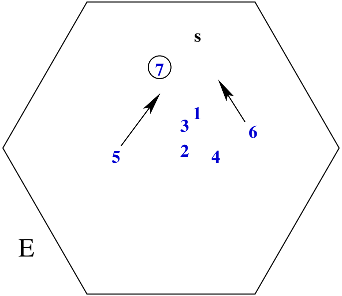
Exploratory Observation Morphogenesis (XOM)



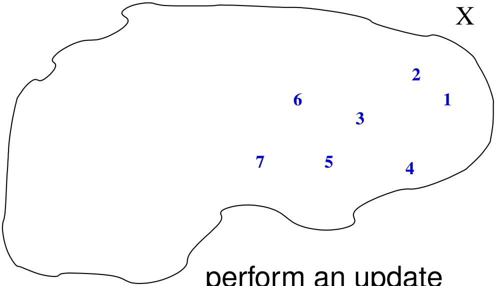
Exploratory Observation Morphogenesis (XOM)



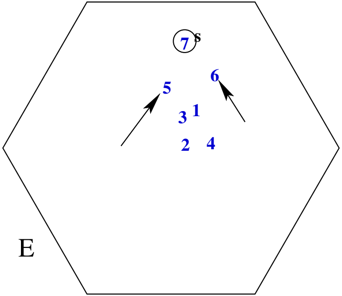
use neighborhood cooperation



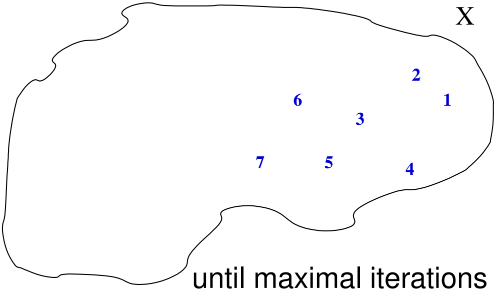
Exploratory Observation Morphogenesis (XOM)



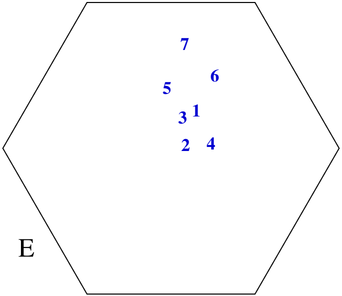
perform an update



Exploratory Observation Morphogenesis (XOM)



until maximal iterations reached



Cost function

Variation of the winner definition following Heskes:

$$\Psi(\boldsymbol{s}) = \boldsymbol{x}^i \text{ where } \sum_j h_\sigma(d_\mathcal{X}(\boldsymbol{x}^i, \boldsymbol{x}^j)) d_\mathcal{E}(\boldsymbol{s}, \boldsymbol{y}^j) \ ,$$

allows to define a cost function for the XOM:

$$E_{\text{XOM}} \sim \int \sum_i \delta_{\Psi(\boldsymbol{s}),\boldsymbol{x}^i} \sum_{j=1}^N h_\sigma(d_\mathcal{X}(\boldsymbol{x}^i, \boldsymbol{x}^j)) d_\mathcal{E}(\boldsymbol{s}, \boldsymbol{y}^j) p(\boldsymbol{s}) d\boldsymbol{s}$$

δ denotes the Kronecker delta

New XOM inspired cost function

Generalized Kullback Leibler (KL) divergence

$$E_{GKL}(p \,||\, q) = \int \left[p(\boldsymbol{x}) \log \left(\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})} \right) \right] - [p(\boldsymbol{x}) - q(\boldsymbol{x})]\, d\boldsymbol{x}$$

$$0 \leq p(\boldsymbol{x}), q(\boldsymbol{x}) \leq 1$$

Neighborhood functions $h_\sigma(d_{\mathcal{X}}(\boldsymbol{x}^i, \boldsymbol{x}^j))$ and $g_\gamma(d_{\mathcal{E}}(\boldsymbol{s}, \boldsymbol{y}^j))$

$$E_{\text{KLX}} \sim \int \sum_i \delta_{\Psi^{\text{GKL}}(\boldsymbol{s}), \boldsymbol{x}^i} \sum_j \left[h_\sigma^{ij} \ln \left(\frac{h_\sigma^{ij}}{g_\gamma^j} \right) - h_\sigma^{ij} + g_\gamma^j \right] p(\boldsymbol{s}) d\boldsymbol{s}$$

New XOM inspired Learning Rules

Derive the online learning rule from $\frac{\partial E_{\mathrm{KLX}}}{\partial \boldsymbol{y}^k}$ for a given sampling vector $\boldsymbol{s}$:

$$\boldsymbol{y}^k = \boldsymbol{y}^k - \eta \frac{\partial g_\gamma^k}{\partial \boldsymbol{y}_k} \left(1 - \frac{h_\sigma^{ik}}{g_\gamma^k}\right)$$

New XOM inspired Learning Rules

Derive the online learning rule from $\frac{\partial E_{\mathrm{KLX}}}{\partial \boldsymbol{y}^k}$ for a given sampling vector $\boldsymbol{s}$:

$$\boldsymbol{y}^k = \boldsymbol{y}^k - \eta \frac{\partial g_\gamma^k}{\partial \boldsymbol{y}_k} \left(1 - \frac{h_\sigma^{ik}}{g_\gamma^k}\right)$$

In case of Gaussian neighborhood function $g_\gamma^j = \exp\left(\frac{-d_{\mathcal{E}}(\boldsymbol{s}, \boldsymbol{y}^j)}{2\gamma^2}\right)$:

$$\boldsymbol{y}^k = \boldsymbol{y}^k - \frac{\eta}{\gamma^2} \frac{\partial d_{\mathcal{E}}(\boldsymbol{s}, \boldsymbol{y}^k)}{\partial \boldsymbol{y}^k} \left(h_\sigma^{ik} - g_\gamma^k\right)$$

New XOM inspired Learning Rules

Derive the online learning rule from $\frac{\partial E_{\text{KLX}}}{\partial \boldsymbol{y}^k}$ for a given sampling vector $\boldsymbol{s}$:

$$\boldsymbol{y}^k = \boldsymbol{y}^k - \eta \frac{\partial g_\gamma^k}{\partial \boldsymbol{y}_k}\left(1 - \frac{h_\sigma^{ik}}{g_\gamma^k}\right)$$

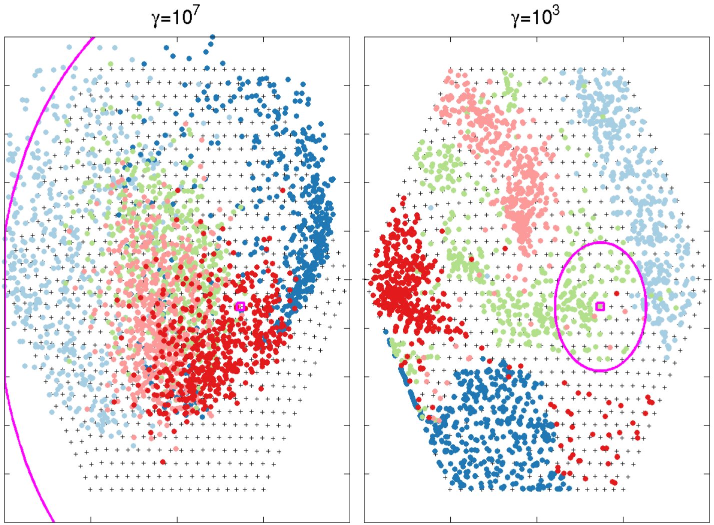
In case of Gaussian neighborhood function $g_\gamma^j = \exp\left(\frac{-d_{\mathcal{E}}(\boldsymbol{s},\boldsymbol{y}^j)}{2\gamma^2}\right)$:

$$\boldsymbol{y}^k = \boldsymbol{y}^k - \frac{\eta}{\gamma^2}\frac{\partial d_{\mathcal{E}}(\boldsymbol{s},\boldsymbol{y}^k)}{\partial \boldsymbol{y}^k}\left(h_\sigma^{ik} - g_\gamma^k\right)$$

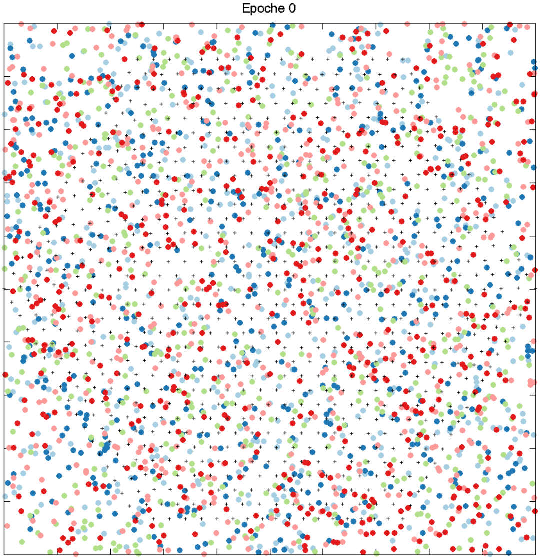
In case of t-distributed $g_\gamma^j = \left(1 + d_{\mathcal{E}}(\boldsymbol{s},\boldsymbol{y}^j)/\gamma\right)^{\left(-\frac{\gamma+1}{2}\right)}$:

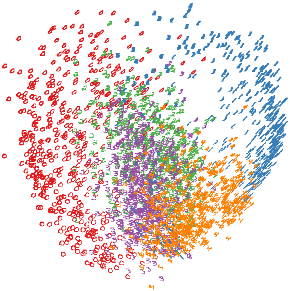
$$\boldsymbol{y}^k = \boldsymbol{y}^k - \frac{\eta}{2}\frac{\partial d_{\mathcal{E}}(\boldsymbol{s},\boldsymbol{y}^k)}{\partial \boldsymbol{y}^k}\alpha\left(h_\sigma^{ik} - g_\gamma^k\right) \rightarrow \alpha = \frac{1}{\left(1 + \frac{d_{\mathcal{E}}(\boldsymbol{s},\boldsymbol{y}^k)}{\gamma}\right)}$$

Influence of γ in t-NEXOM

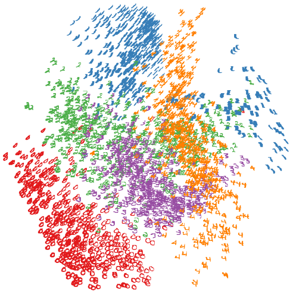


Training video



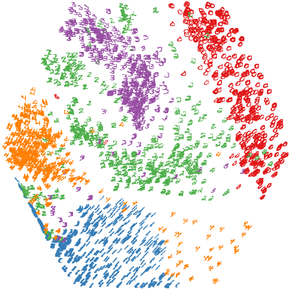


(a) PCA (37.7%)



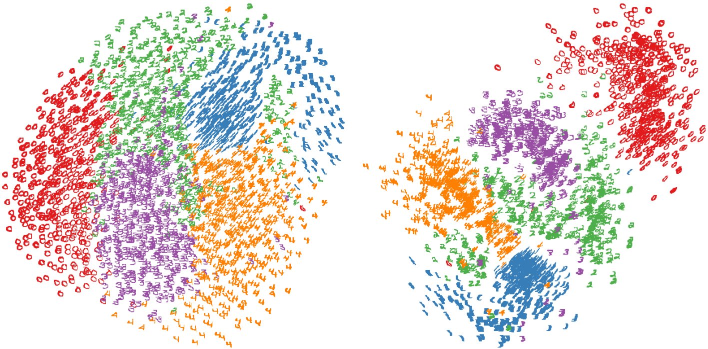
(b) NEXOM (23.8%)

USPS Digits



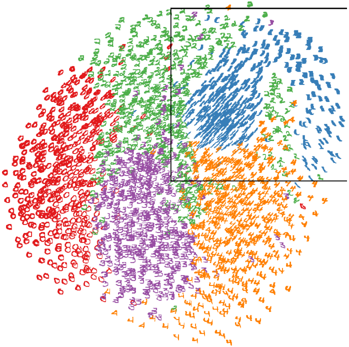
(c) t-NEXOM (4.6%)

Without structure hypothesis

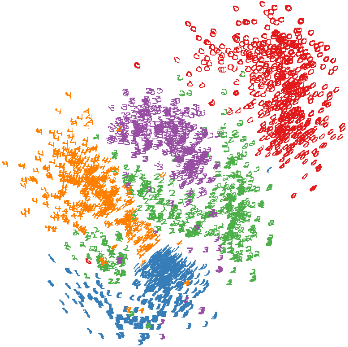


(d) NEXOM (14.4%) (e) t-NEXOM (4.5%)

Without structure hypothesis

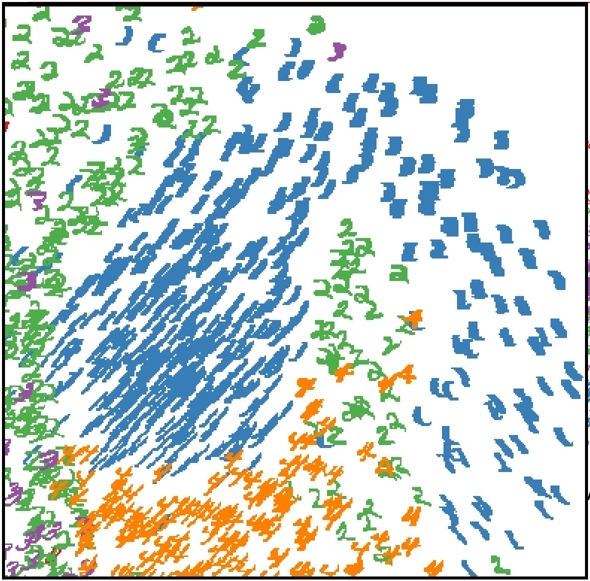


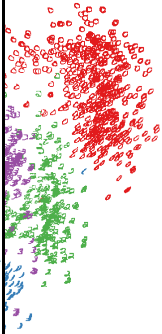
(f) NEXOM (14.4%)



(g) t-NEXOM (4.5%)

Without structure hypothesis





M (4.5%)

Summary

Neighbor Embedding XOM:

- conceptually links between
  - fast sequential online learning
  - direct divergence optimization
- competitive trade-off:
  - high embedding quality
  - low computational expense
- different distributions and learning rules