Nonlinear dimensionality reduction viewed as information retrieval

Jarkko Venna and Samuel Kaski

Helsinki Institute for Information Technology & Adaptive Informatics Research Centre
Laboratory of Computer and Information Science
Helsinki University of Technology
P.O. Box 5400, FI-02015 TKK - Finland

Summary

1. The application:
   Visual exploration of the neighborhood or proximity relationships in the data in functional genomics, image retrieval etc.
2. Preserved property:
   Probability of data point i being a neighbor of j in the input and output space.
3. Ideally we would like to preserve:
   Primary goal: All nearest neighbors of a point should be the same before and after dimensionality reduction. Secondary goals: Order within the neighborhood should be preserved and there should not be too drastic distortions in the distances overall.
4. Was the application goal a substantial part of the dimensionality reduction method:
   Yes, the dimensionality reduction method was primarily designed to take into account the application goals.

Visual exploration of neighborhood relationships in data

We view information visualization from the user perspective, as an information retrieval problem. Assuming that the task of the user is to understand the proximity relationships in the original high-dimensional data set, the task of the visualization algorithm is to construct a display that helps in this task. For a given data point, the user wants to know which other data points are its neighbors, and the visualization should reveal this for all data points, as well as possible.

The goal is to make the data set more understandable, by making the similarity relationships between data points explicit through visualizations.

Application areas:

- Visual exploration of a new data set. Do the data points make homogenous areas or clusters? Are there clear trends in the data.
- Creating preliminary hypotheses on unknown data based on known samples. Are there continuous areas with only unknown data? Are there unknown data points that are similar (proximate) to known data points.
- Interfaces to high-dimensional data stores.

Background: Stochastic Neighbor embedding [1]

- Define the probability p_{ij} of the point i being a neighbor of point j in the input space and q_{ij} output space as

$$p_{ij} = \frac{\exp\left(-\frac{d(\mathbf{x}_i,\mathbf{x}_j)^2}{\sigma_i^2}\right)}{\sum_{k\neq i}\exp\left(-\frac{d(\mathbf{x}_i,\mathbf{x}_k)^2}{\sigma_i^2}\right)}, \qquad q_{ij} = \frac{\exp\left(-\frac{\|\mathbf{y}_i-\mathbf{y}_j\|^2}{\sigma_i^2}\right)}{\sum_{k\neq i}\exp\left(-\frac{\|\mathbf{y}_i-\mathbf{y}_k\|^2}{\sigma_i^2}\right)}. \tag{1}$$

- The SNE algorithm searches for the configuration of points $\mathbf{y}_i$ that minimizes the KL-divergence D between the probability distributions in the input and output spaces, averaged over all points. The cost function is

$$E_{\mathrm{SNE}} = E_i[D(p_i,q_i)] \propto \sum_i D(p_i,q_i) = \sum_i \sum_{j\neq i} p_{ij}\log\frac{p_{ij}}{q_{ij}}, \tag{2}$$

where E_i is the average over data samples i.

Retrieval of Neighbors

- We assume the k closest points to be neighbors with a high probability and the rest with a very low probability.

$$p_{ij} = \begin{cases} a \equiv \frac{1-\delta}{k}, & \text{if point } j \text{ is among the } k \text{ nearest neighbors of } i \text{ in the input space} \\ b \equiv \frac{\delta}{N-k-1}, & \text{otherwise} \end{cases} \tag{3}$$

and similarly in the output space

$$q_{ij} = \begin{cases} c \equiv \frac{1-\delta}{r}, & \text{if point } j \text{ is among the } r \text{ nearest neighbors of } i \text{ in the visualization} \\ d \equiv \frac{\delta}{N-r-1}, & \text{otherwise} \end{cases} \tag{4}$$

where r is the neighborhood size in the output space.

- Each KL-divergence between the two distributions in SNE can be divided into four parts:

$$D(p_i,q_i) = \sum_{p_{ij}=a,q_{ij}=c} a\log\frac{a}{c} + \sum_{p_{ij}=a,q_{ij}=d} a\log\frac{a}{d} + + \sum_{p_{ij}=b,q_{ij}=c} b\log\frac{b}{c} + \sum_{p_{ij}=b,q_{ij}=d} b\log\frac{b}{d} \tag{5}$$

$$= C_{\mathrm{TP}}N_{\mathrm{TP}} + C_{\mathrm{MISS}}N_{\mathrm{MISS}} + C_{\mathrm{FP}}N_{\mathrm{FP}} + C_{\mathrm{TN}}N_{\mathrm{TN}}. \tag{6}$$

- If δ is very small then this can be approximated:

$$D_{KL}(p_i,q_i) \approx \frac{N_{MISS}}{k}C. \tag{7}$$

- Minimizing Eq. 7 is the same as maximizing recall

$$\text{recall} = \frac{N_{TP}}{k} = 1 - \frac{N_{MISS}}{k}. \tag{8}$$

- **In summary: A new finding that SNE maximizes a smoothed form of recall.**

Abstract

Nonlinear dimensionality reduction has so far been treated either as a data representation problem or as a search for a lower-dimensional manifold embedded in the data space. A main application for both is information visualization, to make visible the neighborhood or proximity relationships in the data, but neither approach has been designed to optimize this task. We give such visualization a new conceptualization as an information retrieval problem; a projection is good if neighbors of data points can be retrieved well based on the visualized projected points. This makes it possible to rigorously quantify goodness in terms of precision and recall. A method is introduced to optimize retrieval quality; it turns out to be an extension of Stochastic Neighbor Embedding, one of the earlier nonlinear projection methods, for which we give a new interpretation: it optimizes recall.

Neighbor Retrieval Visualizer (NeRV)

- Reversing the KL-divergence in Eq. 6 results in

$$D(q_i,p_i) \approx \frac{N_{FP}}{r}C, \tag{9}$$

- Minimizing Eq. 7 is the same as maximizing precision

$$\text{precision} = 1 - \frac{N_{FP}}{r}. \tag{10}$$

- By making a λ-parameterized combination of the (smoothed) precision and recall, we get:

$$E_{\mathrm{NeRV}} = \lambda E_i[D(p_i,q_i)] + (1-\lambda)E_i[D(q_i,p_i)] = \lambda\sum_i\sum_{j\neq i}p_{ij}\log\frac{p_{ij}}{q_{ij}} + (1-\lambda)\sum_i\sum_{j\neq i}q_{ij}\log\frac{q_{ij}}{p_{ij}}. \tag{11}$$

- We call the new method that optimizes (11) *Neighbor Retrieval Visualizer (NeRV)*.

Computational complexity

The computational complexity of the gradient step in the NeRV algorithm is $\mathcal{O}(n^3)$.

NeRV has two faster alternatives with a computational complexity of $\mathcal{O}(n^2)$:

- fNeRV (unpublished), is based on an information-geometrically motivated approximation of the NeRV cost function.
- LocalMDS [4], is a more heuristically motivated algorithm.

Experimental Results

Projection of a sphere

$\lambda = 0$ $\lambda = 0.8$ $\lambda = 1.0$

Three projections of data that lies on the surface of a 3D sphere. **Left** The projection tries to maximize precision. **Middle** The projection is a compromise between precision and recall. **Right** The projection tries to maximize recall.

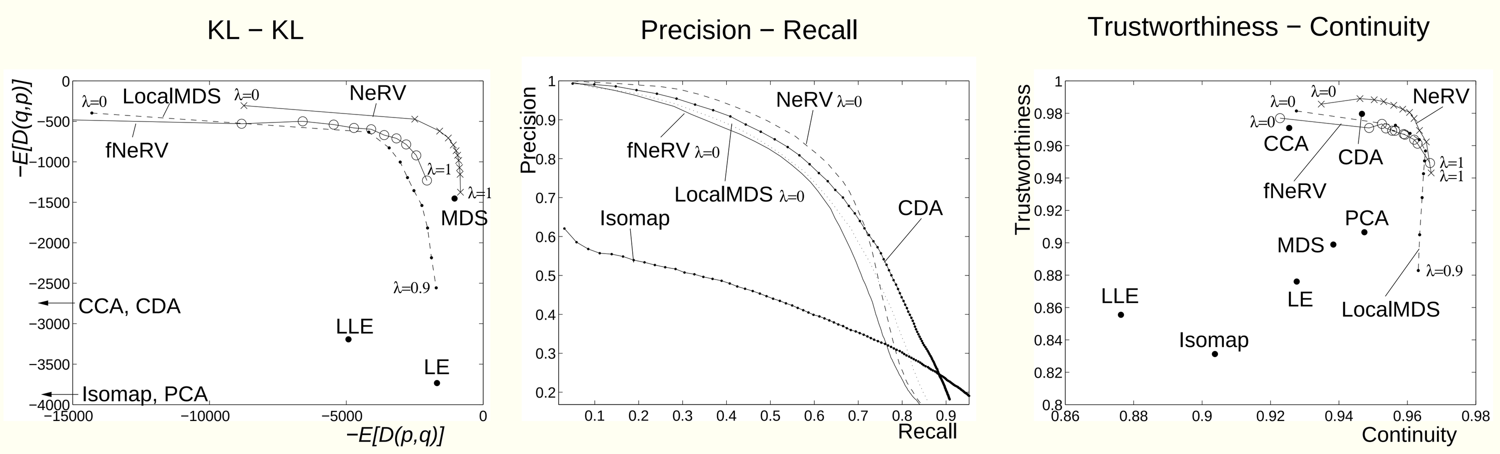
For more information

jarkko.venna@tkk.fi samuel.kaski@tkk.fi

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

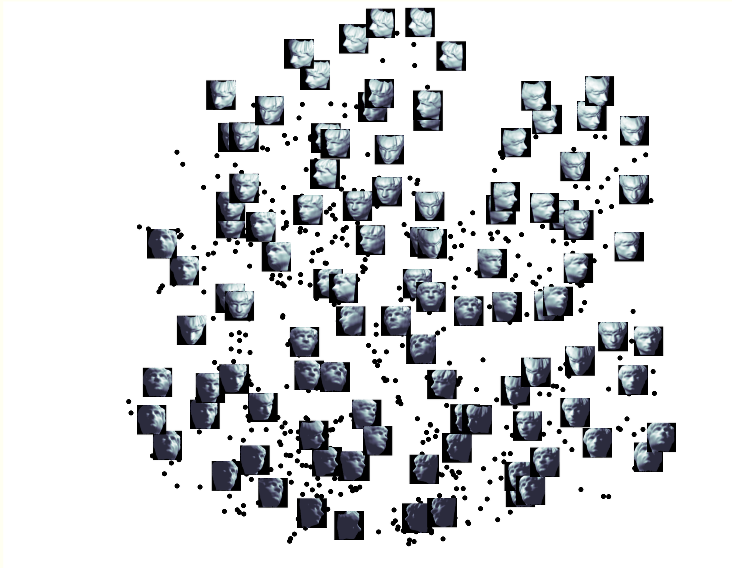
Experimental Results

Projecting face images

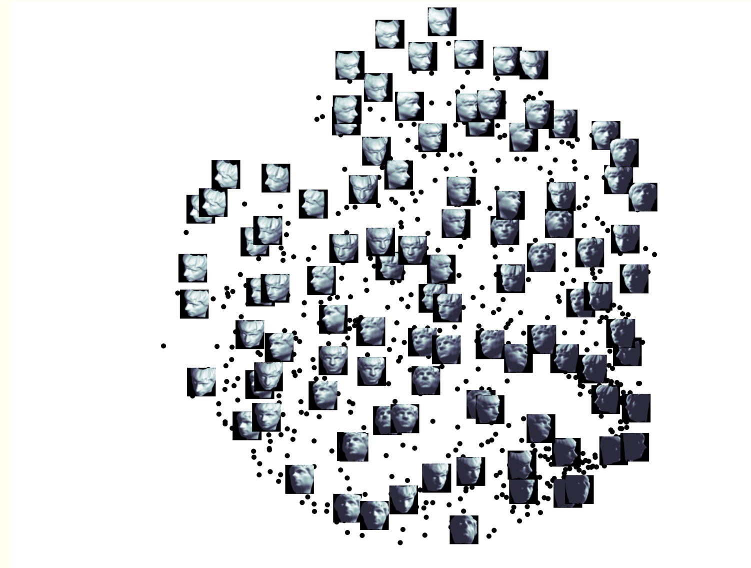


Results on projecting the face dataset [3] to two dimensions. KL-KL curves (left), precision–recall curves (middle) and trustworthiness–continuity [2] curves (right) as a function of λ. Other nonlinear projection methods have been added for reference. The precision–recall curves have been calculated with 20 nearest neighbors in the input space as the set of relevant items; the number of retrieved items (neighbors) is varied from 1 to 100. Only the reference method that achieved the highest precision and the highest recall, and the λ values that had the largest area under the curve are included for clarity. The KL–KL curve and the trustworthiness-continuity curve are calculated using 20 nearest neighbors. On each plot the best performance is in the top right corner. Methods; NeRV, LocalMDS, fNeRV: a faster approximative version of NeRV, PCA: Principal Component Analysis, MDS: metric Multidimensional Scaling, LLE: Locally Linear Embedding, LE: Laplacian Eigenmap, CCA: Curvilinear Component Analysis, CDA: CCA using geodesic distances and Isomap.

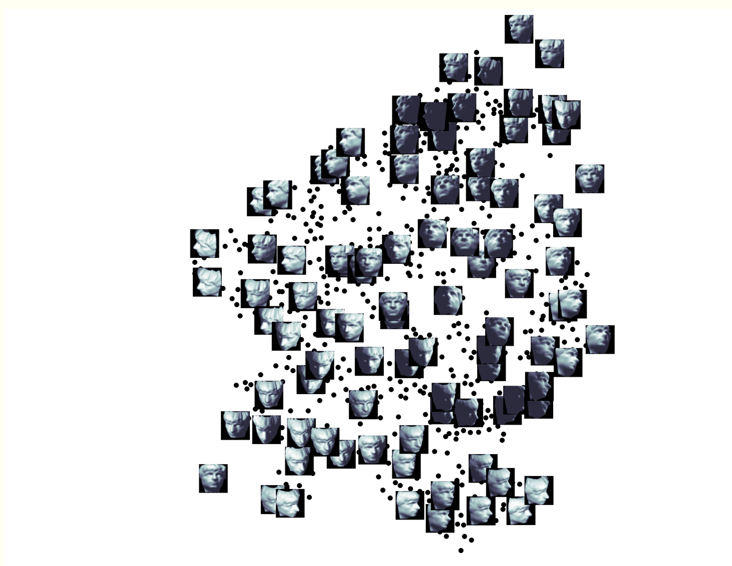
Visualizations of the face data set



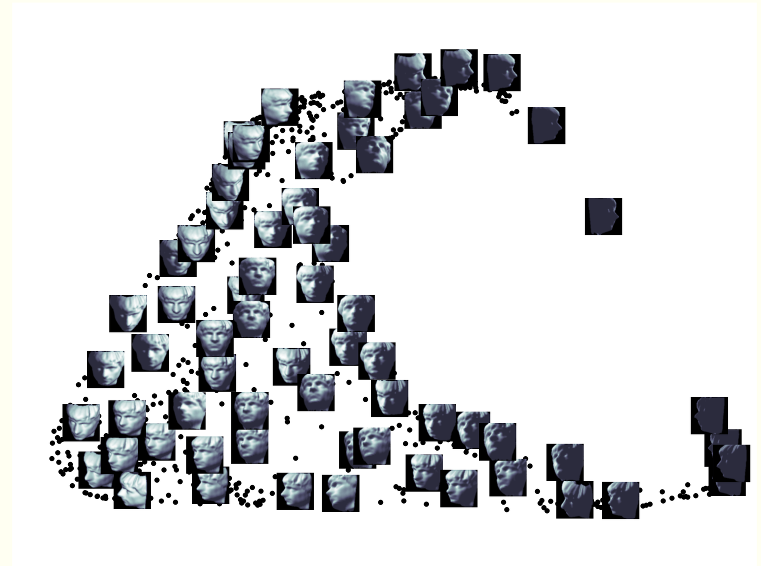
NeRV $\lambda = 0.1$



MDS



CDA



Laplacian Eigenmap

References

[1] Geoffrey Hinton and Sam Roweis. Stochastic neighbor embedding. In *Advances in Neural Information Processing Systems 15*, pages 833–840. MIT Press, 2002.

[2] Samuel Kaski, Janne Nikkilä, Merja Oja, Jarkko Venna, Petri Törönen, and Eero Castrén. Trustworthiness and metrics in visualizing similarity of gene expression. *BMC Bioinformatics*, 4:48, 2003.

[3] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, 2000.

[4] Jarkko Venna and Samuel Kaski. Local multidimensional scaling. *Neural Networks*, 19:889–899, 2006.