Non-linear Dimensionality Reduction

t-Distributed Stochastic Neighbor Embedding

Xiaoyu Xue

Microsoft BingAds UCM

Table of contents

Dimensionality Reduction

Dimensionality Reduction

Definition of dimensionality reduction:

Given a set of data $X = [\mathbf{x}_1, \ldots, \mathbf{x}_m]$, $\mathbf{x}_i \in \mathbb{R}^n$, find a map $f : \mathbb{R}^n \rightarrow \mathbb{R}^d$, make $\mathbf{y}_i = f(\mathbf{x}_i)$ and $d \ll n$. Where $f = (f_1, \ldots, f_n)$, $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$

Linear Dimensionality Reduction

If f_i is a linear map, $f = P = [\mathbf{p}_1, \ldots, \mathbf{p}_n], \mathbf{y}_i = P^T \mathbf{x}_i$

Dimensionality reduction: Some Assumptions

1. High-dimensional data often lies on or near a much lower dimensional, curved manifold
2. A good way to represent data points is by their low-dimensional coordinates.
3. The low-dimensional representation of the data should capture information about high-dimensional pairwise distances.

Dimensionality Reduction

- **Linear Dimensionality Reduction**: PCA(Principal Components Analysis), LDA(Linear Discriminant Analysis), MDS(Classical Multidimensional Scaling)
- **None-Linear Dimensionality Reduction**: Isomap(Isometric Mapping), LLE(Locally Linear Embedding), LE(Laplacian Eigenmaps), **tSNE**(t-Distributed Stochastic Neighbor Embedding)

Stochastic Neighbor Embedding

Stochastic Neighbor Embedding

Define the similarity of data point $\mathbf{x}_i$ in original space as conditional probability $p_{j|i}$. It is the probability that $\mathbf{x}_i$ would pick $\mathbf{x}_j$ as its neighbor under a Gaussian centered at $\mathbf{x}_i$

$$p_{j|i} = \frac{\exp\left(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma_i^2\right)}{\sum_{k\neq i}\exp(-\|\mathbf{x}_i - \mathbf{x}_k\|^2/2\sigma_i^2)}$$

In low-dimensional space, define the similarity $q_{j|i}$

$$q_{j|i} = \frac{\exp\left(-\|\mathbf{y}_i - \mathbf{y}_j\|^2\right)}{\sum_{k\neq i}\exp(-\|\mathbf{y}_i - \mathbf{y}_k\|^2)}$$

Cost function of SNE

If the map points $\mathbf{y}_i$ and $\mathbf{y}_j$ correctly model the similarity between the high-diminsional datapoints $\mathbf{x}_i$ and $\mathbf{x}_j$, the conditional probability $p_{j|i}$ and $q_{j|i}$ will be equal. Use the Kullback-Leibler divergences to minimize the mismatch:

$$Cost = \sum_i KL(P_i||Q_i) = \sum_i \sum_j p_{j|i} \log \frac{p_{j|i}}{q_{j|i}}$$

To minimize the cost function using gradient descent:

$$\frac{\partial Cost}{\partial \mathbf{y}_i} = 2\sum_j (p_{j|i} - q_{j|i} + p_{i|j} - q_{i|j})(\mathbf{y}_i - \mathbf{y}_j)$$

Stochastic Neighbor Embedding

About the cost function:

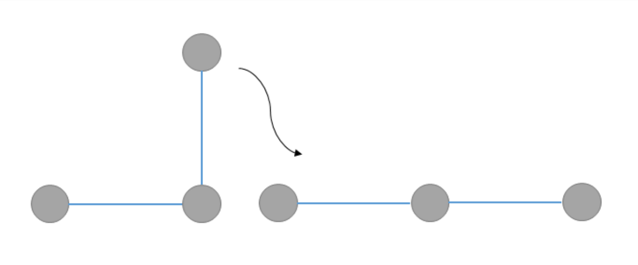
$$Cost = \sum_i KL(P_i||Q_i) = \sum_i \sum_j p_{j|i} \log \frac{p_{j|i}}{q_{j|i}}$$

- Large $p_{j|i}$ modeled by small $q_{j|i}$, Big penalty !
- Small $p_{j|i}$ modeled by large $q_{j|i}$, Small penalty !
- It is asymmetric and mainly preserves local similarity structure of data !

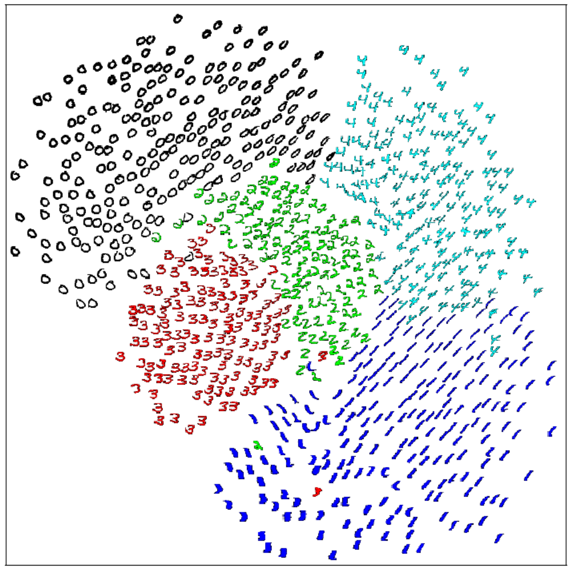
Stochastic Neighbor Embedding

The "Crowding" problem

- Try to model the local structure of data in the map !
- Dissimilar points have to be modeled as too far apart in the map !
- SNE does not have gaps between classes !



Stochastic Neighbor Embedding



t-Distributed Stochastic Neighbor Embedding

t-Distributed Stochastic Neighbor Embedding

Symmetric SNE by using joint probability distribution instead of conditional probability distribution.

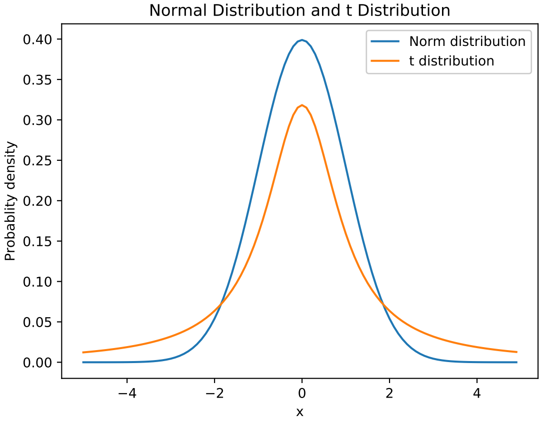
$$p_{ij} = \frac{p_{i|j} + p_{j|i}}{2m}$$

Cost function becomes:

$$Cost = KL(P||Q) = \sum_{i} \sum_{j} p_{ji} \log \frac{p_{ji}}{q_{ji}}$$

t-Distributed Stochastic Neighbor Embedding

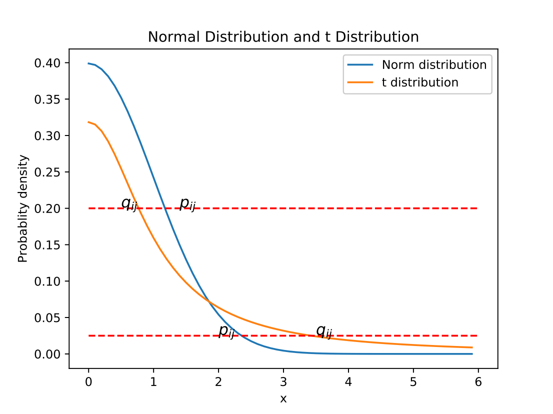
Use t distribution (Heavy-tailed distribution) to model similarity in low-dimensional space to release the "Crowding" problem



t-Distributed Stochastic Neighbor Embedding

The joint probabilities q_{ij} are defined as

$$q_{ij} = \frac{(1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2)^{-1}}{\sum_{k!=l}(1 + \|\mathbf{y}_k - \mathbf{y}_l\|^2)^{-1}}$$



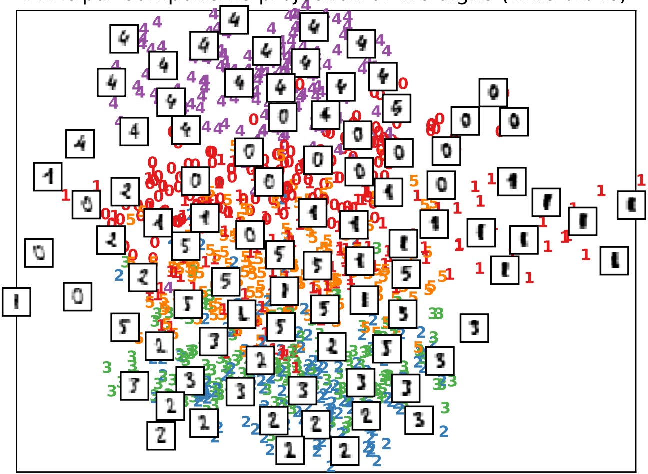
Hardwriting digit visualization

Hardwriting digit visualization

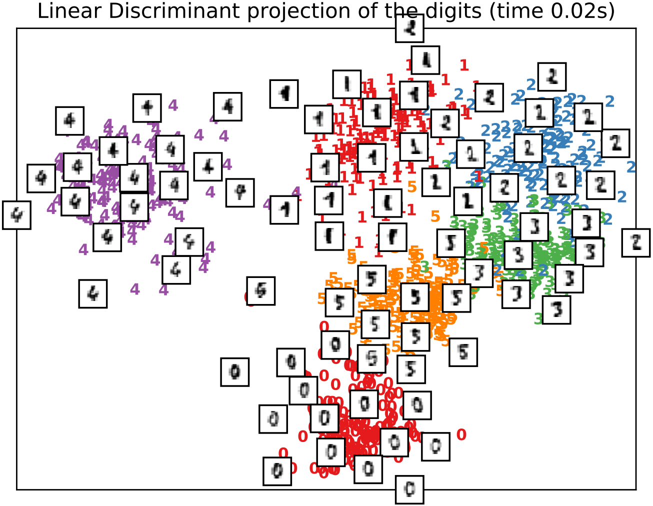
A selection from the 64-dimensional digits dataset

Hardwriting digit visualization

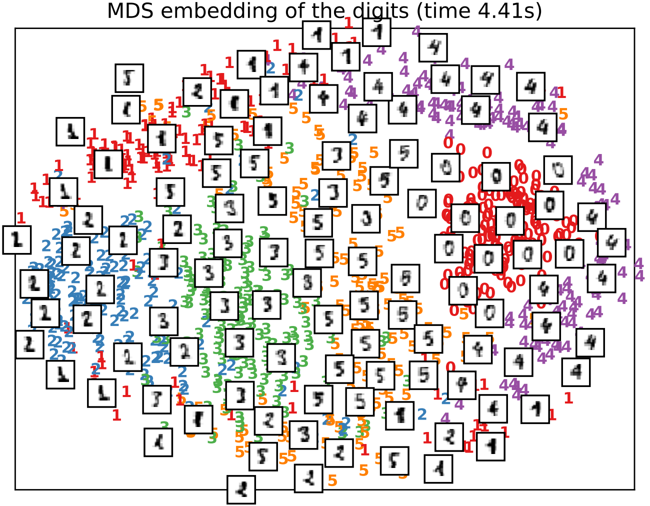
Principal Components projection of the digits (time 0.04s)



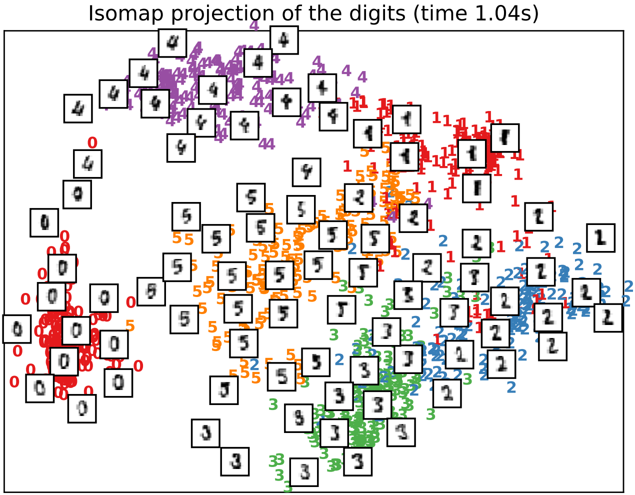
Hardwriting digit visualization



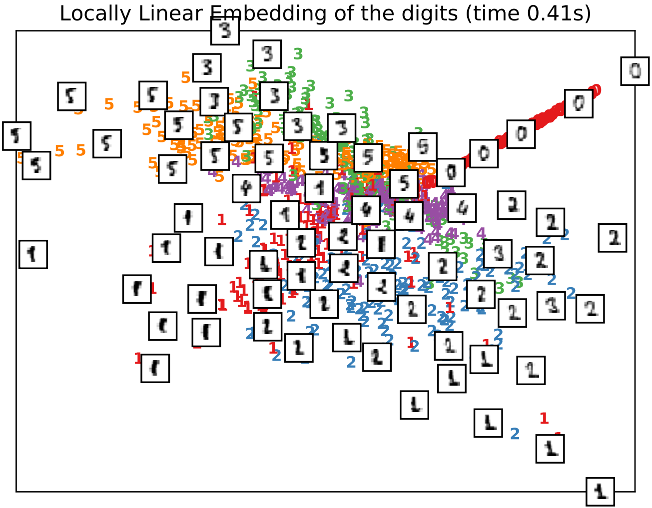
Hardwriting digit visualization



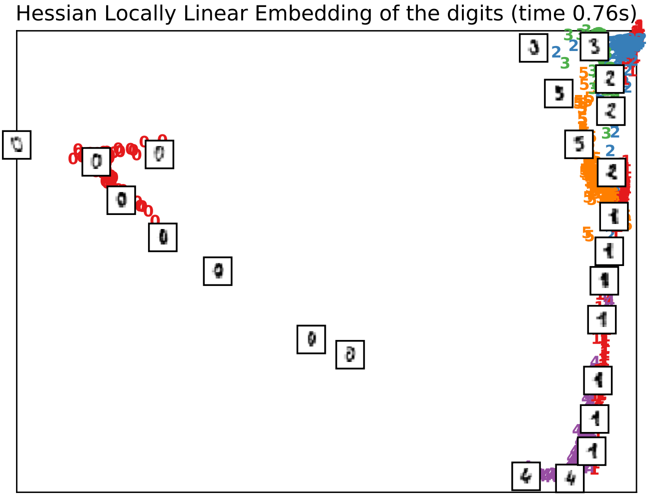
Hardwriting digit visualization



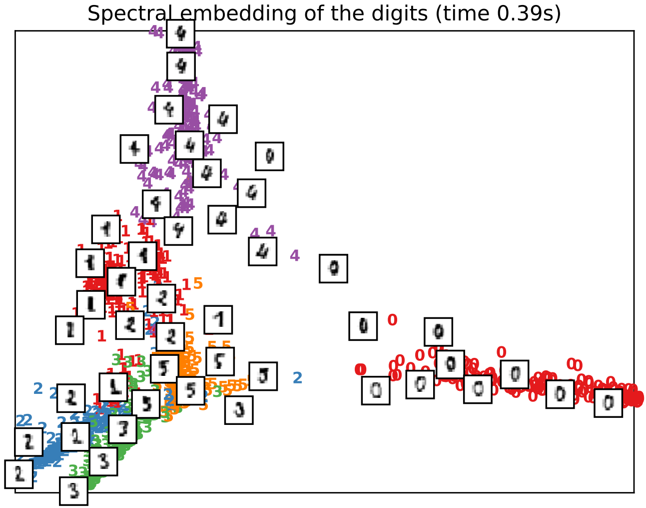
Hardwriting digit visualization



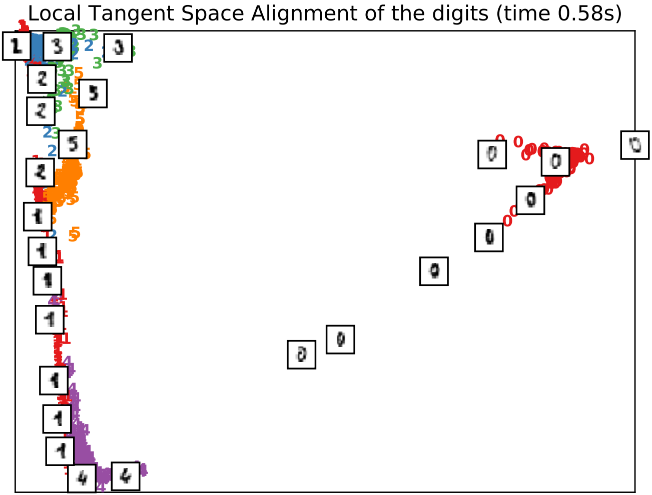
Hardwriting digit visualization



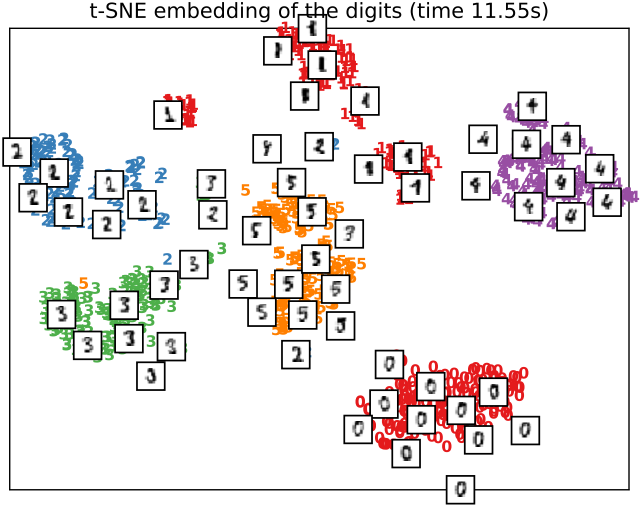
Hardwriting digit visualization



Hardwriting digit visualization



Hardwriting digit visualization



Demo

Demo

Hero images visualization:

`https://onefoldmedia.com/sites/default/super_t-sne`

Resource

Resource

- Stochastic Neighbor Embedding
- isualizing Data using t-SNE
- Local Linear Embedding
- scikit-learn

Thanks!

Thank you !