Visualizing High-Dimensional Data using t-SNE

Authors: **L. van der Maaten** and **G. Hinton**

Presented by: **Oisin Turbitt** and **Anna Ivahnenko**

The University of Edinburgh, Informatics Department

Introduction

- t-Distributed Stochastic Neighbour Embedding (t-SNE) is a visualisation technique useful for exploratory data analysis
- Nonlinear multidimensional scaling (MDS) method for dimensionality reduction
- MDS finds the configuration of points in Euclidean space that represent the dissimilarities of the inputs
- Improvements over original SNE method:

1. Solves crowding problem of points in the middle of embedding by using the heavy-tailed Student-t distribution
2. Minimizes Kullback-Leibler (KL) divergence cost function between high and low dimensional pairwise similarity probability distributions

Theory

- For a high dimensional dataset, $\mathbf{X} = \mathbf{x_1}, \mathbf{x_2} \ldots, \mathbf{x_N}$ compute the joint probability of the pairwise similarities of $\mathbf{x_i}$ and $\mathbf{x_j}$ as defined as:

where: $$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N}$$

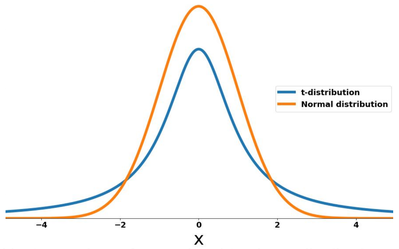
$$p_{j|i} = \frac{exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma_i^2)}{\sum_k \sum_{l \neq k} exp(-\|\mathbf{x}_k - \mathbf{x}_l\|^2/2\sigma_i^2)} \qquad p_{i|i} = 0$$

- Bandwidth of the Gaussian kernel, $\boldsymbol{\sigma_i}$, is set so the perplexity of the $\mathbf{P_i}$ equals a predefined perplexity, $\mathbf{u}$ Ensures $\boldsymbol{\sigma_i}$ adapts to cluster density.

$$u(P_i) = 2^{H(P_i)} \qquad H(P_i) = -\sum_j p_{j|i} log_2 p_{j|i}$$

- Goal is to learn $\mathbf{y_i}$ and $\mathbf{y_j}$ which represents the low dimensional mapping of $\mathbf{x_i}$ and $\mathbf{x_j}$ in the D-dimensional embedding, $\mathbf{Y} = \mathbf{y_1}, \mathbf{y_2}, \ldots, \mathbf{y_N}$. Pairwise similarities of these points, $\mathbf{q_{ij}}$, is given by:

$$q_{ij} = \frac{(1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2)^{-1}}{\sum_k \sum_{l \neq k} (1 + \|\mathbf{y}_k - \mathbf{y}_l\|^2)^{-1}} \qquad q_{ii} = 0$$



(1) Comparison of Normal and Student-t distribution

t-SNE Algorithm

Input : data set $X = \{x_1, x_2, ..., x_N\}$
Output: low-dimensional embedding $Y^T = \{y_1, y_2, ..., y_N\}$
set parameters: perplexity u, number of iterations T, learning rate η, momentum $\alpha(t)$.
begin
  compute pairwise similarities $p_{j|i}$
  set $p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N}$
  sample initial solution $Y^0 = \{y_1, y_2, ..., y_N\}$ from $\mathcal{N}(0, 10^{-4}I)$
  for $t = 1$ **to** T **do**
    compute low-dimensional pairwise similarities q_{ij}
    compute gradient $\frac{\partial C}{\partial Y}$
    set $Y^t = Y^{t-1} + \eta \frac{\partial C}{\partial Y} + \alpha(t)(Y^{t-1} - Y^{t-2})$
  end
end

Student t-Distribution

- Captures global and local structure of similar high dimensional objects as clusters a low dimensional embedding
- Heavy-tailed normalised Student-t distribution kernel allows for dissimilar high-dimensional objects are modelled by distant low-dimensional points
- Acts like a repulsive force between dissimilar points and exaggerates the distances between points in the low dimensional representation
- This promotes movement of clusters in the embedding to more accurately model small pairwise distances, improving the optimisation

Optimization

- To calculate $\mathbf{y_i}$, the t-SNE algorithm will minimize the KL divergence between $\mathbf{p_{ij}}$ and $\mathbf{q_{ij}}$ using:

$$C(\varepsilon) = KL(P\|Q) = \sum_i \sum_{j \neq i} p_{ij} log \frac{p_{ij}}{q_{ij}}$$

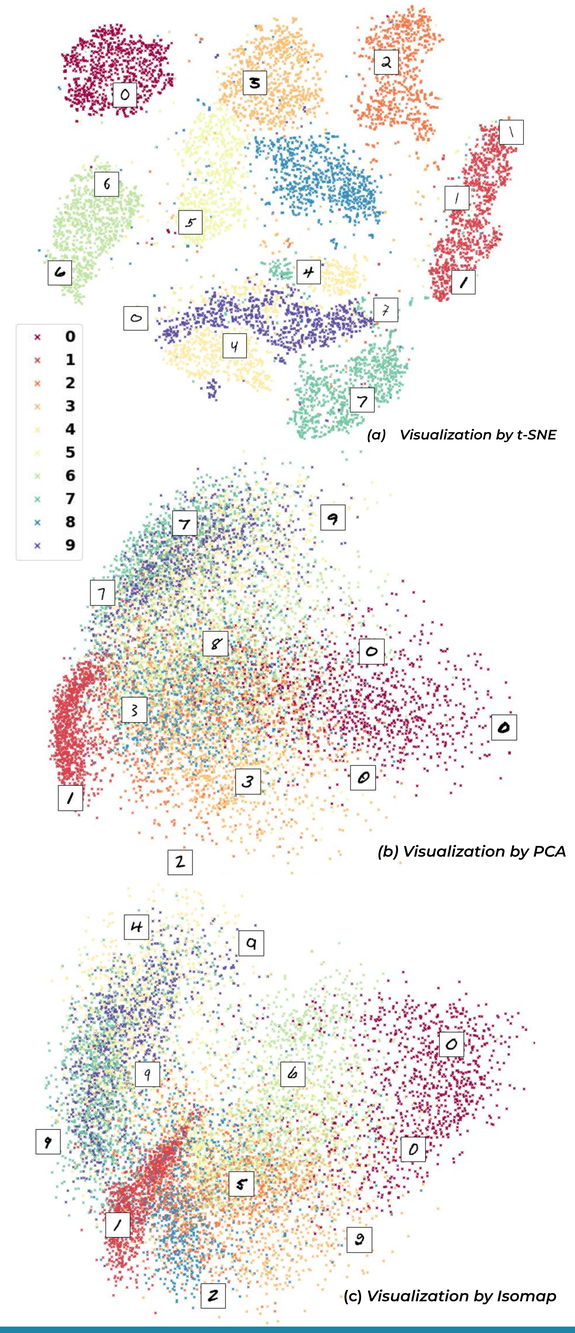
- Models similar inputs with large values of $\mathbf{p_{ij}}$ and nearby points in the embedding space with large values of $\mathbf{q_{ij}}$
- Cost function is minimized by using stochastic gradient descent with momentum and an adaptive learning rate

$$\frac{\partial C}{\partial \mathbf{y}_i} = 4 \sum_j (p_{ij} - q_{ij})(\mathbf{y}_i - \mathbf{y}_j)(1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2)^{-1}$$

- L2 penalty proportional to the sum squared distances of the embedded points from the origin is added to the cost function to force map points to stay close together at the start of the optimization
- "Early exaggeration" initially multiplies all $\mathbf{p_{ij}}$ by 4 to focus modelling large $\mathbf{p_{ij}}$ and $\mathbf{q_{ij}}$, creates small distant clusters in the embedding to result in good global organization

Results

Comparing the results of t-SNE with PCA and Isomap techniques using 10,000 random MNIST images:



(a) Visualization by t-SNE

(b) Visualization by PCA

(c) Visualization by Isomap

Discussion

- Both PCA and Isomap struggle to keep the local and the global data structure, resulting in overlapping clusters.
- t-SNE produces interpretable results on both local and global scales, with well separated clusters which capture subtle differences in inputs
- It is a visualisation tool only. Distances in the embedding do not relate to the location of objects in the high dimensional space, but reflect the joint probabilities under KL divergence
- Non-convex cost function, provides a different solution for separate runs
- Heavily influenced by choice of perplexity. Typically set between 5 - 50
- Heuristic: larger perplexity for larger datasets
- Ineffective on datasets with high intrinsic dimensionality
- For D > 3 , local structure will not be preserved. Higher degrees of freedom t-distributions may be more suitable
- Memory and computational complexities of t-SNE are $O(N^2)$ - limits application to extremely large datasets

Extensions

- Can use different distance measures other than Euclidean
- Autoencoder methods can be used as inputs to t-SNE to produce better results for complex datasets
- Barnes Hut approximation of gradients allows of complexity of $O(NlogN)$. Enables use with datasets of millions of objects
- t-SNE is nonparametric. Parametric version can be obtained by training deep neural network using the t-SNE cost function
- Multiple maps t-SNE for available for non-metric data

References

van der Maaten, L. and Hinton, G. (2008). Visualizing Data using t-SNE. Journal of Machine Learning Research, 9 (2579-2605), pp.2579 - 2605.

Code available at:
Turbitt, O., oisinturbitt/DME. Available at: https://github.com/oisinturbitt/DME.