Visualizing Data using t-SNE (van der Matten and Hinton 2008)

Feburary 2024

Laurens van der Maaten

FOLLOW

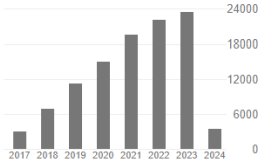
GET MY OWN PROFILE

Distinguished Research Scientist, Meta AI
Verified email at meta.com - Homepage
Machine Learning Computer Vision

TITLE	CITED BY	YEAR
Densely Connected Convolutional Networks G Huang, Z Liu, L van der Maaten, KQ Weinberger IEEE Conference on Computer Vision and Pattern Recognition	42921	2016
Visualizing data using t-SNE L van der Maaten, G Hinton The Journal of Machine Learning Research 9 (2579-2605), 85	41005	2008
Dimensionality reduction: A comparative review LJP Van der Maaten, EO Postma, HJ Van den Herik Technical Report TiCC TR 2009-005	3754 *	2009
Accelerating t-SNE using Tree-Based Algorithms L Van Der Maaten The Journal of Machine Learning Research 15 (1), 3221-3245	2826	2014
Clevr: A diagnostic dataset for compositional language and elementary visual reasoning J Johnson, B Hariharan, L Van Der Maaten, L Fei-Fei, C Lawrence Zitnick, ... Proceedings of the IEEE conference on computer vision and pattern ...	2200	2017
Exploring the limits of weakly supervised pretraining D Mahajan, R Girshick, V Ramanathan, K He, M Paluri, Y Li, A Bharambe, ... Proceedings of the European conference on computer vision (ECCV), 181-196	1466	2018

Cited by VIEW ALL

	All	Since 2019
Citations	111111	95182
h-index	53	43
i10-index	97	78



Public access VIEW ALL

1 article not available
17 articles available

Based on funding mandates

Geoffrey Hinton

FOLLOW

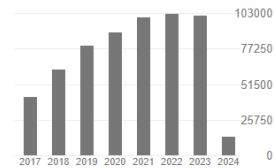
GET MY OWN PROFILE

Emeritus Prof. Computer Science, University of Toronto
Verified email at cs.toronto.edu - Homepage
machine learning psychology artificial intelligence cognitive science computer science

TITLE	CITED BY	YEAR
Imagenet classification with deep convolutional neural networks A Krizhevsky, I Sutskever, GE Hinton Advances in neural information processing systems 25	149946 *	2012
Deep learning Y LeCun, Y Bengio, G Hinton Nature 521 (7553), 436-44	75703	2015
Dropout: a simple way to prevent neural networks from overfitting N Srivastava, G Hinton, A Krizhevsky, I Sutskever, R Salakhutdinov The journal of machine learning research 15 (1), 1929-1958	48860	2014
Visualizing data using t-SNE L van der Maaten, G Hinton Journal of Machine Learning Research 9 (Nov), 2579-2605	41485	2008
Learning representations by back-propagating errors DE Rumelhart, GE Hinton, RJ Williams Nature 323 (6088), 533-536	36977	1986
Learning internal representations by error-propagation DE Rumelhart, GE Hinton, RJ Williams Parallel Distributed Processing: Explorations in the Microstructure of ...	32784	1986
Learning multiple layers of features from tiny images A Krizhevsky, G Hinton	27773	2009

Cited by VIEW ALL

	All	Since 2019
Citations	756277	486885
h-index	182	129
i10-index	461	346



Public access VIEW ALL

1 article not available
8 articles available

Based on funding mandates

Outline

Introduction

Goal:

- visualize high-dimensional (HD) data by giving each data point a location in a two or three dimensional map.
- preserve significant structure of the high-dimensional data $\mathcal{X} = \{x_1, x_2, ..., x_n\}$ in the low-dimensional (LD) $\mathcal{Y} = \{y_1, y_2, ..., y_n\}$ map.

Popular Linear Techniques

- Traditional linear techniques: PCA and classical multidimensional scaling (MDS; Torgerson 1952)
- Linear techniques focus on keeping the low-dimensional representations of dissimilar data points far apart
- It is usually more important to keep the low-dimensional representations of very similar data points close together. This is hard with a linear mapping

Popular Nonlinear Techniques

- Aim: to preserve the local structure
- Sammon mapping (Sammon 1969), curvilinear components analysis (CCA; Demartines and Herault 1997), Stocastic Neighbor Embedding (SNE; Hinton and Roweis 2002), Isomap (Tenenbaum et al 2000), Maximum Variance Unfolding (MVU; Weiberger et al 2004), Locally Linear Embedding (LLE; Roweis and Saul 2000), and Laplacian Eigenmaps (Belkin and Niyogi 2002)
- They are often not very successful at visualizing real HD data
- Most of the techniques are not capable of retaining both the local and the global structure of the data in a single map

In this paper

- The authors describe a way of converting a HD data set into a matrix of pairwise similarities
- What are the problems of the existing method, Stochastic Neighbor Embedding (SNE)? How did the authors extend SNE to t-SNE?
- t-SNE can visualize the resulting similarity data and capture both local and global structure

Stochastic Neighbor Embedding

Convert the HD Euclidean distances between data points into conditional probabilities that represent similarities. The similarity of data point x_j to data point x_i is the conditional probability $p_{j|i}$

$$p_{j|i} = \frac{\exp(-\|x_i - x_j\|^2/2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|x_i - x_k\|^2/2\sigma_i^2)}$$

, where σ_i^2 is the variance of the Gaussian that is centered on the data point x_i

For the low-dimensional (LD) counterparts y_i and y_j of the HD data points x_i and x_j, compute a similar conditional probability $q_{j|i}$

$$q_{j|i} = \frac{\exp(-\|y_i - y_j\|^2)}{\sum_{k \neq i} \exp(\|y_i - y_k\|^2)}$$

Stochastic Neighbor Embedding

SNE aims to find a LD data representation that minimizes the mismatch between $p_{j|i}$ and $q_{j|i}$. SNE minimized the sum of Kullback-Leibler divergences over all data points using a gradient descent method. The cost function C is given by

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

, where P_i represents the conditional probability distribution over all other data points given data point x_i and Q_i represents the conditional probability distribution over all other data points given data point y_i

Stochastic Neighbor Embedding

SNE performs a binary search for the value of σ_i that produces a P_i with a fixed perplexity that is specified by the user. The perplexity is defined as

$$Perp(P_i) = s^{H(P_i)}$$

, where $H(P_i)$ is the Shannon entropy of P_i measured in bits

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

The perplexity can be interpreted as a smooth measure of the effective number of neighbors.

Stochastic Neighbor Embedding

The minimization of the cost function is performed using a gradient descent method.

$$\frac{\partial C}{\partial y_i} = 2\sum_j (p_{j|i} - q_{j|i} + p_{i|j} - q_{i|j})(y_i - y_j)$$

the gradient update with a momentum term is given by

$$\mathcal{Y}^{(t)} = \mathcal{Y}^{(t-1)} + \eta\frac{\partial C}{\partial \mathcal{Y}} + \alpha(t)(\mathcal{Y}^{(t-1)} - \mathcal{Y}^{(t-2)})$$

,where $\mathcal{Y}^{(t)}$ indicates the solution at iteration t, η indicates the learning rate, and $\alpha(t)$ represents the momentum at iteration t

Stochastic Neighbor Embedding

Problems of SNE

- the cost function is difficult to optimize and
- "crowding problem"

t-SNE alleviates these problems by

1. it uses a symmetrized version of the SNE cost function with simpler gradients
2. it uses a Student-t distribution rather than a Gaussian to compute the similarity between two points in the LD space
3. t-SNE employs a heavy-tailed distribution in the LD space to alleviate both crowding problem and the optimization problems of SNE

Symmetric Stochastic Neighbor Embedding

- Previously, minimize the sum of the KL divergence between the conditional probability $p_{j|i}$ and $q_{j|i}$
- Now, consider minimizing a single KL divergence between joint probability distributions, P, in HD and Q, in LD

$$C = KL(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

- symmetric property: $p_{ij} = p_{ji}$ and $q_{ij} = q_{ji}$ for $\forall i, j$
- Pairwise similarities in LD map q_{ij} are given by

$$q_{ij} = \frac{\exp(-\|y_i - y_j\|^2)}{\sum_{k \neq l} \exp(-\|y_j - y_l\|^2)}$$

in HD space p_{ij} is

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

Symmetric Stochastic Neighbor Embedding

The main advantage of the symmetric version of SNE is the simpler form of its gradient

$$\frac{\partial C}{\partial y_i} = 4 \sum_j (p_{ij} - q_{ij})(y_i - y_j)$$

The Crowding Problem

- Suppose that the manifold has 10 intrinsic dimensions and is embedded within a space of much higher dimensionality
- In ten dimensions, it is possible to have 11 datapoints that are mutually equal distant and there is no way to model this faithfully in 2-d map
- "crowding problem": the area of the two-dimensional map that is available to accommodate moderately distant data points will not be large enough. Hence, if we want to model the small distances accurately in the map, most of the points that are at a moderate distance from data point i will have to be placed much too far away

The Crowding Problem

- The crowding problem prevents gaps from forming between the natural clusters
- The crowding problem is not specific to SNE, but that it also occurs in other local techniques for multidimensional scaling such as Sammon mapping
- An attempt to address crowding problem is UNI-SNE (Cook et al, 2007), by introducing a small mixing proportion, ρ, so however far apart two map points are, q_{ij} can never fall below $\frac{2\rho}{n(n-1)}$. This means for points that are far apart in HD, q_{ij} will always larger than p_{ij}
- UNI-SNE usually outperforms standard SNE, but the optimization of the UNI-SNE cost function is tedious...

Mismatch Tails can Compensate for Mismatched Dimensionalities

- We still aim to solve the crowding problem
- We are hoping that data points, that are far apart in HD, have q_{ij} larger than p_{ij}
- In LD space, we can use a probability distribution that has much heavier tails than a Gaussian

Mismatch Tails can Compensate for Mismatched Dimensionalities

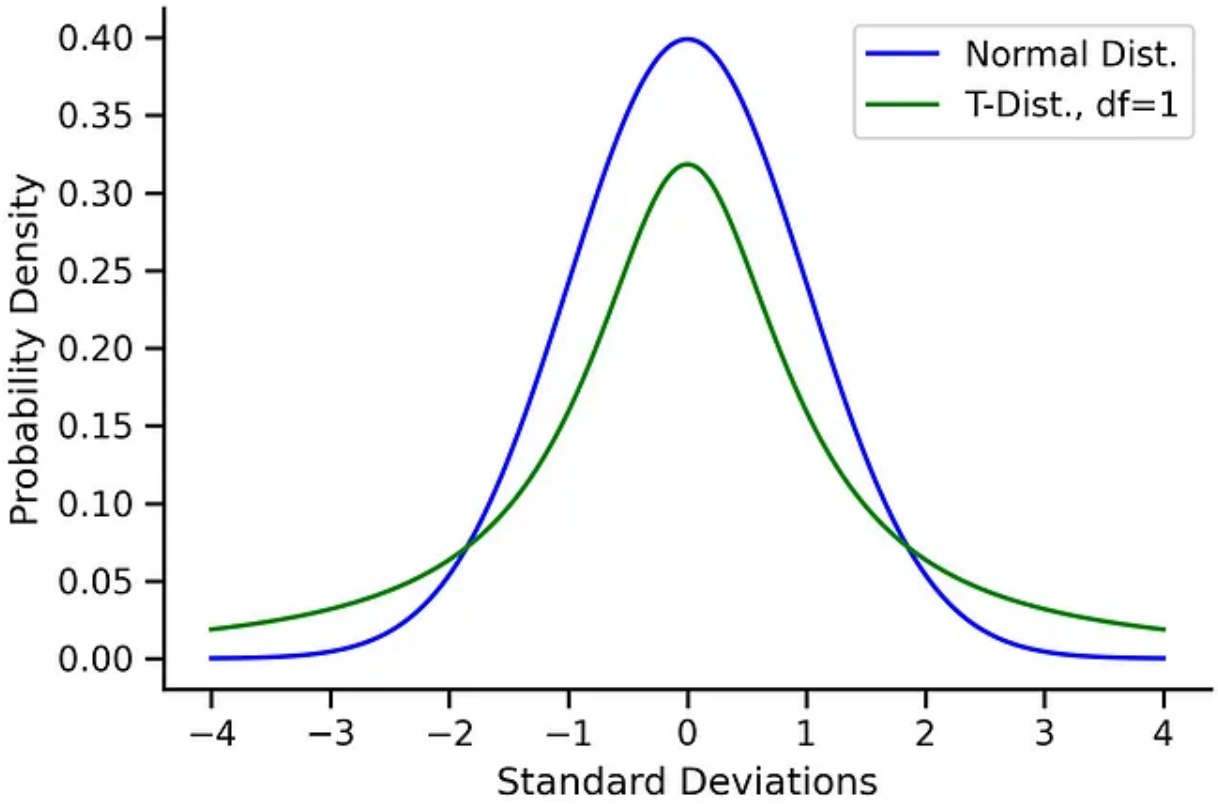
In t-SNE, the authors employ a Student t-distribution with one degree of freedom as the heavy-tailed distribution in the LD map.

$$q_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k \neq l}(1 + \|y_k - y_l\|^2)^{-1}}$$

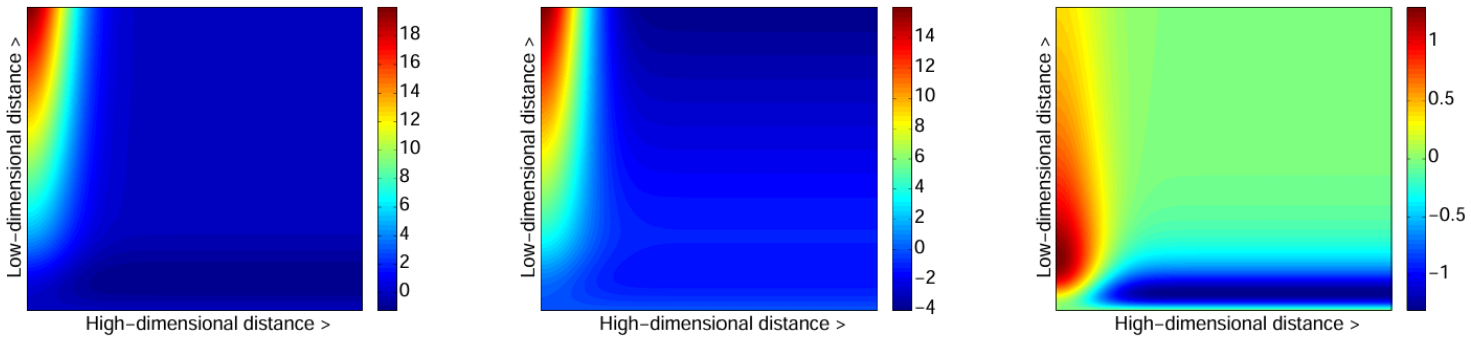
The gradient of the KL divergence between P and the Student t-based joint probability distribution Q is

$$\frac{\partial C}{\partial y_i} = 4 \sum_j (p_{ij} - q_{ij})(y_i - y_j)(1 + \|y_i - y_j\|^2)^{-1}$$

Mismatch Tails can Compensate for Mismatched Dimensionalities



Mismatch Tails can Compensate for Mismatched Dimensionalities



(a) Gradient of SNE. (b) Gradient of UNI-SNE. (c) Gradient of t-SNE.

Figure 1: Gradients of three types of SNE as a function of the pairwise Euclidean distance between two points in the high-dimensional and the pairwise distance between the points in the low-dimensional data representation.

Algorithm

Algorithm 1: Simple version of t-Distributed Stochastic Neighbor Embedding.

Data: dataset $\mathcal{X} = \{x_1, x_2, ..., x_n\}$,
cost function parameters: perplexity $Perp$,
optimization parameters: number of iterations T, learning rate η, momentum $\alpha(t)$.
Result: low-dimensional data representation $\mathcal{Y}^{(T)} = \{y_1, y_2, ..., y_n\}$.
begin
  compute pairwise affinities $p_{j|i}$ with perplexity $Perp$ (using Equation 1)
  set $p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}$
  sample initial solution $\mathcal{Y}^{(0)} = \{y_1, y_2, ..., y_n\}$ from $\mathcal{N}(0, 10^{-4}I)$
  for *t=1* **to** *T* **do**
    compute low-dimensional affinities q_{ij} (using Equation 6)
    compute gradient $\frac{\delta C}{\delta \mathcal{Y}}$ (using Equation 7)
    set $\mathcal{Y}^{(t)} = \mathcal{Y}^{(t-1)} + \eta \frac{\delta C}{\delta \mathcal{Y}} + \alpha(t) \left(\mathcal{Y}^{(t-1)} - \mathcal{Y}^{(t-2)} \right)$
  end
end

Optimization Methods for t-SNE

- The simple procedure uses a momentum term to reduce the number of iterations required and it works best if the momentum term is small until the map points have become moderately well organized
- **early compression**: to force the map points to stay close together at the start of the optimization. This is implemented by adding an additional L2-penalty to cost function that is proportional to the sum of squared distances of the map points from the origin
- **early exaggeration**: to multiply all of the p_{ij}'s by, for example 4, in the initial stages of the optimization

Experiments

- To evaluate t-SNE, the authors present experiments in which t-SNE is compared to seven other non-parametric techniques for dimensionality reduction
- Methods: (1) Sammon mapping; (2) Isomap; (3) LLE; (4) CCA; (5) SNE; (6) MVU; (7) Laplacian Eigenmaps

Experimental Data Sets

Datasets	Number of samples	Number of features	$n \times p$
MNIST	6,000 images	$28 \times 28 = 784$ pixels	$6{,}000 \times 784$
Olivetti faces	400 images 40 ind $\times$ 10 images/ind	$92 \times 112 = 10{,}304$ pixels	$400 \times 10{,}304$
COIL-20	1,440 images 20 obj $\times$ 72 orientations	$32 \times 32 = 1{,}024$ pixels	$1{,}440 \times 1{,}024$
Word features	1,000 most common words	100	$1{,}000 \times 100$
Netflix	Ratings for 500 most popular movies	30 hidden units trained by a Restrictive Boltzmann Machine	$17{,}770 \times 30$

Experimental Setup

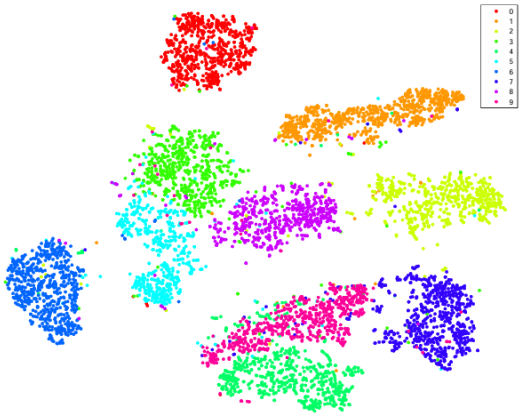
1. Apply PCA to reduce the dimensionality of the data to 30
2. Use different dimension reduction methods to convert the 30-dimensional representation to a two-dimensional map
3. show the resulting map as a scatterplot, labelled by their class information

Experimental Setup

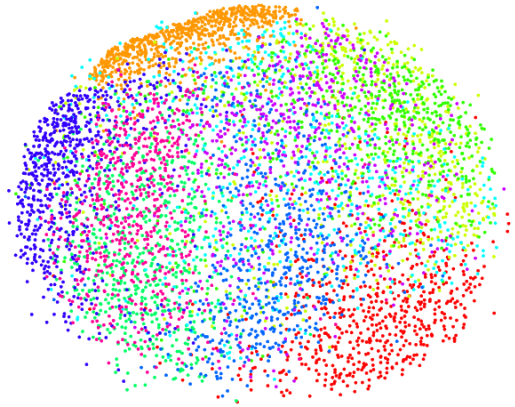
Technique	*Cost function parameters*
t-SNE	$Perp = 40$
Sammon mapping	none
CCA	$\lambda = 3s^2$
SNE	$Perp = 40$
Isomap	$k = 12$
MVU	$k = 12$
LLE	$k = 12$
Laplacian Eigenmaps	$k = 12 \quad \sigma = 1$

Table 1: Parameter settings for the experiments.

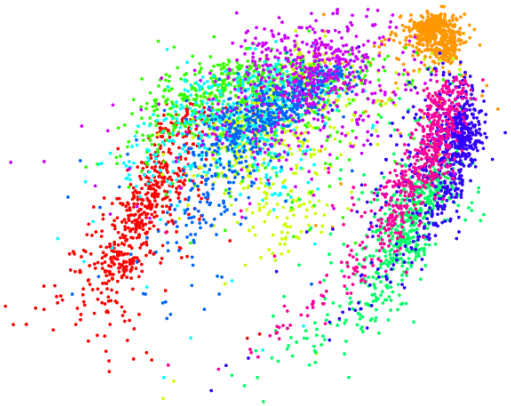
Results



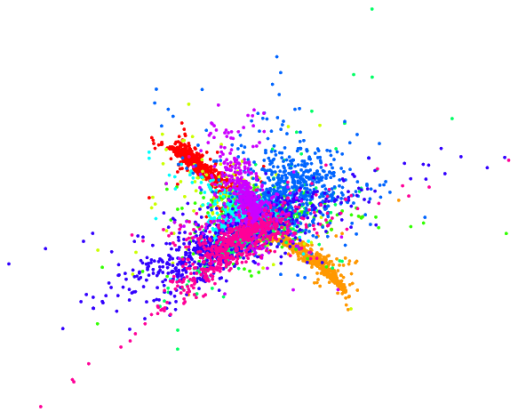
(a) Visualization by t-SNE.



(b) Visualization by Sammon mapping.



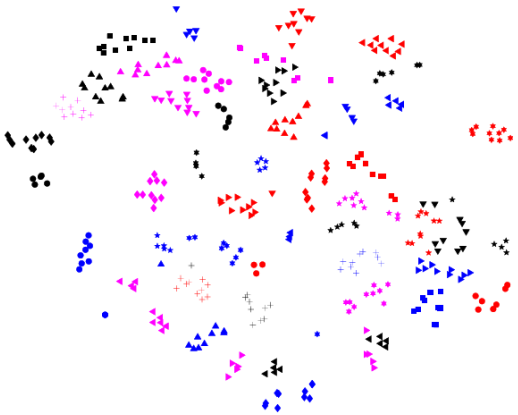
(c) Visualization by Isomap.



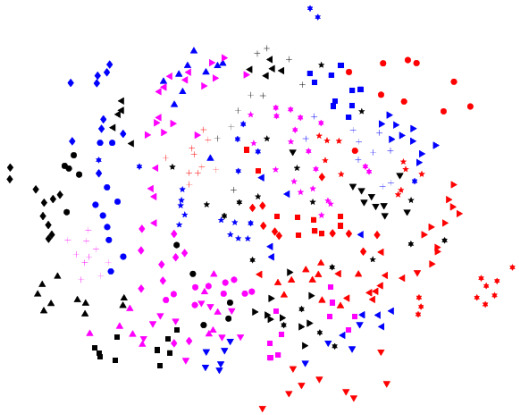
(d) Visualization by LLE.

Figure 2: Visualizations of 6,000 handwritten digits from the MNIST dataset.

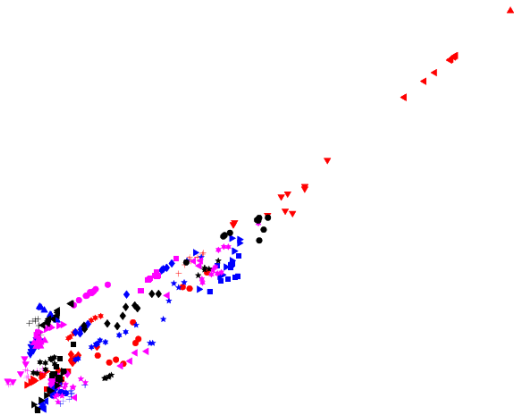
Results



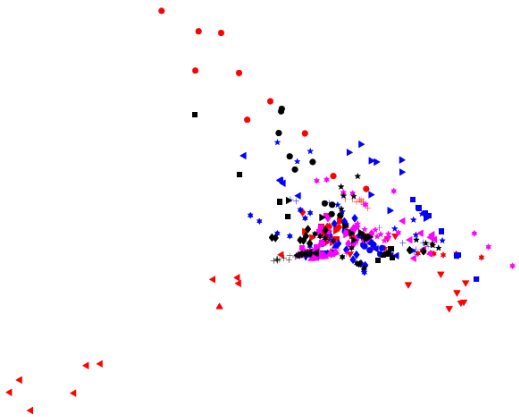
(a) Visualization by t-SNE.



(b) Visualization by Sammon mapping.



(c) Visualization by Isomap.

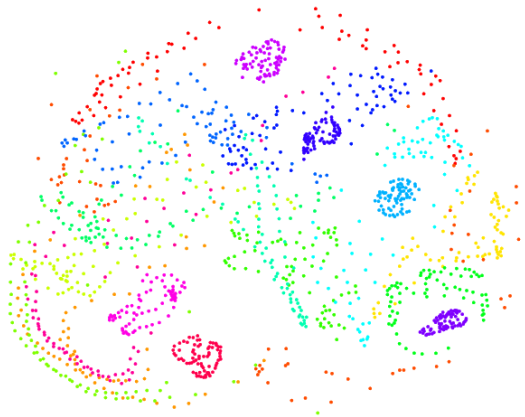


(d) Visualization by LLE.

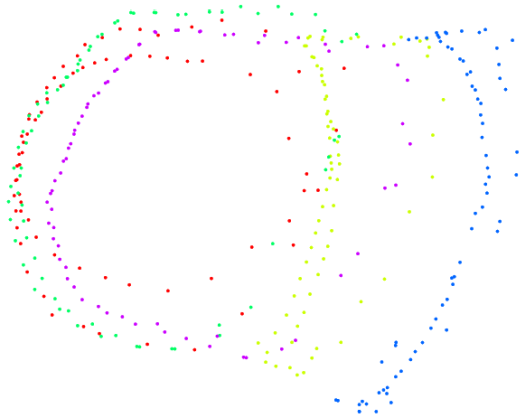
Figure 3: Visualizations of the Olivetti faces dataset.

Results

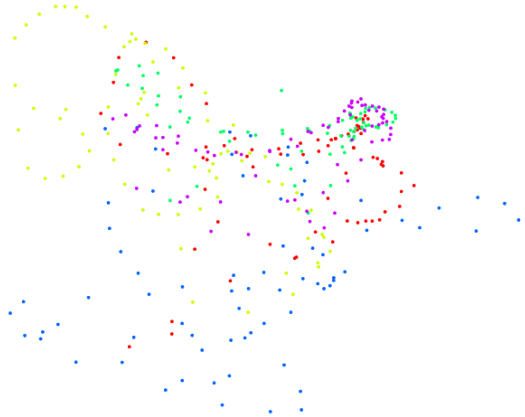
(a) Visualization by t-SNE.



(b) Visualization by Sammon mapping.



(c) Visualization by Isomap.



(d) Visualization by LLE.

Figure 4: Visualizations of the COIL-20 dataset.

Applying t-SNE to Large Data Sets

Like many other visualization techniques, t-SNE has a computational and memory complexity that is quadratic in the number of data points. This makes it infeasible to apply the standard version to large datasets, so the authors introduced the random walk version of t-SNE:

1. choose a desired number of neighbors and create a neighbor graph for all the data points
2. For each of the landmark points, define a random walk starting at that landmark point and terminate as soon as it lands on another landmark point

Applying t-SNE to Large Data Sets

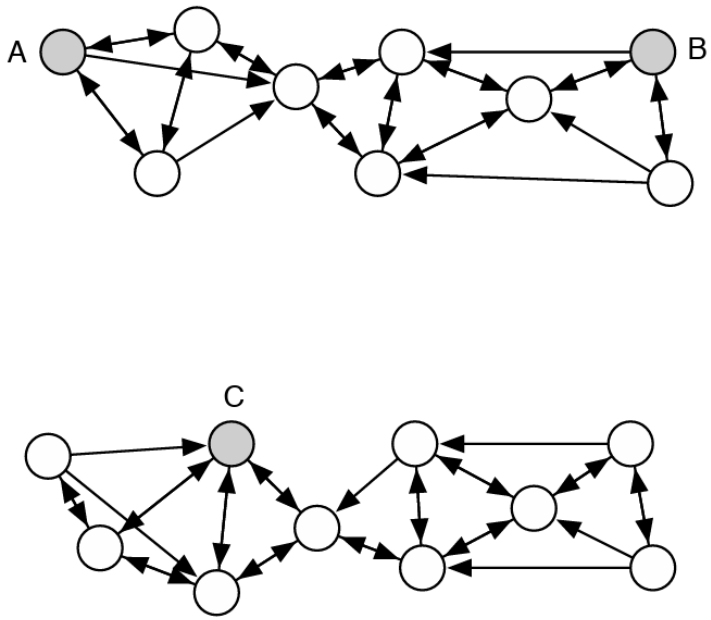


Figure 5: An illustration of the advantage of the random walk version of t-SNE over a standard landmark approach. The shaded points A, B, and C are three (almost) equidistant landmark points, whereas the non-shaded datapoints are non-landmark points. The arrows represent a directed neighborhood graph where $k = 3$. In a standard landmark approach, the pairwise affinity between A and B is approximately equal to the pairwise affinity between A and C. In the random walk version of t-SNE, the pairwise affinity between A and B is much larger than the pairwise affinity between A and C, and therefore, it reflects the structure of the data much better.

Applying t-SNE to Large Data Sets

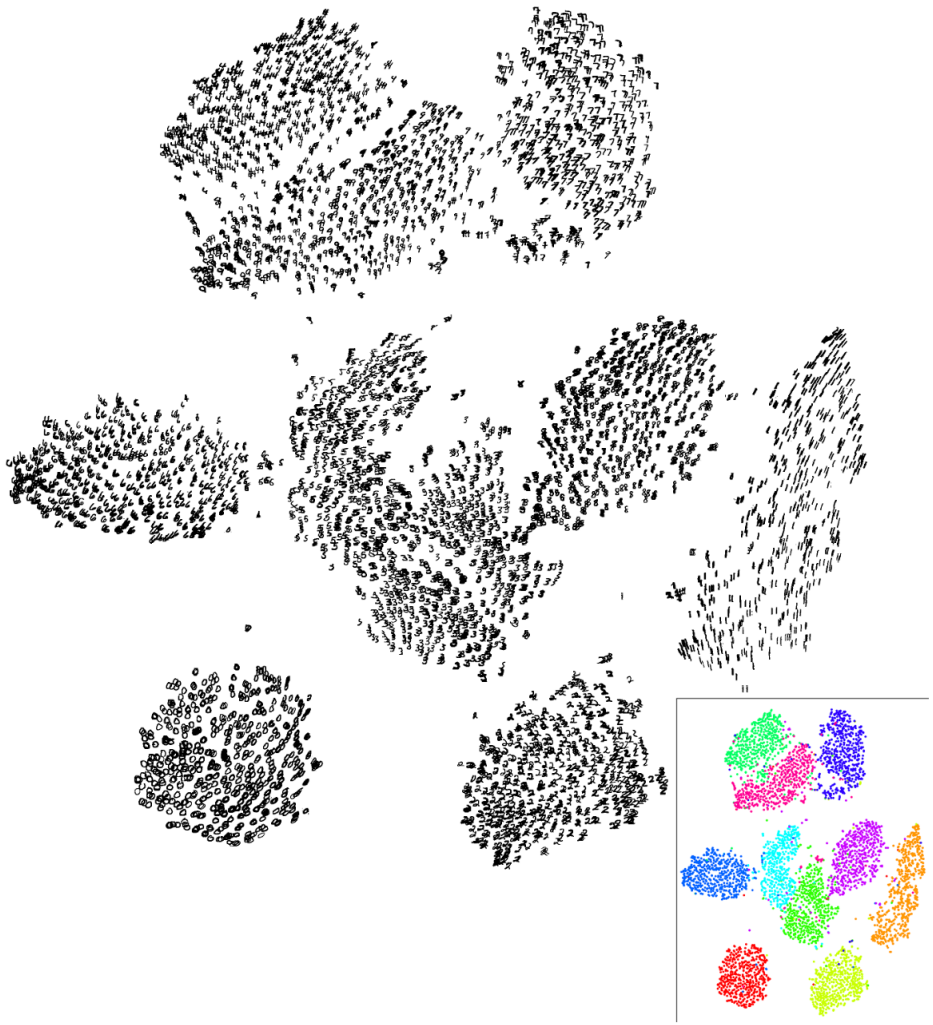


Figure 6: Visualization of 6,000 digits from the MNIST dataset produced by the random walk version of t-SNE (employing all 60,000 digit images).

Weakness Discussed in the Paper

1. it is unclear to show how t-SNE performs on general dimensionality reduction tasks
2. the relatively local nature of t-SNE makes it sensitive to the curse of the intrinsic dimensionality of the data
3. t-SNE is not guaranteed to converge to a global optimum of its cost function

Other Weakness

1. Relatively slow
2. Not great at pre-processing features for prediction
3. Has hyperparameters
4. Sensitive to initialization conditions
5. Need to be careful when incorporating categorical variables
6. Sensitive to scale
7. Cannot handle missing data

Conclusion

- t-SNE is a new technique for visualization of similarity data that is capable of retaining the local structure and global structure
- Both the computational and the memory complexity of t-SNE are $\mathcal{O}(n^2)$
- random-walk version of t-SNE for visualizing large data set