T-distributed stochastic neighbor embedding (t-SNE)

Nguyen The Phong Le Thien Toan Do Thi Huong Trang
Nguyen Minh Thu

May 17th, 2020

Contents

Chapter 1

Introduction

Visualization of high-dimensional data is an important problem in many different domains and deals with data of widely varying dimensionality. There are a lot of methods to visualize high-dimensional data have been proposal. However, following development of the world, data sets also bigger than before and contain thousands of high-dimensional datapoints. So, we need methods that can decrease dimensional without losing too much information on the original data set.

Dimensionality reduction methods convert the high-dimensional dataset $X = \{x_1, x_2, ...x_n\}$ into two or three-dimensional data $Y = \{y_1, y_2, ..., y_n\}$ that can be displayed as 2-D or 3-D scatter plot. Various techniques for this problem have been proposed that differ in the type of structure they preserve.

Traditional dimensionality reduction techniques suchs as Principal Component Analysis (PCA [3]) and classical multidimensional scaling[5] are linear techniques that focus on separating dissimilar datapoints. On the other hand, various nonlinear reduction techniques as Sammon mapping[4], curvilinear components analysis[1] aim to keep the low-dimensional representations of very similar datapoints close together .

In this report, we will introduce another method to convert a high-dimensional data set into a matrix of pairwise similarities is t-Distributed Stochastic Neighbor Embedding(t-SNE),a non-linear technique for dimensionality reduction for visualizing the resulting similarity data. t-SNE is capable of capturing much of the local structure of the high-dimensional data while also revealing global structure such as the presence of clusters at several scales. It is extensively applied in image processing, NLP, genomic data and speech processing. .

Chapter 2

Theoretical Background

2.1 Stochastic Neighbor Embedding (SNE)

Before we approach t-SNE, we consider Stochastic Neighbor Embedding (SNE) is presented by Hinton and Roweis [2]. This method starts by converting the high-dimensional Euclidean distances between datapoints into conditional probability that represent similarities.
For the high-dimensional datapoints x_i and x_j, let $p_{j|i}$ which the conditional probability is the similarity of datapoint x_j to datapoint x_i. It means that t x_i would pick x_j as its neighbor if neighbors were picked in proportion to their probability density under a Gaussian centered at x_i. Mathematically, the conditional probability $p_{j|i}$ is given by:

$$p_{j|i} = \frac{exp(-\|x_i - x_j\|^2/2\sigma_i^2)}{\sum_{k \neq i} exp(-\|x_i - x_k\|^2/2\sigma_i^2)} \tag{2.1}$$

Where σ_i is the variance of the Gaussian that is centered on datapoints. Since we are only interested in modeling pairwise similarities, we set the value of $p_{i|j}$ to zero.
For the low-dimensional counterparts y_i and y_j of the high-dimensional datapoints x_i and x_j, it is possible to compute a similar conditional probability, which we denote by $q_{j|i}$. Variance of the Gaussian is employed in the computation of the conditional probabilities $q_{j|i}$ is set to $\frac{1}{\sqrt{2}}$. Hence, modeling the similarity of map point y_j to map point y_i by:

$$q_{j|i} = \frac{exp(-\|y_i - y_j\|^2)}{\sum_{k \neq i} exp(-\|y_i - y_k\|^2)} \tag{2.2}$$

Where $q_{i|i} = 0$. If map points y_i and y_j correctly model the similarity between the high-dimensional datapoints x_i and x_j, the conditional probabilities $q_{j|i}$ and $p_{j|i}$ will be equal. SNE aims to find a low-dimensional data representation that minimizes the mismatch between $q_{j|i}$

and $p_{j|i}$. The minimizes the sum of Kullback-Leibler divergences over all datapoints using a gradient descent method. The cost function C is given by

$$C = \sum_i KL(P_i||Q_i) = \sum_i \sum_j p_{j|i} log \frac{p_{j|i}}{qj|i} \tag{2.3}$$

in which P_i represents the conditional probability distribution over all other datapoints given datapoints x_i, and Q_i represents the conditional distribution over all other map points given map points y_i.

The remaining parameter to be selected is the variance i of the Gaussian that is centered over each high-dimensional datapoint, x_i. 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 can be interpreted as a smooth measure of the effective number of neighbors.

The perplexity is defined as:

$$Perp(P_i) = 2^{H(P_i)} \tag{2.4}$$

Where $H(P_i)$ is the Shannon entropy of P_i measured in bits:

$$H(P_i) = -\sum_j p_{j|i} log_2 p_{j|i} \tag{2.5}$$

The minimization of the cost function uses a gradient descent method:

$$\frac{\delta C}{\delta y_i} = 2 \sum_j (p_{j|i} - q_{j|i} + p_{i|j} - q_{i|j})(y_i - y_j) \tag{2.6}$$

In order to speed up the optimization and to avoid poor local minima, a relatively large momentum term is added to the gradient.The gradient update with a momentum term is given by:

$$Y^(t) = Y^{(t-1)} + \eta \frac{\delta C}{\delta Y} + \alpha(t)(Y^{(t-1)} - Y^{(t-2)} \tag{2.7}$$

Where:

- $Y^{(t)}$ indicates the solution at iteration t
- η indicates the learning rate
- $\alpha(t)$ represents the momentum at iteration t.

2.2 t-Distributed Stochastic Neighbor Embedding (t-SNE)

2.2.1 Symmetric SNE

We can also possible to minimize the sum of the Kullback-Leibler divergences between a joint probability distribution, P, in the high-dimensional space and a joint probability distribution, Q, in the low-dimensional space: :

$$C = KL(P\|Q) = \sum_i \sum_j p_{ij} log \frac{p_{ij}}{q_{ij}} \tag{2.8}$$

We set p_{ii} and q_{ii} to zero, again.
This type is call symmetric SNE, since it has the property that $p_{ij} = p_{ji}$ and $q_{ij} = q_{ji}$ for $\forall i, j$.
The pairwise similarities in the low-dimensional map q_{ij} are given by:

$$q_{ij} = \frac{exp(-\|y_i - y_j\|^2)}{\sum_{k \neq l} exp(-\|y_k - y_l\|^2)} \tag{2.9}$$

The pairwise similarities in the high-dimensional space p_{ij} are given by:

$$p_{ij} = \frac{exp(-\|x_i - x_j\|^2/2\sigma_i^2)}{\sum_{k \neq l} exp(-\|x_k - x_l\|^2/2\sigma_i^2)} \tag{2.10}$$

This causes problems when a high-dimensional datapoint x_i is an outlier (all pairwise distances $\|x_i - x_j\|^2$ are large for x_i), the value of p_{ij} are extremely small for all j, so the location of its low-dimensional map point y_i has very little effect on the cost function. In order to circumvent this problem, we define the joint probabilities p_{ij} in the high-dimensional space to be the symmetrized conditional probabilities :

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

This ensures that $\sum_j p_{ij} > \frac{1}{2n}$ for all datapoints x_i, as a result of which each datapoint x_i makes a significant contribution to the cost function.
The gradient of symmetric SNE is given by:

$$\frac{\delta C}{\delta y_i} = 4 \sum_j (p_{ij} - q_{ij})(y_i - y_j) \tag{2.11}$$

2.2.2 The Crowding Problem

There are many reasons why the pairwise distances in a two-dimensional map cannot faithfully model distances between points on the higher dimensional manifold. For example, consider a data set with 10 dimensions, it is possible to have 11 datapoints that are mutually equidistant and there is no way to model this faithfully in a two-dimensional map.
A related problem is the different distribution of pairwise distances in the two spaces. Imagine a set of datapoints that a uniformly distributed around the center i of a sphere. When we project the points onto a 2-D map, the area of the 2-D map is not enough to accommodate all the points so most of the points that are at a moderate distance from datapoint i will have to be placed much too far away in the 2-D map. In SNE, the spring that connected i to these points will then exert a very small attractive force. Although these forces a small but a large number of them will crushes the points together toward the center of the map. This crowding problem is not unique to SNE but also occurs in other local techniques for multidimensional scaling such as Sammon mapping.
There was an attempt to address the crowding problem presented by Cook et al. (2007) by adding a slight repulsion to all springs. This repulsion is created by introducing a uniform background model with a small mixing proportion ρ. This technique is called UNI-SNE and although it usually outperform standard SNE, the optimization of the UNI-SNE cost function is tedious. The best optimization method is to start by setting the background mixing proportion to zero (i.e. performing standard SNE) and only after SNE cost function has been optimized using simulated annealing, the background mixing proportion can be increased to allow some gaps to form between natural clusters. Optimizing the UNI-SNE cost function directly does not work because if two parts of a cluster get separated early on in the optimization, there is no force to pull them back together.

2.2.3 t-SNE algorithm

In order to alleviate the crowding problem and simplify the cost function of SNE, we use t-Distributed Stochastic Neighbor Embedding (t-SNE) technique.
The cost function used by t-SNE differs from the one used by SNE in two ways:

- Uses a symmetrized version of the SNE cost function.
- Uses a Student-t distribution rather than a Gaussian to compute similarity between two points in the low-dimensional space

We have a natural way of alleviating the crowding problem that works as follows. In the high-dimensional space, we convert distances into probabilities using a Gaussian distribution. In the low-dimensional map, we can use a probability distribution that has much heavier tails

than a Gaussian to convert distances into probabilities. This allows a moderate distance in the high-dimensional space to be faithfully modeled by a much larger distance in the map. More specifically,t-SNE uses a Student t-distribution with one degree of freedom in the low-dimensional map. The joint probabilities q_{ij} are defined as:

$$q_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k \neq l}(1 + \|y_k - y_l\|^2)^{-1}} \tag{2.12}$$

The gradient of the Kullback-Leibler divergence between P and the Student-t based joint probability distribution Q is given by:

$$\frac{\delta C}{\delta y_i} = 4 \sum_j (p_{ij} - q_{ij})(y_i - y_j)(1 + \|y_i - y_j\|^2)^{-1} \tag{2.13}$$

Simple version of t-Distributed Stochastic Neighbor Embedding:

Data: data set $X = x_1, x_2, ..., x_n$,
cost function parameters: perplexity Perp,
optimization parameters: number of iterations T, learning rate η, momentum $\alpha(t)$.
Result: low-dimensional data representation $Y^{(T)} = y_1, y_2, ..., y_n$
begin
  compute pairwise affinities $p_{j|i}$ with perplexity Perp (using Equation (2.1))
  set $p_{ij} = \frac{p_{i|j} + p_{j|i}}{2n}$
  sample initial solution $Y^{(0)} = y_1, y_2, ..., y_n$ from $N(0, 10^{-4}I)$
  for $t = 1$ **to** T **do**
    compute low-dimensional affinities q_{ij} (using Equation (2.12))
    compute gradient $\frac{\delta C}{\delta Y}$ (using Equation (2.13))
    set $Y^{(t)} = Y^{(t-1)} + \eta \frac{\delta C}{\delta Y} + \alpha(t)(Y^{(t-1)} - Y^{(t-2)})$
  end
end

Chapter 3

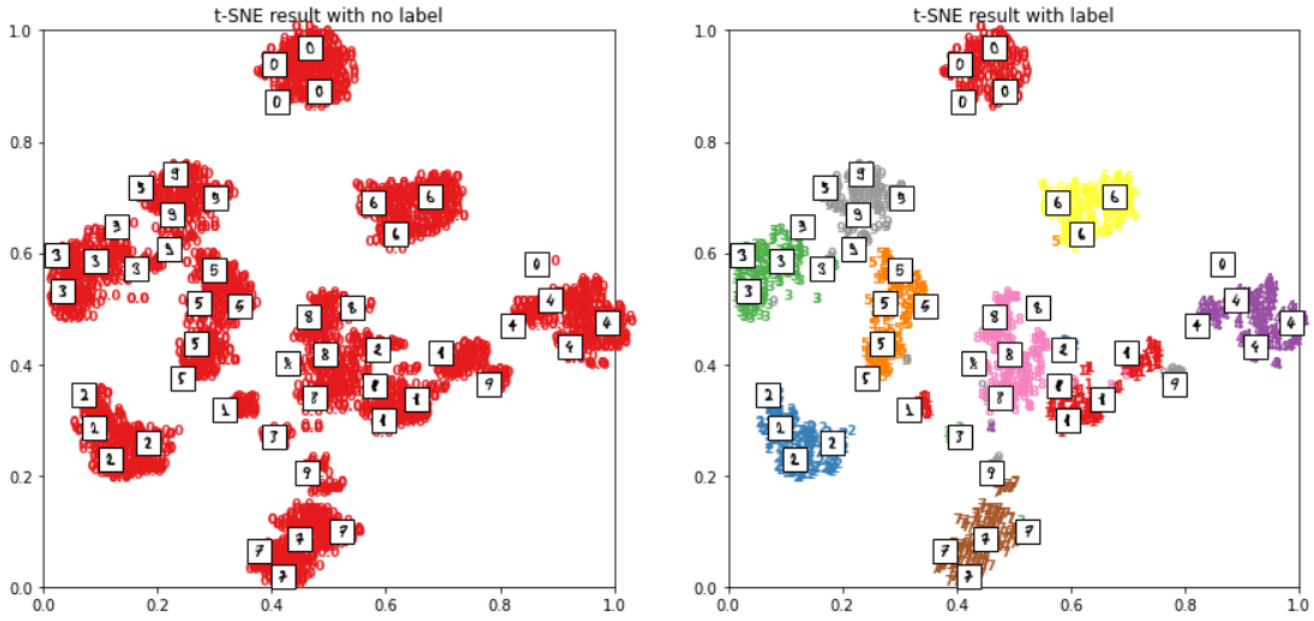
Applications

t-SNE visualization can help gain insight on the structure of data. Thus, it can be useful in many fields that study data: machine learning, health care, genetics, language processing, etc.
On general, t-SNE applications can be:

- Exploring data for classification tasks
- Evaluate the result of embedding techniques

3.1 Exploring data for classification tasks

Using the MNIST dataset, t-SNE can visualize the data with somewhat clear clusters.



It indicates that the classes are seperatable. As the result, we can obtain high performance on classification task even with simple models, without any data preprocessing.

Method	Accuracy	Precision	Recall	F1
Decision Tree	0.840741	0.840741	0.840741	0.840741
KNNs	0.981481	0.981481	0.981481	0.981481
Neural Network	0.974074	0.974074	0.974074	0.974074

But what about more complex dataset with no clear seperation between classes. To illustrate this point, Urban Land Cover Data Set will be used. t-SNE couldn't clearly seperate the classes.

And if we pass this data into classification models, the result will be poor

Method	Accuracy	Precision	Recall	F1
Decision Tree	0.727811	0.727811	0.727811	0.727811
KNNs	0.382643	0.382643	0.382643	0.382643
Neural Network	0.420118	0.420118	0.420118	0.420118

KNNs and Neural Network perform worse than random guess (less than 0.5 accuracy). There are three possible reasons:

- The classes are not seperable
- The number of features is nearly as large as number of samples (KNNs suffers from curse of dimension)
- The number of samples is relatively low with that number of features (not enough data for neural network to train)

But what if we preprocess the data first. For illustration purpose, the data will go through Standard Scaling and PCA. The result is better in both t-SNE and Classification accuracy.

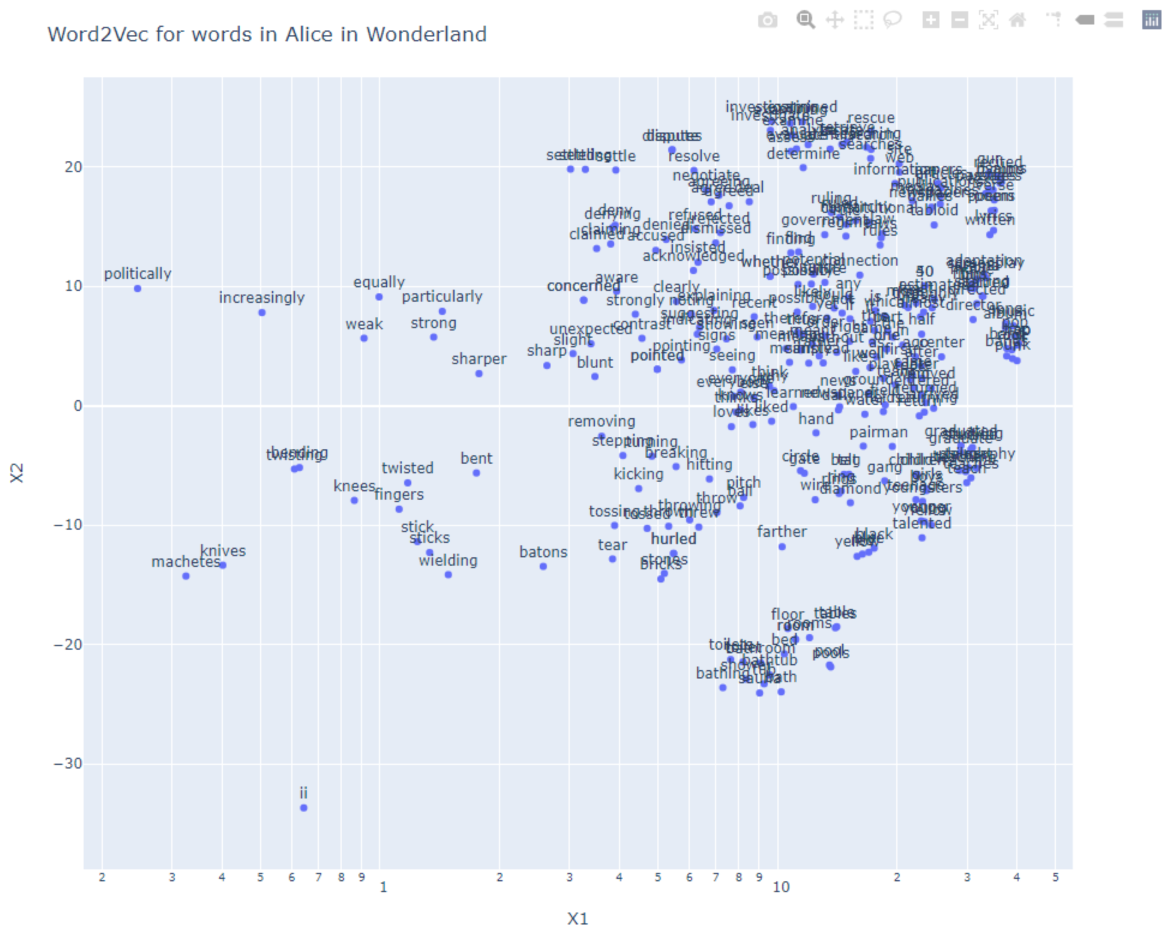
Method	Accuracy	Precision	Recall	F1
Decision Tree	0.619329	0.619329	0.619329	0.619329
KNNs	0.708087	0.708087	0.708087	0.708087
Neural Network	0.733728	0.733728	0.733728	0.733728

So, by looking at t-SNE plot, we can somewhat decide if the data need more preprocessing before passing into Classification models.

3.2 Evaluate the result of embedding techniques

We will consider the embedding problem for language. The goal is to convert words into some kind of data that a machine can perform on. Word2Vec is a method to solve that problem. It transform words into their corresponding vectors.
After apply Word2Vec, we can use t-SNE to visualize if our embedding process is satisfactory. Below is the plot of words from the novel Alice in Wonderland.
We can see t-SNE can not separate a lot of words. But checking some clusters, we can see some positive result:

- Words with same origin tend to stay close with each other
- Some cluster contains many words with similar meaning (For example the top right cluster contains many verb like analyze, examine, investigate, etc.)


Word2Vec for words in Alice in Wonderland
X1
X2
20
10
0
−10
−20
−30
politically
increasingly
equally
particularly
weak
strong
sharper
sharp
blunt
concerned
aware
pointed
removing
bent
twisted
knees
fingers
stick
sticks
wielding
batons
knives
machetes
tear
hurled
kicking
hitting
farther
hand
pairman
talented
floor
bed
pool
bath
resolve
determine
rescue
web
tabloid
written
ii

Chapter 4

Conclusion

Although t-SNE shown advantage properties than other techniques for data visualization, but this method still have three potential weakness:

- It is unclear how t-SNE performs on general dimensionality reduction tasks. We are not obvious how t-SNE will perform on the more general task of dimensionality reduction (i.e, when the dimensionality of the data is not reduced to two or three, but to $d > 3$ dimensions).

- The relatively local nature of t-SNE makes it sensitive to the curse of the intrinsic dimensionality of the data. t-SNE reduces the dimensionality of data mainly based on local properties of the data, which makes t-SNE sensitive to the cruse of the intrinsic dimensionality of the data (Bengio, 2007). In data sets with a high intrinsic dimensionality and an underlying manifold that is highly varying, the local linearity assumption on the manifold that t-SNE implicitly makes (by employing Euclidean distances between near neighbors) may be violated.

- t-SNE is not guaranteed to converge to a global optimum of its cost function. A nice property of most state-of-the-art dimensionality reduction techniques (such as classical scaling, Isomap, LLE, and diffusion maps) is the convexity of their cost functions. A major weakness of t-SNE is that the cost function is not convex, as a result of which several optimization parameters need to be chosen. The constructed solutions depend on these choices of optimization parameters and may be different each time t-SNE is run from an initial random configuration of map points.

This report represents one of the best techniques for the visualization of similarity data that is capable of retaining the local structure of the data while also revealing some important global structure (such as clusters at multiple scales). Both the computational and the memory complexity of t-SNE are $O(n^2)$, but through a landmark approach that makes it possible to

successfully visualize large real-world data sets with limited computational demands. Our experiments on a variety of data sets show that t-SNE outperforms existing state-of-the-art techniques for visualizing a variety of real-world data sets.

Bibliography

[1] Demartines, P. and Hérault, J. (1997). Curvilinear component analysis: A self-organizing neural network for nonlinear mapping of data sets. *IEEE Transactions on neural networks*, 8(1):148–154.

[2] Hinton, G. E. and Roweis, S. T. (2003). Stochastic neighbor embedding. In *Advances in neural information processing systems*, pages 857–864.

[3] Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417.

[4] Sammon, J. W. (1969). A nonlinear mapping for data structure analysis. *IEEE Transactions on computers*, 100(5):401–409.

[5] Torgerson, W. S. (1952). Multidimensional scaling: I. theory and method. *Psychometrika*, 17(4):401–419.