t-Distributed Stochastic Neighbor Embedding (t-SNE)

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

John von Neumann Institute - HCM VNU

May 20th, 2020

Content

- Introduction - What is t-SNE?
- Background and theory
- Applications of t-SNE
- Conclusion

Introduction

- Visualization of high-dimensional data is an important problem
- 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

Introduction

We have two typed dimensionality reduction data:

- Linear techniques as Principle Component Ananlysis, classical multidimensional scaling...
- Non-linear techniques as Stochastic Neighbor Embedding, curvilinear components analysis...

Introduction

- Linear techniques focus on keeping the low-dimensional representations of dissimilar datapoints far apart.
- Non-linear techniques often are used to keep the low-dimensional representaions of very similar datapoints close together, which is typically not possible with a linear mapping

Introduction

What is t-SNE?

- t-SNE is a non-linear visualization technique
- Based on Stochastic Neighbor Embedding (Hinton and Roweis, 2002)
- Much easier to optimize and reducing the tendency to crowd points together in the center of the map
- t-SNE is better than existing techniques at creating a single map that reveals structure at many different scales

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. The conditional probability $p_{j|i}$ is given by:

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

Where σ_i is the variance of the Gaussian that is centered on datapoints

Theoretical Background

SNE

Modeling the similarity of map point y_j to map point y_i by:

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

- Variance of the Gaussian is set to $\frac{1}{\sqrt{2}}$
- Where $q_{i|i} = 0$

SNE aims to find a low-dimensional data representation that minimizes the mismatch between $q_{j|i}$ and $p_{j|i}$

The cost function C is given by

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

- P_i represents the conditional probability distribution over all other datapoints given datapoints x_i
- Q_i represents the conditional distribution over all other datapoints given datapoints y_i

The perplexity is calculated by

$$Perp(P_i) = 2^{H(P_i)}$$

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

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

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

Theoretical Background

SNE

The gradient descent of SNE:

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

The gradient update with a momentum term

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

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

Kullback-Leibler divergences between a jointprobability distribution:

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

The pairwise similarities in the low-dimensional map q_{ij} :

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

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

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

The joint probabilities p_{ij} in the high-dimensional space

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

The gradient of symmetric SNE:

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

SNE and other local techniques such as Sammon mapping suffer from "crowding problem"

It is where it's impossible to model distance correctly when project data-points from higher to lower dimensions.

Mathematically, when the dimensions are high, the gradient acts as attractive forces between points will cause the datapoints to be squashed together.

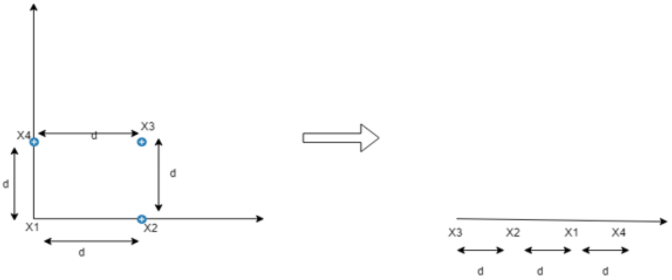


Figure 8 – Crowding problem

The t-SNE can solve all the aforementioned problems with SNE.

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

The pairwise similarities in the low-dimensional map q_{ij} :

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

The gradient of t-SNE:

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

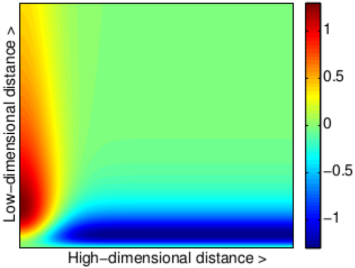


Figure: t-SNE heatmap

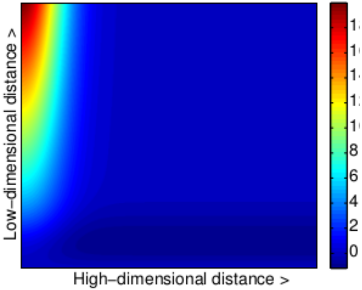


Figure: SNE heatmap

Figure: Comparison between SNE and t-SNE heatmap

Theoretical Background
t-SNE

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

Applications of t-SNE

t-SNE visualization can help gain insight on the structure of data.

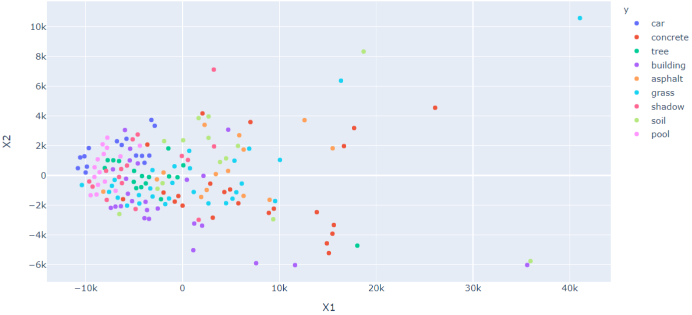
Used in many fields that study data: machine learning, health care, genetics, language processing, etc.

On general, t-SNE applications can be used in:

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

Applications of t-SNE

Exploring data for classification tasks

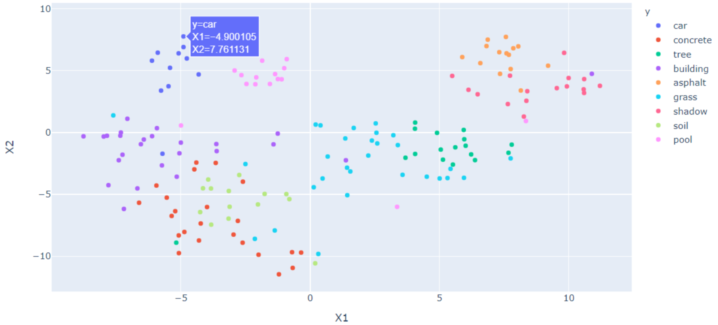


Method	Accuracy	Precision	Recall	F1
Decision Tree	0.727811	0.727811	0.727811	0.727811
KNNs	0.382643	0.382643	0.382643	0.382643
Neural Network	0.420118	0.420118	0.420118	0.420118

Figure: t-SNE on original data (Urban Land Cover dataset)

Applications of t-SNE

Exploring data for classification tasks



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

Figure: t-SNE on preprocessed data (Urban Land Cover dataset)

Applications of t-SNE

Evaluate the result of embedding techniques

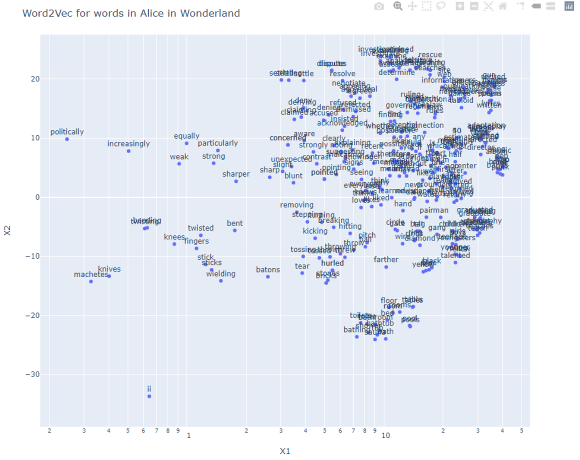


Figure: t-SNE on Word2Vec result

Conclusion

- This is 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
- It possible to successfully visualize large real-world data sets with limited computational demands.
- t-SNE outperforms existing state-of-the-art techniques for visualizing a variety of real-world data sets.

Conclusion

Three potential weakness

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

The end
Thank you for your attention