t-SNE Report

Donal Loitam

July 27, 2022

Contents

1 Introduction

- (t-SNE) t-Distributed Stochastic Neighbor Embedding is a non-linear dimensionality reduction algorithm used for exploring high-dimensional data
- t-SNE is iterative so unlike PCA you cannot apply it on another dataset
- t-SNE is something called **nonlinear dimensionality reduction**. What that means is this algorithm allows us to separate data that cannot be separated by any straight line

Figure 1: Linearly nonseperable data

2 Key Points of the Algorithm

- **Probability Distribution for high dimensional space** :
  Let the scatter plot below be our dataset.It has 3 different classes and you can easily distinguish them from each other.

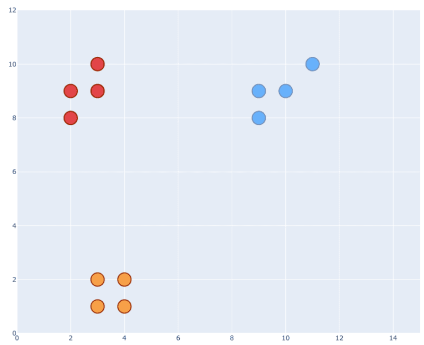


Figure 2: Scatter plot of the data

  - The first part of the algorithm is to create a probability distribution that represents similarities between neighbors. What is "similarity"?
  - " Similarity of datapoint x_j to datapoint x_i is the conditional probability $p_{j|i}$, that 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 ".
  - We've picked one of the points from the dataset. Now we have to pick another point and calculate Euclidean Distance between them $|x_i|x_j|$

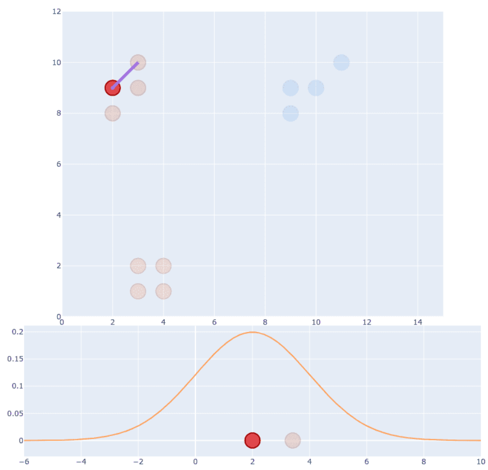


Figure 3: Calculating similarity of x_i, x_j

- The next part of the original paper states that it has to be **proportional to probability density under a Gaussian centered at** x_i. So we have to generate Gaussian distribution with mean at x_i, and place our distance on the X-axis.
- After calculating the first point we have to do the same thing for every single point out there.

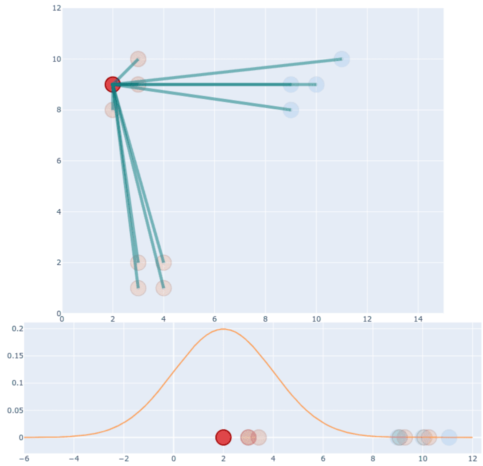


Figure 4: Calculating similarity of x_i, x_j $(\forall i, j)$

- **SCALING THE UNSCALED SIMILARITY VALUES :**

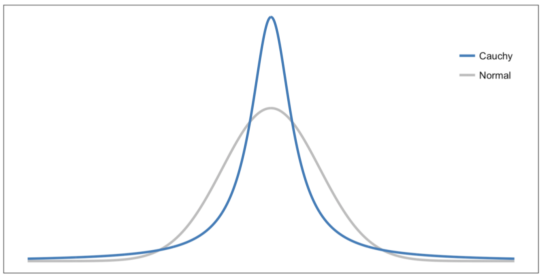
$$\text{scaled score} = \frac{\text{score}}{\text{sum of all scores}} \tag{1}$$

- **Dealing with different distances :** If we take two points and try to calculate conditional probability between them then values of $p_{i|j}$ and $p_{j|i}$ will be different:
  The reason for that is because they are coming from two different distributions.

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

- **Repeat Step 1, but for corresponding Low-Dimensional Space**:
  - The next part of t-SNE is to create low-dimensional space with the same number of points as in the original space.
  - The goal of this algorithm is to find similar probability distribution in low-dimensional space. The most obvious choice for new distribution would be to use Gaussian again.
  - That's not the best idea, unfortunately. One of the properties of Gaussian is that it has a "short tail" and because of that it creates a crowding problem.
  - To solve that we're going to use **t-distribution , (hence the "t" in t-SNE)** with a single degree of freedom
  - This gives us a second set of probabilities (Q_{ij}) in the low dimensional space.
  - The heavy tails allow for better modeling of far apart distances.

Normal vs Cauchy (Students-T) Distribution



- **Low-dimensional space similarity(Q_{ij}) must reflect those of high-dimensional space (P_{ij}) as best as possible :**
  - We measure the difference between the probability distributions of the two-dimensional spaces using Kullback-Liebler divergence (KL)
  - Finally, we use gradient descent to minimize our KL cost function.
  - You can treat that gradient as repulsion and attraction between points. A gradient is calculated for each point and describes how "strong" it should be pulled and the direction it should choose

3 Some Questions

1.
Ans.
2.
Ans.
3.
Ans :
4.
Ans.
5.
Ans.