Data Visualization: T-SNE and UMAP

Kuznetsov Dmitriy
National Research University
Higher School of Economics, Faculty of Computer Science
Moscow, Russian Federation
Email: dskuznetsov_4@edu.hse.ru

***Abstract*—In this paper I view two ways of data visualization: t-SNE and UMAP. The first of them is well-known algorithm, which has its own advantages and disadvantages. The second one is quite new for most part of data engineers. I want to investigate UMAP pros and cons against t-SNE.**

***Index Terms*—Data visualization; t-SNE; UMAP;**

Introduction

Often programmers, analysts and researchers, all those who work with large amounts of data, face the problem: to find certain patterns in the sample. When we begin to work with new information before using any complex models or approaches, it is simply necessary to determine the origin of the data, their form, in particular how different features of objects interact with each other. A good understanding of the structure of the data helps to better choose not only the way to preprocess the sample, but also the choice of a competent model to build an estimate for the distribution, which is to some extent the ultimate goal.

For an initial study, it is often useful to look at the main sample statistics: sample mean, sample variance, or some percentiles. This helps to represent the distribution density of the components of a hypothetical random variable (in fact, there is no exact distribution because some normal noise is almost always present in the data). In this or other way, at the stage of the first interaction with the data, we always try to get the outlines of the density of the multidimensional distribution, to build a multidimensional model of our data in our head. Instead of fantasizing, we can try to draw our data using computer technology. Trying to do this, we are immediately faced with the problem: how to depict the data presented in multidimensional space? This problem is solved by visualization methods presented in this article.

Two visualization algorithms will be presented in the following sections: *t-distributed Stochastic Neighbor Embedding* (t-SNE)[3] and *Uniform manifest Approximation and Projection* (UMAP)[2]. T-SNE is a well-known visualization algorithm that has earned its popularity due to the distinct separation of classes from each other, unlike many other visualization methods. The goal of this approach is to keep the relative distances between objects as good as possible when mapping data from multidimensional space to a plane. Due to this, a good separability of data clusters is achieved. The basis of the t-SNE lies in the statistical method of construction of projection data, in contrast to UMAP, which describes the structure of data using topology. UMAP has a high level of mathematical background, which makes it possible to achieve high performance in the benchmark. The authors of UMAP claim that they managed to overtake t-SNE in terms of execution time and quality of approximation. In this article, we will try to understand this issue, figuring out how each model works.

Main part

In this section, we consider both methods of constructing a projection on the plane and highlight the features of the models, based on their method of visualization. In the conclusion of the Chapter, we compare the results of both methods on popular examples.

First of all, let's formulate the problem. Let there be a sample $X = (x_1, \dots, x_n)$ obtained from some unknown multidimensional distribution d. Our task is to obtain a new sample $Y = (y_1, \dots, y_n)$, but in the new space $\mathbb{R}^2$, such that the sample was as close as possible to the original in the context of the structure.

Before proceeding directly to the consideration of t-SNE, let's deal with a slightly simpler method: SNE. This method solves the problem as follows:

We define for each object of the initial sample a random variable $d_i \sim \mathrm{N}(0, \sigma_i)$ in R. This random variable will describe the distance from some object x to the object x_i (note that this value is random). We have a finite number of points in the sample, then it is easy to conclude the following statement from Bayes' theorem and the law of total probability:

$$p_{ji} = Prob[x = x_j | x_i] = \frac{\exp\{-\|x_i - x_j\|^2 / 2\sigma_i^2\}}{\sum_{t \neq i} \exp\{-\|x_i - x_t\|^2 / 2\sigma_i^2\}}$$

The distribution parameter is found using perplexity evaluation (these technical details will not help us much to answer the desired question).

Similarly, we define a conditional distribution for our projection, with the only difference that we believe that now the distribution is standard:

$$q_{ji} = Prob[y = y_j|y_i] = \frac{\exp\{-\|y_i - y_j\|^2\}}{\sum_{t\neq i}\exp\{-\|y_i - y_t\|^2\}}$$

Our task is to find the second sample so that the conditional distributions of both samples are closest. To estimate the distance of the two distributions from each other, we use the KullbackLeibler divergence:

$$D(P\|Q) = \sum_i p_i \log\frac{p_i}{q_i}$$

We want the distance to be minimal. In this case, our model should solve the problem of minimizing the following functionality:

$$\mathrm{L}(p, q) = \sum_i\sum_j p_{ji}\log\frac{p_{ji}}{q_{ji}}$$

This problem can be solved by any method of smooth optimization, for example, gradient descent.

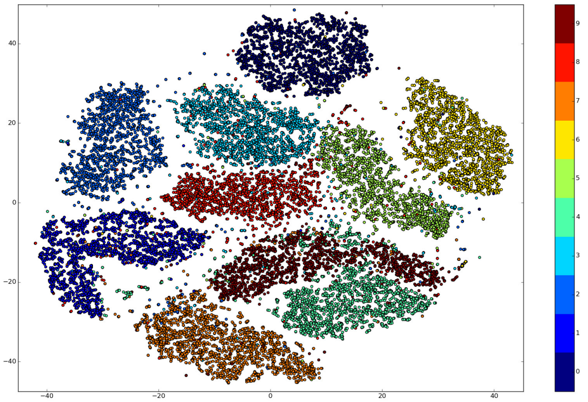
As a result of the work of SNE, we obtain a sample on the plane closest to the original multidimensional in the sense of the Kullback-Leibler divergence.

T-SNE differs from the algorithm just considered only in the distribution for a two-dimensional sample: instead of the standard normal distribution, we assume Student's t-distribution with one degree of freedom. Now the distribution for the two-dimensional sample is as follows:

$$q_{ji} = Prob[y = y_j|y_i] = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{t\neq i}(1 + \|y_i - y_t\|^2)^{-1}}$$

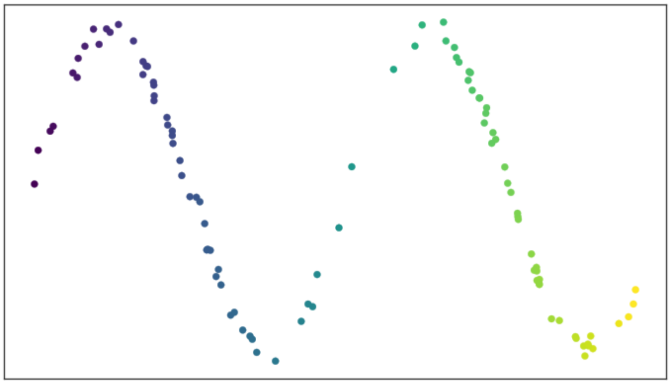
The authors argue that this has significantly improved the convergence of the algorithm and increased the inter-cluster distance, which directly affects the quality of visualization.

The image below shows the result of t-SNE's work on the MNIST dataset.

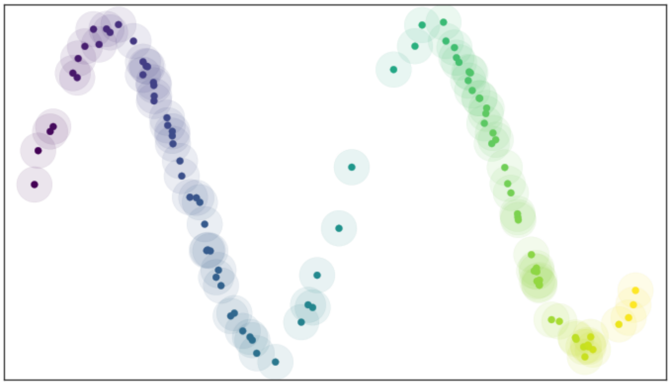


Now let's consider UMAP: To understand the mathematical correctness of this algorithm high mathematical preparation is required, so this article is limited to the superficial analysis of the model.

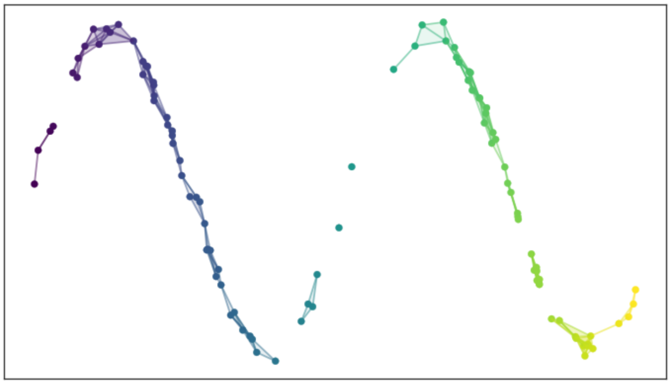
UMAP solves the problem we set earlier as follows: Suppose we have some sample (let's imagine that this is a multidimensional sample):



Let's consider some finite closed cover of this set (as a cover element, take the closed neighborhood of each sample object).

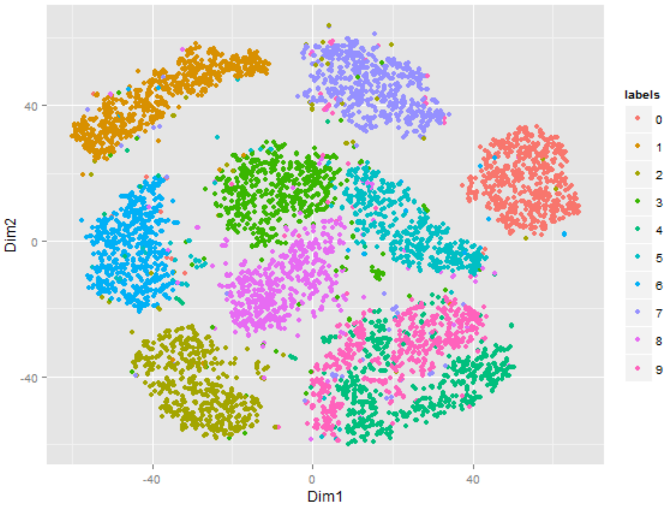


We will build on this cover and sample a simplical complex, i.e. triangulate our topological space defined on the set of objects.

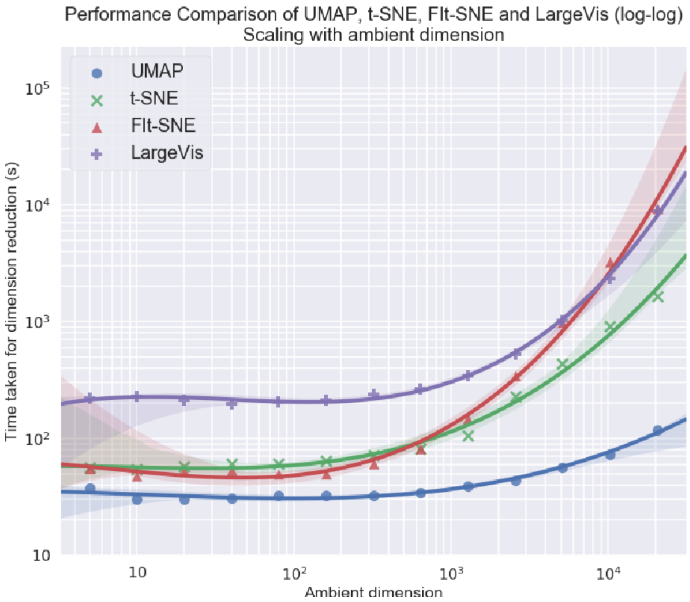


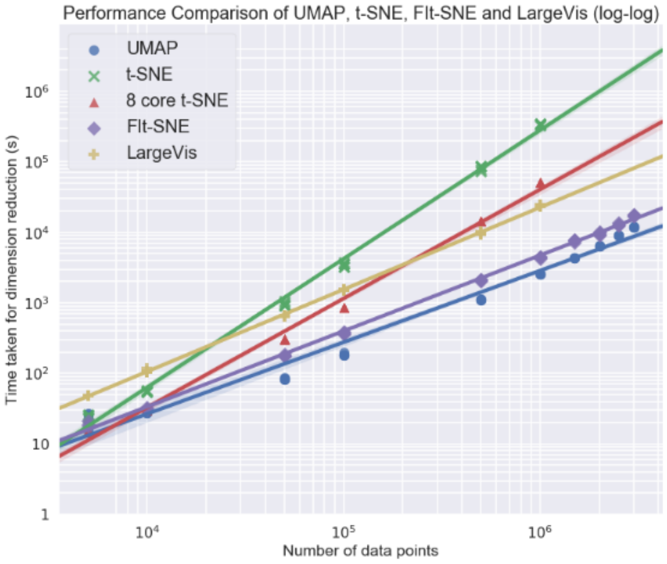
According to the Nerve theorem, the new topological space is homotopy equivalent to the original sample. Thus, we described our data in a simplical complex in some way without losing any information about the data structure. Now, let us construct not an arbitrary, but a simplical complex consisting strictly of simplices of dimension at most 1. In this case, we obtained some connectivity graph. Now we just need to find some Euclidean space in which the image of the graph will accurately convey the structure of the original data. Thus, we reduced the original data visualization problem to the *Graph Layout problem* (it can be solved by other methods) with the help of some knowledge about topological spaces. In this article, we will not focus on the intricacies of the implementation of this method in practice. One has only to note that in reality it is not as easy as in theory. This approach will only work on the assumption that the data is uniformly distributed. However, after some practical additions, the essence of the algorithm is preserved.

This is how the result of UMAP on the MNIST dataset looks like:



Let's move directly to the comparison of models: The authors of UMAP argue that the main advantage of their model over t-SNE (which is currently one of the best methods of visualization) is evaluating time and and the comparable quality of the approximation. First of all, we note that both models have a high inter-cluster distance. Both models achieve high quality in mixed density separation. Moreover, both models are constructed in such a way that as a result of the work the proportions of distances between objects are preserved (the closer the object was in the original space, the closer it is in the new one). However, for many users it is no secret that t-SNE can hardly cope with the visualization of datasets with a large number of features or samples (this is also noticed by the authors of the articles). Therefore, before rendering such samples, it must be preoprocessed using one of the methods of dimension reduction, e.g., PCA based on SVM-decomposition. This assumption not only increases the visualization time, but also the quality of the approximation, because usually when using PCA, the exact decomposition is not considered, but the estimate for the matrix decomposition is found. Also unlike most the implementations of t-SNE, UMAP does not trumpet using space-trees (used in better t-SNE implementations). Every model has its drawbacks, UMAP is no exception. Unlike some of the vusualization models and methods of dimension reduction, the results of UMAP are uninterpretable, that is why the result informativity decreases. For example, the new feature space produced by PCA is the axises of greatest variance. This trait of UMAP turns out to be a serious drawback when using the visualization method, because our goal is to understand the nature of the data as much as possible. As a confirmation of the above statements, I propose to consider the performance on the dataset *Google News*. On this benchmark UMAP shows results on the order of magnitude exceeding the results of t-SNE:





CONCLUSION

In conclusion, it is necessary to sum up the findings. T-SNE and UMAP show similar data visualization results in a low-dimensional feature space. However, as we have already found out with increasing the dimension of the space, the execution time and the quality of the t-SNE visualization deteriorates. Also, this model has a poor resistance to *the problem of high-dimensional*. UMAP in turn has complexity and quality much higher. These advantages make UMAP the most attractive method for visualizing data in case if we want to look exclusively at the data structure without interpreting the resulting feature space in any way. Both articles and the models proposed in them make a great contribution to the development of data engineering, since they significantly increased the level of quality of models in the visualization models class. Before writing this paper, I was not familiar with the UMAP algorithm. During the analysis of the author's article, it was a surprise for me to get acquainted with an extraordinary approach to data visualization, having some experience in this area before. From my personal experience, the problem of t-SNE with high-dimensional data was critical for me, that is why I preferred other less qualitative, but faster models than t-SNE when choosing the visualization method. After the advent of UMAP, there is an excellent alternative to t-SNE when working with big data. In any case, in the field of machine learning and data analysis, it is not clear that one model is better than the other. The solution of each problem is non-deterministic, so the choice of data visualization model depends entirely on the task and the source data, but my personal preference at this moment I give UMAP.

References

[1] V. A. Vasiliev *Introduction to topology*

[2] Leland McInnes, John Healy, James Melville *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction*

[3] Laurens van der Maaten, Geoffrey Hinton *Visualizing Data using t-SNE*