CCTS – Data visualization with t-SNE and clustering with DBSCAN

Chiara Maccani, Samuele Piccinelli, Tommaso Stentella, and Cristina Venturini
(Dated: May 15, 2021)

In the field of machine learning, unsupervised learning refers to a class of algorithms that learns patterns from unlabeled data. Clustering in particular is an unsupervised learning technique used to group data together based on shared similar characteristics and broadly used in many applications such as market research, pattern recognition, data analysis and image processing.
We investigate the behavior of 2 different algorithms, namely t-SNE and DBSCAN by employing 2 labeled datasets. Some fundamental aspects of these algorithms, which are relevant in many data based applications, are explored and the resulting discussion offers an opportunity for highlighting the fundamental gist of machine learning: there's no general rule that applies to all datasets but each algorithm must be picked based on the data itself.

INTRODUCTION

The purpose of this work is to test the functionalities of two algorithms: t-SNE and DBSCAN.
t-distributed stochastic neighbor embedding (t-SNE) is a non-linear technique for visualizing high-dimensional data, first introduced by G. Hinton *et al* [1] in 2008. t-SNE preserves local similarities and space relationships while creating a lower dimensional representation of multi-dimensional data.
The DBSCAN algorithm is a clustering method that relies on a density-based notion of clusters and was first proposed by M. Ester *et al* [2] in 1996. Given a set of feature points, DBSCAN groups together points that are closely packed while marking as outliers points that lie in low-density regions.
More precisely, we highlight their dependencies from the data structure and the hyper-parameters, with the goal of understanding the different use cases within the framework of unsupervised learning.
The adopted workflow is as follows: the chosen algorithm is executed multiple times with different parameters and then performance is evaluated. Such assessment is obtained either by visual cue or by means of the *Normalized Mutual Information* (NMI) between predicted clusters and labeled ones.

METHODS

The analysis is performed using `Python`: both t-SNE and DBSCAN are available as `sklearn` [3] objects, so these versions are used.
As mentioned before, we consider 2 datasets with different structures: as a consequence the analysis we perform differs in the two cases.
First Dataset. The first dataset is composed by embedded manifolds, which represent 3 clusters with a closed linear structure. Given $N = 800$ data points, the first 10% belongs to cluster 0 (*red*), the next 30% to cluster 1 (*green*) and the last 60% to cluster 2 (*blue*). A visual representation is given in Fig. 1.

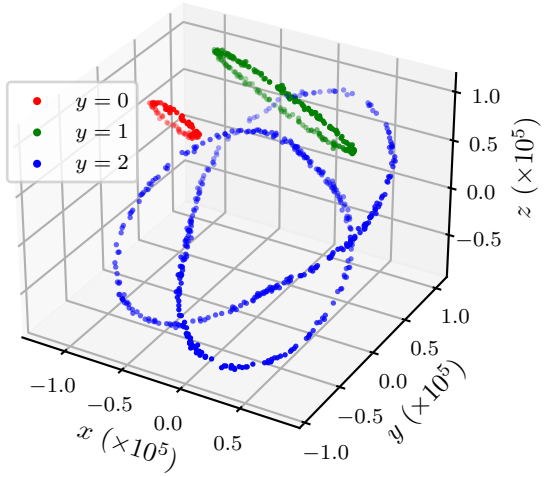


Figure 1. First dataset projected along the first 3 dimensions.

t-SNE. The main hyper-parameter of t-SNE is the *perplexity* (perp). We choose 3 different values: perp $= [5, 30, 100]$. 2 initialization routines are explored: random and principal component analysis (PCA) based. The optimal value of the perplexity parameter is the one that allows the algorithm to visually group the data in the best way according to the known labels. The dimension of the latent space is 2.

DBSCAN. In this section we analyze the clustering process with DBSCAN in $d = 5$ dimensions. DBSCAN has two main hyper-parameters to be tuned: eps and minPts. To understand a good range for the cutoffs value of eps, we sort the minimum distances to the k^{th} nearest neighbor in ascending order and plot them. For this part, we refer to the works in [4] and [5]. In the latter, for any given $k \leq 1$ the authors define a function mapping each point to the distance from its k^{th} nearest neighbor (implemented in the `NearestNeighbors.kneighbors()` command in the `scikit` package). The suggestion is to fix $k = 2d - 1$, with d being the dimension of the feature space. The ideal value for eps would be then equal to the distance value at the "crook of the elbow", i.e. the point

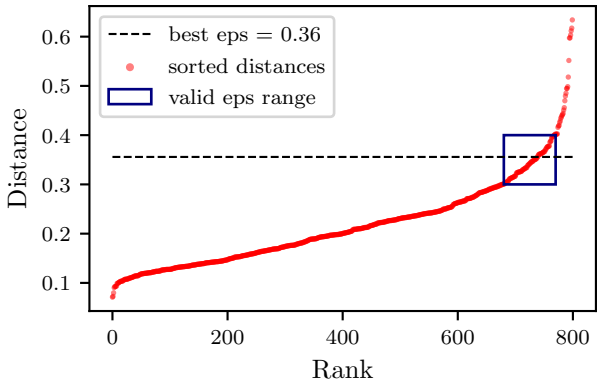
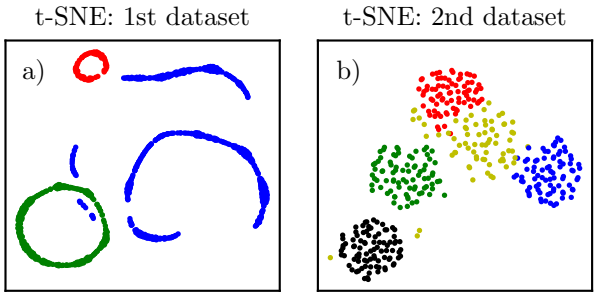


Figure 2. Sorted distances to the k^{th} neighbor.



PCA, perp = 45 PCA, perp = 40

Figure 3. t-SNE applied on both datasets.

of maximum curvature in the nearest neighbor curve. In order to find it we calculate the slopes $\frac{y_n - y_{n-1}}{x_n - x_{n-1}}$ and evaluate the percentage difference for each consecutive point. The best value for eps is given by the corresponding ordinate showed by the black dashed line in Fig. 2. minPts is fixed to $k+1$ as suggested by the cited literature.
The algorithm is run for such values of eps and minPts and performance is evaluated through the NMI score. The value of NMI quantifies the difference between true and predicted labels: a value of 1.00 stands for 100% accuracy.
We then test the algorithm on a broad set of hyperparameters, to empirically assess how performance changes and show the results in a heatmap.

Second Dataset. The second dataset is composed by $N = 400$ sequences of length $L = 36$ formed by bits (0 or 1) generated with equal probability. For each of those, some bits are overwritten following a pattern which is chosen through a uniform distribution among a set of 5 possible ones. The labels associated to each sample are the number of the corresponding pattern, $y = \{0, 1, 2, 3, 4\}$. The goal is to build a model capable of recognizing the presence of these patterns.

t-SNE. In order to visualize the data, dimensionality is reduced through t-SNE from $L = 36$ to $L' = 2$. 6 values of the perplexity parameter (perp $= [5, 20, 35, 40, 45, 500]$) and both initialization routines are tested.
We fix perp $= 40$ and use PCA initialization for the following analysis.

DBSCAN. DBSCAN is applied to the original dataset, using as metric first the $L1$ norm and then the $L2$ norm. We perform an optimization on the parameters of the algorithm comparing the DBSCAN's predicted labels and the real ones through NMI.

DBSCAN on t-SNE output. The same procedure is applied to the data transformed through the optimal t-SNE model. The $L2$ norm (euclidean metric) is used.

DBSCAN on PCA output. When dealing with high dimensional datasets, finding relevant neighbors becomes hard. In order to deal with this, one of the possible solutions is to perform a dimensionality reduction, here through PCA (`sklearn` object as `decomposition.PCA`), and to then apply the clustering algorithm on the reconstructed data. In particular, we retain the first 4 principal components and then apply DBSCAN, varying parameters in order to obtain the best NMI score.

KMeans. We also test the KMeans algorithm (`sklearn` object as `cluster.KMeans`) on the original dataset, with the goal of evaluating its performance, always through NMI score.

RESULTS

First Dataset. Visualization of the data through t-SNE is well achieved with perp $\in [30; 70]$ and both PCA and random initialization (Fig. 3a). However, PCA best maintains the original structure of the data, even for small values of perp (e.g. 30), for which random initialization fails. In [6], [7] the authors cover the topic of the different initialization. The considerations they make, although concerning specific applications to single-cell transcriptomics, can be generalized. The advantage of using PCA-based initialization (an *informative* initialization) is that it preserves the global structure of the data. Such structure is injected into the t-SNE embedding which is then maintained during the course of t-SNE optimization of the fine structure [6].
All assessments concerning t-SNE results are made by visual cue: knowing the labels of the points, plotting the results makes evident which combinations of parameters are successful in capturing the data structure.
In fig. 4, we show the results of DBSCAN with HP estimated with the methods explained in *Methods*. The parameter Nc is the number of clusters found by the algorithm; the unclassified points are shown in light-grey. Performance when using parameters suggested by the literature is poor (Fig. 4a). A working example is shown in Fig. 4b: the parameters come from empirical assessment and are chosen to be as close as possible to the ones suggested in literature. All values of HP considered in the study are shown in the heatmap in Fig. 5, which showcases the corresponding NMI score. It can be observed

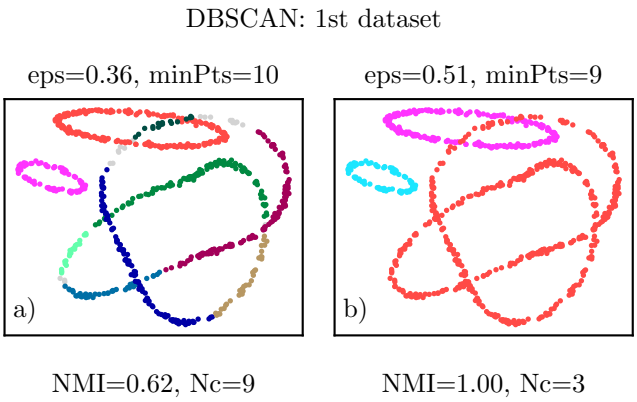


Figure 4. DBSCAN on 1st dataset for 2 HP configurations.

that, for growing values of minPts, one can reach a score close to unity, provided a consequently larger value of eps. In particular, for minPts = 10 the highest score is

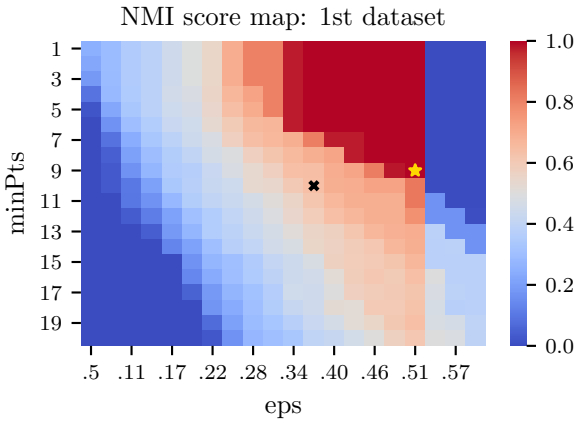


Figure 5. NMI score map, 1st dataset: × for 4a, ⋆ for 4b.

NMI= 0.83 with eps = 0.51. For greater values of eps the algorithm fails.

Second Dataset. What emerges from the t-SNE analysis is that the optimal range for perp is around a few tens: for the extreme values the cluster structure is not conserved. PCA initialization does not yield significant improvement from the random one. Clusters $y = 1$ and $y = 4$, since they share similar sequences, are closest in the initial space. t-SNE maintains this relation in the latent space.

For the original dataset, in most cases DBSCAN fails to find any cluster and in the case of large parameters it groups all the samples in a single cluster (NMI ≤ 0.68) (Fig. 7a). The original samples are binary and high dimensional and in this space the notion of density drifts away from our intuition. The samples could be represented as vertices of an L-dimensional unitary cube. The regular and discrete nature of the data is reflected in the minimum distances between samples and consequently in the ability of the DBSCAN algorithm of propagating through the cluster. Moreover the set of possible distances is not enough to have well separated clusters, since it is very limited both by the regularity and the high-dimensionality. As a consequence, clusters do not present well defined separation volumes. This situation is most critical for DBSCAN. On the other hand, mapping the data to a new 2-dim space with t-SNE allows a magnification of the densities. This method results in better distinguishable clusters with NMI reaching 0.92 (Fig. 7b). In this approach, however, the majority of the learning of the cluster structure is done by t-SNE. Furthermore, it is, in general, bad practice to apply a density or distance based algorithm on the output of t-SNE. In fact, t-SNE does not preserve distances nor density, only to some extent nearest-neighbors. This means that t-SNE will sometimes work, but one will never know whether the found clusters are real, or just artifacts [8]. However, in this case, since we have the labels, we can safely apply the algorithm and verify *a posteriori* if the results we find are in agreement with reality.

We perform PCA reduction to 4 dimensions. This choice

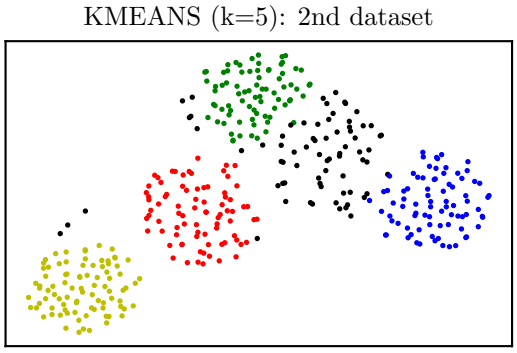


Figure 6. KMEANS on 2nd dataset.

is heuristic: the first 4 principal components (PC) account only for 34% of the total variance but from the 4th PC onward the drop in percentage of two consecutive variance ratios is $< 0.1\%$. In other words, the curve of the eigenvalues of the covariance matrix displays a slope in the order of 10^{-3} starting from the 4th value on. This seems to suggest that the information makes up for a small part of the total data and the rest is mostly "noise" [9]. This is justified *a posteriori* by how the data are generated, i.e. by enforcing a specific pattern in a random bit sequence. In fact, of the whole 36-bit string only $\sim 30\%$ on average is relevant signal (8 to 16 bits of pattern). By reconstructing the data using just this subset of PCs and applying DBSCAN, we reach NMI = 0.89 (Fig. 7c). Applying PCA de-noising impacts majorly the performance of DBSCAN. This confirms that a preliminary analysis that tries to infer on the nature of the data is a decisive approach to unsupervised learning. A well-thought dimensionality reduction can improve results and be safely applied even without knowing the correct labels. Still, it's worth noting that being able to confirm one's hypothesis on the

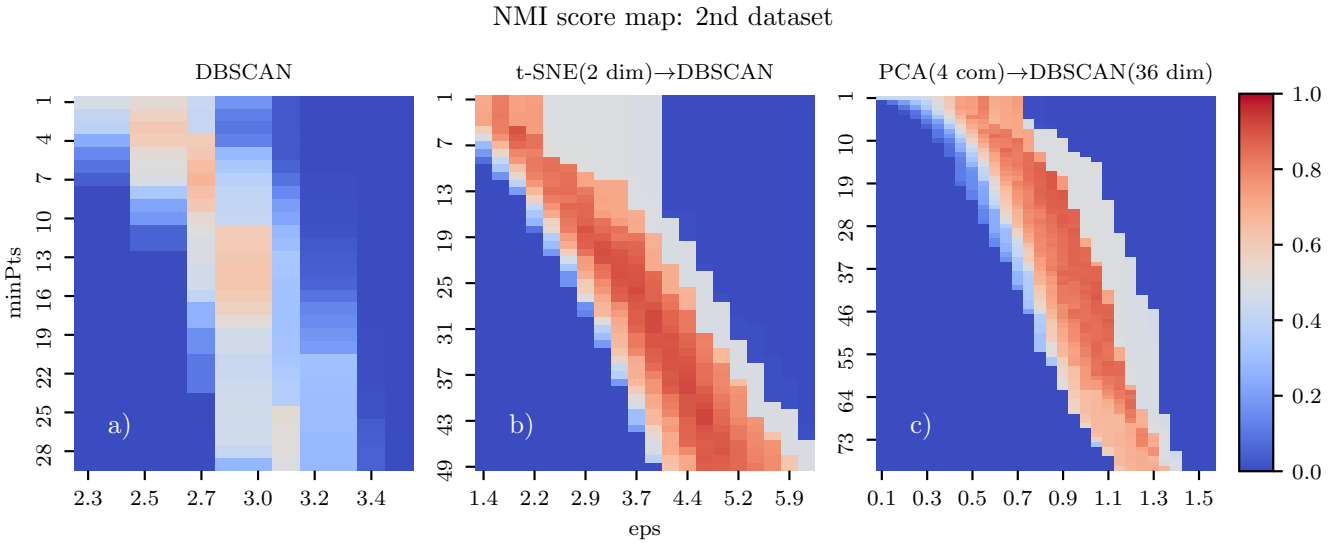


Figure 7. NMI score map, 2nd dataset.

data structure is a considerable advantage.
We also implement the KMEANS algorithm, which is able to perform far better than DBSCAN, reaching a NMI = 0.97 on the 36-dim data (Fig. 6). This highlights once again how density is not a good descriptor for this dataset and that a distance based approach yields a better performance.

CONCLUSION

In this work we present a comparison between the performances of different algorithms, in the context of unsupervised learning, analyzing datasets with different structures. This kind of evaluation is made possible by knowing the true labels. What emerges is a dependence of the performance on the hyper-parameters and on the chosen algorithm. The main differences between the two datasets were the presence or absence of a wide spatial density range and the shape of the clusters. While in the case of the first dataset both t-SNE and DBSCAN perform well, for the second one, because of the high dimensionality of the feature space, analysis is more complex. A distance-based clustering method and a dimensionality reduction proved better than density-based clustering.
The bottom line is to pick the algorithm with a data-based approach, especially when dealing with high dimensional datasets: choosing an off-the-shelf method is not enough.

[1] L. van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(2):2579–2605, 2008.

[2] E. Martin et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, page 226–231, 1996.

[3] F. Pedregosa et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[4] N. Rahmah et al. Determination of optimal epsilon (eps) value on DBSCAN algorithm to clustering data on peatland hotspots in sumatra. *IOP Conference Series: Earth and Environmental Science*, 31, 2016. https://doi.org/10.1088/1755-1315/31/1/012012.

[5] J. Sander et al. Density-based clustering in spatial databases: The algorithm GDBSCAN and its applications. *Data Mining and Knowledge Discovery*, 2(2):169–194, 1998. https://doi.org/10.1023/a:1009745219419.

[6] E. Becht et al. Dimensionality reduction for visualizing single-cell data using UMAP. *Nature Biotechnology*, 37(1):38–44, 2018. 10.1038/nbt.4314.

[7] D. Kobak et al. The art of using t-SNE for single-cell transcriptomics. *Nature Communications*, 10(1), 2019. 10.1038/s41467-019-13056-x.

[8] E. Schubert and M. Gertz. Intrinsic t-stochastic neighbor embedding for visualization and outlier detection. In *Similarity Search and Applications*, pages 188–203. Springer International Publishing, 2017.

[9] I. T. Jolliffe. Principal component analysis and factor analysis. In *Principal Component Analysis*. Springer, 2002.