



## Visualizing high-dimensional datasets using t-SNE

---

Before ending this chapter, I want to introduce the reader to a very powerful algorithm called **t-Distributed Stochastic Neighbor Embedding (t-SNE)**, which can be employed to visualize high-dimensional dataset also in 2D plots. In fact, one of the hardest problems that every data scientist has to face is to understand the structure of a complex dataset without the support of graphs. This algorithm has been proposed by Van der Maaten and Hinton (in **Visualizing High-Dimensional Data Using t-SNE**, Van der Maaten L.J.P., Hinton G.E., *Journal of Machine Learning Research* **9** (Nov), 2008), and can be used to reduce the dimensionality trying to preserve the internal relationships. A complete discussion is beyond the scope of this book (but the reader can check out the aforementioned paper and **Mastering Machine Learning Algorithms**, Bonaccorso G., Packt Publishing, 2018), however, the key concept is to find a low-dimensional distribution so as to minimize the Kullback-Leibler divergence between it and the data generating process. Clearly, a few mathematical tricks are needed to carry out this task efficiently, but the only fundamental concept that is useful for a beginner is the **perplexity**, which is defined as follows:

$$Perplexity(p) = 2^{H(p)}$$

In other words, the perplexity is directly proportional to the entropy of distribution (when computed using  $\log_2(x)$ , it becomes a sort of inverse operation) and by minimizing it, we reduce the uncertainty about the target distribution, given the original data generating process. Therefore, all t-SNE implementations require that you specify the target perplexity (normally in the range 10 ÷ 30), and the computational time grows inversely proportionally.



To show the power of this algorithm, let's suppose that we want to visualize the digits of the MNIST dataset in a bidimensional plot to check whether similar digits are closer than different ones. Let's start by loading and normalizing the dataset:

Copy

```
import numpy as np

from sklearn.datasets import load_digits

digits = load_digits()
X = digits['data'] / np.max(digits['data'])
```

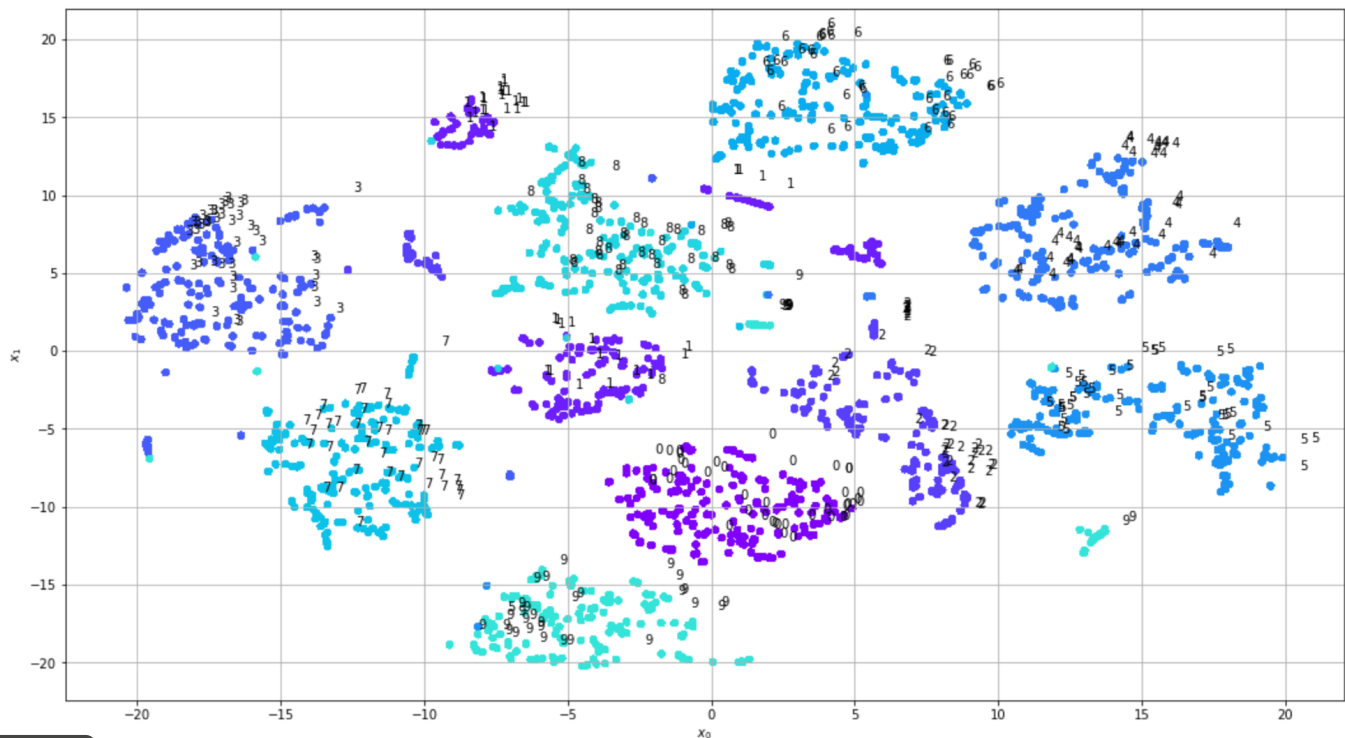
At this point, we can instantiate the scikit-learn `TSNE` class with `n_components=2` and `perplexity=20`, and fit it with the original dataset (which has 64 dimensions):

Copy

```
from sklearn.manifold import TSNE

tsne = TSNE(n_components=2, perplexity=20, random_state=1000)
X_tsne = tsne.fit_transform(X)
```

The final result is shown in the following screenshot (where 400 samples are shown):



## t-SNE plot of the MNIST dataset (limited to 400 digits)

As it's possible to see, the digits are grouped (clustered) coherently with their original distribution with only a few errors, which is probably due to the excessive deformations (different `random_state` values can yield slightly different final configurations). However, t-SNE is extremely powerful, and most of the current implementations are fast enough to process very large high-dimensional datasets in a short amount of time. Therefore, I suggest that you employ it whenever it's helpful to have a graphical representation of the data. This algorithm, which belongs to the family of manifold learning ones, implicitly confirms the manifold assumption. In fact, even if we have 64 original dimensions, the samples are grouped into dense chunks that can be represented by using a smaller number of dimensions. In this particular case, we have also obtained an implicit clustering because many blocks are not only very dense, but they are separated from the other ones and, considering the central point (centroid), it's possible to predict if a digit belongs to the group by computing the distance from the center. In general, this is not always so simple, but we have a further confirmation that the dimensionality reduction has a strong mathematical rationale and it mainly works thanks to the internal structure of real datasets.

---

[◀ Previous Section \(/book/big\\_data\\_and\\_business\\_intelligence/9781789347999/3/ch03lvl1sec30/visualizing-high-dimensional-datasets-using-t-sne/\)](/book/big_data_and_business_intelligence/9781789347999/3/ch03lvl1sec30/visualizing-high-dimensional-datasets-using-t-sne/)

[Next Section ▶ \(/book/big\\_data\\_and\\_business\\_intelligence/9781789347999/3/ch03lvl1sec31/visualizing-high-dimensional-datasets-using-t-sne/\)](/book/big_data_and_business_intelligence/9781789347999/3/ch03lvl1sec31/visualizing-high-dimensional-datasets-using-t-sne/)

---

