How to Use t-SNE Effectively

Academic Review by William Steimel

Sources

- Wattenberg, et al., "How to Use t-SNE Effectively", Distill, 2016. http://doi.org/10.23915/distill.00002
  - MARTIN WATTENBERG Google Brain
  - FERNANDA VIÉGAS Google Brain
  - IAN JOHNSON Google Cloud
  - Oct. 13, 2016
- What is Distill?
  - A journal
  - Mission
    - "Distill is dedicated to clear explanations of machine learning"

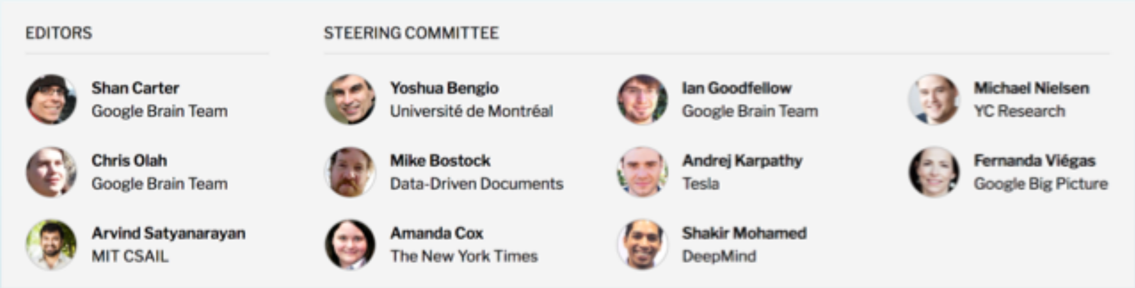


Table of Contents

Motivation

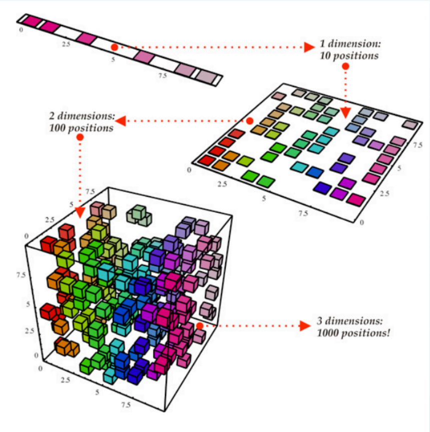
- I chose this topic as I saw T-SNE used in a Music Generation Research Paper for visualizing similarities between notes
- I have not studied Dimensionality Reduction before so I wanted to try understanding one of the popular techniques

Introduction

- This is a more practical tutorial based on a Dimensionality Reduction technique called t-SNE
- t-Distributed Stochastic Neighbor Embedding
  - Introduced by van der Maaten and Hinton in 2008 (Link)
- t-SNE has become popular in the field of machine learning as it can transform high-dimensional datasets into informative two-dimensional data maps
- This paper aims to teach practitioners how to interpret t-SNE results

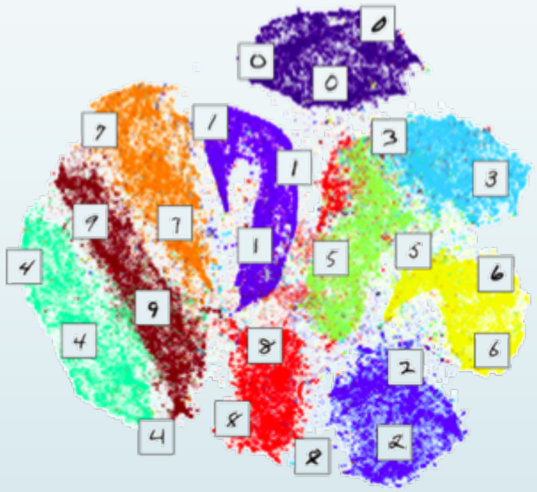
Introduction

- "The aim of dimensionality reduction is to preserve as much of the significant structure of the high-dimensional data as possible in the low-dimensional map."
- **Other Dimensionality Techniques Include:**
- Linear (Traditional Techniques)
  - Principal Components Analysis (PCA; Hotelling 1933)
  - Classical multidimensional scaling (MDS; Torgerson, 1952)
- Non-Linear
  - Sammon Mapping (Sammon, 1969)
  - Curvilinear components analysis (CCA; Demartines and Herault, 1997)
  - Stochastic Neighbor Embedding (SNE; Hinton and Roweis, 2002)
  - Isomap (Tenenbaum et al., 2000)
  - Maximum Variance Unfolding (MVU; Weinberger et al., 2004)
  - Locally Linear Embedding (LLE; Roweis and Saul, 2000)
  - Laplacian Eigenmaps (Belkin and Niyogi, 2002)



Introduction

- How does t-SNE function?
  - t-SNE functions to take a set of points in a high-dimensional space and transform them to accurate representations of these points in a lower dimensional space (usually 2-dimensional)
  - With MNIST Dataset Example:
    - Image is 28x28 meaning 784 dimensions
    - T-SNE can reduce this to two dimensions
  - The algorithm is considered non-linear and adapts to the underlying data distribution
  - t-SNE also has a tunable parameter called "perplexity"



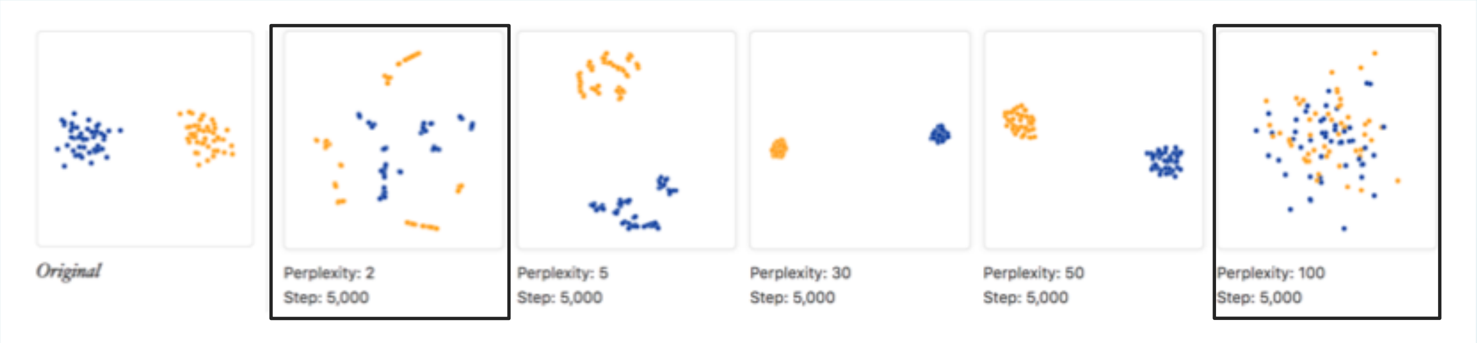
Those Hyperparameters really matter

- This section discusses t-SNE hyperparameters of perplexity and number of iterations
- It begins with a tutorial of a dataset with two widely separated clusters in 2 dimensions.

Original

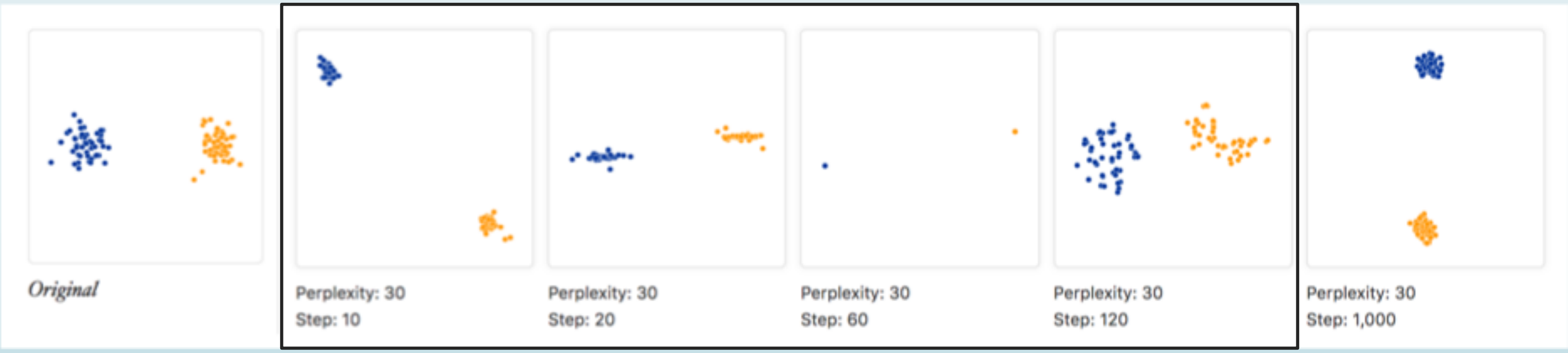
Those Hyperparameters really matter

- **Perplexity**
  - The diagrams below show t-SNE plots for five different perplexity values
    - The original authors (van der Maaten and Hinton) of t-SNE suggested a range between (5-50) in their original paper
    - The writer of this tutorial points out that results get a little strange outside of this range
      - Perplexity 2- Local variations seem to dominate
      - Perplexity 100 – Behavior becomes unexpected- The author asserts that perplexity should be below the number of data points to get meaningful results



Those Hyperparameters really matter

- **Number of Iterations**
- The below shows five runs with different iterations at the same perplexity
  - The first four were stopped before stability (10, 20, 60, 120)
  - It is important to specify enough iterations so that the algorithm converges (reaches a stable configuration)
  - There's no fixed number of steps that will bring a stable configuration and different datasets will have different requirements

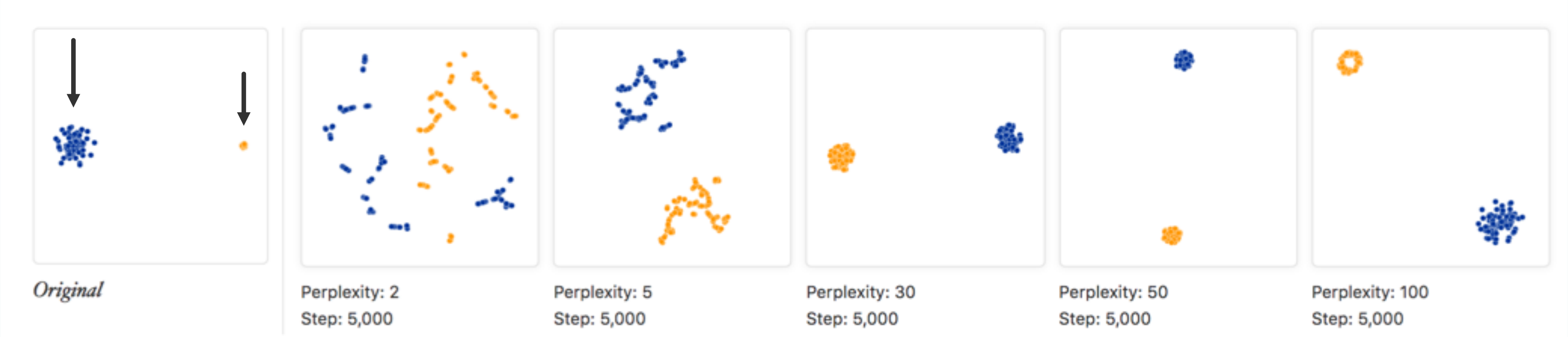


Those Hyperparameters Really Matter

- Do multiple runs with the same Hyperparameters achieve the same results?
  - According to the authors, In this simple example the same global shape is returned but certain datasets will return very different results.
- The rest of this tutorial uses a step size of 5000 as this is enough to reach convergence for the examples in this paper.

Cluster Sizes in a t-SNE plot mean nothing

- If you look at the original data you can see there are two clusters with different standard deviations. One cluster is 10 times as dispersed (spread apart) as the other.
  - The two clusters look to be similar sizes in t-SNE plots
  - What is called "Density equalization" is a predictable feature of t-SNE
  - Dense clusters are expanded, spare clusters are contracted (Evening out the cluster size)
  - "You cannot see relative sizes of clusters in a t-SNE plot"



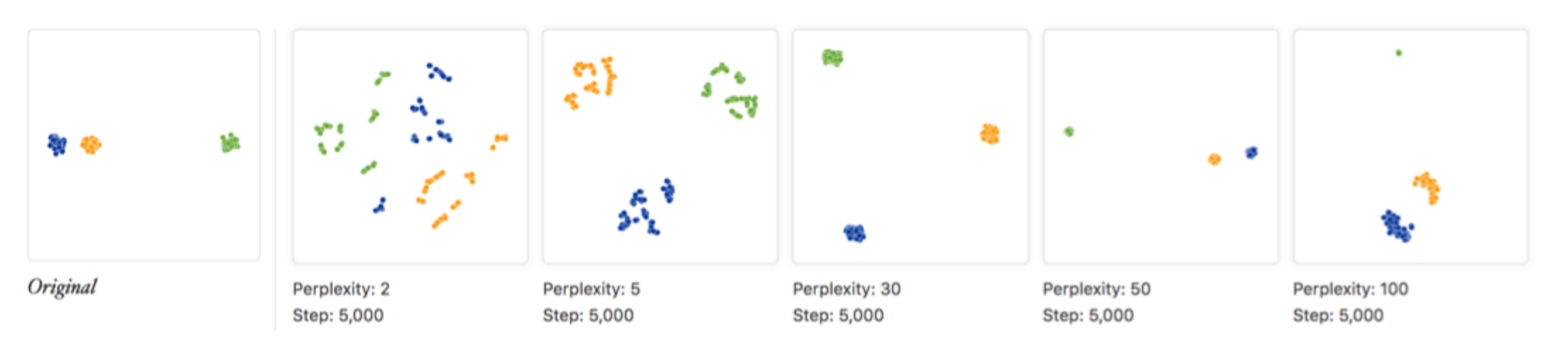
Distance between clusters might not mean anything

- The next section discusses distance between clusters and t-SNE
- The next diagrams show three gaussians of 50 points each with one pair being 5 times as far apart as another pair


Original

Distance between clusters might not mean anything

- Perplexity 50 gives us the best result indicative of the original data's global geometry.
- Since perplexity 50 gave us a good result does that mean we can use perplexity 50 for all datasets to capture global geometry?



Distance between clusters might not mean anything

No!

- The below example is 3 gaussians with 200 points each:
  - Now none of the perplexity values represent the global geometry well
  - Accurate representation of global geometry requires fine-tuning of the perplexity hyperparameter
  - “The Basic message is that distances between well separated clusters in t-SNE plot may mean nothing.”

Random Noise doesn’t always look random

- What about Random Data?
  - The below Diagram shows random data of 500 points drawn from a unit Gaussian distribution in 100 dimensions.
  - Lets see how t-SNE performs under these conditions

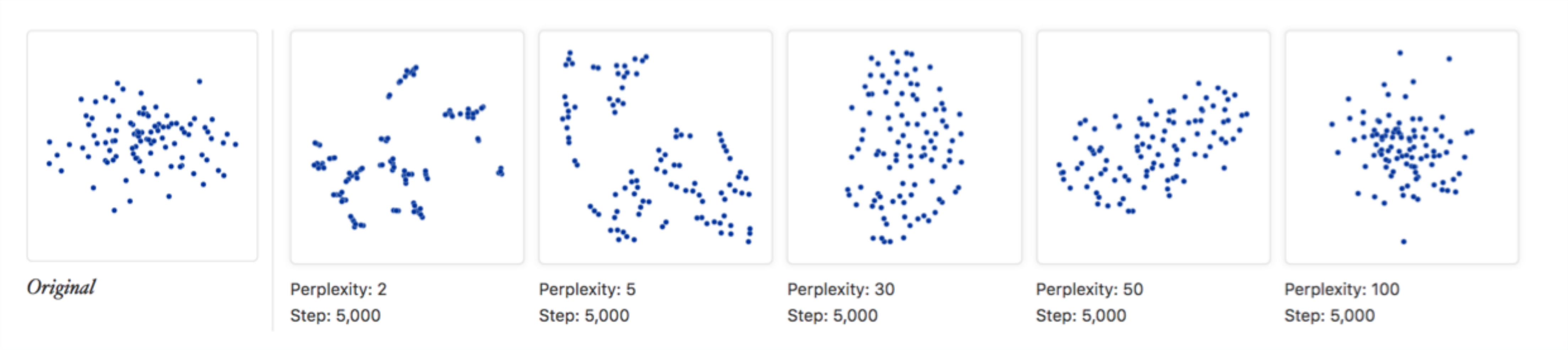
Original

Random Noise doesn't always look random

- The below shows random data plotted with t-SNE at various perplexities
- **Perplexity 2-** Seems to show defined clusters
  - These clusters are random noise (low perplexity values often lead to this type of behavior)
  - Recognizing these clumps as random noise is an important part of reading t-SNE plots

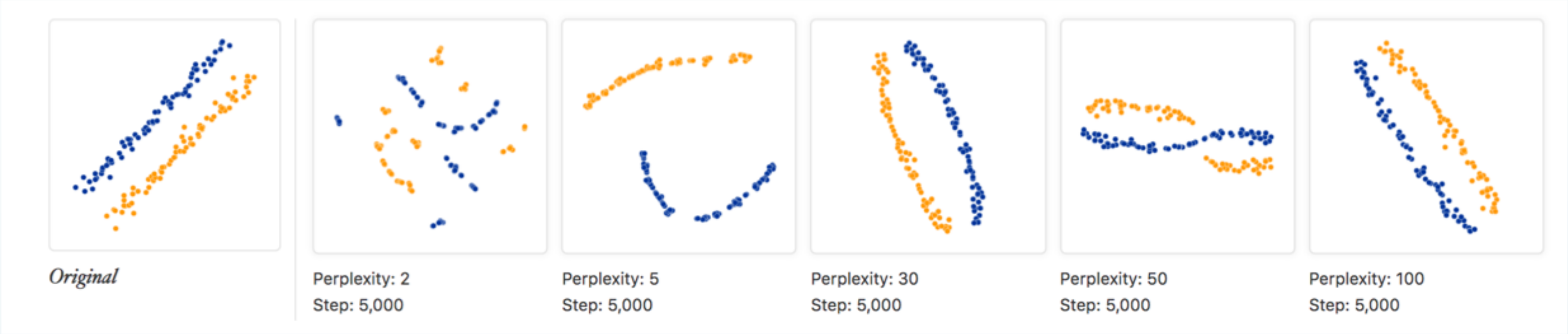
You can see some shapes, sometimes

- The next example takes a look at an axis-aligned Gaussian Distribution in 50 dimensions, where the standard deviation in coordinate i is 1/i.
  - This is essentially long ellipsoidal cloud of points
- Low Perplexity- Meaningless clumping and clustering takes shape
- High Perplexity- Elongated shape becomes more apparent



You can see some shapes, sometimes

- The next example takes two clusters of 75 points each arranged in parallel lines with a little noise.
- For some range of perplexity the clusters look correct but the lines are slightly curved outward in the t-SNE diagram.
  - t-SNE usually expands denser regions of data and since the middles of the clusters have less empty space the algorithm magnifies them



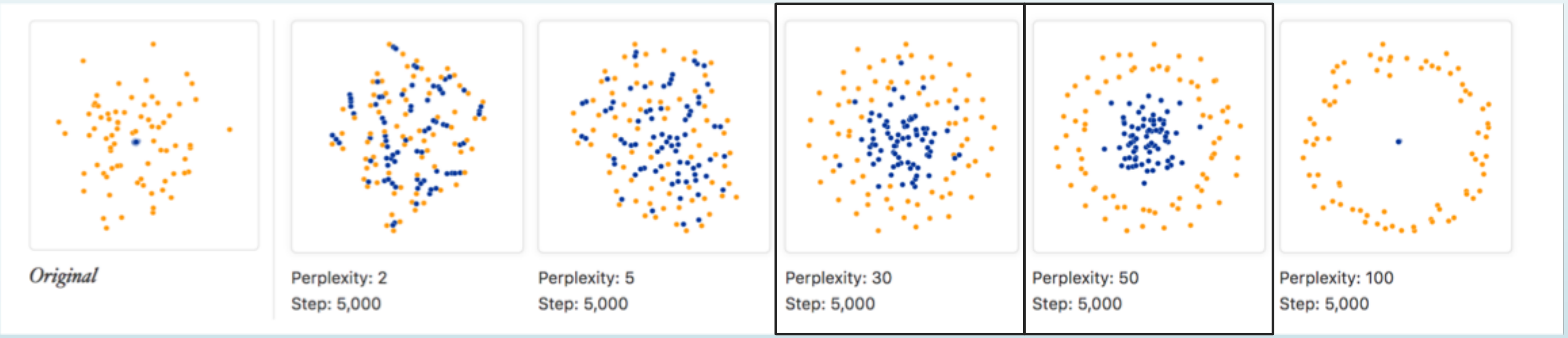
For topology, you may need more than one plot

- Sometimes you can derive topological information from t-SNE plots but may require views at multiple perplexities
  - The next example shows two groups of 75 points in 50 dimensional space sampled from two symmetric gaussian distributions
  - One distribution is 50 times more tightly dispersed (blue) than the other
  - Essentially, the smaller distribution is contained in the larger one.

Original

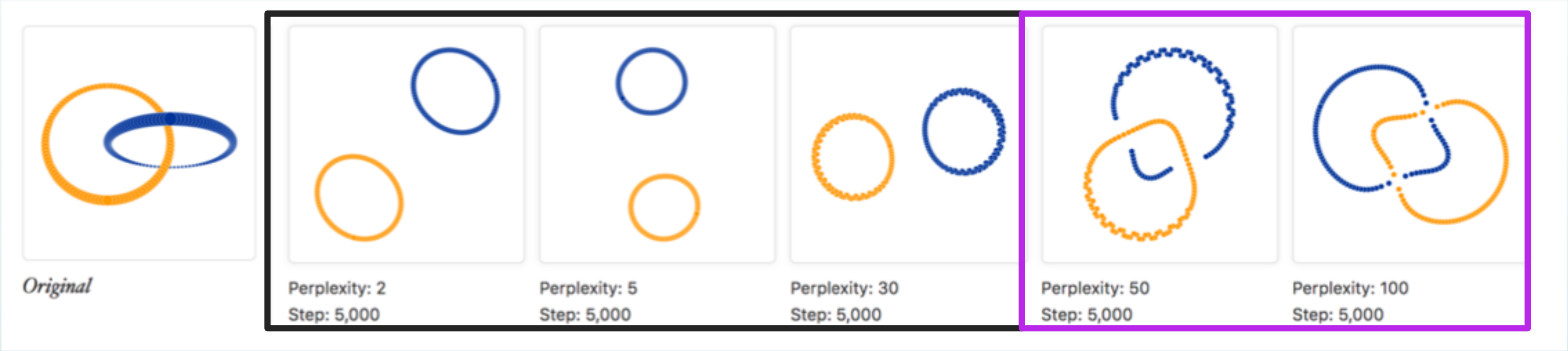
For topology, you may need more than one plot

- Perplexity 30 – Shows the basic topology correctly, but t-SNE exaggerates the size of the smaller group of points
- Perplexity 50 – Outer points become a circle
- Lets look at more complex types of topology



For topology, you may need more than one plot

- The next example takes a set of points that trace a link or a knot in three dimensions.
- Looking at multiple perplexity values gives the most complete picture
  - Low perplexity – two completely separate loops
  - High perplexity – global connectivity



For topology, you may need more than one plot

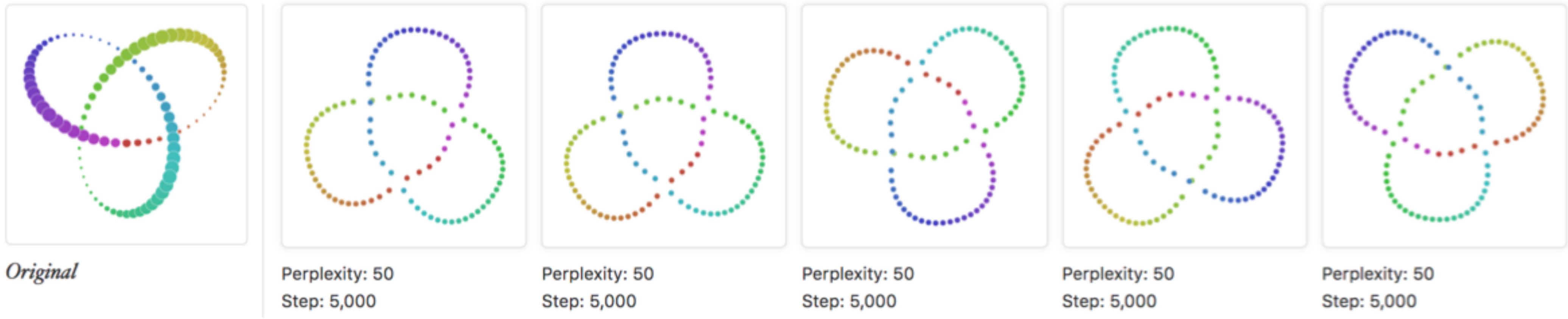
- The Trefoil knot is another example that is interesting
  - Multiple runs of the t-SNE affects the outcome

Original

For topology, you may need more than one plot

Perplexity 2: The algorithm settles twice for a circle but three times results in solutions with artificial breaks

Perplexity 50: Visually identical results which shows some problems are easier to optimize than others

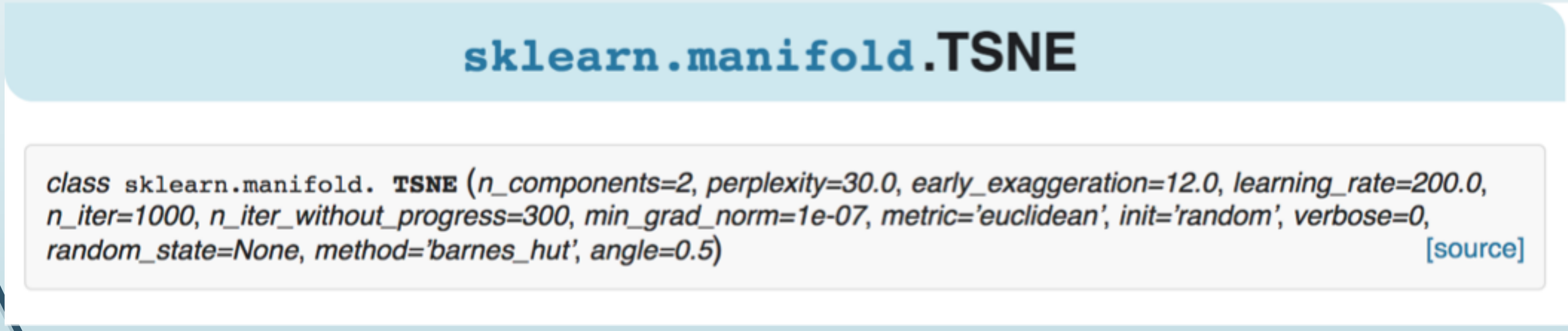


Conclusion

- There are many reasons why t-SNE is very popular
  - Flexibility
  - Can find structure where other dimensionality algorithms cannot
- There are however some challenges with t-SNE
  - Flexibility makes it harder to interpret
- Its important to study how t-SNE behaves on simple cases to develop an intuition on more complex examples
- I look forward to use t-SNE for Dimensionality Reduction in the future with Music Generation visualization and Machine Learning

Other Sources

- SKLearn Documentation
  - http://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html
- T-SNE Playground
  - https://distill.pub/2016/misread-tsne/
- Simple explanation of T-Sne
  - https://www.youtube.com/watch?v=NEaUSP4YerM&t=424s



sklearn.manifold.TSNE

class `sklearn.manifold.` **`TSNE`** (*n_components=2, perplexity=30.0, early_exaggeration=12.0, learning_rate=200.0, n_iter=1000, n_iter_without_progress=300, min_grad_norm=1e-07, metric='euclidean', init='random', verbose=0, random_state=None, method='barnes_hut', angle=0.5*) [source]