Dimensionality Reduction t-SNE

t-distributed Stochastic Neighbour Embedding

Review Unsupervised Learning

Discuss how this can be applied to reduce the number of columns in the dataset using PCA

Understanding Manifolds

A manifold can be an underlying structure
Locally Euclidean

An analogy:

Earth in maps = 2D
Actual Earth = 3D

High Dimensional data can be visualized as resting on a curved surface inside a space with even more dimensions. The goal of manifold learning is to discover a lower- dimensional space that still faithfully conveys the critical structures and relationships inherent in the data.

Manifold Learning

Powerful unsupervised learning approach with machine learning, used for dimensionality reduction.
Works on the theory that High-Dim-Data often resides on a lower-dim manifold that is embedded within the higher-dim space

Why?
The objective of manifold learning is to reveal this lower-dimensional structure without relying on pre-established classifications.

- **Isomap**: sklearn.manifold.Isomap
- **Locally Linear Embedding (LLE)**: sklearn.manifold.LocallyLinearEmbedding
- **t-Distributed Stochastic Neighbor Embedding (t-SNE)**: sklearn.manifold.TSNE
- **Spectral Embedding**: sklearn.manifold.SpectralEmbedding
- **Multidimensional Scaling (MDS)**: sklearn.manifold.MDS

These are some examples of Manifold learning but we will focus on t-SNE

Simplified overview of t-SNE

t-Distributed Stochastic Neighbour Embedding (t-SNE) is a powerful method for visualizing complex, high-dimensional data in a space of fewer dimensions. It's particularly effective for exploring and analyzing data, as it retains the local structure and reveals patterns and clusters.

1. **Calculating Pairwise Similarities**: Starts with figuring out how similar each data point is to every other point, using euclidean distance to measure. closer = high prob of being neighbours
2. **Dimensional Mapping**: Finds new lower dim space where the probabilities are preserved as much as possible. Puts similar points together and dissimilar points apart which keeps the data's structure.
3. **Utilizing Probability Distributions**: In the high-dimensional space, it uses a Gaussian distribution, while in the lower-dimensional space, it opts for a t-distribution, which helps prevent different points from overlapping too much.
4. **Optimizing Representation**: The point of t-SNE is minimizing the difference between the two probability distributions, ensuring that the low-dimensional representation reflects the high-dimensional relationships accurately.

Components of t-SNE

• **n_components**: This parameter specifies the dimension of the embedded space. By default, it is set to 2, meaning that t-SNE will create a 2D representation of the data. • **perplexity**: Perplexity controls the balance between preserving local and global structures. It determines the number of nearest neighbours used in the algorithm. Larger datasets usually require a larger perplexity. A value between 5 and 50 is recommended. Different values can significantly affect the results. We will discuss this more. • **early_exaggeration**: This parameter controls the spacing between natural clusters in the original space. Larger values increase the space between clusters in the embedded space. • **learning_rate**: The learning rate influences how the data points move during optimization. It is usually in the range [10.0,

1000.0]. If too high, data points may form a 'ball'; if too low, points may compress into a dense cloud. The 'auto' option sets the learning rate based on the sample size. • **n_iter**: The maximum number of iterations for optimization. The algorithm iteratively refines the embedding. • **metric**: The distance metric used to compute pairwise distances between samples. By default, it's 'euclidean'. • **init**: The initialization method for the embedding. 'pca' is commonly used for dense data. • **random_state**: Seed for random number generation.

Perplexity

Balance between local and global data structures during the embedding process

- It approximates the number of effective nearest neighbors, guiding the density of the local manifold of each data point.
- Essentially, perplexity is a measure of the expected density around a point and impacts how t-SNE calculates probabilities for nearby points.

Higher Perplexity

2. **Impact on t-SNE Visualizations**:
   - **With Higher Perplexity**:
     - An increase in perplexity generally allows t-SNE to capture broader data trends, potentially enhancing the visibility of global structures.
     - Clusters may appear more separated, aiding in the discernment of distinct data groupings.
     - However, this can sometimes lead to an exaggeration of space between clusters, which might not accurately reflect the true data distances.
     - High perplexity values can be particularly useful for datasets with intricate structures, like nested clusters.

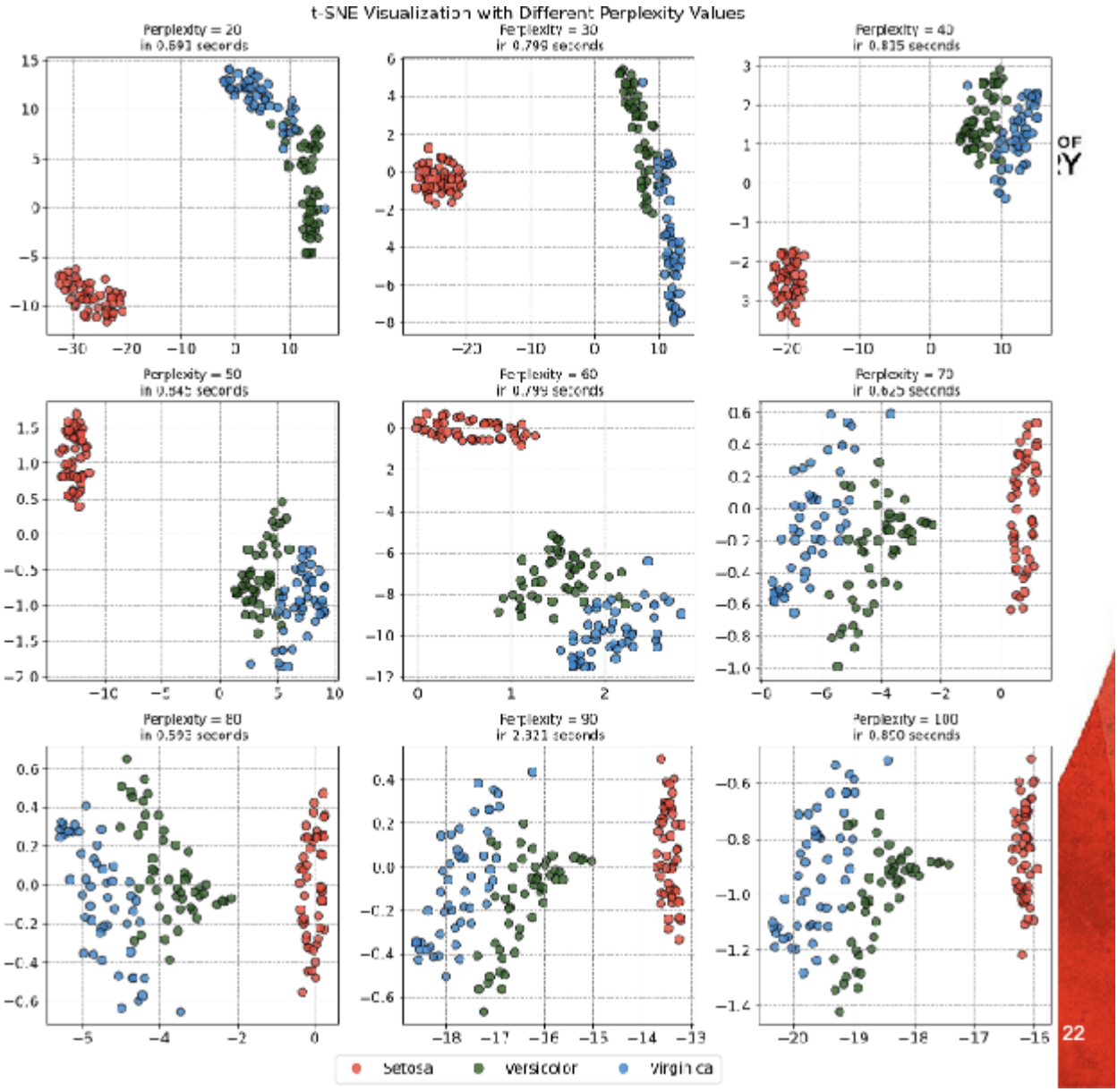
Lower Perplexity

2. **Impact on t-SNE Visualizations**:
   - **With Lower Perplexity**:
     - A lower perplexity value emphasizes the preservation of local data intricacies, often resulting in denser clusters.
     - This can be beneficial for highlighting fine-grained patterns but may merge or obscure broader data relationships.
     - Overlapping or densely packed clusters are common when the perplexity is set too low, which might complicate the interpretation of the embedding.

Choosing the right Perplexity

3. **Selecting an Appropriate Perplexity**:
   - The optimal perplexity setting is dependent on the dataset's characteristics and the specific insights you're seeking.
   - While the typical range for perplexity lies between 5 and 50, the best value is often found through iterative experimentation.
   - Adjusting perplexity in conjunction with other t-SNE parameters, such as the learning rate and the number of iterations, can further refine the resulting visualization.

Remember, t-SNE is a non-linear dimensionality reduction technique, and its results can vary **significantly with different hyperparameter settings**. It's always recommended to try multiple perplexity values and compare the resulting embeddings to ensure the most informative representation of your data.



Here is the impact of perplexity in the data

Advantages of t-SNE

1. **Preserves Local Structure**: Keeps local structure and data intact. High-dim neighbours will remain neighbours in low-dim 2. **Captures Non-Linear Relationships**: t-SNE can capture complex non-linear relationships between features. This makes it suitable for datasets where the relationship between variables is not linear. 3. **Robust to Outliers**: t-SNE is less sensitive to outliers than many other dimensionality reduction techniques.

Disadvantages of t-SNE

1. **Computationally Intensive**: Requires significant computational resources especially for large Datasets. 2. **Not Ideal for Preprocessing**: t-SNE not typically used for preprocessing data for predictive modeling. Does not maintain global structure of data which is VERY important for predictive models 3. **Sensitive to Initial Conditions**: results from t-SNE can vary from the initial conditions and random seed used. Different runs of the algo can give varied results and produce diff visuals which is annoying when you want consistent results