Visualizing Data using t-SNE

Team Number: 55 | Team Bash Party

- Keshav Bajaj (2019115010)
- Tejasvi Chebrolu (2019114005)
- Naman Ahuja (2019101042)
- B Vaibhaw Kumar (2019112021)

Problem Statement

Visualization of high-dimensional data is an important problem in many different domains, and deals with data of widely varying dimensionality. Cell nuclei that are relevant to breast cancer, for example, are described by approximately 30 variables (Street et al., 1993), whereas the pixel intensity vectors used to represent images or the word-count vectors used to represent documents typically have thousands of dimensions.

Dimensionality Reduction methods convert the high-dimensional data set $X = \{x_1, x_2, ..., x_n\}$ into two or three-dimensional data $Y = \{y_1, y_2, ..., y_n\}$ that can be displayed in a scatterplot. 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. Various techniques for this problem have been proposed that differ in the type of structure they preserve.

We plan on implementing **t-SNE**, a way of converting a high-dimensional data set into a matrix of pairwise similarities, for visualising the resulting similarity data. **t-SNE** is capable of capturing much of the local structure of the high-dimensional data very well, while also revealing global structure such as the presence of clusters at several scales. It is also **non-linear** which means that it can separate data that cannot be separated by any straight line.

Goals and Approach

Goals

▼ Compare our implementation of **t-SNE** to the implementation given in the paper.

▼ See the differences in the scatter plots generated.

▼ Compare our implementation of **t-SNE** to Sammon Mapping, Isomap, and LLE on the same dataset.

▼ We will use existing implementations for the other techniques.

▼ See the differences in the scatter plots generated.

▼ Compare our implementation of regular **t-SNE** to our implementation of modified **t-SNE**.

▼ Compare the performances of the implementations with respect to memory, time, space, implementation ease, etc.

▼ See the differences in the scatter plots generated.

Conclusion

By the end of the project, we want to have a quantitative and qualitative understanding of the pros and cons of **t-SNE**. When it would be a good idea to use it, and when it would not be a good idea. We would have a working implementation of:

- **t-SNE:**
  - Code
  - Scatterplots for different datasets
- **Modified t-SNE:**
  - Code
  - Scatterplots for different datasets

Approach

▼ **Introduction**

▼ Algorithm

Algorithm 1: Simple version of t-Distributed Stochastic Neighbor Embedding.

Data: data set $\mathcal{X} = \{x_1, x_2, ..., x_n\}$,
cost function parameters: perplexity $Perp$,
optimization parameters: number of iterations T, learning rate η, momentum $\alpha(t)$.
Result: low-dimensional data representation $\mathcal{Y}^{(T)} = \{y_1, y_2, ..., y_n\}$.
begin
  compute pairwise affinities $p_{j|i}$ with perplexity $Perp$ (using Equation 1)
  set $p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}$
  sample initial solution $\mathcal{Y}^{(0)} = \{y_1, y_2, ..., y_n\}$ from $\mathcal{N}(0, 10^{-4}I)$
  for t=1 **to** T **do**
    compute low-dimensional affinities q_{ij} (using Equation 4)
    compute gradient $\frac{\delta C}{\delta \mathcal{Y}}$ (using Equation 5)
    set $\mathcal{Y}^{(t)} = \mathcal{Y}^{(t-1)} + \eta \frac{\delta C}{\delta \mathcal{Y}} + \alpha(t)\left(\mathcal{Y}^{(t-1)} - \mathcal{Y}^{(t-2)}\right)$
  end
end

- **Datasets**
  - MNIST
  - Olivetti
- **PCA**
  - Run PCA on all the datasets to reduce the dimensionality to 30.
- **Run Other DRAs**
  - Run the other DRAs on the given datasets.
- **Implement t-SNE**
  - Implement the regular **t-SNE** and the modified **t-SNE** and note their performance.
- **Compare Scatter Plots**
  - Manually look at the scatter plots and see which methods performed better.
- **Future Work**
  - Look at approaches for using **t-SNE** for large datastes.
  - Look at methods to improve the time-complexity of the algorithm.

Milestones and Timeline

Timeline

Timeline	Milestones
@November 1, 2021 → November 7, 2021	Literautre Review
@November 7, 2021	Project Proposal Submission
@November 7, 2021 → November 10, 2021	Creation of pipeline and preparing dataset
@November 11, 2021 → November 12, 2021	Run PCA on Datasets
@November 13, 2021	Running other DRAs
@November 14, 2021 → November 16, 2021	Convert Euclidean distances into conditional probabilities that represent similarity
@November 17, 2021 → November 20, 2021	Mid Project Evaluations
@November 21, 2021 → November 27, 2021	Implementing improvements and suggestions based on mid evals
@November 28, 2021 → November 29, 2021	Testing
@November 30, 2021	Write Report and make presentation
@December 1, 2021 → December 4, 2021	Final Presentation
@December 4, 2021	Final Report submission

References:

1. van der Maaten, Laurens & Hinton, Geoffrey. (2008). Viualizing data using t-SNE. Journal of Machine Learning Research. 9. 2579-2605.
2. *Statquest: T-sne, clearly explained*. (n.d.). Retrieved 7 November 2021, from https://www.youtube.com/watch?v=NEaUSP4YerM
3. Erdem (burnpiro), K. (2020, April 22). *T-sne clearly explained*. Medium. https://towardsdatascience.com/t-sne-clearly-explained-d84c537f53a