Improving Sampling-Based t-SNE Performance Using Dijkstra’s Algorithm for Approximate Distance Computation

Introduction

In today’s world, high-dimensional data is very common. Also, it is hard to interpret for humans. So, proper visualisation is crucial to help people make sense of it. That’s where **t-SNE** comes in:

T-distributed Stochastic Neighbour Embedding is:

- An algorithm for visualising high-dimensional data to lower (usually 2 or 3) dimensions
- Captures non-linear relationships
- Detect patterns and clusters in the data
- Perplexity is the main parameter; it controls global vs local representation

Sampling-based t-SNE is:

- t-SNE ran on a sample of the data
- Useful for fine-tuning t-SNE (adjust parameters on the sample, then extrapolate to the full dataset) [1]

The neighbourhood graph between samples is reconstructed on every run, which is **slow and costly**

Research Question

Can Dijkstra's algorithm be used to improve the speed of sample-based t-SNE?

Methodology

Approximate the distances by traversing the full neighbourhood graph. We run Dijkstra’s algorithm from every sample point until we reach k=3*perplexity sample points.

Alternatively, we apply the Dijkstra algorithm on the full affinity matrix, to directly obtain the sample affinity matrix.

Furthermore, we use a modified version of Dijkstra:

- Optimised to work on sparse graphs
- Stops after reaching k sample points

Results

Quantitative analysis:

Comparing affinity matrices from the three methods

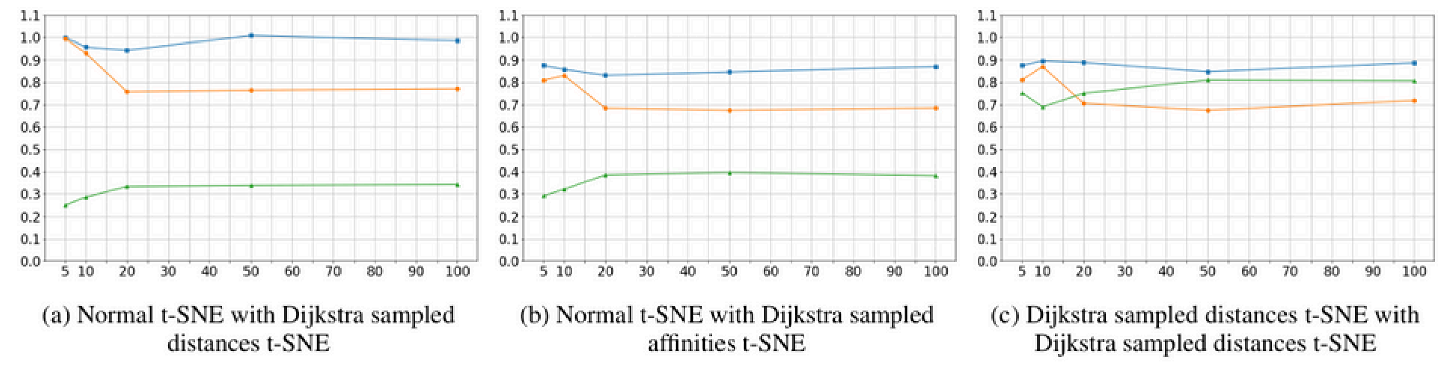
(a) Normal t-SNE with Dijkstra sampled distances t-SNE (b) Normal t-SNE with Dijkstra sampled affinities t-SNE (c) Dijkstra sampled distances t-SNE with Dijkstra sampled distances t-SNE

Figure 1: Line graphs comparing the three methods based on the affinity matrices for different perplexity values for the MNIST dataset. On the Y axis, the three different measures are visualised. On the X axis lie the different perplexity values. The blue line with square markers represents the Ratio of Neighbour Count. The orange line with circle markers represents the Ratio of Shared Indices. The green line with triangle markers represents the Similarity.

Comparing the runtime of the algorithms

Method	Setup	Perplexity				
		5	10	20	50	100
Normal	-	137.1	118.0	94.0	69.9	52.8
Dijkstra on Distances	19.5	**80.4**	**71.7**	**53.4**	87.9	140.9
Dijkstra on Affinities	39.2	**82.3**	**70.6**	73.2	92.7	151.4

Table 1: Total execution time per method per perplexity in seconds for the MNIST dataset. Averaged over 5 runs, calculated using $k = 300$. We observe that our methods execute faster for lower perplexities.

Qualitative analysis:

Comparing the visualisations of the algorithms

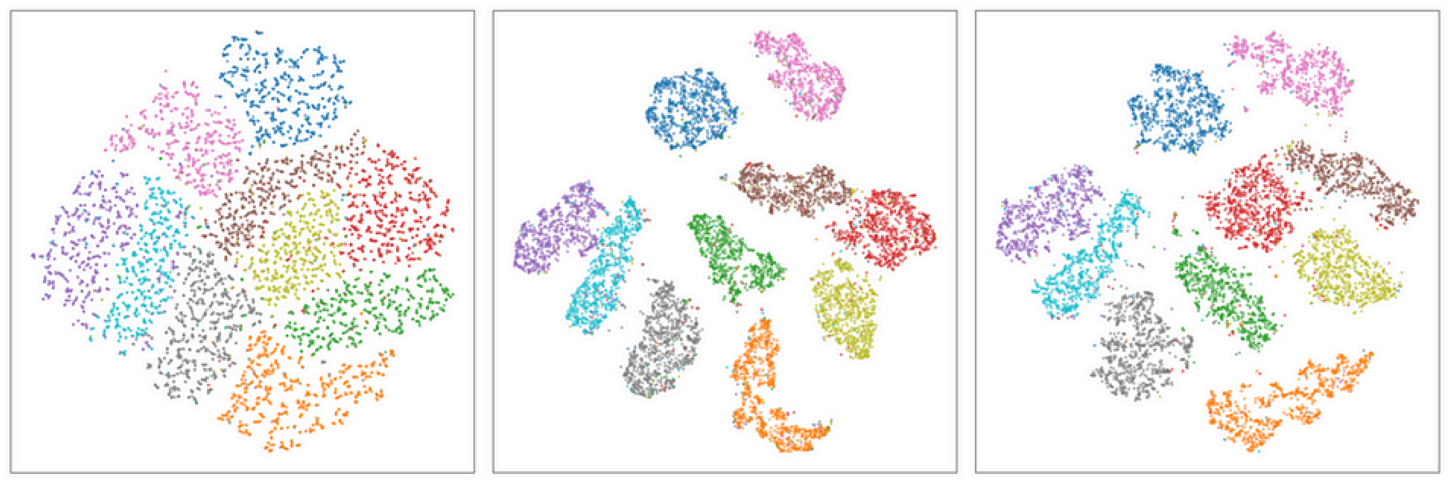
(a) Visualisation using normal t-SNE (b) Visualisation using sampled distances with Dijkstra t-SNE (c) Visualisation using sampled affinities with Dijkstra t-SNE

Figure 5: Comparison of visualisations of the three different methods analysed in our paper for the MNIST dataset. We can see that (b) and (c) have more distinct clusters than (a), but maintain the quality of the visualisations.

Author: Filip Markov (f.i.markov-1@student.tudelft.nl)
Supervisor: Martin Skrodzki
Responsible professor: Klaus Hildebrandt

Discussion

Our results show:

- **Affinity matrix comparison**
  - Our methods perform somewhat similarly
  - Dijkstra on distances is more similar to the normal implementation
- **Runtime comparison**
  - Both of our implementations perform up to 40% faster for lower perplexity values
  - Dijkstra on distances performs better than Dijkstra on affinities overall
- **Visualisation comparison**
  - Our methods cluster the data more
  - Dijkstra on affinities leaves more points between clusters

Conclusion

Our approach:

- Lowers runtime for low perplexity values
- Still produces useful visualisations, even if they are different

But:

- It is slower for higher perplexity values
- Visualisations differ from standard t-SNE

Future Work

- Further testing
  - Datasets
  - Parameters
- Use similar graph traversal algorithms, such as A* or Bellman-Ford
- Attempt to create better visualisations instead of improving runtime

[1] Martin Skrodzki, Nicolas F. Chaves-de Plaza, Thomas Höllt, Elmar Eisemann, and Klaus Hildebrandt. Navigating Perplexity: A linear relationship with the data set size in t-SNE embeddings, December 2024. arXiv:2308.15513 [cs].