Sample-Based t-SNE Embeddings

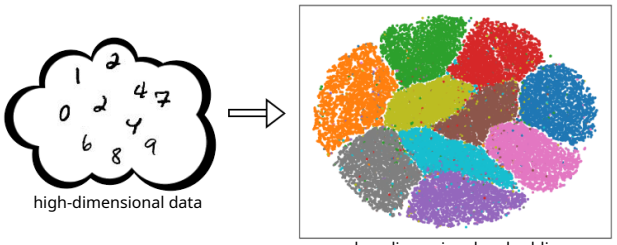
How different sampling strategies influence the quality of low-dimensional embeddings

TU Delft

Background

Visualising data is vital to understand it, identify patterns, and for presenting results.

But how do we do that in higher dimensions?



The solution is using dimensionality reduction techniques like t-SNE: an algorithm that embedds high-dimensional data by giving both the high and low dimensional points a distribution, which are then updated to be as similar to each other as possible.

While t-SNE works very well for data visualisation, it is also rather slow, with a runtime complexity of $O(n^2)$ per iteration.

There has been some research into algorithmic improvements to make it faster, but large datasets remain a challenge.

What if we instead sample to reduce the dataset size.

Research Question

How does the used sampling technique influence the resulting embedding of a high-dimensional dataset using t-SNE?

- What different sampling techniques are there?
- What is the effect of using them and applying t-SNE on the obtained sample?
- How do the resulting embeddings compare?
- How does embedding a sample change the runtime of the algorithm?

Methodology

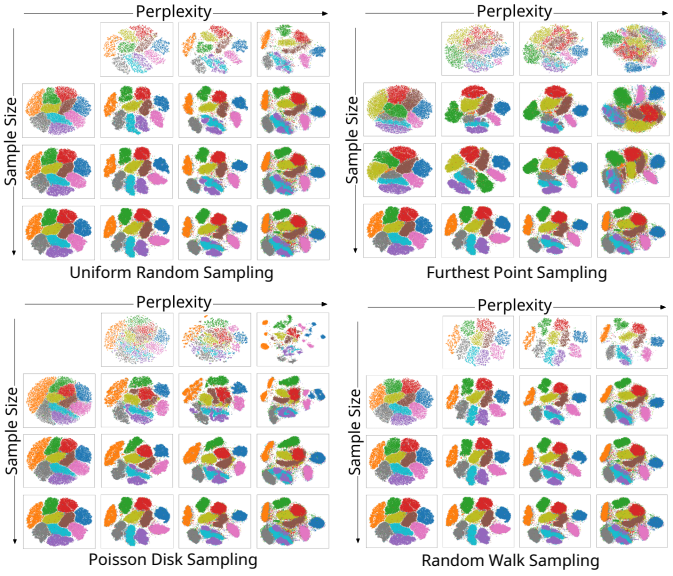
Compute t-SNE embeddings on sampled data with different sampling rates and perplexity values.

Sampling algorithms to look at:

- Uniform Random Sampling
- Furthest Point Sampling (FPS)
- Poisson Disk Sampling (PDS)
- Random Walk Sampling (RWS)

Visual Results

How good do the embeddings produced given different sampling rates and perplexity values look?



Uniform random seems to produce quite consistent results.

FPS seems quite inconsistent, and loses an entire cluster.

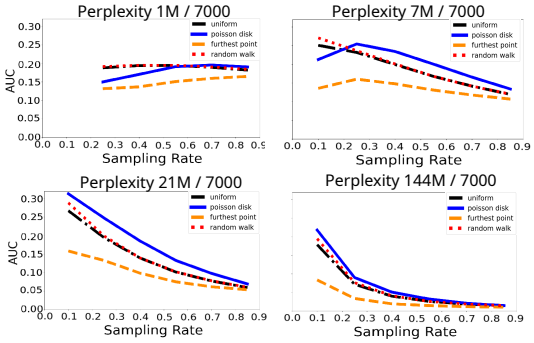
PDS seems quite consistent again, but has more merging and (useful) splitting of clusters.

RWS seems very similar to uniform random sampling.

Numerical Results

How good are the embeddings numerically?

Area under the precision recall curve (AUC): measure of how well high-dimensional neighbours are preserved in low-dimensional embedding, where 1 is perfect value.



Poisson disk sampling seems to perform the best.

Runtime of Approach

How long does it take to actually sample and embed (sampled) data?

Sample Size \ Perplexity			
	40.2949	40.0240	59.6335
55.3980	80.8241	144.3457	762.6433
75.8105	165.1294	384.0978	2378.9994
103.8373	307.4926	796.8412	5102.6103

Time (in seconds) to embed data

- Uniform: 0.004 seconds
- FPS: 90 samples / second
- PDS: 8,000 samples / second
- RWS: 50 minutes

So sampling is clearly helpful.

Uniform random sampling or PDS take little time to sample, making use of this speed up.

Conclusion and Future Work

Uniform random sampling seems to produce consistent results that are representative of the full dataset.

Poisson disk sampling seems to maintain high-dimensional neighbours very well through sub-clustering of samples.

Future work suggestion: Experiment on more datasets, and with more sampling methods to strengthen the findings.

Em Ketterer (e.l.ketterer@student.tudelft.nl)

Research Project CSE3000

Supervisors: Klaus Hildebrandt, Martin Skrodzki