ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

Estimated neighbour sets and smoothed sampled global interactions are sufficient for a fast approximate t-SNE.

Pierre Lambert[1], Edouard Couplet[1], Cyril de Bodt[1], and John A. Lee[1 2] *

1- Université catholique de Louvain - ICTEAM/ELEN
Place du Levant 3 L5.03.02, 1348 Louvain-la-Neuve - Belgium

2- Université catholique de Louvain - IREC/MIRO
Avenue Hippocrate 55 B1.54.07, 1200 Brussels - Belgium

Abstract. To minimise its loss function, the popular method of nonlinear dimensionality reduction t-SNE requires $\mathcal{O}(N^2)$ computations. As its applications often involve large datasets, fast approximations have been developed, such as Barnes-Hut t-SNE and FIt-SNE. Most fast approximations to t-SNE require the embedding dimensionality to be small, typically 2 or 3, limiting the use of t-SNE to data visualisation. Additionally, the effective computation time of the current accelerated t-SNE algorithms stays too high for a comfortable interactive visual exploration of data. This paper proposes an accelerated approximation to t-SNE with iterations of complexity $\mathcal{O}(NK)$, which does not rely on the use of a model to capture information about the low-dimensional space, relieving the computational burden of high dimensionality of the embedding space. For this purpose, the proposed method approximates neighbour sets and keeps track of smoothed estimations of long-range interactions in $\mathcal{O}(NK)$ time. The method is qualitatively tested on a handful of datasets and shows comparable results to existing fast neighbour embedding methods in the context of data visualisation. Code is available at `https://github.com/PierreLambert3/c_fast_hSNE.git`.

1 Introduction

Dimensionality reduction (DR) with t-SNE [1] has raised much interest and is now widely used in many application domains, noticeably in computational biology [2]. Part of this popularity stems from the availability of fast approximations of t-SNE. The original t-SNE scales in $\mathcal{O}(N^2)$ per iteration, due to considering all pairwise interactions. In Barnes-Hut t-SNE [3], space-partitioning trees are used, namely, vantage-point trees to identify the K nearest neighbours in the high-dimensional(HD) data space, which are involved in short-range attractive forces, whereas Barnes-Hut quad-trees are used in the low-dimensional(LD) embedding space to aggregate and approximate long-range repulsive forces. Embedding data points is then inspired by methods that where originally developed to solve N-body problems in physics and astronomy, and thus to produce force-directed layouts in mechanics. Such binary or quaternary trees allows the computational

*PL is a FRIA grantee of the Fonds de la Recherche Scientifique - FNRS. JAL is a Research Director with the F.R.S.-FNRS.

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

complexity to reduce to $\mathcal{O}(N \log_2 N)$. More recent methods include FIt-SNE [4], LargeVis [5], and UMAP [6], where the complexity can further drop down to $\mathcal{O}(N)$ per iteration, using fast multi-pole approximations or 'negative sampling' of points to estimate the repulsive forces. Recent works [7, 8] have also investigated modifying the weight of the tail in the LD neighbor distribution, which impacts the shape of the mismatch with the Gaussians in HD, yielding in turn a spectrum of embeddings where clusters might be more or less separated.

This paper proposes a fast approximation to t-SNE ($\mathcal{O}(N)$) which does not rely on modeling the LD space with data structures. This removes the usual restrictions on the dimensionality of the LD embedding space, potentially opening t-SNE to new applications. In addition, the method learns the pairwise high dimensional relationships iteratively, allowing instantaneous visual feedback to the user when changing a hyperparameter or launching the method.

The rest of this paper is organized as follows. Section 2 goes through a brief reminder of t-SNE and its variable tail weight variants, sometimes coined ht-SNE, where 'h' refers to heavy or heavier tails. Section 3 describes the proposed method, which implements a fast ht-SNE without any specific data structure or model of the LD space, apart from a list of neighbours. Section 4 reports some results on various data sets. Section 5 concludes this paper and sketches perspectives for future work.

2 Variable tail weight in t-SNE

Neighbour embedding (NE) gathers methods of DR like stochastic neighbor embedding (SNE) [9], t-distributed SNE (t-SNE) [1], and UMAP [6]. These perform DR by preserving soft neighborhoods with pairwise similarities. In SNE, Gaussian kernels are used to smoothly (and derivably) model neighborhoods in both HD and LD space. The minimisation of the Kullback-Leibler divergences between the HD and LD distributions yields the low dimensional embedding. To enhance the separation of points into clusters, t-SNE forces a mismatch between LD and HD similarities by modeling the LD relationships with a Student t-distribution. Formally, let $\mathbf{X} = [\mathbf{x}_i]_{1 \leq i \leq N}$ and $\mathbf{Y} = [\mathbf{y}_i]_{1 \leq i \leq N}$ denote the data coordinates in the HD and LD spaces, respectively. Then, the symmetric pairwise affinities p_{ij} are

$$p_{j|i} = \frac{\exp -\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma_i^2}{\sum_{k \neq i} \exp(-\|\mathbf{x}_i - \mathbf{x}_k\|^2 / 2\sigma_i^2)}\,, \quad p_{i|i} = 0\,, \quad \text{and } p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N}\ , \quad (1)$$

where the radii σ_i are adjusted to comply with a user-set perplexity. In the LD space, the similarities are defined

$$q_{ij} = \frac{(1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2)^{-1}}{\sum_{k \neq l} (1 + \|\mathbf{y}_k - \mathbf{y}_l\|^2)^{-1}}\,, \quad q_{ii} = 0\ . \quad (2)$$

The joint KL divergence $L = \mathrm{KL}(P||Q) = \sum_{i \neq j} p_{ij} \log(p_{ij}/q_{ij})$ is here the loss to be minimised using gradient descent with momentum. The gradient of the

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

KL divergence is $\frac{\partial L}{\partial \mathbf{y}_i} = 4\sum_{j\neq i}(p_{ij}-q_{ij})(1+\|\mathbf{y}_i-\mathbf{y}_j\|)^{-1}(\mathbf{y}_i-\mathbf{y}_j)$, where p_{ij} and q_{ij} are responsible for attractive and repulsive forces between $\mathbf{y}_i$ and $\mathbf{y}_j$, respectively. Extended versions of t-SNE exists, like HSSNE [8], where the LD kernel can have tails of varying weights, revealing different views on the data. Here, a (non-integer) power α parameterizes the LD kernel, like in [7], instead of degrees of freedom. This gives

$$w_{ij} = \left(1+\frac{\|\mathbf{y}_i-\mathbf{y}_j\|^2}{\alpha}\right)^{-\alpha}, \quad q_{ij} = \frac{w_{ij}}{\sum_{k\neq l} w_{kl}}, \quad q_{ii} = 0 \ . \tag{3}$$

The gradient to ht-SNE becomes $\frac{\partial L}{\partial \mathbf{y}_i} = 4\sum_j (p_{ij}-q_{ij}) w_{ij}^{1/\alpha}(\mathbf{y}_i-\mathbf{y}_j)$.

3 The proposed method: ht-SNE

Typically, fast t-SNE accelerations first compute the HD similarities P and then start iteratively updating the LD coordinates for a set duration. In data visualisation, this forces the user to wait after each action before receiving a visual feedback from the iterative part of the optimisation. Ideally, one could imagine a version of (h)t-SNE without a set number of iterations, taking place within an interactive software environment where user-driven inputs take effect immediately and seamlessly on screen. This would allow for a more insightful exploration of the data by dynamically visualising the effects of hyperparameter changes, such as the LD similarity kernel, perplexity, learning rate, or distance metric. The proposed method allows such a setup by intertwining a neighbour discovery phase and an LD coordinate optimisation phase during optimisation, both at $\mathcal{O}(NK)$ cost.

For the ith point number, let the approximate twin sets of HD and LD neighbours be denoted $\hat{\mathfrak{N}}_i^{HD}$ and $\hat{\mathfrak{N}}_i^{LD}$. The size of $\hat{\mathfrak{N}}_i^{HD}$ is set to 3 times the perplexity, and $\hat{\mathfrak{N}}_i^{LD}$ to a small value between 5 and 30. During neighbour discovery, each point refines both sets by looking at candidate points. Exploration is done by uniformly sampling candidates across the dataset. Exploitation is ensured by generating candidate points from neighbours of neighbours as in [5], and from points from the twin set: candidate indices for $\hat{\mathfrak{N}}_i^{LD}$ are sampled from $\hat{\mathfrak{N}}_i^{HD}$ and reciprocally. This heuristic closes a convenient positive-feedback loop between the minimisation of the cost function and the quality of the neighbour estimations. Better neighbour approximations mean more accurate gradients, and therefore a better embedding. A better embedding means that when generating candidates for $\hat{\mathfrak{N}}_i^{LD}$ by looking at neighbours of neighbours in LD, the hit rate gets higher. Conversely, from LD to HD, a more accurate $\hat{\mathfrak{N}}_i^{LD}$ means better candidates for $\hat{\mathfrak{N}}_i^{HD}$ when looking at $\hat{\mathfrak{N}}_i^{LD}$. The two-way feedback between embedding and estimated neighbours allows the method to 'recycle' some of the computations carried out by t-SNE to further refine $\hat{\mathfrak{N}}_i^{LD}$ directly and $\hat{\mathfrak{N}}_i^{HD}$ indirectly. Since the embedding changes from iteration to iteration, $\hat{\mathfrak{N}}_i^{LD}$ needs to be refined at each iteration. On the other hand, the HD positions remain static throughout the process, it therefore becomes inefficient to pursue refinement of the HD sets at each iteration. For this reason, $\hat{\mathfrak{N}}_i^{HD}$ has a probability of being updated equal to the exponential average through time of the percentage of points that were updated in the HD neighbour estimation recently, this value cannot be lower than 0.02 to prevent missing too many opportunities without impacting the effective performance.

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

At the heart of any version of t-SNE are the matrices P and Q. Here, the HD similarities $\hat{P}$ are sparse approximations similar to other accelerated versions of t-SNE [3, 4], they are recomputed with a probability corresponding to the smoothed percentage of changes in HD neighbours as described above. Whenever $\hat{\mathfrak{N}}_i^{HD}$ is updated, $\sigma_\mathbf{i}$ is updated to comply with the desired perplexity.

To approximate q_{ij}, it is necessary to estimate the value of $\sum_{k\neq l} w_{kl}$. Here, an exponential average through time is used to represent the sum. To do so, an accumulator is initialised at 0 at each iteration, whenever a value for q_{ij} for two points i and j has to be computed, the accumulator is increased by the value. At the end of each iteration, the accumulator is scaled to simulate N∗N contributions, it is then used to update the value of the average through time.

The gradient of ht-SNE can be rewritten around this division in neighbour sets. Let $\frac{\partial L}{\partial \mathbf{y}_{i|j}} = 4(p_{ij} - q_{ij}) w_{ij}^{1/\alpha}(\mathbf{y}_i - \mathbf{y}_j)$ be the gradient on $\mathbf{y}_i$ resulting from interacting with the point number j, the ht-SNE gradient can be written $\frac{\partial L}{\partial \mathbf{y}_i} = \sum_j \frac{\partial L}{\partial \mathbf{y}_{i|j}}$. The gradient for the parameters in $\mathbf{y}_i$ can be rewritten as

$$\frac{\partial L}{\partial \mathbf{y}_i} = \sum_{j \in \hat{\mathfrak{N}}_i^{LD}} \frac{\partial L}{\partial \mathbf{y}_{i|j}} + \sum_{\substack{j \in \hat{\mathfrak{N}}_i^{HD} \\ \& \; j \notin \hat{\mathfrak{N}}_i^{LD}}} \frac{\partial L}{\partial \mathbf{y}_{i|j}} + \sum_{\substack{j \notin \hat{\mathfrak{N}}_i^{LD} \\ \& \; j \notin \hat{\mathfrak{N}}_i^{HD}}} \frac{\partial L}{\partial \mathbf{y}_{i|j}} \quad . \tag{4}$$

Knowing $\hat{P}$ and the estimation of $\sum_{k\neq l} w_{kl}$, computing the first two elements of $\frac{\partial L}{\partial \mathbf{y}_i}$ is fast. Because of the heavy tail of $\hat{Q}$, relative to $\hat{P}$, the third element of $\frac{\partial L}{\partial \mathbf{y}_i}$ is dominated by the repulsive distant interactions between the ith point and the rest of the data set. These far-reaching interactions on i are approximated by sampling a number (here 40) of random indices j across the dataset.

This paper uses Nesterov's momentum [10], as it performs well for t-SNE [11] and can bring visually smoother transitions. Changing α can reveal different aspects of the data. However, this can cause shrinkage of over-expansion of certain zones of the embedding, making visual interpretation difficult. For this reason, the momenta are split into attractive forces and repulsive forces, the repulsive momenta are scaled by a parameter ϕ that can be modified accordingly. Moreover, when increasing the attraction/repulsion ratio through changes in ϕ, a similar effect to early exaggeration [3, 4, 2] can be obtained.

4 Results

The proposed method is tested on 3 datasets: the 60.10^3 observations of the MNIST train set; COIL-20, which consists of 1440 photographs of 20 rotated objects, drawing manifolds in the shape of rings; and Anuran calls, available on the UCI machine learning database.

In Fig. 1.A, MNIST (left) and COIL-20 (right) are reduced to 6 dimensions, the axes are grouped by 2 to produce 3 embeddings. The proposed method appears to produce clear clusters and to utilise the available degrees of freedom to organise the points together. For instance, on COIL-20, some rings that tend to be cut when projecting in 2d instead organise as a pile of rings when seen through certain angles.

Panel B compares the proposed method with FIt-SNE on COIL-20, showing similar results. If using the algorithm in an interactive environment, data exploration is greatly enhanced when tweaking and experimenting with α and ϕ as well as when taking advantage of the information contained in the motion of points. Panel C attempts to

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

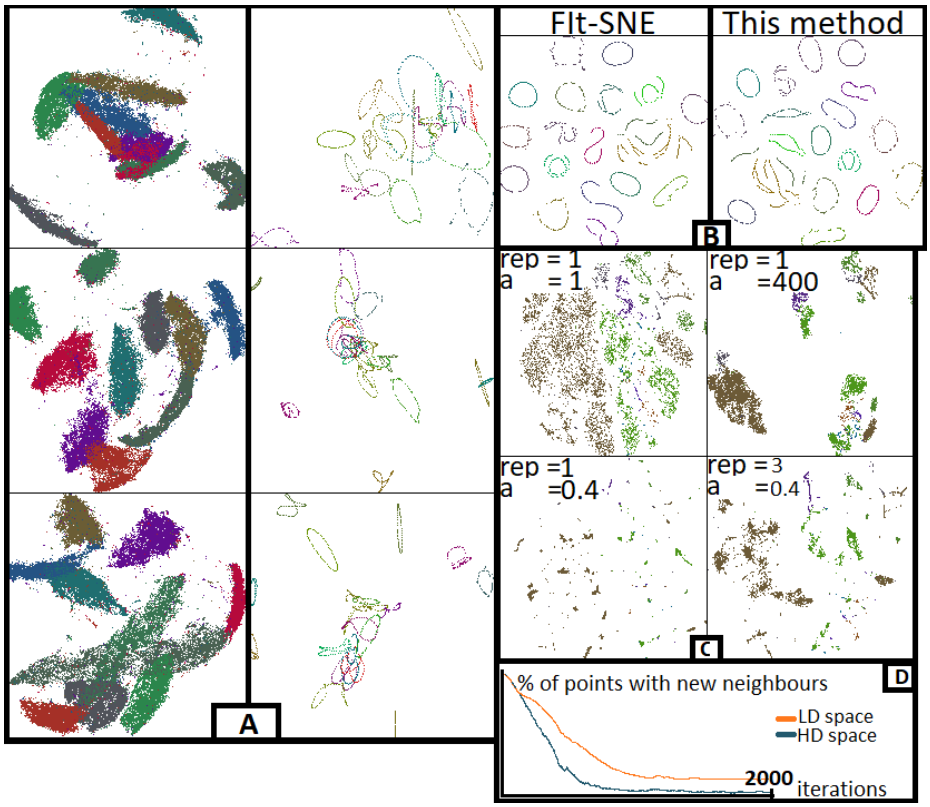

Fig. 1: **A**: Triptychs of 6-dimensional t-SNE embeddings of MNIST and COIL-20, the three embedding axes correspond to dimensions (1,2),(3,4), and (5,6). **B**: Qualitative comparison of the proposed method to FIt-SNE, on COIL-20. **C**: Example intended use of this algorithm, where varying α ("a") and ϕ ("rep") can reveal different aspects of the data, such as a global hierarchy using small distribution tails (high α) and a finer-grained cluster organisation using large distribution tails. **D**: Percentage of points whose LD or HD neighbour sets where updated, across 2000 iterations on MNIST.

demonstrate this by showing the effect of changes to α and ϕ on the embedding. With reasonable dataset sizes (less than roughly 20.10^3), transitions from one embedding to the other are seamless and take a few seconds on a CPU from 2021. Changing ϕ can be particularly useful to stretch out dense clusters when the LD tails are particularly heavy (bottom two embeddings). This panel illustrates that the method produces results that can vary greatly with the choice of hyperparameters, a tendency that needs to be addressed in most unsupervised methods. The proposed algorithm is designed with interactivity at the forefront: it is intended to be used in tandem with human input to modify various hyperparameters on the fly (learning rate, perplexity, ϕ, α, distance metric in HD, kernel in LD), and quickly assess the impact of the hyperparameter changes. The interactivity loop can be helpful in the task of visual data exploration, in particular when used in conjunction with quality assessment criteria and additional data analysis methods. However, when producing an embedding with more than three dimensions, a more thorough search of the hyperparameter space remains to be done in order to develop heuristics for the choice of hyperparameter values.

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

Panel D shows the evolution of the percentage of points whose sets $\hat{\mathfrak{N}}_i^{LD}$ and $\hat{\mathfrak{N}}_i^{HD}$ were updated, across the first 2000 iterations on MNIST with fixed hyperparameters. The change in LD neighbours decreases to a constant higher than 0, due to slight changes in the embedding that remain at convergence (these are barely noticeable visually when watching the embedding). This observation suggests that a decaying learning rate could be beneficial once the user is satisfied with the hyperparameter choices.

5 Conclusion

The proposed acceleration to (h)t-SNE runs at a competitive speed without restricting the dimensionality of the embedding space. Moreover, the method continuously updates and refines the HD relationships, allowing for a visual representation that can react quickly and seamlessly to user inputs. Further works include technical aspects such as a GPU implementation, and a more prospective path by studying the properties of the LD space when targeting an embedding dimensionality that exceeds the usual 2 or 3.

References

[1] L. van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[2] D. Kobak and P. Berens. The art of using t-sne for single-cell transcriptomics. *Nat Commun 10, 5416*, 2019.

[3] Laurens van der Maaten. Accelerating t-sne using tree-based algorithms. *Journal of Machine Learning Research*, 15(93):3221–3245, 2014.

[4] George C. Linderman, Manas Rachh, Jeremy G. Hoskins, Stefan Steinerberger, and Yuval Kluger. Fast interpolation-based t-sne for improved visualization of single-cell rna-seq data. *Nature methods*, 16:243 – 245, 2019.

[5] Jian Tang, Jingzhou Liu, Ming Zhang, and Qiaozhu Mei. Visualizing large-scale and high-dimensional data. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, page 287–297, Republic and Canton of Geneva, CHE, 2016. International World Wide Web Conferences Steering Committee.

[6] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction, 2020.

[7] Dmitry Kobak, George Linderman, Stefan Steinerberger, Yuval Kluger, and Philipp Berens. Heavy-tailed kernels reveal a finer cluster structure in t-sne visualisations. In Ulf Brefeld, Elisa Fromont, Andreas Hotho, Arno Knobbe, Marloes Maathuis, and Céline Robardet, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 124–139, Cham, 2020. Springer International Publishing.

[8] Zhirong Yang, Irwin King, Zenglin Xu, and Erkki Oja. Heavy-tailed symmetric stochastic neighbor embedding. In Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 22. Curran Associates, Inc., 2009.

[9] Geoffrey E Hinton and Sam Roweis. Stochastic neighbor embedding. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems*, volume 15. MIT Press, 2002.

[10] Yurii Nesterov. A method for solving the convex programming problem with convergence rate $o(1/k^2)$. *Proceedings of the USSR Academy of Sciences*, 269:543–547, 1983.

[11] Pierre Lambert, Cyril de Bodt, Michel Verleysen, and John A. Lee. Squadmds: A lean stochastic quartet mds improving global structure preservation in neighbor embedding like t-sne and umap. *Neurocomputing*, 503:17–27, 2022.