ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

Nesterov momentum and gradient normalization to improve t-SNE convergence and neighborhood preservation, without early exaggeration

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

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. Student t-distributed stochastic neighbor embedding (t-SNE) finds low-dimensional data representations allowing visual exploration of data sets. t-SNE minimises a cost function with a custom two-phase gradient descent. The first phase is called early exaggeration and involves a hyper-parameter whose value can be tricky and time-consuming to set. This paper proposes another way to optimise the cost function without early exaggeration. Empirical evaluation shows that the proposed method of optimization converges faster and yields competitive results in terms of neighborhood preservation.

1 Effective visualization of data with t-SNE

High-dimensional (HD) data is abundant in our digital world, with medical images, genomics, or financial transactions, to cite only a few examples. Most of the time, the data records lie in HD spaces and dependencies or relationships cannot be visualised directly by humans, turning intuitive understanding of HD data sets into a challenging task. Dimensionality reduction (DR) tackles the problem by providing their users with low-dimensional (LD) representations of the data, enabling visual exploration.

Representing HD data into a less expressive LD space requires concessions as information is usually lost in the process. Many families of DR algorithms exist, each one focusing on the preservation of information of different nature; some methods aim to preserve distances between points [1], others called neighbour embeddings (NE) [2] attempt to preserve the local neighbourhoods around the data points. Neighbour embedding is increasingly popular due to its capability to mitigate the concentration of norms and distances [3] and the availability of fast, approximate and yet accurate implementations [4, 5], making NE applicable to very large data sets [6, 7].

Algorithms that carry out NE, such as Student t-distributed stochastic neighborhood embedding (t-SNE) [8], usually minimise a cost function in a non-parametric way, by adjusting iteratively the coordinates of the data points in the

*PL is a FRIA grantee of the Fonds de la Recherche Scientifique - FNRS. JAL is a Research Director with the F.R.S.-FNRS. EC is supported by a FSR grant (UCLouvain). CdB is supported by Service Public de Wallonie Recherche under grant n°2010235-ARIAC by DIGITALWALLONIA4.AI.

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

LD space with gradient descent. This optimization process can be interpreted as a force-directed layout or mechanical system converging to equilibrium where attractive and repulsive forces are exerted on pairs of LD points.

In practice, t-SNE and its variants come with custom implementations of gradient descent, whose course is typically split in two successive phases, namely, *early exaggeration*, a relatively short preliminary phase where the attractive forces are deliberately amplified, followed by a longer second phase, where the gradients come back to their genuine value and the LD coordinates are fine-tuned with gradient and momentum. Intuitively, amplification of attractive forces in early exaggeration contributes to momentarily shrinking the clusters and increasing the inter-cluster gaps, to compensate for the otherwise local nature of NE. Even if default settings are provided, the user can specify the length and factor of force amplification in the early exaggeration phase [8, 9, 5]. This peculiar hyper-parameter, coined *early exaggeration* after the phase it is active in, was introduced in an effort to help the clusters move freely within the embedding during the early iterations, by concentrating the sources of repulsive forces into zones of tightly-packed points [8].

From the users' broader perspective, data exploratory analysis is an interactive and possibly iterative process where the users select different subsets of points, DR methods, and hyper-parameters in order to observe the data through multiple virtual lenses. However, deciding on a value for the early exaggeration amplification factor can be quite technical and hinders the overall visualisation process by diverting attention from the data to the optimiser. This paper hence proposes a new optimiser for t-SNE, which eliminates the need for early exaggeration. It is compared to the standard t-SNE optimiser on 10 data sets. The results show that the proposed optimiser achieves competitive neighbourhood preservation and faster convergence. The experiments also support the claim that the best values for early exaggeration can vary widely across different data sets, when using the standard optimiser.

Section 2 provides an overview of the t-SNE algorithm. Section 3 introduces the proposed optimisation method. Section 4 describes the experiments, reports their results, and discuss them.

2 t-SNE with early exaggeration

This section briefly overviews t-SNE and the role of *early exaggeration* in its optimisation.

Methods of NE such as t-SNE aim to preserve the neighbourhoods around each data point; t-SNE [8] defines pairwise similarities in the HD and LD space to model the neighbourhoods as smooth functions of the locations of the data points $\mathbf{X} = [\mathbf{x}_i]_{1 \leq i \leq N}$. Considering a data set of N points in the HD space, let δ_{ij} and d_{ij} be the pairwise distance between the i^{th} and j^{th} observations in the HD and LD spaces, respectively, for $i \in \mathcal{I} = \{1, ..., N\}$ and $j \in \mathcal{I} \setminus \{i\}$. The

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

pairwise similarities σ_{ij} and s_{ij} in both space are defined as

$$\sigma'_{ij} = \frac{\exp(-\pi_i \delta_{ij}^2/2)}{\sum_{k\in\mathcal{I}\setminus\{i\}} \exp(-\pi_i \delta_{ik}^2/2)}, \sigma_{ij} = \frac{\sigma'_{ij} + \sigma'_{ji}}{2N}, \;\; s_{ij} = \frac{(1+d_{ij}^2)^{-1}}{\sum_{k\in\mathcal{I},l\in\mathcal{I}\setminus\{k\}}(1+d_{kl}^2)^{-1}} \;\; . \tag{1}$$

The similarity between a point and itself is set to 0 in both spaces. The precision π_i modulates the similarities to capture information relevant to a user-defined scale. Embeddings are produced by minimising the mismatch between the HD and LD pairwise similarities. As the similarities are normalized, they can be interpreted as probabilities of the points to be the neighbor of one another. For this reason, t-SNE defines the mismatch as the Kullback-Leibler (KL) divergence $C_{t-\mathrm{SNE}}(\mathbf{X}) = \sum_{i\in\mathcal{I},j\in\mathcal{I}\setminus\{i\}} \sigma_{ij} \log(\sigma_{ij}/s_{ij})$. Minimisation is carried out with gradient descent.

Early exaggeration is an empirical tweak that consists in multiplying all HD similarities with a constant greater than one during the first iterations of the optimisation, typically one fifth or quarter of them. This changes temporarily the cost function and its gradient, with an increased tendency of similar HD points to group together in the LD embedding, because the similarities s_{ij} sum up to 1 while the σ_{ij} sum up to a larger value.

3 Proposed t-SNE optimizer, without early exaggeration

This section introduces the proposed optimisation method, which removes the need for early exaggeration during the first phases of optimisation.

Early exaggeration aims to help the clusters in formation move inside the embedding by accentuating the forces at hand and making the clusters smaller. Without early exaggeration, data points in movement are more likely to encounter large groups of dissimilar points and be subject to barriers or large fluctuations in the gradients. This paper proposes an alternative optimisation method without early exaggeration, where a strong momentum compensates for the occasional impeding forces, giving points enough inertia to pass the barrier of local gradient fluctuations. This work also normalises the gradients before updating the momenta.

The proposed method uses Nesterov's momentum [10], which makes (non-stochastic) gradient descent converge faster. If J denotes a cost function to be minimised, θ^t a parameter at iteration t, v^t its momentum, and $\nabla_\theta J(\theta)$ the gradient of J with respect to θ, Nesterov's momentum update the parameter θ^t following the pair of rules:

$$\begin{aligned} v^t &= \gamma \cdot v^{t-1} - \eta \nabla_\theta J(\theta^{t-1} + \gamma \cdot v^{t-1}) \;\; , \\ \theta^t &= \theta^{t-1} + v^t \;\; . \end{aligned}$$

The proposed optimiser uses a very strong momentum with $\gamma = 0.995$; the learning rate η is set to 1.

In order to keep a steady learning rate across data sets of varied sizes, the gradient $\nabla_{\mathbf{x}_i} C_{t-\mathrm{SNE}}$ for each parameter $\mathbf{x}_i$ is multiplied by $\frac{\sqrt{2N}}{100\|\nabla_{\mathbf{X}} C_{t-\mathrm{SNE}}\|_2}$ before being used in the momentum update, as a normalization. Operator $\|\cdot\|_2$

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

denotes the L_2 norm, while $\sqrt{2N}$ is the norm of a unit vector of the same dimensionality as $\nabla_{\mathbf{X}} C_{t-\mathrm{SNE}}$, and $1/100$ is empirical. This scaling term yields better experimental results and complies with the documented recommendation of increasing the learning rate of t-SNE with the data set size [7, 6].

4 Experiments, Results, and Discussion

The classical, legacy optimizer of t-SNE and the proposed optimiser without early exaggeration are here compared. The Barnes-Hut [4] acceleration of t-SNE is used in both cases. The methods were tested on 10 data sets of size N and dimensionality M [11]: airfoil self-noise ($N = 1502$, $M = 5$), abalone ($N = 4176$, $M = 8$), COIL-20 ($N = 1440$, $M = 1024$), Statlog landsat satellite ($N = 4434$, $M = 36$), Gaussian blobs ($N = 1000$, $M = 25$), forest cover type ($N = 3000$, $M = 54$), Californian housing data set ($N = 2500$, $M = 8$), gesture phase segmentation [12] ($N = 4000$, $M = 18$), plant ($N = 1000$, $M = 4$), Anuran ($N = 4000$, $M = 22$), and single-cell RNA-seq data from the adult mouse cortex ($N = 23822$, $M = 50$). Some of the data sets are random subsamples of larger data sets; the features in the RNA-seq data set are the first 50 principal components of selected scaled gene expression levels, as in [6]. The embeddings produced with and without early exaggeration are assessed quantitatively using the area under the curve (AUC) of the relative neighbourhood preservation R_{NX} indicator [13]; the closer the AUC gets to 1, the better the neighbourhood preservation.

The standard t-SNE optimisation relies on the default hyper-parameter values as specified in [9]; the perplexity is set to 30, the embeddings are initialised with PCA, and the optimisation runs for 1000 iterations. For each data set, regular t-SNE is run 12 times in total, as a grid search for the best combination of 2 learning rates and 6 early exaggeration factors. Early exaggeration factor e_e is picked in $\{1, 4, 8, 12, 16, 20\}$, while the learning rate is either 200 or $\max(\frac{N}{e_e}, 50)\}$. The former value 200 is a common default value for the learning rate, whereas $\max(\frac{N}{e_e}, 50)$ stems from works showing that greater learning rates are preferable on large data sets [6, 7, 9].

Concerning the proposed method, it relies on the same perplexity and number of iterations as the standard optimiser. It is also initialised with PCA whitening and thus the initial embedding is scaled to have unit standard deviation along both axes. The learning rate is set to 1 and there is no early exaggeration; no grid search is necessary here and therefore one run per data set suffices.

Table 1 displays the area under the curve (AUC) of the R_{NX} curve for both optimisers on each dataset, with the best scores highlighted in bold face. For the standard method, the AUC of the best run during the hyper-parameter grid searches is shown along with the corresponding early exaggeration factor. The results indicate that the proposed method performs competitively with the legacy optimiser on the considered data sets. Furthermore, this experiment emphasizes the time-consuming challenge of tuning the early exaggeration factor, as the best values vary widely across different data sets.

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

Data set	Proposed opt.	Legacy opt.	Early exag.
blobs	0.324	0.324	4
forest	**0.569**	0.555	20
plant	**0.694**	0.677	20
Anuran	**0.561**	0.535	1
COIL20	**0.671**	0.663	1
Abalone	0.594	**0.596**	4
airfoil	0.671	**0.688**	1
satellite	**0.577**	0.560	1
housing	**0.587**	0.573	1
RNAseq	**0.484**	0.466	4

Table 1: AUC of the R_{NX} curves for both optimizers across 10 data sets; the best value of early exaggeration for the standard method is reported.

Figure 1 shows the evolution of cost function $C_{t-\mathrm{SNE}}(\mathbf{X})$across the iterations for both optimizers. The blue curves correspond to the legacy optimizer, with early exaggerations of 4 and 1 in the leftmost and rightmost plots, respectively. The orange curves correspond to the proposed method. At the 1000^{th} iteration, the KL-divergence between the HD and LD similarities is seen to be lower with the proposed method than with the legacy optimiser for both data sets. Moreover, the proposed method seems to stabilize much faster close to its final arrangement.

The end of the early exaggeration is clearly visible at 250 iterations on the blue curve of the leftmost plot. A slight change of regime can be observed on the other plot as well, despite early exaggeration is set to 1 and hence turned off. This break in the curve is due to a change in the momentum parameter carried out in the standard implementation at the same time as the early exaggeration phase ends.

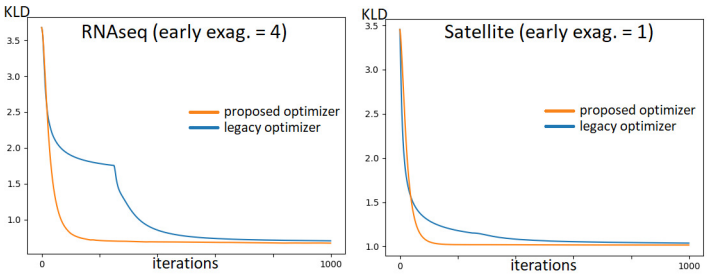

Fig. 1: Evolution of the t-SNE cost function across iterations.

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

5 Conclusions and outlook

This work proposes a new optimizer for the cost function of t-SNE, which frees the users from tediously searching an optimal early exaggeration factor. The legacy and proposed optimizers are compared, the latter producing competitive results across 10 data sets. The proposed optimisation method remains to be tested on very large data sets, which is the aim of further works.

References

[1] Jan de Leeuw. Modern multidimensional scaling: Theory and applications (second edition). *Journal of Statistical Software*, 14, 10 2005.

[2] 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.

[3] John A. Lee and Michel Verleysen. Shift-invariant similarities circumvent distance concentration in stochastic neighbor embedding and variants. *Procedia Computer Science*, 4:538–547, 2011. Proceedings of the International Conference on Computational Science, ICCS 2011.

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

[5] 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.

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

[7] Anna Belkina, Christopher Ciccolella, Rina Anno, Richard Halpert, Josef Spidlen, and Jennifer Snyder-Cappione. Automated optimized parameters for t-distributed stochastic neighbor embedding improve visualization and analysis of large datasets. *Nature Communications*, 10, 11 2019.

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

[9] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[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] Moshe Lichman. UCI Machine Learning repository, 2013.

[12] Renata C. B. Madeo, Clodoaldo A. M. Lima, and Sarajane M. Peres. Gesture unit segmentation using support vector machines: Segmenting gestures from rest positions. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, SAC '13, page 46–52, New York, NY, USA, 2013. Association for Computing Machinery.

[13] J. A. Lee, D. H. Peluffo-Ordóñez, and M. Verleysen. Multi-scale similarities in stochastic neighbour embedding: Reducing dimensionality while preserving both local and global structure. *Neurocomputing*, 169:246–261, 2015.