ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

Stochastic quartet approach for fast multidimensional scaling

Pierre Lambert[1], Cyril de Bodt[1], Michel Verleysen[1], John A. Lee[1,2]

1- UCLouvain.be - ICTEAM/ELEN
Place du Levant 3 L5.03.02, 1348 Louvain-la-Neuve - Belgium

2- UCLouvain.be - IREC/MIRO
Avenue Hippocrate 55 B1.54.07, 1200 Brussels - Belgium

Abstract. Multidimensional scaling is a statistical process that aims to embed high-dimensional data into a lower-dimensional, more manageable space. Common MDS algorithms tend to have some limitations when facing large data sets due to their high time and spatial complexities. This paper attempts to tackle the problem by using a stochastic approach to MDS which uses gradient descent to optimise a loss function defined on randomly designated quartets of points. This method mitigates the quadratic memory usage by computing distances on the fly, and has iterations in $\mathcal{O}(N)$ time complexity, with N samples. Experiments show that the proposed method provides competitive results in reasonable time. Public codes are available at `https://github.com/PierreLambert3/SQuaD-MDS.git`.

1 Multidimensional scaling and its limitations

Dimensionality reduction (DR) is the process of mapping high-dimensional (HD) observations into a lower-dimensional (LD) space such that the LD embedding is a faithful representation of the HD data. The main DR uses are in machine learning, to curb the curse of dimensionality, and in visualisation. Mapped data can reveal structures that would lay hidden from the human perception if left in HD. Typically, some information is lost by the DR and, therefore, each DR method has a take on what kind of information should be preserved and what can be lost. Used frequently in visualisation, t-SNE [1] aims at retaining the neighbourhood of each point according to a distance metric and a perplexity, which reflects the size of the neighbourhood to preserve. While t-SNE excels at retaining local structures, sufficiently remote points tend to be considered equally distant by the algorithm and, therefore, the larger-scale structures can be distorted. Such distortions can lead to erroneous conclusions by the human user, who might overestimate the dissimilarity between two clusters that are distant in the LD embedding. For this reason, using multiple DR paradigms in conjunction is a good practice in visualisation: another embedding that preserves distances instead of neighbourhoods would have prevented this erroneous conclusion.

This paper considers metric multidimensional scaling (MDS): a DR technique that produces a LD embedding such that the pairwise distances in LD reflect those in HD. MDS minimises a cost function which, in its simplest form, is the sum of the squared differences between distances in HD and the Euclidean distances in LD. A common strategy to optimize this cost function is based on

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

the SMACOF algorithm [2], involving iterations of $\mathcal{O}(N^2)$ time complexity with N samples. This quickly becomes problematic when N grows. SMACOF also requires the full matrix of pairwise HD distances, consuming $\mathcal{O}(N^2)$ memory.

Some alternatives with lower time or memory complexities exist, such as the iterative spring-based model [3], where each point is subject to forces computed according to two constant-sized sets of points. This enables iterations with $\mathcal{O}(N)$ time complexity. However, the sets are subject to iterative refinements, making the total number of required iterations increase with the size of the data set. A different divide-and-conquer approach [4] applies MDS on sub-matrices of the HD distance matrix and then stitches together the results. Another take on the problem is proposed in landmark MDS [5], which first embeds a small subset of the data in LD and then arranges the rest of the points with respect to these landmarks. Alternatively, the effects of high computational complexities can be reduced by focusing on certain parts of the LD embedding thanks to the input of a user [6].

While these accelerated MDS methods provide interesting results, they often involve multi-step approaches that complicate the algorithm and tend to make it monolithic in the sense that the optimisation process becomes an intractable block once the optimisation started. This paper proposes an algorithm that has $\mathcal{O}(N)$ time complexity and does not require a full distance matrix, making it more memory-efficient. In addition to showing that the algorithm can yield good distance correlation between the HD and LD spaces, this paper shows that by using a simple gradient descent optimisation, the algorithm is open for hybrid approaches, where gradients of different types are mixed.

Section 2 presents the algorithm, while Section 3 is dedicated to an empirical assessment of its quality and speed performance. Section 4 sketches some conclusions and perspectives.

2 Proposed method

The proposed algorithm performs MDS by using gradient descent on a cost function defined on quartets of points. Before defining the gradients, this section starts by explaining how to articulate a distance scaling problem around the use of quartets.

2.1 Quartet-based optimisation

Let us consider a quartet formed by four points that has a HD distance matrix δ and a LD Euclidean distance matrix d. If these points have a LD embedding that perfectly preserves their HD distances, then the relative distance between the i^{th} and the j^{th} points in LD, d^{r}_{ij}, also perfectly matches their relative distance in HD, δ^{r}_{ij}. These relative distances are defined as

$$d^{\text{r}}_{ij} = d_{ij} \Big/ \Big(\sum_{a=1}^{3} \sum_{b=a+1}^{4} d_{ab} \Big) \quad \text{and} \quad \delta^{\text{r}}_{ij} = \delta_{ij} \Big/ \Big(\sum_{a=1}^{3} \sum_{b=a+1}^{4} \delta_{ab} \Big) \ .$$

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

A cost function for the preservation of relative distances within this quartet of points can be defined:

$$C_{\text{quartet}} = \sum_{a=1}^{3} \sum_{b=a+1}^{4} (\delta_{ab}^{\text{r}} - d_{ab}^{\text{r}})^2 \quad . \tag{1}$$

To apply this quartet approach to the whole data set, the algorithm iteratively and randomly splits the observations in groups of 4, and the gradient for each point is computed according to (1). This use of quartets should not be confused with triplet methods such as stochastic triplet embedding [7], as they work with data of different natures. Triplet embeddings use partial similarity rankings of the form "A is more similar to B than to C" often resulting from human judgement, whereas the proposed fast MDS method uses HD observations.

Using quartets of points gives constant-time gradients and has a natural analogy to triangulation (1 point versus 3 others) when applied to a target dimension of 2. Quartets are also the reason why the full HD distance matrix isn't needed : only 6 HD distances are used at any instant. It is noteworthy that using relative distances precludes any preservation of scale: the LD distances tend to correlate to those in HD, but the scale in LD will stay close to the scale at which the embedding is initialised.

2.2 The gradients

Considering a quartet of points with a LD Euclidean distance matrix d and a HD distance matrix δ, I is the 4x4 identity matrix and x_q is the value taken for some LD dimension by the q^{th} point in the quartet. For the sake of concision, only the gradients for one term in the sum of (1) is shown here, as the other terms in the sum share similar forms:

$$\frac{\partial(\delta_{ij}^{\text{r}} - d_{ij}^{\text{r}})^2}{\partial x_q} = \frac{2}{S}(d_{ij}^{\text{r}} - \delta_{ij}^{\text{r}})\left(\frac{I_{qi}(x_q - x_j) + I_{qj}(x_q - x_i)}{d_{ij}} - d_{ij}^{\text{r}} \sum_{b \in \{1,..,4\} \setminus q} \frac{x_q - x_b}{d_{qb}}\right)$$

with $S = \sum_{i<j} d_{ij}$. The motivation for using relative distances is that each of the six LD distances in the quartet contributes to the update for each LD point, as opposed to direct distance preservation where only three distances would contribute. This results in the behaviour illustrated in Fig. 1, where each point moves to minimise the error on the quartet as a whole.

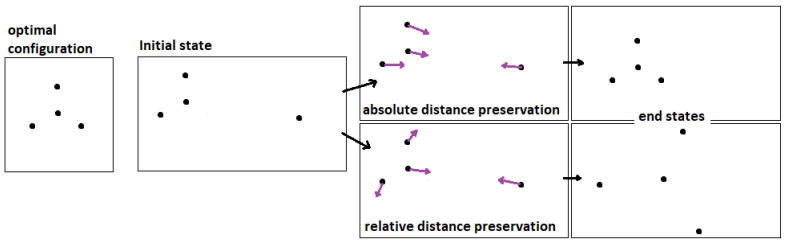

Fig. 1: Difference in behaviour when using relative distances and raw distances.

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

3 Experiments and discussion

To assess the performance of the proposed algorithm in terms of speed and distance preservation, it is compared to the SMACOF-based MDS [2] on various data sets. Section 3.1 details our implementation, while Section 3.2 describes the results.

3.1 Implementation

Our implementation uses a gradient descent optimizer in its simplest form: the gradients are scaled by a learning rate and then directly subtracted to the positions of the LD points. During the first part of the optimisation, the HD distances are squared before being transformed to relative distances. This helps the optimisation by exaggerating the differences between the distances within a quartet. Because of the stochastic nature of the optimisation, the updates on the position of the points tend to be erratic; this isn't a problem during the first part of the optimisation, but it renders the fine-tuning of the final positions difficult. To counter this behaviour, it is important that the learning rate decays to small values. We determined experimentally that, for embeddings initialised with a standard deviation around 10, a learning rate starting in the hundreds that decays to 10^{-3} is adequate. Having it decay to greater values tends to reduce the quality of the results. All the following embeddings using the proposed fast MDS are given the same hyper-parameters. They result from a single run of 1000 iterations, with a target dimension of 2 and a PCA initialisation scaled to have a standard deviation of 10. Non-PCA initialisations can produce similar results, but they typically require a larger number of iterations before convergence. The code can be found at: `https://github.com/PierreLambert3/SQuaD-MDS.git`.

3.2 Empirical assessment

Seven data sets are featured in this paper, with size N and dimensionality M. Abalone: (N, M) = (4177, 8); coil20: (N, M) = (1440, 1024); a subset of MNIST digits: (N, M) = (3400, 784); a subset of MNIST fashion: (N, M) = (4000, 784); satellite: (N, M) = (4434, 36); a subset of Gisette: (N, M) = (3700, 4900); RNAseq: (N, M) = (10^4, 50). Most sets are available from the UCI machine learning repository. RNAseq contains single-cell data used in [8]; the same 50 principal components from the selected features are used, as in [8].

Table 1 shows the Pearson correlation between the pairwise distances in HD and LD for various models on the data sets. The 1st row shows results obtained with PCA. The 2nd row shows the best result of 10 different initialisations for the SMACOF MDS algorithm as implemented by scikit-learn. The 3rd row is also the result of a MDS using SMACOF but this time initialised with PCA. The 4th row is our fast MDS algorithm in its default version, and the last row is an experiment where the HD distances for the quartets are first transformed non-linearly before being used by the cost function. This transformation is of the form $\delta'_{ij} = 1 - \exp(-(\delta_{ij} - \delta_{\min})/(2\sigma))$ with σ being the standard deviation of the HD

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

distances in the quartet and $\delta_{\min}$ the smallest HD distance in the quartet. This transformed distance is then divided by the sum of the transformed distances to become relative. The purpose of this transformation is to increase the contrast between the HD distances within the quartet in the hope of reducing the effect of the concentration of norms [9] when the input dimension is high. In terms of correlation of distances, Table 1 shows that the proposed method can provide competitive results, and that its results tend to be particularly good when the data dimension is high.

	abalone	satellite	RNAseq	digits	fashion	coil20	gisette
PCA	0.82	0.98	0.93	0.50	0.69	0.76	0.45
10x MDS	0.82	0.93	0.93	0.63	0.82	0.80	0.84
PCA-MDS	0.99	0.99	0.90	0.66	0.89	0.84	0.67
fastMDS	0.98	0.97	0.95	0.73	0.94	0.86	0.85
fastMDS-rbf	0.98	0.96	0.96	0.75	0.92	0.87	0.89

Table 1: Correlation of the distances in LD and HD; the data sets are of increasing dimensionality from left to right.

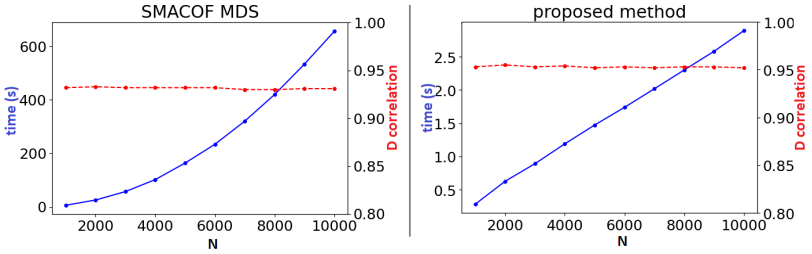

Fig. 2: Evolution of computation time and distance correlation for samples of RNAseq of size varying between 10^3 and 10^4 with steps of 10^3.

Figure 2 shows that the fast MDS algorithm runs in linear time with N, as opposed to the quadratic scaling of SMACOF MDS; the distance correlation is stable with different sizes N, which can be surprising considering that the proportion of HD distances that are used during the optimisation decreases when N increases. Actually, the number of HD distances used by the proposed algorithm is *number_of_iterations*6_distances_per_quartet*number_of_quartets*, in this case $10^3 * 6 * \frac{N}{4}$, but the total number of HD distances is $\frac{N*(N-1)}{2}$. This supports the intuition that when N is high, using only a subset of the distances can be sufficient for a good embedding, which is a principle of landmark-based approaches.

Figure 3 shows some 2-D embeddings of a subset of RNAseq; 'D correlation' is the Pearson correlation between HD and LD distances and 'R_{NX} AUC' is the AUC of the $R_{\text{NX}}(K)$ curves [10], which reaches 1 when the multi-scale neighbourhoods of the data are well preserved. Since the proposed method relies on a simple gradient descent, it is possible to combine it with another gradient-based algorithm. In the 3$^{\text{rd}}$ scatter plot, we added the quartet based distance gradients to those of t-SNE to obtain a hybrid embedding. The figure shows that

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

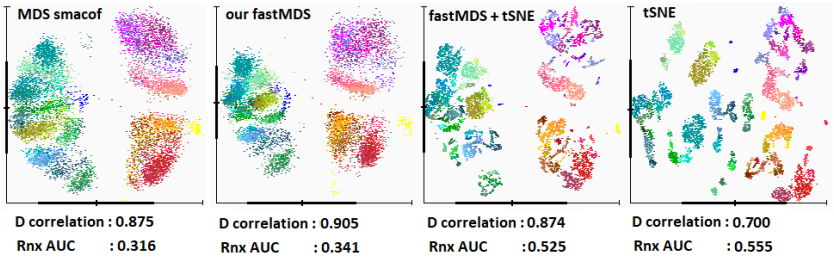

Fig. 3: 2-D embeddings of a subset of RNAseq of size 9300.

this hybrid approach can preserve neighbourhoods quite well while keeping the distances meaningful.

4 Conclusion and perspectives

Experiments show that the proposed fast MDS method produces competitive results while keeping the algorithm simple and the computational complexities low. We may have only scratched the surface of the possibilities that a stochastic quartet-based approach can offer; we can imagine an extension that would enable the use of weighted distances, the use of different quartet-based metrics, or an adaptation for ordinal MDS. Another perspective is to further study the use of quartet-based distance gradients in a hybrid approach to DR, by mixing the distance preservation and the neighbourhood preservation paradigms.

References

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

[2] J. de Leeuw and P. Mair. Multidimensional scaling using majorization: Smacof in r. *Journal of Statistical Software, Articles*, 31(3):1–30, 2009.

[3] M. Chalmers. A linear iteration time layout algorithm for visualising high-dimensional data. *Proceedings of the 7th conference on Visualization*, pages 127 – 131, 12 1996.

[4] Tzeng, Henry Lu, and Wen-Hsiung Li. Multidimensional scaling for large genomic data sets. *BMC bioinformatics*, 9:179, 02 2008.

[5] V. Silva and J. Tenenbaum. Sparse multidimensional scaling using landmark points. *Technology*, 01 2004.

[6] M. Williams and T. Munzner. Steerable, progressive multidimensional scaling. In *IEEE Symposium on Information Visualization*, pages 57–64, 2004.

[7] L.J.P. van der Maaten and K.Q. Weinberger. Stochastic triplet embedding. *Proceedings of the IEEE International Workshop on Machine Learning for Signal Processing*, 2012.

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

[9] D. Francois, V. Wertz, and M. Verleysen. The concentration of fractional distances. *IEEE Trans. Knowl. Data Eng.*, 19(7):873–886, 2007.

[10] 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.