



Institute for Information
and Communication Technologies,
Electronics and Applied Mathematics

Robust and Fast Neighbor Embedding Algorithms

Cyril de Bodt

Thesis submitted in partial fulfillment
of the requirements for the degree of
Docteur en sciences de l'ingénieur et technologie

Dissertation committee:

Prof. Michel Verleysen (UCLouvain, advisor)
Prof. John A. Lee (UCLouvain, advisor)
Prof. Jean-Charles Delvenne (UCLouvain, Belgium)
Prof. Laurent Jacques (UCLouvain, Belgium)
Prof. Barbara Hammer (Bielefeld University, Germany)
Prof. Benoît Frénay (University of Namur, Belgium)
Prof. Jean-Pierre Raskin (UCLouvain, chair)

Version of March 31, 2020.

Acknowledgments

IT is sometimes difficult to realize that this thesis is actually already approaching its end, perhaps even more during the present period of general containment, but I guess that it is better to conclude it with the feeling that it was a bit too short rather than a bit too long. Conducting this work has been a personal aspiration for long, and I am well aware that fulfilling it during these last few years has only been possible thanks to the tremendous help of wonderful persons. As a very modest testimony of my profound gratitude, I would like to take the time to acknowledge them.

I first wish to thank my advisors, Prof. Michel Verleysen and Prof. John A. Lee. For your guidance, your kindness, your trust. For the time you have been devoting to me, and your persistent attention to make yourselves available. For your answers at any time, hour and day, independently of the proximity of the current deadline. For your precious advice and the uncountable great moments spent together. For our comforting conversations and your constant attention and remarkably helpful sympathy during the private defense and all our meetings. For the opportunities you have been offering me, the scientists I have been meeting thanks to you, the incredible ESANN conferences, magnificent travels, funny lunches and entertaining walks. I consider myself as extremely lucky to have been learning by your sides, and will always remember your continuous support. You impress me as professors, delight me as friends, inspire me as persons.

I would also like to thank Prof. Jean-Charles Delvenne and Prof. Laurent Jacques, for joining my supervisory panel and my thesis jury. Either in the context of remarkable lectures or insightful discussions, your clarity of thought is a model of accomplishment. In addition to your expertise, I have especially been enjoying your sympathy and pleasing company. Your nice suggestions and comments during the private defense moreover substantially helped me to improve this manuscript.

I thank Prof. Barbara Hammer and Prof. Benoît Frénay for accepting to be part of my thesis jury. Your readiness and friendliness during our interactions have been very appreciated. I am honored that this thesis is evaluated

by as renowned and talented researchers as you. I very much enjoyed our discussions during the private defense, as well as the enhancements that you pointed out for this thesis.

I thank Prof. Jean-Pierre Raskin for chairing my thesis jury. Your kindness and promptness in responding to my messages have been boosting in many regards. Your involvement and your organization during the private defense tremendously facilitated its smooth running.

I am grateful to the Fonds de la Recherche Scientifique de Belgique (F.R.S.-FNRS) for their funding to carry out this project. I also thank the Université catholique de Louvain (UCLouvain) and the ICTEAM institute in particular for providing me with more than convenient facilities in order to complete this thesis. The supercomputing resources of the CISM/UCLouvain and the Consortium des Equipements de Calcul Intensif en Fédération Wallonie Bruxelles (CÉCI) have been notably useful for some computational aspects of this work.

I thank the numerous colleagues and friends from ICTEAM and beyond, with whom sharing breaks and playing tennis or padel have been highly pleasant, as well as attending many interesting seminars, memorable ESANN conferences and other workshops with amazing atmosphere in different countries. I also thank the administrative and technical staff members for their good mood, reactivity and daily assistance. Meeting Daniel, Walter, Francesco and several visitors and foreign researchers has been particularly enriching, and I would like to thank them too for our fruitful collaborations and enjoyable conversations.

I thank Gildas, Zélie, Tanguy, Arthur, Salomé, Andrew, Laura, Corentin, Fabienne, Christian and the (very) many (very) soon to come, for your smiles, joy of life, happiness, delicious dishes and the wonderfully relaxing times spent together. I feel privileged to have been introduced to a lifestyle that was totally new to me, and which has been changing much of my daily habits!

I thank my brothers, Jérémy and Grégory, and my parents. For your understanding, your support, your encouragements. For the fantastic moments spent with you and for all these hilarious evenings. Thank you Jérémy for your communicative laughs and your cheerfulness. Thank you Grégory for your listening and your care for others. Thank you Dad for your availability and your interest in all occasions. Thank you Mom for your attention and your ever goodwill.

Finally, I thank Dounia. Your presence, help, persistence, resilience, drive, dedication, determination, happiness, fabulous enthusiasm and extraordinary serenity in all circumstances have been crucial to achieve this thesis, and even more fundamental for me. Thank you for all of this, and for everything else.

Abstract

IN numerous machine learning (ML) settings, complex data mining tasks necessitate user interaction and cannot be completely automated. This interaction can take place in the context of a data exploratory phase, during which data visualization helps determining and refining the application needs. Since most ML databases are nowadays high-dimensional (HD), their visualization entails considering approaches from nonlinear dimensionality reduction (DR). This field aims at creating meaningful low-dimensional (LD) versions of HD data. The advent of neighbor embedding (NE) techniques impressively improved state-of-the-art DR performances.

Nevertheless, the data exploratory requirements in current ML applications involve generalizing modern NE algorithms in several aspects. Namely, they should be robustly able to handle unconventional data types, such as incomplete databases which are omnipresent in data analysis. Also, they must be sufficiently fast to process very large data sets, being ubiquitous presently. This thesis contributes to both of these aspects.

Regarding the ability to deal with incomplete data sets, common missing data imputation techniques are not suited to nonlinear DR, as they at best enable applying a DR scheme on the expected database. Since NE approaches are nonlinear, this differs from minimizing their expected cost function. The thesis addresses this limitation by proposing a general methodology to compute the LD embedding minimizing the cost function expectation, thanks to the multiple imputation framework.

As to the development of fast NE schemes, multi-scale techniques are of great interest among NE methods as they account for the global HD organization to define the LD space, delivering outstanding DR quality. Their time complexity in the number of data samples however prevents tackling large-scale databases. The thesis addresses this difficulty by presenting fast multi-scale NE algorithms which account for the dense nature of the multi-scale similarities, providing high quality embeddings of very big data sets.

The robust and fast NE algorithms designed in this thesis hence open the path to enhanced HD data exploration in ML through visualization.

Author's publications

IN the context of this thesis, several projects ensued in scientific publications, which are listed hereunder. Their relations with the different sections of this dissertation are detailed as well.

Journal articles

[dBMVL19]: de Bodt, C., Mulders, D., Verleysen, M., & Lee, J. A. (2019). Nonlinear Dimensionality Reduction with Missing Data using Parametric Multiple Imputations. *IEEE Transactions on Neural Networks and Learning Systems*, 30(4), 1166-1179.

- The content of Chapter 3 corresponds to this paper.
- © 2019 IEEE.

[dBMVL20]: de Bodt, C., Mulders, D., Verleysen, M., & Lee, J. A. (2020). Fast Multi-scale Neighbor Embedding. *IEEE Transactions on Neural Networks and Learning Systems*. Under review.

- The content of Chapter 4 describes this paper, which is currently under review after a first round of revisions.

Conference proceedings (peer-reviewed)

[dBMVL18a]: de Bodt, C., Mulders, D., Verleysen, M., & Lee, J. A. (2018). Extensive assessment of Barnes-Hut t -SNE. In *ESANN* (pp. 135-140).

- The content of Section 4.2 is related to this paper.

[dBMVL18b]: de Bodt, C., Mulders, D., Verleysen, M., & Lee, J. A. (2018). Perplexity-free t -SNE and twice Student tt -SNE. In *ESANN* (pp. 123-128).

- The content of Section 2.2 and the experiments in Section 4.4 refer to this paper.

In addition, some publications resulted from other works which are not discussed in this manuscript. They are provided below.

Journal articles

[**MdBB⁺19**]: Mulders, D., de Bodt, C., Bjelland, J., Pentland, A. S., Verleysen, M., & de Montjoye, Y.-A. (2019). Inference of Node Attributes from Social Network Assortativity. *Neural Computing and Applications*, DOI: 10.1007/s00521-018-03967-z.

[**MdBL⁺20**]: Mulders, D., de Bodt, C., Lejeune, N., Courtin, A., Liberati, G., Verleysen, M., & Mouraux, A. (2020). Dynamics of the perception and EEG signals triggered by tonic warm and cool stimulation. *PLoS One*.

[**LSdBL⁺20**]: López-Sánchez, D., de Bodt, C., Lee, J. A., González Arrieta, A., & Corchado, J. M. (2020). Tuning Database-Friendly Random Projection Matrices for Improved Distance Preservation on Specific Data. *Pattern Recognition*. Under review.

[**SdBL⁺20**]: Serna, W., de Bodt, C., Lee, J. A., Alvarez, A. M., Verleysen, M., & Orozco, A. A. (2020). Semi-supervised neighbor embedding with multi-scale neighborhood preservation. *Journal of Artificial Intelligence Research*. Under review.

Conference proceedings (peer-reviewed)

[**dBMLS⁺19**]: de Bodt, C., Mulders, D., López-Sánchez, D., Verleysen, M., & Lee, J. A. (2019). Class-aware *t*-SNE: cat-SNE. In *ESANN* (pp. 409-414).

[**CdBV⁺20**]: Crecchi, F., de Bodt, C., Verleysen, M., Lee, J. A., & Bacciu, D. (2020). Perplexity-free Parametric *t*-SNE. In *ESANN*.

[**MdBB⁺17**]: Mulders, D., de Bodt, C., Bjelland, J., Pentland, A. S., Verleysen, M., & de Montjoye, Y.-A. (2017). Improving Individual Predictions using Social Networks Assortativity. In *WSOM+* (pp. 127-134). IEEE.

- [**MdBV⁺17**]: Mulders, D., de Bodt, C., Verleysen, M., Bjelland, J., Pentland, A. S., & de Montjoye, Y.-A. (2017). Improving Individual Predictions using Social Networks Assortativity. In *Benelearn* (pp. 134-136).
- [**MdBL⁺19**]: Mulders, D., de Bodt, C., Lejeune, N., Lee, J. A., Mouraux, A., & Verleysen, M. (2019). Tensor factorization to extract patterns in multimodal EEG data. In *ESANN* (pp. 601-606).
- [**MdBL⁺18b**]: Mulders, D., de Bodt, C., Lejeune, N., Mouraux, A., & Verleysen, M. (2018). Spatial filtering of EEG signals to identify periodic brain activity patterns. In *LVA/ICA* (pp. 524-533).
- [**MdBL⁺18a**]: Mulders, D., de Bodt, C., Lejeune, N., Mouraux, A., & Verleysen, M. (2018). Linear Periodic Discriminant Analysis of Multidimensional Signals. In *ICONIP* (pp. 476-487, vol. 6).

Contents

Acknowledgments	i
Abstract	iii
Author's publications	v
Contents	ix
1 Introduction	1
1.1 Dimensionality reduction	2
1.2 Robust and fast neighbor embedding	5
2 Neighbor embedding algorithms	9
2.1 Single-scale methods	9
2.2 Multi-scale methods	19
3 Nonlinear dimensionality reduction with missing data using parametric multiple imputations	29
3.1 Gaussian mixture models for HD and incomplete data	32
3.1.1 Missing data management	32
3.1.2 Sweep inversion operator	34
3.1.3 High-dimensional extension	35
3.1.4 EM algorithm time complexity	36
3.1.5 Meta-parameter tuning	36
3.1.6 EM initialization	37
3.1.7 Conditional mixture model sampling	38
3.2 Multiple imputations for dimensionality reduction with missing data	38
3.3 Experimental results	42
3.3.1 Quality assessment	45
3.3.2 Experimental setting	46
3.3.3 Results	47

3.4	Discussion	55
4	Fast neighbor embedding	57
4.1	Accelerating single-scale neighbor embedding methods	59
4.2	Extensive quality assessment of Barnes-Hut t -SNE	64
4.3	Accelerating multi-scale neighbor embedding methods	78
4.4	Experimental results	87
4.4.1	Comparison between fast and non-fast methods	88
4.4.2	Scalability analysis of the fast methods	100
4.5	Discussion	104
5	Conclusions	109
	Bibliography	115
	List of symbols and acronyms	131
	Index	143

Chapter 1

Introduction

MACHINE learning (ML) consists in developing mathematical methods learning from data, for instance to perform predictions on new ones [Bis06]. Classification and regression, two important branches of the field, have been widely studied during the last decades, in such a way that many solutions currently exist to answer a broad range of data-driven modeling tasks. On the other hand, it is more and more accepted in the ML community that a complex data mining task cannot be fully automated and that an exploratory phase, a “mining” step requiring the user feedback, is necessary [Mun14]. The real quantitative needs are indeed ill-defined in numerous applications and some compromises must be found by interacting with the user [CCR⁺19]. Traditionally, the user typically only observes the evaluation of the ML algorithm cost function, from which it is very difficult to evaluate the adequacy of the performances and provide a relevant feedback [Mun14]. This is why, in increasingly many contexts, data visualization is embedded in the exploratory phase, to enable the user to generate hypotheses and discover data relationships helping to refine the application needs and modify some algorithm parameters based on the visual feedback [DPATBEG19, ZJ20]. Since high-dimensional (HD) data are now ubiquitous in ML tasks, visualizing them necessitates relying on visualization techniques from ML-based nonlinear dimensionality reduction (NLDR) [BMH⁺19, Pla13, KB19].

For this reason, Section 1.1 first summarizes some aspects of the field of dimensionality reduction for data visualization. Then, Section 1.2 introduces the developments conducted in this thesis, which aim at designing robust and fast NLDR algorithms to ease HD data exploration in ML contexts through visualization.

1.1 Dimensionality reduction

Dimensionality reduction (DR) aims at faithfully computing relevant low-dimensional (LD) representations of high-dimensional data sets, mainly for exploratory visualization or to foil the curse of dimensionality [LV07, BBH12, CP97]. This curse encompasses the inherent difficulties to cope with HD data and motivated the development of adequate approaches to extract meaningful LD features [LZZ⁺18, ZNWL19]. In visualization context, the relevance of a LD embedding is typically related to the HD neighborhood preservation.

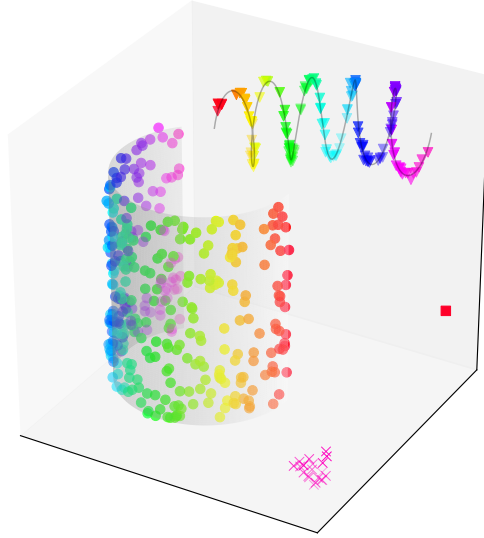
Figure 1.1 illustrates this concept using a toy example. Markers in Fig. 1.1a stand for 3-D data points. One can notice that the circles are distributed on a 2-D surface being half a cylinder, the triangles follow a 1-D manifold which is a spiral, the crosses form a small cluster, and the single square appears to be an outlier, since it is isolated and far from other data samples. As some noise can typically impact measurements in real settings, small perturbations make the circles and triangles slightly deviate from their respective 2-D and 1-D subspaces.

Applying a DR algorithm to data from Fig. 1.1a should reveal the different kinds of identified structures. Figure 1.1b achieves this objective, in which the 2-D coordinates are derived by a neighbor embedding DR technique known as the Neighborhood Retrieval Visualizer (NeRV) [VPN⁺10]. In this 2-D embedding, the circles now delimit a piece of plane, meaning that NeRV properly unrolled the half cylinder. A 1-D string correctly represents the HD spiral, a cluster of crosses can be observed, and the square is depicted as an outlier.

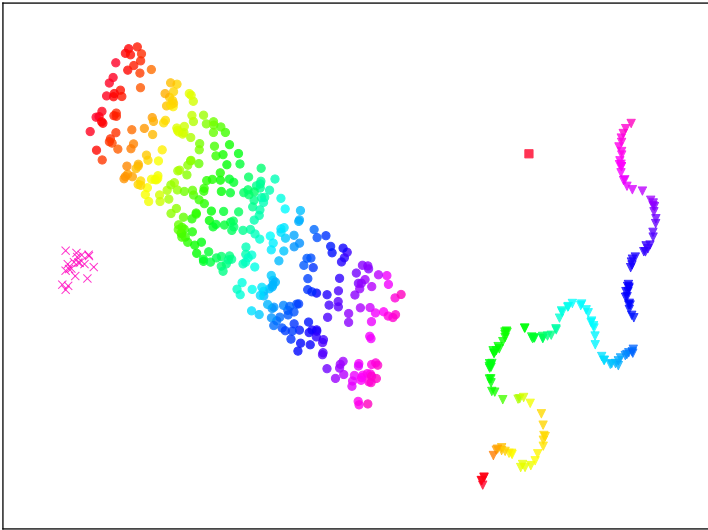
Therefore, from Fig. 1.1a to Fig. 1.1b, dissimilar HD data items are mapped far apart in the LD space, while similar HD data items are represented close one to another in the LD embedding. Such a behavior is highly desired when applying a DR scheme to obtain faithful LD versions of HD data, and drives the design of DR methods. One should stress that the colors in Fig. 1.1 are employed only as visual cues to enable linking the corresponding data points from Fig. 1.1a to Fig. 1.1b, and are not provided to the DR algorithm.

In practice, users can face much larger data dimensions, preventing direct data visualization [BMH⁺19, CCR⁺19]. In this case, analyzing a representation such as Fig. 1.1a is impossible. This further motivates designing DR algorithms providing LD embeddings of the data, to enable users to intuitively grasp the content of HD data and get insights of the HD structures through visualization [DPATBEG19, Pla13], similarly as in Fig. 1.1b.

Through the history of DR, different paradigms have emerged in order to formalize the neighborhood preservation principle in terms of mappings from HD to LD coordinates [LV07, BBH12], like the reproduction of dis-



(a) 3-D data set.



(b) 2-D embedding.

Figure 1.1: (a) Artificial data set in 3-D. (b) 2-D representation of the data set depicted in (a), obtained thanks to a modern DR method.

tances [BG05] or neighborhoods [Koh90, HR02]. Early linear models such as principal component analysis (PCA) [Jol86, Jol02, Hot33] and classical metric multidimensional scaling (MDS) [BG05] project the HD vectors by maximizing the variance or dot product preservation.

These approaches being limited to linear mappings, nonlinear metric MDS variants [Sam69] refine them by defining (weighted) distance preservation schemes, based on either Euclidean or approximated geodesic measures [Ten98, TDSL00, LLDV00, LLV02, LLV04]. Other studies define affinity matrices driving the computation of the LD embedding [BN02, RS00, NLKC06, SSM97, SVGVDM03]. In the visualization context however, these methods hardly outperform the older techniques [LV07, LV09, vdMH08], possibly because they reduce to applying classical MDS in an unknown feature space [SSM98, HLMS04]. The norm concentration phenomenon indeed particularly affects distance preserving schemes [FWV07], since it leads pairwise distances to become more and more similar as the dimension increases [LV11].

In contrast, genuine neighbor embedding (NE) algorithms such as Stochastic Neighbor Embedding (SNE) [HR02] and variants [vdMH08, VPN⁺10] mitigate this phenomenon by defining neighbor probabilities, or similarities, and confronting them between the HD and LD spaces to obtain the LD coordinates [LV11, MHM18], leading to outstanding DR performances. These achievements motivated the development of numerous SNE-based techniques, for instance with alternative mixtures of divergences as cost functions [VPN⁺10, VK07, BHBV12, LRB⁺13], heavy-tailed LD similarities such as in t -SNE [vdMH08, YKXO09, GSH15], and improved optimization procedures [CP10, BHV⁺11, vdM10, VCP12, PL13, VCP13, GMH12, NJC04, YWO10, MH05, Vla19].

Originally, NE methods required the user to specify some perplexity for the neighborhood probability distributions, resulting in a preservation oriented towards neighborhoods of the corresponding size. The perplexity is indeed a hyper-parameter which can be interpreted as a soft neighborhood size. These methods may hence be viewed as single-scale, with fixed resolution. More recently, multi-scale NE approaches were introduced by combining neighborhood probabilities tuned at various data scales [LPOV15, dBMVL18b], leading to meta-parameter-free techniques. Compared to their single-scale counterparts, the multi-scale schemes preserve neighborhoods of all sizes substantially better thanks to their ability to consider both local and global structures while computing the LD embedding. This enables for instance to reproduce small neighborhoods as well as or better than t -SNE, while considerably improving the preservation of larger ones [LPOV15].

1.2 Robust and fast neighbor embedding

Despite all these important advances, facilitating HD data exploration in current ML applications still requires to generalize modern DR schemes, namely to allow the user to rely on visualization algorithms that are able to deal with the actual challenging databases faced in practice nowadays. This entails designing DR schemes with **robustness and flexibility** in order to deliver meaningful and appropriate embeddings of non-traditional types of data which are presently frequently encountered in ML tasks, such as incomplete databases, and with sufficient **speed** to enable tackling large-scale data sets in reasonable computation time, hence without preventing data exploration. Therefore, the objective of this thesis is to develop robust and fast neighbor embedding algorithms, leveraging the recent tremendous achievements in DR to ultimately create new opportunities for more efficient data exploratory phases. Besides robustness, flexibility and speed, one must stress that it is also crucial for nonlinear DR techniques to deliver very high quality embeddings, to faithfully and efficiently guide the exploratory data analysis. In this line, the multi-scale NE setting seems particularly attractive as a strong basis to frame the algorithms to design in this thesis.

Concerning **robustness and flexibility**, the Big Data era, despite its name, not only entails very large databases, but overall diverse and unconventional types of data. For instance, incomplete data sets are nowadays more and more widely considered in machine learning [GLSGFVV09, DV12, BE10]. While missing values occur for many different reasons depending on the data type, any inference task on such incomplete database should take the missing entries into account to avoid biased conclusions [EDVL13, ELVB14]. Discarding the samples with missing features is most often not acceptable as it drastically reduces the data set size as the proportion of missing values increases [SEM⁺16]. Surprisingly, despite all the recent advances in DR, the field had not yet designed methods directly able to deal with incomplete data sets. As missing data related problems are now ubiquitous in data analysis, such a DR methodology has been developed in the context of this thesis [dBMVL19]. It first computes a high-dimensional extension of parametric Gaussian mixture models of the feature space distribution to account for the conditional distribution of the missing values given the observed ones. It then finds the most faithful LD embedding on average to the complete data case, by minimizing the cost function expectation thanks to the multiple imputation paradigm. Being applicable with any DR scheme optimizing a cost function, it experimentally outperforms alternative approaches, statistically significantly over the missing entries introduced in the numerous considered data sets. Free and user-friendly implementations are publicly available online at <https://github.com/cdebodt>, enabling easy usage of the proposed approach.

As to **speed**, neighbor embedding DR methods typically have a time complexity which scales quadratically with the number of samples in the data set [HR02]. Their application is therefore totally unaffordable on large databases, which become omnipresent in data analysis [PTM⁺19]. Some studies hence proposed fast schemes and lower-complexity approximations for SNE-like methods [YPK13, VCP14]. As an example, the Barnes-Hut (BH) acceleration of t -SNE approximates t -SNE gradient in almost linear time [vdM14]. Although some works reported impressive computation time gains over t -SNE [vdM14], surprisingly few studies analyzed its DR quality. Thanks to recently developed neighborhood-based DR performance criteria [LV09, LV10], the BH t -SNE quality has been examined during this thesis on a large number of real-world and artificial data sets [dBMVL18a]. The experimental results suggest that t -SNE and its BH acceleration perform equally well. The usage of the BH variant can therefore only be favored, thanks to its reduced time complexity.

Meanwhile, the approximation techniques as employed in BH t -SNE can unfortunately only be applied in the single-scale neighbor embedding framework. Indeed, the multi-scale HD similarities can be shown to be dense [LPOV15], at the opposite of the sparse ones of t -SNE [vdM14]. On the other hand, this non-sparsity is a key ingredient of multi-scale algorithms and allows them to impressively outperform single-scale schemes in terms of DR quality [LPOV15]. Indeed, while t -SNE delivers embeddings preserving very well local neighborhoods around the data points [vdMH08, LRB⁺13], multi-scale methods also account for the global structure of the HD space, preserving larger neighborhoods as well [LPOV15]. Therefore, fast accelerations of multi-scale methods have been developed in the frame of this thesis [dBMVL20] to enable their large-scale use, thanks to space-partitioning trees and randomized sub-samplings of the HD data at different scales. They run in nearly linear time with the number of data samples, with an additional logarithmic factor standing from the multi-scale behavior of the methods. Extensive experiments highlight that the proposed accelerated algorithms are both faster than the original, non-accelerated multi-scale schemes by orders of magnitude, and much better preserving the HD neighborhoods than state-of-the-art single-scale techniques. These results are statistically significant over the random HD data sub-samplings, leading to LD embeddings exhibiting a high quality.

This thesis thus generally aims at laying out foundations and steps towards improved HD data exploration, with the final goal of more tightly connecting and bridging the gap between users and ML algorithms in data analysis applications through visualization [Mun14, DPATBEG19]. For this purpose, it introduces neighbor embedding DR algorithms that are both **robust** to missing values in databases, and **fast** enough to conveniently tackle

large-scale data sets, while maintaining the faithfulness and adequacy of the provided LD representations thanks to the multi-scale NE framework.

This thesis dissertation is structured as follows: first, Chapter 2 summarizes some essential aspects of neighbor embedding DR methods by reviewing both single-scale (Section 2.1) and multi-scale (Section 2.2) NE algorithms. Then, Chapter 3 presents the proposed methodology for managing incomplete data set in nonlinear DR [dBMVL19]. Afterwards, Chapter 4 considers fast accelerations of neighbor embedding approaches, by first synthesizing some lower-complexity approximations for single-scale methods (Section 4.1), then experimentally studying their DR quality (Section 4.2) [dBMVL18a], and thereafter introducing the proposed fast multi-scale neighbor embedding schemes (Section 4.3) [dBMVL20]. Finally, Chapter 5 discusses the DR algorithms designed in the context of this thesis, and sketches perspectives for future works and challenges to further enhance HD data exploration in machine learning visualization settings.

Chapter 2

Neighbor embedding algorithms

DIMENSIONALITY reduction (DR) aims to compute LD representations of HD data sets which best preserve their structures. This means that the data points LD neighborhoods should reflect the HD ones. Several paradigms tackled this problem, from the rigid distance preservation methods to the successful neighbor embedding approaches [LV07]. Among these, Stochastic Neighbor Embedding (SNE) first defined the concept of probabilities of being neighbors [HR02], while more recent multi-scale techniques refine these to yield impressive performances [LPOV15]. SNE and some of its variants like t -SNE [vdMH08] are first reviewed in Section 2.1. Their multi-scale extensions are then presented in Section 2.2.

2.1 Single-scale methods

Given a set $\Xi = \{\xi_i\}_{i=1}^N$ of N points lying in a $\mathcal{H}$ -dimensional metric space (HDS), let $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$ refer to its representation in a P -dimensional metric space, termed as the LD space (LDS), with $P \leq \mathcal{H}$. The distances between the i^{th} and j^{th} points are denoted by δ_{ij} and d_{ij} in the HDS and LDS respectively. Starting from them, single-scale HD and LD similarities between pairs of data points are defined by SNE [HR02] for each $i \in \mathcal{I} := \{1, \dots, N\}$

and $j \in \mathcal{I} \setminus \{i\}$,

$$\sigma_{ij\star} = \frac{\exp(-\pi_{i\star} \delta_{ij}^2/2)}{\sum_{k \in \mathcal{I} \setminus \{i\}} \exp(-\pi_{i\star} \delta_{ik}^2/2)} > 0, \quad \sigma_{ii\star} = 0, \quad (2.1)$$

$$s_{ij\star} = \frac{\exp(-p_{i\star} d_{ij}^2/2)}{\sum_{k \in \mathcal{I} \setminus \{i\}} \exp(-p_{i\star} d_{ik}^2/2)} > 0, \quad s_{ii\star} = 0, \quad (2.2)$$

with $\pi_{i\star} > 0$ and $p_{i\star} > 0$ denoting the precisions of the HD and LD Gaussian functions, which are equal to the inverse of their variance. These precisions are determined below in this section. The values of $\sigma_{ii\star}$ and $s_{ii\star}$ are set to zero because SNE is mainly focused on modeling pairwise similarities [vdMH08]. Also, one can easily observe that, for each $i \in \mathcal{I}$,

$$\sum_{j \in \mathcal{I} \setminus \{i\}} \sigma_{ij\star} = 1, \quad (2.3)$$

$$\sum_{j \in \mathcal{I} \setminus \{i\}} s_{ij\star} = 1. \quad (2.4)$$

Therefore, for each $i \in \mathcal{I}$, the similarities gathered in $\boldsymbol{\sigma}_{i\star} := \{\sigma_{ij\star}, j \in \mathcal{I} \setminus \{i\}\}$ and $\boldsymbol{s}_{i\star} := \{s_{ij\star}, j \in \mathcal{I} \setminus \{i\}\}$ define distributions and can be considered as probabilities.

The similarities (2.1) and (2.2) can be shown to remarkably enable neighbor embedding algorithms to circumvent the distance concentration phenomenon [LV11]. This phenomenon is a manifestation of the curse of dimensionality [LV07], which refers to the difficulty to deal with counter-intuitive and unexpected properties of HD spaces. In particular, the norm concentration phenomenon arises from the influence of the HD space dimension Π on the statistical distribution of norms and distances [FWV07]. Indeed, as an example, one can consider HD vectors $\boldsymbol{\xi}_1, \dots, \boldsymbol{\xi}_N$ in $\mathbb{R}^\Pi$ which follow a Gaussian distribution with zero-mean and covariance matrix $\varpi \mathbf{I}_\Pi \in \mathbb{R}^{\Pi \times \Pi}$, where $\mathbf{I}_\Pi$ denotes the identity matrix of order Π and $\varpi > 0$ is a strictly positive scalar. In such a case, for all $i \in \mathcal{I}$, the Euclidean norm $\|\boldsymbol{\xi}_i\|_2 / \varpi = \sqrt{\boldsymbol{\xi}_i^T \boldsymbol{\xi}_i} / \varpi$ of $\boldsymbol{\xi}_i$ normalized by ϖ , where $\boldsymbol{\xi}_i^T$ denotes the transpose of $\boldsymbol{\xi}_i$, has a chi distribution with Π degrees of freedom. Therefore, $\|\boldsymbol{\xi}_i\|_2^2 / \varpi^2$ follow a chi-square distribution with Π degrees of freedom. Its mean is equal to Π , its standard deviation to $\sqrt{2\Pi}$ and its mode is situated at $\max(\Pi - 2, 0)$. For this reason, increasing the dimension Π shifts this mode to larger values without sufficient thickening of the distribution of squared Euclidean norms, since the ratio $\sqrt{2\Pi} / \Pi$ between the standard deviation and the mean tends to 0. This leads squared Euclidean norm to concentrate [FWV07].

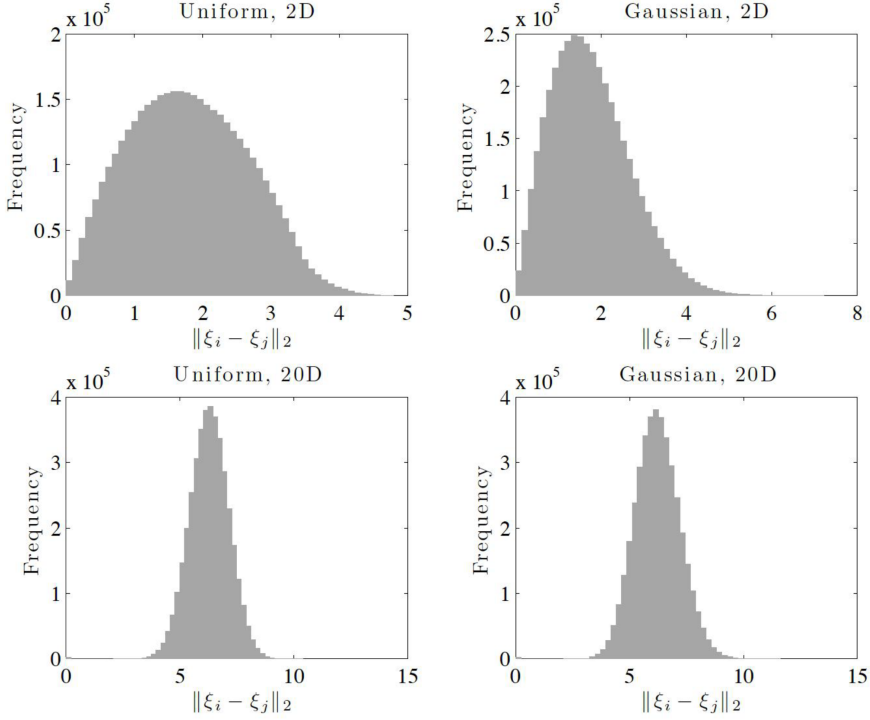


Figure 2.1: Histograms of Euclidean distances between pairs of points in data sets generated using uniform (left) and Gaussian (right) distributions for their samples. The dimensions of the vectors are set to 2 (above) and 20 (below). Figure from the authors of [LV11].

Similarly, if the HD vectors $\xi_1, \dots, \xi_N$ in $\mathbb{R}^H$ independently follow a Gaussian distribution with mean $\mu \in \mathbb{R}^H$ and covariance matrix $\varpi \mathbf{I}_H$, then for all $i \in \mathcal{I}$ and $j \in \mathcal{I} \setminus \{i\}$, $(\xi_i - \xi_j)$ has a Gaussian distribution with zero-mean and covariance matrix $\sqrt{2}\varpi \mathbf{I}_H$. Thus, $\|\xi_i - \xi_j\|_2^2 / (2\varpi^2)$ follow a chi-square distribution with H degrees of freedom. The squared Euclidean distances between pairs of distinct elements hence also concentrate.

Numerous other data distributions depict this concentration phenomenon [FWV07]. It can be visually observed in Fig. 2.1, which displays empirical distributions of pairwise distances in simulated data sets, and in Fig. 2.2, which similarly shows the histogram of pairwise distances in a real data set.

The norm concentration difficulty is an important impediment towards faithful dimensionality reduction and prominently limits the efficiency of distance preserving DR schemes [LV11]. Indeed, DR algorithms involving

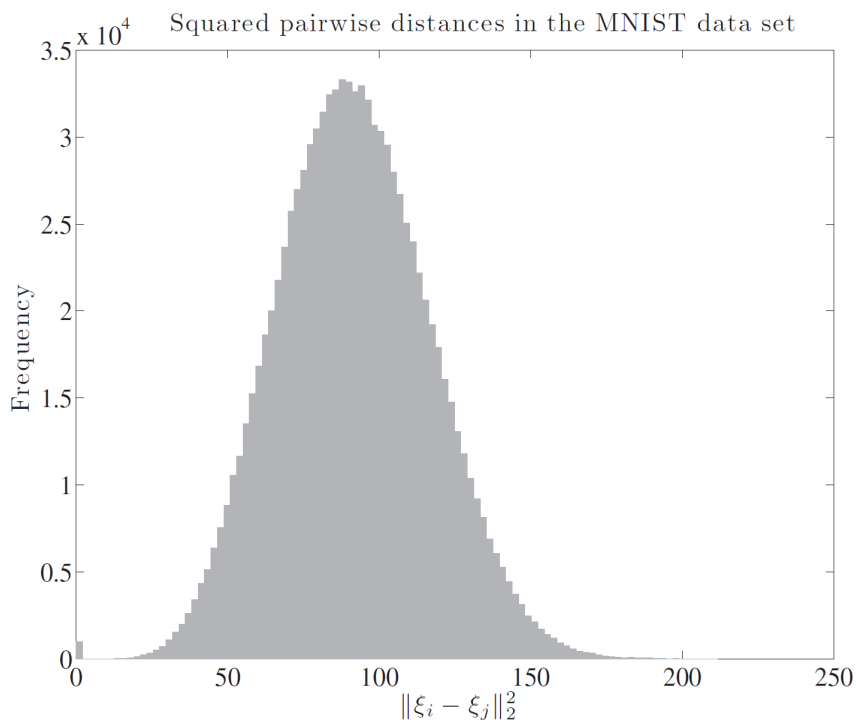


Figure 2.2: Histogram of squared pairwise Euclidean distances for 10^3 points from the MNIST data set of handwritten digit images [LBBH98]. The dimension of this database is equal to 784. Figure from [LV11].

cost functions based on squared differences between HD and LD distances [Sam69, DH97] in fact attempt to directly match distance distributions with distinct shapes in both spaces. The irremediable difference between these distributions, as illustrated in Fig. 2.1, hence greatly affects the performances of these DR methods. Therefore, as the distance concentration is related to the dimensionality, DR techniques must be designed by accounting for the dimensions of the HD and LD spaces. Ideally, one should consider distance transformations that cancel the concentration, independently of the space dimension [LV11].

In this view, the similarities (2.1) elegantly exhibit a shift invariance with respect to squared distances [LV11], as

$$\sigma_{ij\star} = \sigma_{ij\star} \frac{\exp(\gamma_i^2)}{\exp(\gamma_i^2)} = \frac{\exp(\gamma_i^2 - \pi_{i\star} \delta_{ij}^2/2)}{\sum_{k \in \mathcal{I} \setminus \{i\}} \exp(\gamma_i^2 - \pi_{i\star} \delta_{ik}^2/2)}, \quad (2.5)$$

where γ_i is a shift value. Therefore, the similarities (2.1) are not impacted by whether squared distances δ_{ij}^2 or shifted ones $\delta_{ij}^2 - 2\gamma_i^2/\pi_{i\star}$ are considered. One must hence not estimate the shift γ_i value before matching the HD and LD similarities (2.1) and (2.2). This enables to conveniently employ neighbor embedding schemes regardless of the HD dimension $\mathcal{H}$, while alleviating the norm concentration phenomenon [LV11, MHM18].

To determine the HD precision in (2.1), a binary search, also termed as a bisection method, on $\pi_{i\star}$ sets it to obtain a user-provided perplexity $K_\star \in]1, N-1[$, which can be interpreted as a soft neighborhood size [LRB⁺13], for the distribution $\sigma_{i\star}$,

$$\pi_{i\star} \text{ s.t. } \log K_\star = \sum_{j \in \mathcal{I} \setminus \{i\}} \sigma_{ij\star} \log \frac{1}{\sigma_{ij\star}} =: e_{i\star}(\pi_{i\star}) \in]0, \log(N-1)[. \quad (2.6)$$

The star index in $K_\star$ is for consistency with Section 2.2. In practice, only a finite maximum number of iterations is tolerated during the binary search, to avoid infinite loops when it is impossible to tune $\pi_{i\star}$ in order to obtain a perplexity $K_\star$ for $\sigma_{i\star}$, for instance when the HD distances in $\{\delta_{ij}, j \in \mathcal{I} \setminus \{i\}\}$ are all equal. However, it can be noticed that in such a case, $\sigma_{ij\star} = \frac{1}{N-1}$ for all $j \in \mathcal{I} \setminus \{i\}$, independently of the value of the precision $\pi_{i\star}$, as expected since all the HD data points in $\{\xi_j, j \in \mathcal{I} \setminus \{i\}\}$ are equally similar to ξ_i according to the HD distances.

On the other hand, the LD precisions $p_{i\star}$ are fixed to 1. Actually, one can notice that if the LD distances are Euclidean and if the LD precisions $p_{i\star}$ are set to the same value $p_\star > 0$ for all $i \in \mathcal{I}$, then the actual value of $p_\star$ has no importance. Indeed in this case, as $d_{ij}^2 = \mathbf{x}_i^T \mathbf{x}_j$ with $\mathbf{x}_i^T$ referring to the transpose of $\mathbf{x}_i$, the LD precisions in (2.2) only multiply all LD coordinates by a common factor $\sqrt{p_\star}$, which similarly rescales all LD features. As the

visual inspection of a LD space is insensitive to such LD zoom, the value of p_* does not matter. Furthermore, it can be noted that LD distances are typically Euclidean in visualization setting. It indeed corresponds to the natural distance for the visual analysis of a LD embedding.

It is noteworthy that the right-hand side of (2.6) is the “entropy” of the HD similarities in $\{\sigma_{ij*}, j \in \mathcal{I} \setminus \{i\}\}$, denoted by e_{i*} for convenience. It is a function of the HD precision π_{i*} . One can show that if $\pi_{i*} > 0$ and if the HD distances in $\{\delta_{ij}, j \in \mathcal{I} \setminus \{i\}\}$ are not all equal, then e_{i*} is a monotonically decreasing function of π_{i*} , equal to $\log(N-1)$ for $\pi_{i*} \rightarrow 0$ and to 0 for $\pi_{i*} \rightarrow +\infty$ [VCP13]. Thanks to the chain rule, the derivative of e_{i*} with respect to π_{i*} can indeed be computed as

$$\frac{de_{i*}}{d\pi_{i*}}(\pi_{i*}) = -\frac{1}{2} \sum_{j \in \mathcal{I} \setminus \{i\}} (1 + \log \sigma_{ij*}) \sigma_{ij*} \left(-\delta_{ij}^2 + \sum_{k \in \mathcal{I} \setminus \{i\}} \delta_{ik}^2 \sigma_{ik*} \right) \quad (2.7)$$

$$= \frac{1}{2} \sum_{j \in \mathcal{I} \setminus \{i\}} \log(\sigma_{ij*}) \sigma_{ij*} \left(\delta_{ij}^2 - \sum_{k \in \mathcal{I} \setminus \{i\}} \delta_{ik}^2 \sigma_{ik*} \right) \quad (2.8)$$

$$= \frac{-\pi_{i*}}{4} \left(\sum_{j \in \mathcal{I} \setminus \{i\}} \delta_{ij}^4 \sigma_{ij*} - \left(\sum_{j \in \mathcal{I} \setminus \{i\}} \delta_{ij}^2 \sigma_{ij*} \right)^2 \right) \quad (2.9)$$

$$= \frac{-\pi_{i*}}{4} \sum_{j \in \mathcal{I} \setminus \{i\}} \sigma_{ij*} \left(\delta_{ij}^2 - \sum_{k \in \mathcal{I} \setminus \{i\}} \delta_{ik}^2 \sigma_{ik*} \right)^2. \quad (2.10)$$

Therefore, as $\sigma_{ij*} > 0$ and $\pi_{i*} > 0$, $\frac{de_{i*}}{d\pi_{i*}}(\pi_{i*}) < 0$ if the HD distances in $\{\delta_{ij}, j \in \mathcal{I} \setminus \{i\}\}$ are not all equal, and $\frac{de_{i*}}{d\pi_{i*}}(\pi_{i*}) = 0$ if these distances are all equal. Thus, if the distances in $\{\delta_{ij}, j \in \mathcal{I} \setminus \{i\}\}$ are not all equal, e_{i*} is monotonically decreasing with π_{i*} . For this reason, the larger K_* , the smaller π_{i*} , according to (2.6). Thanks to these observations, some studies tackle the root-finding problem (2.6) by replacing or combining the binary search with improved, derivative-based algorithms such as Newton’s method [VCP13, LRB⁺13].

The soft Gaussian neighborhood size in (2.1) is quantified by the perplexity, meaning that (2.1) and (2.2) can be interpreted as probabilities for ξ_j and $\mathbf{x}_j$ to be neighbors of ξ_i and $\mathbf{x}_i$, respectively in the HD and LD spaces [HR02]. To compute $\mathbf{X}$, the sum over $i \in \mathcal{I}$ of the Kullback-Leibler (KL) divergences between the distributions σ_{i*} and s_{i*} ,

$$C_* = \sum_{i \in \mathcal{I}, j \in \mathcal{I} \setminus \{i\}} \sigma_{ij*} \log \left(\frac{\sigma_{ij*}}{s_{ij*}} \right), \quad (2.11)$$

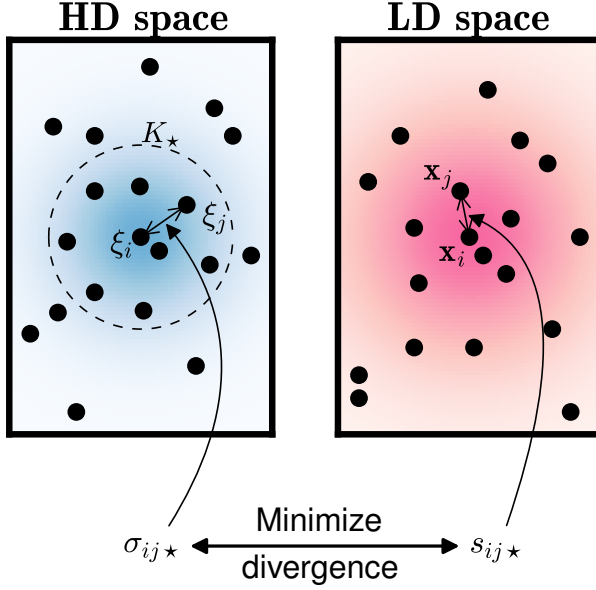


Figure 2.3: Illustration of the main principles of the SNE algorithm.

is minimized by SNE through gradient-based optimization algorithms [HR02]. The gradient of (2.11) with respect to the LD coordinates can be expressed as

$$\frac{\partial C_*}{\partial \mathbf{x}_v} = \sum_{i \in \mathcal{I} \setminus \{v\}} (p_{v*} (\sigma_{vi*} - s_{vi*}) + p_{i*} (\sigma_{iv*} - s_{iv*})) (\mathbf{x}_v - \mathbf{x}_i), \quad (2.12)$$

assuming that the LD distances are Euclidean. Since these LD distances are natural in visualization contexts, the gradients of NE cost functions are all computed using Euclidean LD distances in this thesis dissertation. In particular, these gradients are developed hereunder to ease the comparison between NE algorithms at the end of this section. Figure 2.3 illustrates the application of SNE to compute a LD representation of a HD data set.

Besides symmetrizing SNE HD similarities, *t*-SNE employs a Student *t*-distribution with one degree of freedom in the LDS, circumventing the crowding problem [vdMH08], which refers to the difficulty to reproduce data similarities from large HD volumes to exponentially smaller LD ones:

for $i \in \mathcal{I}$, $j \in \mathcal{I} \setminus \{i\}$,

$$\tau_{ij\star} = \frac{\sigma_{ij\star} + \sigma_{ji\star}}{2N}, \quad \tau_{ii\star} = 0, \quad (2.13)$$

$$t_{ij} = \frac{(1 + d_{ij}^2)^{-1}}{\sum_{k \in \mathcal{I}, l \in \mathcal{I} \setminus \{k\}} (1 + d_{kl}^2)^{-1}}, \quad t_{ii} = 0. \quad (2.14)$$

The t -SNE cost function remains as in SNE,

$$C_{t\star} = \sum_{i \in \mathcal{I}, j \in \mathcal{I} \setminus \{i\}} \tau_{ij\star} \log \left(\frac{\tau_{ij\star}}{t_{ij}} \right). \quad (2.15)$$

Its gradient with respect to the LD coordinates develops as

$$\frac{\partial C_{t\star}}{\partial \mathbf{x}_v} = 4 \sum_{i \in \mathcal{I} \setminus \{v\}} (\tau_{vi\star} - t_{vi}) (\mathbf{x}_v - \mathbf{x}_i) (1 + d_{vi}^2)^{-1}. \quad (2.16)$$

Other variants of SNE were defined by modifying its cost function using mixtures of divergences [BHBV12], such as the Neighborhood Retrieval Visualizer (NeRV) [VPN⁺10] and Jensen–Shannon embedding (JSE) [LRB⁺13].

On the one hand, NeRV is based on the same HD similarities as SNE, but slightly different LD ones [VPN⁺10]: for all $i \in \mathcal{I}$, $j \in \mathcal{I} \setminus \{i\}$,

$$\sigma_{ij\star}^n = \sigma_{ij\star}, \quad \sigma_{ii\star}^n = s_{ii\star}^n = 0, \quad (2.17)$$

$$s_{ij\star}^n = \frac{\exp(-\pi_{i\star} d_{ij}^2/2)}{\sum_{k \in \mathcal{I} \setminus \{i\}} \exp(-\pi_{i\star} d_{ik}^2/2)}. \quad (2.18)$$

The LD precisions are hence set to the HD ones. Moreover, NeRV minimizes a mixture of KL divergences,

$$C_{n\star} = \sum_{i \in \mathcal{I}, j \in \mathcal{I} \setminus \{i\}} (1 - \kappa_n) \sigma_{ij\star}^n \log \left(\frac{\sigma_{ij\star}^n}{s_{ij\star}^n} \right) + \kappa_n s_{ij\star}^n \log \left(\frac{s_{ij\star}^n}{\sigma_{ij\star}^n} \right), \quad (2.19)$$

where $\kappa_n \in [0, 1]$ is a meta-parameter trading off the different terms in the divergences. This blending of two dual KL divergences is termed as a type 1 mixture [CA10]. Mixing dual objective functions is indeed a reminiscent idea already investigated by some DR techniques aiming to directly reproduce pairwise distances [VK05]. The gradient of (2.19) with respect to the LD

points can be computed as

$$\begin{aligned}
\frac{\partial C_{n^*}}{\partial \mathbf{x}_v} = & \sum_{i \in \mathcal{I} \setminus \{v\}} (\mathbf{x}_v - \mathbf{x}_i) \left(\pi_{v^*} (1 - \kappa_n) (\sigma_{vi^*}^n - s_{vi^*}^n) \right. \\
& + \pi_{v^*} \kappa_n s_{vi^*}^n \log \left(\frac{\sigma_{vi^*}^n}{s_{vi^*}^n} \right) \\
& + \pi_{v^*} \kappa_n s_{vi^*}^n \left[\sum_{j \in \mathcal{I} \setminus \{v\}} s_{vj^*}^n \log \left(\frac{s_{vj^*}^n}{\sigma_{vj^*}^n} \right) \right] + \\
& \pi_{i^*} \left[(1 - \kappa_n) (\sigma_{iv^*}^n - s_{iv^*}^n) + \kappa_n s_{iv^*}^n \log \left(\frac{\sigma_{iv^*}^n}{s_{iv^*}^n} \right) \right] \\
& \left. + \pi_{i^*} \kappa_n s_{iv^*}^n \left[\sum_{j \in \mathcal{I} \setminus \{i\}} s_{ij^*}^n \log \left(\frac{s_{ij^*}^n}{\sigma_{ij^*}^n} \right) \right] \right). \quad (2.20)
\end{aligned}$$

Interestingly, NeRV can be interpreted as an algorithm producing an optimal LD embedding with respect to an information retrieval task given the LD visualization [VPN⁺10]. This task aims at finding similar HD data items based on the proximity relationships observed on the LD display. The KL divergences involved in (2.19) can respectively be related to smooth measures of recall and precision of the retrieved LD neighborhoods around the data points in the context of this information retrieval task [VPN⁺10]. The cost function (2.19) balances these two performance measures, and the meta-parameter κ_n enables the user to specify the relative cost of missing similarity relationships in the LD embedding versus depicting dissimilar HD points close one to another in the LD space.

On the other hand, JSE relies on the HD and LD similarities of SNE, but finds the LD coordinates by minimizing another sum of KL divergences,

$$C_{j^*} = \sum_{i \in \mathcal{I}, j \in \mathcal{I} \setminus \{i\}} \frac{1}{1 - \kappa_j} \sigma_{ij^*} \log \left(\frac{\sigma_{ij^*}}{z_{ij^*}} \right) + \frac{1}{\kappa_j} s_{ij^*} \log \left(\frac{s_{ij^*}}{z_{ij^*}} \right), \quad (2.21)$$

where $z_{ij^*} = \kappa_j \sigma_{ij^*} + (1 - \kappa_j) s_{ij^*}$, with $\kappa_j \in]0, 1[$ being again a meta-parameter weighting both terms in the divergence. The combination (2.21) of KL divergences is termed as the type 2 symmetric KL divergence [CA10], or the symmetric generalized Jensen-Shannon divergence [BR82, Lin91]. The gradient of (2.21) with respect to the LD points in $\mathbf{X}$ can be expressed

as

$$\begin{aligned} \frac{\partial \mathcal{C}_{j\star}}{\partial \mathbf{x}_v} = & \sum_{i \in \mathcal{I} \setminus \{v\}} (\mathbf{x}_v - \mathbf{x}_i) \left(\frac{p_{v\star} s_{vi\star}}{\kappa_j} \left[\log \left(\frac{z_{vi\star}}{s_{vi\star}} \right) - \sum_{j \in \mathcal{I} \setminus \{v\}} s_{vj\star} \log \left(\frac{z_{vj\star}}{s_{vj\star}} \right) \right] \right. \\ & \left. + \frac{p_{i\star} s_{iv\star}}{\kappa_j} \left[\log \left(\frac{z_{iv\star}}{s_{iv\star}} \right) - \sum_{j \in \mathcal{I} \setminus \{i\}} s_{ij\star} \log \left(\frac{z_{ij\star}}{s_{ij\star}} \right) \right] \right). \end{aligned} \quad (2.22)$$

It is noteworthy that the gradient of (2.21) involves a modulation factor, equal to $s_{vi\star}$ and $s_{iv\star}$ in the terms of (2.22), which decreases as the LD distances grow [LRB⁺13]. Such a factor is deemed to be desirable in nonlinear DR methods. It indeed enables DR algorithms to tear HD manifolds in LD spaces, as previously observed in the context of DR schemes from the distance preservation paradigm [LLV04, VK06]. This property explains the superiority of Demartines’ curvilinear component analysis [DH97] with respect to Sammon’s nonlinear mapping [Sam69]. It also justifies why t -SNE [vdMH08] typically outperforms SNE [HR02] in practice, since the gradient (2.16) contains a similar modulation factor at the opposite of the gradient (2.12). The gradient (2.20) of the NeRV cost function (2.19) is only partially modulated, by a factor which depends on the value of κ_n . NeRV can hence be expected to exhibit a behavior between SNE and t -SNE. Eventually, both t -SNE and JSE benefit from a full modulation in their gradients.

Meanwhile, t -SNE relies on different neighborhood distributions in the HD and LD spaces, namely, Gauss (2.13) versus Student (2.14). For very high-dimensional data, the intuition of a crowding problem [vdMH08], related to the curse of dimensionality [LV07], motivates tails that are heavier in LD than in HD. Mid-range to distant neighbors are then repelled further away in the LD embedding, which compensates for the relative lack of volume in LD spaces compared to HD ones. This approach tends, however, to over-emphasize clusters or to be too extreme for data with low intrinsic dimensionality [dbMVL18b]. Adapting the degrees of freedom of the Student function (2.14) has been a possible workaround [Maa09].

In this view, JSE is superior to t -SNE as it is based on similarities with the same distributions in both the HD and LD spaces, while involving a modulation factor in the gradient of (2.21) with respect to the LD points. This tends to elucidate why JSE is reported as delivering either competitive or improved experimental performances compared to t -SNE, and that it is efficiently applicable in a broader variety of cases [LRB⁺13].

To illustrate these concepts, one can consider a 3-D data set with $N = 1000$ points uniformly sampled from the surface of a sphere, as depicted in Fig. 2.4a. Starting from this space with $\mathcal{I} = 3$ dimensions, embeddings

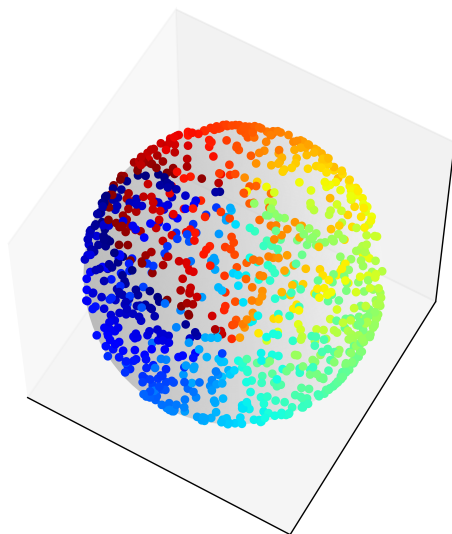
with $P = 2$ dimensions are desired. It is noteworthy that the colors of the data points in Fig. 2.4a are not employed by the DR methods; they only enable linking corresponding samples on the 2-D embeddings and the 3-D sphere in Fig. 2.4a. Figure 2.4b sketches the results of the PCA projection; one can observe that the sphere is crushed, with two hemispheres that are superimposed. A linear mapping of a sphere indeed necessarily leads to such a superimposition, which highlights the limitations of linear DR schemes for data visualization. One would instead wish to obtain a kind of planisphere [LRB⁺13], revealing the local 2-D structure of this spherical shell manifold.

The embeddings provided by SNE, t -SNE, NeRV and JSE are presented respectively in Fig. 2.5a, 2.5b, 2.6a and 2.6b, using a perplexity $K_\star = 30$, $\kappa_n = 0.5$ and $\kappa_j = 0.5$. All algorithms are optimized following suggestions from the literature, using gradient-based procedures from momentum gradient descent [Jac88] for SNE [HR02] and t -SNE [vdMH08] to the limited-memory version of Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) technique [BLNZ95] for NeRV [VPN⁺10] and JSE [LRB⁺13]. NeRV and JSE appear to perform best, by cutting the sphere to unfold it from the 3-D to the 2-D space. On the other hand, SNE seems to be unable to as nicely tear the manifold, while t -SNE even artificially introduces clusters in the embedding. This behavior results from the different similarity distributions employed by t -SNE in the HD and LD spaces, as previously mentioned, and exemplifies the inadequacy of t -SNE for data with low intrinsic dimensionality, such as the sphere. Nevertheless, t -SNE reaches impressive performances in practice on very HD data exhibiting clusters, as reported in the literature [vdMH08] and illustrated in Chapter 4. For this reason, it is extensively used as a powerful data visualization technique [KB19, BCA⁺19].

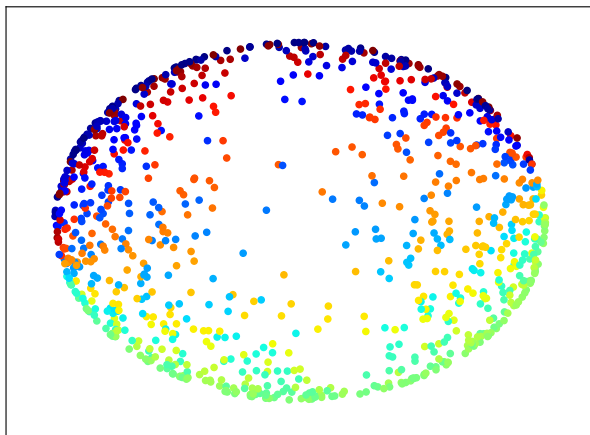
It should be finally mentioned that, for all these SNE-based methods, evaluating their cost functions or their gradient scales in $\mathcal{O}(N^2)$, since it requires computing all $\frac{N(N-1)}{2}$ similarities. Such a time complexity prevents directly tackling databases containing more than a few thousands samples.

2.2 Multi-scale methods

Small perplexities $K_\star$ are typically used in SNE, inducing local transformations of the HD distances in (2.1). It thus mainly focuses on the preservation of small neighborhoods. Indeed, most HD similarities are close to zero, which annihilates the influence of the associated LD similarities on the SNE cost function. Relying on such local distance transformations may hence be questioned as it neglects and prevents considering the average- and large-scale interactions while designing the global structure of the embedding, $K_\star$ being most of the time arbitrary. The preservation of long-range neighborhoods is therefore considerably affected. On the other hand, employing too



(a) Sphere (3-D).



(b) PCA embedding (2-D).

Figure 2.4: (a) Data set consisting in 1000 points uniformly sampled from the surface of a 3-D sphere. The colors associated with the data points are ignored by the DR algorithms and enable identifying corresponding samples between the 2-D embeddings of the sphere and its 3-D version. (b) 2-D embedding of the data set displayed in (a) obtained using PCA.

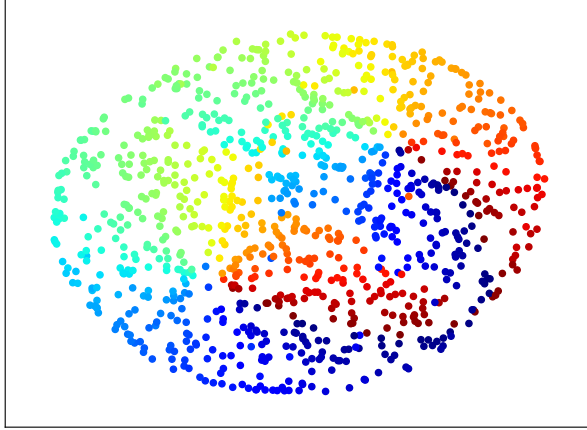
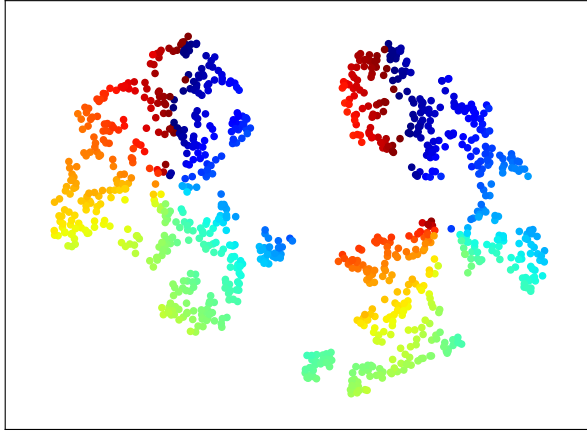
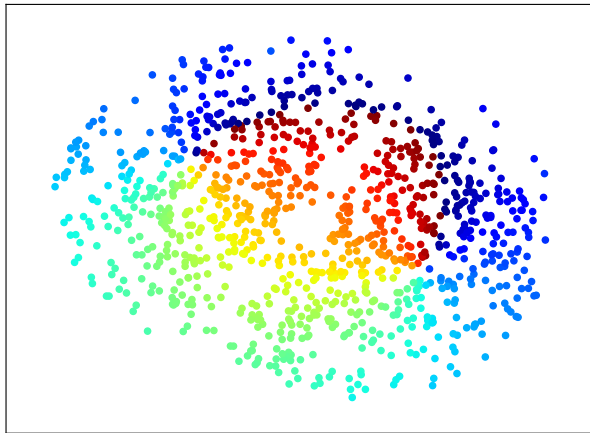
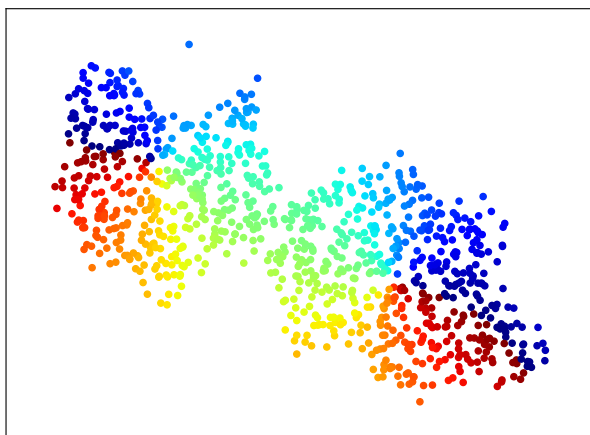
(a) SNE embedding (2-D, $K_{\star} = 30$).(b) t -SNE embedding (2-D, $K_{\star} = 30$).

Figure 2.5: (a) 2-D embedding of the data set displayed in Fig. 2.4a obtained using SNE, with a perplexity $K_{\star} = 30$. (b) 2-D embedding of the data set displayed in Fig. 2.4a obtained using t -SNE, with a perplexity $K_{\star} = 30$.



(a) NeRV embedding (2-D, $K_{\star} = 30$, $\kappa_n = 0.5$).



(b) JSE embedding (2-D, $K_{\star} = 30$, $\kappa_j = 0.5$).

Figure 2.6: (a) 2-D embedding of the data set displayed in Fig. 2.4a obtained using NeRV, with a perplexity $K_{\star} = 30$ and $\kappa_n = 0.5$. (b) 2-D embedding of the data set displayed in Fig. 2.4a obtained using JSE, with a perplexity $K_{\star} = 30$ and $\kappa_j = 0.5$.

large a perplexity leads to nearly-uniform HD similarities, impairing the preservation of small neighborhoods.

However, combining several SNE similarities defined using increasing perplexities enables to capture both local and global relationships and to focus on various data scales at the same time. This reasoning leads to the multi-scale SNE method [LPOV15]. By introducing an additional scale index h , the single-scale similarities are now defined, for $i \in \mathcal{I}$ and $j \in \mathcal{I} \setminus \{i\}$, as

$$\sigma_{ijh} = \frac{\exp(-\pi_{ih}\delta_{ij}^2/2)}{\sum_{k \in \mathcal{I} \setminus \{i\}} \exp(-\pi_{ih}\delta_{ik}^2/2)}, \quad \sigma_{iih} = 0, \quad (2.23)$$

$$s_{ijh} = \frac{\exp(-p_{ih}d_{ij}^2/2)}{\sum_{k \in \mathcal{I} \setminus \{i\}} \exp(-p_{ih}d_{ik}^2/2)}, \quad s_{iih} = 0. \quad (2.24)$$

Multi-scale similarities then average the single-scale ones as

$$\sigma_{ij} = \frac{1}{H} \sum_{h=1}^H \sigma_{ijh}, \quad s_{ij} = \frac{1}{H} \sum_{h=1}^H s_{ijh}, \quad (2.25)$$

where the considered number of scales is denoted by H .

In order to determine the precisions π_{ih} and p_{ih} for the neighborhoods around each $\mathbf{x}_i$ and $\mathbf{x}_i$, one can rely on perplexity values covering the whole data range from 1 to N , selected in a data-driven way without any user intervention. As suggested in [LPOV15], indexed perplexities are introduced as $K_h = 2^{h-1}K_1$, with a small base perplexity like $K_1 = 2$ and $1 \leq h \leq H = \lfloor \log_2(N/K_1) \rfloor$, where $\lfloor \cdot \rfloor$ represents the rounding operator. Precision π_{ih} is then computed as in SNE, using perplexity K_h . The entropy of the HD similarity distributions hence linearly increases across scales. On the other hand, the LD precisions should be determined according to the exponential evolution of the perplexities [LPOV15]. One possibility is $p_{ih} = p_h = K_h^{-2/P}$. Further details for the choice of K_h and p_h can be found in [LPOV15].

The cost function of multi-scale SNE remains as in SNE. Since the entropy of the HD similarities does not depend on $\mathbf{X}$, it develops as

$$C = - \sum_{i \in \mathcal{I}, j \in \mathcal{I} \setminus \{i\}} \sigma_{ij} \log s_{ij}. \quad (2.26)$$

In comparison with Fig. 2.3, Fig. 2.7 depicts the computation of LD coordinates for a bunch of HD data points using multi-scale SNE.

On the other hand, multi-scale t -SNE [dBMVL18b] minimizes

$$C_t = \sum_{i \in \mathcal{I}, j \in \mathcal{I} \setminus \{i\}} \tau_{ij} \log \frac{1}{t_{ij}}, \quad \text{with } \tau_{ij} = \frac{\sigma_{ij} + \sigma_{ji}}{2N}. \quad (2.27)$$

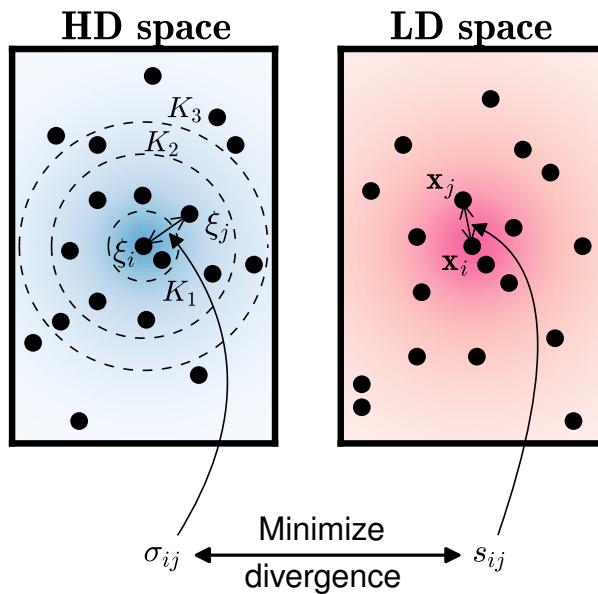


Figure 2.7: Illustration of the main principles of the multi-scale SNE algorithm. In this case, the number H of scales is fixed to 3.

Using such compound HD Gaussian neighborhoods with exponentially growing perplexities in t -SNE leads to multi-scale similarity distributions with intermediate tails, between Gauss and Student [dBMVL18b].

Multi-scale NeRV and JSE can also be defined [LPOV15], by replacing their single-scale similarities with multi-scale ones. Namely,

$$C_n = \sum_{i \in \mathcal{I}, j \in \mathcal{I} \setminus \{i\}} (1 - \kappa_n) \sigma_{ij} \log \left(\frac{\sigma_{ij}}{s_{ij}} \right) + \kappa_n s_{ij} \log \left(\frac{s_{ij}}{\sigma_{ij}} \right) \quad (2.28)$$

is the cost function of multi-scale NeRV. As to multi-scale JSE, its cost function develops as

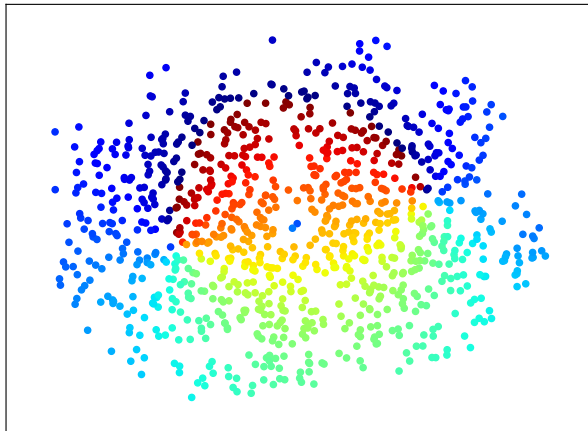
$$C_j = \sum_{i \in \mathcal{I}, j \in \mathcal{I} \setminus \{i\}} \frac{1}{1 - \kappa_j} \sigma_{ij} \log \left(\frac{\sigma_{ij}}{z_{ij}} \right) + \frac{1}{\kappa_j} s_{ij} \log \left(\frac{s_{ij}}{z_{ij}} \right), \quad (2.29)$$

with $z_{ij} = \kappa_j \sigma_{ij} + (1 - \kappa_j) s_{ij}$.

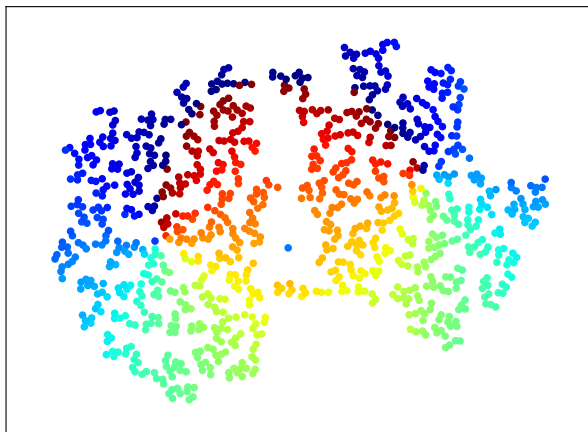
The optimization of SNE-like methods turns out to be difficult [HR02, VPN⁺10, CP10, VCP12], especially when small perplexities are considered since they induce high-frequency oscillations in the cost functions [LPOV15]. Heuristics to attempt avoiding shallow local minima are usually helpful and typically recommended [BCA⁺19]. They include early exaggeration [vdMH08] and multi-scale optimization [LPOV15]. The latter starts the minimization by first only employing the largest scale and then progressively introduces the smaller ones, one at a time.

As there are $\mathcal{O}(\log N)$ scales, evaluating the multi-scale similarities, hence the multi-scale NE cost functions and gradients, costs $\mathcal{O}(N^2 \log N)$, which is then the time complexity of the multi-scale NE schemes. Whereas single-scale algorithms succeed best in reproducing neighborhoods with sizes close to or smaller than K_* but poorly preserve larger ones, broader ranges of neighborhoods are systematically spanned and retrieved by the parameter-free multi-scale approaches. This behavior enables retaining both local and global data structures in the data [LPOV15], while relieving the user from adjusting the perplexity K_* , which is the main meta-parameter in neighbor embedding algorithms [dRBP19].

Figures 2.8a, 2.8b, 2.9a and 2.9b respectively present the 2-D embeddings of the data set displayed in Fig. 2.4a computed by multi-scale SNE, multi-scale t -SNE, multi-scale NeRV and multi-scale JSE, with $\kappa_n = \kappa_j = 0.5$. Compared to the results of single-scale methods depicted in Fig. 2.5a, 2.5b, 2.6a and 2.6b, multi-scale similarities improve the planisphere representation. In particular, they clearly attenuate the cluster magnification behavior of t -SNE in Fig. 2.8b with respect to Fig. 2.5b.

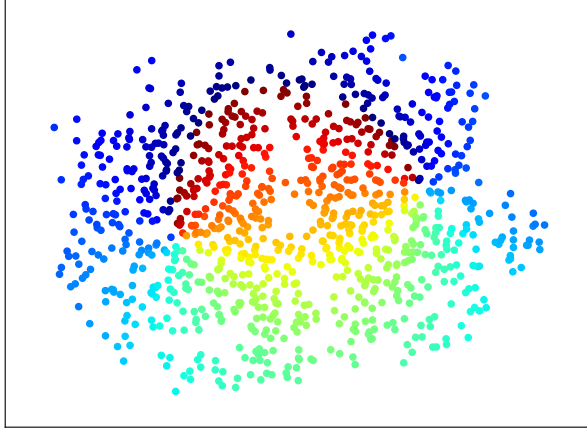


(a) Multi-scale SNE embedding (2-D).

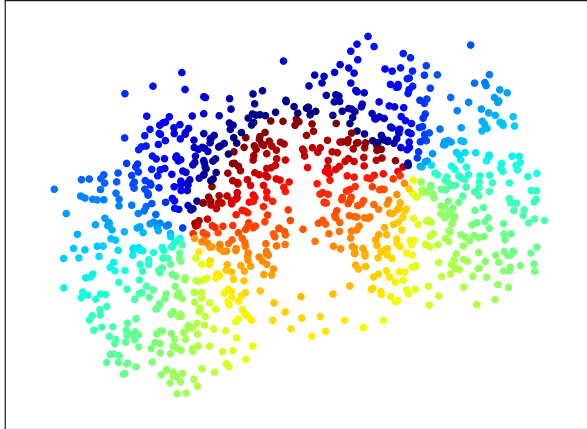


(b) Multi-scale t -SNE embedding (2-D).

Figure 2.8: (a) 2-D embedding of the data set displayed in Fig. 2.4a obtained using multi-scale SNE. (b) 2-D embedding of the data set displayed in Fig. 2.4a obtained using multi-scale t -SNE.



(a) Multi-scale NeRV embedding (2-D, $\kappa_n = 0.5$).



(b) Multi-scale JSE embedding (2-D, $\kappa_j = 0.5$).

Figure 2.9: (a) 2-D embedding of the data set displayed in Fig. 2.4a obtained using multi-scale NeRV, with $\kappa_n = 0.5$. (b) 2-D embedding of the data set displayed in Fig. 2.4a obtained using multi-scale JSE, with $\kappa_j = 0.5$.

Chapter 3

Nonlinear dimensionality reduction with missing data using parametric multiple imputations

INCOMPLETE databases become nowadays more and more frequently encountered in machine learning and data analysis [GLSGFVV09, DV12, BE10, All02]. While missing values occur for many different reasons depending on the data type, any inference task on such incomplete data set should take the missing entries into account in order to avoid biased conclusions [EDVL13, ELVB14]. Discarding the samples with missing features is most often not acceptable as it results in considerable information losses and drastically reduces the data set size as the proportion of missing values increases [SEM⁺16]. Even more importantly in our context, removing these examples would prevent the computation of their LD representations, which greatly alters the SNE-based probabilities of the complete samples and hence their LD positioning.

Quite surprisingly, despite all the recent advances in DR, the field has not yet designed methods directly able to deal with incomplete data sets. In this context, the cost functions optimized by nonlinear DR schemes cannot be directly evaluated as they depend on the incomplete HD samples coordinates. Although it has strong limitations for reasons detailed below, a workaround is to consider a complete HD data set instead, resulting from some imputation. The missing data management is hence considered as a preprocessing step. The imputation may attempt to be consistent with

the data distribution in the feature space [VBBGOR06]. As the features of most data sets are not independent, the conditional distribution of the missing features of the samples given their observed features should indeed be considered in order to derive meaningful insights on the data relationships. This can be accounted for by replacing the missing data with their conditional mean given the observed values, provided that the data distribution can be modeled [YME⁺13]. Nearest neighbor-based imputation is another possibility [HHE03, VHK14, WM05]. In all these cases however, the DR method would at best yield the LD embedding minimizing the cost function evaluated using the expected HD data set. Denoting by Ξ_m a set of incomplete HD data points and by $\mathbf{X}$ the set of their LD representations, the LD embedding would be

$$\arg \min_{\mathbf{X}} C_{\text{DR}}(\mathbf{X}, \mathbb{E}[\Xi_m]), \quad (3.1)$$

where C_{DR} is the cost function of the DR algorithm, and where the expectation is taken over the missing entries in Ξ_m . Missing values can indeed be considered as random variables following the data distribution in the HD space. In practice however, one is rather interested in obtaining the LD embedding minimizing the cost function expectation, yielding the closest cost function value on average to the one that would be obtained in the complete data case:

$$\arg \min_{\mathbf{X}} \mathbb{E}[C_{\text{DR}}(\mathbf{X}, \Xi_m)]. \quad (3.2)$$

This strongly differs from (3.1) when C_{DR} is nonlinear in Ξ_m as in most recent, state-of-the-art, DR methods which are based on nonlinear transformations of the HD samples, such as SNE similarities (2.1), etc. Rather than (3.2), one could suggest to seek for

$$\mathbb{E} \left[\arg \min_{\mathbf{X}} C_{\text{DR}}(\mathbf{X}, \Xi_m) \right]. \quad (3.3)$$

This would however merge LD embeddings specifically designed for different complete HD data sets. As $\mathbf{X}$ is subject to rotations, translations and scaling factors, the obtained LD embedding would not reflect any kind of expected relationship in Ξ_m . More interestingly, one could create an imputed HD data set for each mode of the conditional distribution of the missing values given the observed data and compute their corresponding LD embeddings. Their number would nevertheless rapidly increase with the proportion of missing values. This would result in a huge amount of DR results to analyze and compare for the user, which greatly complicates its exploratory task. This chapter hence focuses on recovering a single relevant LD embedding.

Solving (3.2) requires reliably approximating the cost function expectation on the incomplete database. Whereas considering a single imputed data

set would yield a biased estimate for nonlinear cost functions, a consistent approximation can be obtained under the multiple imputation paradigm [Rub04, LR14], as long as the conditional distribution of the missing features given the observed ones can be modeled [End10]. Gaussian mixture models are particularly attractive for this purpose, as they can approximate arbitrarily well almost any continuous density [Bis06, TSM85]. Their parameters are usually derived through maximum likelihood estimation [DLR77], which can be easily adapted when facing missing values [GJ95, HJ03]. The fitting of Gaussian mixtures being notoriously difficult in very HD spaces, efficient HD extensions exist [BGS07], considerably reducing their number of parameters. Further improvements [LLH06, DCB12] also allow for the application of these models in several fields [SWR10, BMK05, MCG98]. In particular, they are successfully used for distance estimation in numerical data sets with missing values [EDVL13, ELVB14] and adapted extreme learning machines on incomplete data sets [YME⁺13, SEM⁺16].

This chapter proposes to process missing data in DR under the multiple imputation paradigm [dBMVL19]. In particular, a high-dimensional extension of Gaussian mixture models is first fitted on the incomplete database. Several imputed versions of the data set are then created by sampling the conditional distribution of the missing features of the samples given their observed features. Averaging the nonlinear DR cost function evaluations on each complete data set finally approximates its expectation on the incomplete database, enabling to optimize (3.2). This provides a meaningful LD representation of each data point, no matter whether it contains missing values or not. Although this chapter considers multi-scale SNE [LPOV15], the proposed methodology is directly applicable with any other neighbor embedding DR method, and more generally with any DR scheme optimizing some cost function. Free implementations of the presented approach are publicly available online at <https://github.com/cdebodt>, enabling easy usage of the methodology.

This chapter is structured as follows: Section 3.1 reviews Gaussian mixture models and describes their missing data extension as well as a HD variant. This section is strongly inspired from [EDVL13, ELVB14, YME⁺13, SEM⁺16]. The complete multiple imputation methodology for nonlinear DR with missing data is derived in Section 3.2. Its application using multi-scale SNE and other neighbor embedding DR schemes is detailed. Section 3.3 is dedicated to experimental results comparing different methods dealing with missing data on several data sets in which missing values are introduced. Some conclusions are finally drawn in Section 3.4.

3.1 Gaussian mixture models for HD and incomplete data

A G -component Gaussian mixture model of a complete data set $\Xi := \{\xi_i\}_{i=1}^N$ with N independent examples $\xi_i \in \mathbb{R}^\Pi$ consists in G multivariate Gaussians with mean vectors $\mu_1, \dots, \mu_G \in \mathbb{R}^\Pi$, covariance matrices $\Sigma_1, \dots, \Sigma_G \in \mathbb{R}^{\Pi \times \Pi}$ and positive weight coefficients $w_1, \dots, w_G \in \mathbb{R}$ summing to 1, which maximize the model log-likelihood

$$\log \mathcal{L}(\vartheta) = \log \ell(\Xi | \vartheta) = \sum_{i \in \mathcal{I}} \log \left(\sum_{k \in \mathcal{G}} w_k \mathcal{N}(\xi_i | \mu_k, \Sigma_k) \right), \quad (3.4)$$

where $\mathcal{I} := \{1, \dots, N\}$, $\mathcal{G} := \{1, \dots, G\}$, $\vartheta := \{w_k, \mu_k, \Sigma_k; k \in \mathcal{G}\}$ and where $\mathcal{N}(\xi_i | \mu_k, \Sigma_k)$ denotes the evaluation of the probability density function of the k^{th} Gaussian over ξ_i [Bis06, DLR77]. A traditional Expectation-Maximization (EM) algorithm allows maximizing (3.4). It is detailed below for the sake of introducing relevant notations.

After initializing the mean vectors, covariance matrices and weight coefficients, the E-step estimates the probabilities for each example to originate from each Gaussian given the current value of ϑ : for each $i \in \mathcal{I}$ and $k \in \mathcal{G}$,

$$\varphi_{ik} = \frac{w_k \mathcal{N}(\xi_i | \mu_k, \Sigma_k)}{\sum_{j \in \mathcal{G}} w_j \mathcal{N}(\xi_i | \mu_j, \Sigma_j)}, \quad (3.5)$$

$$N_k = \sum_{i \in \mathcal{I}} \varphi_{ik}. \quad (3.6)$$

The M-step then updates ϑ given these probabilities:

$$\mu_k = \frac{1}{N_k} \sum_{i \in \mathcal{I}} \varphi_{ik} \xi_i, \quad (3.7)$$

$$w_k = \frac{N_k}{N}, \quad (3.8)$$

$$\Sigma_k = \frac{1}{N_k} \sum_{i \in \mathcal{I}} \varphi_{ik} (\xi_i - \mu_k) (\xi_i - \mu_k)^T. \quad (3.9)$$

These two steps are iterated until convergence of (3.4) [ELVB14, SEM⁺16].

3.1.1 Missing data management

Different studies extended the Gaussian mixture model to manage missing data [GJ95, HJ03]. Missing values refer to well-defined data entries, unavailable for an unknown reason [LR14]. These missing data entries are

identified, meaning that one knows which coordinates of the samples are observed and missing. Let $\Xi_m = \{\xi_i, O_i\}_{i=1}^N$ denote an incomplete data set with N independent samples $\xi_i \in \mathbb{R}^{\Pi}$, $O_i \subset \{1, \dots, \Pi\}$ indexing the observed features of ξ_i . The complement of O_i is noted M_i , and $\xi_i^{O_i} \in \mathbb{R}^{|O_i|}$ (resp. $\xi_i^{M_i} \in \mathbb{R}^{|M_i|}$) denotes the values of the observed (resp. missing) features of ξ_i . The set Ξ_m^O gathers $\xi_i^{O_i}$ for all $i \in \mathcal{I}$.

As to the missing-data mechanism, this work considers “missing at random” data [LR14]: given Ξ_m^O , the probability of a data entry to be missing is independent from $\xi_i^{M_i}$ for all $i \in \mathcal{I}$. This missing-data mechanism can be shown to be ignorable [GJ95], meaning that maximum likelihood estimators remain consistent when evaluated on the incomplete database [LR14]. The stronger “missing completely at random” assumption, requiring the missingness probability of a data entry to also be independent of Ξ_m^O , is thus not necessary. A Gaussian mixture model for Ξ_m then maximizes the observed log-likelihood

$$\log \ell \left(\Xi_m^O | \vartheta \right) = \sum_{i \in \mathcal{I}} \log \left(\sum_{k \in \mathcal{G}} w_k \mathcal{N} \left(\xi_i^{O_i} | \mu_k, \Sigma_k \right) \right) \quad (3.10)$$

over its parameters ϑ , where $\mathcal{N} \left(\xi_i^{O_i} | \mu_k, \Sigma_k \right)$ refers to the marginal of $\mathcal{N}(\xi_i | \mu_k, \Sigma_k)$ over the observed features of ξ_i . The EM-algorithm described above can be adapted to maximize (3.10), by further computing the conditional expectations $\tilde{\mu}_{k, M_i | O_i} \in \mathbb{R}^{|M_i|}$ and covariances $\tilde{\Sigma}_{k, M_i | O_i} \in \mathbb{R}^{|M_i| \times |M_i|}$ of the missing features of each sample ξ_i given $\xi_i^{O_i}$ under each Gaussian component $k \in \mathcal{G}$ during the E-step.

For all $k \in \mathcal{G}$, $\tilde{\mu}_{k, M_i | O_i}$ and $\tilde{\Sigma}_{k, M_i | O_i}$ are easily derived as the conditional distribution of $\xi_i^{M_i}$ given $\xi_i^{O_i}$ remains a multivariate Gaussian when ξ_i is considered as originating from a Gaussian component [Bis06]. The E-step then develops, for $k \in \mathcal{G}$ and $i \in \mathcal{I}$, as

$$\varphi_{ik} = \frac{w_k \mathcal{N} \left(\xi_i^{O_i} | \mu_k, \Sigma_k \right)}{\sum_{j \in \mathcal{G}} w_j \mathcal{N} \left(\xi_i^{O_i} | \mu_j, \Sigma_j \right)}, \quad (3.11)$$

$$N_k = \sum_{i \in \mathcal{I}} \varphi_{ik}, \quad (3.12)$$

$$\tilde{\mu}_{k, M_i | O_i} = \mu_k^{M_i} + \Sigma_k^{MO_i} \left(\Sigma_k^{OO_i} \right)^{-1} \left(\xi_i^{O_i} - \mu_k^{O_i} \right), \quad (3.13)$$

$$\tilde{\Sigma}_{k, M_i | O_i} = \Sigma_k^{MM_i} - \Sigma_k^{MO_i} \left(\Sigma_k^{OO_i} \right)^{-1} \Sigma_k^{OM_i}, \quad (3.14)$$

where $\mu_k^{M_i}$ and $\mu_k^{O_i}$ denote the components of μ_k indexed by M_i and O_i respectively and where $\Sigma_k^{AB_i}$ refers to the sub-matrix of Σ_k with rows indexed

by A_i and columns by B_i .

Defining $\tilde{\boldsymbol{\xi}}_{ik} \in \mathbb{R}^{\tilde{I}}$ with its components indexed by O_i and M_i equal to $\boldsymbol{\xi}_i^{O_i}$ and $\tilde{\boldsymbol{\mu}}_{k,M_i|O_i}$, the M-step then computes, for $k \in \mathcal{G}$,

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{i \in \mathcal{I}} \varphi_{ik} \tilde{\boldsymbol{\xi}}_{ik}, \quad (3.15)$$

$$w_k = \frac{N_k}{N}, \quad (3.16)$$

$$\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{i \in \mathcal{I}} \varphi_{ik} (\tilde{\boldsymbol{\xi}}_{ik} - \boldsymbol{\mu}_k) (\tilde{\boldsymbol{\xi}}_{ik} - \boldsymbol{\mu}_k)^T + \tilde{\boldsymbol{\Sigma}}_{ik}, \quad (3.17)$$

where $\tilde{\boldsymbol{\Sigma}}_{ik} \in \mathbb{R}^{\tilde{I} \times \tilde{I}}$ is defined as $\tilde{\boldsymbol{\Sigma}}_{ik}^{MM_i} = \varphi_{ik} \tilde{\boldsymbol{\Sigma}}_{k,M_i|O_i}$ and zero elsewhere, $\tilde{\boldsymbol{\Sigma}}_{ik}^{MM_i}$ denoting the sub-matrix of $\tilde{\boldsymbol{\Sigma}}_{ik}$ with rows and columns indexed by M_i .

3.1.2 Sweep inversion operator

One can note that the E-step requires inverting a different matrix for each component and set of observed features, which may scale as $\mathcal{O}(GN)$ in HD data sets with many missing entries. Such data sets are indeed more likely to present a different pattern of observed features for each sample. As performing $\mathcal{O}(GN)$ matrix inversions at each E-step has a significant cost, some studies suggest to employ the sweep inversion operator [ELVB14, LR14], which is used for the experiments in this chapter in Section 3.3. The sweep operator definition along as its application to compute the E-step is therefore described in this section.

Sweeping a symmetric matrix $\mathbf{E} \in \mathbb{R}^{V \times V}$ on its row and column $l \in \{1, \dots, V\}$ results in a symmetric matrix $\mathbf{E}_s = S_{\{l\}}(\mathbf{E}) \in \mathbb{R}^{V \times V}$ where

$$(\mathbf{E}_s)_{ll} = \frac{-1}{(\mathbf{E})_{ll}}, \quad (3.18)$$

$$(\mathbf{E}_s)_{lj} = (\mathbf{E}_s)_{jl} = \frac{(\mathbf{E})_{lj}}{(\mathbf{E})_{ll}} \text{ if } j \neq l, \quad (3.19)$$

$$(\mathbf{E}_s)_{ij} = (\mathbf{E})_{ij} - \frac{(\mathbf{E})_{il}(\mathbf{E})_{lj}}{(\mathbf{E})_{ll}} \text{ if } i \neq l \text{ \& } j \neq l. \quad (3.20)$$

Sweeping over a set of rows and columns can be performed in any order as the sweep operator can be shown to be commutative [LR14, ELVB14]. Reverse sweeping $\mathbf{E}$ on l yields a symmetric matrix $\mathbf{E}_r = S_{\{l\}}^{-1}(\mathbf{E}) \in \mathbb{R}^{V \times V}$ equal to $\mathbf{E}_s$ except that

$$(\mathbf{E}_r)_{lj} = (\mathbf{E}_r)_{jl} = -(\mathbf{E}_s)_{lj} \text{ if } j \neq l. \quad (3.21)$$

Let us denote by $\tilde{\boldsymbol{\mu}}_{ik} \in \mathbb{R}^{\Pi}$ a vector whose components indexed by O_i (resp. M_i) are equal to $\boldsymbol{\mu}_k^{O_i} - \boldsymbol{\xi}_i^{O_i}$ (resp. $\boldsymbol{\mu}_k^{M_i}$). It can be shown [LR14, ELVB14] that, for $i \in \mathcal{I}$ and $k \in \mathcal{G}$,

$$S_{O_i} \begin{pmatrix} \boldsymbol{\Sigma}_k & \tilde{\boldsymbol{\mu}}_{ik} \\ \tilde{\boldsymbol{\mu}}_{ik}^T & 0 \end{pmatrix} = S_{M_i}^{-1} \begin{pmatrix} -\boldsymbol{\Sigma}_k^{-1} & \boldsymbol{\Sigma}_k^{-1} \tilde{\boldsymbol{\mu}}_{ik} \\ \tilde{\boldsymbol{\mu}}_{ik}^T \boldsymbol{\Sigma}_k^{-1} & -\tilde{\boldsymbol{\mu}}_{ik}^T \boldsymbol{\Sigma}_k^{-1} \tilde{\boldsymbol{\mu}}_{ik} \end{pmatrix} \quad (3.22)$$

contains $\tilde{\boldsymbol{\Sigma}}_{k,M_i|O_i}$ in its rows and columns indexed by M_i , $\tilde{\boldsymbol{\mu}}_{k,M_i|O_i}$ in its last column at the rows indexed by M_i and

$$\begin{pmatrix} \boldsymbol{\xi}_i^{O_i} - \boldsymbol{\mu}_k^{O_i} \end{pmatrix}^T \left(\boldsymbol{\Sigma}_k^{O O_i} \right)^{-1} \begin{pmatrix} \boldsymbol{\mu}_k^{O_i} - \boldsymbol{\xi}_i^{O_i} \end{pmatrix} \quad (3.23)$$

in its bottom-right entry, useful to evaluate φ_{ik} . Since $\boldsymbol{\Sigma}_k^{-1} = \mathbf{Q}_k \bar{\boldsymbol{\Lambda}}_k^{-1} \mathbf{Q}_k^T$ as developed in Section 3.1.3, which may be evaluated only once for all $i \in \mathcal{I}$, the right-hand side of (3.22) is faster to compute when $|O_i| > |M_i|$ [ELVB14].

3.1.3 High-dimensional extension

As the covariance matrices dimension is $\Pi \times \Pi$, the number of parameters of Gaussian mixture models scales in $\mathcal{O}(G\Pi^2)$. This makes the fitting of such models difficult in HD spaces, requiring much more samples than available. However adopting low-dimensional representations of each Gaussian function circumvents this problem, leading to the family of high-dimensional data clustering (HDDC) methods [BGS07]. The latter constrains the structure of the covariance matrices computed in the M-step to reduce the number of model parameters.

This work considers the most general HDDC model. For each component $k \in \mathcal{G}$, the eigenvalue decomposition (EVD) of $\boldsymbol{\Sigma}_k$ is first computed after each M-step:

$$\boldsymbol{\Sigma}_k = \mathbf{Q}_k \boldsymbol{\Lambda}_k \mathbf{Q}_k^T, \quad (3.24)$$

where the eigenvalues are sorted in decreasing order on the diagonal of $\boldsymbol{\Lambda}_k$. A new $\Pi \times \Pi$ diagonal matrix $\bar{\boldsymbol{\Lambda}}_k$ is then defined as, for $j \in \{1, \dots, \Pi\}$,

$$(\bar{\boldsymbol{\Lambda}}_k)_{jj} = \begin{cases} (\boldsymbol{\Lambda}_k)_{jj} & \text{if } j \leq d_k, \\ \frac{1}{\Pi - d_k} \sum_{l=d_k+1}^{\Pi} (\boldsymbol{\Lambda}_k)_{ll} & \text{otherwise,} \end{cases} \quad (3.25)$$

for some $d_k \in \{1, \dots, \Pi\}$. Note that $\text{tr}(\boldsymbol{\Lambda}_k) = \text{tr}(\bar{\boldsymbol{\Lambda}}_k)$, with $\text{tr}(\mathbf{A})$ denoting the trace of the matrix $\mathbf{A}$. Covariance matrix $\boldsymbol{\Sigma}_k$ defined in (3.17) is finally replaced with

$$\bar{\boldsymbol{\Sigma}}_k = \mathbf{Q}_k \bar{\boldsymbol{\Lambda}}_k \mathbf{Q}_k^T, \quad (3.26)$$

with $\text{tr}(\boldsymbol{\Sigma}_k) = \text{tr}(\bar{\boldsymbol{\Sigma}}_k)$.

Since the variance along an eigenvector is equal to its associated eigenvalue, the variance in the subspace spanned by the eigenvectors of the $\Pi - d_k$ last eigenvalues of Σ_k is modeled using a single parameter [BGS07]. Hyper-parameter d_k is determined with the scree test [Cat66]: d_k is set to the smallest integer between 1 and Π such that the subsequent successive eigenvalues differences are smaller than a threshold $\eta_k \geq 0$, for which a heuristic derivation is detailed in Section 3.1.5. The eigenvalues are hence replaced with their mean as soon as they become similar to each other. This ensures that $\bar{\Lambda}_k$ remains close to Λ_k , suggesting that the HDDC approximation is reasonable. This motivates the use of the scree test for HDDC models [BGS07].

3.1.4 EM algorithm time complexity

A single EM iteration includes the computation of the E- and M-steps, as well as the application of the HDDC method. The latter entails the derivation of the EVD of the covariance matrices, from which their inverses are deduced. This scales in $\mathcal{O}(G\Pi^3)$ time.

About the M-step, updating the covariance matrices costs $\mathcal{O}(GN\Pi^2)$. As to the E-step, evaluating φ_{ik} for a complete sample scales in $\mathcal{O}(\Pi^2)$. When ξ_i contains missing values, the sweep inversion operator computing (3.22) costs

$$\mathcal{O}(\Pi^2 \min(|O_i|, |M_i|)), \quad (3.27)$$

which allows obtaining φ_{ik} , $\tilde{\mu}_{k,M_i|O_i}$ and $\tilde{\Sigma}_{k,M_i|O_i}$ in

$$\mathcal{O}\left(|O_i|^2|M_i| + |O_i||M_i|^2 + |M_i|^2\right) \quad (3.28)$$

time. As $|O_i| = \Pi - |M_i|$, performing the E-step for all samples and Gaussians hence scales in

$$\mathcal{O}\left(G\Pi \sum_{i \in \mathcal{I}} \Pi \max(1, |M_i|) + |M_i|^2\right). \quad (3.29)$$

The time complexity of an EM iteration then expresses as

$$T_{MG} := \mathcal{O}\left(G\Pi \left(\Pi^2 + N\Pi + \Pi \left(\sum_{i \in \mathcal{I}} |M_i| + \sum_{i \in \mathcal{I}} |M_i|^2\right)\right)\right). \quad (3.30)$$

3.1.5 Meta-parameter tuning

Thresholds η_k as well as the number G of components are hyper-parameters which can be determined using the Bayesian Information Criterion (BIC)

[SEM⁺16]. If the number P_ϑ of parameters of the model characterized by ϑ is smaller than N , the BIC can be expressed as [S⁺78]

$$-2\log \mathcal{L}(\vartheta) + P_\vartheta \log N. \quad (3.31)$$

Otherwise, it is not defined. The numbers P_ϑ of parameters of the HDDC models are expressed in [BGS07] as a function of Π , G and d_k for all $k \in \mathcal{G}$.

The BIC is preferred to Akaike's Information Criterion (AIC) [Aka74] as the latter typically selects too complex models [HT89]. As mentioned in previous studies [ELVB14, SEM⁺16], it is crucial to avoid considering overly complex models, as P_ϑ grows quickly with G and may rapidly become larger than N when dealing with HD data [ELVB14]. This would yield Gaussian mixture models particularly subject to overfitting of the observed data, affecting the quality of the multiple imputations. This motivates employing the BIC criterion [SEM⁺16], which has moreover been shown to be efficient in the context of HDDC models [BGS07].

A simple heuristic then allows determining G [SEM⁺16]: first fit two models with one and two components, then with three if the BIC decreases from $G = 1$ to $G = 2$, and so on as long as the BIC decreases. However, for each considered value of G , the thresholds η_k should still be determined. Adopting a similar strategy as for G , η_k is defined as a decreasing fraction of the covariance trace in this chapter: $\eta_k = \zeta \cdot \text{tr}(\Sigma_k)$, where ζ is first set to 0.5 and then repeatedly divided by 2 until the BIC stops decreasing. Note that ζ is common to all Gaussian components and that when it becomes small, d_k will tend towards Π . HDDC applied to low- or average-dimensional data is hence able to consider full covariances, when N is large enough.

3.1.6 EM initialization

The application of the EM algorithm entails the initialization of the Gaussian means, covariances and weights. This work considers uniform weights w_k at the beginning of the algorithm, while the means μ_k are randomly selected among the complete samples. If there are less than G such samples, some incomplete ones are drawn and sample mean imputation is used to fill their missing features [ELVB14]. The covariances Σ_k are all initialized to the sample covariance of the complete examples (complete-case analysis) when there are more than $\Pi + 1$ complete samples, and to a diagonal matrix with the sample variance of the features otherwise [ELVB14]. This ensures the positive definiteness of Σ_k , which is not the case when computing the covariances separately for all pairs of features using the available examples [EDVL13] (available-case analysis). Other initialization procedures for the Gaussian means, covariances and weights could also be employed [BB16], possibly leading to improved stationary points of the observed log-likelihood (3.10).

The EM algorithm being sensitive to the parameter initialization, several runs should be conducted with different initializations to explore several local maxima. In this chapter, for each considered values of G and ζ , the algorithm is run 10 times with distinct initializations and the one yielding the smallest BIC is kept. Each run of the algorithm stops as soon as one of the following conditions is met: (i) the log-likelihood difference between two EM iterations is smaller than 10^{-5} , (ii) 400 iterations have been performed, or (iii) one of the covariance matrices has only close-to zero eigenvalues.

3.1.7 Conditional mixture model sampling

Multiple imputation schemes require to sample the conditional distribution of the missing features of each example ξ_i given its observed features. As the data set is modeled using a Gaussian mixture model, this conditional distribution also follows a Gaussian mixture with G components and with means $\tilde{\mu}_{k,M_i|O_i}$, covariances $\tilde{\Sigma}_{k,M_i|O_i}$ and weights φ_{ik} [SEM⁺16]. The conditional mean imputation of $\xi_i^{M_i}$ then consists in replacing it with $\sum_{k \in \mathcal{G}} \varphi_{ik} \tilde{\mu}_{k,M_i|O_i}$.

On the other hand, a random sample from the conditional distribution can be obtained by first drawing a component index k from $\mathcal{G}$ with a φ_{ik} probability, then sampling a vector $\mathbf{y}_{ik}$ from a $|M_i|$ -dimensional Gaussian distribution, with zero mean and identity covariance, and finally computing $\tilde{\mu}_{k,M_i|O_i} + \mathbf{C}_{ik} \mathbf{y}_{ik}$, where $\mathbf{C}_{ik}$ is the Cholesky factor of $\tilde{\Sigma}_{k,M_i|O_i}$ [SEM⁺16].

3.2 Multiple imputations for dimensionality reduction with missing data

Given an incomplete data set Ξ_m with N independent examples, neighbor embedding algorithms cannot be directly applied as the examples $\xi_i \in \mathbb{R}^H$ contain unobserved features. However, following (3.2), the expectation, over the missing values, of their cost function may be minimized. Indeed, in the case of multi-scale SNE, the expectation of (2.26) can be expressed as

$$\mathbb{E}[C] = - \sum_{i \in \mathcal{I}, j \in \mathcal{I} \setminus \{i\}} \mathbb{E}[\sigma_{ij}] \log s_{ij}. \quad (3.32)$$

Only the expected HD similarities must hence be evaluated in order to apply multi-scale SNE. This can be performed by modeling the Ξ_m distribution in $\mathbb{R}^H$ through a Gaussian mixture model. One can then sample Ψ times the conditional distribution of the missing features of each sample ξ_i given its observed features. This creates Ψ complete data sets $\Xi^1, \dots, \Xi^\Psi$. Their multi-scale HD similarities $[\sigma_{ij}^1]_{i,j=1}^N, \dots, [\sigma_{ij}^\Psi]_{i,j=1}^N$ may

hence be computed, allowing to approximate the expectation of the multi-scale HD similarities of Ξ_m :

$$\mathbb{E}[\sigma_{ij}] \approx \frac{1}{\Psi} \sum_{l=1}^{\Psi} \sigma_{ij}^l. \quad (3.33)$$

This approach follows the multiple imputation paradigm for dealing with missing data [LR14, Rub04, End10, SEM⁺16]. Notice that computing the multi-scale HD similarities of the complete data set resulting from the imputation of the missing values of each example with their conditional means given the observed features does not approximate the expectation of the multi-scale HD similarities of Ξ_m , as they are non-linear in the HD coordinates. It would yield the LDS of (3.1) instead of (3.2).

The procedure for applying multi-scale SNE to a data set Ξ_m with missing values is illustrated in Fig. 3.1, in comparison with the typical workflow of a single imputation method. It can be summarized as follows:

1. Fit a Gaussian mixture model on Ξ_m , as detailed in Section 3.1;
2. Use this model to extract Ψ random samples of the conditional distribution of the missing features of each example ξ_i given its observed ones, in order to create Ψ complete data sets $\Xi^1, \dots, \Xi^\Psi$;
3. Compute their multi-scale HD similarities and the expected multi-scale HD similarities of Ξ_m using (3.33);
4. Use the approximation to minimize the expected multi-scale SNE cost function (3.32) over the LD coordinates.

The expected multi-scale SNE gradient can be evaluated in the same way as the cost function, by employing (3.33).

Computing the multi-scale HD similarities of the Ψ imputed data sets scales in $\mathcal{O}(\Psi N^2 \log N)$ time. Their average enables to tune the LD coordinates thanks to gradient-based optimization algorithm, each gradient evaluation costing $\mathcal{O}(N^2 \log N)$. The total time complexity of the procedure then scales in $\mathcal{O}(T_{MG} + \Psi N^2 \log N)$, where T_{MG} is defined in Section 3.1.4. The number Ψ of random samplings is set to 100 during the experiments of this chapter, but the results are stable across variations over this meta-parameter.

It should be stressed that, even though this work focuses on the use of multi-scale SNE [LPOV15], the proposed methodology can be directly applied using any other neighbor embedding method, by approximating appropriate expectations. More generally, any DR scheme optimizing some criterion can be employed, as the cost function expectation, as well as its

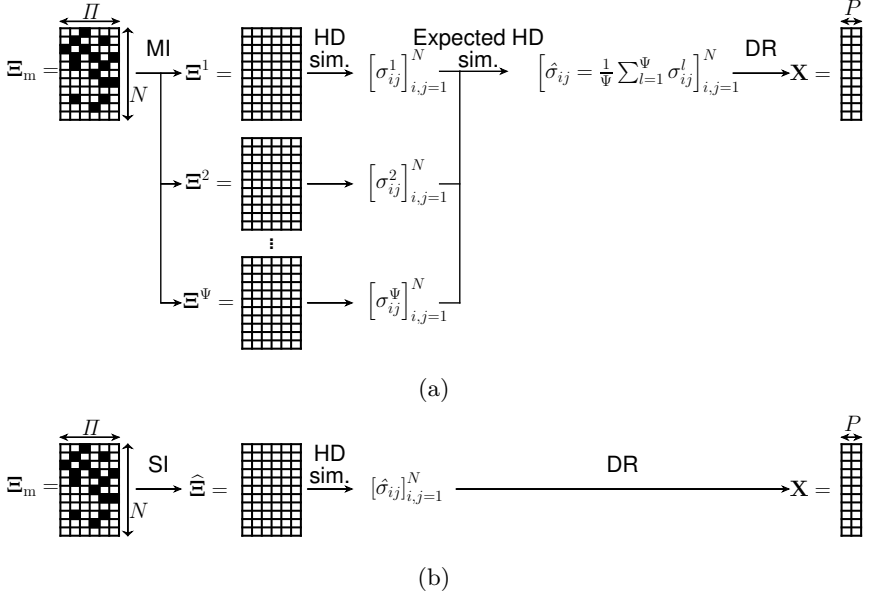


Figure 3.1: (a) Procedure to reduce the dimension of an incomplete data set using multi-scale SNE under the multiple imputation (MI) paradigm. The depicted data sets have one row per sample and one column per feature. Black entries indicate missing values. (b) Illustration of a method managing missing data during a preprocessing step, hence imputing the incomplete data set only once. In this view, the multiple imputation paradigm more faithfully estimates the expected multi-scale HD similarities. SI stands for single imputation, while “sim.” abbreviates “similarities”.

gradient, can always be estimated under the multiple imputation framework after fitting a Gaussian mixture model on the incomplete data set. The rest of this section details the application of the methodology with the SNE, t -SNE, Jensen-Shannon embedding (JSE) and Neighborhood Retrieval Visualizer (NeRV) methods.

SNE can be used in the same way as multi-scale SNE, by estimating the expectations $\mathbb{E}[\sigma_{ij\star}]$ for all $i \in \mathcal{I}$ and $j \in \mathcal{I} \setminus \{i\}$ instead of $\mathbb{E}[\sigma_{ij}]$ and by minimizing, over the LD positions, the expected SNE cost function

$$\mathbb{E}[C_{\star}] \propto - \sum_{i \in \mathcal{I}, j \in \mathcal{I} \setminus \{i\}} \mathbb{E}[\sigma_{ij\star}] \log s_{ij\star}, \quad (3.34)$$

the entropy of the HD similarities being constant with respect to $\mathbf{X}$. Concerning t -SNE [vdMH08], the expectation of its cost function,

$$\mathbb{E}[C_{t\star}] \propto - \sum_{i \in \mathcal{I}, j \in \mathcal{I} \setminus \{i\}} \mathbb{E}[\tau_{ij\star}] \log t_{ij}, \quad (3.35)$$

can be minimized by estimating the HD similarities expectation as previously. Indeed, one can observe that

$$\mathbb{E}[\tau_{ij\star}] = \frac{\mathbb{E}[\sigma_{ij\star}] + \mathbb{E}[\sigma_{ji\star}]}{2N}. \quad (3.36)$$

As to JSE [LRB⁺13], the expectation of its cost function is proportional to

$$\begin{aligned} \mathbb{E}[C_{j\star}] \propto & \sum_{i \in \mathcal{I}, j \in \mathcal{I} \setminus \{i\}} \frac{-1}{1 - \kappa_j} \mathbb{E}[\sigma_{ij\star} \log(z_{ij\star})] + \frac{1}{\kappa_j} s_{ij\star} \log(s_{ij\star}) \\ & - \frac{1}{\kappa_j} s_{ij\star} \mathbb{E}[\log(z_{ij\star})]. \end{aligned} \quad (3.37)$$

Given some current LD coordinates, the HD similarities

$$[\sigma_{ij\star}^l; l = 1, \dots, \Psi]_{i,j=1}^N \quad (3.38)$$

of the Ψ complete data sets resulting from the multiple imputations and defining $z_{ij\star}^l = \kappa_j \sigma_{ij\star}^l + (1 - \kappa_j) s_{ij\star}$, the expected JSE cost function is estimated using

$$\mathbb{E}[\sigma_{ij\star} \log z_{ij\star}] \approx \frac{1}{\Psi} \sum_{l=1}^{\Psi} \sigma_{ij\star}^l \log z_{ij\star}^l, \quad (3.39)$$

$$\mathbb{E}[\log z_{ij\star}] \approx \frac{1}{\Psi} \sum_{l=1}^{\Psi} \log z_{ij\star}^l. \quad (3.40)$$

The expected JSE gradient can be similarly approximated.

Finally, the expectation of the NeRV cost function (2.19) is proportional to

$$\begin{aligned} \mathbb{E}[C_{n\star}] \propto & \sum_{i \in \mathcal{I}, j \in \mathcal{I} \setminus \{i\}} (\kappa_n - 1) \mathbb{E}[\sigma_{ij\star}^n \log(s_{ij\star}^n)] \\ & + \kappa_n \mathbb{E}\left[s_{ij\star}^n \log\left(\frac{s_{ij\star}^n}{\sigma_{ij\star}^n}\right)\right], \end{aligned} \quad (3.41)$$

the LD similarities remaining inside the expectations as their precisions depend on the HD ones. Given some LD positions, the expected NeRV cost function can hence be approximated by estimating the expectations in (3.41) using the HD similarities of the Ψ imputed data sets, as for JSE. The NeRV gradient can be evaluated likewise. The multi-scale versions of t -SNE, NeRV and JSE [LPOV15, dBMVL18b] can also be employed in the same way, by replacing their single-scale similarities with multi-scale ones.

3.3 Experimental results

In order to validate the proposed methodology, several methods are experimentally compared on various data sets, mainly from the UCI Machine Learning Repository [Lic13]:

- Breast Tissue (BrTis.);
- Concrete Compressive Strength (CCS) [Yeh98];
- Connectionist Bench on Vowel Recognition and Deterding Data (ConV.);
- Concrete Slump Test (ConcSl.) [Yeh07];
- A random subset of Optical Recognition of Handwritten Digits (Digits) with 50 samples per digit;
- Energy efficiency (EE) [TX12];
- Ecoli;
- Forest Fires (FFir.) [CM07];
- Fertility (Fert.) [GGDJ⁺12];
- Glass Identification (GLId.);
- Indian Liver Patient (ILP);

- Ionosphere (Iono.);
- Iris;
- Istanbul Stock Exchange (IstSt.) [ABB14];
- Leaf [SMdS13];
- Parkinsons (Park.) [LMR⁺07];
- Planning Relax (PIRel.) [Bha12];
- Seeds;
- Servo;
- User Knowledge Modeling (UsKn.) [KSC13];
- Wholesale customers (WholC.) [Abr11];
- Wine;
- Yacht Hydrodynamics (YaHyd.);
- A 3-D artificial data set (Sphere) with 1500 dots randomly distributed on a unit-radius sphere.

Fig. 3.6 and 3.7 indicate the number of samples N and dimension H of each database.

The data sets being all complete, missing values are randomly and independently introduced during the experiments, corresponding to some fixed proportion of the whole data set, while ensuring that each example and feature contain at least one observed value. The completeness of the data sets enables to quantify the quality of the methods after discarding some data according to different criteria described in Section 3.3.1. The compared methods are:

- Multiple imputations using a mixture of multivariate Gaussians, abbreviated as MIMG, which is the suggested methodology;
- Conditional mean imputation using a mixture of multivariate Gaussians (IMG), which fits a Gaussian mixture model on the incomplete data set and then impute its missing values by their conditional mean given the observed features of their examples in order to apply multi-scale SNE on the resulting data set;

- Multiple imputations using a single multivariate Gaussian (MIG) and conditional mean imputation using a single multivariate Gaussian (IG), which are equivalent to MIMG and IMG except that they use only one Gaussian to model the data set;
- Mean imputation (IMe), which first computes the mean of each feature across its observed values in the data, then fills the missing data using the estimated means of their corresponding features and finally applies multi-scale SNE on the imputed data set;
- Incomplete-case k -nearest neighbor imputation (INN) [VHK14], which imputes each missing feature $j \in M_i$ of every sample ξ_i by the mean of the j^{th} feature of its k nearest neighbors in Ξ_m in order to afterward apply multi-scale SNE on the resulting data set. A data sample ξ_l is eligible as a nearest neighbor if and only if $O_i \subset O_l$ and $j \in O_l$. The distances between ξ_i and its eligible nearest neighbors are evaluated by only considering the features indexed in O_i . When the set of eligible nearest neighbors is empty, the imputation relies on the sample mean of the j^{th} feature. INN improves the complete-case k -nearest neighbor imputation [HHE03] as the latter only considers the complete instances (i.e. $O_l = \{1, \dots, \Pi\}$) as eligible nearest neighbors, which greatly limits their number especially when dealing with large proportions of missing values. The experiments in this chapter consider up to 5 nearest neighbors, as suggested by [VHK14, ELVB14].

Comparing MIMG to IMG highlights the difference between optimizing the expected cost function and reducing the dimension of the expected HD data set. IG and MIG provide insights on the importance of the data set distribution modeling when confronted to IMG and MIMG. IMe ignores both the feature relationships and the evaluation of the multi-scale SNE cost function expectation, and may thus be considered as the baseline method. In the same spirit as IG and IMG, INN aims to model the data distribution but only considers a single imputed data set. It should be noted that, when applied on some incomplete data set, MIMG and IMG (resp. MIG and IG) use the same Gaussian mixture model (resp. Gaussian model).

The time complexity of IMG is similar to MIMG but with $\Psi = 1$, while IG and MIG behave likewise but with $G = 1$. On the other hand, the simplicity of IMe enables to reach a $\mathcal{O}(N\Pi + N^2 \log N)$ time complexity. Finally, INN must first find the eligible nearest neighbors for each sample, which costs $\mathcal{O}(N^2\Pi)$. As the imputation is then straightforward, the method scales in $\mathcal{O}(N^2\Pi + N^2 \log N)$ time.

Multi-scale SNE is used by all methods with $K_1 = 2$ and

$$H = \lfloor \log_2 (N/K_1) \rfloor. \quad (3.42)$$

Two-dimensional embeddings are searched ($P = 2$), and the L-BFGS optimization algorithm [BLNZ95] is employed along with multi-scale optimization [LPOV15]. The LD embedding is initialized using an origin-centered, isotropic Gaussian distribution with a 10^{-4} standard deviation [vdMH08]. Euclidean distances are used in the HD and LD spaces.

3.3.1 Quality assessment

The studied methods are evaluated both by assessing the resulting DR performances and by quantifying the approximation quality of the “true” (i.e. complete-case) multi-scale HD similarities $[\sigma_{ij}]_{i,j=1}^N$ by their estimate. These two measures can be computed thanks to the completeness of the data sets, in which missing values are simulated.

The approximation quality of the HD similarities is quantified by the sum of the KL divergences between the “true” and approximated HD similarities for each data point,

$$Q_{\text{KL}} = \sum_{i \in \mathcal{I}, j \in \mathcal{I} \setminus \{i\}} \sigma_{ij} \log \left(\frac{\sigma_{ij}}{\hat{\sigma}_{ij}} \right), \quad (3.43)$$

where $\hat{\sigma}_{ij}$ is the approximation of σ_{ij} obtained by the considered method. In the MIMG case, it is equal to the right-hand side of (3.33).

As to the DR quality assessment, some studies developed criteria measuring the HD neighborhood preservation in the LDS [LV09, LV10]. This principle is indeed considered as driving the DR quality [VPN⁺10, FC07, CB09]. The HD and LD ranks of ξ_j and $\mathbf{x}_j$ with respect to ξ_i and $\mathbf{x}_i$ can first be defined as

$$\rho_{ij} = |\{k \in \mathcal{I} | \delta_{ik} < \delta_{ij} \text{ or } (\delta_{ik} = \delta_{ij} \text{ and } k < j)\}|, \quad (3.44)$$

$$r_{ij} = |\{k \in \mathcal{I} | d_{ik} < d_{ij} \text{ or } (d_{ik} = d_{ij} \text{ and } k < j)\}|. \quad (3.45)$$

The K -ary neighborhoods of ξ_i and $\mathbf{x}_i$ can then be respectively expressed as

$$\nu_i^K = \{j \in \mathcal{I} | 1 \leq \rho_{ij} \leq K\} \quad (3.46)$$

and

$$n_i^K = \{j \in \mathcal{I} | 1 \leq r_{ij} \leq K\}. \quad (3.47)$$

They index the K nearest neighbors of ξ_i and $\mathbf{x}_i$ in the HDS and LDS. Their average normalized agreement is computed as

$$Q_{\text{NX}}(K) = \frac{1}{N} \sum_{i \in \mathcal{I}} \frac{|\nu_i^K \cap n_i^K|}{K} \in [0, 1]. \quad (3.48)$$

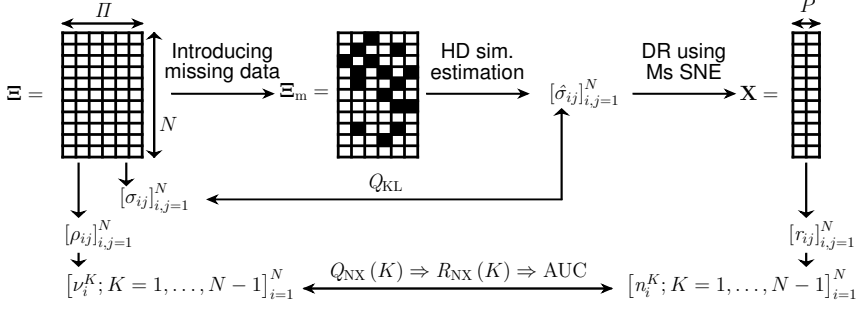


Figure 3.2: Quality assessment employed in the experiments of this chapter. The depicted data sets have one row per sample and one column per feature. Black entries indicate missing values. The notation “sim.” abbreviates “similarities”.

As $\mathbb{E}[Q_{NX}(K)]$ equals $\frac{K}{N-1}$ for random LD points,

$$R_{NX}(K) = \frac{(N-1)Q_{NX}(K) - K}{N-1-K} \quad (3.49)$$

rescales $Q_{NX}(K)$ to enable the comparison of different neighborhood sizes [LRB⁺13]. The $R_{NX}(K)$ curves are typically displayed with a log-scale for K as local neighborhoods usually prevail. Evaluated in $\mathcal{O}(N^2 \log N)$ [LV10], the area under the resulting curve

$$\text{AUC} = \left(\sum_{K=1}^{N-2} K^{-1} \right)^{-1} \cdot \sum_{K=1}^{N-2} \frac{R_{NX}(K)}{K} \in [-1, 1] \quad (3.50)$$

increases with the DR quality, quantified at all scales with an emphasis on small ones [LPOV15].

Both Q_{KL} and the AUC of the computed LD embeddings are used to evaluate the methods during the experiments. Their computation is depicted in Fig. 3.2. Note that for Q_{KL} , the smaller the better whereas for the AUC, the larger the better.

3.3.2 Experimental setting

Missing values are introduced in each data set according to 30 different proportions p_m of its number $N\Pi$ of entries, from 1% to 30%. For each considered proportion p_m of missing values, 200 incomplete data sets are created and all methods are applied on each one of them. Before introducing missing values, Z-score standardization [Bis06] is applied on the data

sets containing features with different ranges, i.e., all except Sphere and Digits. This ensures that missing data for different features have the same importance, which makes the repetitions for a fixed p_m comparable. The repetitions for each p_m enable computing paired statistical tests assessing whether the average performances of the methods are significantly different according to both Q_{KL} and the AUC. For each one of these two scores, two-tailed paired t -tests are employed if their 200 values can be considered to be normally distributed for each p_m and each method, according to D’Agostino and Pearson’s omnibus normality test [d’A71]. Wilcoxon tests are otherwise used. In this chapter, the level of significance is set to $\alpha = 0.01$.

Comparing the performances of several methods for each p_m however consists of multiple hypothesis tests. The Holm–Bonferroni correction is hence applied for each p_m to bound by α the probability to consider as statistically significant at least one non-significant difference [Sha95].

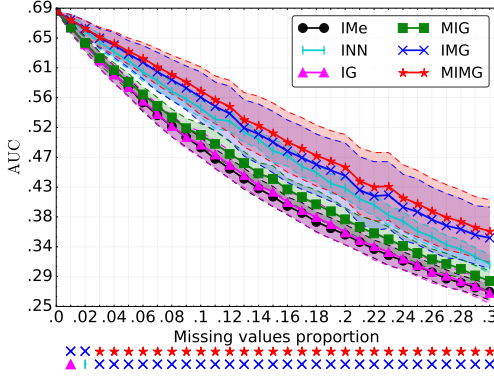
3.3.3 Results

Fig. 3.3 and 3.4 respectively illustrate the results on the Yacht Hydrodynamics ($N = 308$, $\Pi = 6$) and Leaf ($N = 340$, $\Pi = 14$) data sets. On Yacht Hydrodynamics, MIMG is consistently the best method on average for all proportions p_m of missing values, according to both criteria, except for a very limited number of p_m for most of which it is not statistically different from the corresponding best performing method. MIMG is moreover nearly always statistically significantly superior to the second best method, which is almost systematically IMG for the AUC and switches from IMG to MIG for Q_{KL} . It is noteworthy that even though MIG approximates the multi-scale HD similarities for the largest p_m better than MIMG, MIMG still provides better embeddings according to the AUC. This can be explained as, although MIG has a lower Q_{KL} , MIMG may still better approximate the HD similarity distributions around some data points which can have a larger influence for the DR part. For large p_m , the standard deviations of the performances are not negligible, meaning that the missing values positions in the data set strongly influence their impact: if the observed data sufficiently well describe the distribution of the missing features given the observed ones, which is for instance more likely with incomplete samples fairly spread in the data distribution, the distribution modeling of the missing values is easier, enabling better performances. Otherwise, if the missing entries exhibit specific structures, e.g. if they concentrate in some features for some values of observed data, less information may be available about the missing values distribution, affecting the performances. Despite large standard deviations, most results are statistically significant; the performances of the different methods hence vary consistently with respect to the missing-data pattern. IMe and IG are unsurprisingly always out-

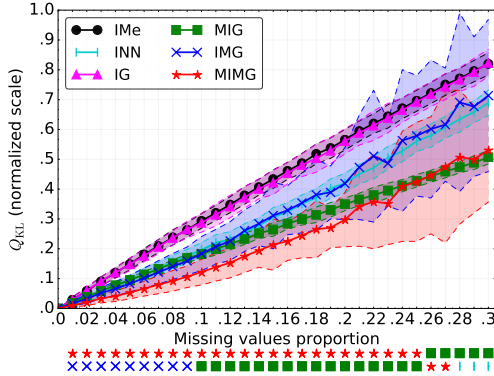
performed by the other methods. The superiority of MIG over IG and of MIMG over IMG moreover stresses the importance of optimizing the expected cost function over the missing data, in order to derive the closest result to the complete data case on average. One can finally observe that the number of Gaussian components chosen in the mixture-based models tends to slightly increase with the proportion p_m of missing values. The growing uncertainty may indeed affect the quality of the BIC criterion for model selection. As the number of observed data decreases with p_m , fitting more Gaussian components may result in slight overfitting of the observed data. This justifies the Q_{KL} increase of MIMG and IMG for large proportions of missing values. Such a phenomenon could probably be mitigated by relying on cross-validation procedures instead of the BIC criterion for selecting the hyper-parameters of the Gaussian mixture, at the expense of increased computation times.

As to the Leaf data set, a similar analysis can be conducted. According to both criteria, MIMG is either always the best method or not statistically different from IMG. The impressive performance drop of INN with p_m is a noticeable difference with Yacht Hydrodynamics. The Leaf data set having more features, increasing the proportion of missing values indeed considerably affects the determination of the nearest neighbors. The number of Gaussian components still slightly increases with p_m . However, no overfitting behavior is observed in this case: the performances of the Gaussian mixture-based methods remain blatantly superior to the ones of the other schemes. They are even more robust with respect to the missing values, since the gap between the methods is increasing with p_m . This confirms that a DR scheme handling missing values should seek for the best possible model of the data distribution.

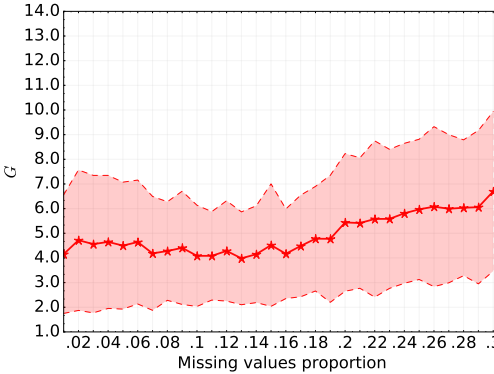
Figure 3.5 further details some of the results obtained on Yacht Hydrodynamics. This data set contains 308 yachts with 22 different hull forms [Lic13]. Each form is represented by 14 boats, which only differ according to their Froude numbers. The 14 Froude numbers are linearly spaced between two values. The embedding obtained by multi-scale SNE on the complete data set hence reveals that the samples are distributed on 22 one-dimensional manifolds. The 14 samples corresponding to some hull form are depicted using the same color in Fig. 3.5a. When introducing missing values in the data set, the $R_{NX}(K)$ curves in Fig. 3.5b show that missing data mainly affect the preservation of small neighborhoods, as the curves corresponding to the different methods are lower for small K values than the one obtained when using the complete data set. Fig. 3.5c illustrates the embedding obtained by MIMG on one of the 200 Yacht Hydrodynamics data set with 5% missing values. We can observe that it tends to recover the structure of the one-dimensional manifolds.



(a)

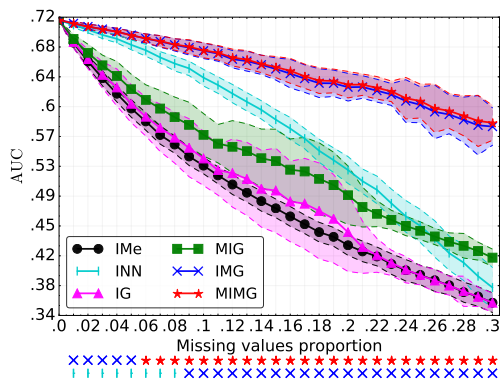


(b)

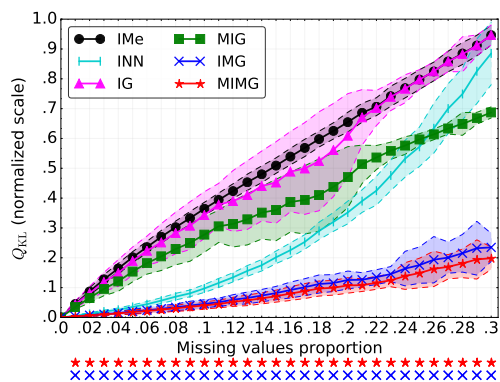


(c)

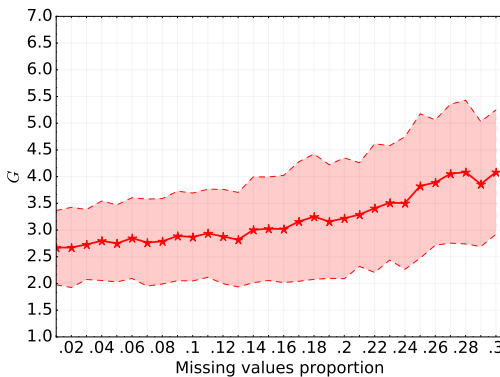
Figure 3.3: (a) depicts the AUC of each method on the Yacht Hydrodynamics data set as a function of the proportion p_m of missing values. The curves give the average AUC over the 200 repetitions for each p_m , while the shaded areas delimit one standard deviation around the mean. The first row of markers below the x -axis indicates the best method on average for each p_m . The second row presents, for each p_m , the best method which is statistically significantly outperformed by the best method on average for the corresponding p_m . (b) is similar, but for the sum of the KL divergences between the “true” and approximated multi-scale HD similarities. The y -axis is normalized, as the absolute value of Q_{KL} brings little insights. (c) indicates, for each p_m , the average number of Gaussian components selected in MIMG and IMG when applying the BIC-based heuristic, the shaded area depicting one standard deviation around the mean.



(a)

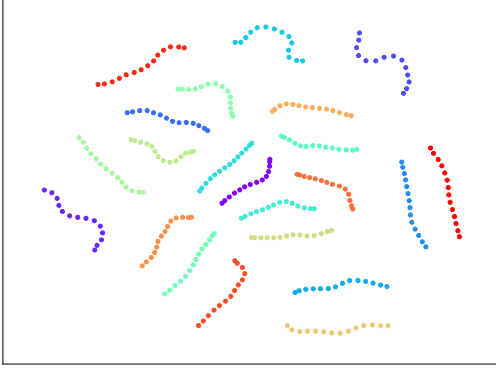
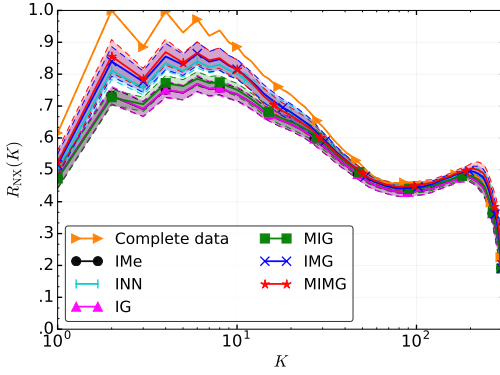
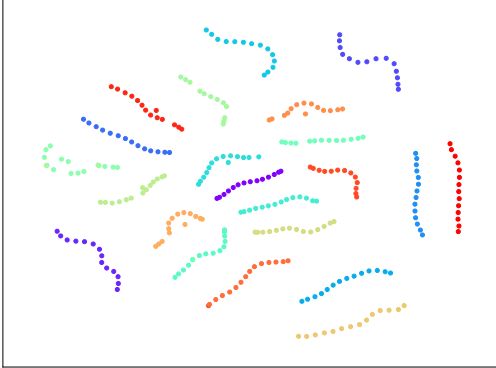


(b)



(c)

Figure 3.4: (a), (b) and (c) are analogous to Figs. 3.3a, 3.3b and 3.3c but for the Leaf data set.

(a) Multi-scale SNE applied on Ξ . AUC = 0.69.(b) $R_{NX}(K)$ curves.

(c) MIMG. AUC = 0.645.

Figure 3.5: (a) depicts the two-dimensional embedding obtained when applying multi-scale SNE on the complete Yacht Hydrodynamics data set. (b) provides the associated $R_{NX}(K)$ curve, as well as the average curves obtained by the different methods on 200 incomplete Yacht Hydrodynamics data sets with 5% missing values. As in Figs. 3.3 and 3.4, the shaded areas delimit one standard deviation around the mean. (c) presents the embedding obtained when applying MIMG on one of the 200 data sets with 5% missing values. The AUC of the embedding is provided below its plot.

Since an analysis such as provided in Figs. 3.3 and 3.4 cannot be presented for all data sets due to space limitations, Figs. 3.6 and 3.7 summarize the remaining results. We can first observe that MIMG is almost systematically top or highly ranked among the different methods according to both the AUC and Q_{KL} . Moreover, when it is (resp. is not) the best performing method, it is often (resp. often not) statistically different from the second best (resp. best) performing one. The few exceptions to these observations are analyzed in more details below.

ConV. is a data set containing numerous small clusters [Lic13]. Unless the number of available samples becomes very large, this kind of data distribution is not easily described using Gaussian mixtures. Fitting many Gaussian components indeed requires to adjust increasingly many model parameters, even when employing HDDC models. On the other hand, as long as the proportion p_m of missing values remains small, nearest neighbor methods impute the missing data using cluster-specific values. The imputation then follows very well the data distribution and, although a single one is considered, it is sufficiently close to the original data set to preserve its clusters. This explains why INN outperforms MIMG for the first few p_m values. Nevertheless, as p_m increases, the computed nearest neighbors are less and less cluster-specific. This drastically reduces the quality of the single imputation, causing the performances of INN to drop. Even though MIMG can probably not model exactly each cluster, it can still model higher-order relationships in the data when facing an increasing number of missing values. This makes MIMG superior to INN for larger p_m . The same kind of analysis holds for Digits and Park., which also contain many clusters [Lic13]. The performances of INN however decrease even more rapidly as the dimensions of these data sets are larger, which makes the determination of the relevant nearest neighbors much more difficult.

On Ecoli, the opposite trend is observed: INN seems to be superior to MIMG for larger p_m according to the AUC. Nevertheless, the performances of the two methods are not statistically significantly different for most p_m . The adaptations of Fig. 3.3a and 3.3b for Ecoli indeed reveal that the difference between both methods is tiny. Concerning the evolution of Q_{KL} with p_m , MIG better estimates the HD similarities than MIMG for larger values of p_m , similarly to the results for Yacht Hydrodynamics. This is also observed on ILP and Servo.

As to Fert. and UsKn., both multiple imputation schemes are efficient according to Q_{KL} . The superiority of MIG over MIMG may result from potential slight overfitting of the mixture model to the observed data. The AUC difference between MIG and MIMG remain however small when looking at the adaptation of Fig. 3.3a for these data sets. In particular for Fert., most methods are statistically equivalent for small p_m according to

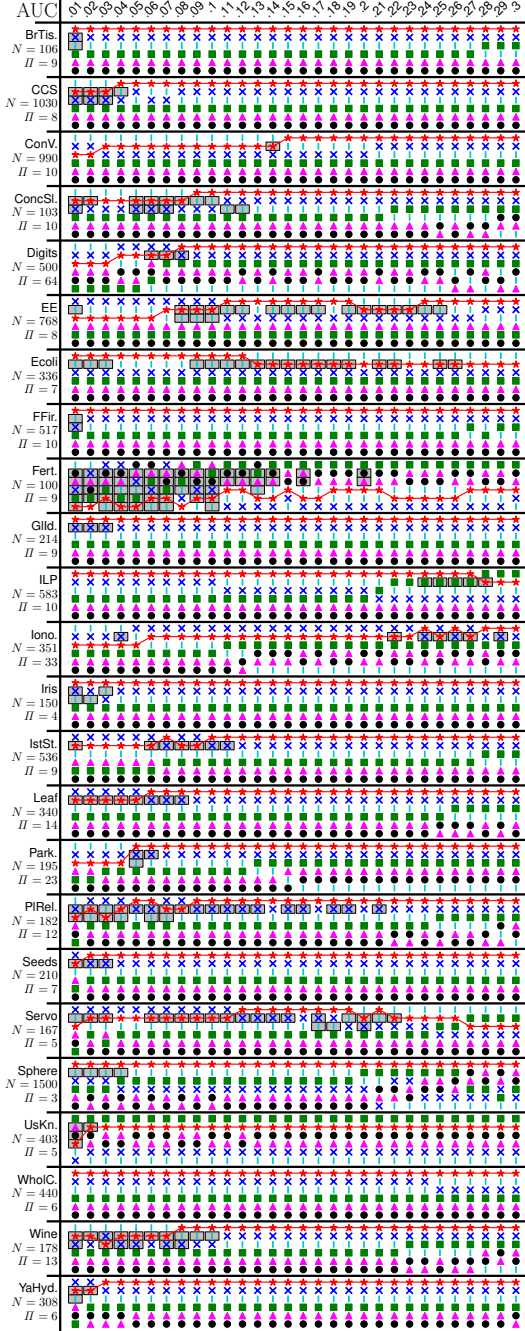


Figure 3.6: Summary of the performances in terms of AUC for all data sets. The results for each data set are displayed horizontally, using six rows of markers. These refer to the same six methods as in Figs. 3.3 and 3.4. There is one column per proportion p_m of missing values. For each data set and p_m , the six markers are sorted vertically from the best performing method on average at the top (i.e. largest average AUC) to the worst average one at the bottom (i.e. smallest average AUC). The averages are computed across the 200 replications for each p_m . The markers corresponding to MIMG are linked. A grey shaded area is displayed behind a marker when the corresponding method is **not** statistically significantly outperformed by the best method on average for the associated p_m , except for the top marker of the columns, as a method can obviously not be significantly different from itself.

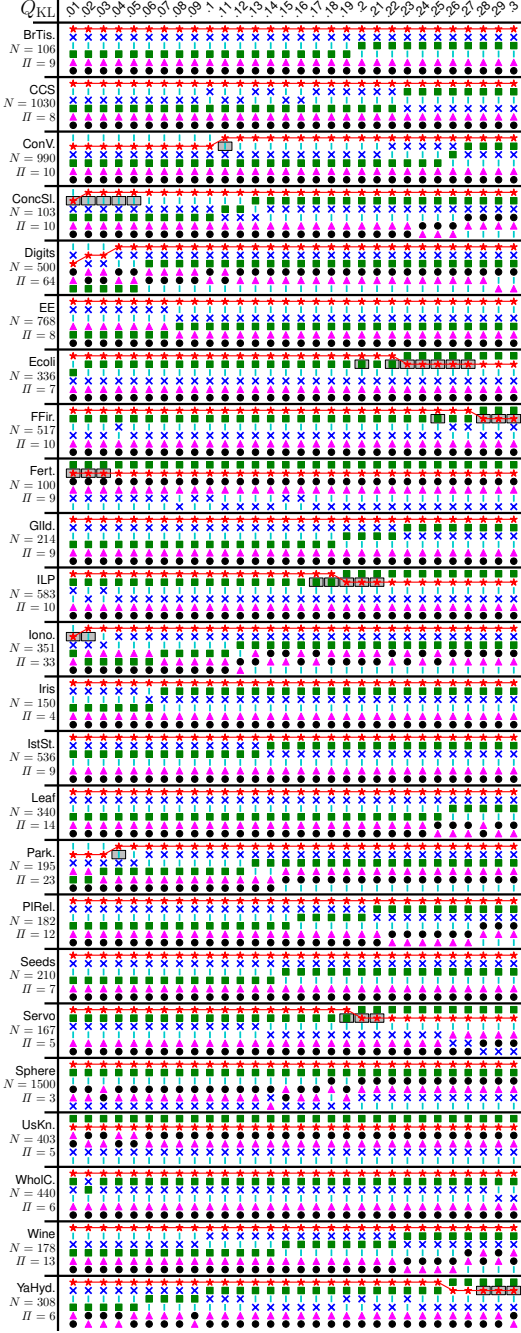


Figure 3.7: Summary of the performances in terms of Q_{KL} for all data sets. The results for each data set are displayed horizontally, using six rows of markers. These refer to the same six methods as in Figs. 3.3 and 3.4. There is one column per proportion p_m of missing values. For each data set and p_m , the six markers are sorted vertically from the best performing method on average at the top (i.e. smallest average Q_{KL}) to the worst average one at the bottom (i.e. largest average Q_{KL}). The averages are computed across the 200 replications for each p_m . The markers corresponding to MIMG are linked. A grey shaded area is displayed behind a marker when the corresponding method is **not** statistically significantly outperformed by the best method on average for the associated p_m , except for the top marker of the columns, as a method can obviously not be significantly different from itself.

the AUC, possibly because this data set is hardly structured; the missing values management hence does not much impact the HD neighborhood preservation.

For Iono. and the first few p_m of EE, MIMG better approximates the HD similarities than IMG from the Q_{KL} viewpoint. However, IMG is superior according to the AUC, but only with a tight margin when observing the adaptation of Fig. 3.3a for these data sets. The tiny AUC improvement of IMG over MIMG is probably due to a better estimation of the HD similarity distributions around some data points which have a larger influence on the HD neighborhood preservation.

Generally, multiple imputation schemes tend to outperform their single imputation versions, MIMG and MIG being respectively globally superior to IMG and IG. IMe and IG appear almost systematically as the weakest approaches. The performances of INN quickly drop as p_m increases, especially on data sets with larger dimensions such as Iono., Digits or Park.

3.4 Discussion

This chapter introduces the missing data processing in DR. After modeling the incomplete data set distribution using a Gaussian mixture extension for HD data, several imputed versions of the data set are created by sampling the conditional distribution of the missing features of the examples given their observed features. These complete data sets are employed to reliably estimate the DR algorithm cost function expectation over the incomplete database. Its minimization yields the LD embedding with the closest cost function value on average with respect to the complete data case. This multiple imputation approach hence provides a relevant LD representation of each, possibly incomplete, data point. Experimental results show that this procedure outperforms alternative methods while being generally applicable with any DR scheme optimizing some criterion.

The Gaussian mixture model as well as the multiple imputation paradigm inevitably come up with an increase of the computation time complexity compared to single imputation schemes. Their superior performances however justify the use of additional computing resources whenever possible. A situation in which the proposed approach may not be most efficient is when facing extremely HD data: Gaussian mixtures would then be difficult to fit due to their increasing numbers of parameters, even with HDDC methods [BGS07]. This motivates the development of alternative models of the data distribution, such as non-parametric estimations, to study whether they could better deal with the inherent sparsity of very HD spaces and reduce the time complexity of the approach.

Chapter 4

Fast neighbor embedding

DESPITE their impressive results in terms of DR quality, single- and multi-scale NE methods respectively have $\mathcal{O}(N^2)$ and $\mathcal{O}(N^2 \log N)$ time complexity when employed with N data samples. Although the logarithmic factor further induced by the multi-scale behavior has a minor impact, the quadratic dependency is not suited to big data sets and confines the use of NE methods to a few thousands data points. An obvious workaround is to employ a NE method solely on a small random subset of the database, enabling its visualization for exploratory analysis. However, analyzing a reduced part of a data set often only reveals a limited fraction of the HD data relationships [PG12]. Another option is to use the HD and LD positions of the selected data samples to project the remaining data points after learning a parametric function between the HD and LD spaces [Maa09], or by using interpolation [WYY⁺10] or an out-of-sample extension [CP10]. Nevertheless, learning such mappings is typically difficult since the NE method only processes a small part of the database [vdM14], hence ignoring the information carried by most of the data points on the underlying HD structures [vdMH08]. Some studies experimentally show that applying NE methods on a subset of a database poorly generalizes as it largely neglects the HD data interactions [YPK13], affecting the projection quality of the remaining samples and altering the structure of the data. Considering big data sets is further demonstrated to always be advantageous to tune visualizations, regardless of whether the whole database or only a portion needs to be analyzed [YPK13]. Therefore, some works attempt to incorporate information from the entire data structure when visualizing a part of the samples, such as landmark *t*-SNE [vdMH08]. However, while preventing to observe the whole database, this approach entails considerable running times and memory requirements [YPK13].

For these reasons, lower-complexity approximations were introduced for

single-scale methods, either relying on tree-based computations [YPK13, vdM14], fast multipole methods (FMMs) [VCP14], negative sampling techniques (NSTs) [TLZM16] or Gaussian mixture modeling approximations [PG12]. Although they effectively reduce computation times by reaching near-linear time complexity with respect to the number of data samples, Gaussian mixtures are typically difficult to tune in (very) HD spaces [BGS07] and require to determine their number of components. A poor tuning or an inadequate number of mixture components deteriorates the quality of the HD modeling and hence of the LD embedding. On the other hand, tree-based schemes, FMMs and NSTs achieve impressive performances in terms of both quality and speedup, ensuing in methods with $\mathcal{O}(N \log N)$ and $\mathcal{O}(N)$ time complexity provided that the HD neighbor probabilities can be sparsely encoded. The tree-based algorithms and FMMs poorly scale with the LD dimension, but the latter is typically as small as 2 or 3 for visualization purposes. However, the sparsity of the HD neighbor probability distributions is a key ingredient of tree-based methods, FMMs and NSTs. This fits the single-scale framework well, in which the neighbor probability distributions remain local around the data points, but not the multi-scale one. Indeed, multi-scale neighbor probability distributions span much larger levels of data organization to account for both local and global neighboring structures when tuning the LD embedding. This defines dense neighbor probability distributions, preventing to accelerate multi-scale NE methods as the single-scale ones.

Therefore, after reviewing and assessing tree-based accelerations of single-scale NE techniques, this chapter focuses on developing fast multi-scale NE schemes with $\mathcal{O}(N \log^2 N)$ time complexity instead of $\mathcal{O}(N^2 \log N)$, able to manage and process large-scale databases with hundreds of thousands data points for visualization [dBMVL20]. Appropriate neighbor sets are first computed for all samples using vantage-point trees defined on random sub-samplings of the data set at all scales. This enables preserving the multi-scale behavior of the methods by accounting for global data structures in the obtained neighbor sets. The cost function of the considered multi-scale scheme is then minimized using these sets with a Barnes-Hut (BH) algorithm to efficiently approximate the LD data interactions and avoid LD crushing phenomena, i.e. overlapping LD points. Simulations on numerous data sets evidence that the proposed fast multi-scale methods outperform state-of-the-art single-scale approaches such as *t*-SNE in terms of LD embedding quality quantified using neighborhood preservation criteria, while at the same time being considerably faster than non-accelerated NE algorithms. These results are statistically significant with respect to the random sub-samplings of the data.

This chapter is organized as follows: Section 4.1 first describes lower-

complexity approximations for fast single-scale schemes. Section 4.2 then extensively compares t -SNE [vdMH08] and its Barnes-Hut (BH) acceleration [vdM14] using neighborhood preservation-based DR quality criteria, to quantify the faithfulness of the approximation and study its suitability for the multi-scale setting [dBMVL18a]. Afterwards, Section 4.3 introduces the proposed randomized accelerations of multi-scale neighbor embedding schemes [dBMVL20]. Section 4.4 details experimental results comparing the performances of (single-scale) t -SNE, multi-scale SNE and multi-scale t -SNE with their respective fast versions on several data sets, in terms of both computation time and DR quality, for varying values of the BH algorithm meta-parameter. The accelerated methods are also validated on large databases on which it is unaffordable to employ the original, non-fast NE algorithms. Section 4.5 discusses the obtained performances.

4.1 Accelerating single-scale neighbor embedding methods

The quadratic time complexity of both single- and multi-scale NE schemes with respect to N limits their applicability to moderate-sized data sets. This motivated accelerating single-scale methods, by harnessing the typical smallness of the user-defined perplexity $K_\star$ with respect to N . This observation enables defining sparse HD similarities in $\mathcal{O}(N \log N)$ time. The cost functions of single-scale approaches and their gradients can then be estimated using tree-based algorithms [YPK13, vdM14] or fast multipole methods [VCP14] in $\mathcal{O}(N \log N)$ and $\mathcal{O}(N)$ time respectively. Although both tree-based and fast multipole schemes cannot deal with large LD dimensions P , their efficiency is impressive in a visualization context, in which $P \leq 3$. This section focuses on tree-based approximations for SNE and t -SNE, while Section 4.3 details the proposed randomized acceleration of multi-scale methods, with $\mathcal{O}(N \log^2 N)$ time complexity. Indeed, as the multi-scale framework relies on perplexities covering all data scales, the induced dense HD similarities hinder designing lower-complexity multi-scale techniques as in the single-scale case. In both this section and Section 4.3, Euclidean LD distances are assumed to compute the gradients of the cost functions.

Due to its normalized Gaussian nature, the HD similarity $\sigma_{ij\star}$ quickly vanishes as the HD distance δ_{ij} grows [YPK13], especially for large precision $\pi_{i\star}$, which drives the decay of the exponential functions. As small perplexities $K_\star$ are typically employed in practice in the single-scale framework [vdMH08], the adjusted HD precision $\pi_{i\star}$ is usually large. A sparse

approximation of $\sigma_{ij\star}$ then consists in

$$\tilde{\sigma}_{ij\star} = \begin{cases} \frac{\exp(-\tilde{\pi}_{i\star}\delta_{ij}^2/2)}{\sum_{k \in \mathcal{I}_{i\star}} \exp(-\tilde{\pi}_{i\star}\delta_{ik}^2/2)} & \text{if } j \in \mathcal{I}_{i\star} \\ 0 & \text{otherwise} \end{cases}, \quad (4.1)$$

with

$$\tilde{\pi}_{i\star} \text{ s.t. } \log K_{\star} = - \sum_{j \in \mathcal{I}_{i\star}} \tilde{\sigma}_{ij\star} \log \tilde{\sigma}_{ij\star}, \quad (4.2)$$

where $\mathcal{I}_{i\star}$ indexes the $\lfloor 3K_{\star} \rfloor$ nearest neighbors of ξ_i in $\Xi \setminus \{\xi_i\}$ [vdM14]. The number of nearest neighbors of ξ_i to consider should indeed monotonically increase with $K_{\star}$, since the entropy of $\sigma_{i\star}$ at the right-hand side of (2.6) is adjusted to be equal to $\log(K_{\star})$ and is monotonically decreasing with $\pi_{i\star}$ when the distances in $\{\delta_{ij}, j \in \mathcal{I} \setminus \{i\}\}$ are not all equal [VCP13], as developed in Section 2.1. Therefore, the greater $K_{\star}$, the greater the entropy $e_{i\star}$ in (2.6), the smaller $\pi_{i\star}$, the flatter the HD similarities in $\sigma_{i\star}$, the more HD neighbors of ξ_i lie in the bulk of the exponential functions of the HD similarities in $\sigma_{i\star}$ before they vanish, and the more nearest neighbors should hence be accounted for in (4.1) for the faithfulness of the approximation. The $\lfloor 3K_{\star} \rfloor$ number is suggested in [vdM14] as it yields impressive DR performances, and is employed in this work to ease the comparison with state-of-the-art literature results.

The nearest neighbor sets are exactly derived by creating a vantage-point tree [Yia93] on Ξ and performing N depth-first searches, each yielding the $\lfloor 3K_{\star} \rfloor$ nearest neighbors of a datum. As an advantage, vantage-point trees only require the availability of a valid distance metric to compare HD objects, and can be employed even when HD vector representations are not available [vdM14]. A sparse estimate for $\tau_{ij\star}$ then develops as

$$\tilde{\tau}_{ij\star} = \frac{\tilde{\sigma}_{ij\star} + \tilde{\sigma}_{ji\star}}{2N}. \quad (4.3)$$

Both matrices $[\tilde{\sigma}_{ij\star}]_{i,j \in \mathcal{I}}$ and $[\tilde{\tau}_{ij\star}]_{i,j \in \mathcal{I}}$ have $\mathcal{O}(K_{\star}N)$ non-zero entries instead of $\mathcal{O}(N^2)$ for their non-sparse versions. The tree is created in $\mathcal{O}(N \log N)$ time and each search of the $\lfloor 3K_{\star} \rfloor$ nearest neighbors of a datum costs $\mathcal{O}(K_{\star} \log N)$ time, hence providing $\mathcal{I}_{i\star}$ for all $i \in \mathcal{I}$ in $\mathcal{O}(K_{\star}N \log N)$ time.

Efficiently evaluating the SNE and t -SNE cost functions and their gradients still entails cheaply estimating their remaining non-sparse, LD terms. A Barnes-Hut (BH) algorithm [BH86] can be developed for this purpose, leading to BH SNE [HR02, YPK13] and BH t -SNE [vdM14]. The computations are made explicit for the gradient of t -SNE cost function below, and can be adapted to the SNE case and to estimate the cost functions.

The gradient of $C_{t\star}$ with respect to the LD points extends as

$$\frac{\partial C_{t\star}}{\partial \mathbf{x}_v} = 4 \sum_{i \in \mathcal{I} \setminus \{v\}} (\tau_{vi\star} - t_{vi}) (\mathbf{x}_v - \mathbf{x}_i) (1 + d_{vi}^2)^{-1} \quad (4.4)$$

$$= 4 \underbrace{\sum_{i \in \mathcal{I} \setminus \{v\}} \tau_{vi\star} \frac{\mathbf{x}_v - \mathbf{x}_i}{1 + d_{vi}^2}}_{\mathbf{f}_v^{t\star}} - \underbrace{\frac{\overbrace{\sum_{i \in \mathcal{I} \setminus \{v\}} \frac{\mathbf{x}_v - \mathbf{x}_i}{(1 + d_{vi}^2)^2}}^{\mathbf{g}_v^t}}{\sum_{k \in \mathcal{I}, l \in \mathcal{I} \setminus \{k\}} (1 + d_{kl}^2)^{-1}}}_{g^t}. \quad (4.5)$$

After replacing $\tau_{vi\star}$ with $\tilde{\tau}_{vi\star}$, the sparsity of $\tilde{\tau}_{vi\star}$ enables estimating $\mathbf{f}_v^{t\star}$ for all $v \in \mathcal{I}$ in $\mathcal{O}(K_\star N)$ time.

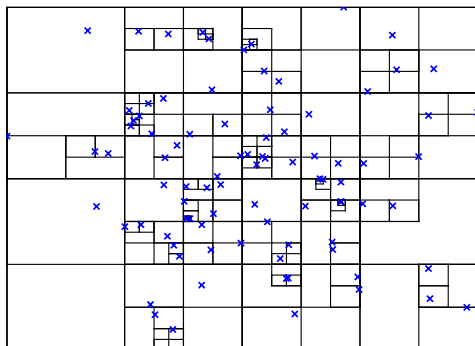
To estimate $\mathbf{g}_v^t / g^t$ for all $v \in \mathcal{I}$ in less than $\mathcal{O}(N^2)$ time, first note that if $d_{ij} \ll d_{vi} \approx d_{vj}$, then $t_{vi} \approx t_{vj}$ [vdM14]. To benefit from this observation and as the LD dimension P is small in visualization tasks, a Barnes-Hut (BH) algorithm can be developed by first constructing a quad- ($P = 2$) or octree ($P = 3$) on $\mathbf{X}$ in $\mathcal{O}(N)$ time [vdM14], as illustrated in Fig. 4.1a. Each node of the tree is associated with a rectangular cell in the LD space, the root containing the whole LD embedding. Each non-leaf node has 4 ($P = 2$) or 8 ($P = 3$) children partitioning the cell into 4 or 8 equal parts at its corners. A leaf contains at most one LD point. For each cell c , let $\mathcal{T}_c \subset \mathcal{I}$ index the LD points inside it. In the node related to c , one can store $|\mathcal{T}_c|$ and the center-of-mass

$$\mathbf{m}_c := \frac{1}{|\mathcal{T}_c|} \sum_{i \in \mathcal{T}_c} \mathbf{x}_i \quad (4.6)$$

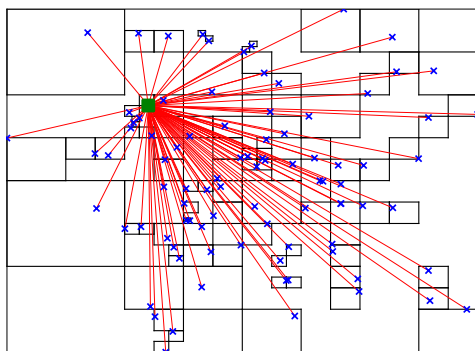
of the points indexed in $\mathcal{T}_c$. Creating the quad- or octree consists in inserting the LD points one after the other, dividing leafs as soon as they contain two LD points, while updating $|\mathcal{T}_c|$ and $\mathbf{m}_c$ of the traversed nodes [vdM14]. Quadtrees can be generalized to any dimension P but are not suitable to large ones, as the number of children per node grows exponentially with P .

Once the tree defined, a depth-first search enables estimating $\mathbf{g}_v^t$ in $\mathcal{O}(\log N)$ average time, if the splits in the tree partition the data into roughly equal parts [YPK13]. At each node with associated cell c during the traversal, one determines whether the terms over $\mathcal{T}_c$ in $\mathbf{g}_v^t$ are similar to each other, which happens when the cell c is sufficiently small and far enough from $\mathbf{x}_v$, in light of the above observation ($d_{ij} \ll d_{vi} \approx d_{vj}$ with i and j in $\mathcal{T}_c$). The condition [BH86]

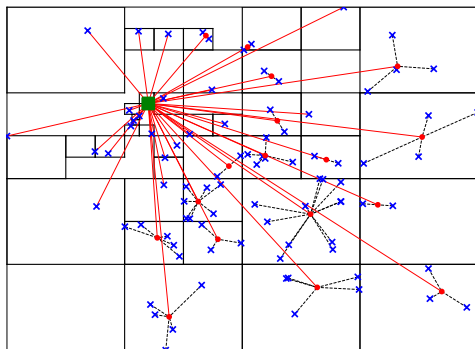
$$\frac{r_c}{d_{v, \mathbf{m}_c}} < \theta \quad (4.7)$$



(a)



(b)



(c)

Figure 4.1: (a) Quadtree of a 2-D data set with crosses as data points. (b) Complete pairwise interactions between the square data point and the remaining crosses. (c) Barnes-Hut approximation of the complete pairwise interactions with the square data point, induced by some threshold θ . The centers-of-mass employed for the approximation are depicted as dots, linked with dotted lines to the crosses belonging to their cells.

formalizes this idea, with r_c the cell diagonal length, $d_{v,\mathbf{m}_c}$ the LD distance between $\mathbf{x}_v$ and $\mathbf{m}_c$, and $\theta \in [0, 1]$ a user-defined threshold parameter trading off accuracy and speed. Larger θ speeds up computations with rougher approximations. The DR quality of the t -SNE acceleration remains, however, stable under variations of θ , as detailed in Section 4.2 [dBMVL18a]. One could also replace r_c in (4.7) with the average distance between $\mathbf{m}_c$ and the LD points indexed in $\mathcal{T}_c$, or the diagonal of the smallest box in cell c which contains all points in $\mathcal{T}_c$. These quantities can be easily derived, respectively in $\mathcal{O}(N \log N)$ time and when creating the tree, and could improve the trade-off between computation time and DR quality.

As soon as (4.7) holds, the depth-first search stops and the part of $\mathbf{g}_v^t$ over $\mathcal{T}_c$ is estimated as

$$\sum_{i \in \mathcal{T}_c} \frac{\mathbf{x}_v - \mathbf{x}_i}{(1 + d_{vi}^2)^2} \approx |\mathcal{T}_c| \frac{\mathbf{x}_v - \mathbf{m}_c}{(1 + d_{v,\mathbf{m}_c}^2)^2}, \quad (4.8)$$

since the terms over $\mathcal{T}_c$ are similar to each other and hence close to their approximation using $\mathbf{m}_c$ as representative for the LD points indexed in $\mathcal{T}_c$. During the same traversal, one can also estimate the sum

$$g_v^t := \sum_{l \in \mathcal{I} \setminus \{v\}} \frac{1}{1 + d_{vl}^2}. \quad (4.9)$$

After completing N passes, approximations $\tilde{\mathbf{g}}_v^t$ and $\tilde{g}_v^t$ of $\mathbf{g}_v^t$ and g_v^t are available for all $v \in \mathcal{I}$. The sum

$$\tilde{g}^t := \sum_{v \in \mathcal{I}} \tilde{g}_v^t \quad (4.10)$$

estimates g^t . Since each traversal costs $\mathcal{O}(\log N)$ average time, $\tilde{g}^t$ and $\tilde{\mathbf{g}}_v^t$ for all $v \in \mathcal{I}$ are obtained in $\mathcal{O}(N \log N)$ average time. Assuming that $K_\star$ is small compared to N , an approximation of $C_{t\star}$ gradient (4.4) is then obtained in $\mathcal{O}(N \log N)$ average time [vdM14], and can be used in gradient-based minimization schemes.

Figure 4.1 illustrates the computation of $\tilde{\mathbf{g}}_v^t$ and $\tilde{g}_v^t$. The square data point corresponds to $\mathbf{x}_v$ in Fig. 4.1b and 4.1c. Figure 4.1b computes $\mathbf{g}_v^t$ and g_v^t exactly, leading to $\mathcal{O}(N^2)$ time complexity when performed for all $v \in \mathcal{I}$. Figure 4.1c sketches their BH approximation in $\mathcal{O}(\log N)$ average time, in which the centers-of-mass of the LD points in farther cells to $\mathbf{x}_v$ summarize their contributions in $\mathbf{g}_v^t$ and g_v^t .

4.2 Extensive quality assessment of Barnes-Hut t -SNE

While the quadratic time complexity of t -SNE in the number N of samples limits its applicability to moderate-size data sets, its BH acceleration [vdM14] approximates the gradient (4.4) in $\mathcal{O}(N \log N)$ time, even though it introduces a new threshold hyper-parameter θ . Although some works reported impressive computation time gains over t -SNE [vdM14], surprisingly few studies analyzed its DR quality. Apart from depicting some data sets LD representations, [vdM14] only measures the 1-nearest neighbor errors of the embeddings of a single labeled data set, for several thresholds and one perplexity. Being solely applicable to labeled data set, this measure at best reveals the nearest HD neighbors retrieval and is at worst unrelated to neighborhood preservation [dBMLS⁺19]. Thus it hardly quantifies the HD clusters preservation in the LD embedding and the HD manifolds unrolling. These are instead well described by analyzing whether larger than 1-size neighborhoods are preserved as well [LV09, LV10].

Therefore, in the sake of designing fast accelerations of multi-scale NE algorithms in Section 4.3, this section first examines the BH t -SNE quality on numerous real-world and artificial data sets, using various perplexities K_* and BH thresholds θ , thanks to neighborhood-based DR performance criteria [LV09]. The aim of this section is hence to assess the suitability of the BH algorithm in the single-scale NE setting, to verify whether it can be a suitable basis for the developments carried in Section 4.3. Since the running time behavior of BH t -SNE with respect to t -SNE is already thoroughly reported in the literature [vdM14, YPK13], this section focuses on analyzing their differences in terms of achieved DR quality. Further performance comparisons in both terms of LD embedding goodness and computation times are presented in Section 4.4. The experimental results of the current section suggest that t -SNE and its BH version perform equally well, the latter being sometimes slightly superior to the former, except for some extremely large perplexities K_* . As small ones are typically preferred in t -SNE, this barely matters in practice. Moreover the DR quality is almost independent of the BH threshold θ . These results combined with its reduced time complexity allow the BH acceleration to outperform t -SNE in all aspects. Its usage can hence only be encouraged.

As detailed in Section 3.3.1, DR quality is usually related to the preservation of the HD neighborhoods in the LD space [VPN⁺10, FC07, CB09, MLGH13, MGL⁺12, MGH10, MGH11, GFD16]. It can be evaluated thanks to the R_{NX} and AUC criteria presented in Section 3.3.1 [LV09, LV⁺08], which are computed in $\mathcal{O}(N^2 \log N)$ time [LV10]. The experiments reported in this section are hence also based on these measures, to quantify

DR quality both locally and globally [LPOV15].

To study diversified public databases, the performance comparison of t -SNE and its BH acceleration is conducted on the following data sets:

- COIL-20 (COIL) [NNM96];
- B. Frey’s images (B.Freys);
- Olivetti face images (Oliv.);
- The test set of the MNIST images of handwritten digits (MNIST) [LBBH98];
- A 3-D artificial data set (Sph) with 3000 dots randomly distributed on a unit-radius sphere;
- A 3-D artificial data set (Clusters) with 3000 dots randomly distributed on eight equal-size Gaussian clusters;
- A 3-D artificial data set (Torus) with 3000 dots randomly distributed on a torus.

In addition, various UCI data sets [Lic13] are considered as well:

- Glass Identification (GId.);
- Breast Tissue (BrTis.);
- Ecoli;
- Wine;
- Parkinsons (Park.) [LMR⁺07];
- Yacht Hydrodynamics (YaHyd.);
- Seeds;
- Concrete Slump Test (ConcSl.) [Yeh07];
- Servo;
- Optical Recognition of Handwritten Digits test set (Digits);
- Haberman’s Survival (Hab.);
- Iris;
- Fertility (Fert.) [GGDJ⁺12];

- Planning Relax (PIRel.) [Bha12];
- Leaf [SMdS13];
- Wholesale customers (WholC.) [Abr11];
- Istanbul Stock Exchange (IstSt.) [ABB14];
- User Knowledge Modeling (UsKn.) [KSC13];
- Indian Liver Patient (ILP);
- SPECTF Heart (SPECTF);
- Concrete Compressive Strength (CCS) [Yeh98];
- Ionosphere (Iono.);
- Connectionist Bench on Vowel Recognition and Deterding Data (ConV.);
- Breast Cancer Wisconsin Diagnostic (BCW).

Figure 4.2 indicates the numbers N of samples and I of features for all the data sets, which are of moderate size to enable handily applying t -SNE. Z-score standardization [Bis06] is applied to the UCI data sets when the ranges of their features differ. Potential data labels are ignored during the DR. The employed optimization schemes and experimental setup are as described in [vdMH08, vdM14]: the LD dimension P is set to 2 and momentum gradient descent with adaptive step size [Jac88] is performed with early exaggeration. Initial LD coordinates are sampled from an origin-centered isotropic Gaussian with 10^{-4} variance, with fixed random seed to facilitate the comparison of the performances. Euclidean distances are used in both the HD and LD spaces. The gradient descent is iterated until the gradient infinite norm vanishes or the objective function do not evolve anymore. The embedding achieving the smallest objective function value is kept. Optimization parameter values are derived from [vdMH08, vdM14].

The considered values of the BH threshold θ range from 0.05 to 0.95 with a 0.05 step. Unlike [vdMH08, vdM14], different perplexities are studied, expressed both absolutely and relatively with respect to the number N of samples. Absolute perplexities are usually employed in the literature, while relative ones aim to ease comparison across data sets. All perplexities are in any case kept above 2; whenever a relative perplexity induces on a data set with small N a value smaller than 2, it is hence forced back to 2. Moreover for each data set, only the absolute perplexities smaller than its number N of samples are considered. Perplexities larger than N would indeed yield nearly uniform HD similarities (2.13).

Figure 4.2 displays the difference in AUC between BH t -SNE and t -SNE for the different data sets, perplexities and BH thresholds. Positive values hence indicates the BH scheme superiority over t -SNE. Observing the y -axis scales of the plots and as the AUC ranges in $[-1, 1]$, it is first noteworthy that the BH acceleration performs extremely similarly to t -SNE, sometimes even slightly better, except occasionally for very large perplexities compared to N . This is, however, hardly detrimental as rather small perplexities are employed in the literature. Second, although a larger threshold θ implies cruder BH estimations of the gradient (4.4), the LD embedding quality is almost independent of θ , most curves being close to horizontal except for some large perplexities. These results suggest that one only needs to set the BH threshold θ to a reasonable value, such as 0.5, to obtain a faithful approximation of t -SNE.

The obtained conclusions also hold when computing the retrieval rate $Q_{\text{NX}}(1)$ of the nearest HD neighbors instead of the AUC in Fig. 4.3. The $Q_{\text{NX}}(1)$ score is a measure which is closer to the 1-nearest neighbor errors in the LD embeddings which are evaluated in [vdM14]. Nevertheless, as $Q_{\text{NX}}(1)$ is limited to quantifying the very local performances of the methods, the results depicted in Fig. 4.3 are inevitably slightly less stable than in Fig. 4.2.

The individual R_{NX} curves and LD embeddings can also be analyzed for each data set, perplexity and BH threshold, yielding the same conclusions as previously observed. For conciseness, these curves and embeddings are not displayed for all data sets but only three of them, namely the MNIST test set (MNIST) [LBBH98], the COIL-20 data set (COIL) [NNM96] and the B. Frey's images (B.Freys). Figure 4.4 illustrates some samples of these three image databases. Figure 4.5, 4.6 and 4.7 depict the difference of the R_{NX} curves of BH t -SNE and t -SNE as a function of the neighborhood size K on MNIST, COIL and B.Freys. Similarly to the results of Fig. 4.2, BH t -SNE and t -SNE deliver comparable curves. For some perplexities, the methods slightly differ for the preservation of local neighborhoods; the reproduction of structures at small scales is indeed likely to be more sensitive to changes in the definitions of the algorithms. In this line, Fig. 4.8, 4.9 and 4.10 enables to more easily observe this phenomenon, while stressing the striking similitude between BH t -SNE and t -SNE. This resemblance is also emphasized in Fig. 4.11, as well as the appreciable stability of the performances of BH t -SNE with respect to the threshold θ .

Finally, Fig. 4.12 and 4.13 display the 2-D embeddings respectively obtained on MNIST and COIL, by t -SNE and BH t -SNE, using a perplexity $K_* = 50$ and a threshold $\theta = 0.5$ for BH t -SNE. The visual inspection of these LD representations also supports the above conclusions: the 2-D spaces computed by t -SNE and BH t -SNE are extremely similar. One

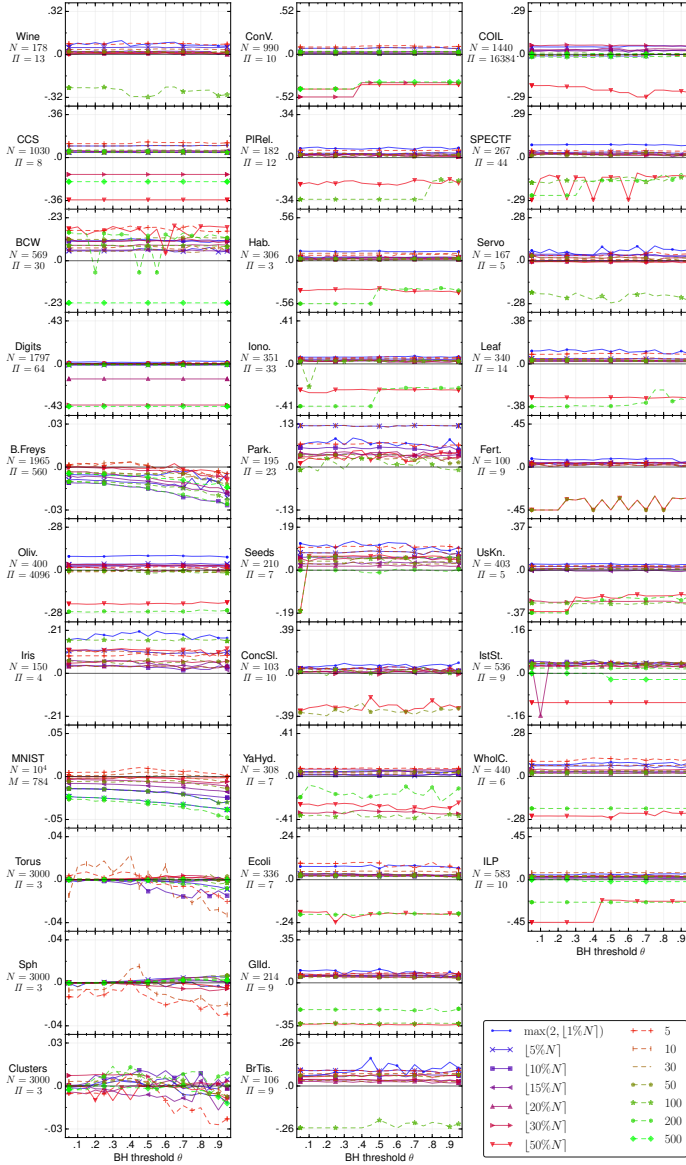


Figure 4.2: Difference in terms of AUC between BH t -SNE and t -SNE as a function of the threshold θ , ranging from .05 to .95 with a .05 step, on all data sets. Positive (negative) values show that the BH scheme out- (under-) performs t -SNE. The legend lists all the studied perplexities K_* . Each perplexity K_* is used only on the data sets with $N > K_*$.

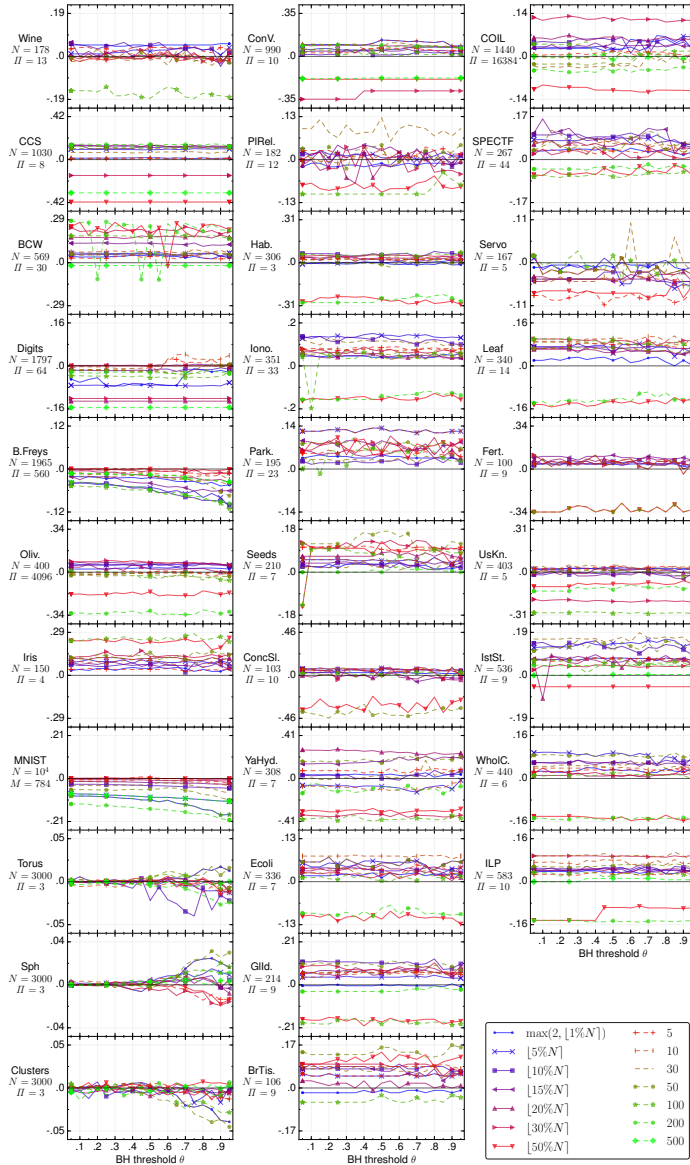
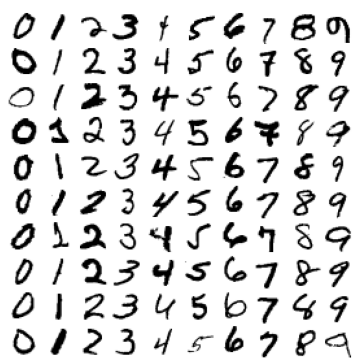
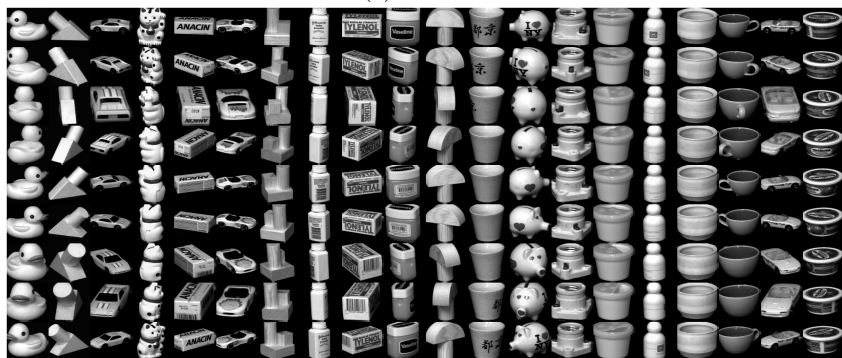


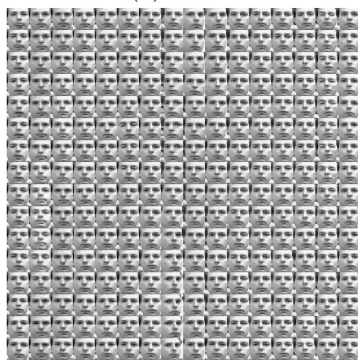
Figure 4.3: Difference in terms of nearest HD neighbors retrieval rate $Q_{NX}(1)$ between BH t -SNE and t -SNE as a function of θ , ranging from .05 to .95 with a .05 step, on all data sets. Positive (negative) values show that the BH scheme out- (under-) performs t -SNE. The legend lists the studied perplexities K_* . Each K_* is used only on the data sets with $N > K_*$.



(a) MNIST.



(b) COIL.



(c) B.Freys.

Figure 4.4: (a) A few samples of the MNIST images of handwritten digits. (b) A few samples of the COIL images of rotated objects. (c) A few samples of the B.Freys images of facial expression of Brendan Frey.

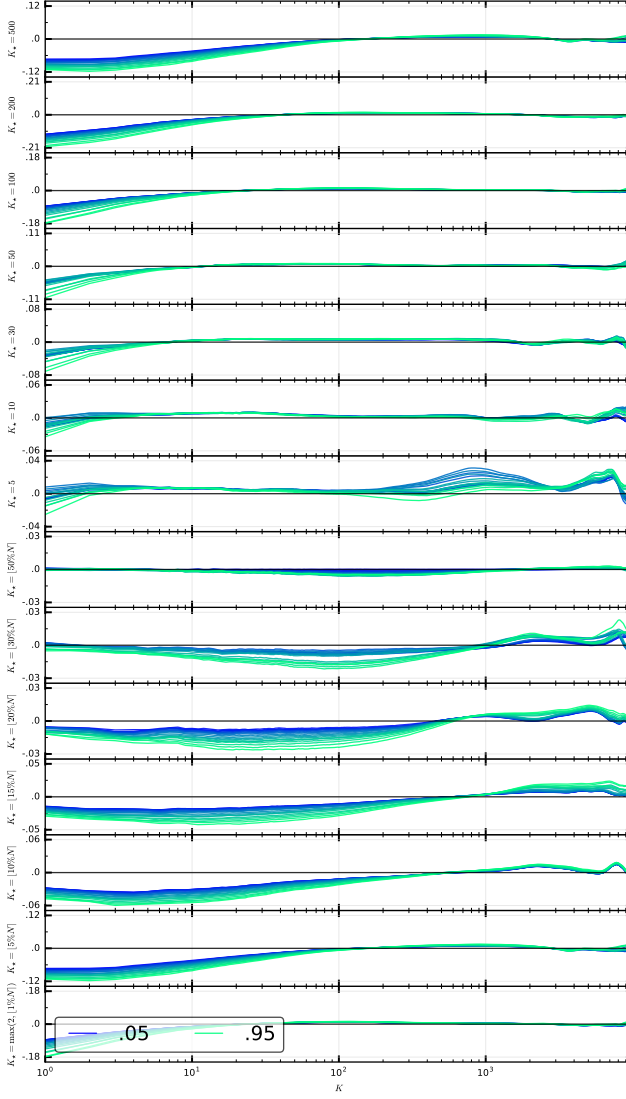


Figure 4.5: Difference of the R_{NX} curves of BH t -SNE and t -SNE as a function of the neighborhood size K on MNIST. Positive (negative) values show that the BH scheme out- (under-) performs t -SNE. For each perplexity K_* indicated on the y -axis, the threshold θ ranges from .05 to .95 with a .05 step, yielding 19 curves. The legend depicts the colors of the curves for θ equal to .05 and .95; the remaining curves follow a linear colormap between these two extreme θ values.

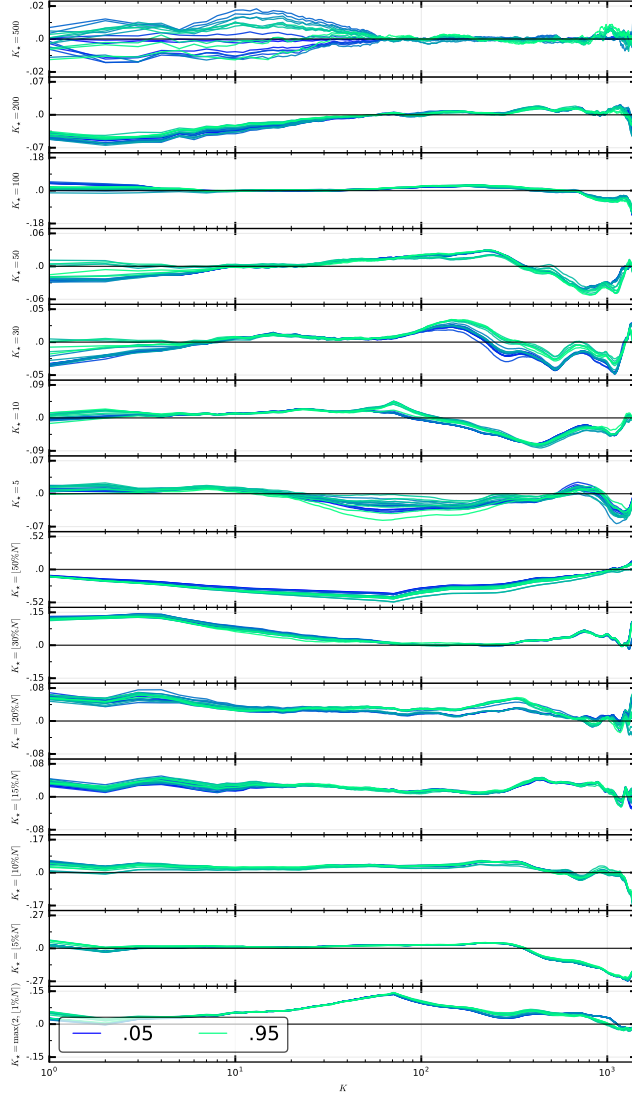


Figure 4.6: Difference of the R_{NX} curves of BH t -SNE and t -SNE as a function of the neighborhood size K on COIL. Positive (negative) values show that the BH scheme out- (under-) performs t -SNE. For each perplexity K_* indicated on the y -axis, the threshold θ ranges from .05 to .95 with a .05 step, yielding 19 curves. The legend depicts the colors of the curves for θ equal to .05 and .95; the remaining curves follow a linear colormap between these two extreme θ values.

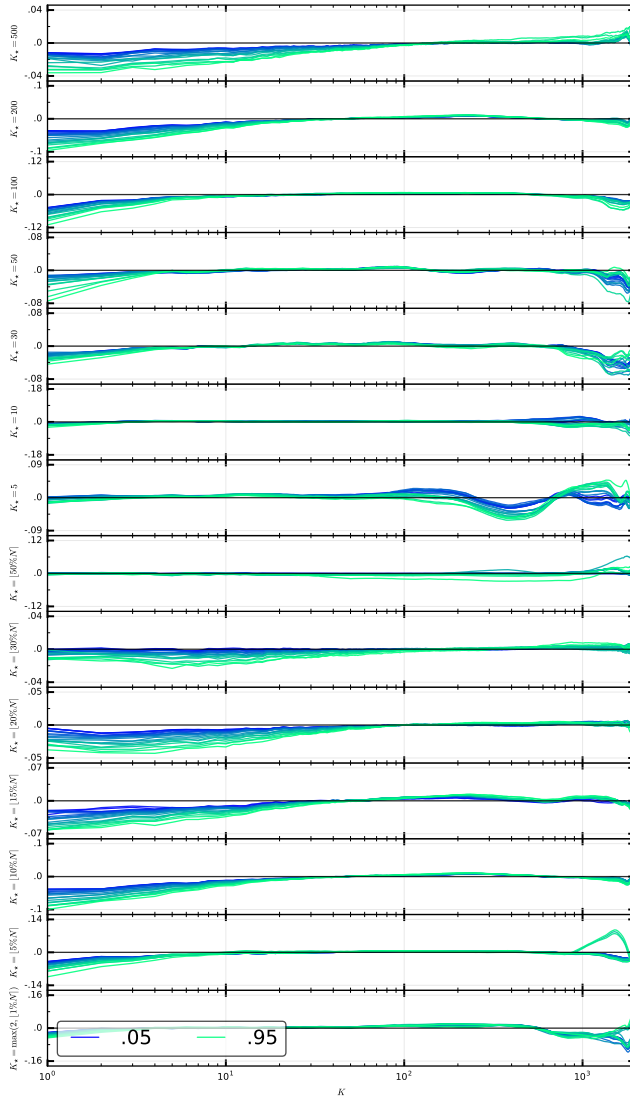
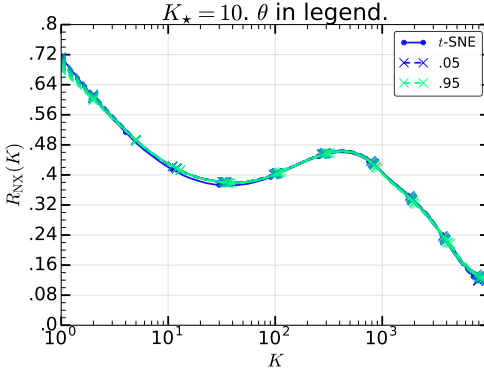
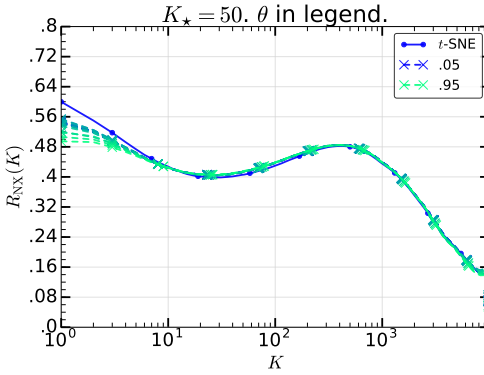


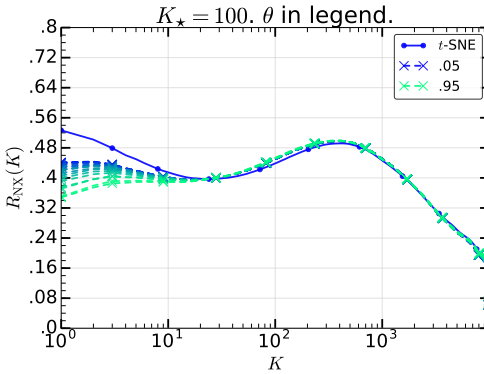
Figure 4.7: Difference of the R_{NX} curves of BH t -SNE and t -SNE as a function of the neighborhood size K on B.Freys. Positive (negative) values show that the BH scheme out- (under-) performs t -SNE. For each perplexity K_* indicated on the y -axis, the threshold θ ranges from .05 to .95 with a .05 step, yielding 19 curves. The legend depicts the colors of the curves for θ equal to .05 and .95; the remaining curves follow a linear colormap between these two extreme θ values.



(a)



(b)



(c)

Figure 4.8: (a) The R_{NX} curves of BH t -SNE and t -SNE as a function of the neighborhood size K on MNIST, using a perplexity $K_\star = 10$. For each K , the highest curve determines the method achieving the best DR performances. The BH threshold θ ranges from .05 to .95 with a .05 step, yielding 19 curves for BH t -SNE. The legend depicts the colors of the BH t -SNE curves for θ equal to .05 and .95; the remaining BH t -SNE curves follow a linear colormap between these two extreme θ values. (b)-(c) Similar to (a) but respectively using a perplexity $K_\star$ equal to 50 and equal to 100.

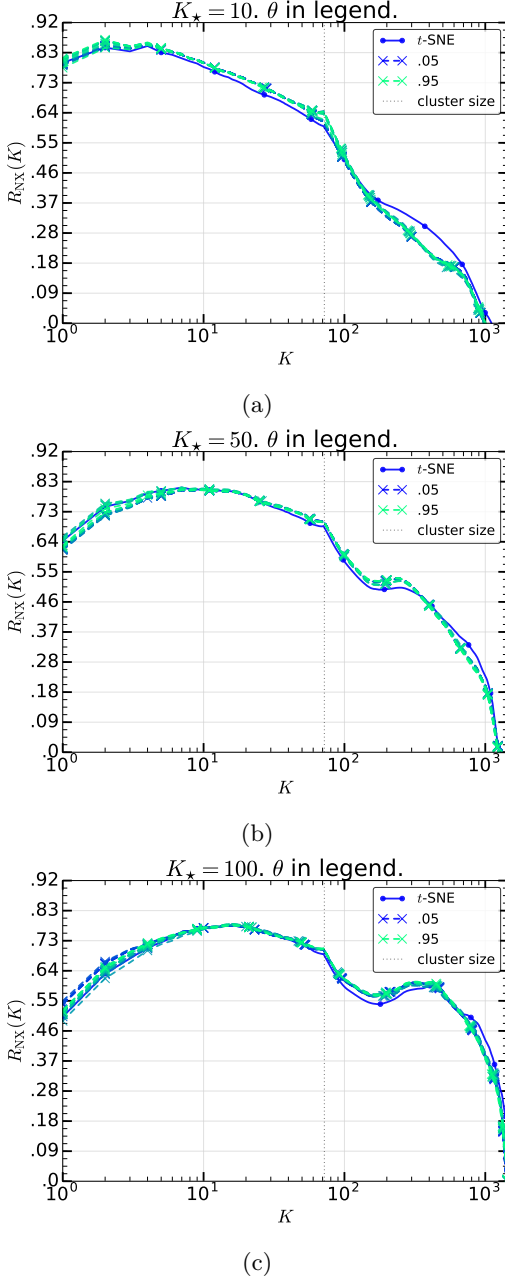
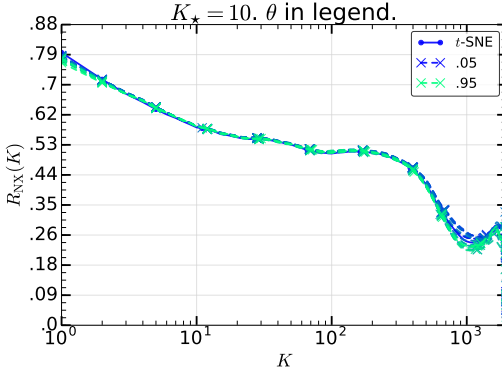
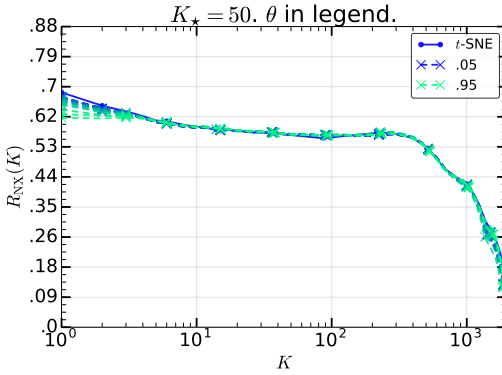


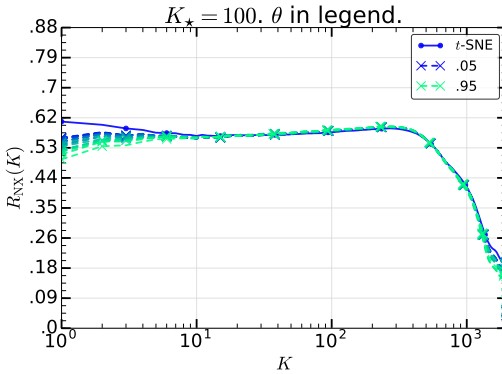
Figure 4.9: (a) The R_{NX} curves of BH t -SNE and t -SNE as a function of the neighborhood size K on COIL, using a perplexity $K_* = 10$. For each K , the highest curve determines the method achieving the best DR performances. The BH threshold θ ranges from .05 to .95 with a .05 step, yielding 19 curves for BH t -SNE. The legend depicts the colors of the BH t -SNE curves for θ equal to .05 and .95; the remaining BH t -SNE curves follow a linear colormap between these two extreme θ values. The vertical dotted line highlights the neighborhood size K value equal to the number of images per object in the COIL database, which is 72. This number provides an intuition of the size of the clusters that can be expected in this HD data set. (b)-(c) Similar to (a) but respectively using a perplexity K_* equal to 50 and equal to 100.



(a)

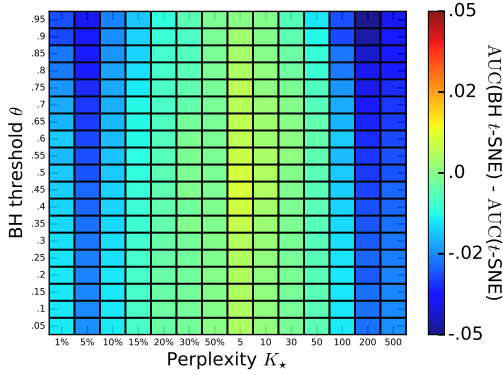


(b)

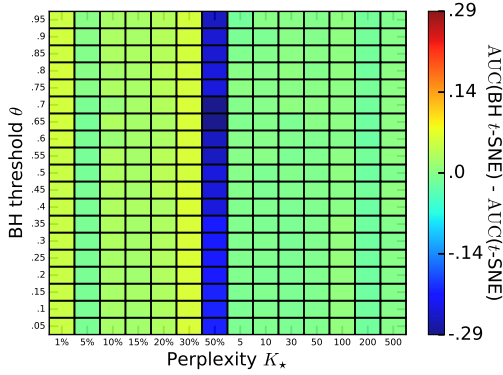


(c)

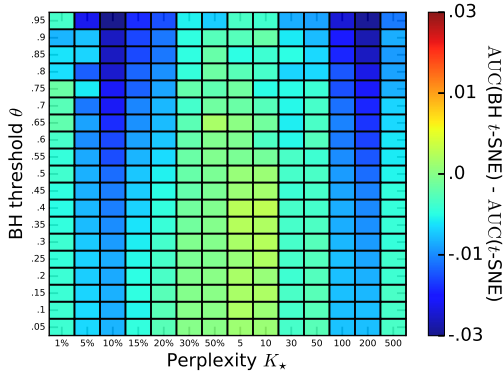
Figure 4.10: (a) The R_{NX} curves of BH t -SNE and t -SNE as a function of the neighborhood size K on B.Freys, using a perplexity $K_* = 10$. For each K , the highest curve determines the method achieving the best DR performances. The BH threshold θ ranges from .05 to .95 with a .05 step, yielding 19 curves for BH t -SNE. The legend depicts the colors of the BH t -SNE curves for θ equal to .05 and .95; the remaining BH t -SNE curves follow a linear colormap between these two extreme θ values. (b)-(c) Similar to (a) but respectively using a perplexity K_* equal to 50 and equal to 100.



(a) MNIST.



(b) COIL.



(c) B.Freys.

Figure 4.11: (a) Heatmap of the difference in terms of AUC between BH t -SNE and t -SNE as a function of the perplexity K_* and of the threshold θ , ranging from .05 to .95 with a .05 step, on MNIST. Positive (negative) values show that the BH scheme out- (under-) performs t -SNE. On the x -axis, the notation $x_{K_*}\%$ indicates that a perplexity of $\max(2, \lfloor x_{K_*} N / 100 \rfloor)$ is employed. (b)-(c) Similar to (a) but respectively on COIL and B.Freys.

can notice in Fig. 4.12 that the HD clusters corresponding to the different MNIST digits are nicely retrieved. Also, in Fig. 4.13, nearly all objects are mapped on 1-D manifolds; such a result is relevant for the COIL database as the HD points associated with each object are distributed on a HD ring, since the lone degree of freedom is the rotation angle of the object images.

Besides, the investigation of these LD embeddings highlights the usefulness of relying on the R_{NX} and AUC quality criteria. They indeed conveniently avoid confronting the t -SNE and BH t -SNE embeddings for all perplexities, BH thresholds and data sets, while providing quantitative description of the performances instead of a qualitative one.

The neighborhood-based DR quality results of this section on numerous real-world and artificial databases hence tend to show that t -SNE and its fast BH version perform extremely similarly, suggesting the BH scheme superiority thanks to its reduced time complexity. Indeed, the Zoutendijk condition also further supports the above observations [Zou60]: as long as the dot product between the true gradient (4.4) and its BH estimate remains positive, gradient descent with adequate step sizes leads to a stationary point of the cost function (2.15). The BH method hence improves t -SNE in all respects, and its usage can thus be preferred [dBML18a]. For these reasons, the BH algorithm seems to be appropriate to accelerate neighbor embedding methods. This motivates designing fast multi-scale NE techniques partly based on this algorithm, as presented in Section 4.3.

4.3 Accelerating multi-scale neighbor embedding methods

A crucial aspect of fast single-scale methods stems from their sparse HD similarities. These rely on nearest neighbor sets $\mathcal{I}_{i\star}$ for each $i \in \mathcal{I}$, with $|\mathcal{I}_{i\star}| = \lfloor 3K_\star \rfloor$. A vantage-point tree [Yia93] computes these sets in $\mathcal{O}(K_\star N \log N)$ time, which reduces to $\mathcal{O}(N \log N)$ if $K_\star \ll N$.

Multi-scale methods combine single-scale similarities tuned at several data scales, employing exponentially growing perplexities. In particular, K_H lies in the $\mathcal{O}(N)$ range, leading to neighbor sets $\mathcal{I}_{iH}$ with $\mathcal{O}(N)$ size for each $i \in \mathcal{I}$, since $|\mathcal{I}_{iH}| = \min(\lfloor 3K_H \rfloor, N - 1)$. As multi-scale HD similarities average the single-scale ones in (2.25), full multi-scale neighbor sets are derived as

$$\mathcal{I}_i = \cup_{h=1}^H \mathcal{I}_{ih} = \mathcal{I} \setminus \{i\}, \quad (4.11)$$

to gather the neighbors from all scales. Therefore, the multi-scale HD similarity matrix determined by these neighbor sets has $\mathcal{O}(N^2)$ non-zero entries. This prevents designing sparse multi-scale HD similarities as in the single-scale framework.

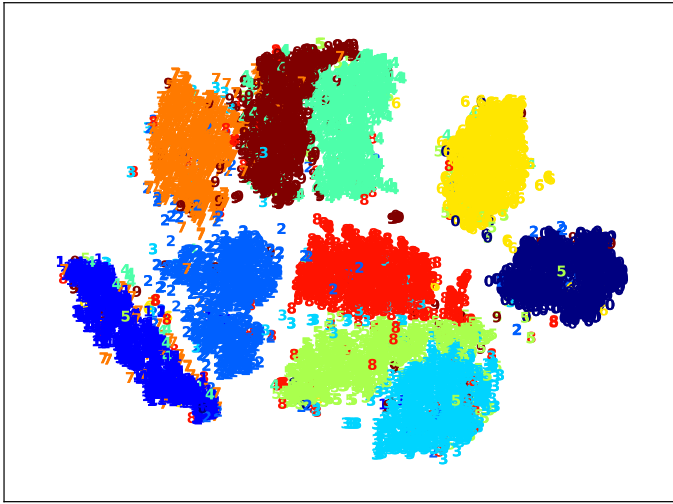
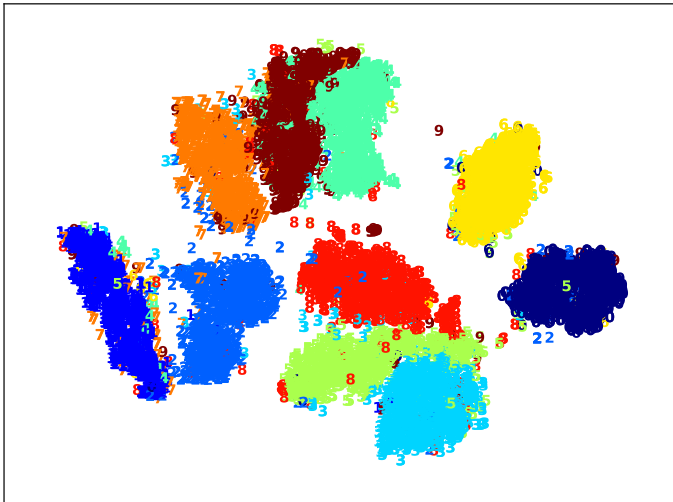
(a) t -SNE ($K_* = 50$).(b) BH t -SNE ($K_* = 50$, $\theta = 0.5$).

Figure 4.12: (a) 2-D embedding of MNIST obtained by t -SNE using a perplexity $K_* = 50$. Each data point corresponds to an image of a handwritten digit; the digit associated with an image is employed to depict its LD coordinates. (b) Similar to (a) but for the 2-D embedding of MNIST obtained by BH t -SNE using a perplexity $K_* = 50$ and a threshold $\theta = 0.5$.

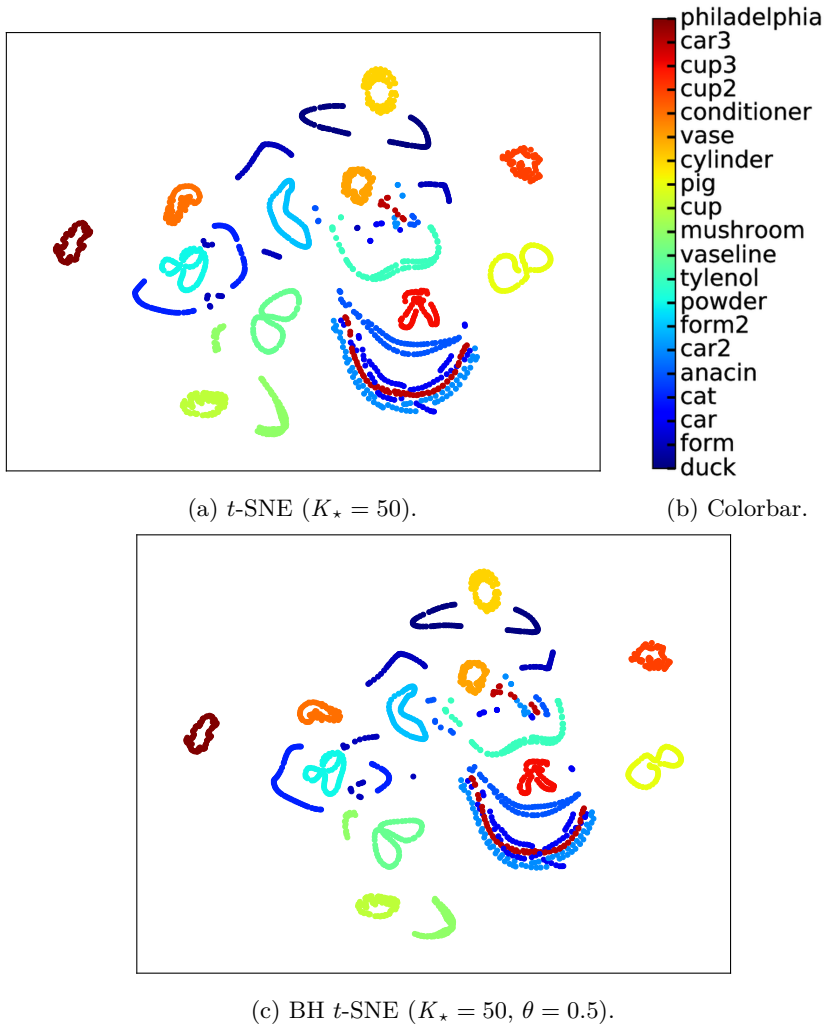


Figure 4.13: (a) 2-D embedding of COIL obtained by t -SNE using a perplexity $K_* = 50$. Each data point corresponds to an image of an object. (b) Colorbar indicating the colors used to depict the LD representations of the COIL objects in (a) and (c). (c) Similar to (a) but for the 2-D embedding of COIL obtained by BH t -SNE using a perplexity $K_* = 50$ and a threshold $\theta = 0.5$.

Single-scale HD similarities with small perplexities excel at modeling local interactions around each data point, while reducing the influence of farther away HD objects. They indeed concentrate their probability mass on the nearest neighbors of each datum, which can hence in turn serve as a remarkable basis to produce sparse approximations. As local structures often rapidly fluctuate, it is in fact crucial to consider the closest neighbors to guarantee the faithfulness of the sparse estimates when processing small scales. On the other hand, single-scale similarities based on large perplexities capture global, large scale interactions in the data. As these global structures emerge from average trends across important amounts of data points, it is little convincing that their estimation requires exhaustive consideration of all samples. Indeed, whereas local interactions are short range and induced by very specific sets of neighbors, global ones result from overall consensuses which can be fairly well approximated using appropriate subsets of data points. In the cost functions of neighbor embedding methods, evaluating the divergences between probability distributions defined on such subsets may reveal sufficient to relevantly shape the LD embedding according to global HD properties.

However, multi-scale similarities combine both types of small and large perplexities, respectively focusing on local and global HD characteristics. As a consequence, to faithfully accelerate multi-scale methods, the design of small and relevant multi-scale neighbor sets, denoted by $\tilde{\mathcal{I}}_i$ for $i \in \mathcal{I}$, entails purposely gathering data points which reasonably model the whole data cloud, while at the same time accurately target the nearest HD neighbors of each ξ_i . The computation of NE cost functions could then be restricted to these subsets $\tilde{\mathcal{I}}_i$ for each $i \in \mathcal{I}$, which would efficiently score the adequacy of the LD coordinates $\mathbf{x}_i$ in the multi-scale spirit, by accounting for both global trends and local interactions in the data.

The hierarchical structure of the multi-scale HD similarities, from small to large perplexities, suggests that the set $\tilde{\mathcal{I}}_i$ should gather data points lying at different ranges of ξ_i . Indeed, as small scales require fine levels of details, $\tilde{\mathcal{I}}_i$ must precisely contain the very close neighbors of ξ_i . But as the scale index increases, only rougher HD characterization is required and a few farther data points from ξ_i can be sampled, providing a higher resolution view. Inspired by this intuition, one can first define a hierarchy $\{\Xi_h\}_{h=1}^H$ of sub-sampled HD data sets at each scale, where Ξ_h is a random sample of Ξ with $\lfloor 2^{1-h}N \rfloor$ elements drawn without replacement. Note that $\Xi_1 = \Xi$. This collection of HD data sets is illustrated in Fig. 4.14.

For each $h \in \{1, \dots, H\}$, a vantage-point tree [Yia93] can be efficiently created on Ξ_h in

$$\mathcal{O} \left(\frac{N}{2^{h-1}} \log \left(\frac{N}{2^{h-1}} \right) \right) \quad (4.12)$$

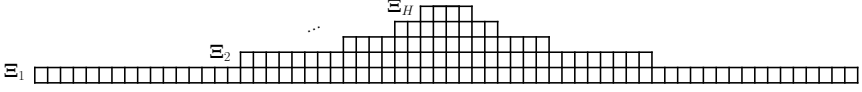


Figure 4.14: Hierarchy of HD data sets at all scales. A square refers to a HD sample.

time. All the trees are then defined in

$$\mathcal{O} \left(\sum_{h=1}^H \frac{N}{2^{h-1}} \log \frac{N}{2^{h-1}} \right) = \mathcal{O} \left(N \log N \sum_{h=1}^H \frac{1}{2^h} \right) \quad (4.13)$$

$$= \mathcal{O}(N \log N) \quad (4.14)$$

time. The h^{th} tree enables computing the set $\tilde{\mathcal{I}}_{ih}$ of the $\lfloor 3K_1 \rfloor$ nearest neighbors of ξ_i in $\Xi_h \setminus \{\xi_i\}$ in

$$\mathcal{O} \left(K_1 \log \left(\frac{N}{2^{h-1}} \right) \right) \quad (4.15)$$

time. The sets $\tilde{\mathcal{I}}_{ih}$ for all $i \in \mathcal{I}$ and $h \in \{1, \dots, H\}$ are hence obtained in $\mathcal{O}(N \log^2 N)$ time, as $K_1 = 2 \ll N$ and as there are $\mathcal{O}(\log N)$ scales. It is worth mentioning that, on average, $\tilde{\mathcal{I}}_{ih}$ sub-samples the set $\mathcal{I}_{ih}$ of the $\lfloor 3K_h \rfloor$ nearest neighbors of ξ_i in $\Xi \setminus \{\xi_i\}$, as $\tilde{\mathcal{I}}_{ih}$ is based on Ξ_h which contains a fraction

$$2^{1-h} = \frac{K_1}{K_h} \quad (4.16)$$

of Ξ .

Defining $\tilde{\mathcal{I}}_i := \cup_{h=1}^H \tilde{\mathcal{I}}_{ih}$ then follows the above intuition: the smallest scale is represented by the exact $\lfloor 3K_1 \rfloor$ nearest neighbors of ξ_i in $\Xi \setminus \{\xi_i\}$. But as the scale grows, Ξ_h more and more roughly summarizes Ξ in such a way that the accounted neighbors $\tilde{\mathcal{I}}_{ih}$ are more and more dispersed in the data cloud in the sake of capturing its global properties. The size of the multi-scale neighbor set $\tilde{\mathcal{I}}_i$ evolves as $\mathcal{O}(\log N)$. Note that depending on N and K_* , $|\tilde{\mathcal{I}}_i|$ may be smaller than $|\mathcal{I}_{i*}|$.

The sets $\tilde{\mathcal{I}}_i$ can be employed to define sparse HD similarities focusing on both local and global data relationships. At scale $h \in \{1, \dots, H\}$, one can establish, for $i, j \in \mathcal{I}$,

$$\tilde{\sigma}_{ijh} = \begin{cases} \frac{\exp(-\tilde{\pi}_{ih} \delta_{ij}^2 / 2)}{\sum_{k \in \tilde{\mathcal{I}}_i} \exp(-\tilde{\pi}_{ih} \delta_{ik}^2 / 2)} & \text{if } j \in \tilde{\mathcal{I}}_i \\ 0 & \text{otherwise} \end{cases}, \quad (4.17)$$

with

$$\tilde{\pi}_{ih} \text{ s.t. } \log K_1 = - \sum_{j \in \tilde{\mathcal{I}}_{ih}} \tilde{\sigma}_{ijh}^\pi \log \tilde{\sigma}_{ijh}^\pi, \quad (4.18)$$

where

$$\tilde{\sigma}_{ijh}^\pi = \begin{cases} \frac{\exp(-\tilde{\pi}_{ih} \delta_{ij}^2/2)}{\sum_{k \in \tilde{\mathcal{I}}_{ih}} \exp(-\tilde{\pi}_{ih} \delta_{ik}^2/2)} & \text{if } j \in \tilde{\mathcal{I}}_{ih} \\ 0 & \text{otherwise} \end{cases}. \quad (4.19)$$

Precision $\tilde{\pi}_{ih}$ is hence fixed in (4.18) thanks to the neighbors $\tilde{\mathcal{I}}_{ih}$.

Sparse multi-scale HD similarities then follow as, for $i, j \in \mathcal{I}$,

$$\tilde{\sigma}_{ij} = \frac{1}{H} \sum_{h=1}^H \tilde{\sigma}_{ijh}, \quad \tilde{\tau}_{ij} = \frac{\tilde{\sigma}_{ij} + \tilde{\sigma}_{ji}}{2N}. \quad (4.20)$$

Their matrices have $\mathcal{O}(N \log N)$ non-zero entries, all derived in $\mathcal{O}(N \log^2 N)$ total time, since $|\tilde{\mathcal{I}}_i|$ scales as $\mathcal{O}(\log N)$.

Sparse multi-scale LD similarities could be defined in a similar way. To match them with (4.20), they should rely on the same sets $\tilde{\mathcal{I}}_i$ for $i \in \mathcal{I}$. Therefore, for each scale $h \in \{1, \dots, H\}$, $i \in \mathcal{I}$ and $j \in \mathcal{I}$, they would develop as

$$\tilde{s}_{ij} = \frac{1}{H} \sum_{h=1}^H \tilde{s}_{ijh}, \quad (4.21)$$

with

$$\tilde{s}_{ijh} = \begin{cases} \frac{\exp(-p_{ih} d_{ij}^2/2)}{\sum_{k \in \tilde{\mathcal{I}}_i} \exp(-p_{ih} d_{ik}^2/2)} & \text{if } j \in \tilde{\mathcal{I}}_i \\ 0 & \text{otherwise} \end{cases}. \quad (4.22)$$

Each set $\tilde{\mathcal{I}}_i$ with $i \in \mathcal{I}$ could possibly be extended with $\mathcal{O}(\log N)$ elements to capture the local and global properties of the LDS, similarly to the sparse multi-scale HD similarities (4.20) which model the local and global structures of the HDS. A hierarchy $\{\mathbf{X}_h\}_{h=1}^H$ of sub-sampled LD data sets at each scale would hence be defined, where $\mathbf{X}_h$ is a random sample of $\mathbf{X}$ with $\lfloor 2^{1-h} N \rfloor$ elements drawn without replacement. For each $h \in \{1, \dots, H\}$, a vantage-point tree would then be created on $\mathbf{X}_h$ to efficiently determine, for each $i \in \mathcal{I}$, the $\lfloor 3K_1 \rfloor$ nearest neighbors of $\mathbf{x}_i$ in $\mathbf{X}_h \setminus \{\mathbf{x}_i\}$. Thus, for each $i \in \mathcal{I}$, the set $\tilde{\mathcal{I}}_i$ would be expanded with $\mathcal{O}(\log N)$ additional elements, consisting of the $\lfloor 3K_1 \rfloor$ nearest neighbors of $\mathbf{x}_i$ in $\mathbf{X}_h \setminus \{\mathbf{x}_i\}$ for $h \in \{1, \dots, H\}$. It is noteworthy that these $\mathcal{O}(\log N)$ supplementary elements should be iteratively updated according to the LD embedding modification after each optimization step.

Unfortunately, experiments revealed that minimizing the KL divergence between (4.20) and (4.21) leads to LD crushing phenomena, meaning that the LD coordinates tend to overlap. Indeed, the size of each set $\tilde{\mathcal{I}}_i$ for $i \in \mathcal{I}$ remains limited to $\mathcal{O}(\log N)$, whether adding the above mentioned $\mathcal{O}(\log N)$ extra elements or not. Therefore, it is likely that $\tilde{\mathcal{I}}_k \cap \tilde{\mathcal{I}}_l = \emptyset$ for some $k, l \in \mathcal{I}$. In this case, the LD similarities (4.22) based on the sets $\tilde{\mathcal{I}}_k$ and $\tilde{\mathcal{I}}_l$ would make the LD coordinates $\mathbf{x}_k$ and $\mathbf{x}_l$ independent one to another, since they would not be considered in the denominators of both their sparse HD and LD similarities (4.17) and (4.22). The resulting lack of LD repulsive forces between them hence enables their LD superposition. Extending this reasoning to a potentially very large number of data point pairs leads to crushed LD embeddings, with poor HD neighborhood preservation.

On the other hand, LD interactions can be efficiently estimated with inspiration from the single-scale setting. This motivates defining hybrid cost functions for fast versions of multi-scale SNE and t -SNE as

$$\tilde{C} = - \sum_{i \in \mathcal{I}, j \in \mathcal{I} \setminus \{i\}} \tilde{\sigma}_{ij} \log s_{ij} \quad (4.23)$$

and

$$\tilde{C}_t = - \sum_{i \in \mathcal{I}, j \in \mathcal{I} \setminus \{i\}} \tilde{\tau}_{ij} \log t_{ij}, \quad (4.24)$$

which quantify the mismatch between the sparse multi-scale HD similarities (4.20) and the LD ones of genuine multi-scale SNE and t -SNE. In (4.23) and (4.24), the HD similarities are normalized only through the data points in $\tilde{\mathcal{I}}_i$ for $i \in \mathcal{I}$ whereas all samples are involved in the LD similarity normalization. Matching these HD and LD similarities then decreases the amplitude of the additional terms in the denominators of the LD similarities, favoring LD spreading, i.e. the tendency to expand the LD space, reported as beneficial to DR quality [vdMH08].

After computing (4.20), a Barnes-Hut (BH) algorithm [BH86] is adapted from Section 4.1 to estimate (4.24) as well as its gradient in $\mathcal{O}(N \log N)$ average time. As to (4.23) and its gradient, we develop their BH approximation in $\mathcal{O}(N \log^2 N)$ average time. It is detailed below for the gradient of (4.23), from which the BH estimation of (4.23) itself can be easily derived.

The gradient of (4.23) with respect to the LD points writes

$$\frac{\partial \tilde{C}}{\partial \mathbf{x}_v} = \frac{1}{H} \sum_{i \in \mathcal{I} \setminus \{v\}} \underbrace{\left(\sum_{h=1}^H p_h s_{vih} \right)}_{f_{vi}} \frac{\tilde{\sigma}_{vi}}{s_{vi}} (\mathbf{x}_v - \mathbf{x}_i) \quad (4.25)$$

$$- \frac{1}{H} \sum_{h=1}^H p_h \underbrace{\left[\sum_{i \in \mathcal{I} \setminus \{v\}} \frac{\tilde{\sigma}_{vi}}{s_{vi}} s_{vih} \right]}_{g_{vh}} \underbrace{\left[\sum_{i \in \mathcal{I} \setminus \{v\}} s_{vih} (\mathbf{x}_v - \mathbf{x}_i) \right]}_{\mathbf{g}_{vh}} \quad (4.26)$$

$$+ \frac{1}{H} \sum_{i \in \mathcal{I} \setminus \{v\}} \underbrace{\left(\sum_{h=1}^H p_h s_{ivh} \right)}_{f_{iv}} \frac{\tilde{\sigma}_{iv}}{s_{iv}} (\mathbf{x}_v - \mathbf{x}_i) \quad (4.27)$$

$$- \frac{1}{H} \sum_{h=1}^H p_h \sum_{i \in \mathcal{I} \setminus \{v\}} s_{ivh} (\mathbf{x}_v - \mathbf{x}_i) \underbrace{\left[\sum_{j \in \mathcal{I} \setminus \{i\}} \frac{\tilde{\sigma}_{ij}}{s_{ij}} s_{ijh} \right]}_{g_{ih}}. \quad (4.28)$$

The denominator of (2.24),

$$\bar{s}_{ih} := \sum_{k \in \mathcal{I} \setminus \{i\}} \exp(-p_h d_{ik}^2/2), \quad (4.29)$$

can first be estimated for all $i \in \mathcal{I}$ and $h = 1, \dots, H$ in $\mathcal{O}(N \log^2 N)$ average time by extending the BH algorithm from Section 4.1. Then, f_{vi} and s_{vi} can be approximated for all $v \in \mathcal{I}$ and $i \in \tilde{\mathcal{I}}_v$ in $\mathcal{O}(N \log^2 N)$ time as $|\tilde{\mathcal{I}}_v|$ is $\mathcal{O}(\log N)$. Afterwards, estimating the right-hand-side of (4.25) for all $v \in \mathcal{I}$ takes $\mathcal{O}(N \log N)$ time, thanks to the sparsity of $\tilde{\sigma}_{vi}$ which reduces the sum over $\mathcal{I} \setminus \{v\}$ to a sum over $\tilde{\mathcal{I}}_v$. Similarly, (4.27) is approximated for all $v \in \mathcal{I}$ in $\mathcal{O}(N \log N)$ time, and g_{vh} for all $v \in \mathcal{I}$ and $h = 1, \dots, H$ in $\mathcal{O}(N \log^2 N)$ time. On the other hand, $\mathbf{g}_{vh}$ satisfies

$$\bar{s}_{vh} \mathbf{g}_{vh} = \sum_{i \in \mathcal{I} \setminus \{v\}} \exp(-p_h d_{vi}^2/2) (\mathbf{x}_v - \mathbf{x}_i), \quad (4.30)$$

where the right-hand-side is estimated in $\mathcal{O}(\log N)$ average time thanks to the BH algorithm. Thus, (4.26) is estimated for all $v \in \mathcal{I}$ in $\mathcal{O}(N \log^2 N)$ average time. Finally, (4.28) extends as

$$\sum_{i \in \mathcal{I} \setminus \{v\}} \left[\sum_{h=1}^H p_h \exp(-p_h d_{iv}^2/2) \frac{g_{ih}}{\bar{s}_{ih}} \right] (\mathbf{x}_v - \mathbf{x}_i). \quad (4.31)$$

To approximate (4.31), the quadtree is extended by storing, in each cell c , $\mathcal{O}(\log N)$ additional attributes consisting of

$$a_h^c := \frac{1}{|\mathcal{T}_c|} \sum_{i \in \mathcal{T}_c} \frac{g_{ih}}{s_{ih}}, \quad (4.32)$$

for $h = 1, \dots, H$, yielding a tree with $\mathcal{O}(N \log N)$ memory cost [YPK13]. Depth-first searching the resulting tree enables estimating the value of (4.31) in $\mathcal{O}(\log^2 N)$ average time, by stopping the traversal as soon as (4.7) holds and approximating the part of (4.31) with $i \in \mathcal{T}_c$ through a sum of $\mathcal{O}(\log N)$ terms,

$$|\mathcal{T}_c| \left[\sum_{h=1}^H p_h \exp(-p_h d_{v, \mathbf{m}_c}^2 / 2) a_h^c \right] (\mathbf{x}_v - \mathbf{m}_c). \quad (4.33)$$

Estimating (4.28) for all $v \in \mathcal{I}$ hence takes $\mathcal{O}(N \log^2 N)$ average time, which is the time complexity of approximating the gradient of (4.23). It is noteworthy that the estimation entails two tree traversals for each $v \in \mathcal{I}$, to first approximate (4.25), (4.26), (4.27) and (4.32), and then estimate (4.28) during the second search. One can thus expect an increased sensitivity of the multi-scale SNE acceleration with respect to the BH threshold θ , compared to fast single- and multi-scale t -SNE, since the estimation quality of the first traversal affects the second one.

The overall time complexity of the presented fast multi-scale schemes evolves as $\mathcal{O}(N \log^2 N)$, importantly without quadratic dependence with N . Note that in fast multi-scale t -SNE, although determining the HD similarities (4.20) takes $\mathcal{O}(N \log^2 N)$ time once at initialization, only $\mathcal{O}(N \log N)$ average time is required to approximate the gradient of (4.24), which is iteratively performed in gradient-based optimization.

It is noteworthy that $j \in \tilde{\mathcal{I}}_i$ does not imply $i \in \tilde{\mathcal{I}}_j$. One may easily extend the multi-scale neighbor sets to ensure this property, but experiments revealed that it does not influence the performances much: only tiny improvements in terms of DR quality were observed, at the expense of slightly increased computation times. As multi-scale t -SNE is defined through symmetric similarities, its fast version extends the sets $\tilde{\mathcal{I}}_i$ to ensure the above property, but not fast multi-scale SNE.

The proposed accelerations are based on the definition of multi-scale neighbor sets $\tilde{\mathcal{I}}_i \subset \mathcal{I} \setminus \{i\}$ for all $i \in \mathcal{I}$, with a limited number of elements emulating the intrinsic hierarchy of the data scales. Indeed, $\tilde{\mathcal{I}}_i$ gathers the closest neighbors of ξ_i as well as farther away HD samples, more diffused in the data cloud. This enables accurately describing the local, quickly fluctuating properties around ξ_i while providing a succinct, higher-order summary of the global HD characteristics. The induced sparse multi-scale HD similarities (4.20) hence model data proximities at different ranges,

unlike their single-scale counterparts (4.1). An additional $\mathcal{O}(\log N)$ factor arises in the time complexity with respect to fast single-scale approaches, with minor impact compared to the linear dependency in N . Matching the HD similarities (4.20) with BH estimates of the multi-scale LD similarities enables efficiently mapping both local and global HD structures in the LDS, while avoiding LD crushing. Similarly to multi-scale SNE and t -SNE, multi-scale NeRV and JSE can be accelerated thanks to the multi-scale neighbor sets along with BH approximations.

4.4 Experimental results

The performances of the presented fast multi-scale methods are studied in the context of two main experiments, in terms of both DR quality and computation time. First, Section 4.4.1 considers several moderate-sized databases, with a few thousands of samples each, which enables applying the original, non-accelerated versions of the algorithms for comparison purposes. The assessed methods are hence multi-scale SNE (Ms SNE), multi-scale t -SNE (Ms t -SNE) and their proposed accelerations (FMs SNE and FMs t -SNE). For the sake of comparison, the results of the single-scale t -SNE and of its Barnes-Hut version (BH t -SNE), detailed in Sections 2.1 and 4.1, are also discussed. The behavior of the fast methods with respect to the BH threshold θ is analyzed to suggest adequate values suitable across multiple databases. Second, Section 4.4.2 deals with large-scale data sets, with tens or hundreds of thousands samples each. The quadratic time complexity of the non-accelerated schemes prevents their application on these databases. Therefore, only scalability of the fast schemes is tested in this setting.

In both Section 4.4.1 and 4.4.2, HD and LD distances are Euclidean and two-dimensional embeddings are targeted ($P = 2$), as typically desired in exploratory visualization. On each database, both fast multi-scale algorithms are applied 30 times, each with different random sub-samplings of the HD data set. Average results are then reported over these replications, along with standard deviations. Moreover, thanks to the repetitions, statistical tests allow evaluating the significance of the difference in average performances of the methods. Paired tests are computed when FMs SNE and FMs t -SNE are confronted to each other, and one-sample tests for the mean otherwise. For each studied score, if their 30 values for each fast multi-scale method are normally distributed according to D’Agostino and Pearson’s omnibus normality test [d’A71], two-tailed t -tests are employed, and Wilcoxon tests otherwise. In this chapter, the test significance level is fixed to $\alpha = 0.05$. As multiple hypothesis tests result from each database analysis, the Holm–Bonferroni correction is used [Sha95]. Non-significant differences are hence considered as statistically significant with probability

bounded by α .

The compared single- and multi-scale algorithms are optimized as advised in their respective papers [vdMH08, vdM14, LPOV15], the proposed fast multi-scale methods being minimized as their non-fast versions. In particular for the latter, K_1 and H are respectively set to 2 and $\lfloor \log_2(N/K_1) \rfloor$, multi-scale optimization [LPOV15] is executed and the L-BFGS algorithm is conveniently employed [BLNZ95]. As to single-scale t -SNE and BH t -SNE, momentum gradient descent is used along with early exaggeration, with parameters as described in [vdMH08, vdM14]. The perplexity K_* is set to 50, as in [vdM14]. In both single- and multi-scale settings, the iterations of the optimization schemes are stopped as soon as (1) the gradient infinite norm is below 10^{-5} , or (2) the cost function relative updates are bounded by 10^{-8} , or (3) a maximum number is reached. The latter maximum number must, however, be carefully defined as different data sets require, obviously, different numbers of iterations before convergence of the LD embedding. Comparing converged and non-converged results of the methods across databases may likely lead to misleading conclusions on the behavior of the algorithms, in terms of both achievable DR quality and computation time. On the other hand, large numbers of samples often require unaffordable amounts of time before reaching convergence of the optimization. For these reasons, the number of iterations of optimization algorithms is limited to 10^5 in Section 4.4.1. As much fewer iterations are typically required for convergence, this threshold is almost equivalent to an unbounded number of iterations. Thus, in Section 4.4.1, the optimization schemes terminate according to the first two above conditions, on the gradient norm and relative cost function updates. This ensures comparability of the performances across data sets while maintaining affordable computation times, since moderate-sized databases are tackled. As to Section 4.4.2, since huge numbers of samples are considered, the maximum number of iterations is reduced to avoid dealing with too long computation times, as one would proceed in a practical context when facing large-scale databases. For BH t -SNE, 10^3 iterations are performed, as suggested in [vdM14]. For FMs SNE and FMs t -SNE, as few as 30 iterations per L-BFGS run are at most executed. All simulations and computation times were obtained thanks to a desktop computer with a 4-core Intel(R) Xeon(R) CPU E3-1245 v5 at 3.5 GHz, with 32 GB RAM.

4.4.1 Comparison between fast and non-fast methods

The data sets studied in this section originate from the UCI Machine Learning Repository [Lic13]:

- Anuran calls MFCCs (Anuran);
- Combined cycle power plant (Plant) [Tüf14, KTG12];

- Musk version 2 (Musk);
- Spambase;
- Gesture phase segmentation (Gesture) [MLP13];
- Statlog landsat satellite (Satellite);
- First-order theorem proving (Theorem) [BHP14];
- The Waveform database generator version 2 (Waveform);
- Isolet;
- Madelon [GGBHD04];
- Abalone;
- Wine quality (Wine) [CCA⁺09].

Their numbers N and H of samples and features are provided in Table 4.1.

Similarly to Sections 3.3.1 and 4.2, the experiments in this section also rely on the R_{NX} and AUC measures, to characterize DR quality by accounting for all scales, with a focus on smaller ones for the AUC [LPOV15].

Figures 4.15, 4.16, 4.17, 4.18 and 4.19 respectively present the results as a function of the BH threshold θ on Anuran, Plant, Musk, Spambase and Gesture.

Non-fast methods. On all databases, Ms SNE is consistently the best method in terms of AUC, at the expense of long computation times. Indeed, multi-scale HD and LD similarities enable very accurate tuning of the LDS, with high resolution as they involve both large and small precisions. Nevertheless, the induced LD Gaussian neighborhoods likely rapidly locally fluctuate, making the gradient of Ms SNE sensitive to small LD variations. Its bad scaling hence causes a slow convergence of the optimization process. Next, the AUC of Ms t -SNE is always second to Ms SNE, except on some databases when small BH thresholds are employed with FMs SNE and on Spambase, on which FMs t -SNE performs better. In any case, Ms SNE and Ms t -SNE systematically outperform single-scale t -SNE and BH t -SNE in terms of DR quality. From the running time perspective, Ms t -SNE is much more affordable than Ms SNE. The heavy-tailed Student t -distributed LD similarities likely induce a more stable gradient regarding LD variations. Ms t -SNE is even faster than single-scale t -SNE. Although the comparison is less straightforward in this last case, since Ms t -SNE and single-scale t -SNE rely on different optimization schemes, one may argue that accounting for global properties in Ms t -SNE further constrains the LDS structure, facilitating the cost function minimization. On the other hand, as t -SNE

focuses on local aspects, the global LDS organization is left undetermined. Shaping it hence probably requires additional time to fix these degrees of freedom.

Fast methods. The AUC of BH t -SNE is stable with respect to the BH threshold θ , in agreement with the results of Section 4.2 [dBML18a]. The same remarkable behavior is preserved in FMs t -SNE, which is consistently and statistically significantly above BH t -SNE and t -SNE. FMs SNE is however much more sensitive to the BH threshold: its average AUC quickly drops with θ , with increasing standard deviation as the threshold increases. This is due to the two successive tree traversals involved in the FMs SNE gradient approximation: as the second traversal employs the results of the first one, augmenting θ not only affects the estimation quality of the first pass but also more severely deteriorates the accuracy of the second one, in a cascading effect. The DR quality of FMs SNE thus rapidly decreases with θ , while small BH thresholds yield high AUC since, in this case, both tree traversals deliver precise approximations. Smaller θ nevertheless induces larger running times for both FMs SNE and FMs t -SNE. Considering the AUC and computation time dependence of FMs SNE with respect to θ , its BH threshold should be carefully determined to ensure a proper balance between DR quality and running time. For this reason, a threshold $\theta = 0.45$ is considered for FMs SNE in the remaining experiments. On the other hand, FMs t -SNE can accommodate with larger BH thresholds thanks to its AUC dependence with θ , enabling to decrease computation time without suffering from significant loss in DR quality. As it can reach high DR quality with low computation time much more easily and as it is intrinsically faster since entailing only one tree traversal per gradient estimation instead of two, one can state that FMs t -SNE is superior to FMs SNE. For the subsequent experiments, $\theta = 0.75$ is used with FMs t -SNE. Noticeably, although both FMs SNE and FMs t -SNE get, as expected, faster as θ increases, BH t -SNE behaves differently. Indeed, on all data sets, the running time of BH t -SNE first decreases with θ , reaches a minimum at around $\theta = 0.2$ and then increases before remaining roughly stable. This intriguing phenomenon may be explained as larger θ accelerates each gradient estimation but also induces coarser approximations. These likely result in more iterations before reaching convergence of the momentum gradient descent. The L-BFGS scheme employed in FMs methods less suffers from these rougher estimations, probably thanks to the successive Hessian approximations it computes, which partly cancel the influence of noisy gradient estimates. Therefore, for most BH thresholds, FMs t -SNE is statistically significantly faster than BH t -SNE. For completeness of the comparison, three thresholds are further employed with BH t -SNE, namely 0.2 as it minimizes running times on Anuran, Musk, Spambase, Gesture and almost Plant, 0.5 as in

[vdM14], and 0.75 as larger value.

Figures 4.20a and 4.20b depict the AUC and running time of the methods as a function of the number of points in sub-samples of Gesture. In terms of AUC, Ms SNE remains the best method, followed by Ms t -SNE and then FMs t -SNE. The last three methods are close, smaller data sets favoring BH t -SNE and FMs SNE being superior on larger ones. Smaller BH thresholds would improve the accuracy of FMs SNE on smaller sub-samples. As to the computation times, FMs t -SNE is consistently and statistically significantly the fastest method, which raises the interest for it in light of its AUC performances. FMs SNE is second and BH t -SNE is indeed faster with $\theta = 0.2$. Using log-log axes as in Fig. 4.20b, the slope of linear regressions of the method timings should reveal their time complexity. One would expect a slope near 2 for t -SNE, above 2 for Ms SNE and Ms t -SNE, above 1 for BH t -SNE and higher than 1 for FMs SNE and FMs t -SNE, as these methods respectively have $\mathcal{O}(N^2)$, $\mathcal{O}(N^2 \log N)$, $\mathcal{O}(N \log N)$ and $\mathcal{O}(N \log^2 N)$ theoretical time complexity. Such slopes are indeed experimentally observed (larger numbers of samples would obviously yield more accurate values at the expense of much larger running times for non-fast methods).

Table 4.1 provides performances of the methods on all databases using specific BH thresholds. Ms SNE keeps the best AUC and slowest computation times. With more affordable running times, Ms t -SNE is always second in AUC except on Spambase on which FMs t -SNE is superior. Ms t -SNE is also faster than t -SNE, except on Spambase and Madelon. FMs t -SNE is consistently and statistically significantly the fastest method and yields the highest AUC among accelerated schemes, except on Plant and Satellite, on which FMs SNE is superior, without statistical significance on Plant, and on Isolet, on which BH t -SNE performs well, without being statistically different from FMs SNE in AUC. FMs t -SNE also statistically significantly beats t -SNE in terms of AUC, except on Waveform and Madelon. Computation time of BH t -SNE is indeed minimized with $\theta = 0.2$, except on Waveform on which $\theta = 0.5$ is slightly faster. FMs SNE is not as fast as FMs t -SNE but still generally results in statistically significantly larger AUC than t -SNE and BH t -SNE, in less or similar time as BH t -SNE and less time than t -SNE and Ms t -SNE. The AUC difference between Ms SNE and FMs SNE is always greater than between Ms t -SNE and FMs t -SNE, probably due to the two tree traversals required for the gradient estimation.

Figure 4.21 illustrates the results of Table 4.1. The high concentration of FMs t -SNE markers in the upper left part of the figure highlights its superiority with respect to the other DR schemes. The performances of FMs SNE and FMs t -SNE being reported for 30 different random sub-samplings, one can visually observe the variability of their performances. While FMs SNE is stable in terms of computation time but slightly less

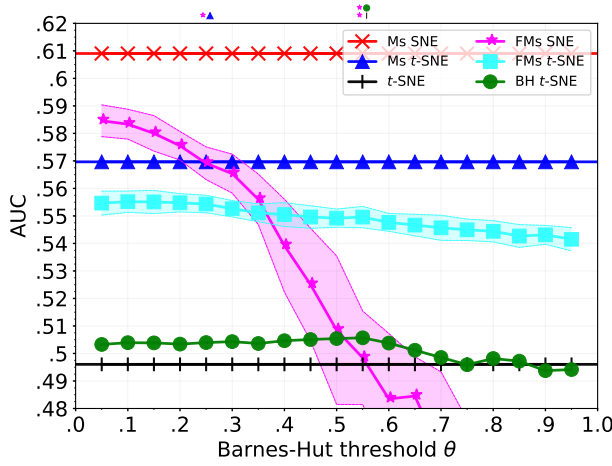
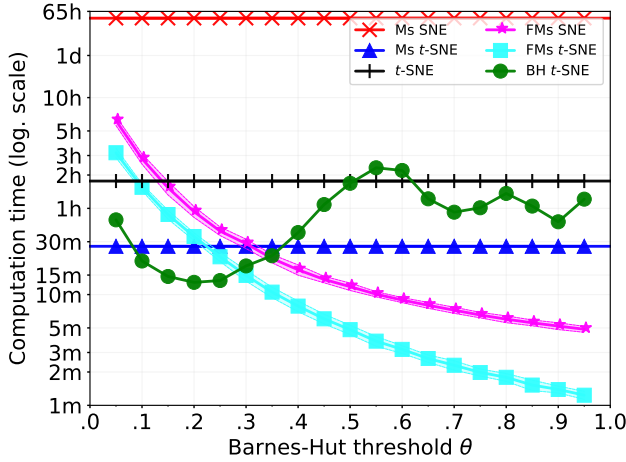
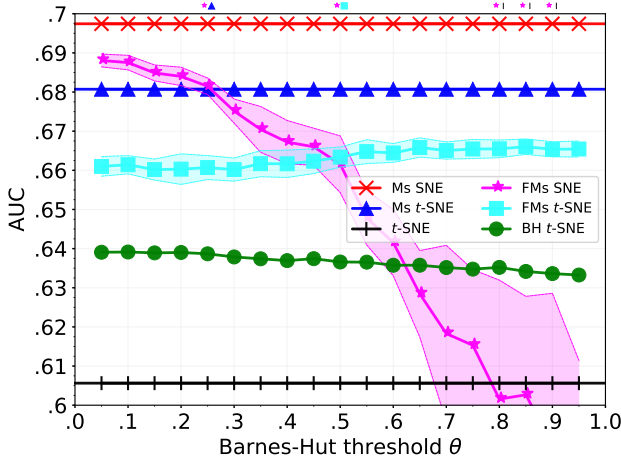
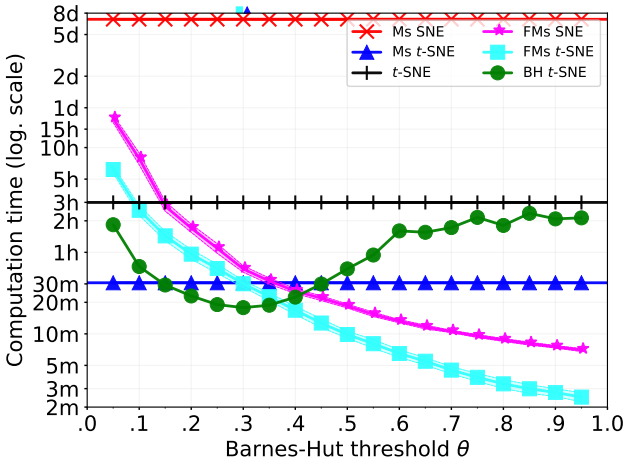
(a) Anuran. AUC (θ).(b) Anuran. Time (θ).

Figure 4.15: (a) AUC of the DR methods on Anuran as a function of the BH threshold θ , from 0.05 to 0.95 with 0.05 step. Average performances over 30 random sub-samplings are depicted for FMs SNE and FMs t -SNE, with shaded areas delimiting one standard deviation around the mean. Above the figure, pairs of markers indicate schemes that are **not** statistically significantly different, for instance Ms t -SNE and FMs SNE for $\theta = 0.25$. (b) Analogous to (a) but for the computation time. Log-scale is used for the y -axis and s, m, h and d refer to second, minute, hour and day.



(a) Plant. AUC (θ).



(b) Plant. Time (θ).

Figure 4.16: (a) Similar to Fig. 4.15a but on Plant. (b) Analogous to (a) but for the computation time. Log-scale is used for the y -axis and s, m, h and d refer to second, minute, hour and day.

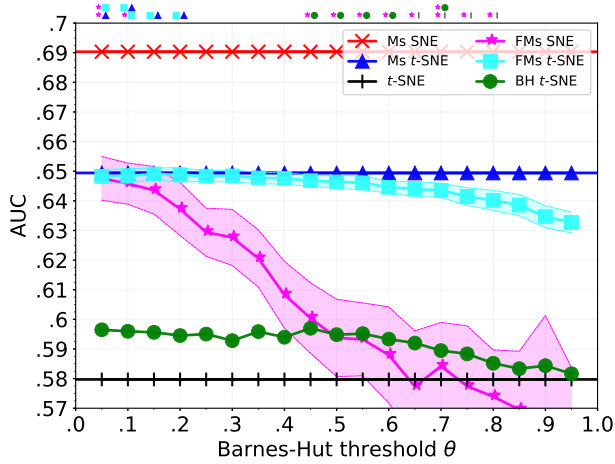
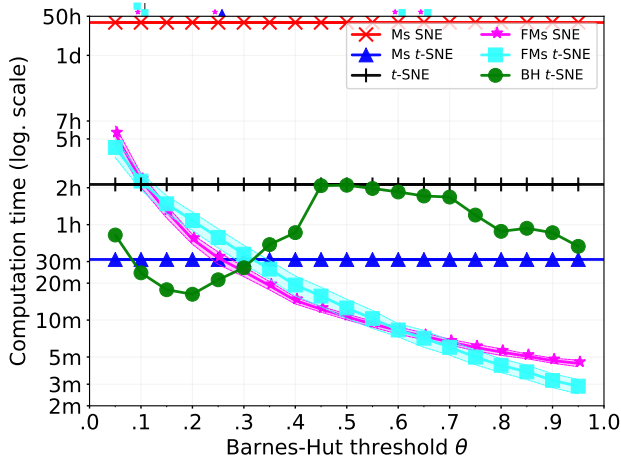
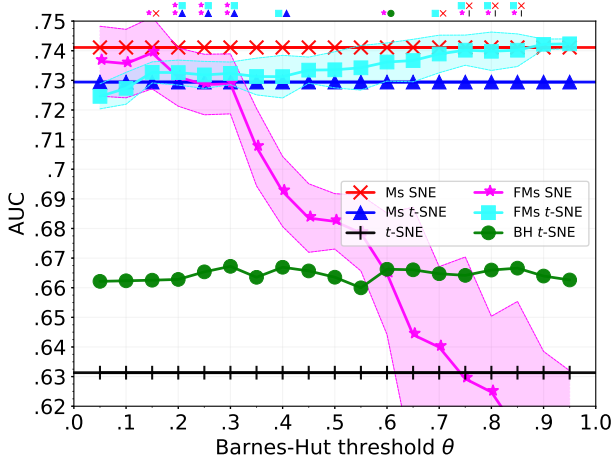
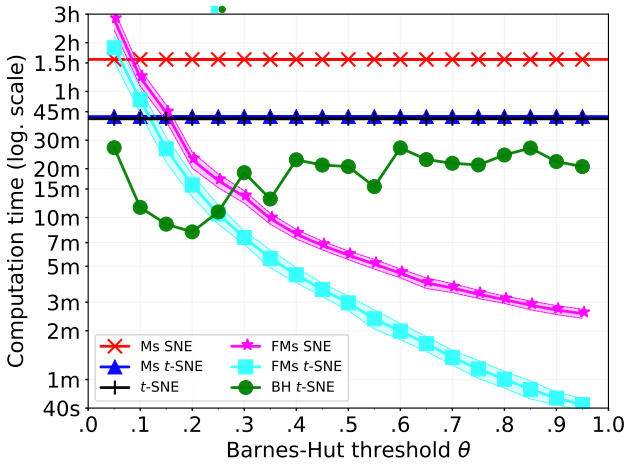
(a) Musk. $AUC(\theta)$.(b) Musk. Time(θ).

Figure 4.17: (a) Similar to Fig. 4.15a but on Musk. (b) Analogous to (a) but for the computation time. Log-scale is used for the y -axis and s, m, h and d refer to second, minute, hour and day.



(a) Spambase. $AUC(\theta)$.



(b) Spambase. $Time(\theta)$.

Figure 4.18: (a) Similar to Fig. 4.15a but on Spambase. (b) Analogous to (a) but for the computation time. Log-scale is used for the y -axis and s, m, h and d refer to second, minute, hour and day.

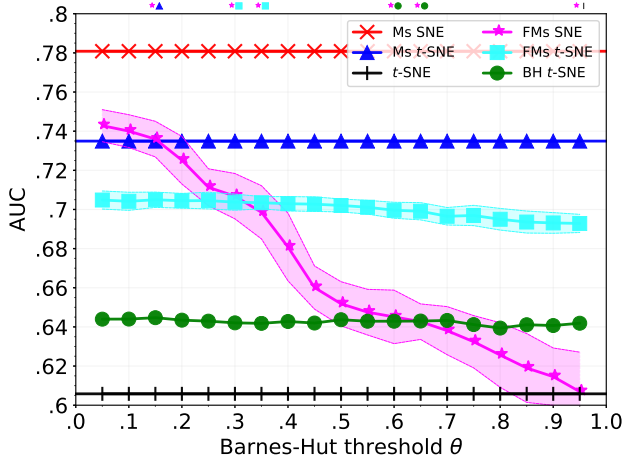
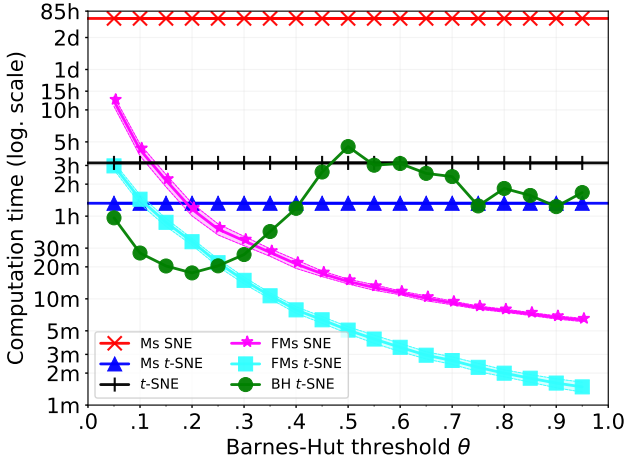
(a) Gesture. $\text{AUC}(\theta)$.(b) Gesture. Time(θ).

Figure 4.19: (a) Similar to Fig. 4.15a but on Gesture. (b) Analogous to (a) but for the computation time. Log-scale is used for the y -axis and s, m, h and d refer to second, minute, hour and day.

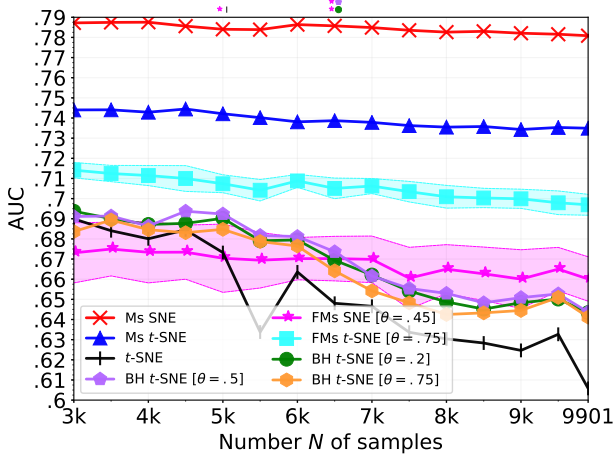
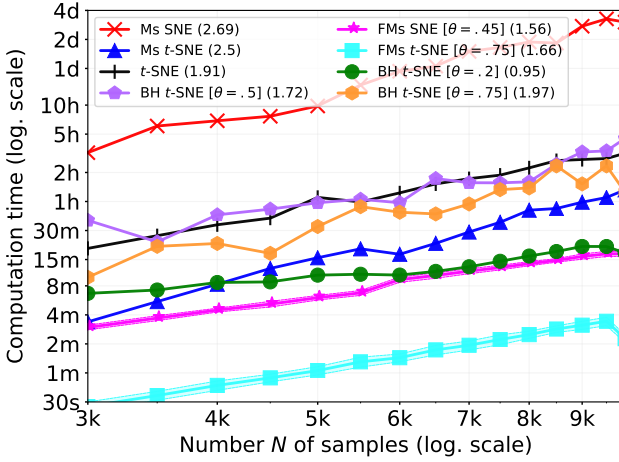
(a) Gesture. $AUC(N)$.(b) Gesture. Time (N).

Figure 4.20: (a) Similar to Fig 4.19a but for sub-samples of the Gesture data set, with fixed BH thresholds θ described in the legend. The number N of samples varies with steps of 500 on the x -axis. (b) Analogous to (a) but for the computation time, on the y -axis. Both axes are in log-scale and s, m, h and d refer to second, minute, hour and day. For each DR algorithm, the slope of a linear regression of its results in log-log scale is provided in parentheses in the legend.

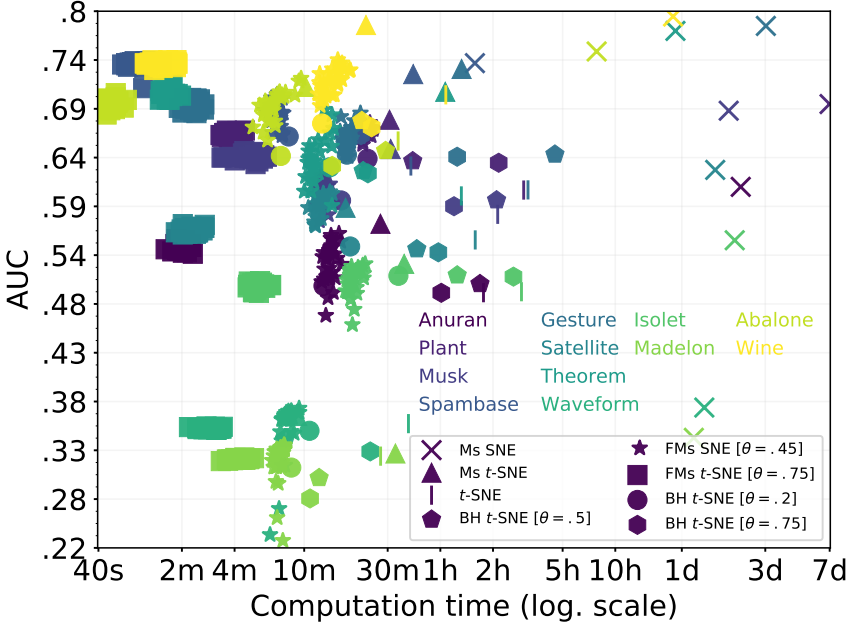
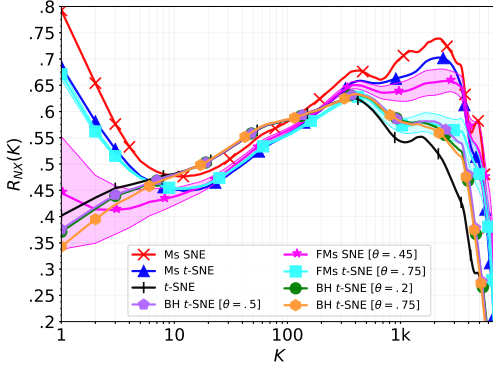


Figure 4.21: AUC of the DR methods on moderate-sized data sets as a function of their computation time in log-scale, with s, m, h and d referring to second, minute, hour and day. Results of FMs SNE and FMs t -SNE are depicted for 30 different random sub-samplings. The markers and colors respectively differentiate the DR methods and data sets. The data sets names are indicated using their associated colors to enable their identification.

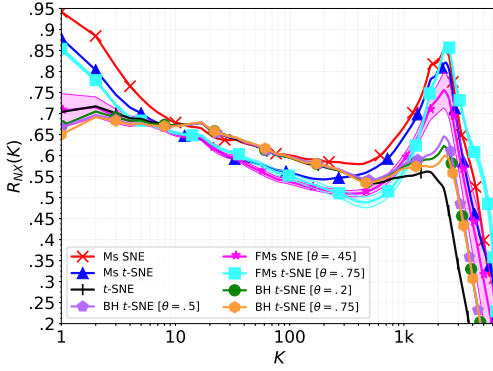
in AUC for reasons detailed below, FMs t -SNE is remarkably consistent across the random sub-samplings. All conclusions drawn from Table 4.1 are depicted in Fig. 4.21.

Figure 4.22 depicts $R_{NX}(K)$ curves for Anuran, Musk and Satellite. Both original and accelerated multi-scale methods outperform single-scale algorithms in terms of local and global neighborhood preservation. Single-scale schemes are only close or slightly above for neighborhood sizes near the perplexity. Indeed, as mentioned in [LPOV15], at the expense of a small loss around the perplexity, the multi-scale behavior uplifts the curve for all other neighborhood sizes. FMs SNE is close to Ms SNE in terms of global neighborhood preservation but does not reproduces local ones as well. FMs t -SNE behaves in the opposite way when compared to Ms t -SNE.

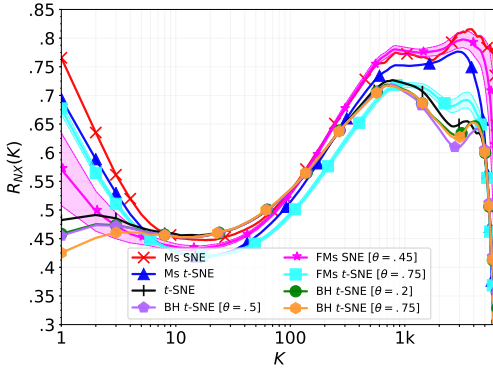
It is noteworthy that in Fig. 4.15, 4.16, 4.17, 4.18, 4.19, 4.20, Table 4.1,



(a) Anuran.



(b) Musk.



(c) Satellite.

Figure 4.22: (a) $R_{NX}(K)$ curves of the DR methods on Anuran as a function of the neighborhood size K , using the same BH thresholds as in Table 4.1. Average curves over 30 random sub-samplings are depicted for FMs SNE and FMs t -SNE, with shaded areas delimiting one standard deviation around the mean, as in Fig. 4.15, 4.16, 4.17, 4.18, 4.19 and 4.20. A log-scale is employed for the neighborhood size K . (b)-(c) Similar but respectively on Musk and Satellite.

Fig. 4.21 and Fig. 4.22, the standard deviations of FMs SNE are systematically larger than for FMs t -SNE. This indicates that FMs SNE is more sensitive to the random sub-samplings than FMs t -SNE, yielding magnified variations of the small neighborhood preservation in Fig. 4.22. This increased sensitivity likely results from the two tree traversals of the FMs SNE gradient estimation. Indeed, the BH algorithm is directly dependent on the LD configuration, which is itself related to the considered HD data sub-sampling. The larger influence of the BH algorithm on FMs SNE than on FMs t -SNE, caused by the two tree passes performed per data point at each iteration, then induces an increased sensitivity of FMs SNE with respect to the HD data sub-samplings.

4.4.2 Scalability analysis of the fast methods

Non-accelerated NE schemes can manage up to a few thousands samples, due to their quadratic time complexity. Fast methods better scale on large data sets, with tens or hundreds of thousands samples. This section hence studies their performances in this last context, using data sets from the UCI Machine Learning Repository [Lic13]:

- Statlog Shuttle (Shuttle);
- Sensorless Drive Diagnosis (Sensorless);
- SGEMM GPU kernel performance (Sgemm) [NC15];
- Covertypes;
- Gas sensors for home activity monitoring (Gas) [HMF⁺16];

Table 4.2 provides their numbers N and H of samples and features.

To assess DR quality, the AUC is no longer suitable as its evaluation requires $\mathcal{O}(N^2 \log N)$ time. Nevertheless, the start of the $R_{\text{NX}}(K)$ curve, with K ranging from 1 to $\bar{K}$, can be computed in $\mathcal{O}(\bar{K}N \log N)$ time, thanks to vantage-point trees. The area $\overline{\text{AUC}}$ of this reduced curve develops as (3.50), but with the upper limits of the sums replaced with $\bar{K}$. The smaller $\bar{K}$ compared to N , the more the $\overline{\text{AUC}}$ focuses on local neighborhoods compared to AUC. In this section, $\bar{K}$ is set to 10^4 , to consider similar neighborhood sizes as in Section 4.4.1.

Table 4.2 gathers the results. On Shuttle and Sensorless, both with tens of thousands samples, FMs SNE performs best in terms of $\overline{\text{AUC}}$, but with the largest computation times. This is due to the $\mathcal{O}(N \log^2 N)$ time complexity of each gradient estimation, moreover involving two tree traversals per data point, whereas the optimization iterations in BH t -SNE and FMs t -SNE only cost $\mathcal{O}(N \log N)$ time, with a single tree pass per sample.

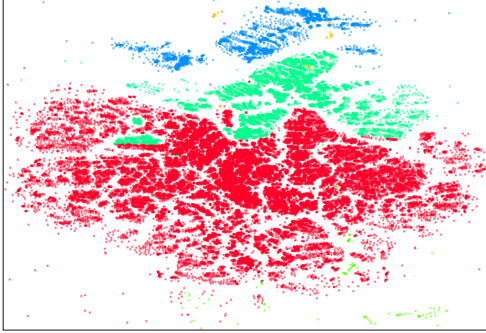
FMs t -SNE is statistically significantly the fastest scheme, also with statistically significantly greater $\overline{\text{AUC}}$ than BH t -SNE. It is noteworthy that increasing θ decreases the running time of BH t -SNE, differently to what is observed in Section 4.4.1. Indeed, computing 10^3 momentum gradient descent steps does not guarantee LDS convergence. Therefore, larger BH thresholds may accelerate BH t -SNE optimization when stopped after 10^3 iterations, as increasing θ yields cheaper BH estimates and faster iteration evaluations.

Figure 4.23 depicts 2-D embeddings of Shuttle as computed by FMs SNE, FMs t -SNE and BH t -SNE. This database contains unbalanced labels, 80% of its samples belonging to a single class. Visually, both fast multi-scale methods tend to more regroup the classes than BH t -SNE. Figure 4.23c moreover presents cluster over-emphasis produced by BH t -SNE, which affects its $\overline{\text{AUC}}$ with respect to FMs SNE and FMs t -SNE. Indeed, although matching Gaussian similarities with heavy-tailed Student t -distributed ones is motivated by the crowding problem to compensate for the relative loss of volume from the HDS to the LDS [vdMH08], it typically introduces artificial clusters in the embedding since mid-range to distant neighbors are further repelled [dBMVL18b]. In Fig. 4.23b, FMs t -SNE is also subject to this phenomenon but to a lesser extent, as multiple Gaussian functions with increasing tails are averaged in the HD space, reducing the gap with the heavy-tailed Student t distribution. In contrast, as suggested by Fig. 4.23a and its dominant $\overline{\text{AUC}}$, FMs SNE is more robust to cluster over-emphasis, thanks to the multi-scale Gaussian similarities it employs in both spaces.

As to Sgemm, Coverttype and Gas, the results of FMs SNE with $\theta = 0.45$ are not reported in Table 4.2 as these databases contain hundreds of thousands samples, inducing intractable computation times when applying the method with 30 different random sub-samplings. Larger thresholds could have been employed, but they yield unstable DR quality as suggested in Fig. 4.15, 4.16, 4.17, 4.18 and 4.19. Although BH t -SNE is the fastest scheme, FMs t -SNE impressively outperforms it in terms of $\overline{\text{AUC}}$. The poor DR quality achieved by BH t -SNE, with the same $\overline{\text{AUC}}$ for all three thresholds, suggests that it is far from convergence, before θ starts influencing the LD positioning. On the other hand, FMs t -SNE delivers fair $\overline{\text{AUC}}$, for instance in 4.5 hours on average on Gas, containing almost one million samples. The 2-D embeddings of this data set illustrate in Fig. 4.24 these results: the LDS computed by FMs t -SNE in Fig. 4.24a indeed reveals a structure related to the three classes of Gas. Such an organization is not obtained by BH t -SNE in Fig. 4.24b and 4.24c, which hardly and fuzzily arrange the data points regardless of the BH threshold θ , leading to a weak $\overline{\text{AUC}}$.

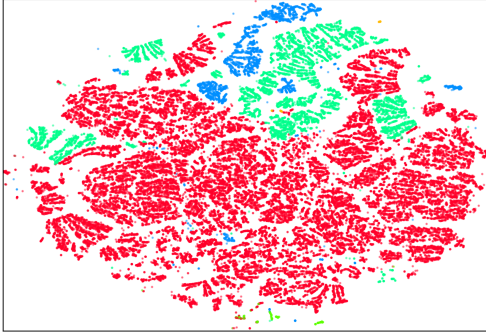
Figure 4.25 further depicts the results of Table 4.2. For Shuttle and

$\overline{\text{AUC}} = 57.58\%$, Time = 36309 s.



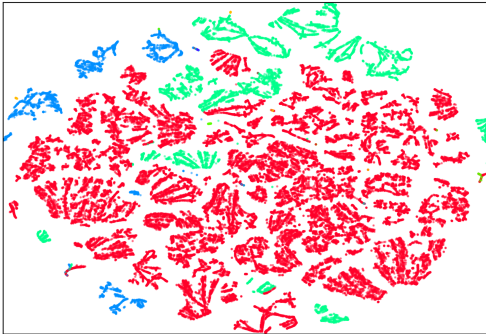
(a) FMs SNE ($\theta = 0.45$).

$\overline{\text{AUC}} = 53.879\%$, Time = 1050 s.



(b) FMs t -SNE ($\theta = 0.75$).

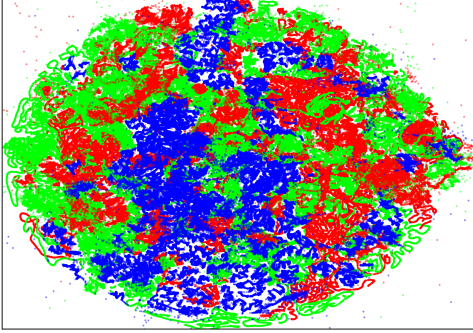
$\overline{\text{AUC}} = 50.448\%$, Time = 2234 s.



(c) BH t -SNE ($\theta = 0.5$).

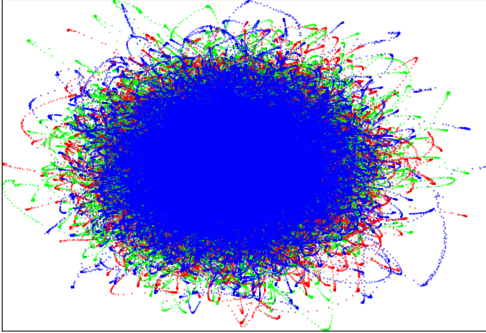
Figure 4.23: (a) 2-D embedding of Shuttle obtained by one of the 30 applications of FMs SNE. The classes in the data set are depicted using different colors. The computation time provided in the title of the figure is in seconds. (b)-(c) Similar but for FMs t -SNE and BH t -SNE. The embedding of BH t -SNE is displayed with $\theta = 0.5$ as it maximizes its $\overline{\text{AUC}}$ on Shuttle among $\{0.2, 0.5, 0.75\}$ in Table 4.2.

$\overline{\text{AUC}} = 45.172\%$, Time = 16504 s.



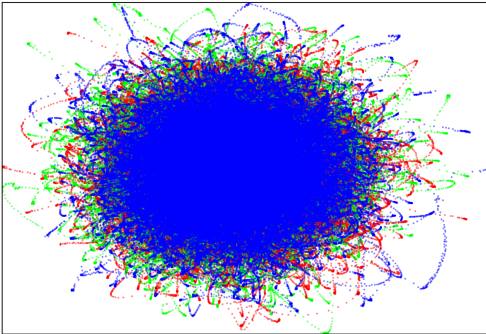
(a) FMs t -SNE ($\theta = 0.75$).

$\overline{\text{AUC}} = 5.283\%$, Time = 16835 s.



(b) BH t -SNE ($\theta = 0.2$).

$\overline{\text{AUC}} = 5.283\%$, Time = 3171 s.



(c) BH t -SNE ($\theta = 0.75$).

Figure 4.24: (a) 2-D embedding of Gas obtained by one of the 30 applications of FMs t -SNE. The classes in the data set are depicted using different colors. The computation time provided in the title of the figure is in seconds. (b)-(c) Similar but for BH t -SNE, respectively with $\theta = 0.2$ and $\theta = 0.75$.

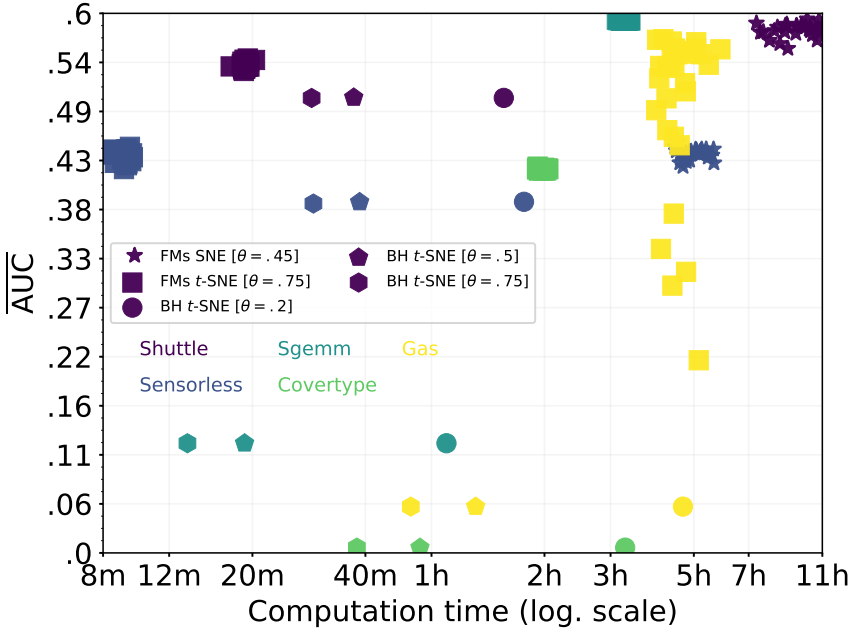


Figure 4.25: Similar to Fig. 4.21 but for the fast DR methods on large-scale data sets, with their $\overline{\text{AUC}}$ along the y -axis.

Sensorless, FMs t -SNE indeed lies in the upper left corner of the figure, while $\text{BH } t$ -SNE is the fastest on Sgemm, Covertypes and Gas, but with poor $\overline{\text{AUC}}$ compared to FMs t -SNE. It is noteworthy that FMs t -SNE has a less stable $\overline{\text{AUC}}$ on Gas across the random sub-samplings compared to all other data sets, due to the challenging nature and large size of Gas.

4.5 Discussion

In DR, NE methods produce LD versions of HD data sets retaining HD neighborhoods impressively well. HD structures extraction drives a lot of attention, leading to efficient frameworks easing subsequent classification [ZNL17, ZNL18, ZNL16] and clustering [ZNG⁺18]. In visualization, multi-scale schemes outperform single-scale ones by accounting for both local and global data interactions, improving the HD neighborhood preservation. Yet, their $\mathcal{O}(N^2 \log N)$ time complexity with N samples limits their applicability to a few thousands data. To tackle larger databases, this chapter develops $\mathcal{O}(N \log^2 N)$ accelerations of the multi-scale NE methods. While fast

single-scale methods harness the intrinsic sparsity of their HD similarities, the proposed fast multi-scale schemes account for the dense nature of the multi-scale HD similarities by defining neighbor sets for each datum incorporating samples summarizing the data cloud from different perspectives. These sets gather the closest HD neighbors to each datum to accurately capture the locally fluctuating structures, as well as HD points sampled with different resolutions to consider the global data cloud properties. With a BH algorithm avoiding LD crushing, the accounted subsets of HD data interactions enable relevantly scoring the LD coordinates quality. The LDS is hence efficiently shaped to reproduce HD relationships at all scales. Although the presented algorithms evolve in $\mathcal{O}(N \log^2 N)$, it is noteworthy that the gradient of FMs t -SNE is iteratively estimated in $\mathcal{O}(N \log N)$ average time. As it requires a single tree traversal per datum for each gradient approximation, it is superior to FMs SNE, as confirmed experimentally. Simulations highlight that fast multi-scale methods consistently and statistically significantly outperform state-of-the-art single-scale schemes in DR quality, while tremendously accelerating their non-fast versions, enabling multi-scale visualization of large data sets.

In extremely HD spaces, the efficiency of the nearest neighbor searches through vantage-point trees may decay, due to the curse of dimensionality [LV07]. In such cases, one can however first reduce the dimension with PCA towards an intermediate dimensionality while preserving most information content [vdM14]. Another option is to employ random projection trees combined with neighbor exploring techniques [TLZM16]. As to future lines of work, our methods can be extended to class-aware settings [ZNL16], inspired by recent successful works [LZN⁺19]. Weights could also be associated to the multi-scale sampled neighbors, proportionally to the number of neglected interactions they represent on average. These interactions would also influence the LDS computation, to improve DR quality. The design of such weights should however consider both the HD and LD coordinates, which are iteratively updated. Another perspective is to study fast multipole methods [VCP14] and negative sampling techniques [TLZM16], to analyze whether they yield better trade-offs between running time and DR quality. In this line, variants of condition (4.7) will be investigated as well, by replacing r_c with the average distance between $\mathbf{m}_c$ and the points in cell c , and the diagonal of the smallest box which contains all points in $\mathcal{T}_c$. These refinements would ease meeting the BH approximation condition, since (4.7) could hold as soon as all points in $\mathcal{T}_c$ are close one to another and far from $\mathbf{x}_v$, even if cell c is large, hence possibly further reducing computation time.

Table 4.2: Performances of the fast DR methods on large-scale data sets, in terms of $\overline{\text{AUC}}$ and of computation time in seconds. For FMs SNE and FMs t -SNE, the provided results are averages over 30 random sub-samplings, with standard deviations in parentheses. In each row, the greatest $\overline{\text{AUC}}$ or smallest time is indicated in bold. Results are in italic (resp. underlined) when **not** statistically significantly different from FMs SNE (resp. FMs t -SNE).

		FMs SNE $\theta = 0.45$	FMs t -SNE $\theta = 0.75$	BH t -SNE	
			$\theta = 0.2$	$\theta = 0.5$	$\theta = 0.75$
Shuttle $N = 58000$ $H = 9$	$\overline{\text{AUC}}$ [%]	57.684 (0.716)	54.009 (0.35)	50.404	50.448
	Time [s]	33448 (3732)	1146 (26)	5610	2234
Sensorless $N = 58509$ $H = 48$	$\overline{\text{AUC}}$ [%]	43.996 (0.576)	<i>43.945 (0.593)</i>	38.934	38.91
	Time [s]	17790 (1257)	550 (21)	6350	2316
Sgmm $N = 241600$ $H = 18$	$\overline{\text{AUC}}$ [%]	—	59.022 (0.057)	12.277	12.277
	Time [s]	—	11812 (254)	3946	1144
Coverttype $N = 581012$ $H = 54$	$\overline{\text{AUC}}$ [%]	—	42.556 (0.104)	0.756	0.756
	Time [s]	—	7083 (151)	11813	3354
Gas $N = 928991$ $H = 11$	$\overline{\text{AUC}}$ [%]	—	49.559 (9.218)	5.283	5.283
	Time [s]	—	16449 (1664)	<u>16835</u>	4719
					3171

Chapter 5

Conclusions

THE goal of this thesis is to develop robust and fast nonlinear dimensionality reduction (DR) algorithms for visualization to ease extended and efficient high-dimensional (HD) data exploration in data analysis tasks. Such exploratory phases are crucial to help defining the application needs, to drive the subsequent design choices and data analyses to conduct, and to appropriately assess the relevance of the performances of the employed machine learning (ML) methods [Pla13, CCR⁺19]. In this context, whereas most ML schemes only provide the user with quantitative indicators such as cost function values, which are difficult to interpret, data visualization offers powerful exploration means [Mun14]. Since numerous ML tasks nowadays deal with HD data, the sake of improving exploratory visualization requires considering the recent progresses made in the development of visualization tools for HD data, in the field of ML-based nonlinear DR [LV07].

Nonlinear DR seeks at faithfully representing HD data sets into low-dimensional (LD) spaces, mostly for exploratory visualization or to dismiss the curse of dimensionality [CP97, BPN16, BBH12]. The relevance of an LD embedding typically relies on the HD neighborhood preservation. Mappings from HD to LD coordinates formalize this neighborhood preservation principle in the context of several paradigms [LV07], such as the reproduction of distances [BG05] or neighborhoods [Koh90, HR02].

Linear projections of the HD vectors include early principal component analysis (PCA) [Jol86] and classical metric multidimensional scaling (MDS) [BG05], driven by the variance or dot product preservation. Nonlinear metric MDS extensions [Sam69] define (weighted) distance preservation schemes relying on either Euclidean or approximated geodesic measures [Ten98, TDSL00, LLV04]. Affinity matrices may also be computed to guide the LD embedding tuning [BN02, RS00, NLKC06, SSM97, Suy08]. However,

these approaches are hardly superior to the older methods in visualization tasks [LV07, LV09, vdMH08], potentially because they can be expressed as classical MDS applied in an unknown feature space [SSM98, HLMS04]. Distance-preserving schemes are indeed particularly affected by the norm concentration phenomenon [FWV07].

On the other hand, modern neighbor embedding (NE) visualization methods such as Stochastic Neighbor Embedding (SNE) [HR02] and variants [vdMH08, VPN⁺10] alleviate this phenomenon by matching neighbor similarity or probability distributions defined in both spaces to compute the LD points [LV11, MHM18]. The resulting outstanding DR performances enable accurately retrieving HD structures in the LD space (LDS) and motivated developing alternative SNE-based models, with heavy-tailed distributions as in *t*-SNE [vdMH08, YKXO09, dBMLS⁺19, GSH15], mixtures of divergences as cost functions [VPN⁺10, VK07, BHBV12, LRB⁺13], enhanced optimization [CP10, BHV⁺11, vdM10, VCP12, PL13, VCP13, GMH12, NJC04, YWO10, MH05], lower-complexity approximations [YPK13, vdM14, VCP14] and multi-scale, meta-parameter-free similarity definitions [LPOV15, dBMVL18b], revealing multiple organization levels of the data.

Notwithstanding these major achievements, enhancing HD data exploration in modern visualization contexts involves developing and extending NE algorithms in different regards, including the two following ones. First, NE schemes need to be **robust** with respect to various data types that are faced by users in practice. Such data types include databases with missing entries, which are often tackled in machine learning [BE10, All02]. Second, NE techniques must be **fast** enough to efficiently process big data sets, without limiting data exploration with too long running times. This thesis studies both aspects and designs NE approaches according to these two requirements for exploratory data visualization in current ML settings.

As to **robustness** to missing data, state-of-the-art NE algorithms can unfortunately not be directly employed on incomplete data sets, which become ubiquitous in machine learning [GLSGFVV09, DV12]. Discarding samples with missing features prevents their LD coordinates computation and deteriorates the complete samples treatment, as detailed in Chapter 3. Common missing data imputation schemes [HHE03] are also unsuitable in the nonlinear DR context. Indeed, even if they model the data distribution in the feature space, they can at best enable the application of a DR scheme on the expected data set. In practice, one would instead like to obtain the LD embedding with the closest cost function value on average with respect to the complete data case. As current NE techniques are nonlinear, the latter embedding results from minimizing the expected cost function on the incomplete database, not from considering the expected data set. This thesis addresses these limitations by developing a general methodology

for nonlinear DR with missing data [dBMVL19], being directly applicable with any DR scheme optimizing some criterion, as presented in Chapter 3. In order to model the feature dependencies, a high-dimensional extension of Gaussian mixture models is first fitted on the incomplete data set, as presented in Section 3.1. It is afterwards employed under the multiple imputation paradigm to obtain a single relevant LD embedding, minimizing the cost function expectation, as developed in Section 3.2. Extensive experiments in Section 3.3 demonstrate the superiority of the suggested framework over alternative approaches. Public implementations are available online at <https://github.com/cdebdot>.

Concerning the design of **fast** NE algorithms, both single-scale and multi-scale methods exhibit a time complexity in the number N of samples which is unaffordable for big data sets, as they evolve respectively in $\mathcal{O}(N^2)$ [HR02, vdMH08, VPN⁺10, LRB⁺13] and $\mathcal{O}(N^2 \log N)$ [LPOV15, dBMVL18b] time. Single-scale methods can be accelerated by relying on the inherent sparsity of the HD similarities they involve [YPK13, VCP14]. In particular, the Barnes-Hut (BH) approximation of t -SNE [vdM14] has a $\mathcal{O}(N \log N)$ average time complexity, while essentially delivering a similar DR quality as the original t -SNE formulation [dBMVL18a], as observed in Section 4.2. Meanwhile, the dense structure of the multi-scale HD similarities prevents developing fast multi-scale schemes in a similar way [LPOV15]. This difficulty is even more unpleasant as the single-scale or multi-scale nature of NE schemes drives their HD neighborhood preservation in the LD space; whereas single-scale methods focus on single-sized neighborhoods through the concept of perplexity [HR02], multi-scale ones preserve neighborhoods in a broader range of sizes and account for the global HD organization to define the LD space [LPOV15]. To open the path to the deployment of multi-scale NE approaches on very big databases, this thesis overcomes their time complexity challenge by designing randomized accelerations of these methods [dBMVL20], which run in $\mathcal{O}(N \log^2 N)$ average time as explained in Section 4.3. To account for all levels of interactions, the HD data are first sub-sampled at different scales, enabling to identify small and relevant neighbor sets for each data point thanks to vantage-point trees. Afterwards, these sets are employed with a BH algorithm to cheaply evaluate the considered cost function and its gradient, enabling large-scale use of multi-scale NE schemes. Thorough experiments in Section 4.4 demonstrate that the proposed accelerations are, statistically significantly over the random HD sub-samplings, both faster than the original multi-scale methods by orders of magnitude, and better preserving the HD neighborhoods than state-of-the-art single-scale schemes, leading to high-quality LD embeddings.

This thesis hence presents robust and fast NE methods able to, on the

one hand, deal with incomplete databases and, on the other hand, tackle large-scale data sets in the multi-scale setting. Such developments ultimately aim at facilitating HD data exploration in ML analysis tasks, by (1) allowing users to rely on algorithms that can effectively be applied on the data sets they encounter in practice, which frequently contain missing entries, and by (2) ensuring that computation times remain manageable in real exploration settings.

Among future perspectives, one can mention that, although Gaussian mixture models efficiently capture the feature distribution and enable to extend the applicability of NE methods to incomplete databases under the multiple imputation paradigm as introduced in Chapter 3, their large number of parameters limits their suitability when facing extremely HD databases. In such cases, non-parametric estimators of the data density could possibly better scale with the data dimension, by directly accounting for the sparsity of these very HD spaces. In addition, these data distribution models would also probably be more appropriate to data sets containing categorical information, which are also common in machine learning problems. Gaussian mixtures are indeed not primarily developed to purposely consider categorical features.

About the design of fast multi-scale NE algorithms, fast multipole methods [VCP14] can be confronted to the Barnes-Hut technique to determine whether they could better balance computation time with DR quality. Moreover, another line of further works finds inspiration from parametric DR mappings [BBH12, GSH15, Maa09, CdBV⁺20]. Such techniques are indeed well adapted to real-time application scenarios, as new data points can be processed on the fly once the NLDR projection has been learned. In this line, parametric NLDR embeddings can hence be very appropriate to conveniently expedite the data treatment, ensuing in a more pleasant data exploration for the user thanks to reduced running times.

In addition to robustness and speed, the current needs in HD data exploration in ML contexts challenge modern NLDR algorithms according to other aspects as well. In particular, enhancing the interaction between users and DR methods is key to efficiently explore the structure of large-scale databases in diverse tasks. Indeed, enabling the user to provide relevant feedbacks on the visualization helps to appropriately update it in an iterative way to focus on different aspects of the HD data. While DR schemes necessarily map data points by introducing some errors in the LD embeddings with respect to the HD structures due to the dimension reduction from the HD to the LD space, the faithfulness of a LD representation could be interactively depicted to the user besides the LD embedding itself. The DR quality criteria measuring the HD neighborhood preservation in the LD space as summarized in Section 3.3.1 would be very well suited for this pur-

pose. The LD errors could also be individually measured and represented, by evaluating the quality assessment criteria for each point separately. Interacting with the user would then help to visualize and identify the points which are incorrectly displayed in the LD embedding. Such interaction however crucially requires developing dedicated softwares to enable its practical deployment [Mun14]. Links between the NLDR and information visualization communities should therefore be reinforced [KMRV15], with the ultimate goal of more tightly connecting and bridging the gap between users and ML algorithms in data analysis frameworks.

Bibliography

- [ABB14] Oguz Akbilgic, Hamparsum Bozdogan, and M Erdal Balaban. A novel Hybrid RBF Neural Networks model as a forecaster. *Statistics and Computing*, 24(3):365–375, 2014. DOI 10.1007/s11222-013-9375-7.
- [Abr11] Nuno Abreu. *Análise do perfil do cliente Recheio e desenvolvimento de um sistema promocional*. PhD thesis, ISCTE-IUL, Lisbon, 2011.
- [Aka74] Hirotugu Akaike. A new look at the statistical model identification. *IEEE Trans. Autom. Control*, 19(6):716–723, 1974.
- [All02] Paul D Allison. Missing data: Quantitative applications in the social sciences. *British Journal of Mathematical and Statistical Psychology*, 55(1):193–196, 2002.
- [BB16] Johannes Blömer and Kathrin Bujna. Adaptive seeding for gaussian mixture models. In *Pacific-asia conference on knowledge discovery and data mining*, pages 296–308. Springer, 2016.
- [BBH12] Kerstin Bunte, Michael Biehl, and Barbara Hammer. A general framework for dimensionality-reducing data visualization mapping. *Neural Computation*, 24(3):771–804, 2012.
- [BCA⁺19] Anna C Belkina, Christopher O Ciccolella, Rina Anno, Richard Halpert, Josef Spidlen, and Jennifer E Snyder-Cappione. Automated optimized parameters for t-distributed stochastic neighbor embedding improve visualization and analysis of large datasets. *Nature communications*, 10(1):1–12, 2019.
- [BE10] Amanda Baraldi and Craig Enders. An introduction to modern missing data analyses. *Journal of school psychology*, 48(1):5–37, 2010.

- [BG05] Ingwer Borg and Patrick J F Groenen. *Modern Multidimensional Scaling: Theory and applications*. Springer Science & Business Media, 2005.
- [BGS07] Charles Bouveyron, Stéphane Girard, and Cordelia Schmid. High-dimensional data clustering. *Computational Statistics & Data Analysis*, 52(1):502–519, 2007.
- [BH86] J. Barnes and P. Hut. A hierarchical $O(N \log N)$ force-calculation algorithm. *Nature*, 324(6096):446–449, 1986.
- [Bha12] Rajen Bhatt. Planning-Relax Dataset for Automatic Classification of EEG Signals’. *UCI Machine Learning Repository*, 2012.
- [BHBV12] Kerstin Bunte, Sven Haase, Michael Biehl, and Thomas Villmann. Stochastic neighbor embedding (SNE) for dimension reduction and visualization using arbitrary divergences. *Neurocomputing*, 90:23–45, 2012.
- [BHP14] J.P. Bridge, S.B. Holden, and L.C. Paulson. Machine learning for first-order theorem proving: learning to select a good heuristic. *Journal of automated reasoning*, 53(2):141–172, 2014.
- [BHV⁺11] Kerstin Bunte, Barbara Hammer, Thomas Villmann, Michael Biehl, and Axel Wismüller. Neighbor embedding XOM for dimension reduction and visualization. *Neurocomputing*, 74(9):1340–1350, 2011.
- [Bis06] Christopher Bishop. *Pattern recognition and machine learning*. Springer, 2006.
- [BLNZ95] Richard H Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, 1995.
- [BMH⁺19] Etienne Becht, Leland McInnes, John Healy, Charles-Antoine Dutertre, Immanuel WH Kwok, Lai Guan Ng, Florent Ginhoux, and Evan W Newell. Dimensionality reduction for visualizing single-cell data using UMAP. *Nature biotechnology*, 37(1):38, 2019.
- [BMK05] Caroline Beunckens, Geert Molenberghs, and Michael G Kenward. Direct likelihood analysis versus simple forms

- of imputation for missing data in randomized clinical trials. *Clinical Trials*, 2(5):379–386, 2005.
- [BN02] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in neural information processing systems*, pages 585–591, 2002.
- [BPN16] Javad Birjandtalab, M Baran Pouyan, and Mehrdad Nourani. Nonlinear dimension reduction for EEG-based epileptic seizure detection. In *2016 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*, pages 595–598. IEEE, 2016.
- [BR82] Jacob Burbea and C Rao. On the convexity of some divergence measures based on entropy functions. *IEEE Transactions on Information Theory*, 28(3):489–495, 1982.
- [CA10] Andrzej Cichocki and Shun-ichi Amari. Families of alpha-beta-and gamma-divergences: Flexible and robust measures of similarities. *Entropy*, 12(6):1532–1568, 2010.
- [Cat66] Raymond B Cattell. The scree test for the number of factors. *Multivariate behavioral research*, 1(2):245–276, 1966.
- [CB09] Lisha Chen and Andreas Buja. Local multidimensional scaling for nonlinear dimension reduction, graph drawing, and proximity analysis. *Journal of the American Statistical Association*, 104(485):209–219, 2009.
- [CCA⁺09] P. Cortez, A. Cerdeira, F. Almeida, T. Matos, and J. Reis. Modeling wine preferences by data mining from physico-chemical properties. *Decision Support Systems*, 47(4):547–553, 2009.
- [CCR⁺19] Matthew C Cieslak, Ann M Castelfranco, Vittoria Roncalli, Petra H Lenz, and Daniel K Hartline. t-distributed stochastic neighbor embedding (t-SNE): A tool for eco-physiological transcriptomic analysis. *Marine Genomics*, page 100723, 2019.
- [CdBV⁺20] Francesco Crecchi, Cyril de Bodt, Michel Verleysen, John A. Lee, and Davide Bacciu. Perplexity-free parametric t-SNE. In *ESANN*, 2020.

- [CM07] P. Cortez and A. Morais. A Data Mining Approach to Predict Forest Fires using Meteorological Data. In *Proc. of the 13th EPIA - Portuguese Conference on Artificial Intelligence*, pages 512–523, 2007.
- [CP97] Miguel Carreira-Perpinán. A review of dimension reduction techniques. Technical Report CS-96-09, Dept. of Comp. Sci. Univ. of Sheffield., Jan 1997.
- [CP10] Miguel Carreira-Perpinán. The elastic embedding algorithm for dimensionality reduction. In *ICML*, volume 10, pages 167–174, 2010.
- [d’A71] Ralph B d’Agostino. An omnibus test of normality for moderate and large size samples. *Biometrika*, 58(2):341–348, 1971.
- [dBMLS⁺19] Cyril de Bodt, Dounia Mulders, Daniel López-Sánchez, Michel Verleysen, and John A. Lee. Class-aware t-SNE: cat-SNE. In *ESANN*, pages 409–414, 2019.
- [dBMVL18a] Cyril de Bodt, Dounia Mulders, Michel Verleysen, and John A. Lee. Extensive assessment of Barnes-Hut t-SNE. In *ESANN*, pages 135–140, 2018.
- [dBMVL18b] Cyril de Bodt, Dounia Mulders, Michel Verleysen, and John A. Lee. Perplexity-free t-SNE and twice Student t-SNE. In *ESANN*, pages 123–128, 2018.
- [dBMVL19] Cyril de Bodt, Dounia Mulders, Michel Verleysen, and John A. Lee. Nonlinear Dimensionality Reduction with Missing Data using Parametric Multiple Imputations. *IEEE Trans. Neural Netw. Learn. Syst.*, 30(4):1166–1179, April 2019. DOI: 10.1109/TNNLS.2018.2861891.
- [dBMVL20] Cyril de Bodt, Dounia Mulders, Michel Verleysen, and John A. Lee. Fast Multi-scale Neighbor Embedding. *IEEE Trans. Neural Netw. Learn. Syst.*, 2020. Under review.
- [DCB12] Olivier Delalleau, Aaron Courville, and Yoshua Bengio. Efficient em training of gaussian mixtures with missing data. *preprint arXiv:1209.0521*, 2012.
- [DH97] Pierre Demartines and Jeanny Hérault. Curvilinear component analysis: A self-organizing neural network for nonlinear mapping of data sets. *IEEE Transactions on neural networks*, 8(1):148–154, 1997.

- [DLR77] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.
- [DPATBEG19] Alex Diaz-Papkovich, Luke Anderson-Trocmé, Chief Ben-Eghan, and Simon Gravel. Umap reveals cryptic population structure and phenotype heterogeneity in large genomic cohorts. *PLoS genetics*, 15(11), 2019.
- [dRBP19] Gustavo H de Rosa, José RF Brega, and João P Papa. How optimizing perplexity can affect the dimensionality reduction on word embeddings visualization? *SN Applied Sciences*, 1(12):1673, 2019.
- [DV12] Gauthier Doquire and Michel Verleysen. Feature selection with missing data using mutual information estimators. *Neurocomputing*, 90:3–11, 2012.
- [EDVL13] Emil Eirola, Gauthier Doquire, Michel Verleysen, and Amaury Lendasse. Distance estimation in numerical data sets with missing values. *Information Sciences*, 240:115–128, 2013.
- [ELVB14] Emil Eirola, Amaury Lendasse, Vincent Vandewalle, and Christophe Biernacki. Mixture of Gaussians for distance estimation with missing data. *Neurocomputing*, 131:32–42, 2014.
- [End10] Craig K Enders. *Applied missing data analysis*. Guilford Press, 2010.
- [FC07] Stephen France and Douglas Carroll. Development of an agreement metric based upon the rand index for the evaluation of dimensionality reduction techniques, with applications to mapping customer data. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pages 499–517. Springer, 2007.
- [FWV07] Damien Francois, Vincent Wertz, and Michel Verleysen. The concentration of fractional distances. *IEEE Trans. Knowl. Data Eng.*, 19(7):873–886, 2007.
- [GFD16] Andreea Griparis, Daniela Faur, and Mihai Datcu. A dimensionality reduction approach to support visual data mining: Co-ranking-based evaluation. In *2016 Int. Conf. on Communications*, pages 391–394. IEEE, 2016.

- [GGBHD04] I. Guyon, S. Gunn, A. Ben-Hur, and G. Dror. Result analysis of the nips 2003 feature selection challenge. In *NIPS*, pages 545–552, 2004.
- [GGDJ⁺12] David Gil, Jose Luis Girela, Joaquin De Juan, M Jose Gomez-Torres, and Magnus Johnsson. Predicting seminal quality with artificial intelligence methods. *Expert Systems with Applications*, 39(16):12564–12573, 2012.
- [GJ95] Zoubin Ghahramani and Michael Jordan. Learning from incomplete data. *Tech.rep., Lab Memo 1509, CBCL Paper 108, MIT AI Lab*, 1995.
- [GLSGFVV09] Pedro J García-Laencina, José-Luis Sancho-Gómez, Aníbal R Figueiras-Vidal, and Michel Verleysen. K nearest neighbours with mutual information for simultaneous classification and missing data imputation. *Neurocomputing*, 72(7):1483–1493, 2009.
- [GMH12] Andrej Gisbrecht, Bassam Mokbel, and Barbara Hammer. Linear basis-function t-SNE for fast nonlinear dimensionality reduction. In *The International Joint Conference on Neural Networks*, pages 1–8. IEEE, 2012.
- [GSH15] Andrej Gisbrecht, Alexander Schulz, and Barbara Hammer. Parametric nonlinear dimensionality reduction using kernel t-SNE. *Neurocomputing*, 147:71–82, 2015.
- [HHE03] Eduardo Hruschka, Estevam Hruschka, and Nelson Ebecken. Evaluating a nearest-neighbor method to substitute continuous missing values. In *Australasian Joint Conference on Artificial Intelligence*, pages 723–734, 2003.
- [HJ03] Lynette Hunt and Murray Jorgensen. Mixture model clustering for mixed data with missing information. *Computational Statistics & Data Analysis*, 41(3):429–440, 2003.
- [HLMS04] Jihun Ham, Daniel D Lee, Sebastian Mika, and Bernhard Schölkopf. A kernel view of the dimensionality reduction of manifolds. In *Proceedings of the twenty-first international conference on Machine learning*, page 47. ACM, 2004.
- [HMF⁺16] R. Huerta, T. Mosqueiro, J. Fonollosa, N. Rulkov, and I. Rodriguez-Lujan. Online decorrelation of humidity and temperature in chemical sensors for continuous monitoring. *Chemometrics and Intelligent Laboratory Systems*, 157:169–176, 2016.

- [Hot33] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933.
- [HR02] Geoffrey Hinton and Sam Roweis. Stochastic neighbor embedding. In *NIPS*, volume 15, pages 833–840, 2002.
- [HT89] Clifford Hurvich and Chih-Ling Tsai. Regression and time series model selection in small samples. *Biometrika*, 76(2):297–307, 1989.
- [Jac88] R. A. Jacobs. Increased rates of convergence through learning rate adaptation. *Neural networks*, 1(4):295–307, 1988.
- [Jol86] Ian T Jolliffe. Principal component analysis and factor analysis. In *Principal component analysis*, pages 115–128. Springer, 1986.
- [Jol02] I. Jolliffe. *Principal component analysis*. Wiley Online Library, 2002.
- [KB19] Dmitry Kobak and Philipp Berens. The art of using t-sne for single-cell transcriptomics. *Nature communications*, 10(1):1–14, 2019.
- [KMRV15] Daniel A Keim, Tamara Munzner, Fabrice Rossi, and Michael Verleysen. Bridging information visualization with machine learning (dagstuhl seminar 15101). In *Dagstuhl Reports*, volume 5. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2015.
- [Koh90] Teuvo Kohonen. The self-organizing map. *Proc. IEEE*, 78(9):1464–1480, 1990.
- [KSC13] H Tolga Kahraman, Seref Sagiroglu, and Ilhami Colak. Developing intuitive knowledge classifier and modeling of users’ domain dependent data in web. *Knowledge-Based Systems*, 37:283–295, 2013.
- [KTG12] Heysem Kaya, Pınar Tüfekci, and Sadık Fikret Gürgen. Local and global learning methods for predicting power of a combined gas & steam turbine. In *Int. Conf. ETCEE*, pages 13–18, 2012.
- [LBBH98] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

- [Lic13] Moshe Lichman. UCI Machine Learning repository, 2013. URL: <http://archive.ics.uci.edu/ml>.
- [Lin91] Jianhua Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory*, 37(1):145–151, 1991.
- [LLDV00] John A. Lee, Amaury Lendasse, Nicolas Donckers, and Michel Verleysen. A robust non-linear projection method. In *ESANN*, pages 13–20, 2000.
- [LLH06] Tsung I Lin, Jack C Lee, and Hsiu J Ho. On fast supervised learning for normal mixture models with missing information. *Pattern Recognition*, 39(6):1177–1187, 2006.
- [LLV02] John A. Lee, Amaury Lendasse, and Michel Verleysen. Curvilinear Distance Analysis versus Isomap. In *ESANN*, volume 2, pages 185–192, 2002.
- [LLV04] John Aldo Lee, Amaury Lendasse, and Michel Verleysen. Nonlinear projection with curvilinear distances: Isomap versus curvilinear distance analysis. *Neurocomputing*, 57:49–76, 2004.
- [LMR⁺07] Max Little, Patrick McSharry, Stephen Roberts, Declan Costello, and Irene Moroz. Exploiting nonlinear recurrence and fractal scaling properties for voice disorder detection. *BioMedical Engineering OnLine*, 6(23), 2007.
- [LPOV15] John A Lee, Diego H Peluffo-Ordóñez, and Michel Verleysen. Multi-scale similarities in stochastic neighbour embedding: Reducing dimensionality while preserving both local and global structure. *Neurocomputing*, 169:246–261, 2015.
- [LR14] Roderick JA Little and Donald B Rubin. *Statistical analysis with missing data*. John Wiley & Sons, 2014.
- [LRB⁺13] John A Lee, Emilie Renard, Guillaume Bernard, Pierre Dupont, and Michel Verleysen. Type 1 and 2 mixtures of kullback-leibler divergences as cost functions in dimensionality reduction based on similarity preservation. *Neurocomputing*, 112:92–108, 2013.
- [LSdBL⁺20] Daniel López-Sánchez, Cyril de Bodt, John A. Lee, Angélica González Arrieta, and Juan M. Corchado. Tuning Database-Friendly Random Projection Matrices for Im-

- proved Distance Preservation on Specific Data. *Pattern Recognition*, 2020. Under review.
- [LV07] John A Lee and Michel Verleysen. *Nonlinear dimensionality reduction*. Springer Science & Business Media, 2007.
- [LV⁺08] J. A. Lee, M. Verleysen, et al. Rank-based quality assessment of nonlinear dimensionality reduction. In *ESANN*, pages 49–54, 2008.
- [LV09] John A Lee and Michel Verleysen. Quality assessment of dimensionality reduction: Rank-based criteria. *Neurocomputing*, 72(7):1431–1443, 2009.
- [LV10] John A Lee and Michel Verleysen. Scale-independent quality criteria for dimensionality reduction. *Pattern Recognition Letters*, 31(14):2248–2257, 2010.
- [LV11] 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.
- [LZN⁺19] Yun Liu, Rui Zhang, Feiping Nie, Xuelong Li, and Chris Ding. Supervised Dimensionality Reduction Methods via Recursive Regression. *IEEE Trans. Neural Netw. Learn. Syst.*, 2019. DOI: 10.1109/TNNLS.2019.2940088.
- [LZZ⁺18] Xuelong Li, Han Zhang, Rui Zhang, Yun Liu, and Feiping Nie. Generalized Uncorrelated Regression with Adaptive Graph for Unsupervised Feature Selection. *IEEE Trans. Neural Netw. Learn. Syst.*, 30(5):1587–1595, 2018.
- [Maa09] Laurens van der Maaten. Learning a parametric embedding by preserving local structure. In *Artificial Intelligence and Statistics*, pages 384–391, 2009.
- [MCG98] Andrew Morris, Martin Cooke, and Phil Green. Some solution to the missing feature problem in data classification, with application to noise robust asr. In *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing, 1998.*, volume 2, pages 737–740. IEEE, 1998.
- [MdBB⁺17] Dounia Mulders, Cyril de Bodt, Johannes Bjelland, Alex Pentland, Michel Verleysen, and Yves-Alexandre de Montjoye. Improving individual predictions using social net-

- works assortativity. In *12th International Workshop on Self-Organizing Maps and Learning Vector Quantization, Clustering and Data Visualization (WSOM+)*, pages 127–134. IEEE, 2017.
- [MdB⁺19] Dounia Mulders, Cyril de Bodt, Johannes Bjelland, Alex Pentland, Michel Verleysen, and Yves-Alexandre de Montjoye. Inference of node attributes from social network assortativity. *Neural Computing and Applications*, pages 1–21, 2019. doi:<https://doi.org/10.1007/s00521-018-03967-z>.
- [MdB⁺18a] Dounia Mulders, Cyril de Bodt, Nicolas Lejeune, André Mouraux, and Michel Verleysen. Linear periodic discriminant analysis of multidimensional signals. In *International Conference on Neural Information Processing (ICONIP)*, pages 476–487. Springer, 2018.
- [MdB⁺18b] Dounia Mulders, Cyril de Bodt, Nicolas Lejeune, André Mouraux, and Michel Verleysen. Spatial filtering of EEG signals to identify periodic brain activity patterns. In *International Conference on Latent Variable Analysis and Signal Separation (LVA/ICA)*, pages 524–533. Springer, 2018.
- [MdB⁺19] Dounia Mulders, Cyril de Bodt, Nicolas Lejeune, John A. Lee, André Mouraux, and Michel Verleysen. Tensor factorization to extract patterns in multimodal EEG data. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, pages 601–606, 2019.
- [MdB⁺20] Dounia Mulders, Cyril de Bodt, Nicolas Lejeune, Arthur Courtin, Giulia Liberati, Michel Verleysen, and André Mouraux. Dynamics of the perception and EEG signals triggered by tonic warm and cool stimulation. *PLoS One*, 2020.
- [MdBV⁺17] Dounia Mulders, Cyril de Bodt, Michel Verleysen, Johannes Bjelland, Alex Pentland, and Yves-Alexandre de Montjoye. Improving individual predictions using social networks assortativity. In *Proceedings of the Twenty-Sixth Benelux Conference on Machine Learning (BENELEARN)*, pages 134–136, 2017.

- [MGH10] Bassam Mokbel, Andrej Gisbrecht, and Barbara Hammer. On the effect of clustering on quality assessment measures for dimensionality reduction. In *NIPS workshop on Challenges of Data Visualization*, 2010.
- [MGH11] Bassam Mokbel, Andrej Gisbrecht, and Barbara Hammer. Quality assessment measures for dimensionality reduction applied on clustering. *ICOLE 2010*, page 75, 2011.
- [MGL⁺12] Bassam Mokbel, Sebastian Gross, Markus Lux, Niels Pinkwart, and Barbara Hammer. How to quantitatively compare data dissimilarities for unsupervised machine learning? In *IAPR Workshop on Artificial Neural Networks in Pattern Recognition*, pages 1–13. Springer, 2012.
- [MH05] Roland Memisevic and Geoffrey Hinton. Improving dimensionality reduction with spectral gradient descent. *Neural networks*, 18(5-6):702–710, 2005.
- [MHM18] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.
- [MLGH13] Bassam Mokbel, Wouter Lueks, Andrej Gisbrecht, and Barbara Hammer. Visualizing the quality of dimensionality reduction. *Neurocomputing*, 112:109–123, 2013.
- [MLP13] R.C.B. Madeo, C.A.M. Lima, and S.M. Peres. Gesture unit segmentation using support vector machines: segmenting gestures from rest positions. In *Proc. SAC*, pages 46–52. ACM, 2013.
- [Mun14] Tamara Munzner. *Visualization Analysis and Design*. CRC Press, 2014.
- [NC15] C. Nugteren and V. Codreanu. Cltune: A generic auto-tuner for opencl kernels. In *MCSoc*, pages 195–202. IEEE, 2015.
- [NJC04] Kijoeng Nam, Hongmo Je, and Seungjin Choi. Fast stochastic neighbor embedding: a trust-region algorithm. In *IJCNN*, volume 1, pages 123–128. IEEE, 2004.
- [NLKC06] Boaz Nadler, Stephane Lafon, Ioannis Kevrekidis, and Ronald R Coifman. Diffusion maps, spectral clustering and

- eigenfunctions of fokker-planck operators. In *Advances in neural information processing systems*, pages 955–962, 2006.
- [NNM96] S. A. Nene, S. K. Nayar, and H. Murase. Columbia object image library (COIL-20), 1996.
- [PG12] Jaakko Peltonen and Konstantinos Georgatzis. Efficient optimization for data visualization as an information retrieval task. In *MLSP*, pages 1–6, Santander, Spain, September 2012. IEEE.
- [PL13] Jaakko Peltonen and Ziyuan Lin. Multiplicative update for fast optimization of information retrieval based neighbor embedding. In *International Workshop on Machine Learning for Signal Processing*, pages 1–6, 2013.
- [Pla13] Alexander Platzer. Visualization of SNPs with t-SNE. *PloS one*, 8(2), 2013.
- [PTM⁺19] Nicola Pezzotti, Julian Thijssen, Alexander Mordvintsev, Thomas Höllt, Baldur Van Lew, Boudewijn PF Lelieveldt, Elmar Eisemann, and Anna Vilanova. GPGPU linear complexity t-sne optimization. *IEEE transactions on visualization and computer graphics*, 26(1):1172–1181, 2019.
- [RS00] Sam Roweis and Lawrence Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.
- [Rub04] Donald B Rubin. *Multiple imputation for nonresponse in surveys*, volume 81. John Wiley & Sons, 2004.
- [S⁺78] Gideon Schwarz et al. Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464, 1978.
- [Sam69] John W Sammon. A nonlinear mapping for data structure analysis. *IEEE Trans. Comput.*, 100(5):401–409, 1969.
- [SdBL⁺20] Walter Serna, Cyril de Bodt, John A. Lee, Andrés M. Alvarez, Michel Verleysen, and Alvaro A. Orozco. Semi-supervised neighbor embedding with multi-scale neighborhood preservation. *Journal of Artificial Intelligence Research*, 2020. Under review.
- [SEM⁺16] Dušan Sovilj, Emil Eirola, Yoan Miche, Kaj-Mikael Björk, Rui Nian, Anton Akusok, and Amaury Lendasse. Extreme

- learning machine for missing data using multiple imputations. *Neurocomputing*, 174:220–231, 2016.
- [Sha95] Juliet Popper Shaffer. Multiple hypothesis testing. *Annual review of psychology*, 46(1):561–584, 1995.
- [SMdS13] Pedro FB Silva, Andre RS Marcal, and Rubim M Almeida da Silva. Evaluation of features for leaf discrimination. *Springer Lecture Notes in Computer Science*, 7950:197–204, 2013.
- [SSM97] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Kernel principal component analysis. In *International Conference on Artificial Neural Networks*, pages 583–588. Springer, 1997.
- [SSM98] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10(5):1299–1319, 1998.
- [Suy08] Johan Suykens. Data visualization and dimensionality reduction using kernel maps with a reference point. *IEEE Trans. Neural Netw.*, 19(9):1501–1517, 2008.
- [SVGVDM03] Johan Suykens, Tony Van Gestel, Joos Vandewalle, and Bart De Moor. A support vector machine formulation to PCA analysis and its kernel version. *IEEE Trans. on neural networks*, 14(2):447–450, 2003.
- [SWR10] Russell J Steele, Naisyin Wang, and Adrian E Raftery. Inference from multiple imputation for missing data using mixtures of normals. *Statistical methodology*, 7(3):351–365, 2010.
- [TDSL00] Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.
- [Ten98] Joshua B Tenenbaum. Mapping a manifold of perceptual observations. In *Advances in neural information processing systems*, pages 682–688, 1998.
- [TLZM16] Jian Tang, Jingzhou Liu, Ming Zhang, and Qiaozhu Mei. Visualizing large-scale and high-dimensional data. In *Proc. of the 25th int. conf. on WWW*, pages 287–297. International World Wide Web Conferences Steering Committee, 2016.

- [TSM85] D Michael Titterington, Adrian FM Smith, and Udi E Makov. *Statistical analysis of finite mixture distributions*. Wiley,, 1985.
- [Tüf14] Pınar Tüfekci. Prediction of full load electrical power output of a base load operated combined cycle power plant using machine learning methods. *Int. J. Electrical Power & Energy Systems*, 60:126–140, 2014.
- [TX12] Athanasios Tsanas and Angeliki Xifara. Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools. *Energy and Buildings*, 49:560–567, 2012.
- [VBBGOR06] Stef Van Buuren, Jaap Brand, C Groothuis-Oudshoorn, and Donald Rubin. Fully conditional specification in multivariate imputation. *Journal of statistical computation and simulation*, 76(12):1049–1064, 2006.
- [VCP12] Max Vladymyrov and Miguel Carreira-Perpinán. Partial-hessian strategies for fast learning of nonlinear embeddings. In *ICML*, volume 29, pages 345–352, 2012.
- [VCP13] Max Vladymyrov and Miguel Carreira-Perpinán. Entropic affinities: Properties and efficient numerical computation. In *ICML*, pages 477–485, 2013.
- [VCP14] Max Vladymyrov and Miguel Carreira-Perpinan. Linear-time training of nonlinear low-dimensional embeddings. In *Artificial Intelligence and Statistics*, pages 968–977, 2014.
- [vdM10] Laurens van der Maaten. Fast optimization for t-SNE. In *NIPS Workshop on Challenges in Data Visualization*, 2010.
- [vdM14] Laurens van der Maaten. Accelerating t-SNE using tree-based algorithms. *Journal of machine learning research*, 15(1):3221–3245, 2014.
- [vdMH08] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.
- [VHK14] Jason Van Hulse and Taghi M Khoshgoftaar. Incomplete-case nearest neighbor imputation in software measurement data. *Information Sciences*, 259:596–610, 2014.

- [VK05] Jarkko Venna and Samuel Kaski. Local multidimensional scaling with controlled tradeoff between trustworthiness and continuity. In *Proceedings of WSOM*, volume 5, pages 695–702. Citeseer, 2005.
- [VK06] Jarkko Venna and Samuel Kaski. Local multidimensional scaling. *Neural Networks*, 19(6-7):889–899, 2006.
- [VK07] Jarkko Venna and Samuel Kaski. Nonlinear dimensionality reduction as information retrieval. In *Artificial Intelligence and Statistics*, pages 572–579, 2007.
- [Vla19] Max Vladymyrov. No pressure! addressing the problem of local minima in manifold learning algorithms. In *Advances in Neural Information Processing Systems*, pages 678–687, 2019.
- [VPN⁺10] Jarkko Venna, Jaakko Peltonen, Kristian Nybo, Helena Aidos, and Samuel Kaski. Information retrieval perspective to nonlinear dimensionality reduction for data visualization. *Journal of Machine Learning Research*, 11(Feb):451–490, 2010.
- [WM05] Ito Wasito and Boris Mirkin. Nearest neighbour approach in the least-squares data imputation algorithms. *Information Sciences*, 169(1):1–25, 2005.
- [WYY⁺10] Jinjun Wang, Jianchao Yang, Kai Yu, Fengjun Lv, Thomas Huang, and Yihong Gong. Locality-constrained linear coding for image classification. In *CVPR*, pages 3360–3367. IEEE, 2010.
- [Yeh98] I-Cheng Yeh. Modeling of strength of high-performance concrete using artificial neural networks. *Cement and Concrete research*, 28(12):1797–1808, 1998.
- [Yeh07] I-Cheng Yeh. Modeling slump flow of concrete using second-order regressions and artificial neural networks. *Cement and Concrete Composites*, 29(6):474–480, 2007.
- [Yia93] P. Yianilos. Data structures and algorithms for nearest neighbor search in general metric spaces. In *SODA*, volume 93, pages 311–321, 1993.
- [YKXO09] Zhirong Yang, Irwin King, Zenglin Xu, and Erkki Oja. Heavy-tailed symmetric stochastic neighbor embedding. In

- Advances in neural information processing systems*, pages 2169–2177, 2009.
- [YME⁺13] Qi Yu, Yoan Miche, Emil Eirola, Mark Van Heeswijk, Eric Séverin, and Amaury Lendasse. Regularized extreme learning machine for regression with missing data. *Neurocomputing*, 102:45–51, 2013.
- [YPK13] Zhirong Yang, Jaakko Peltonen, and Samuel Kaski. Scalable optimization of neighbor embedding for visualization. In *ICML (2)*, pages 127–135, 2013.
- [YWO10] Zhirong Yang, Chiwei Wang, and Erkki Oja. Multiplicative updates for t-SNE. In *MLSP*, pages 19–23. IEEE, 2010.
- [ZJ20] Bo Zhou and Wenfei Jin. Visualization of single cell rna-seq data using t-sne in r. In *Stem Cell Transcriptional Networks*, pages 159–167. Springer, 2020.
- [ZNG⁺18] Rui Zhang, Feiping Nie, Muhan Guo, Xian Wei, and Xuelong Li. Joint Learning of Fuzzy k-Means and Nonnegative Spectral Clustering With Side Information. *IEEE Trans. Image Process.*, 28(5):2152–2162, 2018.
- [ZNL16] Rui Zhang, Feiping Nie, and Xuelong Li. Regularized Class-Specific Subspace Classifier. *IEEE Trans. Neural Netw. Learn. Syst.*, 28(11):2738–2747, 2016.
- [ZNL17] Rui Zhang, Feiping Nie, and Xuelong Li. Self-Weighted Supervised Discriminative Feature Selection. *IEEE Trans. Neural Netw. Learn. Syst.*, 29(8):3913–3918, 2017.
- [ZNL18] Rui Zhang, Feiping Nie, and Xuelong Li. Semisupervised Learning With Parameter-Free Similarity of Label and Side Information. *IEEE Trans. Neural Netw. Learn. Syst.*, 30(2):405–414, 2018.
- [ZNWL19] Rui Zhang, Feiping Nie, Yunhai Wang, and Xuelong Li. Unsupervised Feature Selection via Adaptive Multimeasure Fusion. *IEEE Trans. Neural Netw. Learn. Syst.*, 30(9):2886–2892, 2019.
- [Zou60] G. Zoutendijk. *Methods of feasible directions: a study in linear and non-linear programming*. Elsevier, 1960.

List of symbols and acronyms

$\tilde{g}^t$	Approximation of g^t (p. 63)
$\tilde{g}_v^t$	Approximation of g_v^t (p. 63)
$\hat{\mathbf{g}}_v^t$	Approximation of $\mathbf{g}_v^t$ (p. 63)
a_h^c	Additional attribute stored in cell c of the space-partitioning tree to approximate (4.31) (p. 86)
AUC	Area under the $R_{NX}(K)$ curve displayed with a log-scale for K (p. 46)
θ	Barnes-Hut threshold (p. 63)
$ \mathcal{A} $	Cardinality of set $\mathcal{A}$ (p. 33)
C_{DR}	Cost function of a DR algorithm (p. 30)
$C_{j\star}$	Cost function of single-scale JSE (p. 17)
C_j	Cost function of multi-scale JSE (p. 25)
C_n	Cost function of multi-scale NeRV (p. 25)
C	Cost function of multi-scale SNE (p. 23)
C_t	Cost function of multi-scale t -SNE (p. 23)
$C_{n\star}$	Cost function of single-scale NeRV (p. 16)
$\tilde{C}$	Cost function of fast multi-scale SNE (p. 84)
$\tilde{C}_t$	Cost function of fast multi-scale t -SNE (p. 84)
$C_\star$	Cost function of single-scale SNE (p. 14)

- $C_{t\star}$ _____ Cost function of single-scale t -SNE (p. 16)
- C_{ik} _____ Cholesky factor of $\tilde{\Sigma}_{k,M_i|O_i}$ (p. 38)
- $\mathbf{y}_{ik}$ — Vector drawn from a $|M_i|$ -dimensional Gaussian distribution, with zero mean and identity covariance (p. 38)
- $\Sigma_k^{AB_i}$ Sub-matrix of Σ_k with rows indexed by A_i and columns by B_i (p. 33)
- $\tilde{\Sigma}_{k,M_i|O_i}$ _____ Conditional covariance of the missing features of the sample ξ_i given $\xi_i^{O_i}$ under the component $k \in \mathcal{G}$ of a Gaussian mixture model (p. 33)
- $\bar{\Sigma}_k$. Matrix replacing Σ_k defined in (3.17) when using HDDC models (p. 35)
- Σ_i . Covariance matrix of the i^{th} component of a Gaussian mixture model (p. 32)
- $\tilde{\Sigma}_{ik}^{MM_i}$ _____ Sub-matrix of $\tilde{\Sigma}_{ik}$ with rows and columns indexed by M_i (p. 34)
- $\tilde{\Sigma}_{ik}$ _____ Matrix in $\mathbb{R}^{\Pi \times \Pi}$ defined as $\tilde{\Sigma}_{ik}^{MM_i} = \varphi_{ik} \tilde{\Sigma}_{k,M_i|O_i}$ and zero elsewhere (p. 34)
- $\bar{s}_{ih}$ _____ Denominator of s_{ijh} (p. 85)
- δ_{ij} _____ HD distance between ξ_i and ξ_j (p. 9)
- d_k _____ Hyper-parameter in HDDC methods, belonging to $\{1, \dots, \Pi\}$ (p. 35)
- Π _____ Dimension of the HD space, equal to the number of features of the HD data set (p. 9)
- P _____ Dimension of the LD space (p. 9)
- d_{ij} _____ LD distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ (p. 9)
- $d_{v,\mathbf{m}_c}$ _____ LD distance between $\mathbf{x}_v$ and $\mathbf{m}_c$ (p. 63)
- $\sigma_{i\star}$ — Set of the single-scale SNE HD similarities with respect to ξ_i , equal to $\{\sigma_{ij\star}, j \in \mathcal{I} \setminus \{i\}\}$ (p. 10)
- $s_{i\star}$ — Set of the single-scale SNE LD similarities with respect to $\mathbf{x}_i$, equal to $\{s_{ij\star}, j \in \mathcal{I} \setminus \{i\}\}$ (p. 10)
- $\|\xi_i\|_2$ _____ Euclidean norm of ξ_i (p. 10)
- $e_{i\star}$ — Entropy of the single-scale SNE HD similarities in $\{\sigma_{ij\star}, j \in \mathcal{I} \setminus \{i\}\}$ (p. 13)
- $\mathbb{E}[\cdot]$ _____ Expectation operator (p. 30)

$\bar{\Lambda}_k$ _____ $\Pi \times \Pi$ diagonal matrix defined in (3.25) (p. 35)

Λ_k _____ Diagonal matrix containing the eigenvalues of Σ_k sorted in decreasing order (p. 35)

$\mathbf{Q}_k$ _____ Matrix containing the eigenvectors of Σ_k in its columns (p. 35)

ζ _____ Scalar employed to tune η_k in Section 3.1.5 (p. 37)

f_{vi} _____ A term of $\frac{\partial \tilde{C}}{\partial \mathbf{x}_v}$, defined in (4.25) (p. 85)

g_{vh} _____ A term of $\frac{\partial \tilde{C}}{\partial \mathbf{x}_v}$, defined in (4.26) (p. 85)

$\mathbf{g}_{vh}$ _____ A term of $\frac{\partial \tilde{C}}{\partial \mathbf{x}_v}$, defined in (4.26) (p. 85)

$\mathbf{f}_v^{\text{t}\star}$ _____ Term in $\frac{\partial C_{\text{t}\star}}{\partial \mathbf{x}_v}$ (p. 61)

g^{t} _____ Term in $\frac{\partial C_{\text{t}\star}}{\partial \mathbf{x}_v}$ (p. 61)

g_v^{t} _____ Defined in (4.9) (p. 63)

$\mathbf{g}_v^{\text{t}}$ _____ Term in $\frac{\partial C_{\text{t}\star}}{\partial \mathbf{x}_v}$ (p. 61)

$\tilde{\mathcal{I}}_{ih}$ _____ Set of the indexes of the $\lfloor 3K_1 \rfloor$ nearest neighbors of ξ_i in $\Xi_h \setminus \{\xi_i\}$ (p. 82)

$\mathcal{I}_i$ _____ Full multi-scale neighbor set of ξ_i (p. 78)

$\tilde{\mathcal{I}}_i$ _____ Set of the indexes of the multi-scale neighbors of ξ_i in $\Xi \setminus \{\xi_i\}$ (p. 81)

$\mathcal{I}_{ih}$ _____ Set of the indexes of the $\min(\lfloor 3K_h \rfloor, N - 1)$ nearest neighbors of ξ_i in $\Xi \setminus \{\xi_i\}$ (p. 78)

$\mathcal{I}_{i\star}$ _____ Set of the indexes of the $\lfloor 3K_\star \rfloor$ nearest neighbors of ξ_i in $\Xi \setminus \{\xi_i\}$ (p. 60)

$\mathcal{I}$ _____ Set of the indexes of the data points, equal to $\{1, \dots, N\}$ (p. 9)

$\mathcal{G}$ _____ Set of the indexes of the components of a Gaussian mixture model, equal to $\{1, \dots, G\}$ (p. 32)

M_i _____ Set of the indexes of the missing features of ξ_i , which is a subset of $\{1, \dots, \Pi\}$ (p. 33)

O_i _____ Set of the indexes of the observed features of ξ_i , which is a subset of $\{1, \dots, \Pi\}$ (p. 33)

$\mathcal{T}_c$ — Set of the indexes of the LD points inside cell c of a space-partitioning tree (p. 61)

$z_{ij\star}^l$ ————— Equal to $\kappa_j \sigma_{ij\star}^l + (1 - \kappa_j) s_{ij\star}$ (p. 41)

z_{ij} ————— Equal to $\kappa_j \sigma_{ij} + (1 - \kappa_j) s_{ij}$ (p. 25)

$z_{ij\star}$ ————— Equal to $\kappa_j \sigma_{ij\star} + (1 - \kappa_j) s_{ij\star}$ (p. 17)

ν_i^K ——— K -ary neighborhood of ξ_i , which is the set of the indexes of its K nearest neighbors in the HD space (p. 45)

n_i^K ——— K -ary neighborhood of $\mathbf{x}_i$, which is the set of the indexes of its K nearest neighbors in the LD space (p. 45)

$\overline{K}$ — Upper bound for K to compute the area $\overline{\text{AUC}}$ of the $R_{\text{NX}}(K)$ curve (p. 100)

α ————— Level of significance of the statistical tests (p. 47)

$\log \mathcal{L}(\vartheta)$ — Likelihood of a Gaussian mixture model with parameters ϑ (p. 32)

$\mathbf{I}_\Pi$ ————— Identity matrix of order Π (p. 10)

$\mathcal{N}(\xi_i^{O_i} | \mu_k, \Sigma_k)$ — Marginal of $\mathcal{N}(\xi_i | \mu_k, \Sigma_k)$ over the observed features of ξ_i (p. 33)

$\mathcal{N}(\xi_i | \mu_k, \Sigma_k)$ ——— Evaluation of the probability density function of the k^{th} component of a Gaussian mixture model over ξ_i (p. 32)

ϑ ————— Set of the parameters of a Gaussian mixture model, equal to $\{w_k, \mu_k, \Sigma_k; k \in \mathcal{G}\}$ (p. 32)

$\mathbf{m}_c$ ————— Center-of-mass of the points indexed in $\mathcal{T}_c$ (p. 61)

$\tilde{\mu}_{k, M_i | O_i}$ ——— Conditional expectation of the missing features of the sample ξ_i given $\xi_i^{O_i}$ under the component $k \in \mathcal{G}$ of a Gaussian mixture model (p. 33)

$\mu_k^{M_i}$ ————— Components of μ_k indexed by M_i (p. 33)

$\mu_k^{O_i}$ ————— Components of μ_k indexed by O_i (p. 33)

$\tilde{\mu}_{ik}$ — Vector in $\mathbb{R}^\Pi$ whose components indexed by O_i (resp. M_i) are equal to $\mu_k^{O_i} - \xi_i^{O_i}$ (resp. $\mu_k^{M_i}$) (p. 35)

μ_i — Mean vector of the i^{th} component of a Gaussian mixture model (p. 32)

N ————— Number of data points in a data set (p. 9)

N_k _____ Equal to $\sum_{i \in \mathcal{I}} \varphi_{ik}$ (p. 32)

G _____ Number of components in a Gaussian mixture model (p. 32)

Ψ _____ Number of imputations in multiple imputations techniques (p. 38)

P_{ϑ} _____ Number of parameters of the Gaussian mixture model characterized by ϑ (p. 37)

H _____ Number of scales in multi-scale methods (p. 23)

$\ell(\Xi | \vartheta)$ _____ Likelihood of a Gaussian mixture model with parameters ϑ on a data set Ξ (p. 32)

w_i _____ Weight coefficient of the i^{th} component of a Gaussian mixture model (p. 32)

p_m _____ Proportion of missing values introduced in a data set with respect to its number NII of entries (p. 46)

$\pi_{i\star}$ _____ Precision of the single-scale SNE HD similarities $\sigma_{ij\star}$ for $j \in \mathcal{I} \setminus \{i\}$ (p. 13)

π_{ih} _____ Precision of the single-scale SNE HD similarities σ_{ijh} for $j \in \mathcal{I} \setminus \{i\}$ (p. 23)

p_h _____ Precision of the single-scale SNE LD similarities s_{ijh} for $i \in \mathcal{I}$ and $j \in \mathcal{I} \setminus \{i\}$ (p. 23)

p_{ih} _____ Precision of the single-scale SNE LD similarities s_{ijh} for $j \in \mathcal{I} \setminus \{i\}$ (p. 23)

$p_{\star}$ _____ Value of $p_{i\star}$ for all $i \in \mathcal{I}$ (p. 13)

$p_{i\star}$ _____ Precision of the single-scale SNE LD similarities $s_{ij\star}$ for $j \in \mathcal{I} \setminus \{i\}$ (p. 13)

$K_{\star}$ _____ Perplexity (p. 13)

K_h _____ Perplexity at scale h (p. 23)

Q_{KL} _____ Sum of the KL divergences between the “true” and approximated HD similarities for each data point when introducing missing entries in a HD data set (p. 45)

$Q_{\text{NX}}(K)$ _____ Average normalized agreement of the K nearest neighbor sets of the data points in the HD and LD spaces (p. 45)

$\mathbb{R}$ _____ Real numbers (p. 10)

- $\overline{\text{AUC}}$ — Area under the $R_{\text{NX}}(K)$ curve for K ranging from 1 to $\overline{K}$ (p. 100)
- ρ_{ij} — HD rank of ξ_j with respect to ξ_i (p. 45)
- r_{ij} — LD rank of $\mathbf{x}_j$ with respect to $\mathbf{x}_i$ (p. 45)
- $R_{\text{NX}}(K)$ — Rescaled version of $Q_{\text{NX}}(K)$ to account for the fact that $\mathbb{E}[Q_{\text{NX}}(K)]$ equals $\frac{K}{N-1}$ for random LD points (p. 46)
- $[\cdot]$ — Rounding operator (p. 23)
- r_c — Cell c diagonal length in a space-partitioning tree (p. 63)
- $\mathbf{E}_r$ — Symmetric matrix resulting from reverse sweeping $\mathbf{E}$ on its row and column l (p. 34)
- $S_{\mathcal{A}}^{-1}(\mathbf{E})$ — Reverse sweep operator applied on the set of rows and columns indexed in $\mathcal{A}$ of the symmetric matrix $\mathbf{E}$ (p. 34)
- γ_i — Shift value in (2.5) for the HD similarities $\sigma_{ij\star}$ for $j \in \mathcal{I} \setminus \{i\}$ (p. 13)
- $\hat{\sigma}_{ij}$ — Approximation of σ_{ij} obtained by a method managing missing values in a data set (p. 45)
- σ_{ij}^l — Multi-scale SNE HD similarity between ξ_i and ξ_j from Ξ^l (p. 38)
- τ_{ij} — Multi-scale t -SNE HD similarity between ξ_i and ξ_j (p. 23)
- σ_{ij} — Multi-scale SNE HD similarity between ξ_i and ξ_j (p. 23)
- $\sigma_{ij\star}$ — Single-scale SNE HD similarity between ξ_i and ξ_j using perplexity $K_{\star}$ (p. 10)
- $\sigma_{ij\star}^l$ — Single-scale SNE HD similarity between ξ_i and ξ_j from Ξ^l , using perplexity $K_{\star}$ (p. 41)
- $\sigma_{ij\star}^n$ — Single-scale NeRV HD similarity between ξ_i and ξ_j using perplexity $K_{\star}$ (p. 16)
- $\tau_{ij\star}$ — Single-scale t -SNE HD similarity between ξ_i and ξ_j using perplexity $K_{\star}$ (p. 16)
- σ_{ijh} — Single-scale SNE HD similarity between ξ_i and ξ_j using perplexity K_h (p. 23)
- s_{ijh} — Single-scale SNE LD similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$, using perplexity K_h for σ_{ijh} (p. 23)
- s_{ij} — Multi-scale SNE LD similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$ (p. 23)

$s_{ij\star}$ — Single-scale SNE LD similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$, using perplexity $K_\star$ for $\sigma_{ij\star}$ (p. 10)

$s_{ij\star}^n$. Single-scale NeRV LD similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$, using perplexity $K_\star$ for $\sigma_{ij\star}^n$ (p. 16)

t_{ij} ————— t -SNE LD similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$ (p. 16)

$\tilde{\pi}_{i\star}$ ————— Precision of the sparse single-scale SNE HD similarities $\tilde{\sigma}_{ij\star}$ for $j \in \mathcal{I} \setminus \{i\}$ (p. 60)

$\tilde{\pi}_{ih}$ ————— Precision of the sparse single-scale SNE HD similarities $\tilde{\sigma}_{ijh}$ for $j \in \mathcal{I} \setminus \{i\}$ (p. 83)

$\tilde{\tau}_{ij}$ ————— Sparse multi-scale t -SNE HD similarity between ξ_i and ξ_j (p. 83)

$\tilde{\sigma}_{ij}$ ————— Sparse multi-scale SNE HD similarity between ξ_i and ξ_j (p. 83)

$\tilde{\tau}_{ij\star}$ ————— Sparse estimate for $\tau_{ij\star}$ (p. 60)

$\tilde{\sigma}_{ij\star}$ ————— Sparse approximation of $\sigma_{ij\star}$ (p. 60)

$\tilde{\sigma}_{ijh}^\pi$ ————— Defined in (4.19) to tune $\tilde{\pi}_{ih}$ (p. 83)

$\tilde{\sigma}_{ijh}$ ————— Sparse approximation of σ_{ijh} (p. 82)

$\tilde{s}_{ijh}$ ————— Sparse approximation of s_{ijh} (p. 83)

$\tilde{s}_{ij}$ ————— Sparse multi-scale SNE LD similarity between ξ_i and ξ_j (p. 83)

$\mathbf{E}$ ————— Symmetric matrix (p. 34)

$\mathbf{E}_s$. Symmetric matrix resulting from sweeping $\mathbf{E}$ on its row and column l (p. 34)

$\mathcal{S}_{\mathcal{A}}(\mathbf{E})$. Sweep operator applied on the set of rows and columns indexed in $\mathcal{A}$ of the symmetric matrix $\mathbf{E}$ (p. 34)

T_{MG} Time complexity of an EM iteration to fit a Gaussian mixture model on an incomplete data set (p. 36)

φ_{ik} — Probability for the HD example ξ_i to originate from the k^{th} component of a Gaussian mixture model (p. 32)

$\text{tr}(\mathbf{A})$ ————— Trace of the matrix $\mathbf{A}$ (p. 35)

η_k ————— Threshold in the scree test to tune d_k (p. 36)

κ_j ————— Meta-parameter trading off the terms in $C_{j\star}$ (p. 17)

κ_n	Meta-parameter trading off the terms in C_{n*} (p. 16)
Ξ	Complete HD data set (p. 9)
Ξ^i	i^{th} imputed HD data set (p. 38)
Ξ_h	Sub-sampled HD data set at scale h (p. 81)
Ξ_m	Incomplete HD data set (p. 30)
$\tilde{\xi}_{ik}$	Vector in $\mathbb{R}^I$ with its components indexed by O_i and M_i equal to $\xi_i^{O_i}$ and $\tilde{\mu}_{k, M_i O_i}$ (p. 34)
$\xi_i^{M_i}$	Missing features of ξ_i (p. 33)
Ξ_m^O	Set gathering $\xi_i^{O_i}$ for all $i \in \mathcal{I}$ (p. 33)
$\xi_i^{O_i}$	Observed features of ξ_i (p. 33)
ξ_i^T	Transpose of ξ_i (p. 10)
ξ_i	i^{th} HD point in a data set (p. 9)
$\mathbf{X}$	LD data set (p. 9)
$\mathbf{X}_h$	Sub-sampled LD data set at scale h (p. 83)
$\mathbf{x}_i^T$	Transpose of $\mathbf{x}_i$ (p. 13)
$\mathbf{x}_i$	LD representation of ξ_i (p. 9)
K	Neighborhood size in DR quality assessment criteria (p. 45)
t -SNE	Student t -distributed SNE (p. 15)
INN	Incomplete-case k -nearest neighbor imputation (p. 44)
MIG	Multiple imputations using a single multivariate Gaussian (p. 44)
IG	Conditional mean imputation using a single multivariate Gaussian (p. 44)
MIMG	Multiple imputations using a mixture of multivariate Gaussians (p. 43)
IMG	Conditional mean imputation using a mixture of multivariate Gaussians (p. 43)
IMe	Mean imputation (p. 44)
AIC	Akaike's Information Criterion (p. 37)

Anuran	_____	Anuran calls MFCCs (p. 88)
B.Freys	_____	B. Frey's database of images (p. 65)
BCW	_____	Breast Cancer Wisconsin Diagnostic (p. 66)
BH	_____	Barnes-Hut (p. 60)
BH SNE	_____	Barnes-Hut acceleration of SNE (p. 60)
BH t -SNE	_____	Barnes-Hut t -SNE (p. 60)
BIC	_____	Bayesian Information Criterion (p. 36)
BrTis.	_____	Breast Tissue (p. 42)
CCS	_____	Concrete Compressive Strength (p. 42)
Clusters	Artificial 3-D data set with 3000 dots distributed on eight equal-size Gaussian clusters (p. 65)	
COIL	_____	COIL-20 data set (p. 65)
ConcSl.	_____	Concrete Slump Test (p. 42)
ConV.	_____	Connectionist Bench on Vowel Recognition and Deterding Data (p. 42)
Digits	_____	Optical Recognition of Handwritten Digits (p. 42)
DR	_____	Dimensionality reduction (p. 2)
EE	_____	Energy efficiency (p. 42)
EM	_____	Expectation-Maximization (p. 32)
EVD	_____	Eigenvalue decomposition (p. 35)
Fert.	_____	Fertility (p. 42)
FFir.	_____	Forest Fires (p. 42)
FMMs	_____	Fast multipole methods (p. 58)
FM s t -SNE	_____	Fast multi-scale t -SNE (p. 87)
FM s SNE	_____	Fast multi-scale SNE (p. 87)
Gas	_____	Gas sensors for home activity monitoring (p. 100)
Gesture	_____	Gesture phase segmentation (p. 89)

GIId.	_____	Glass Identification (p. 42)
Hab.	_____	Haberman's Survival (p. 65)
HD	_____	High-dimensional (p. 1)
HDDC	_____	High-dimensional data clustering (p. 35)
HDS	_____	HD space (p. 9)
ILP	_____	Indian Liver Patient (p. 42)
Iono.	_____	Ionosphere (p. 43)
IstSt.	_____	Istanbul Stock Exchange (p. 43)
JSE	_____	Jensen–Shannon embedding (p. 16)
KL	_____	Kullback–Leibler (p. 14)
L-BFGS	_____	Limited-memory Broyden–Fletcher–Goldfarb–Shanno (p. 19)
LD	_____	Low-dimensional (p. 2)
LDS	_____	LD space (p. 9)
MDS	_____	Classical metric multidimensional scaling (p. 4)
ML	_____	Machine learning (p. 1)
MNIST	_____	The test set of the MNIST database containing images of handwritten digits (p. 65)
Ms t -SNE	_____	Multi-scale t -SNE (p. 87)
Ms SNE	_____	Multi-scale SNE (p. 87)
Musk	_____	Musk version 2 (p. 89)
NE	_____	Neighbor embedding (p. 4)
NeRV	_____	Neighborhood Retrieval Visualizer (p. 16)
NLDR	_____	Nonlinear dimensionality reduction (p. 1)
NSTs	_____	Negative sampling techniques (p. 58)
Oliv.	_____	Olivetti database of face images (p. 65)
Park.	_____	Parkinsons (p. 43)

PCA	Principal component analysis (p. 4)
Plant	Combined cycle power plant (p. 88)
PIRel.	Planning Relax (p. 43)
Satellite	Statlog landsat satellite (p. 89)
Sensorless	Sensorless Drive Diagnosis (p. 100)
Sgemm	SGEMM GPU kernel performance (p. 100)
Shuttle	Statlog Shuttle (p. 100)
SNE	Stochastic Neighbor Embedding (p. 9)
SPECTF	SPECTF Heart (p. 66)
Sph	Artificial 3-D data set with 3000 dots distributed on a unit-radius sphere (p. 65)
Sphere	Artificial 3-D data set with 1500 dots distributed on a unit-radius sphere (p. 43)
Theorem	First-order theorem proving (p. 89)
Torus	Artificial 3-D data set with 3000 dots distributed on a torus (p. 65)
UsKn.	User Knowledge Modeling (p. 43)
Waveform	The Waveform database generator version 2 (p. 89)
WholC.	Wholesale customers (p. 43)
Wine	Wine quality (p. 89)
YaHyd.	Yacht Hydrodynamics (p. 43)

Index

- Akaike's Information Criterion (AIC), 37
- Barnes-Hut (BH) t -SNE, 60
- Bayesian Information Criterion (BIC), 36
- Complete data set, 9
- Complete-case k -nearest neighbor imputation, 44
- Conditional mean imputation using a mixture of multivariate Gaussians (IMG), 43
- Conditional mean imputation using a single multivariate Gaussian (IG), 44
- Conditional mixture model sampling, 38
- Crowding problem, 15
- Curse of dimensionality, 2
- Dimensionality reduction (DR), 2
- DR quality assessment, 45
- Early exaggeration, 25
- Expectation-Maximization (EM) algorithm, 32
- Fast multi-scale t -SNE, 84
- Fast multi-scale SNE, 84
- Fast multipole methods (FMMs), 58
- Gaussian mixture model for complete data, 32
- Gaussian mixture model for incomplete data, 32
- High-dimensional data clustering (HDDC) methods, 35
- Holm–Bonferroni correction, 47
- Ignorable missing-data mechanism, 33
- Incomplete data set, 33
- Incomplete-case k -nearest neighbor imputation (INN), 44
- Jensen–Shannon embedding (JSE), 17
- Kullback–Leibler (KL), 14
- LD crushing phenomena, 84
- LD spreading, 84
- Mean imputation (IMe), 44
- Missing at random data, 33
- Missing completely at random data, 33
- Missing values, data or entries, 32
- Multi-scale JSE, 25
- Multi-scale neighbor embedding, 19
- Multi-scale neighbor sets, 81
- Multi-scale NeRV, 25
- Multi-scale optimization, 25
- Multi-scale SNE, 23
- Multiple imputations, 39
- Multiple imputations using a mixture of multivariate Gaussians (MIMG), 43
- Multiple imputations using a single multivariate Gaussian (MIG), 44

- Negative sampling techniques (NSTs),
58
- Neighbor embedding (NE), 4
- Neighborhood preservation criteria, 45
- Neighborhood Retrieval Visualizer (NeRV),
16
- Nonlinear dimensionality reduction (NLDR),
1
- Norm concentration phenomenon, 10
- Octree, 61
- Perplexity, 13
- Perplexity-free, multi-scale t -SNE, 23
- Quadtree, 61
- Random projection trees, 105
- Ranks, 45
- Reverse sweep operator, 34
- Scree test, 36
- Shift invariance, 13
- Single imputation, 29
- Single-scale neighbor embedding, 9
- Stochastic Neighbor Embedding (SNE),
9
- Student t -distributed SNE (t -SNE), 15
- Sweep inversion operator, 34
- Vantage-point tree, 60