UCLouvain

epl

École polytechnique de Louvain

Visualization of data with missing entries using non-linear dimensionality reduction

Author: **Justin Ansotte**
Supervisors: **John Lee, Michel Verleysen**
Readers: **Cyril de Bodt, Pierre Lambert, Edouard Couplet**
Academic year 2022–2023
Master [120] in Computer Science and Engineering

Abstract

This master's thesis studies the field of dimensionality reduction in the context of datasets with missing entries, extending prior work [3] to higher dimensions. Gaussian Mixture Models can be employed to manage missing data by modeling the data distribution, generating pseudo-complete datasets through multiple imputations. In addition, extensions to handle and attend for missing data are introduced. Gaussian Mixture Models are known to scale poorly in high dimension. To address this issue, parametrizations designed to make the model more parsimonious are also implemented. From the imputed datasets, high-dimensional similarities are derived to compute low-dimensional embeddings. At the end of this work, the performance on dimensionality reduction tasks performed with the imputed datasets is assessed on increasing dimensions, then evaluated against alternative imputation techniques. While effective within 50 dimensions, challenges arise in higher dimensional spaces due to the curse of dimensionality and numerical approximations. Other parametrizations from the High Dimensional Data Clustering family [6] show similar performances, suggesting to rely on alternative imputation paradigms when dealing with very high dimensions.

Keywords: Dimensionality Reduction, Missing Data, High-Dimensional Data, Gaussian Mixture Models, Curse of Dimensionality, Data Analysis

Acknowledgments

I want to thank my friends and family for their continuous support, my supervisors Lee John and Verleysen Michel as well as Lambert Pierre, member of the research team. Lastly, I would particularly like to thank de Bodt Cyril for his time, his advice and his brilliant work in [3].

Contents

List of contributions :

- Reviews the state of the art in both the Dimensionality Reduction and Missing Data fields
- Studies the combined methodology for very high dimensions
- Proposes suggestions for further studies

Chapter 1

Introduction

Dimensionality reduction aims to represent high-dimensional data into a lower-dimensional space while preserving its inherent properties. This is valuable in machine learning, as algorithms can struggle with a large number of features, thereby addressing the curse of dimensionality. These techniques are commonly employed to pre-process data too large for some model, or to visualize high-dimensional clusters in a two- or three-dimensional space.

Furthermore, working with data that contains missing entries is a more realistic scenario when applying these methods to real-life data. With the growth of data science fields, the amount of data generated grows exponentially and incomplete databases become more common. To handle these missing values, various techniques exist to retain as much information as possible rather than discarding whole bits of information due to partial gaps.

The work of Cyril de Bodt, Dounia Mulders, Michel Verleysen and John A. Lee [3] studies dimensionality reduction with missing data. To compute the low-dimensional embeddings, it is argued that one would be more interested in obtaining the closest cost function on average rather than the one obtained with the expected dataset. Gaussian Mixture Models are used to model the data distribution in order to extract random samples of the conditional distribution of the missing features. Multiple imputed datasets are created for which the HD similarities are computed using Multi-scale SNE, a non-linear dimensionality reduction technique. The authors tackle the challenge of fitting GMMs in high-dimensional spaces by reducing the relevant dimension of the mixture components with a high-dimensional extension [6] of the classic GMM framework. The motivation behind this approach is the observation that high-dimensional data often resides in different low-dimensional subspaces within the original space.

This paper follows the work of [3] and [6]. It extends on [3] by challenging the feasibility of the method in much higher dimensions. The scope is expanded to nine other HDDC models from [6]. Although these models are labeled as "high-dimensional", in both papers no more than 100 dimensions are tested. By adopting stricter parametrizations, it is questioned whether higher dimensions can be reached, or simply what could be the possible advantages of using those rather than the most general model. This paper is structured as follows: Chapters 2 and 3 review the state of the art in respectively the dimensionality reduction and missing data management fields. Chapter 4 reviews the methodology used. Chapter 5 is dedicated to experimental results on different datasets. Lastly, conclusions are drawn in Chapter 6.

1.1 Notations

Throughout this thesis, please refer to this reference table for abbreviations.

DR	dimensionality reduction
HD	high-dimensional
LD	low-dimensional
HDS	HD Space
LDS	LD Space
PCA	Principal Component Analysis
NLDR	Non-Linear Dimensionality Reduction
SNE	Stochastic Neighbor Embedding
t-SNE	t-distributed SNE
MsSNE	Multi-scale SNE
KNN	K-Nearest Neighbors
GMM	Gaussian Mixture Model(s)
ML	Maximum Likelihood
CF	Closed Form
FG	Full Gaussian
IP	Iterative Procedure

Table 1.1: Abbreviations

Chapter 2

Dimensionality reduction

This chapter proposes an overview of the Dimensionality reduction field. An introduction to the concept of HD data is given in Section 2.1, a brief overview of Feature Selection is proposed in Section 2.2 to introduce relevant concepts and Feature Extraction is explained in Section 2.3, in which we present the appropriate techniques for this work.

2.1 High-dimensional data

Handling low-dimensional (LD) data is typically more straightforward compared to high-dimensional (HD) data. LD data describes scenarios that are more understandable and interpretable. It often allows for a clearer intuition of the data context and how the tools utilizing the data will perform. In the case of HD data, these scenarios lose their concrete representations, making it challenging to form mental images. Intuitions become considerably harder to develop, and comprehending the entire context simultaneously becomes nearly impractical. However, the most significant shift occurs in the ratio between the number of observations and the dimensionality. Working with HD data necessitates exponentially more data points.

The goal of Dimensionality Reduction (DR) is to summarize a large set of features into a smaller one. This transformation should be without redundancy and represent the original information contained in the set as well as it can. High-dimensional (HD) data is encountered in many fields [23] : image processing, multivariate data analysis, data mining, ... One could want to understand and classify existing data, performing unsupervised machine learning tasks. Or infer and generalize new data, supervised machine learning in this case. However, when the dimensionality grows, the properties of the easily visualizable 2D and 3D Euclidean spaces start behaving weirdly and curious events arise. This is referred to as the *curse of dimensionality*, or the *empty space phenomenon*.

Curse of dimensionality Coined by Richard E. Bellman [18], the curse of dimensionality refers to various phenomena when working with high-dimensional spaces (HDS). It was observed that without making simplifying assumptions, the amount of data samples needed to accurately estimate a function involving multiple variables on a specific domain increases exponentially as the number of dimensions increases. Think of n points on a 1-dimensional grid, for example. With increasing dimensions d, the number of points evolve with n^d.

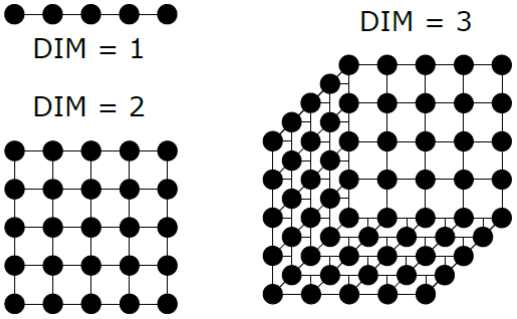

Figure 2.1: d-dimensional grid

Example : hyper-volume of cubes and spheres This example is taken from the course [LELEC2870 - Machine learning : regression, deep networks and dimensionality reduction] given by John Lee and Michel Verleysen, which is extracted from their work on DR [23].
In a d-dimensional space, a sphere and the corresponding circumscribed cube (all edges equal the sphere diameter) lead to the following volume formulas :

$$\begin{aligned} V_{\text{sphere}}(r) &= \frac{2\pi^{d/2} r^d}{d\Gamma(d/2)}, \\ V_{\text{cube}}(r) &= (2r)^d, \end{aligned} \tag{2.1}$$

where r is the radius of the sphere and Γ is the gamma function s.t. $\Gamma(n) = (n-1)!$ with $n > 0$. Surprisingly, the ratio V_{sphere}/V_{cube} tends to zero when d increases :

$$\lim_{d\to\infty} \frac{V_{\text{sphere}}(r)}{V_{\text{cube}}(r)} = \frac{\pi^{d/2}}{d2^{d-1}\Gamma(d/2)} = 0 \tag{2.2}$$

In intuitive terms, as the dimensionality increases, the hypercube has more and more corners. Consequently, the hypersphere becomes smaller and smaller, resulting in a reduction of volume. Eventually, the shape starts resembling a heap of spikes. To illustrate this concept, let us assign the value of $1/2$ to r, V_{cube} tends to 1, whereas V_{sphere} approaches 0. As the dimensionality increases, the volume of the sphere vanishes (see Figure 2.2).

Blessing of dimensionality The reminder of this chapter presents techniques developed to tackle the curse of dimensionality, "fighting against" the problems it causes. However, as opposed to the curse of dimensionality, and this is mentioned only in the interest of providing a global view of what working with HD data means, another term has been introduced in the late 90s : the *blessing of dimensionality*. The regularity of having many "identical" dimensions over which one can "average" is a fundamental tool [11]. HD data, like a two-sided coin, has its own advantages and challenges. As a rapidly growing field of research, it deserves significant attention and exploration for the future. In the work of [11], various directions for progress in this field were already suggested, providing valuable insights and potential avenues for further advancements.

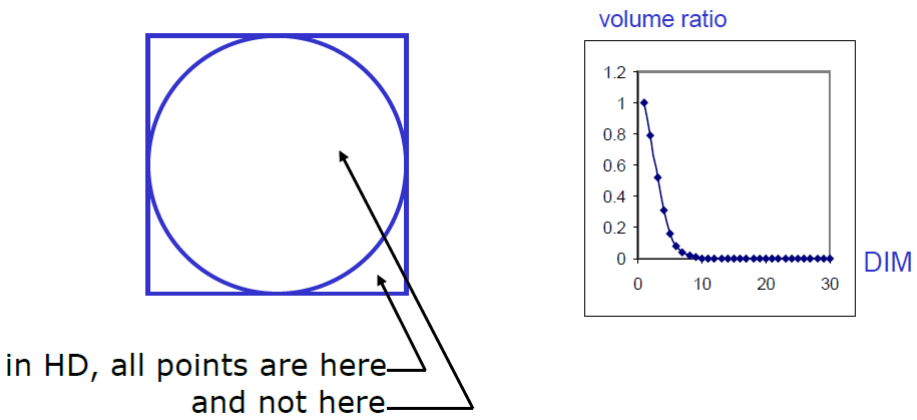

Figure 2.2: As the dimension grows, the volume of the sphere vanishes

2.2 Feature selection

Feature selection techniques reduce the dimensionality by selecting a subset of relevant features. The optimal feature subset is a subset of all relevant features. Relevancy can be defined as [21] :

Definition 1 *A feature $x_i \in X$ is relevant to a target concept C if there exists a pair of examples A and B in the instance space such that A and B differ only in their assignment to x_i and $C(A) \neq C(B)$.*

In the context of classification or prediction tasks, the relevancy of a feature x_i is a measure of how useful it is to predict the target feature y. Several metrics and statistical tests can be used to select the subset of relevant features.

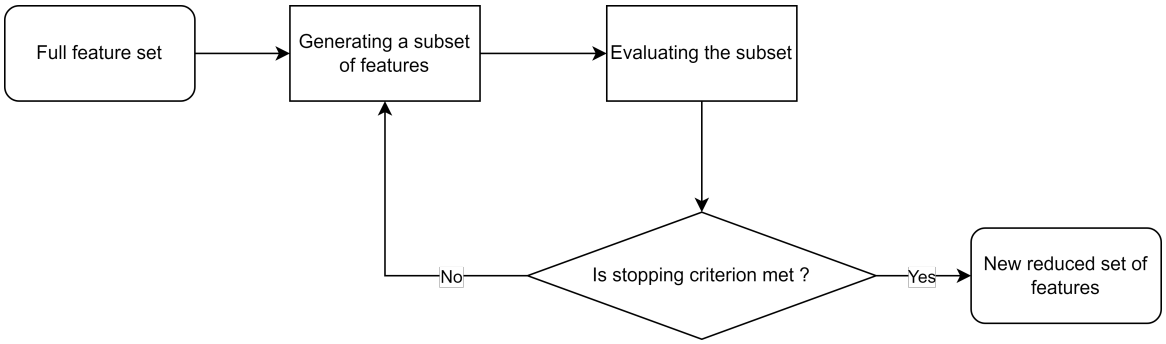

Figure 2.3: General process of feature selection

Definition 2 *The correlation between random variable x and random variable y is defined as*

$$\rho_{xy} = \frac{E[(x-E[x]) \cdot (y-E[y])]}{\sqrt{E\left[(x-E[x])^2\right] \cdot E\left[(y-E[y])^2\right]}}$$

This correlation measures linear dependencies and is always in the range $[-1; +1]$ with 0 indicating no linear relation. The logic behind using correlation for feature selection is that

good variables correlate highly (positively or negatively) with the target feature and variables should be uncorrelated among themselves as to avoid redundancy. Visualizing correlations in a heat map is a well-known method in feature selection. Be aware of two things :

- Correlation does not measure nonlinear relations → low correlation does not mean absence of relationship.
- High correlation does not mean causality.

Users are typically expected to provide their expertise and domain knowledge when dealing with correlation in their dataset. Moreover, correlation is challenging to define when there are more than two variables, and the number of correlations increases exponentially with the number of features. Consequently, correlation-based selection methods often suffer from scalability issues in high-dimensional (HD) datasets.

Entropy and mutual information

Definition 3 *The entropy of a random variable is a measure on its uncertainty s.t.*

$$\begin{aligned} H(y) &= -E[\log(P[y])] \\ &= -\sum_{y\in\Omega} \log(P[y])P[y] \quad \textit{when } Y \textit{ is discrete} \\ &= -\int \log(P[y])\mathrm{d}y \quad \textit{when } Y \textit{ is continuous} \end{aligned}$$

Conditional entropy $H(y|x)$ measures the uncertainty on y when x is known s.t. $H(y|x) = H(y,x) - H(x)$. Finally, mutual information is defined as :

Definition 4 *Mutual information is the difference in entropy between y and x when x is known s.t.*

$$I(y;x) = H(y) - H(y|x) = H(x) - H(x|y)$$

The advantage mutual information has over correlation is that it can identify non-linear relationships between variables. If X is a subset of features, its relevance may still be evaluated. But estimating $I(y;x)$ is difficult in HD because of the curse of dimensionality. In general, all estimators suffer from it.

2.3 Feature extraction

Figure 2.4: General process of feature extraction

Feature extraction methods build new features from the original dataset. It constructs new variables carrying a large part of the global information while reducing the dimensionality of the data. Many techniques have been developed since the early 20^{th} century which can be classified, amongst other things, as being either linear or non-linear. A very complete overview of the field and existing techniques is given in [23].

2.3.1 Linear Dimensionality Reduction

Principal Component Analysis (PCA)

Possibly the most well-known and easy-to-apply feature extraction technique, (PCA)[29] aims to transform the original features into a new set of uncorrelated variables called principal components. These principal components are ranked by variance, where the first component accounts for the highest variance, the second component for the second highest, and so on. The idea is to project the HD vectors by maximizing the variance or dot product preservation. It can be shown that the principal components are eigenvectors of the data's covariance matrix, hence they are often computed using SVD.

Algorithm 1 Principal Component Analysis (PCA)

Require: Data matrix $\mathbf{X}$ with n samples and p features
Ensure: Principal components $\mathbf{V}$ and projected data $\mathbf{Z}$
1: Center the data : $\mathbf{X} \leftarrow \mathbf{X} - \frac{1}{n}\mathbf{X}\mathbf{1}_n^T$
2: Compute the covariance matrix : $\mathbf{\Sigma} \leftarrow \frac{1}{n-1}\mathbf{X}^T\mathbf{X}$
3: Compute the eigenvalues λ_i and eigenvectors $\mathbf{v}_i$ of $\mathbf{\Sigma}$
4: Sort the eigenvectors based on the descending order of eigenvalues
5: Select the first k eigenvectors corresponding to the k largest eigenvalues
6: Form the matrix $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k]$
7: Project the data onto the new space : $\mathbf{Z} \leftarrow \mathbf{X}\mathbf{V}$

There is, however, no assurance that a small number of principal components with the highest variance will necessarily contain the essential information needed for the task it was used for. In some cases, the crucial information might be embedded in linear combinations of variables with low variance rather than high. PCA is built upon assumptions that are overly limiting, particularly regarding the separation of latent variables. Even if the goal is solely dimensionality reduction, a remaining and somewhat troublesome assumption is that the relationships between observed variables are either linear or close to linear [23].

2.3.2 Non-Linear Dimensionality Reduction (NLDR)

The motivation behind using non-linear techniques for dimensionality reduction is rooted in the nature of complex and high-dimensional data. Large datasets often contain hidden information that requires transformation to be effectively understood by humans. Remember that one of the uses of DR is visualizing HD data in a LD space. Non-linear techniques allows to unfold the manifold instead of merely projecting it in a linear fashion. This preservation of distances along the manifold helps in capturing the intricate relationships between data points and enables a more faithful representation of the underlying data structure. Many techniques exist such as distance-based, spectral methods (ISOMAP [35],[36]; CCA [10]; CDA [25],[24], LLE [31],[33]), Self-Organizing Maps [42], or Auto-encoders [43] to name a few. There is one family in particular which is of high interest in this work as it is the chosen NLDR methodology in [3] : *similarity-based embedding.*

Similarity-based embedding

SNE : Rather than trying to preserve distances from HD to LD, this family of methods tries to preserve the HD neighborhoods in LD. In other words, if points are 'close' to each other in HD, they should appear close to each other in LD. Stochastic Neighbor Embedding (SNE) [19] starts by converting the HD Euclidean distances between data points into conditional probabilities, represented by similarities. The similarity of the example ξ_j to ξ_i is the conditional probability σ_{ij} that ξ_i would pick ξ_j as its neighbor if neighbors were picked in proportion to their probability density under a Gaussian centered at ξ_i. This probability is given by :

$$\sigma_{ij} = \frac{\exp\left(-\delta_{ij}^2 / \left(2\lambda_i^2\right)\right)}{\sum_{k,k\neq i} \exp\left(-\delta_{ik}^2 / \left(2\lambda_i^2\right)\right)} \tag{2.3}$$

where λ_i is the variance of the Gaussian that is centered on data point ξ_i and $\delta_{ij} = \left\|\boldsymbol{\xi}_i - \boldsymbol{\xi}_j\right\|_2$ is the HD Euclidean distance. Once an expression for the similarity in HD is derived, an expression for the conditional probability between two points in LD must also be defined. This is done by means of soft neighborhood in the LD space, which is given by :

$$s_{ij} = \frac{\exp\left(-d_{ij}^2/2\right)}{\sum_{k,k\neq i} \exp\left(-d_{ik}^2/2\right)} \tag{2.4}$$

where $d_{ij} = \left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2$ is the LD Euclidean distance. The motivation behind using similarities is that the conditional probabilities in HD and LD should be equal if the map points x_i and x_j correctly model the similarity between ξ_i and ξ_j. The Kullback-Leibler divergence is used to measure the faithfulness with which s_{ij} models σ_{ij}. A gradient descent method is used to minimize the sum of KL divergences expressed as :

$$D_{\mathrm{KL}}\left(\boldsymbol{\sigma}_i \| \mathbf{s}_i\right) = \sum_{j=1}^{N} \sigma_{ij} \log\left(\sigma_{ij}/s_{ij}\right) \tag{2.5}$$

A very good mental representation of this minimization is given in [41] : "*Physically, the gradient may be interpreted as the resultant force created by a set of springs between the map point*

x_i and all other map points x_j. All springs exert a force along the direction $(x_i - x_j)$. The spring between x_i and x_j repels or attracts the map points depending on whether the distance between the two in the map is too small or too large to represent the similarities between the two high-dimensional data points."

Algorithm 2 Stochastic Neighbor Embedding (SNE)

1: Choose size K of neighborhoods in HDS
2: Convert hard neighborhoods into soft ones $\leftarrow \sigma_{ij}$
3: Adjust all bandwidths : $\log(K) = -\sum_{j=1}^{N} \sigma_{ij} \log \sigma_{ij}$
4: Define soft neighborhoods in LDS $\leftarrow s_{ij}$
5: Minimize KL divergences $\leftarrow D_{\mathrm{KL}}\left(\boldsymbol{\sigma}_i \| \mathbf{s}_i\right)$

t-SNE : The cost function of SNE is notably difficult to optimize. It also faces a problem referred to as the "*crowding problem*". Indeed, it arises due to the limited space available on the two-dimensional map to accommodate moderately distant data points, in stark contrast to the relatively larger area available to accommodate nearby data points. Continuing on the spring analogy : "*The spring connecting data point i to each of these too-distant map points will thus exert a very small attractive force. Although these attractive forces are very small, the very large number of such forces crushes together the points in the center of the map, which prevents gaps from forming between the natural clusters.*" An other technique, t-SNE [41], changes the similarities in LD, using a Student t-distribution with one degree of freedom as the heavy-tailed distribution rather than a Gaussian. The joint probabilities s_{ij} are defined as

$$s_{ij} = \frac{\left(1 + d_{ij}^2\right)^{-1}}{\sum_{k,l,k \neq l} \left(1 + d_{kl}^2\right)^{-1}} \tag{2.6}$$

This has several benefits :

- The map's representation of joint probabilities are (almost) invariant to changes in the scale of the map for map points that are far apart.
- Large clusters of points that are far apart interact in the same way as individual points.
- Evaluating a point under a t-Student distribution is computationally faster than under a Gaussian.
- The optimization of the t-SNE cost function is much easier than the optimization of the cost function of SNE.
- Focusing on the local structure of the data will tend to extract clustered local groups of samples, making it particularly beneficial for visualizing data in very entangled datasets.

However, t-SNE still suffers from a quadratic computational and memory complexity with the number of data points, limiting its applicability to datasets with a large number of data points. This limitation presents a challenge for this research thesis, which aims to explore techniques in very high-dimensional spaces where an exponential number of data points is needed. Another weakness of t-SNE lies in its relatively local nature, making it susceptible to the curse of the intrinsic dimensionality of the data.

Definition 5 *The intrinsic dimensionality of a random vector y is the minimal number of parameters or latent variables needed to describe y [23].*

In datasets with high intrinsic dimensionality and a manifold that exhibits significant variation, the local linearity assumption implicit in t-SNE may be transgressed. Consequently, t-SNE might be less effective on datasets with very high intrinsic dimensionality. To address this limitation, [41] suggests using an Auto-encoder as the model generation technique. Auto-encoders can better identify highly varying manifolds compared to a local method such as t-SNE, potentially requiring fewer data points to learn an appropriate solution. A study of the performance of Auto-encoders is proposed in [43].

Multi-scale SNE This is the method selected in [3] for the NLDR which this work extends. SNE, as mentioned earlier, relies on local transformations of high-dimensional distances, which limits the consideration of average- and large-scale interactions when defining the global structure of the embedding. To overcome this, an additional scale index h is introduced, which redefines the similarities to account for various data scales simultaneously. This parameter-free multi-scale approach preserves the neighborhoods in a broader range of sizes and retains local as well as global structures in the data [22].
The precision π_i, determined through binary search, is introduced to achieve a user-defined perplexity K_*[1] for the distribution $\sigma_i = \sigma_{ij}, j \in \mathcal{I}\backslash i$. By combining multiple SNE similarities with increasing perplexities, both local and global relationships can be captured effectively. The new multi-scale similarities are defined as follows :

$$\begin{aligned} \sigma_{hij} &= \frac{\exp\left(-\pi_{hi}\delta_{ij}^2/2\right)}{\sum_{k\in\mathcal{I}\backslash\{i\}}\exp\left(-\pi_{hi}\delta_{ik}^2/2\right)}, \sigma_{hii} = 0, \\ s_{hij} &= \frac{\exp\left(-p_{hi}d_{ij}^2/2\right)}{\sum_{k\in\mathcal{I}\backslash\{i\}}\exp\left(-p_{hi}d_{ik}^2/2\right)}, s_{hii} = 0. \end{aligned} \tag{2.7}$$

Multi-scale similarities are then expressed as :

$$\sigma_{ij,ms} = \frac{1}{L}\sum_{h=1}^{L}\sigma_{hij}, s_{ij,ms} = \frac{1}{L}\sum_{h=1}^{L}s_{hij} \tag{2.8}$$

with L the number of scales. The cost function of MsSNE is given by :

$$C_{MsSNE} = -\sum_{i\in\mathcal{I},j\in\mathcal{I}\backslash\{i\}}\sigma_{ij,ms}\log s_{ij,ms} \tag{2.9}$$

The use of multi-scale optimization is recommended to tackle the challenge of optimizing cost functions in SNE-like methods.

[1] K_* controls the balance between preserving local and global structures in the data's low-dimensional representation. Perplexity is a smooth measure of the effective number of neighbors. A lower perplexity value encourages the algorithm to focus on preserving local structures, while a higher value encourages more emphasis on global structures.

2.3.3 Quality assessment

A Receiver Operating Characteristic (ROC) Curve is a function used to assess the performance of a classification model. It plots as coordinates of the True Positive Rate (TPR) and False Positive Rate (FPR), respectively defined as $TPR = \frac{TP}{TP+FN}$ and $FPR = \frac{FP}{FP+TN}$, both ranging from $[0; 1]$. The Area Under the Curve (AUC) is defined as the integral of the ROC Curve, which provides an aggregate of performance across all possible classification thresholds. It also ranges from $[0; 1]$ and is to be interpreted as the percentage of correctness of the assessed model.

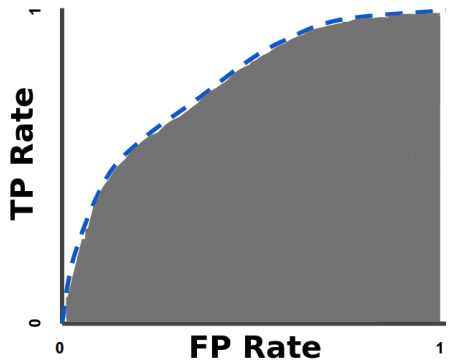

Figure 2.5: AUC (Area under the ROC Curve)

Criteria measuring the HD neighborhood preservation in the LDS have been studied to assess the DR quality. From [3], it is possible to define an AUC for MsSNE. The HD and LD ranks of ξ_j and x_j with respect to ξ_i and x_i are defined as :

$$\begin{aligned} \rho_{ij} =& \left| \left\{ k \in \mathcal{I} \mid \delta_{ik} < \delta_{ij} \text{ or } \left(\delta_{ik} = \delta_{ij} \text{ and } k < j \right) \right\} \right| \\ r_{ij} =& \left| \left\{ k \in \mathcal{I} \mid d_{ik} < d_{ij} \text{ or } \left(d_{ik} = d_{ij} \text{ and } k < j \right) \right\} \right| . \end{aligned} \tag{2.10}$$

The K neighborhoods of $\boldsymbol{\xi}_i$ and $\mathbf{x}_i$ can be denoted as $\nu_i^K = \left\{ j \in \mathcal{I} \mid 1 \leq \rho_{ij} \leq K \right\}$ and $n_i^K = \left\{ j \in \mathcal{I} \mid 1 \leq r_{ij} \leq K \right\}$ for the HDS and LDS respectively. The average normalized agreement between these neighborhoods is computed as :

$$Q_{NX}(K) = \sum_{i \in \mathcal{I}} \frac{\left| \nu_i^K \cap n_i^K \right|}{KN} \in [0, 1] \tag{2.11}$$

This is a measure that quantifies how many of the K-Nearest Neighbors (KNN) in HD are also among the KNN in LD, and vice versa. In other words, it provides an assessment of how well the neighborhoods are preserved during the DR process. $Q_{NX}(K)$ can be rescaled to enable the comparison of different neighborhood sizes as $\mathbb{E}\left[Q_{NX}(K)\right]$ equals $K/(N-1)$ for random LD points, which allows the definition of the curves :

$$R_{NX}(K) = \frac{(N-1)Q_{NX}(K) - K}{N - 1 - K} \tag{2.12}$$

These are usually presented using a logarithmic scale for K since local neighborhoods are often more prominent. The resulting AUC increases with the DR quality :

$$AUC = \left(\sum_{K=1}^{N-2} K^{-1} \right)^{-1} \cdot \sum_{K=1}^{N-2} R_{NX}(K)/K \in [-1, 1] \tag{2.13}$$

Chapter 3

Missing data management

This chapter covers the notion of missing data in the Machine Learning field. A brief explanation to the nature of the missingness is given in Section 3.1, relevant notations are introduced in Section 3.2, the concept of multiple imputations is put in comparison to single imputation in Section 3.3 and the family of Gaussian Mixture Models (GMMs) is introduced in Section 3.4 for which we derive an extension to handle missing values [12]. Finally, a high-dimensional extension of GMMs, the HDDC models [6], are presented in Section 3.5.

3.1 Missing data mechanisms

Missing data can arise for various reasons, and in some cases the reasons for the missingness are unknown. It may come from malfunctioning hardware, data entry errors which had to be removed or corrupted data in general. When handling missing data, it is important to first consider the mechanism behind the missingness of the information, as to make valid statistical hypotheses later. Three types of missing data mechanisms can be defined [32] :

- Missing completely at random (MCAR)
- Missing at random (MAR)
- Missing not at random (MNAR)

MNAR is quite self-explanatory, knowing the mechanism behind the non-randomness of the missing values is a powerful information to integrate in the analysis. MCAR describes randomly distributed data across the variable which are unrelated to the other variables. The probability of a data to go missing is independent of any observation in the data set. MAR implies that the missing data are not randomly distributed but they are accounted for by other observed variables. Given the set of observed features Ξ^O, the probability of a data entry to be missing is independent from the sample ξ^{M_i} for all $i \in \mathcal{I}$.

The missing-data mechanism can be shown to be ignorable [16], meaning that maximum likelihood estimators remain consistent when evaluated on the incomplete database. Still, this work makes the assumption of MAR data.

3.2 Notations

Taking the notation of [3], let $\boldsymbol{\Xi} = \{\boldsymbol{\xi}_i, O_i\}_{i=1}^{N}$ be the incomplete data set with N independent samples $\boldsymbol{\xi}_i \in \mathbb{R}^H, O_i \subset \{1, \ldots, H\}$ indexing the observed features of $\boldsymbol{\xi}_i$. The complement of O_i is noted M_i, and $\boldsymbol{\xi}_i^{O_i}$ (resp. $\boldsymbol{\xi}_i^{M_i}$) denotes the values of the observed (respectively missing) features of $\boldsymbol{\xi}_i$. The set $\boldsymbol{\Xi}^O$ gathers $\boldsymbol{\xi}_i^{O_i}$ for all $i \in \mathcal{I}$.

3.3 Multiple Imputation

Multiple Imputation (MI) [26] refers to the procedure of replacing each missing value by a vector containing at least 2 imputed values. Replacing $\xi_i^{M_i}$ by the first component in its vector of imputations creates the first complete data set, the second creates the second complete data set and so on.

Single Imputation	Multiple Imputation
Advantages	
Allows complete-data analysis	Same
Advantages for cases where collection and analysis are performed separately (i.e. confidentiality constraints)	Same but also reflects the uncertainty as to which values to impute.
Missing data problem handled once	Same
Disadvantages	
Treats the missing value as known once imputed	Solved, can reflect sampling variability
Uncertainty often underestimated	Solved Takes more effort

Table 3.1: Single Imputation (SI) and Multiple Imputation (MI)

3.4 Gaussian Mixture Models

A Gaussian Mixture Model (GMM) is a statistical model that assumes that a set of data points is generated from a combination of a limited number of Gaussian distributions with unknown parameters. In the GMM framework, each class is represented by a Gaussian probability density function defined as follows :

$$f(x, \theta) = \sum_{k=1}^{G} w_k \phi(x, \theta_k) \tag{3.1}$$

Here ϕ represents a p-variate normal density with parameter $\theta_k = \{\mu_k, \sum_k\}$, the mean and covariance respectively. The mixing proportions, w_k, indicate the contribution of each component of the mixture. A G-component Gaussian mixture model of a complete dataset $\boldsymbol{\Xi}_c := \{\boldsymbol{\xi}_i\}_{i=1}^{N}$ consists of N independent examples $\boldsymbol{\xi}_i \in \mathbb{R}^H$. This model is composed of G multivariate Gaussian,

each with mean vectors $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_G \in \mathbb{R}^H$, covariance matrices $\boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_G \in \mathbb{R}^{H \times H}$, and positive, unit-sum weight coefficients $w_1, \ldots, w_G \in \mathbb{R}$ which maximize the model log-likelihood

$$\log p\left(\boldsymbol{\Xi}_c \mid \theta\right) = \sum_{i \in \mathcal{I}} \log \left(\sum_{k \in \mathcal{G}} w_k \mathcal{N}\left(\boldsymbol{\xi}_i \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right) \right) \tag{3.2}$$

where $\mathcal{I} := \{1, \ldots, N\}, \quad \mathcal{G} := \{1, \ldots, G\}, \quad \theta := \{w_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k; k \in \mathcal{G}\}$ and where $\mathcal{N}\left(\boldsymbol{\xi}_i \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)$ represents the probability density function evaluation of the k^{th} Gaussian over $\boldsymbol{\xi}_i$. As a reminder, the likelihood is defined as [26] :

Definition 6 *The likelihood function $L_Y(\boldsymbol{\Xi}_c \mid \theta)$ is any function of the parameter θ proportional to the the density function $f_Y(\boldsymbol{\Xi}_c \mid \theta)$.*

To maximize equation (3.2), a traditional Expectation-Maximization (EM) algorithm is employed. The algorithm involves the following steps :

1. Initialization : Start by initializing the mean vectors μ_k, covariance matrices Σ_k, and weight coefficients w_k.

2. E-step : Calculate the probabilities t_{ik} for each example to originate from each Gaussian, given the current value of θ. This is done for each $i \in \mathcal{I}$ and $k \in \mathcal{G}$.

$$t_{ik} = \frac{w_k \mathcal{N}\left(\boldsymbol{\xi}_i \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)}{\sum_{j \in \mathcal{G}} w_j \mathcal{N}\left(\boldsymbol{\xi}_i \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\right)}, \quad N_k = \sum_{i \in \mathcal{I}} t_{ik} \tag{3.3}$$

3. M-step : Update θ given the following probabilities :

$$\begin{aligned} \boldsymbol{\mu}_k &= \frac{1}{N_k} \sum_{i \in \mathcal{I}} t_{ik} \boldsymbol{\xi}_i, \quad w_k = \frac{N_k}{N}, \\ \boldsymbol{\Sigma}_k &= \frac{1}{N_k} \sum_{i \in \mathcal{I}} t_{ik} \left(\boldsymbol{\xi}_i - \boldsymbol{\mu}_k\right) \left(\boldsymbol{\xi}_i - \boldsymbol{\mu}_k\right)^T. \end{aligned} \tag{3.4}$$

4. Repeat step 2 & 3 until convergence of (3.2).

3.4.1 GMM to handle missing data

The standard EM algorithm for fitting Gaussian mixture models has been extended to handle data with missing values [12]. A Gaussian mixture model for $\boldsymbol{\Xi}$ then maximizes the observed log-likelihood

$$\log p\left(\boldsymbol{\Xi}^O \mid \theta\right) = \sum_{i \in \mathcal{I}} \log \left(\sum_{k \in \mathcal{G}} w_k \mathcal{N}\left(\boldsymbol{\xi}_i^{O_i} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right) \right) \tag{3.5}$$

over its parameters θ, where $\mathcal{N}\left(\boldsymbol{\xi}_i^{O_i} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)$ refers to the marginal of $\mathcal{N}\left(\boldsymbol{\xi}_i \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)$ over the observed features of $\boldsymbol{\xi}_i$. To handle for the observed and missing data, it is possible to adapt the EM-algorithm to maximize the new log-likelihood (3.5) :

1. Initialization :
   - μ_k are initialized by randomly selecting values from the observed data, samples without missing values if available. Otherwise, Gaussian means using incomplete samples are initialized to impute the missing values with the sample mean.
   - Σ_k are initialized either with the sample covariance of the data (ignoring samples with missing values) or as diagonal matrices, using only the sample variance of each variable.
   - w_k are initialized using uniform values.
2. E-Step[1] : The conditional expectations $\tilde{\boldsymbol{\mu}}_{k,M_i|O_i} \in \mathbb{R}^{|M_i|}$ and covariances $\tilde{\boldsymbol{\Sigma}}_{k,M_i|O_i} \in \mathbb{R}^{|M_i|\times|M_i|}$ of the missing features of each sample $\boldsymbol{\xi}_i$ given $\boldsymbol{\xi}_i^{O_i}$ under each Gaussian component $k \in \mathcal{G}$ are computed as :

$$\begin{aligned} t_{ik} &= \frac{w_k \mathcal{N}\left(\boldsymbol{\xi}_i^{O_i} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)}{\sum_{j\in\mathcal{G}} w_j \mathcal{N}\left(\boldsymbol{\xi}_i^{O_i} \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\right)}, \quad N_k = \sum_{i\in\mathcal{I}} t_{ik} \\ \tilde{\boldsymbol{\mu}}_{k,M_i|O_i} &= \boldsymbol{\mu}_k^{M_i} + \boldsymbol{\Sigma}_k^{MO_i}\left(\boldsymbol{\Sigma}_k^{OO_i}\right)^{-1}\left(\boldsymbol{\xi}_i^{O_i} - \boldsymbol{\mu}_k^{O_i}\right) \\ \tilde{\boldsymbol{\Sigma}}_{k,M_i|O_i} &= \boldsymbol{\Sigma}_k^{MM_i} - \boldsymbol{\Sigma}_k^{MO_i}\left(\boldsymbol{\Sigma}_k^{OO_i}\right)^{-1}\boldsymbol{\Sigma}_k^{OM_i} \end{aligned} \tag{3.6}$$

   where $\boldsymbol{\mu}_k^{M_i}$ and $\boldsymbol{\mu}_k^{O_i}$ denote the components of $\boldsymbol{\mu}_k$ indexed by M_i and O_i [2].
3. M-step : Similar as (3.4), except that the component means are estimated using the imputed data vectors and the covariance matrix estimate includes an extra term to account for covariances related to the imputed values. With the feature $\tilde{\boldsymbol{\xi}}_{ik} \in \mathbb{R}^H$ with components indexed by O_i and M_i equal to $\boldsymbol{\xi}_i^{O_i}$ and $\tilde{\boldsymbol{\mu}}_{k,M_i|O_i}$, we compute for $k \in \mathcal{G}$:

$$\begin{aligned} \boldsymbol{\mu}_k &= \frac{1}{N_k}\sum_{i\in\mathcal{I}} t_{ik}\tilde{\boldsymbol{\xi}}_{ik}, \quad w_k = \frac{N_k}{N} \\ \boldsymbol{\Sigma}_k &= \frac{1}{N_k}\sum_{i\in\mathcal{I}} t_{ik}\left(\tilde{\boldsymbol{\xi}}_{ik} - \boldsymbol{\mu}_k\right)\left(\tilde{\boldsymbol{\xi}}_{ik} - \boldsymbol{\mu}_k\right)^T + \tilde{\boldsymbol{\Sigma}}_{ik} \end{aligned} \tag{3.7}$$

   where $\tilde{\boldsymbol{\Sigma}}_{ik} \in \mathbb{R}^{H\times H}$ is defined as $\tilde{\boldsymbol{\Sigma}}_{ik}^{MM_i} = t_{ik}\tilde{\boldsymbol{\Sigma}}_{k,M_i|O_i}$ and zero elsewhere, $\tilde{\Sigma}_{ik}^{MM_i}$ denoting the sub-matrix of $\tilde{\Sigma}_{ik}$ with rows and columns indexed by M_i.
4. Repeat step 2&3 until convergence of (3.5)

One drawback of this model is that it requires estimating full covariance matrices, resulting in an increase in the number of parameters with the square of the data dimension. This makes it often not even possible to apply a conventional Gaussian Mixture Model. However, considering the empty space phenomenon (2.1), it can be assumed that HD data tends to reside around subspaces with lower dimension compared to the one of the original space. Following that intuition, low-dimensional class-specific subspaces are introduced, which reduce the number of parameters that need to be estimated. This is the High-Dimensional extension of GMMs as proposed in [6] and inspired from the attempt at making GMM more parsimonious in [8].

[1] It is worth noting that, as in [3], [12] the sweep inversion operator can be used to accelerate matrix inversions in the E-step. For further informations, refer to [3], [12].

[2] $\Sigma_k^{AB_i}$ refers to the sub-matrix of Σ_k with rows indexed by A_i and columns by B_i.

3.5 High-Dimensional extension of GMM : HDDC

This section presents a high-dimensional extension for GMMs. In an attempt at making the models more parsimonious, [6] declines several parametrizations. The intuition as well as the general model is presented in Section 3.5.1, then some parametrizations are presented in Section 3.5.2. The properties of these models' parameters are given in Section 3.5.3.

3.5.1 The Gaussian model $[a_{kj}b_kQ_kd_k]$

As in the classical GMM framework, class conditional densities are assumed to be Gaussian. Let Q_k be the orthogonal matrix with the eigenvectors of $\sum_k$ as columns. The class conditional covariance matrix Δ_k is therefore defined in the eigenspace of $\sum_k$, in other words, its eigenvalue decomposition, by :

$$\Delta_k = Q_k^t \Sigma_k Q_k \tag{3.8}$$

From this decomposition, modifications on the eigenvalues can be applied in order to reduce the intrinsic dimension of the feature space and more. These are fully detailed in the next section. The covariance matrix is a diagonal matrix which contains the eigenvalues of $\sum_k$. It is further assumed that Δ_k is divided into two blocks :

$$\Delta_k = \left(\begin{array}{ccc|ccc} a_{k1} & 0 & \dots & & & \\ 0 & \ddots & & & 0 & \\ \vdots & & a_{kd_k} & & & \\ \hline & & & b_k & & \vdots \\ & 0 & & & \ddots & 0 \\ & & & \dots & 0 & b_k \end{array}\right) \begin{array}{l} \left.\begin{array}{l}\\\\\\\end{array}\right\} d_k \\ \left.\begin{array}{l}\\\\\\\end{array}\right\} (H-d_k) \end{array} \tag{3.9}$$

with $a_{kj} > b_k, j = 1, \dots, d_k$, and where $d_k \in \{1, \dots, H\}$ is unknown.

The class-specific subspace $\mathbb{E}_k$ is defined as the affine space spanned by the d_k eigenvectors associated with the eigenvalues a_{kj}, s.t. $\mu_k \in \mathbb{E}_k$ and $\mathbb{E}_k \oplus \mathbb{E}_k^{\perp} = \mathbb{R}^p$. Within the subspace $\mathbb{E}_k^{\perp}$, the variance is modeled by a single parameter b_k. The projections of x on $\mathbb{E}_k$ and $\mathbb{E}_k^{\perp}$ are

$$\begin{aligned} P_k(x) &= \tilde{Q}_k \tilde{Q}_k^t (x - \mu_k) + \mu_k \\ P_k^{\perp}(x) &= \bar{Q}_k \bar{Q}_k^t (x - \mu_k) + \mu_k \end{aligned} \tag{3.10}$$

where $\tilde{Q}_k$ consists of the d_k first columns of Q_k supplemented by $(H - d_k)$ zero columns, and $\bar{Q}_k = \left(Q_k - \tilde{Q}_k\right)$. The dimension d_k of the subspace $\mathbb{E}_k$, referred to as the specific subspace of the k-th group, can be considered as its intrinsic dimension. It represents the number of dimensions required to describe the main features of this group. From [8], a notation for the reference model is derived as $[a_{kj}b_kQ_kd_k]$. This is the model used in [3],[12]. HDDC stands for "High Dimensional Data Clustering".

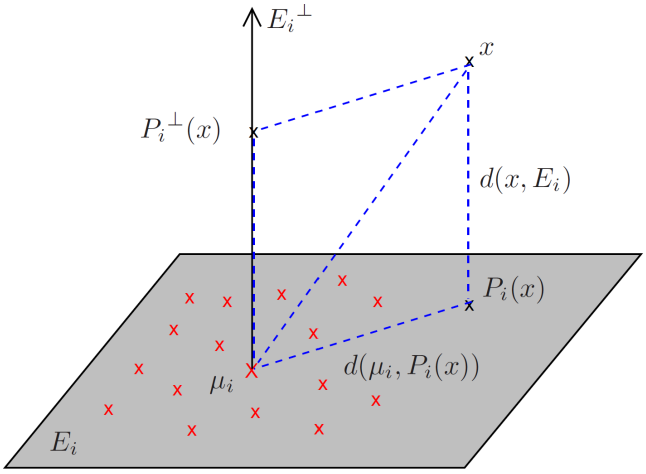

Figure 3.1: The subspaces $\mathbb{E}_k$ and $\mathbb{E}_k^\perp$ of the k-th mixture component.
From [6] where they use i instead of k to enumerate the Gaussian components.

3.5.2 The different parametrizations

By fixing some parameters to be common within or between classes, particular models which correspond to different regularizations are derived. The demonstrations showing how to obtain the estimators for each model are directly taken from [6]. Let us take a step back, to the classical GMM framework, for which it is possible [15] to write the log-likelihood (3.2) as :

$$-2\log(L) = \sum_{k=1}^{G} N_k \sum_{j=1}^{H} \left(\log\left(\delta_{kj}\right) + \frac{1}{\delta_{kj}} q_{kj}^t \Sigma_k q_{kj} \right) + c^{st} \tag{3.11}$$

where δ_{kj} is the j-th diagonal coefficient of Δ_k and q_{kj} is the j-th column of Q_k. This does not account for missing and observed data yet. The GMM extension for missing data is built above what is presented here, this new notation only serves the mathematical justifications for the estimators that follow. Refer to section 3.4.1 for the GMM extension used later in this work.

In [6], 32 models corresponding to different parametrizations are introduced. They are distinctly separated in classes : with free orientations, common dimensions, common orientations and common covariance matrices. Given the redundancy associated with studying all 32 models, 10 models were appropriately chosen for this work as to study all unique parametrizations.

#	Notation
free orientations	
0	$[a_{kj}b_kQ_kd_k]$
1	$[a_{kj}bQ_kd_k]$
2	$[a_kb_kQ_kd_k]$
3	$[ab_kQ_kd_k]$
common dimensions	
4	$[a_{kj}b_kQ_kd]$
5	$[a_jb_kQ_kd]$
6	$[a_kb_kQ_kd]$
7	$[abQ_kd]$
common orientations	
8	$[a_kb_kQd]$
common covariance matrices	
9	$[abQd]$

Table 3.2: Selected HDDC models

Models with free orientations

The ML estimators of model parameters are closed form (CF)[3] for this category of models.
Subspace $\mathbb{E}_k$: The d_k first columns of Q_k are estimated by the eigenvectors associated with the d_k largest eigenvalues λ_{kj} of Σ_k. From [6], it can be proven that the log-likelihood can be rewritten as :

$$-2\log(L) = \sum_{k=1}^{G} N_k \left(\sum_{j=1}^{d_k} \log\left(a_{kj}\right) + (H - d_k)\log\left(b_k\right) + \frac{\mathrm{Tr}\left(\Sigma_k\right)}{b_k} + \sum_{j=1}^{d_k} \left(\frac{1}{a_{kj}} - \frac{1}{b_k} \right) \lambda_{kj} \right) + c^{st} \tag{3.12}$$

Hence, the objective of minimizing $-2\log(L)$ with respect to λ_{kj} can be equivalently achieved by minimizing the quantity $\sum_{k=1}^{G} N_k \sum_{j=1}^{d_k} \left(\frac{1}{a_{kj}} - \frac{1}{b_k} \right) \lambda_{kj}$. Since $\left(\frac{1}{a_{kj}} - \frac{1}{b_k} \right) < 0, \forall j = 1, \ldots, d_k$, it is necessary for λ_{kj} to be as large as possible. Consequently, the column vector q_{kj}, for all $j = 1, \ldots, d_k$, is estimated using the eigenvector associated with the j-th largest eigenvalue of Σ_k.

/0 $a_{kj}b_kQ_kd_k$ This corresponds to the most general model, which is used in [3],[12]. It will serve as the reference and the frame of comparison for the nine others. The partial derivatives of $-2\log(L)$ with respect to a_{kj} and b_k are :

$$-2\frac{\partial \log(L)}{\partial a_{kj}} = N_k \left(\frac{1}{a_{kj}} - \frac{\lambda_{kj}}{a_{kj}^2} \right) \text{ and } -2\frac{\partial \log(L)}{\partial b_k} = \frac{N_k\left(H - d_k\right)}{b_k} - \frac{N_k}{b_k^2} \left(\mathrm{Tr}\left(\Sigma_k\right) - \sum_{j=1}^{d_k} \lambda_{kj} \right) \tag{3.13}$$

[3] that is formed with constants, variables and a finite number of standard operations and functions.

The condition $\frac{\partial \log(L)}{\partial a_{kj}} = 0$ and $\frac{\partial \log(L)}{\partial b_k} = 0$ lead to the following estimators :

$$\begin{aligned} \hat{a}_{kj} &= \lambda_{kj} \\ \hat{b}_k &= \frac{1}{(H - d_k)} \left(Tr(\Sigma_k) - \sum_{j=1}^{d_k} \lambda_{kj} \right) \end{aligned} \tag{3.14}$$

In the remainder of this section, only the partial derivatives and their implications are presented. The conditions follow the same logic, unless explicitly given.

/1 $a_{kj} b Q_k d_k$ This first parametrization can be seen as modeling the noise in $E_k^{\perp}$ by a single parameter b. In other terms, b_k is common across all Gaussian components. $\hat{a}_{kj}$ is given by (3.14) and the partial derivative of $-2\log(L)$ with respect to b is :

$$-2\frac{\partial \log(L)}{\partial b} = \frac{n(H - \xi)}{b} - \frac{1}{b^2} \sum_{k=1}^{G} N_k \left(\mathrm{Tr}\left(\Sigma_k\right) - \sum_{j=1}^{d_k} \lambda_{kj} \right) \tag{3.15}$$

where $n = \frac{\sum_{k=1}^{G} N_k \lambda_{kj}}{\lambda_j}$. The estimator for b is given by :

$$\hat{b} = \frac{1}{(H - \xi)} \left(Tr(W) - \sum_{k=1}^{G} w_k \sum_{j=1}^{d_k} \lambda_{kj} \right) \tag{3.16}$$

where $\xi = \sum_{k=1}^{G} w_k d_k$ and $W = \sum_{k=1}^{G} w_k \Sigma_k$ is the within-covariance matrix.

/2 $a_k b_k Q_k d_k$ This can be seen such that each conditional covariance matrix Δ_k contains only two different eigenvalues, which according to [6] is an efficient way to regularize the estimation of Σ_k. Fixing a_{kj} to a_k replaces the d_k first eigenvalues by their mean. This is a similar transformation as to b_k in the general model **/0**. It has also been shown that the ML estimator of the intrinsic dimensions d_k is asymptotically consistent for this model [5].
The partial derivative of $-2\log(L)$ with respect to a_k is :

$$-2\frac{\partial \log(L)}{\partial a_k} = \frac{N_k d_k}{a_k} - \frac{N_k}{a_k^2} \sum_{j=1}^{d_k} \lambda_{kj} \tag{3.17}$$

The estimator of a_k is given by :

$$\hat{a}_k = \frac{1}{d_k} \sum_{j=1}^{d_k} \lambda_{kj} \tag{3.18}$$

and $\hat{b}_k$ is given by (3.14).

/3 $a b_k Q_k d_k$ Here, a_{kj} is fixed to a single eigenvalue a shared across all Gaussian components. The partial derivative of $-2\log(L)$ with respect to a is :

$$-2\frac{\partial \log(L)}{\partial a} = \frac{n\xi}{a} - \frac{1}{a^2} \sum_{k=1}^{G} N_k \sum_{j=1}^{d_k} \lambda_{kj} \tag{3.19}$$

The estimator of a is given by :

$$\hat{a} = \frac{1}{\xi} \sum_{k=1}^{G} w_k \sum_{j=1}^{d_k} \lambda_{kj} \tag{3.20}$$

and $\hat{b}_k$ is given by (3.14).

Models with common dimensions

Here, d_k is common within the classes. This implies choosing a single d, which leads to a constant number of unchanged eigenvalues across all components of the GMM. As such, the intrinsic dimensions of the subspace $\mathbb{E}_k$ is the same for all groups. This dimension d_k was determined using the scree test for the general model. However no information is given in [6] on how to choose a value for d when considering common dimensions.

$$\text{In this thesis, we fix } d = min_k(\overrightarrow{d}),$$

with $\overrightarrow{d}$ the vector containing the intrinsic dimensions d_k. This is motivated by the idea that d_k was chosen iteratively for each of the G components to mitigate the "risk" of exploding. Indeed, replacing the eigenvalues with their mean when they become too familiar to each other ensures that our approximation of Σ_k remains close to its original value. Moreover, a lower d leads to fewer parameters which is, as shown in Chapter 4, a criterion for a lower BIC, a metric used in our model selection process.

/4 $a_{kj}b_kQ_kd$ The estimators of a_{kj} and b_k are given by (3.14).

/5 $a_jb_kQ_kd$ The partial derivative of $-2\log(L)$ with respect to a_j is :

$$-2\frac{\partial \log(L)}{\partial a_j} = \frac{n}{a_j} - \frac{1}{a_j^2} \sum_{k=1}^{G} N_k \lambda_{kj} \tag{3.21}$$

The condition $\frac{\partial \log(L)}{\partial a_j} = 0$ and the relation $\sum_{k=1}^{G} N_k \lambda_{kj} = n\lambda_j$ gives the estimator for a_j :

$$\hat{a}_j = \lambda_j \tag{3.22}$$

where λ_j is the j-th largest eigenvalue of W. The estimator of b_k is given by (3.14).

/6 $a_kb_kQ_kd$ The estimator of a_k is given by (3.18) and b_k by (3.14).

/7 abQ_kd For this model, the expressions for $\hat{a}$ and $\hat{b}$ in the free dimensions context can be simplified to :

$$\begin{aligned} \hat{a} &= \frac{1}{d} \sum_{j=1}^{d} \lambda_j \\ \hat{b} &= \frac{1}{(H-d)} \left(\text{Tr}(W) - \sum_{j=1}^{d} \lambda_j \right) \end{aligned} \tag{3.23}$$

Models with common orientations

This family of model is divided in two : with free dimensions or common dimensions. In [6], only the models with common dimensions are described. Indeed, they use a simple iterative procedure (IP) to obtain the estimators, whereas with free dimensions the Maximum Likelihood (ML) estimation always requires a complex iterative estimation based on the FG algorithm.

Subspace $\mathbb{E}_k$: Given a_k and b_k, the d first columns of Q are estimated by the eigenvectors associated to the d largest eigenvalues of the matrix M defined by :

$$M\left(a_1,\ldots,a_G,b_1,\ldots,b_G\right)=\sum_{k=1}^{G}N_k\left(\frac{1}{b_k}-\frac{1}{a_k}\right)\Sigma_k \tag{3.24}$$

The log-likelihood (3.11) can be rewritten as :

$$-2\log(L)=\sum_{k=1}^{G}N_k\left(d\log\left(a_k\right)+(H-d)\log\left(b_k\right)\right)-\sum_{j=1}^{d}q_j^t(B-A)q_j+\mathrm{Tr}(B)+c^{st} \tag{3.25}$$

where $A=\sum_{k=1}^{G}\frac{N_k}{a_k}\Sigma_k$ and $B=\sum_{k=1}^{G}\frac{N_k}{b_k}\Sigma_k$.

/8 a_kb_kQd The partial derivatives of $-2\log(L)$ with respect to a_k and b_k are :

$$-2\frac{\partial\log(L)}{\partial a_k}=\frac{N_kd}{a_k}-\frac{N_k}{a_k^2}\sum_{j=1}^{d}q_j^t\Sigma_kq_j \text{ and } -2\frac{\partial\log(L)}{\partial b_k}=\frac{N_k(H-d)}{b_k}-\frac{N_k}{b_k^2}\left(\mathrm{Tr}\left(\Sigma_k\right)-\sum_{j=1}^{d}q_j^t\Sigma_kq_j\right) \tag{3.26}$$

Given Q, estimators of a_k and b_k are :

$$\begin{aligned}\hat{a}_k(Q)&=\frac{1}{d}\sum_{j=1}^{d}q_j^t\Sigma_kq_j\\ \hat{b}_k(Q)&=\frac{1}{(H-d)}\left(\mathrm{Tr}\left(\Sigma_k\right)-\sum_{j=1}^{d}q_j^t\Sigma_kq_j\right)\end{aligned} \tag{3.27}$$

Models with common covariance matrices

The ML estimators of model parameters are CF for this category of models. The log-likelihood (3.11) can be rewritten as :

$$-2\log(L)=n\left(\sum_{j=1}^{d}\log\left(a_j\right)+(H-d)\log(b)+\frac{\mathrm{Tr}(W)}{b}+\sum_{j=1}^{d}\left(\frac{1}{a_j}-\frac{1}{b}\right)q_j^tWq_j\right)+c^{st} \tag{3.28}$$

/9 $abQd$ The partial derivative of $-2\log(L)$ with respect to a is :

$$-2\frac{\partial\log(L)}{\partial a}=\frac{nd}{a}-\frac{n}{a^2}\sum_{j=1}^{d}q_j^tWq_j \tag{3.29}$$

This is the most restrictive model, leading to it having the fewest number of parameters. Estimators for a and b are given by (3.23).

3.5.3 Properties of the HDDC models

In [6], the properties of the parameters of all models are explored through a comparative study, of which an extract is presented in Table 3.3 to illustrate the differences between the 10 selected models. Let us define ρ the number of parameters required for the estimation of means and proportions as well as $\bar{\tau}$ and τ the number of parameters required for the estimation of $\tilde{Q}_k$ and $\tilde{Q}$ respectively :

$$\begin{aligned} \rho &= GH + G - 1 \\ \bar{\tau} &= \sum_{k=1}^{G} d_k \left[H - (d_k + 1)/2\right] \\ \tau &= d\left[H - (d+1)/2\right] \end{aligned} \tag{3.30}$$

Finally, $D = \sum_{k=1}^{G} d_k$. For asymptotic orders, [6] assumes $G \ll d \ll H$.

Model	Number of parameters	Asymptotic order	Example with $G = 4$, $d = 10, H = 100$	ML estimation
$[a_{kj}b_kQ_kd_k]$	$\rho + \bar{\tau} + 2G + D$	GHd	4231	CF
$[a_{kj}bQ_kd_k]$	$\rho + \bar{\tau} + G + D + 1$	GHd	4228	CF
$[a_kb_kQ_kd_k]$	$\rho + \bar{\tau} + 3G$	GHd	4195	CF
$[ab_kQ_kd_k]$	$\rho + \bar{\tau} + 2G + 1$	GHd	4192	CF
$[a_{kj}b_kQ_kd]$	$\rho + G(\tau + d + 1) + 1$	GHd	4228	CF
$[a_jb_kQ_kd]$	$\rho + G(\tau + 1) + d + 1$	GHd	4198	CF
$[a_kb_kQ_kd]$	$\rho + G(\tau + 2) + 1$	GHd	4192	CF
$[abQ_kd]$	$\rho + G\tau + 3$	GHd	4186	CF
$[a_kb_kQd]$	$\rho + \tau + 2G + 1$	Hd	1357	IP
$[abQd]$	$\rho + \tau + 3$	Hd	1351	CF

Table 3.3: From [6], study of the different parametrizations
(CF = Closed Form; IP = Iterative Procedure)

Chapter 4

Methodology

This chapter encompasses the methodology examined in this study, consolidating the theoretical groundwork laid in previous chapters. Section 4.1 outlines the intuition for dimensionality reduction with missing data, Section 4.2 presents the methodology applied by [3] while Section 4.3 elaborates on the extension proposed by this research.

4.1 A new objective

Due to DR techniques' limitations when dealing with incomplete datasets, evaluating cost functions optimized by nonlinear DR methods becomes complex. This complexity arises from relying on the coordinates of incomplete high-dimensional (HD) samples. A workaround involves using a complete HD dataset through imputation to align with data distribution in the feature space. Since features in most datasets aren't independent, considering conditional distribution of missing features given observed ones is crucial. This consideration allows for deriving meaningful insights on the relationships within the data. Traditional approaches for addressing this include replacing missing data with their conditional mean based on the observed features (assuming the data distribution can be modeled), or utilizing nearest neighbor-based imputation methods. However, these approaches would, at best, yield a low-dimensional (LD) embedding that minimizes the cost function when evaluated using the expected HD dataset. Consequently, this leads to the following LD embedding :

$$\arg\min_{\mathbf{X}} C(\mathbf{X}, \mathbb{E}[\mathbf{\Xi}]) \tag{4.1}$$

with Ξ a set of incomplete HD data points and $\mathbf{X}$ the set of their LD representations. However, one would be more 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 :

$$\arg\min_{\mathbf{X}} \mathbb{E}[C(\mathbf{X}, \mathbf{\Xi})]. \tag{4.2}$$

Intuitively, the expectation is just moved one step later in the process but this has big consequences for nonlinear cost functions. To address the optimization of (4.2), the cost function expectation must be approximated on the incomplete database and this is where the multiple imputation paradigm proves useful as it provides a consistent approximation if the conditional distribution of the missing features given the observed ones can be modeled [13].

4.2 Procedure step-by-step

As stated in Chapter 1, the missing-data management is addressed immediately in the first step of the process. The previous section shows that in order to obtain the closest cost function to the one yielded by the complete dataset, multiple imputed complete datasets must be derived, which are obtained from fitting a GMM model on Ξ. From [3], it is possible to summarize the procedure of applying a NLDR to Ξ, a data set with missing values, as :

i. Fitting the Gaussian Mixture Model on Ξ

This first step refers to the methodology presented in Section 3.4.1. The goal is to maximize the observed log-likelihood (3.5) over the model parameters θ using the EM-algorithm. Since the EM algorithm is sensitive to parameter initialization, multiple runs with different initializations are conducted to explore various local maxima. Additionally, LD representations (Section 3.5) of each Gaussian function are adopted to address the scaling issue of the GMM, which is typically $\mathcal{O}\left(GH^2\right)$. Several models from the HDDC family will be tested for this purpose, further discussed in Section 4.3.

For HDDC models with free dimension d_k, the intrinsic dimension is determined using the *scree test*. Specifically, d_k is set to the smallest integer between 1 and H for which the subsequent successive differences in eigenvalues are smaller than a threshold η_k. The threshold η_k and the number G of components are determined using the Bayesian Information Criterion (BIC). If the number of parameters P_θ (Table 3.3) of the model characterized by θ is smaller than N, the number of independent examples, the BIC can be expressed as :

$$-2 \log \mathcal{L}(\theta) + P_\theta \log N \tag{4.3}$$

where $\mathcal{L}(\theta)$ is the model likelihood. The BIC is also used to choose the number of components G. It is chosen by fitting two models, with $G = 1$ and $G = 2$. If the BIC is lower with an additional component, $G = 3$ is tested and so on. For each value of G, η_k is defined as a decreasing fraction of the covariance trace such that $\eta_k = \zeta \cdot \operatorname{tr}(\Sigma_k)$, starting with $\zeta = 0.5$ and dividing its value by two as long as the BIC decreases. The selected model is the one with the lowest BIC. The number of parameters grows quickly with the dimensionality of the problem, so considering parsimonious models is essential (see Section 4.4).

ii. Using the GMM model to perform multiple imputation by extracting Ψ random samples in order to create Ψ complete datasets $\Xi_c^1, ..., \Xi_c^\Psi$

To obtain a random sample from the conditional distribution, follow these steps :

1. Draw a component index k from $\mathcal{G}$ with a probability t_{ik}
2. Sample a vector $\mathbf{y}_{ik}$ from an $|M_i|$-dimensional Gaussian distribution with zero mean and identity covariance
3. Compute $\tilde{\mu}_{k,M_i|O_i} + \mathbf{C}_{ik}\mathbf{y}_{ik}$, where $\mathbf{C}_{ik}$ is the Cholesky factor of $\tilde{\mathbf{\Sigma}}_{k,M_i|O_i}$

The Cholesky factorization is defined as the decomposition of a Hermitian, positive-definite matrix into the product of a lower triangular matrix and its conjugate transpose.

iii. Averaging the nonlinear DR cost function evaluations on each Ξ_c^l to approximate its expectation on the incomplete database

It is important to state that any DR scheme optimizing some criterion can be used instead of MsSNE. The reasons to why it is the chosen method in this work are described in Section 2.3.2. [3] proposes comparisons of different DR methods.

Applying the MsSNE methodology presented in Section 2.3.2, the expectation over the missing values is minimized s.t. :

$$\mathbb{E}\left[C_{\text{MsSNE}}\right] = -\sum_{i\in\mathcal{I},j\in\mathcal{I}\setminus\{i\}} \mathbb{E}\left[\sigma_{ij,\ \text{ms}}\right] \log s_{ij,\ \text{ms}}. \tag{4.4}$$

From the Ψ imputed complete datasets, their multi-scale HD similarities are computed, allowing to approximate the expectation of the multi-scale HD similarities of Ξ :

$$\mathbb{E}\left[\sigma_{ij,\ \text{ms}}\right] \approx \frac{1}{\Psi}\sum_{l=1}^{\Psi} \sigma_{ij,\ \text{ms}}^{l} \tag{4.5}$$

Finally, the approximation quality of the HD similarities is defined by the sum of the KL divergences between the "true" and approximated similarities for each data point :

$$Q_{\text{KL}} = \sum_{i\in\mathcal{I},j\in\mathcal{I}\setminus\{i\}} \sigma_{ij,\ \text{ms}} \log\left(\sigma_{ij,\ \text{ms}}/\hat{\sigma}_{ij,\ \text{ms}}\right) \tag{4.6}$$

with $\hat{\sigma}_{ij,\ \text{ms}}$ estimated by (4.5). This metric will be used alongside the AUC (2.13) to assess the quality of the methodology.

4.3 Trying other parametrizations

The referenced works, [3] and [6], focused at most on relatively "low high dimensions", 64 and 100 respectively. However, real-world data can come in much higher dimensions, ranging from hundreds to even thousands. Now, armed with the theoretical foundations from previous works and already some exclusive insights presented in the previous chapters, the methodology is put to the test in higher dimensions.

In [3], the implementation of the $[a_{kj}b_kQ_kd_k]$ model from the HDDC extension successfully addressed the challenging task of fitting a GMM model for up to 64 dimensions. Reducing the dimensionality of the specific subspace of each group led to a more parsimonious model, adhering to the general idea that "simpler is better". However, there is uncertainty about whether this general model is restrictive enough to fit GMMs on very high dimensions. To explore this further, stricter parametrizations, as presented in Table 3.2, are considered to observe performance differences and potential breakthroughs these could allow.

Out of the 32 presented models, the selection of the 10 models for analysis is based on two criteria :

- Targeting specific parametrizations rather than combining multiple ones, to avoid redundancy and better understand the impact of each restriction.

- Excluding models that use the FG (Full Gaussian) procedure for the Maximum Likelihood (ML) estimation, as it is a very complex procedure.

The questions raised in this section are, to the best of our knowledge, exclusive to this work.

4.4 A side note on parsimonious models

Definition 7 *A parsimonious model is a model that accomplishes the desired level of explanation or prediction with as few predictor variables as possible.*

When considering the BIC (4.3) for model selection, a lower BIC implies either fewer explanatory variables, better fit or both. Hence, it is a good estimate of parsimony. The Gaussian Mixture framework is highly parameterized, which naturally yields inference problems in HDS[4]. Constraining, and by extension simplifying, the general $[a_{kj}b_kQ_kd_k]$ model could allow us to regularize it in high-dimension.

Chapter 5

Experimental results

In this chapter, simulations are performed using the HDDC models which were specifically implemented in Python for this thesis. For reproduction purposes, a description of the virtual environment used is given in Section 5.1. A study of the number of parameters for each model on increasing dimensions is presented in Section 5.2. Then, the limits of the reference model are challenged and established in Section 5.3. The nine new models are compared to the tests performed on the reference model in Section 5.4. Finally, the relevancy of the method in very high dimensions is questioned in Section 5.5 by comparing it to other imputation methods.

5.1 Implementation

To perform the numerical implementation and generate the results presented in this chapter, the open demo provided by [3] served as the starting point. It can be accessed on the GitHub of Cyril de Bodt. Following the authors' recommendation, the virtual environment used for the implementation is as follows:

```
Python: 3.7.5
Numpy: 1.17.4
Numba: 0.46.0
Scipy: 1.3.2
Matplotlib: 3.1.1
Scikit-learn: 0.22
Pandas: 0.25.3
```

5.2 $P(\theta)$ with growing dimensionality

To further extend the study presented in Table 3.3, variations on the parameter H, the dimension of the input data, are studied. This additional analysis will provide valuable insights into the performance of the HDDC models across input data dimensions ranging from 5 to 100. By utilizing (3.30) and considering the asymptotic order $G \ll d \ll H$, it is deduced that the original dimensionality H and the "retained" dimensionality d_k of the dataset play significant roles in determining the number of parameters in the HDDC models.

The results of the parameters study are presented in Figure 5.1. Analysis is performed for $d_k = [0, 1, 5, 10]$, $G = 4$, and d_k constant across all components.

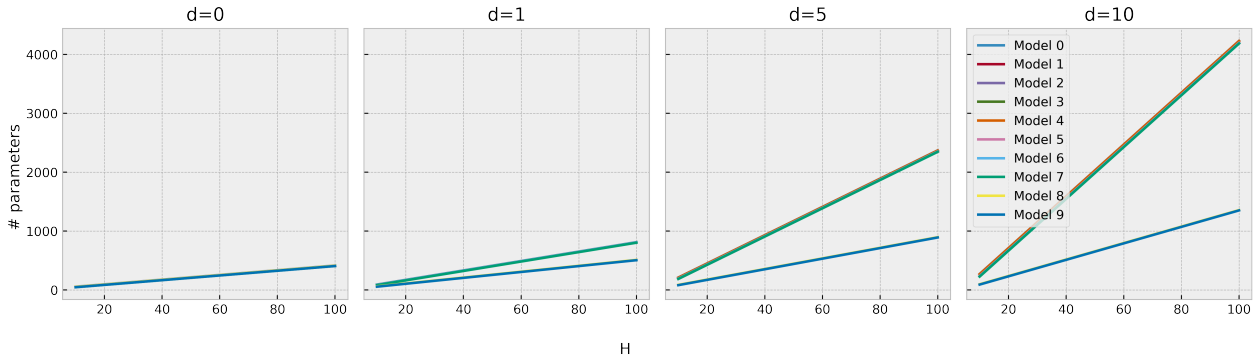

Figure 5.1: Parameters study

If we were to let the model select $d_k = 0$ by letting the scree test unbounded and if the difference between a_{k1} and a_{k2}, sorted in decreasing order, was already too small, then the covariance matrix would only hold eigenvalues simplified to b_k. As a result, τ and $\bar{\tau}$ would simplify to 0 in (3.30). Forcing the models to retain at least some of the largest eigenvalues avoid including them with the b or b_k last eigenvalues, allowing to better differentiate the models and study the impact of dimensionality on the other parameters of the model.

Models 8 and 9, which implement common orientations and common covariance matrices, show slower growth with increasing dimensionality. This aligns with the expected results from Table 3.3, as these models represent the strictest regularizations. Besides these two, considering $P(\theta)$ only, all models seem quite similar and are nearly indistinguishable from each other. The different restrictions are an attempt at reducing the number of parameters of the model. In the case where $P(\theta)$ is approximately the same for all models, it can be expected that they produce similar results. Following that intuition, if all models select $G = 1$, the different parametrizations become less different from one another and similar performance is to be expected.

5.3 Limits of the general model

In [3], experiments are conducted for dimensions up to 64, corresponding to the Optical Recognition of Handwritten Digits (Digits) dataset. Before exploring other models, it is important to understand what the limits of our reference model $[a_{kj} b_k Q_k d_k]$ are. It has already been established that the methodology is limited in terms of input data as both the GMM framework and the SNE techniques come with exhaustive computation time. This section tries to quantify the limit in terms of dimensions. Using artificial isotropic Gaussian blobs datasets, or simply BLOBs, is a flexible method to vary the dimensionality of the dataset. It creates multi-class datasets by allocating each class one or more normally distributed clusters of points.

5.3.1 Simulations

From complete datasets, 5% of missing values is artificially introduced, while making sure that no entire example or feature goes missing.

Experiment 1 : increasing dimensions with free number G of Gaussian components

The following table summarizes the experimental setting :

Models	Dimensions	Data Points	Clusters	Gaussian Components
[0]	[3 $\longrightarrow$ 400]	500	3	Free, chosen with BIC

This experiment examines the methodology's performance across increasing dimensionality. Figure 5.2 demonstrates its remarkable success for dimensions up to 50. However, beyond this threshold, the quality of the LD embeddings degrades rapidly. The most concerning observation is the complete failure of the methodology at 300+ dimensions.
This observation highlights the limitations of the current approach when dealing with very high-dimensional data. While it exhibits promising results for moderately high dimensions, it struggles to handle extremely high-dimensional datasets effectively.

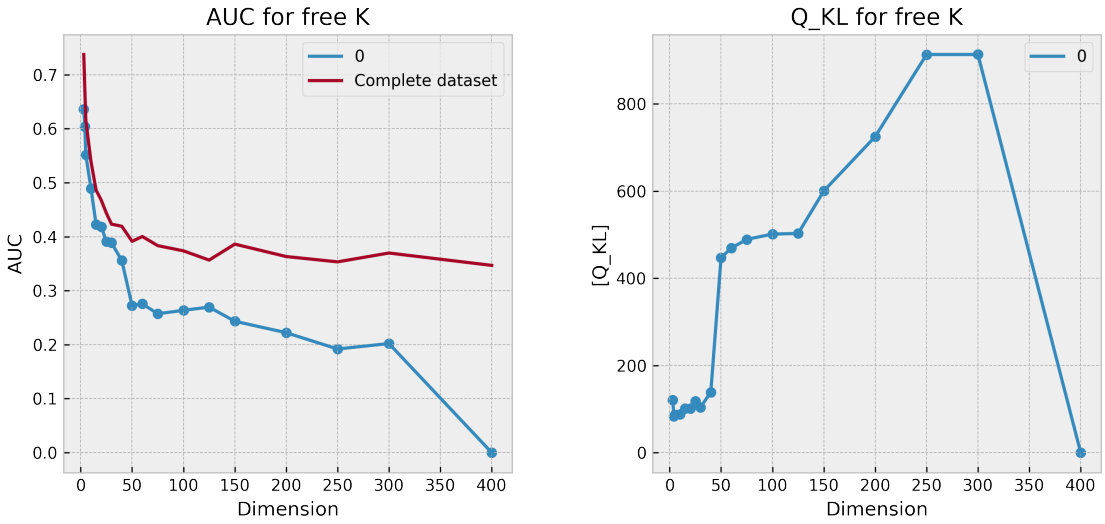

Figure 5.2: $[a_{kj} b_k Q_k d_k]$

A few hypotheses can be established on why it fails at 300+ dimensions :

1. A sufficient (and exponential) amount of data is required for a given dimension. Otherwise the empty space phenomenon leads to numerical instability at some point during the process.
2. The chosen test dataset (BLOBs) is inappropriate for some reason.
3. The problem is intrinsically numerical : GMMs scale poorly in HD as they need to compute a determinant of a matrix with dimension $(H \times H)$. This is numerically unstable.

Experiment 2 : fixed HD on different dataset size

In this next experiment, the first hypothesis is challenged by studying the impact of the ratio between the quantity of data and the number of dimensions. An arbitrarily very high dimension of 200 is fixed and the performance for BLOBs dataset varying across a wide range of sizes is studied.

Models	Dimensions	Data Points	Clusters	Gaussian Components
[0]	[200]	[10 $\longrightarrow$ 320]	3	Free, chosen with BIC

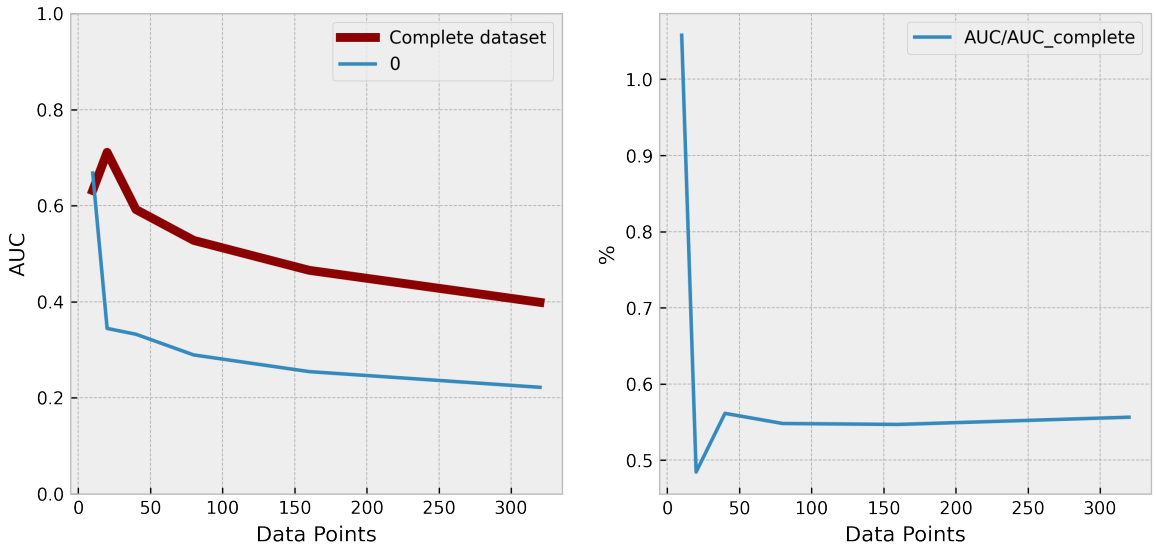

Figure 5.3: Performance across growing input data size

Figure 5.3 demonstrates that the methodology doesn't fail even with a very small ratio (here $\frac{1}{20}$). In HDS, an exponential number of data points are required to effectively describe the space. While more than 320 points should be used to fill the 200-dimensional space, this would be too computationally intensive. The performance remains constant in the range [10 $\longrightarrow$ 320] due to this limitation, and the spike at $p = 10$ is merely noise. Furthermore, having fewer data points than parameters renders the BIC (4.3) inappropriate, as the problem is underdetermined, much like having more weights than data points in a linear regression scenario.

Experiment 3 : reference model on UCI datasets

In [3], the methodology is tested on many UCI Machine Learning repository datasets. In this experiment, a real dataset of very high dimensionality is studied to challenge the second hypothesis. The dataset is ISOLET [9].

Models	Dimensions	Data Points	Dataset	Gaussian Components
[0]	[617]	[500]	ISOLET	Free, chosen with BIC

The exact same error arises, dismissing the second hypothesis. The problem appears to be purely numerical, meaning that without changes, the general model will never work for very high dimensions.

Experiment 4 : Initialization of $\det(\Sigma_k)$ with increasing dimensions

As to study whether the problem does indeed arise from numerical approximations, the average value of the first estimation of $\det(\Sigma_k)$, which happens during the initialization of the EM algorithm, is tracked through increasing dimensions.

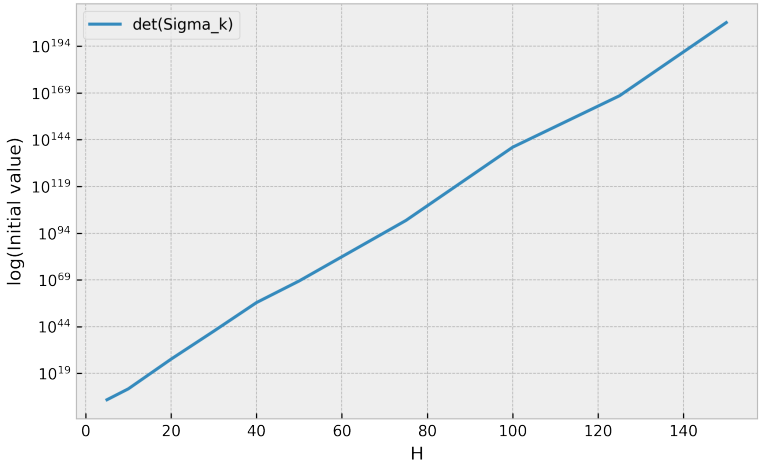

Figure 5.4: $\max_k(\det(\Sigma_k))$ for increasing dimensions

The determinant becomes excessively large and numerically approximated to 'inf'. This issue is inherent to GMM as an initial estimate of $\det(\Sigma_k)$ is necessary for the EM algorithm.

During experimentation, another problem arising from the high-dimensional (HD) nature of the data was identified. In the EM algorithm, even if $\det(\Sigma_k)$ after initialization is large, as long as an initial value can be obtained it will quickly converge to significantly smaller values within two or three iterations. However, after a few more iterations, extremely small values are sometimes computed, often approaching zero and being treated as such numerically. To address this, an additional constraint was introduced in the GMM fitting process :

1. Fit mixture models with more components until the BIC increases or until there are more parameters than samples
2. For a given number G of components, fit n times to retain the best one in terms of BIC
3. Fit mixture models with a decreasing threshold (ζ_k) following the scree test (as long as the BIC decreases)
4. For a given threshold ζ_k, fit n times and retain the best one in terms of BIC
5. **Monitor determinant computations, proceed to the next ζ_k if :**
   - **Any $|\det(\Sigma_k)|$ for $k \in G$ falls below 10^{-50} after the E-step[1].**
   - **Any** $\det$ **during computation of a pdf for t_{ik} is zero (see 5.3.2).**

[1]In Python, the maximum integer e such that 10^e is in the range of representable finite floats is 308. It follows normal double precision, typical of 64-bit floating point numbers. The exact range is $\pm 2.22507385850720 \times 10^{-308}$ to $\pm 2.22507385850720 \times 10^{308}$ with a precision of about 15 significant decimal digits.

While this only partially solves the problem of handling higher dimensionality, it makes the methodology more stable for dimensions where at least an initial value of $\det(\Sigma_k)$ is obtained after the initialization. It stops the process when values are getting dangerously close to being numerically approximated to zero, which would lead to a zero division.

5.3.2 First observations

With very high dimensions, two problems related to numerical computation arise. It happens during the initialization of the EM algorithm and during the E-Step 2 for which the marginal pdf of the multivariate Gaussian over the observed features to compute t_{ik} is needed and computed as :

$$f_O = \frac{\exp(a/2) * (2\pi)^{N_O/2}}{\sqrt{\det(S_O)}} \tag{5.1}$$

with a corresponding to twice the argument of the exponential in the pdf of the marginal multivariate Gaussian over the observed values of the current example [12], N_O the number of observed features of the current example and $\det(S_O)$ the determinant of the sub-matrix over the observed feature of the current example of the covariance matrix of the multivariate Gaussian. If $\det(S_O)$ is approximated to zero, a zero division error occurs. An additional stopping criterion is implemented to circumvent this problem.

More dramatically, the initialization of $\det(\Sigma_k)$ grows exponentially with the dimensionality of the problem, as shown in Figure 5.4. The challenge for the new parametrizations is whether they can succeed in lowering the values of $\det(\Sigma_k)$ and stabilizing the eigenvalues λ_{kj} of that same covariance matrix.

5.4 Comparing HDDC models

5.4.1 Quality assessment for dimensions lower than 100

Before assessing the problem of very high dimensionality, the performance of the other models on datasets tested in [3] is studied. Namely, the Yacht Hydrodynamics, Wine [1], Indian Liver Patient (ILP) [30], Energy efficiency (EE) [39] and Iris [14] datasets. Random subsets are taken when the number of input data exceeds 400 to accelerate the computation.

Experiment 5 : new parametrizations with free G

	Ref	1	2	3	4	5	6	7	8	9	Complete
$\text{Iris}_{(150\times4)}$	.718	.725	.729	.712	.715	**.73**	.716	.718	.723	.718	.786
$\text{Hydro}_{(308\times6)}$	.652	.65	.653	**.666**	.645	.661	.659	.631	.64	.631	.688
$\text{EE}_{(400\times8)}$	**.592**	.592	.58	.58	.592	.592	.58	.58	.58	.58	.67
$\text{ILP}_{(400\times10)}$	.582	.599	.579	.59	.572	.599	.592	**.6**	.56	.599	.624
$\text{Wine}_{(178\times13)}$	.579	**.596**	.584	.587	.58	.58	.578	.589	.579	.589	.643

Table 5.1: AUC with different parametrizations of HDDC (see Table 3.2)

For EE, the performance is basically identical across all models. This can be attributed to the fact that only one Gaussian component was fitted for each model. As a result, there is limited freedom for each model to showcase the uniqueness of its restrictions. The parametrizations essentially simplify to similar transformations when G is set to 1.

Across each dataset, the best performing model varies, although the differences compared to the reference model are small. The non-convex nature of the problem implies that each model could potentially perform best through fortuitous initialization. Still, one could make a suggestion to explore a few restrictions in the process of fitting an HDDC model, similarly to the selection of G and ζ_k. While this process extends the computation time in the GMM fitting step by the number of tested models, it potentially represents a small but valuable avenue for improving the methodology presented in [3] at a scale of 1-5% in the $AUC_{\Xi}/AUC_{complete}$ ratio.

5.4.2 Higher dimensions

Experiment 6 : all models on increasing dimensions with free G

Similarly to the first experiment 5.3.1, BLOBs are used to test the performance of the models on increasing dimensions. As shown in previous experiments, the dimensionality of the problem leads to large or small determinants approximated to 'inf' or 0. We implement the more restrictive parametrizations and their estimators for the eigenvalues presented in Section 3.5.2 as an attempt to solve this problem. Individual plots for each model are given in Annex A.

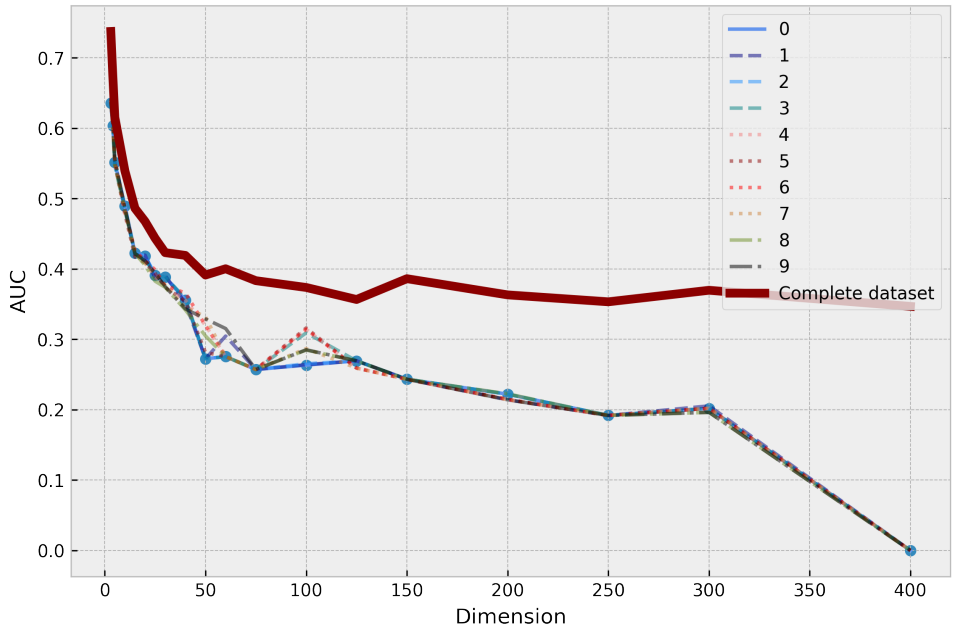

Figure 5.5: AUC for all models on increasing dimensions

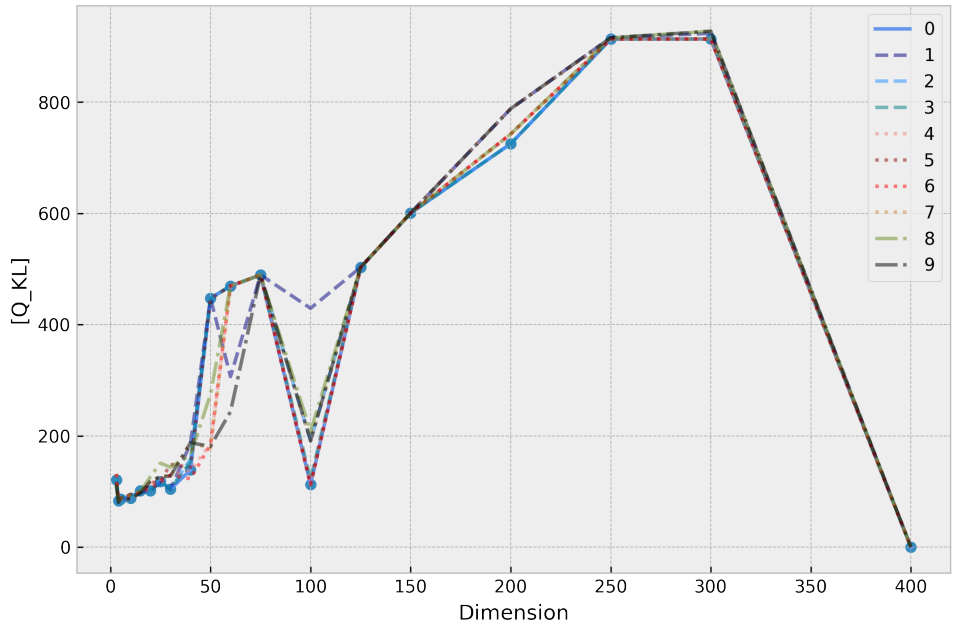

Figure 5.6: Q_{KL} for all models on increasing dimensions

All models encounter failure when dealing with dimensions beyond 300. As it was the case for the reference model, a failure occurs during the initialization of the EM algorithm, where the determinant of Σ_k is excessively large and computationally approximated to infinity. This problem arises immediately even with $G = 1$ and the largest threshold. None of the parametrizations prove robust enough to handle this issue, meaning that none of them have the opportunity to attempt any computations.

Conducting an analysis akin to Experiment 4 reveals that in HDS the value of $\det(\Sigma_k)$ after initialization is almost identical for all models. This similarity shows that the restrictions are not powerful enough to mitigate the numerical instability, causing all parametrizations to behave similarly.

Since the determinant is equal to the product of all a_{kj} and b_k eigenvalues, one could argue that manipulating the logarithm of the determinant, the sum of the logarithms of each eigenvalue, could be an effective way of stabilizing its computation. However this does not effectively address the problem, given that the EM algorithm's computations are also exacerbated by the high dimensionality of the dataset. This prompts a question about the viability of remaining within the GMM framework in extremely high dimensions, given its evident limitations.

5.5 Relevancy of the method for very high dimensions

After investigating the maximum feasible dimensionality, our focus naturally shifts to a comparison between the methodology and other imputation techniques. This comparison aims to offer insights into identifying the most suitable imputation technique in the context of NLDR as dimensionality expands.

Experiment 7 : Comparison with other imputation methods

Given the minimal performance variations observed across different parametrizations in previous experiments, this analysis exclusively studies the general model for comparisons. The comparison is conducted against three other imputation techniques using BLOBs datasets through increasing dimensionality. We monitor the performance in terms of the AUC and the Q_{KL}.

1. Single Imputation with feature mean : Missing values are replaced by the mean along each feature.
2. Single Imputation with KNN [38] : Missing values are replaced using the mean value from $K = 5$ nearest neighbors and using uniform weights.
3. Single Imputation with Iterative imputation [40], [7] : Missing values are replaced by modeling each feature with missing values as a function of other features in a round-robin fashion. The estimator used is a Bayesian ridge model [27], [37].
4. Multiple Imputation using GMMs : The proposed methodology, refer to Section 4.2.

The three methods implemented for comparison lead to the LD embedding given in (4.1), while the proposed methodology yields (4.2).

Dimensions	Data Points	Dataset
$[5 \longrightarrow 400]$	[400]	BLOBs

From Figure 5.7-5.9, the methodology demonstrates competitiveness with other techniques up until around 50 dimensions. Beyond this point, it succumbs more severely to the curse of dimensionality than others, and its performance deteriorates rapidly. The elbow of the *AUC* is still higher than that of simple imputation with the feature mean, however the performance becomes similar around the point where numerical instability causes it to fail completely.

The superiority of the proposed methodology has been proven with real datasets in [3]. However, it appears that the method becomes impractical for datasets with a dimensionality exceeding 50. At such high dimensions, more conventional methods prove more advantageous, not only in terms of the resulting LD embeddings but also, more crucially, in terms of computation time. The time complexity of alternative imputation techniques is substantially lower than that of the proposed method.

For dimensions surpassing 300, fitting a Gaussian Mixture Model, even with its HDDC extension, to perform multiple imputation appears to be simply not possible. In contrast, other methods still manage to produce results.

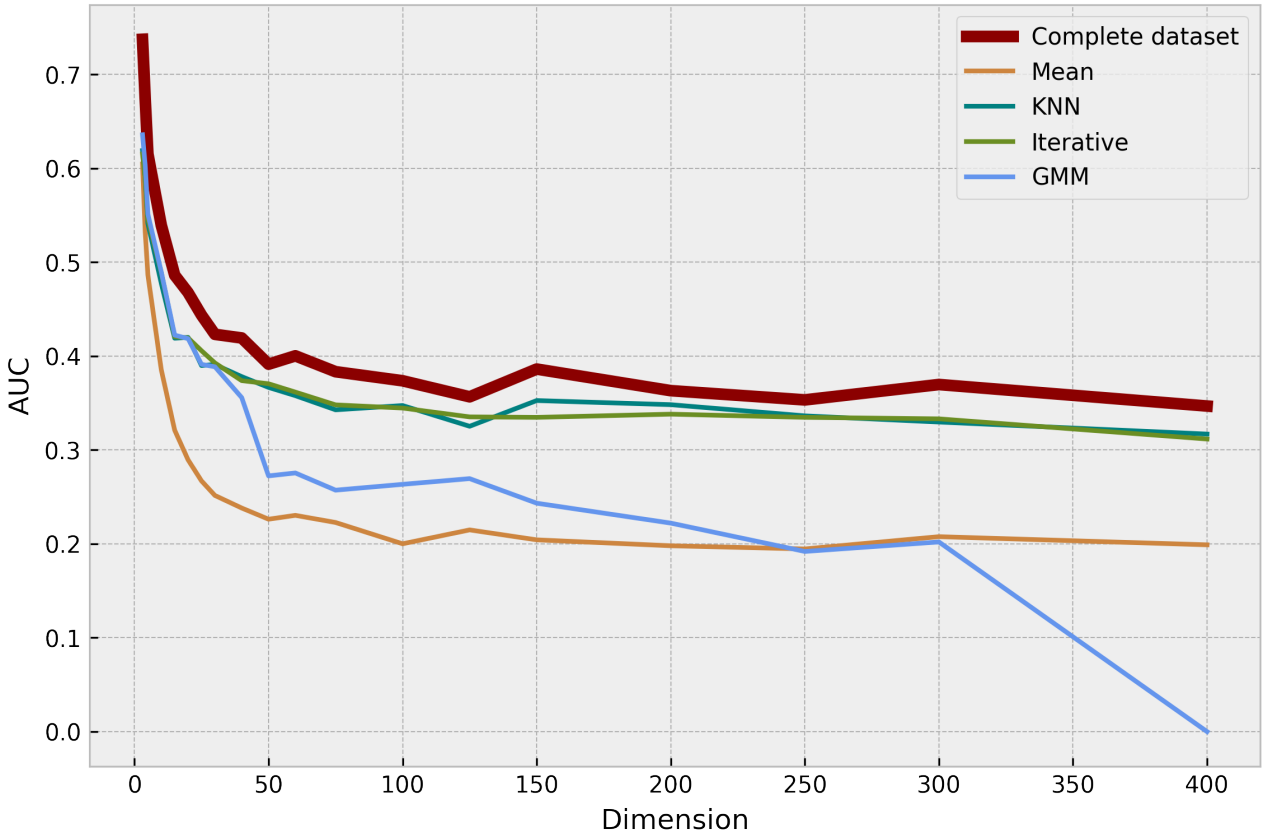

Figure 5.7: AUC comparison for other imputation methods on increasing dimensions to perform MsSNE

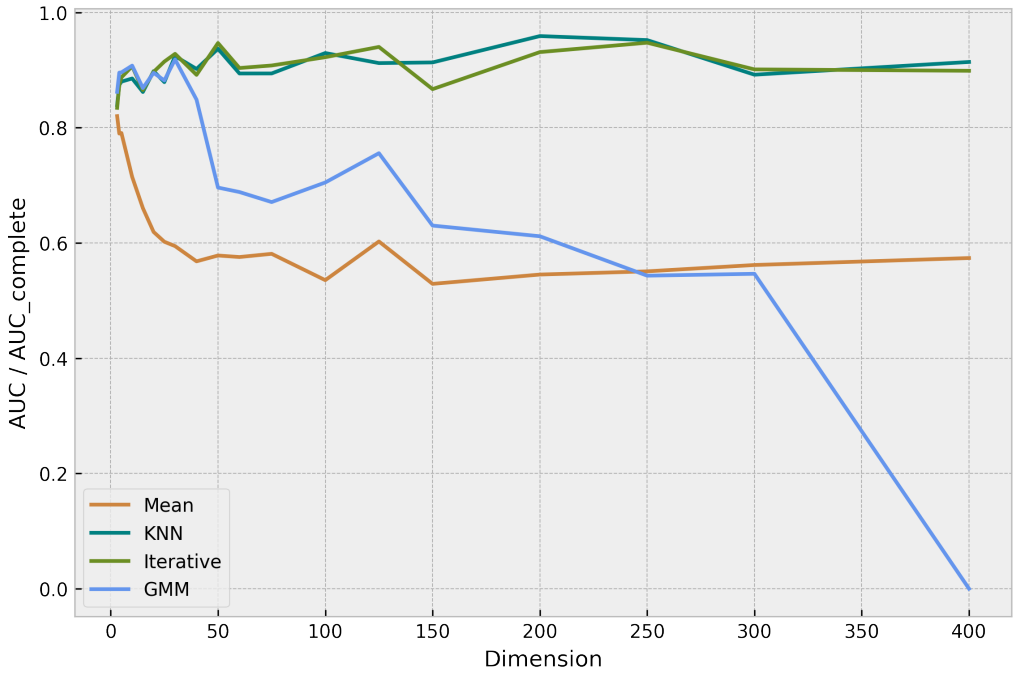

Figure 5.8: $AUC/AUC_{\text{complete}}$ ratio,
accuracy of imputation in the context of NLDR

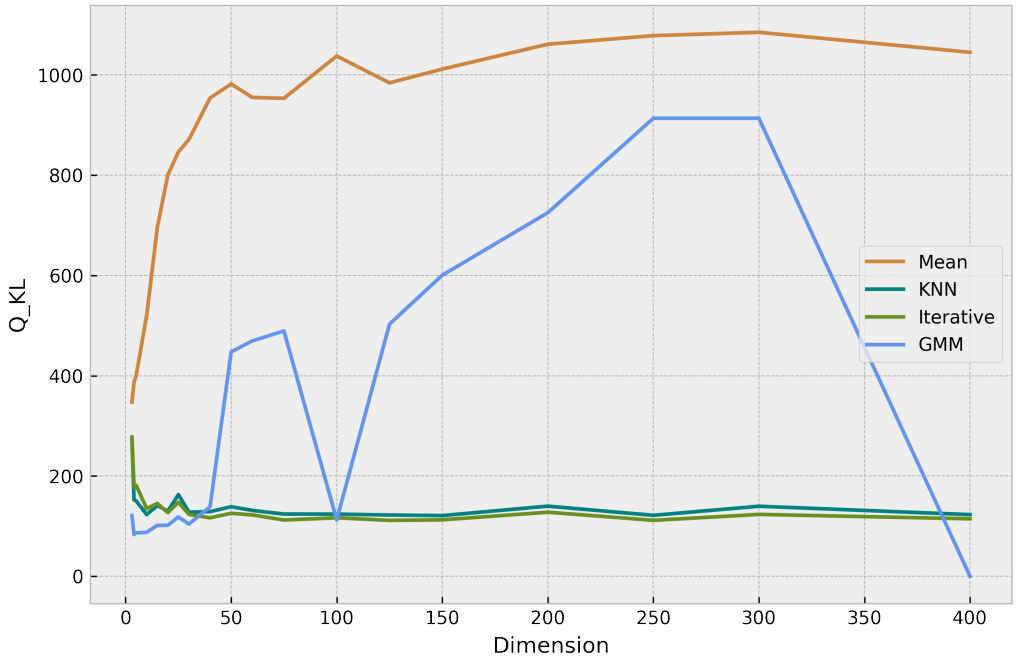

Figure 5.9: Q_{KL} comparison for other imputation methods
on increasing dimensions to perform MsSNE

Chapter 6

Conclusions

This master thesis featured an exploration of the Dimensionality reduction and Missing data fields. An overview of each was presented, depicting the state of the art. The two were then combined, extending on the work of [3] and [6]. The methodology proposed in [3] has been challenged by tackling higher dimensions. Anticipating its limits, the parametrizations depicted in [6] were presented as a potential solution. In Chapter 5, an analysis of these constrained methods' parameters was presented, before diving in several experiments. First, the limits of the methodology on the reference model were explicitly established. Consequently, the proposed parametrizations were then examined on both artificial and real UCI datasets in an attempt to go beyond these limitations.

The results demonstrated a notable similarity among these parametrizations, and very little variation to the most general model. Still, it is suggested that the parametrizations may be of use to compensate the inherent non-convexity of the problem. In the process of fitting a GMM model, just as G mixture components and different values of the threshold ζ_k are tested, different parametrizations could be considered.

Given that the parametrizations offer no help in overcoming the dimensionality cap dictated by numerical limitations, other imputation techniques were studied for comparison. It is argued that the proposed methodology is relevant for up to 50 dimensions. It then suffers considerably from the curse of dimensionality and the resource-intensive computations required for fitting Gaussian Mixture Models within higher-dimensional spaces. Yet, provided with sufficient computational resources and a dataset of reasonable dimensionality, the methodology thrives. This validates the intuition of trying to obtain the closest cost function to the one we would have with the complete dataset (4.2), rather than performing the dimensionality reduction on the expected dataset (4.1).

Further works

Exploring non-parametric methods could prove to be a promising avenue for improvement. Iterative imputation techniques, treating missing values as target variables for regression models, are actively researched [2] and continually expanded [20], [28]. Framing missing data imputation as a predictive task allows for the application of various supervised algorithms, including Deep

Learning techniques [17], [34]. These methods can directly benefit from the advancements in the field of machine learning. Furthermore, undertaking an exhaustive assessment of imputation techniques in high-dimensional spaces would be highly valuable. This task would involve an extensive exploration of appropriate techniques for increasing dimensionality, in the continuation of what is studied in this work.

Bibliography

[1] Stefan Aeberhard and M. Forina. *Wine*. UCI Machine Learning Repository. DOI: https://doi.org/10.244 1991.

[2] Melissa J Azur et al. "Multiple imputation by chained equations: what is it and how does it work?" In: *International journal of methods in psychiatric research* 20.1 (2011), pp. 40–49.

[3] Cyril de Bodt et al. "Nonlinear dimensionality reduction with missing data using parametric multiple imputations". In: *IEEE Transactions on Neural Networks and Learning Systems* 30.4 (2018), pp. 1166–1179.

[4] Charles Bouveyron and Camille Brunet-Saumard. "Model-based clustering of high-dimensional data: A review". In: *Computational Statistics & Data Analysis* 71 (2014), pp. 52–78.

[5] Charles Bouveyron, Gilles Celeux, and Stéphane Girard. "Intrinsic dimension estimation by maximum likelihood in isotropic probabilistic PCA". In: *Pattern Recognition Letters* 32.14 (2011), pp. 1706–1713.

[6] Charles Bouveyron, Stéphane Girard, and Cordelia Schmid. "High-dimensional data clustering". In: *Computational statistics & data analysis* 52.1 (2007), pp. 502–519.

[7] Samuel F Buck. "A method of estimation of missing values in multivariate data suitable for use with an electronic computer". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 22.2 (1960), pp. 302–306.

[8] Gilles Celeux and Gérard Govaert. "Gaussian parsimonious clustering models". In: *Pattern recognition* 28.5 (1995), pp. 781–793.

[9] Ron Cole and Mark Fanty. *ISOLET*. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/ 1994.

[10] Pierre Demartines and Jeanny Hérault. "CCA: Curvilinear component analysis". In: *Proceedings of 15th workshop GRETSI*. 1995, pp. 15–9.

[11] David L Donoho et al. "High-dimensional data analysis: The curses and blessings of dimensionality". In: *AMS math challenges lecture* 1.2000 (2000), p. 32.

[12] Emil Eirola et al. "Mixture of Gaussians for distance estimation with missing data". In: *Neurocomputing* 131 (2014), pp. 32–42.

[13] Craig K Enders. *Applied missing data analysis*. Guilford Publications, 2022.

[14] R. A. Fisher. *Iris*. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C56C76. 1988.

[15] Bernhard N Flury. "Common principal components in k groups". In: *Journal of the American Statistical Association* 79.388 (1984), pp. 892–898.

[16] Zoubin Ghahramani and Michael I Jordan. "Learning from incomplete data". In: (1995).

[17] Lovedeep Gondara and Ke Wang. "Mida: Multiple imputation using denoising autoencoders". In: *Advances in Knowledge Discovery and Data Mining: 22nd Pacific-Asia Conference, PAKDD 2018, Melbourne, VIC, Australia, June 3-6, 2018, Proceedings, Part III 22*. Springer. 2018, pp. 260–272.

[18] PC Hammer. *Adaptive control processes: a guided tour (R. Bellman)*. 1962.

[19] Geoffrey E Hinton and Sam Roweis. "Stochastic neighbor embedding". In: *Advances in neural information processing systems* 15 (2002).

[20] Shahidul Islam Khan and Abu Sayed Md Latiful Hoque. "SICE: an improved missing data imputation technique". In: *Journal of big Data* 7.1 (2020), pp. 1–21.

[21] Vipin Kumar and Sonajharia Minz. "Feature selection: a literature review". In: *SmartCR* 4.3 (2014), pp. 211–229.

[22] 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". In: *Neurocomputing* 169 (2015), pp. 246–261.

[23] John A Lee, Michel Verleysen, et al. *Nonlinear dimensionality reduction*. Vol. 1. Springer, 2007.

[24] John Aldo Lee, Amaury Lendasse, Michel Verleysen, et al. "Curvilinear distance analysis versus Isomap." In: *ESANN*. Vol. 2. 2002, pp. 185–192.

[25] John Aldo Lee et al. "A robust nonlinear projection method". In: *European Symposium on Artificial Neural Networks (ESANN'2000)*. 2000.

[26] Roderick JA Little and Donald B Rubin. *Statistical analysis with missing data*. Vol. 793. John Wiley & Sons, 2019.

[27] David JC MacKay. "Bayesian interpolation". In: *Neural computation* 4.3 (1992), pp. 415–447.

[28] Sanaz Nikfalazar et al. "Missing data imputation using decision trees and fuzzy clustering with iterative learning". In: *Knowledge and Information Systems* 62 (2020), pp. 2419–2437.

[29] Karl Pearson. "LIII. On lines and planes of closest fit to systems of points in space". In: *The London, Edinburgh, and Dublin philosophical magazine and journal of science* 2.11 (1901), pp. 559–572.

[30] Bendi Ramana and N. Venkateswarlu. *ILPD Indian Liver Patient Dataset*. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5D02C. 2012.

[31] Sam T Roweis and Lawrence K Saul. "Nonlinear dimensionality reduction by locally linear embedding". In: *science* 290.5500 (2000), pp. 2323–2326.

[32] Donald B Rubin. "Inference and missing data". In: *Biometrika* 63.3 (1976), pp. 581–592.

[33] Lawrence K Saul and Sam T Roweis. "Think globally, fit locally: unsupervised learning of low dimensional manifolds". In: *Journal of machine learning research* 4.Jun (2003), pp. 119–155.

[34] Indro Spinelli, Simone Scardapane, and Aurelio Uncini. "Missing data imputation with adversarially-trained graph convolutional networks". In: *Neural Networks* 129 (2020), pp. 249–260.

[35] Joshua Tenenbaum. "Mapping a manifold of perceptual observations". In: *Advances in neural information processing systems* 10 (1997).

[36] Joshua B Tenenbaum, Vin de Silva, and John C Langford. "A global geometric framework for nonlinear dimensionality reduction". In: *science* 290.5500 (2000), pp. 2319–2323.

[37] Michael E Tipping. "Sparse Bayesian learning and the relevance vector machine". In: *Journal of machine learning research* 1.Jun (2001), pp. 211–244.

[38] Olga Troyanskaya et al. "Missing value estimation methods for DNA microarrays". In: *Bioinformatics* 17.6 (2001), pp. 520–525.

[39] Athanasios Tsanas and Angeliki Xifara. *Energy efficiency*. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C51307. 2012.

[40] Stef Van Buuren and Catharina GM Oudshoorn. *Multivariate imputation by chained equations*. 2000.

[41] Laurens Van der Maaten and Geoffrey Hinton. "Visualizing data using t-SNE." In: *Journal of machine learning research* 9.11 (2008).

[42] Chr Von der Malsburg. "Self-organization of orientation sensitive cells in the striate cortex". In: *Kybernetik* 14.2 (1973), pp. 85–100.

[43] Yasi Wang, Hongxun Yao, and Sicheng Zhao. "Auto-encoder based dimensionality reduction". In: *Neurocomputing* 184 (2016), pp. 232–242.

Appendix A

Individual BLOBs plot for each model

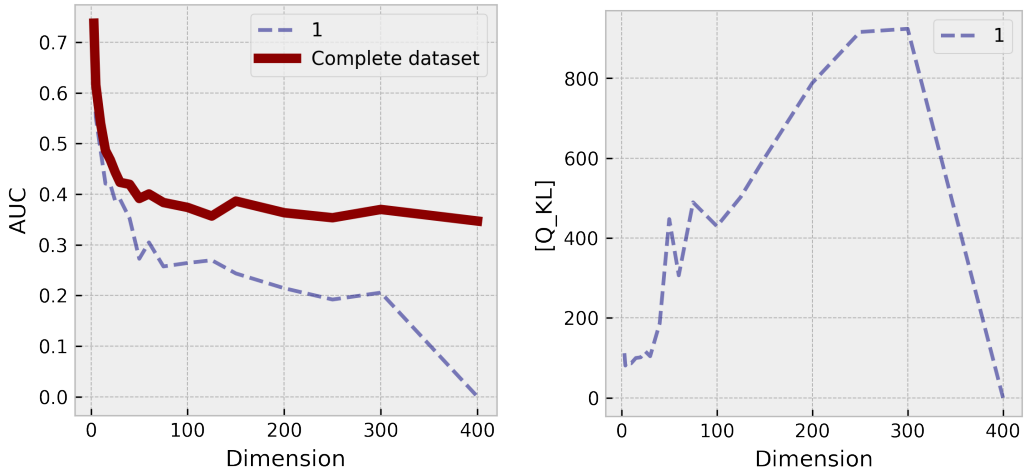

Figure A.1: $[a_{kj}bQ_kd_k]$

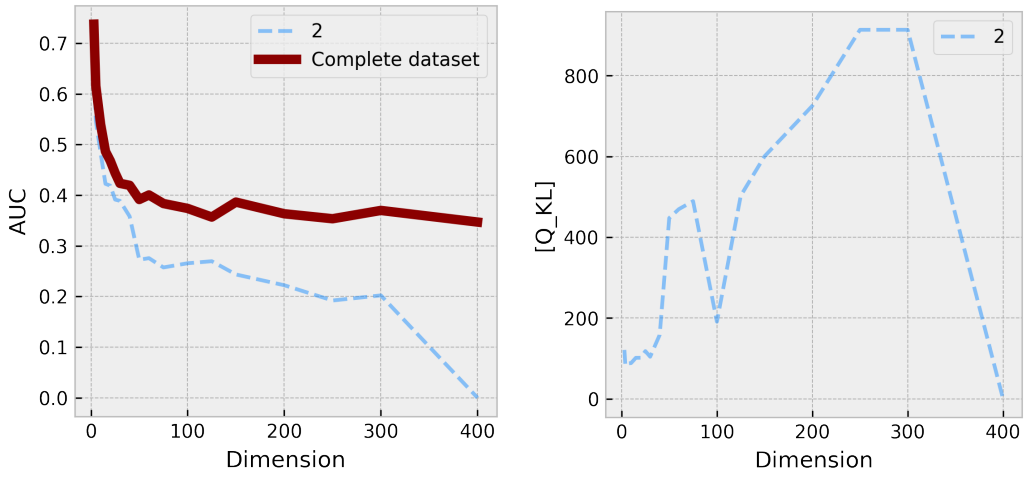

Figure A.2: $[a_kb_kQ_kd_k]$

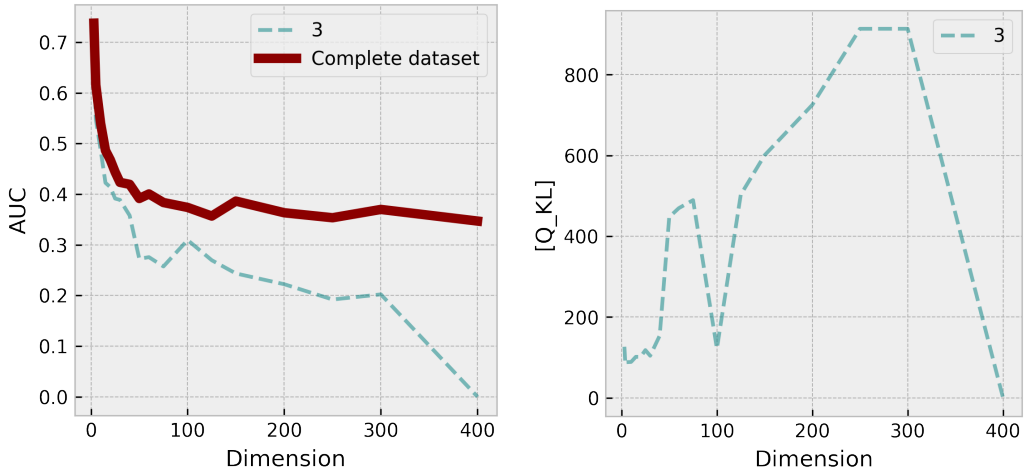

Figure A.3: $[ab_kQ_kd_k]$

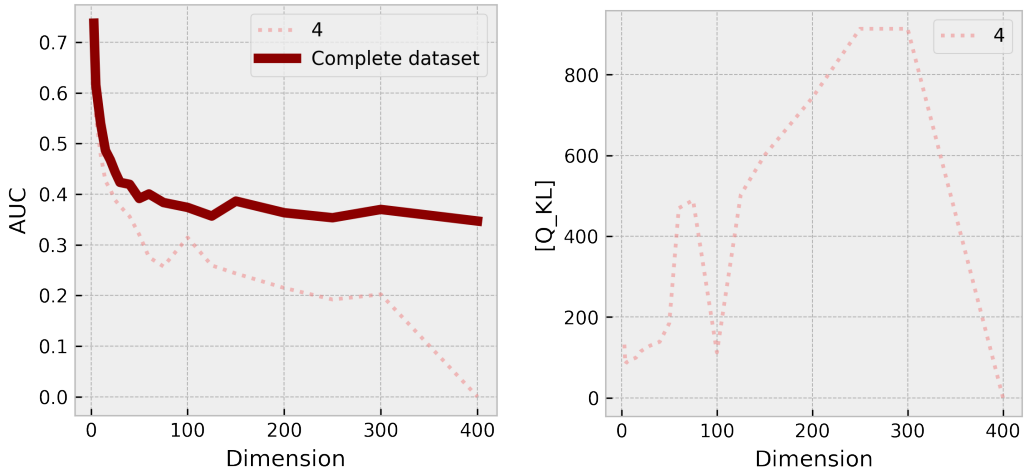

Figure A.4: $[a_{kj}b_kQ_kd]$

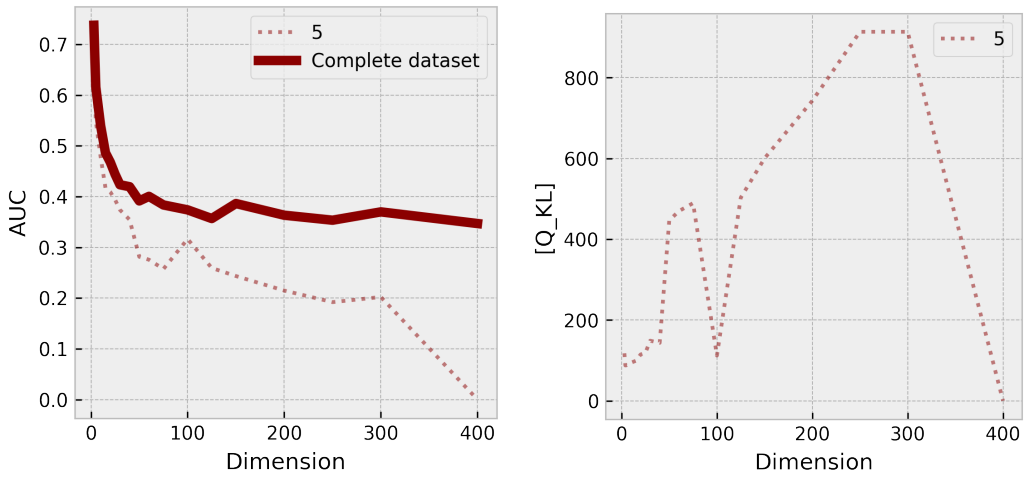

Figure A.5: $[a_jb_kQ_kd]$

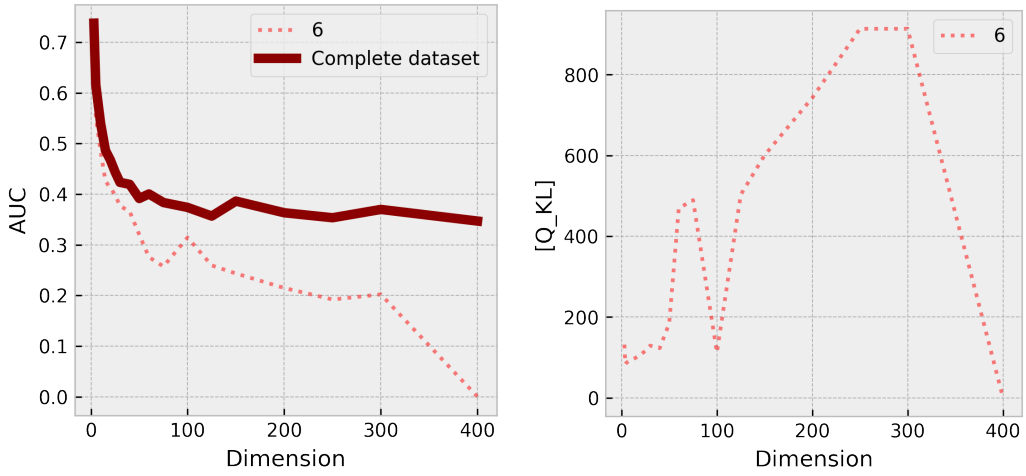

Figure A.6: $[a_k b_k Q_k d]$

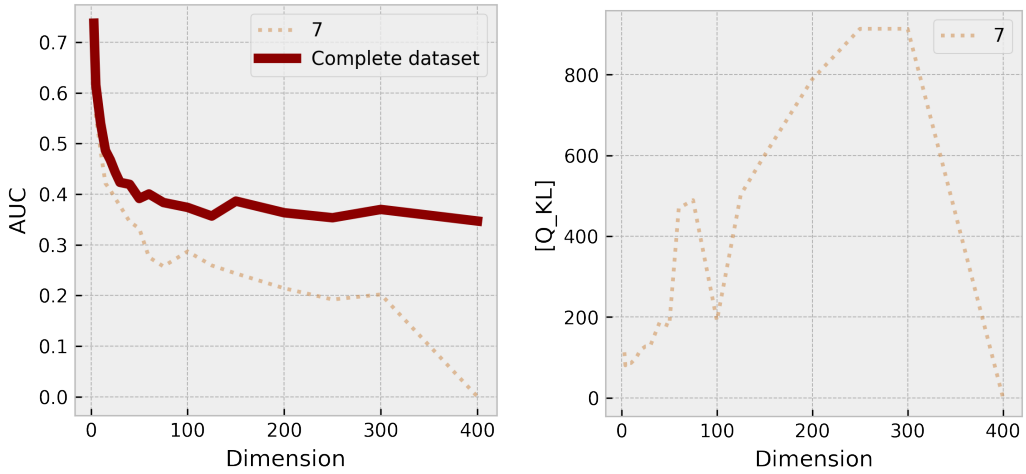

Figure A.7: $[abQ_k d]$

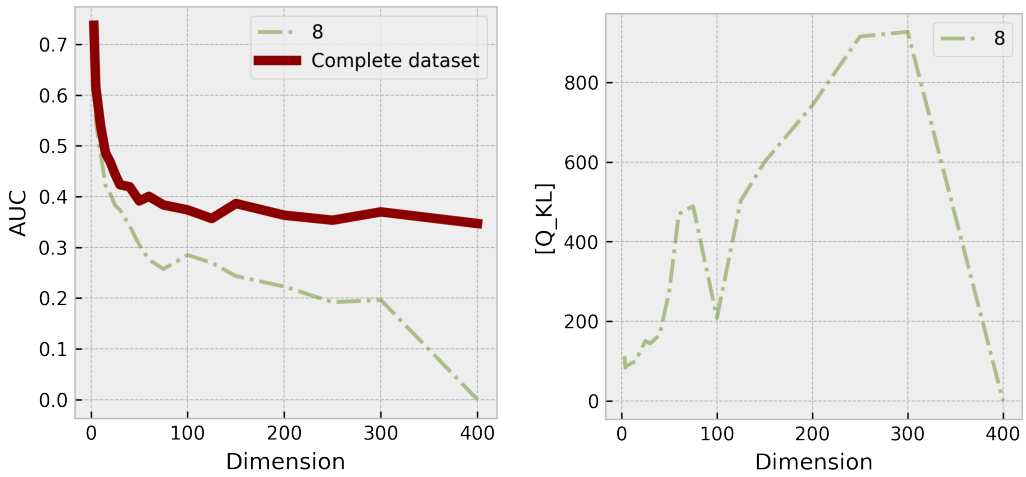

Figure A.8: $[a_k b_k Q d]$

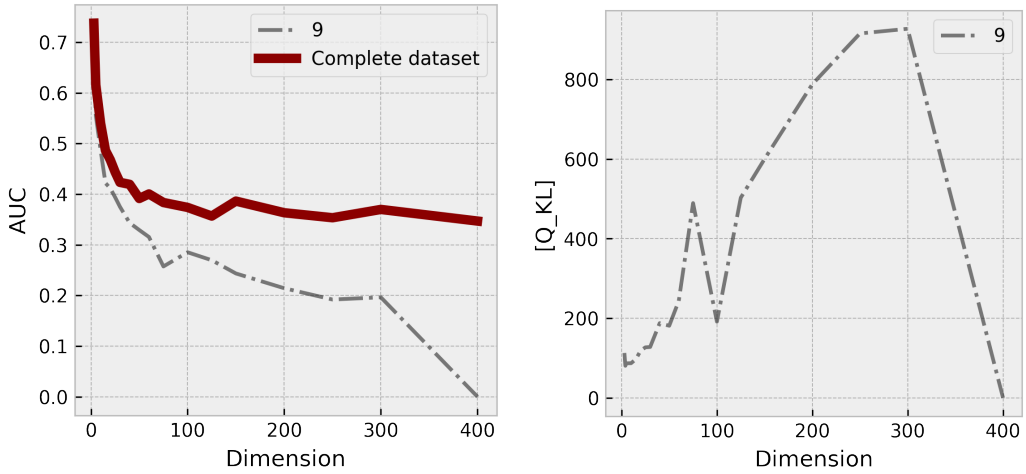

Figure A.9: $[abQd]$

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain
Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | **www.uclouvain.be/epl**