MLPCD, Assignment 1: Curse of Dimensionality

MARIIA HRYTSENKO (261885), mariia.hrytsenko@tu-dortmund.de
NOAH GÖVERT (225862), noah.goevert@tu-dortmund.de
VLADISLAV MALYARCHUK (222742), vladislav.malyarchuk@tu-dortmund.de
CONSTANTIN WEBER (222529), constantin2.weber@tu-dortmund.de

1. Introduction

The "curse of dimensionality" refers to the various phenomena that arise when analyzing and organizing data in high-dimensional spaces.

As the number of dimensions increases, the volume of the space increases exponentially, causing data points to become sparse. This sparsity means that the distance between any two points increases, making it difficult to find meaningful patterns.

Additionally, high-dimensional spaces complicate tasks such as clustering and classification due to the increased computational complexity and the risk of overfitting. Understanding the curse of dimensionality is crucial in fields like machine learning and data mining, where managing high-dimensional data is common.

2. Data Generation

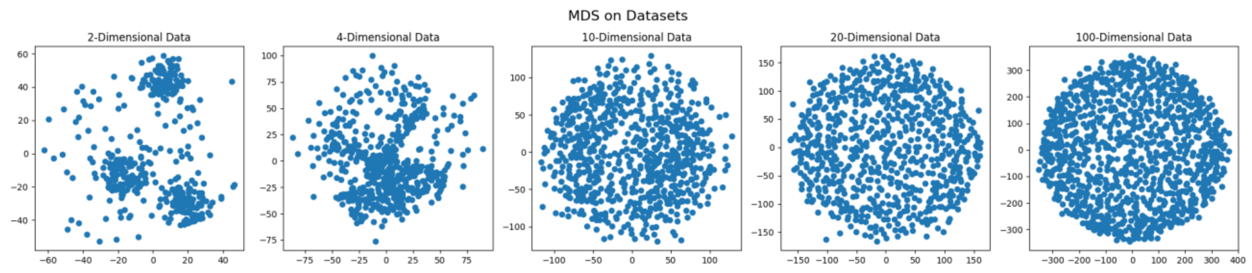
To explore the effects of dimensionality, we generated several synthetic datasets with varying dimensions and characteristics. We generated five datasets with the following characteristics:

1. **2-dimensional data**: 3 clusters, 100 points per cluster, radius of 7, with noise.
2. **4-dimensional data**: 4 clusters, 100 points per cluster, radius of 6, with noise.
3. **10-dimensional data**: 5 clusters, 100 points per cluster, radius of 5, with noise.
4. **20-dimensional data**: 5 clusters, 100 points per cluster, radius of 4, with noise.
5. **100-dimensional data**: 10 clusters, 100 points per cluster, radius of 3, no noise.

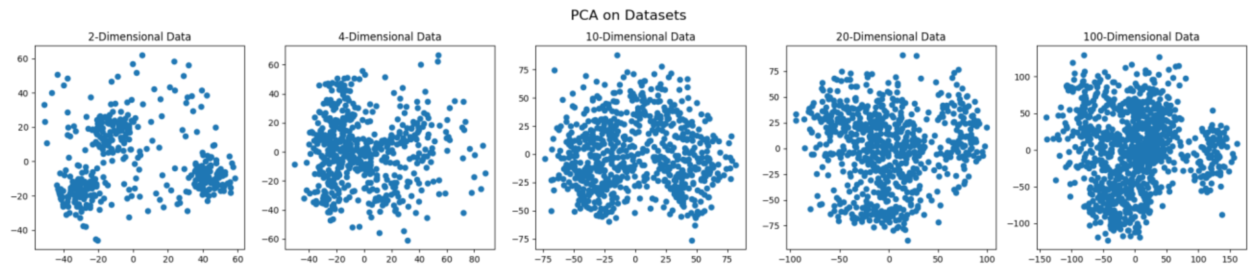
For each dataset, we chose a receding radius for the clusters. This means that the clusters are denser in the center and less dense towards the edges. As the total dimensionality increases, we also increase the amount of clusters in the dataset, as well as the number of noise points.

To better visualize the curse of dimensionality, we will omit the noise points in the final dataset (100 dimensions) completely. As will be seen in the plots below, even then will visualization and clustering be complicated.

To visualize these datasets, we employed two techniques: Multidimensional Scaling (MDS) and Principal Component Analysis (PCA). These techniques reduce the dimensionality of the data, allowing us to plot them in two dimensions.



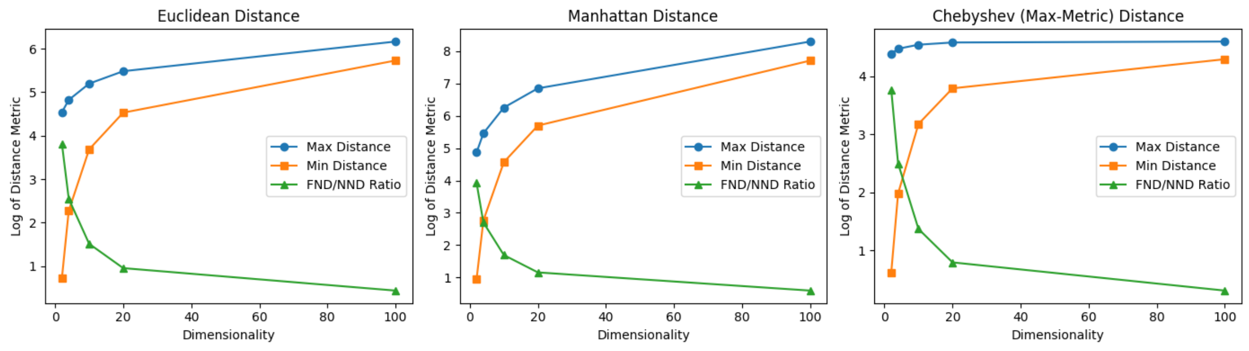
MDS aims to preserve the pairwise distances between points, making it useful for understanding the structure of high-dimensional data.



PCA, on the other hand, transforms the data to maximize variance along the principal components, effectively summarizing the data with fewer dimensions. After observing the stark differences between the multiple visualization techniques, we've opted for PCA in later chapters, as it allows us to show way more differences in high-dimensional datasets than MDS.

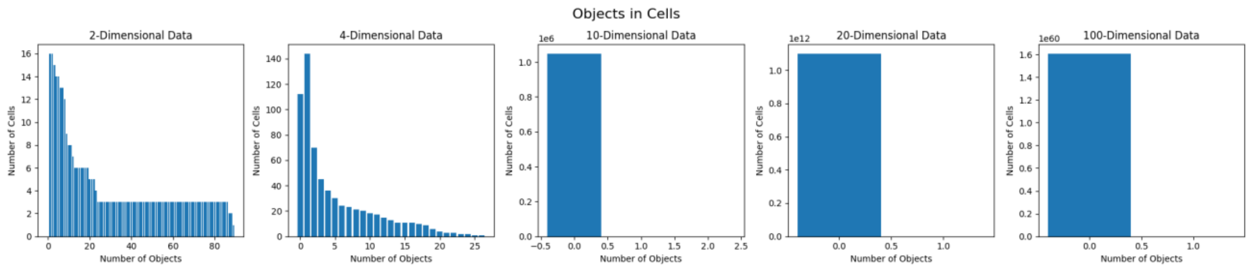
3. Knowledge Discovery

First, we calculated and compared three distance metrics: Euclidean, Manhattan, and Chebyshev distances:



As dimensionality increases, the average distances between points tend to increase. Interestingly, the differences between the metrics became less pronounced in higher dimensions, highlighting the challenges posed by the curse of dimensionality. This occurs because the high-dimensional space becomes sparse, and points are further apart on average.

This sparsity is especially noticeable when partitioning data into multidimensional cells and calculating the number of objects per each cell:

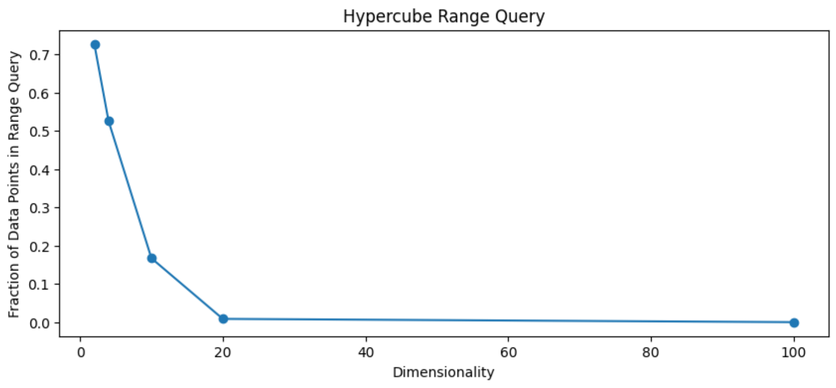


In the 2-dimensional dataset, most cells contain at least one object. Interestingly, absolutely no cells contain 0 objects – the number starts to increase in the 4-dimensional dataset, where about 112 cells (or about 44% of all cells) contain no objects.

Starting from the 10-dimensional dataset onward, the data points are no longer well-separated. Only a few cells cover two objects (only 8 in our dataset), and some also cover one object (692 cells in our dataset). Most cells cover no objects at all (99.99% in our dataset). As the dimensionality increases, the overwhelming majority of cells in these datasets cover absolutely no objects, and absolutely none contain more than one.

This is the curse of dimensionality in action. As the number of dimensions increases, the data points become sparser, and the number of cells covering multiple objects reduces. This makes it harder to cluster the data points.

This is further portrayed when executing a hypercube range query over the dataset. Let us define a length of s = 0.8 for each side of the dataset (normalized to [0, 1]), i.e. 80% of the full length. When performing this range query over the datasets, only the following fractions of the entire dataset are captured:

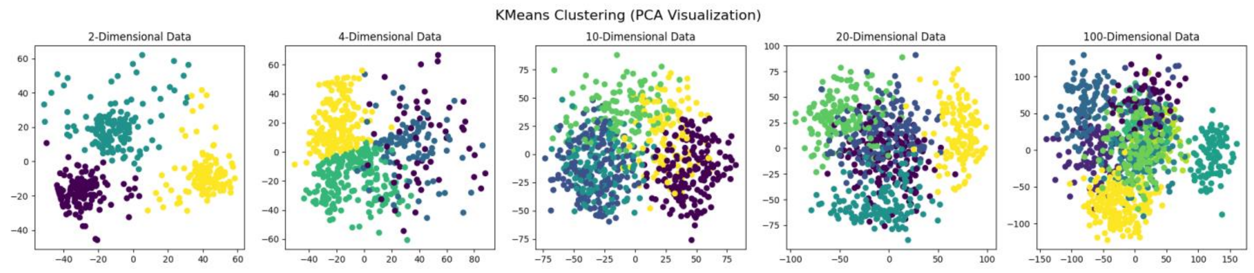


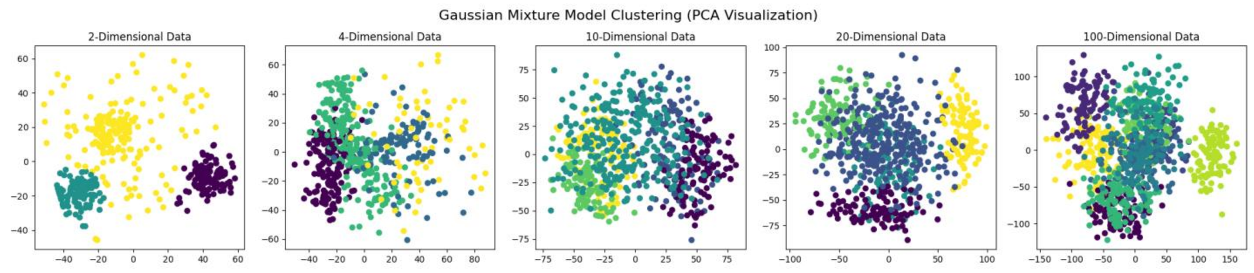
As can be seen, the fraction of objects captured by the query decreases as the number of dimensions increases. This is because the data points become sparser as the number of dimensions increases. In fact, the fraction of objects captured by the query decreases exponentially as dimensionality increases linearly. For example, in the case of 20 dimensions, only 6 objects are captured by the query, i.e. less than 1% of all data points. In the case of 100 dimensions, no objects are captured by the query.

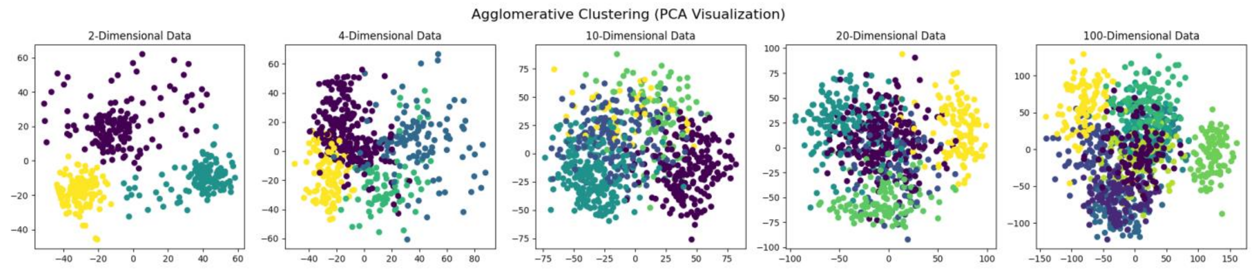
In order to attempt to cluster the objects, we applied three algorithms:

1. **K-Means**: This algorithm partitions data into k clusters by minimizing the within-cluster variance. We chose k based on the known number of clusters in each dataset.
2. **Gaussian Mixture Models (GMM):** This probabilistic model assumes that the data is generated from a mixture of several Gaussian distributions. GMM can model more complex cluster shapes than K-Means. The number of components in GMM was set to match the number of clusters.
3. **Agglomerative Clustering**: This hierarchical method builds clusters by merging pairs of clusters iteratively. Again, we set the number of clusters to match the true cluster count.

The results are portrayed below:







Our observations are that, despite the shortcomings that follow with increasing dimensionality of the datasets, all three algorithms were more or less capable of grouping data into meaningful clusters.

The quality of the clustering does nevertheless seem to drop off for the 100-dimensional data, despite us generating these clusters with smaller radiuses. Were we to generate clusters with even bigger radiuses, it is doubtless that the algorithms would have considerable issue with finding meaningful differences between clusters.

4. Conclusion

Our experiments demonstrated the challenges posed by high-dimensional data, including the curse of dimensionality, increased sparsity, and difficulty in clustering. Visualization techniques like MDS and PCA, along with distance metric analysis, helped us understand these challenges.

Despite the limitations, clustering algorithms are still able to provide useful insights, though their effectiveness varied across datasets. This study highlights the importance of choosing appropriate methods and understanding the inherent difficulties in working with high-dimensional data.