Assignment Report

Assignment 4: Latent Variable Models

Artem Bisliouk (abisliou), Elizaveta Nosova (enosova)

December 8, 2024

1 Probabilistic PCA

a) PPCA Plot

The plots in Figure 1 show 10,000 points generated from the PPCA, where the number of features is 2, the number of latent variables is set to 1, and the noise (σ^2) varies from 0 to 10. The weight vector is represented by an arrow, and remains constant as the noise level increases.

In the absence of the noise, as illustrated in the top left plot in Figure 1, all generated data points are found to lie on the line defined by the weight vector. As the noise rises, the data points begin to spread perpendicularly to the weight vector. The overall shape transitions from a straight line to an ellipse, which becomes wider as σ^2 increases. When $\sigma^2 = 0.5$, as in the original function configuration (the top right plot in Figure 1), the data still align predominantly along the weight vector, but the perpendicular spread is more pronounced.

The bottom plots in Figure 1 illustrate that as the noise value continues to increase, the elliptical shape of the generated data becomes increasingly distorted. The ellipse shrinks in the weight vector's direction, while widening in other directions. When $\sigma^2 = 10$, as seen in the bottom right plot of Figure 1, the generated data becomes almost circular in shape, and the relationship with the weight vector is no longer discernible.

b) MLE for PPCA

The calculation of σ_{MLE} employs the formula in Equation (1)

$$\sigma^2_{\text{MLE}} = \frac{1}{D-L}\sum_{j=L+1}^{D} \lambda_j \tag{1}$$

Since for `toy_ppca` D is set to 2, when L equals 2 as well, there are no λ_j to be included in the sum, and as a consequence, no elements to average (D - L = 0), as a consequence,

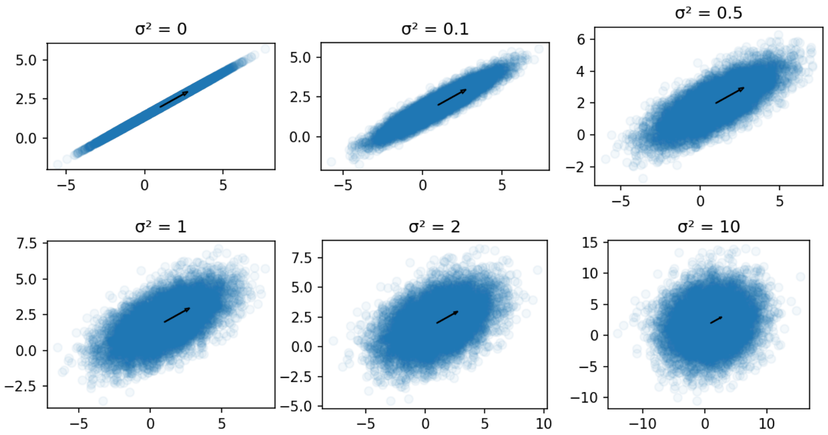

Figure 1: PPCA generated data points with varying amount of noise

$\sigma_{MLE} = 0$. This outcome is expected as σ_{MLE} is meant to describe the average variance of discarded dimensions. When L = D, no dimensions have been discarded.

c) Conditional Negative Log-Likelihood: Implemented in the Jupiter notebook.

d) Number Latent Variables

The scree plot of the negative log-likelihood as a function of the number of latent variables is presented in Figure 2 on the left. Although the interpretation of the scree plots frequently can be subjective, in this case we clearly see a point of maximal curvature at L = 20.

For another approach, 75% of the data was used as a train split, while 25% was held out for validation. On the right plot in Figure 2 we can see the negative log-likelihood on the validation set as a function of L. It can be distinctly seen that the minimal negative log-likelihood value is achieved at L = 20.

Therefore, both the scree plot and the maximizing negative log-likelihood on the validation set coincide that 20 latent variables have been used.

2 Gaussian Mixture Models

a) GMM Parameters and Cluster Characteristics

The probability density function for Gaussian Mixture Model (GMM) which can be viewed as a ”soft clustering” is defined as:

$$p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x \mid \mu_k, \Sigma_k),$$

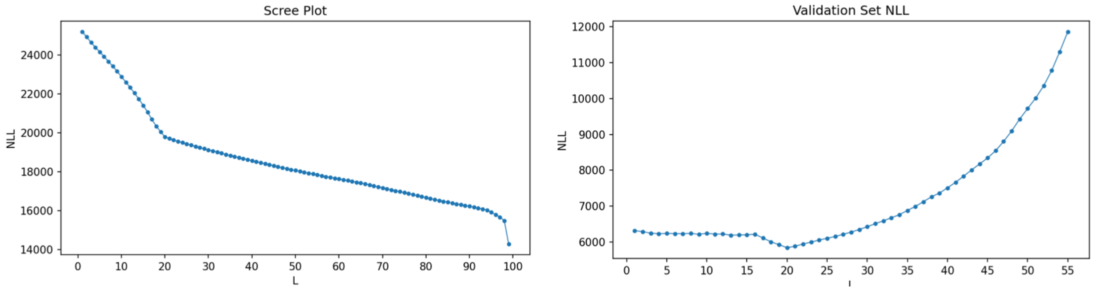

Figure 2: The scree plot (left) and negative log-likelihood of validation data (right)

where we have the following set of parameters:

- π_k is the mixture weight for the k-th component, satisfying $\sum_{k=1}^{K} \pi_k = 1$ and $\pi_k \geq 0$. It determines the relative size of the clusters (proportion of points belonging to each component),
- μ_k is the mean vector defining the center of the k-th Gaussian component. As a consequence, it defines the location of the cluster center in the feature space,
- Σ_k is the covariance matrix, defining the spread and orientation of the k-th Gaussian and thus it controls the shape of the cluster.

As it shown on Figure 2 means μ determine the spatial locations of the clusters, and their positions in the plot align with the specified μ values in the code. The light blue cluster, located at $[0, 0]$, represents the central cluster. The blue and light green clusters, centered at $[10, 0]$ and $[-10, 0]$, respectively, extend symmetrically along the horizontal axis. Similarly, the green and red clusters, positioned at $[0, 10]$ and $[0, -10]$, occupy vertical symmetry around the origin.

The mixing coefficients π define the relative sizes of the clusters, which are evident in their densities and proportions in the plot. The red cluster at $[0, -10]$, with the highest $\pi = 0.35$, dominates the plot with the largest size and density. The blue cluster at $[10, 0]$, corresponding to $\pi = 0.2$, is smaller but still prominent. Similarly, the light green cluster at $[-10, 0]$, with $\pi = 0.25$, appears slightly larger than the blue cluster, as expected. The green and light blue clusters, both with $\pi = 0.1$, are noticeably smaller and less dense, emphasizing their lower mixing proportions. This distribution showcases how π directly controls the number of points assigned to each cluster, significantly influencing their visual prominence.

The covariance matrices Σ govern the shape and spread of the clusters, creating the varying geometries observed in the plot. The elongated form of the red cluster at $[0, -10]$ indicates a covariance matrix with a higher variance along one axis, suggesting an anisotropic distribution. In contrast, the compact circular shapes of the light blue cluster at $[0, 0]$ and the blue cluster at $[10, 0]$ imply nearly isotropic covariance matrices, with similar variances in all directions. The light green cluster at $[-10, 0]$ displays moderate elongation, hinting at unequal variances along its principal axes. Lastly, the

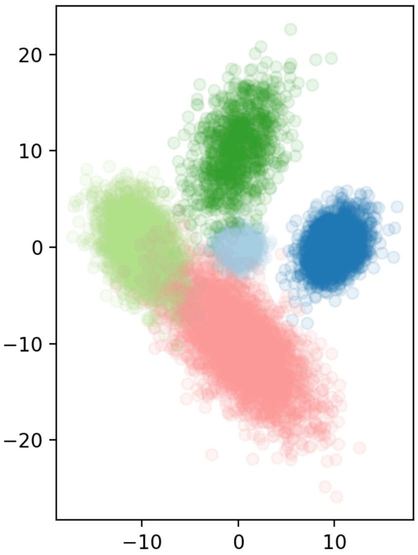

Figure 3: Generated GMM dataset with five clusters. Colors represent distinct clusters as described in the text.

green cluster at $[0, 10]$ exhibits a shape similar to the red cluster but with a smaller size, indicating slightly higher variance along one axis. These observations highlight the impact of Σ on cluster geometry, influencing both the orientation and spread of each cluster, thus adding complexity to the dataset structure.

b) K-Means Clustering

The K-Means clustering results (Figure 4) differ significantly from the GMM clustering (Figure 3) in both methodology and outcomes. K-Means uses hard cluster assignments, where each data point is assigned to a single cluster, resulting in non-overlapping clusters. In contrast, GMM employs soft clustering, assigning membership probabilities for each point to all clusters. This distinction highlights that K-Means can be seen as a specific case of GMM that uses a simplified Expectation-Maximization (EM) algorithm. In this ”hard” version of EM, each data point is assigned entirely to the most likely cluster (i.e., cluster membership probability $w_{ik} = p(z_i = k \mid \mathbf{x}_i, \boldsymbol{\theta}^{(t)})$ is set to 1 for the most probable cluster and 0 for all others). For example, in the GMM plot, points near the boundaries between clusters, such as the light blue and green clusters, have overlapping regions, reflecting uncertainty in cluster membership. K-Means, however, forces a strict boundary, such as the clear separation of the blue and red clusters, regardless of the actual distribution of points.

Cluster sizes in K-Means are relatively uniform compared to GMM. This uniformity

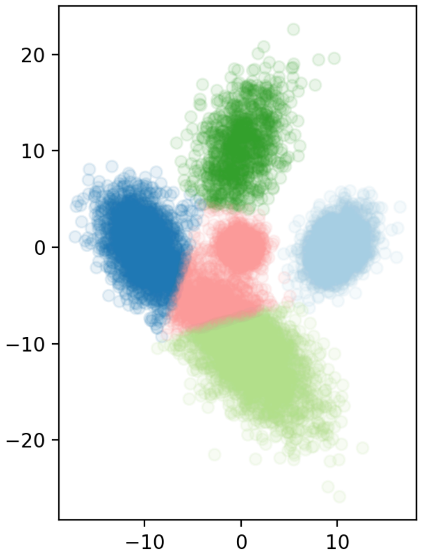

Figure 4: K-Means clustering with five clusters. Colors represent hard cluster assignments determined by K-Means.

arises because K-Means minimizes the total sum of squared residuals (SSR), effectively forcing each cluster to share an equal proportion of the dataset. This behavior corresponds to $\pi_k = 1/K$ in GMM with hard EM algorithm, where clusters are equally likely. In contrast, the GMM plot shows variation in cluster sizes driven by mixing coefficients π. For example, the red cluster in GMM, with $\pi = 0.35$, is much larger than the light blue cluster with $\pi = 0.1$. In the K-Means result, the red cluster is more comparable in size to the other clusters, indicating a forced uniformity.

K-Means assumes spherical clusters, akin to setting $\Sigma_k = I$ in GMM, where each Gaussian component is isotropic. This constraint results in K-Means clusters that are roughly circular in shape, as observed in the plot (e.g., the red and green clusters). In contrast, GMM allows for flexibility in cluster shapes and orientations based on covariance matrices Σ. For example, the elongated shape of the red cluster in GMM (Figure 3) suggests higher variance along one axis, which is not captured by the K-Means result. Similarly, the light green cluster in GMM shows slight elongation, whereas K-Means forces it into a spherical form.

In summary, the comparison highlights key differences between K-Means and GMM clustering. K-Means simplifies clustering with hard assignments, uniform cluster sizes, and spherical shapes, while GMM captures the inherent complexity of the data through soft assignments, varied sizes, and flexible shapes. These differences make GMM more suitable for capturing nuanced patterns in data with varying densities and distributions, as seen in the original dataset.

d) Comparison of GMM and K-Means Clustering (K=5)

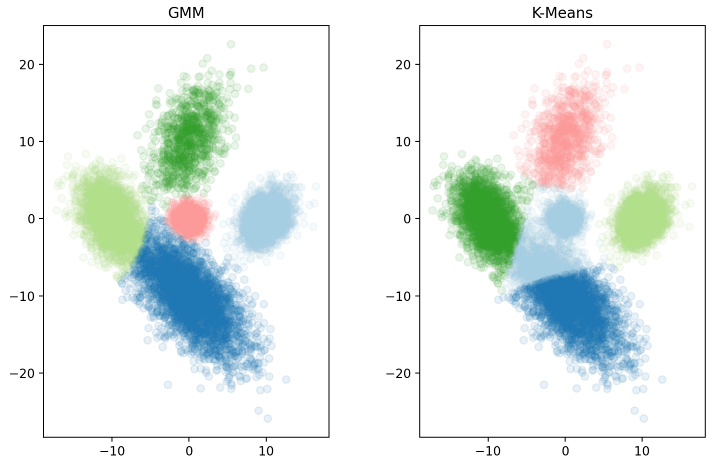

Figure 5: Comparison of GMM and K-Means clustering results for $K = 5$. Left: GMM clustering. Right: K-Means clustering.

The clustering results for $K = 5$ reveal key differences between the GMM and K-Means approaches, as shown in Figure 5. These differences stem from the fundamental distinctions in how the two algorithms assign points to clusters and model cluster structures. While GMM uses a probabilistic model with soft cluster assignments, K-Means relies on hard cluster assignments based on proximity to centroids, leading to sharp differences in cluster boundaries, sizes, and shapes.

In the GMM plot, clusters exhibit smooth and natural boundaries due to the probabilistic nature of assignments. Each data point is labeled based on the most likely component ($\arg\max W[i, k]$), which allows for overlapping and flexible decision boundaries. For example, the green cluster red cluster share a smooth transition region, reflecting uncertainty near their boundary. This behavior arises from the Expectation-Maximization (EM) algorithm's E-step, where responsibilities (W) for each point are computed based on the likelihood under each Gaussian component. In contrast, K-Means imposes hard boundaries, as evident in the sharp division between the blue and the light blue clusters, ignoring the true distribution of edge points.

Cluster sizes also differ significantly between the two methods. GMM captures the variability in the dataset by adapting the mixing coefficients (π) and covariance matrices (Σ) of the Gaussian components. This flexibility is evident in the elongated blue cluster, which spans a wider range along one axis, and the compact red cluster. In contrast, K-Means enforces uniform cluster sizes by minimizing the total sum of squared residuals (SSR), which distributes points more evenly across clusters.

The shapes of the clusters further highlight the flexibility of GMM. Gaussian components in GMM are defined by their covariance matrices, enabling elliptical boundaries that can capture varying spreads and orientations. This is particularly visible in the curved boundary between the blue cluster and the red cluster. K-Means, on the other hand, assumes isotropic cluster shapes, resulting in spherical clusters with straight-line boundaries. This assumption fails to capture the true structure of the dataset, as seen in the constrained shape of the green cluster in the K-Means plot.

In conclusion, GMM clustering provides a more flexible and accurate representation of the data compared to K-Means. By incorporating probabilistic assignments, elliptical boundaries, and varying cluster sizes, GMM captures the true underlying structure of the dataset. K-Means, while computationally efficient, imposes stricter assumptions, leading to less nuanced boundaries and more uniform clusters. This comparison underscores the advantages of GMM in scenarios with overlapping or non-spherical clusters, as demonstrated by the dataset in Figure 5.

e) GMM and K-Means Clustering Comparison (K=4 and K=6)

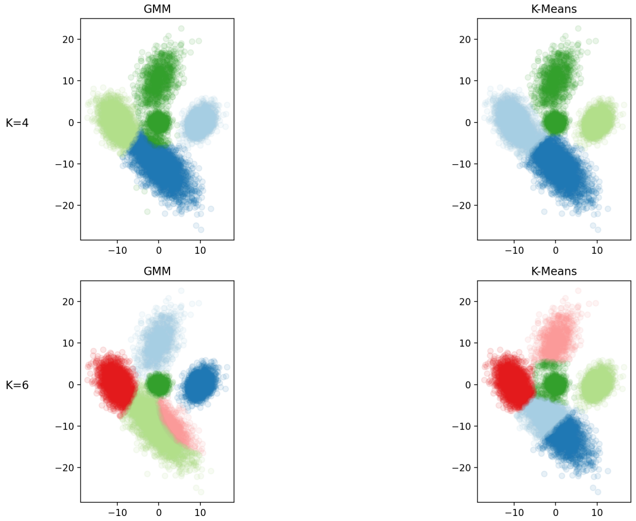

Figure 6: Comparison of GMM and K-Means clustering results for $K = 4$ and $K = 6$. Top row: $K = 4$ (Left: GMM, Right: K-Means). Bottom row: $K = 6$ (Left: GMM, Right: K-Means).

The clustering results for $K = 4$ and $K = 6$ (Figure 6) demonstrate the impact of changing the number of clusters on the behavior of GMM and K-Means. These results reveal key differences in how each algorithm handles merging or splitting clusters and how the data is partitioned.

For $K = 4$, GMM merges clusters that were distinct for $K = 5$. This creates broader clusters that span larger regions of the dataset, with smooth and probabilistic boundaries. For example, the green cluster now encompasses points that were part of two distinct clusters in $K = 5$. The merging behavior in GMM reflects the probabilistic nature of the EM algorithm, which adapts to maximize the likelihood of the data by combining clusters with overlapping densities. In contrast, K-Means with $K = 4$ produces hard, linear boundaries, merging the cluster near the origin. This creates an elongated, artificial-looking cluster that fails to capture the true distribution of points.

For $K = 6$, GMM introduces an additional, smaller cluster, visible as a distinct light red cluster near the center of the dataset. This new cluster primarily captures outliers or noise points that were probabilistically assigned to larger clusters in $K = 5$. The remaining clusters retain their shapes and sizes, with smooth decision boundaries. In K-Means, however, the additional cluster results in a strict partitioning of the dataset, forcing the origin cluster to split into two smaller clusters. This leads to linear, parallel boundaries that divide the dataset evenly but lack the flexibility of GMM's elliptical boundaries.

The behavior of GMM across different K values highlights the adaptability of the EM algorithm. For fewer clusters ($K = 4$), GMM merges Gaussian components with overlapping densities to maintain model simplicity. For more clusters ($K = 6$), it introduces additional components to capture finer details or sparsely distributed points. This flexibility ensures that GMM clusters remain consistent with the true data distribution, as seen in the retention of smooth, curved boundaries and the ability to handle outliers.

K-Means, by contrast, rigidly partitions the dataset into equally sized, isotropic clusters regardless of the data's underlying structure. This is particularly evident for $K = 6$, where the addition of a sixth cluster results in a loss of coherence for the origin cluster, which is split into two. The lack of consideration for density or orientation in K-Means often leads to artificial boundaries that fail to reflect the natural relationships in the data.

In conclusion, the comparison across $K = 4$, $K = 5$, and $K = 6$ demonstrates the superior flexibility and robustness of GMM in capturing complex data distributions. While K-Means is computationally simpler, its reliance on linear boundaries and uniform cluster sizes limits its ability to model nuanced patterns, as evidenced by the results in Figure 6.

Ehrenwörtliche Erklärung

Ich versichere, dass ich die beiliegende Bachelor-, Master-, Seminar-, oder Projektarbeit ohne Hilfe Dritter und ohne Benutzung anderer als der angegebenen Quellen und in der untenstehenden Tabelle angegebenen Hilfsmittel angefertigt und die den benutzten Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen. Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben wird.

Declaration of Used AI Tools

Tool	Purpose	Where?	Useful?
ChatGPT	Rephrasing	Throughout	+
DeepL	Style Edits	Throughout	++
GPT-4	Code debugging	Throughout	+-

Unterschrift
Mannheim, den 8. Dezember 2024