Report - Machine Learning (HWS 2024)

Assignment 4: Latent Variable Models

Arne Huckemann (ahuckema), Elise Wolf (eliwolf)

December 8, 2024

.

1. Introduction

This report investigates two fundamental techniques in probabilistic modeling: Gaussian Mixture Models and Probabilistic Principal Component Analysis. For GMMs, we examine clustering and density estimation by implementing both the Expectation-Maximization algorithm and K-Means clustering. PPCA analysis focuses on variance decomposition and model fitting through Maximum Likelihood Estimation. Both methods are evaluated using synthetic datasets, with emphasis on computational accuracy and interpretability of results. Additionally, the analysis aims to uncover latent structures and components, offering insights into the behavior and effectiveness of these techniques in capturing underlying data distributions.

2. Task 1: Probabilistic PCA

2.1. Task 1a: Toy Data

The `ppca_gen` function is designed to simulate data that follows the Probabilistic Principal Component Analysis (PPCA) model. It requires several key parameters, including the number of data points N, the dimensionality of the observed data D, the number of latent variables (principal components) L, the noise variance σ^2, and the mean of the data μ. If not explicitly specified, the function uses default values for μ, the eigenvalues λ_L (variances along the principal components), and the orthogonal eigenvector matrix Q (principal components). The outputs of the function include the weight matrix W, noise Eps sampled from a standard normal distribution scaled by $\sqrt{\sigma^2}$, and the observed data points X, which are calculated as

$$X = Z \cdot W^\top + \mu + \text{Eps},$$

where Z represents the latent variables.

A scatter plot generated by the function displays 10,000 data points, providing a visual representation of the probabilistic PCA structure (Figure 1). With a noise variance of $\sigma^2 = 0.5$, the data points form a diagonal distribution along the weight vector $[1.749, 0.970]$ corresponding to the first principal component. In this configuration, the data points primarily fall within the range $[-2, 4]$ on the x-axis and $[0, 4]$ on the y-axis. This diagonal alignment reflects the underlying structure captured by the PPCA model, while the noise introduces some spread around the principal component.

The influence of noise variance σ^2 on the distribution of data points is clearly observable (Figure 2 and 3). When the noise variance is reduced to $\sigma^2 = 0.1$, the data points become more tightly clustered around the weight vector, resulting in a narrower distribution. This is evident as the points are concentrated within $[-2, 6]$ on the x-axis and $[0, 4]$ on the y-axis. The reduction in noise variance decreases the randomness in the data, making the alignment along the principal component more pronounced and visually distinct. Conversely, increasing the noise variance leads to a broader spread of data points, diminishing the clarity of the diagonal distribution and making the underlying structure less apparent.

2.2. Task 1 b: Maximum Likelihood Estimation

When implementing Maximum Likelihood Estimation (MLE) for Probabilistic Principal Component Analysis (PPCA), setting $L = D$ (where $L = 2$ and $D = 2$ in this case) results in $\sigma^2_{\text{MLE}} = 0$. This occurs because there are no residual dimensions between L and D to account for any variance not captured by the first L eigenvalues. Consequently, the variance of the discarded dimensions, σ^2_{MLE}, becomes zero.

This reflects that the model explains all the variance in the data through the latent dimensions alone, leaving no additional variance to attribute to noise. Formally, σ^2_{MLE} is computed as:

$$\sigma^2_{\text{MLE}} = \frac{1}{D - L} \sum_{i=L+1}^{D} \lambda_i,$$

where λ_i are the eigenvalues of the covariance matrix of the centered data. When $L = D$, the summation is over an empty set, leading to:

$$\sigma^2_{\text{MLE}} = 0.$$

Thus, the model fully captures the data's variance through its principal components, leaving no remaining variance to assign to noise.

2.3. Task 1c: Negative Log-Likelihood

To compute the Negative Log-Likelihood for a given Probabilistic Principal Component Analysis model and dataset, we begin by defining the covariance matrix C. The covariance matrix is expressed as

$$C = WW^T + \sigma^2 I,$$

where W is the weight matrix, σ^2 is the variance of the isotropic noise, and I is the identity matrix.

The next step involves calculating the log-determinant of the covariance matrix C. This can be computed as

$$\log |C| = \log(\text{np.linalg.det}(C)),$$

where the determinant is computed numerically using appropriate linear algebra routines.

Additionally, we define the empirical covariance matrix S as

$$S = \frac{1}{N}\sum_{n=1}^{N}(x_n - \mu)(x_n - \mu)^T = \frac{1}{N}X_{\text{centered}}^T X_{\text{centered}},$$

where N is the number of data points, μ is the data mean, and X_{centered} represents the mean-centered data matrix.

Finally, the Negative Log-Likelihood (NLL) is given by

$$\text{NLL} = \frac{N}{2}\left(D\log(2\pi) + \log|C| + \text{tr}(C^{-1}S)\right),$$

where D is the dimensionality of the data. This formula is derived from the reference: *Probabilistic Machine Learning: An Introduction* (Online version, November 23, 2024), ensuring a rigorous and consistent computation.

2.4. Task 1d: Discover the Secret!

In this section, we determine the optimal number of latent variables (L) used in the probabilistic PCA model that generated the secret dataset. This analysis is performed using two approaches: (i) studying the scree plot (Figure 4) and (ii) utilizing validation data (Figure 5). The results from both methods are then compared.

(i) Studying the Scree Plot:

The scree plot provides a visual representation of the eigenvalues of the covariance matrix in descending order. It is a standard tool for identifying the number of principal components that capture most of the variance in the dataset. Typically, the plot exhibits a steep decline followed by a plateau, with the "elbow" or kink indicating the optimal number of latent variables.

In the scree plot for this dataset, we observe a pronounced kink between the 20th and 21st eigenvalues. This suggests that the first 20 principal components effectively capture the majority of the variance in the data, with minimal gains in variance explained by adding components beyond this point.

(ii) Using Validation Data:

To further verify the number of latent variables, we used a validation-based approach. The dataset was split into training and validation subsets. For varying values of L, we computed the negative log-likelihood (NLL) on the validation set. The NLL serves as a measure of model fit, where lower values indicate a better fit.

The plot of NLL versus the number of latent variables reveals that the NLL reaches its minimum at $L = 20$. This confirms that the model with 20 latent variables achieves the best trade-off between model complexity and fit to the data.

Comparison of Results:

Both methods, the scree plot and validation data, converge on the same result: the optimal number of latent variables is $L = 20$. This consistency between methods reinforces the reliability of the conclusion.

Based on the scree plot and validation data, we conclude that the optimal number of latent variables (L) for the PPCA model is 20. This finding ensures that the model captures the underlying structure of the data without unnecessary complexity.

3. Task 2: Gaussian Mixture Models

3.1. Task 2a: Toy data

The generated toy dataset, visualized in Figure 6, reveals five distinct clusters. Each cluster exhibits unique characteristics in terms of size, orientation, and density. Below, we describe the properties of each cluster in detail:

1. Pink Cluster:

This cluster spans approximately between -10 and 10 on the x-axis and -2 to -20 on the y-axis. It is elongated diagonally from the top-left to the bottom-right. Representing 35% of the total data points, it is the largest and densest cluster, dominating the overall structure of the dataset.

2. Light Green Cluster:

Located between -15 and -5 on the x-axis and -7 to 7 on the y-axis, this cluster aligns directionally with the pink cluster, giving the impression of an extension. It is slightly less dense than the pink cluster and accounts for 25% of the dataset.

3. Light Blue Cluster:

This cluster is centered near the origin ($[0, 0]$) with a radius of approximately 3. Its structure is circular and isotropic, distinguishing it from the other clusters. Representing only 10% of the data points, it is the smallest cluster.

4. Dark Blue Cluster:

Positioned between 5 and 15 on the x-axis and -5 to 5 on the y-axis, this cluster is diagonally oriented from the bottom-left to the top-right. It is slightly more compressed than the light green cluster and contributes 20% of the data points.

5. Dark Green Cluster:

The core of this cluster lies between -5 and 5 on the x-axis and 5 to 15 on the y-axis. It is vertically elongated but slightly rotated, following a bottom-left to top-right diagonal. This cluster is the most spread-out and least dense, representing 10% of the data points.

The generated dataset exhibits significant diversity in cluster characteristics, as evident from the plot. The pink cluster, being the largest and densest, dominates the visual layout, while the light blue cluster stands out as the smallest and most compact. The light green and dark blue clusters provide intermediate density levels, each with distinct directional orientations. Finally, the dark green cluster adds further complexity by being the most dispersed and least dense. Together, these clusters demonstrate the versatility of the GMM in generating structured but varied data distributions.

3.2. Task 2b: K-Means

The performance of K-Means clustering was evaluated on data generated from a toy Gaussian Mixture Model (GMM) with $K = 5$ as seen in Figure 7. The purpose was to analyze how well K-Means can replicate the structure of the original GMM, which is known for its more flexible, covariance-driven cluster boundaries.

Clusters That Are Similar:
Notably, the dark blue and light green clusters remain nearly unchanged between the GMM and K-Means plots. This consistency can be attributed to the distinct means and well-separated structures of these clusters in the original GMM, which K-Means is capable of handling effectively due to the clear separation between centroids.

Clusters with Significant Differences:
A substantial difference is observed in the light blue cluster of the K-Means plot. This cluster, which originates from the GMM's pink and light green clusters, extends from the center towards the bottom left. This merging indicates that K-Means has shifted the centroid to an overlapping region, resulting in a reclassification of points that originally belonged to separate clusters.

The pink cluster in the K-Means plot is notably smaller. This reduction is due to the reassignment of its data points to the new, larger light blue cluster. Additionally, the boundary between the original pink and light blue clusters has transformed from an elliptical shape to a linear, horizontal decision boundary. This change is a direct consequence of K-Means' reliance on straight-line partitions between centroids, as opposed to GMM's probabilistic model, which supports non-linear, covariance-based boundaries.

Boundary Adjustments:
A clear distinction is seen with the diagonal decision boundary that separates the original pink and light blue clusters, running from the top right to the bottom left. This sharp boundary leads to the inclusion of overlapping points from both the original clusters into the light blue cluster. Conversely, the boundary of the original light green cluster remains almost intact, highlighting its strong density and distinct separation from other clusters, which limits significant point reassignment.

The most striking difference between the clusters stems from K-Means' simplistic method of assigning points based on Euclidean distance to centroids, resulting in linear boundaries. In contrast, GMM's probabilistic approach accommodates elliptical or non-linear shapes, aligning with the data's inherent distribution.

Connection Between Code and K-Means Results:
The initial centroids for K-Means are initialized using the GMM means (μ). However, as K-Means iterates, centroids adjust to minimize intra-cluster variance. This shift explains the expansion of the light blue cluster, as its centroid moves toward denser overlapping regions, causing a reassignment of data points that originally belonged to the pink and

light green clusters. The reduced size of the pink cluster reflects this reallocation, resulting from K-Means' inability to model complex cluster shapes effectively.

While the GMM's mixture proportions (π) influence the initial distribution of cluster densities, K-Means does not incorporate these proportions explicitly. Instead, the final cluster sizes are determined by point proximity and density relative to the centroids. This absence of an explicit density factor results in differences in cluster sizes.

GMM's covariance matrices (Σ) provide flexibility for elliptical cluster boundaries. In contrast, K-Means imposes rigid, linear decision boundaries. This limitation manifests as artificial splits, such as the diagonal and horizontal boundaries observed in the plots, which do not align with the natural data distribution.

3.3. Task 2d+e: Experiment with GMMs for the toy data

The GMM clustering results using $K = 5$ clusters reveal a significant enhancement in capturing the data's underlying structure compared to K-Means (Figure 8. The GMM leverages a probabilistic approach with elliptical decision boundaries, allowing for more nuanced and flexible cluster assignments. This approach contrasts with the K-Means algorithm, which relies on linear boundaries determined by Euclidean distance between points and centroids.

The clustering results from the GMM are obtained through the Expectation- Maximization (EM) algorithm, which iteratively optimizes the parameters of the Gaussian components. This algorithm's role in shaping the final clustering structure can be understood through its key steps:

1. Initialization:

The EM algorithm initializes the means (μ), covariances (Σ), and mixture proportions (π) of the Gaussian components. These initial parameters, often set heuristically or randomly, determine the starting cluster assignments and influence the initial shape of the clustering.

2. E-Step (Expectation):

The algorithm computes the responsibility matrix, indicating the probability of each data point belonging to each Gaussian component. This step allows for soft assignments, where points can have partial affiliations to multiple clusters. The effect of this is evident in the smooth, curved boundaries observed in the GMM plot (e.g., the boundary between Clusters 4 and 5).

3. M-Step (Maximization):

The parameters (μ, Σ, and π) are updated to maximize the likelihood of the data, given the responsibilities computed in the E-step. The means shift towards the weighted average of the points, covariances adjust to reflect the spread and orientation of the data, and proportions update to indicate relative cluster sizes. This leads to more refined cluster shapes (e.g., the larger Cluster 5 and the smaller Cluster 1).

4. Iterative Refinement:

The EM algorithm alternates between the E-step and M-step until convergence. This iterative process results in highly refined cluster boundaries that adapt to even subtle data patterns, such as the curved boundary for Cluster 4 and the inclusion of edge points in Cluster 3.

5. Probabilistic Nature of Assignments:
The EM algorithm's probabilistic approach contrasts with the deterministic assignment of K-Means. This feature is evident in the smoother transitions between clusters in the GMM plot, in contrast to the abrupt boundaries of K-Means.

Comparison Between GMM and K-Means Results
GMM's probabilistic nature and use of covariance matrices allow for smooth and elliptical boundaries that align more accurately with the data's distribution. In contrast, K-Means, constrained by linear partitions, results in artificial, sharp boundaries. Additionally, the GMM assigns points with varying probabilities, enabling partial memberships, which leads to more natural boundary transitions. K-Means, by assigning points solely to the nearest centroid, lacks this flexibility.

The GMM with $K = 5$ produces clusters that better reflect the data's true density and spread. For instance, Cluster 1 appears smaller due to point reassignment, while Cluster 5 expands to capture overlapping data. The GMM clustering shows clear, curved boundaries that adapt to the data's shape, demonstrating the EM algorithm's ability to model complex patterns. In comparison, K-Means plots exhibit linear boundaries that fail to capture these nuanced shapes, leading to artificial splits.

Comparison for Varying Values of K
These resuls can be seen in Figure 9.
For $K = 6$ in the GMM, there are only five clearly distinguishable clusters, with the sixth cluster being minor and likely representing noise. The overall cluster structure remains consistent with that of $K = 5$, except for minor shifts. For instance, the decision boundary between the large bottom cluster and the left cluster mirrors the $K = 5$ plot, while a few pink points near the origin and the left cluster suggest noise points being assigned to the sixth component.

For K-Means with $K = 6$, the clusters are more evenly sized. Compared to $K = 5$, the boundary between the large bottom cluster and the left cluster now includes parts of the cluster near the origin, creating parallel decision boundaries that split these clusters evenly. Consequently, the cluster around the origin loses a significant portion of its size but still captures a few noise points from the upper cluster.

For $K = 4$ in GMM, the clusters merge to maintain overall likelihood. The bottom and left clusters shift slightly, creating a near-vertical decision boundary. The cluster at the origin and the one on the right merge into a single component due to the algorithm's tendency to optimize for the highest likelihood.

In K-Means with $K = 4$, the top and right clusters remain unchanged from the $K = 5$ configuration. However, the cluster around the origin now merges with parts of the left and bottom clusters, resulting in a single, diagonal boundary separating the left and right clusters.

The variations in clustering for different K values highlight the EM algorithm's adaptability. For fewer clusters ($K = 4$), the algorithm merges clusters to maintain likelihood, while for more clusters ($K = 6$), it introduces new components to account for minor details or noise. This adaptability allows GMM to create meaningful, flexible cluster boundaries that align with the true data distribution, unlike the rigid, uniform boundaries of K-Means.

4. Conclusion

The exploration of GMM and PPCA demonstrates their complementary strengths in probabilistic modeling. GMM effectively clusters data while revealing hidden components via the EM algorithm, with results dependent on initialization and the number of clusters. PPCA identifies latent variables and provides robust dimensionality reduction, but issues such as variance overestimation require attention. Validation experiments confirm theoretical expectations, showing consistency between scree plots and cross-validation findings. These analyses underscore the importance of parameter selection and algorithmic precision in leveraging these techniques for real-world applications.

A. Task 1 Plots:

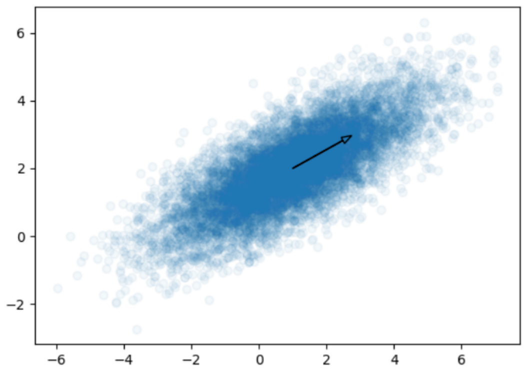

Figure 1: ppca plot of Toy Data with: L=1, sigma2=0.5.

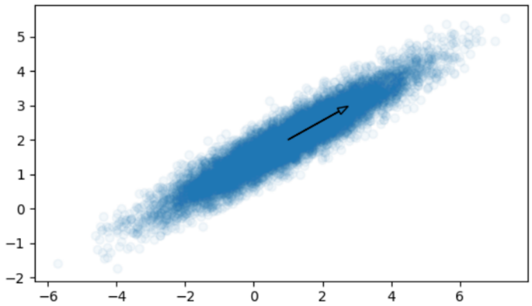

Figure 2: ppca plot 2d of Toy Data with: L=1, sigma2=0.1.

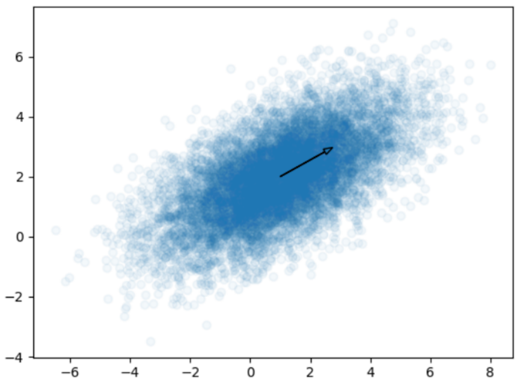

Figure 3: ppca plot 2d Toy Data with: L=1, sigma2=1.

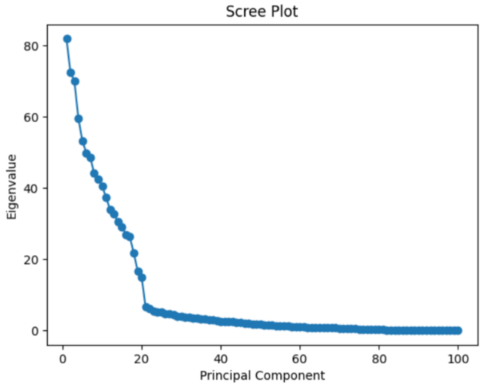

Figure 4: Scree Plot.

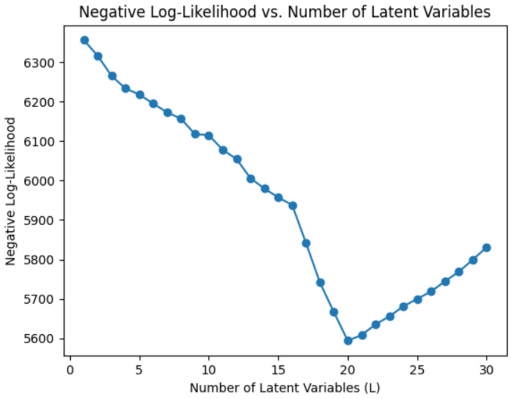

Figure 5: Negative-Log-Likelihood vs. Number of Latent Variables.

B. Task 2 Plots:

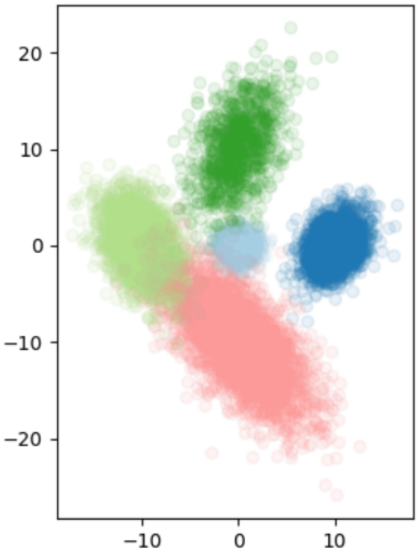

Figure 6: Toy Data using GMM $K = 5$ components.

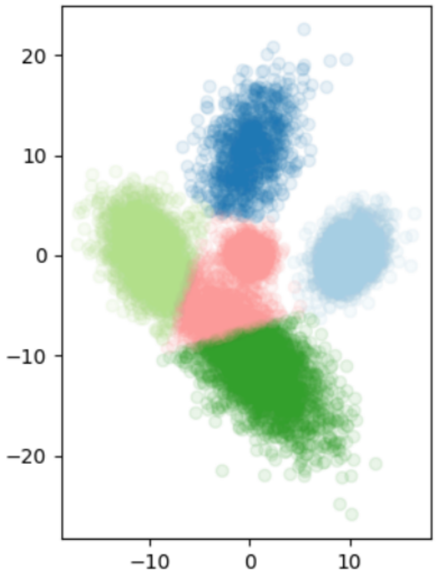

Figure 7: Toy Data using K-means $K = 5$ components.

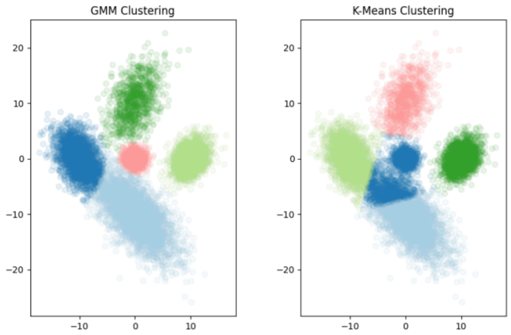

Figure 8: GMM and K-means clustering for $K = 5$ components.

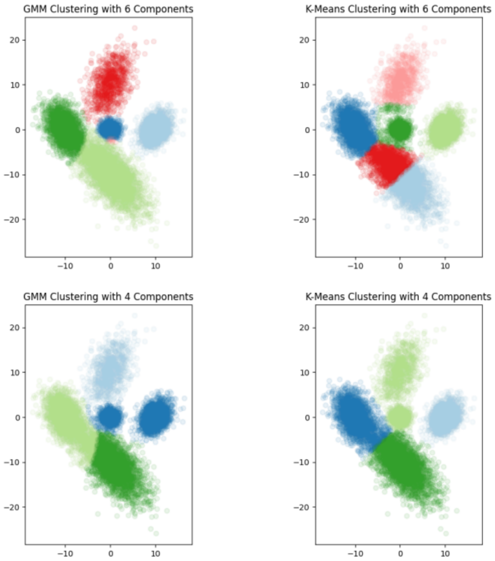

Figure 9: GMM and K-means lustering for $K = 4; 6$ components.

Declaration of Honor

I hereby declare that I have written the enclosed project report without the help of third parties and without the use of other sources and aids other than those listed in the table below and that I have identified the passages taken from the sources used verbatim or in terms of content as such. or content taken from the sources used. This work has not been submitted in the same or a similar form been submitted to any examination authority. I am aware that a false declaration declaration will have legal consequences.

Declaration of Used AI Tools

Tool	Purpose	Where?	Useful?
ChatGPT	Rephrasing	Throughout	+
DeepL	Translation	Throughout	+
Github Copilot	Code generation	a04-lvm.ipynb	+

Signatures
Mannheim, 17. November 2024