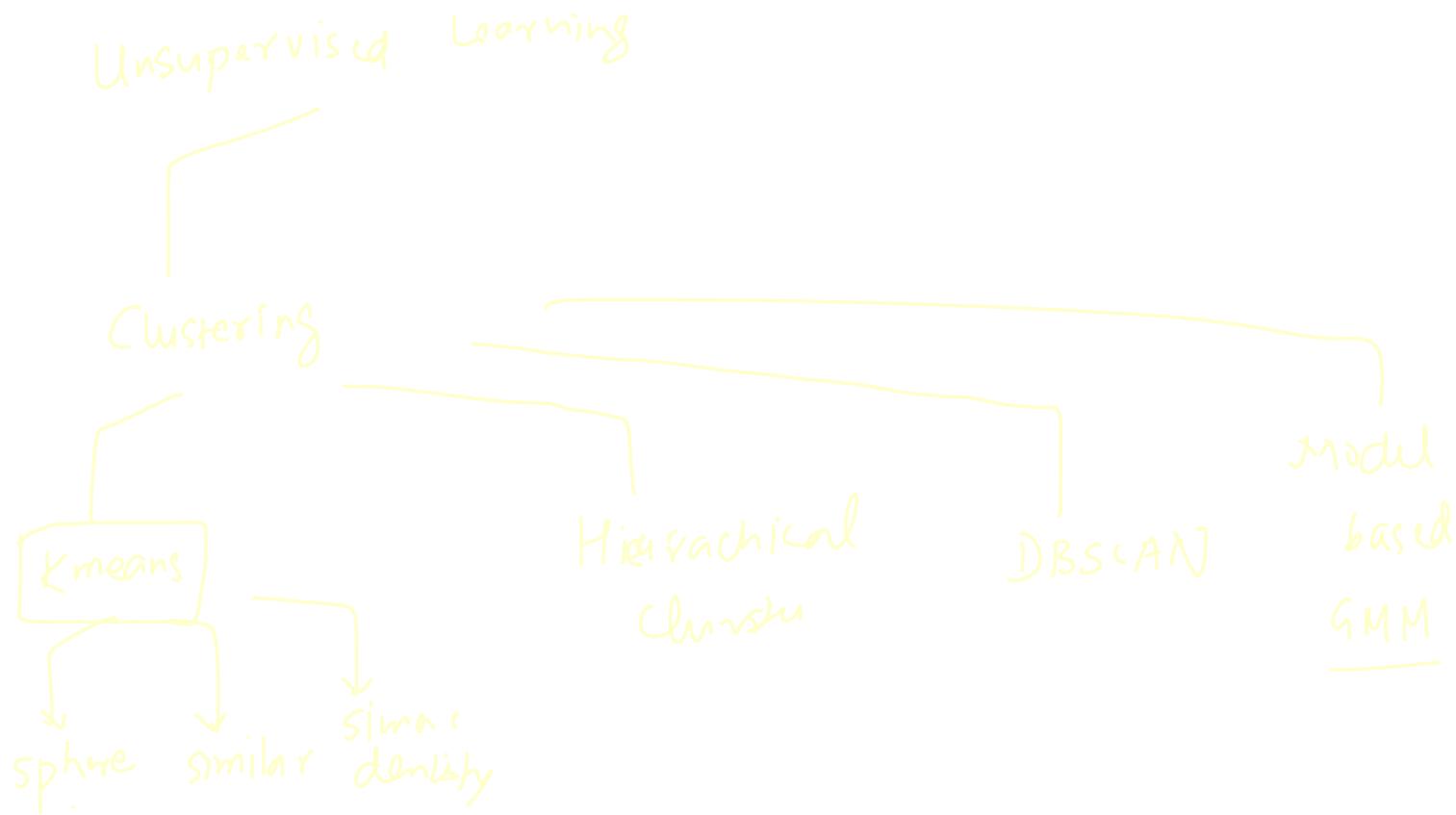


The Why?

05 January 2024 00:26

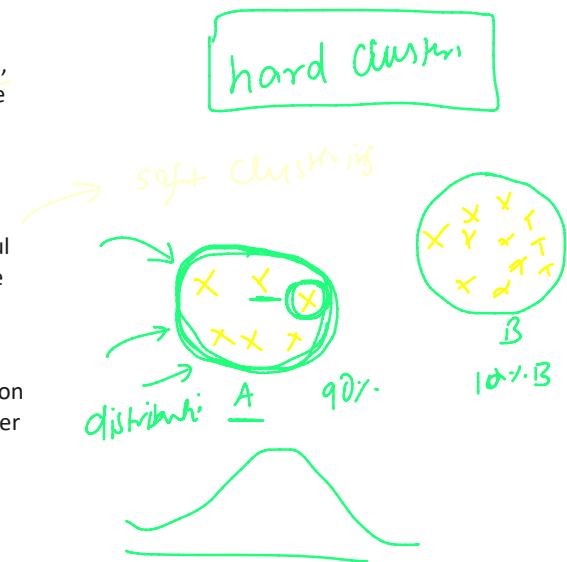


The What?

05 January 2024 14:15

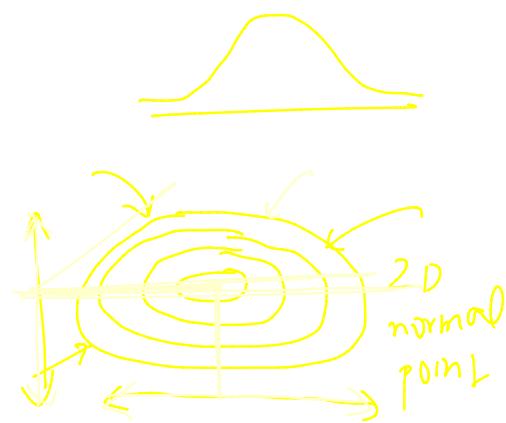
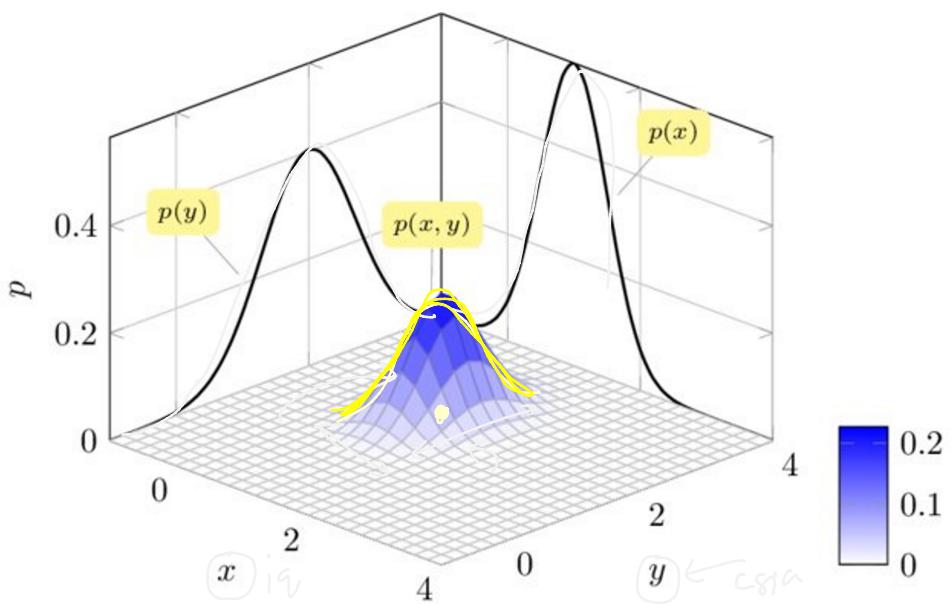
A Gaussian Mixture Model (GMM) is a probabilistic model based clustering technique where we assume that data is generated from a mixture of several Gaussian distributions, each with its own mean and covariance. GMMs are used extensively in statistics, machine learning, and data analysis for tasks like clustering, density estimation, and pattern recognition.

1. **Clustering Capabilities:** Unlike hard clustering methods like K-means, GMMs provide soft clustering, assigning probabilities of membership to each cluster. This is particularly useful in scenarios where data points do not distinctly belong to one group or another but share properties with multiple clusters. It can also cluster data with non-spherical shape.
2. **Density Estimation:** GMMs are used for density estimation, which is the process of estimating the probability distribution underlying a dataset. Understanding this distribution is crucial in many areas, including risk assessment in finance, image processing in computer vision, and anomaly detection in various domains.
3. **Handling Correlated and Complex Data:** With their ability to model different covariance structures, GMMs can handle complex datasets where features are correlated, unlike algorithms that assume feature independence.
4. **Anomaly Detection and Novelty Detection:** GMMs can be used to identify data points that do not fit well into any of the Gaussian components, which is useful for anomaly or outlier detection.



Multivariate Normal Distribution

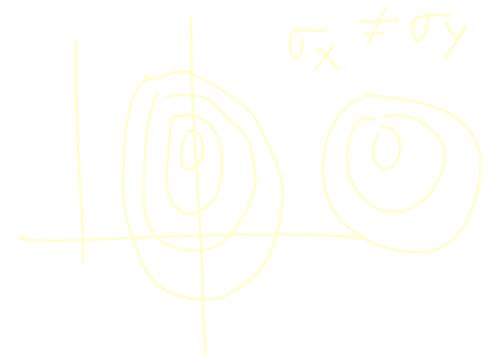
05 January 2024 00:27



$$\sigma_{xy} = 0$$



$$\sigma_x = \sigma_y$$



$$\sigma_x \neq \sigma_y$$



$$\begin{aligned} & \text{2D Normal} & & \text{2D } 2 \times 2 \\ & N(\mu, \Sigma) & & \text{covariance matrix} \\ & [\mu_x \mu_y] & & \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{yx} & \sigma_y^2 \end{bmatrix} \end{aligned}$$

Geometric Intuition 2D

05 January 2024 11:33

EM Algorithm

k means \rightarrow distance / centroid
GMM \rightarrow prob / parameters

clustering technique

\rightarrow model based

GMM

②

n



mixture

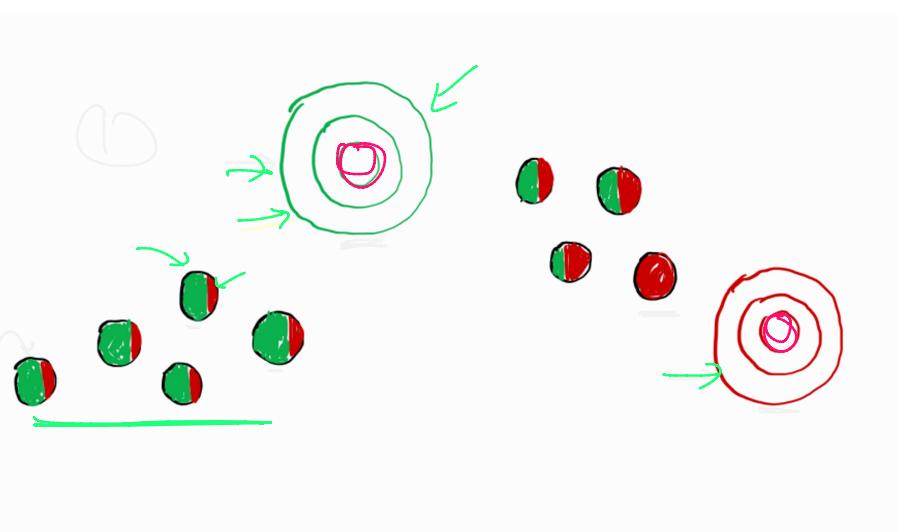
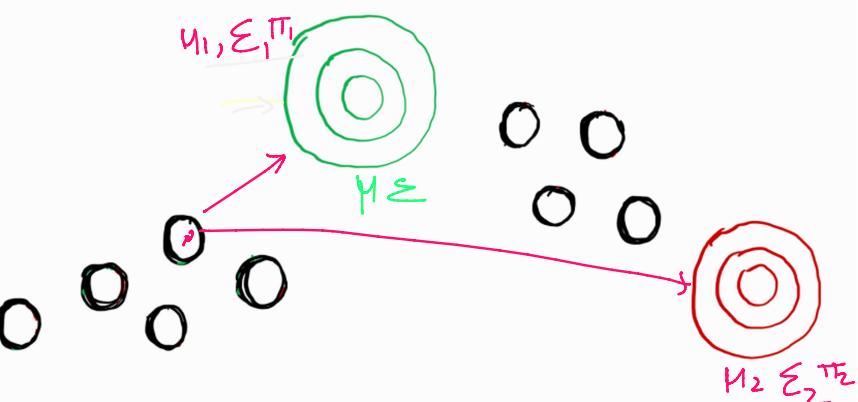
\rightarrow density estimation

③ $\overline{\text{soft clustering}}$

mixture weight

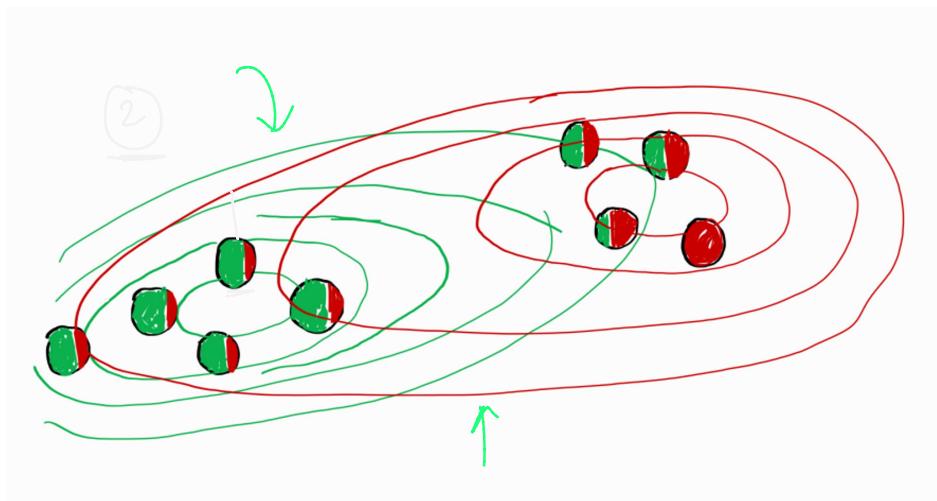
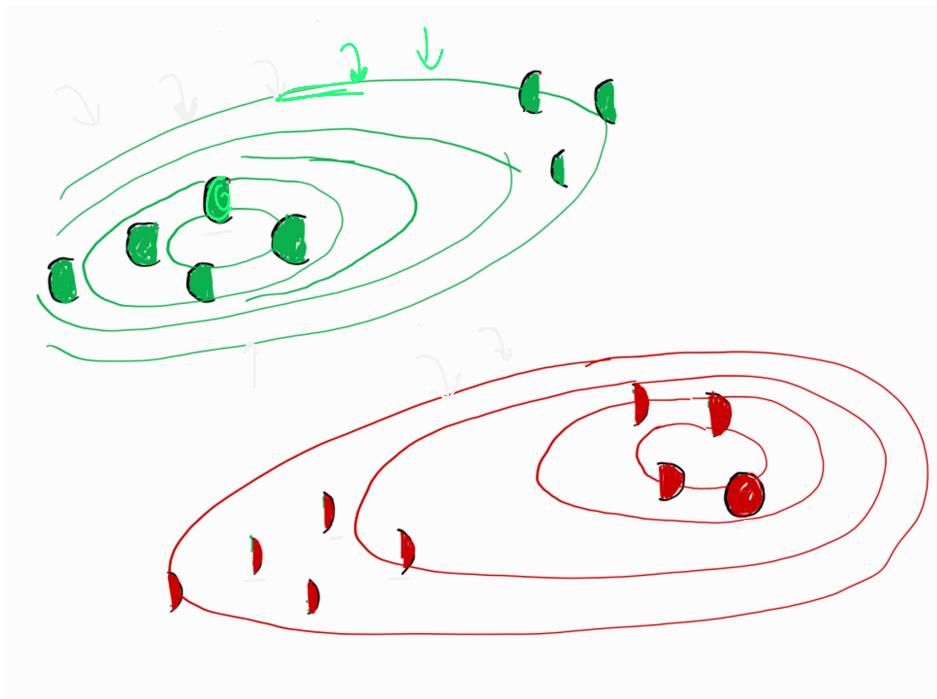
$$\pi \rightarrow \frac{5}{9}$$

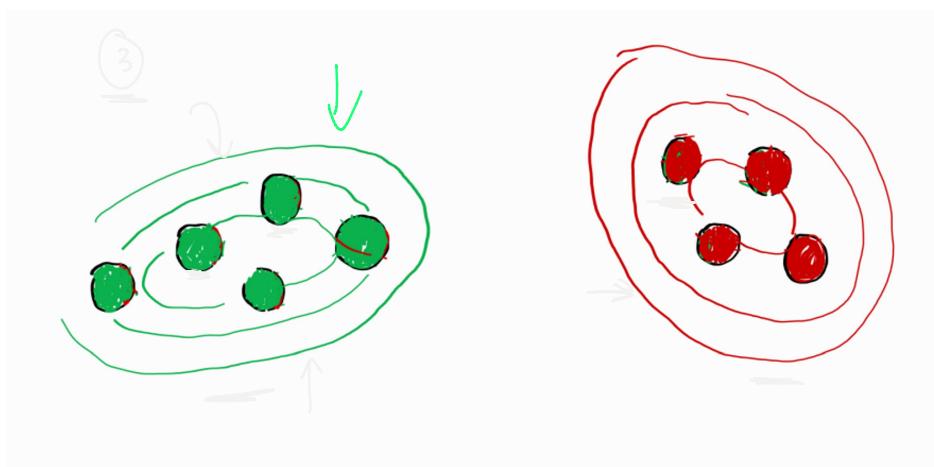
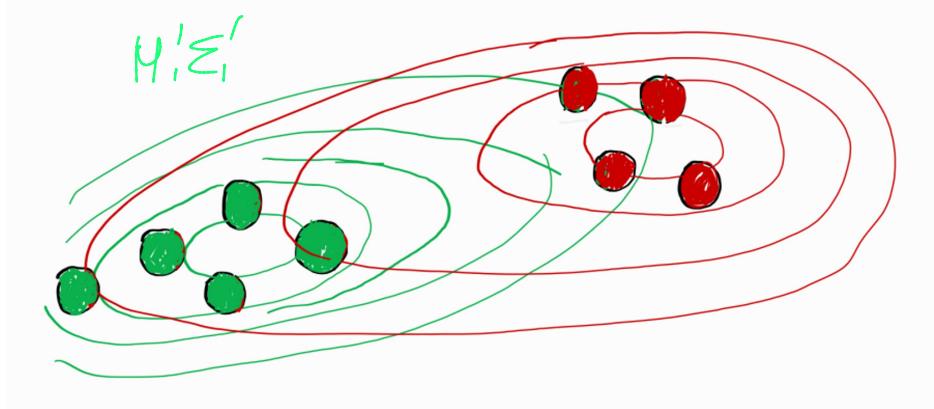
$$\pi_2 = \frac{4}{9}$$



→ EM algorithm

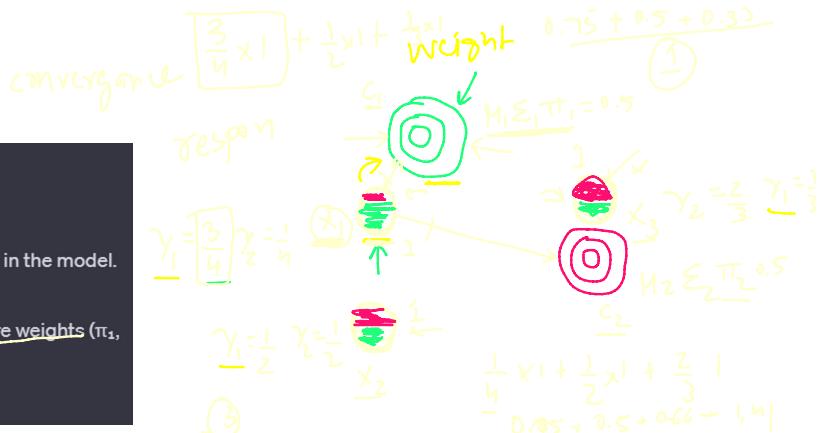
EM algorithm





1) kmean
 $\frac{\text{distance}}{\text{Prob}}$ | $\frac{\text{centroid}}{\text{parameters}}$

2) soft vs hard

**Initialization** $K=2$

- Choose the Number of Components (K):**
 - Decide on the number of Gaussian distributions (components) to use in the model.
- Initialize Parameters:**
 - Initialize the means ($\mu_1, \mu_2, \dots, \mu_k$), covariances ($\Sigma_1, \Sigma_2, \dots, \Sigma_k$), and mixture weights ($\pi_1, \pi_2, \dots, \pi_k$) for each component.
 - These can be initialized randomly or based on some heuristic.

Expectation-Maximization Algorithm

Repeat the following steps until convergence (i.e., until the change in the parameters becomes negligible or a maximum number of iterations is reached).

Expectation Step (E-Step)**Calculate Responsibilities:**

- For each data point x_i and each component k , calculate the responsibility $\gamma_k(x_i)$ using the formula:

$$\gamma_k(x_i) = \frac{\pi_k \cdot P(x_i | \mu_k, \Sigma_k)}{\sum_{j=1}^K \pi_j \cdot P(x_i | \mu_j, \Sigma_j)}$$

Here, $P(x_i | \mu_k, \Sigma_k)$ is the probability density of x_i under the k -th Gaussian distribution, calculated using the Gaussian probability density function.

i = data
 $K=2$

Calculate Responsibilities

$$\rightarrow \gamma_k(x_i) = \frac{\pi_k \cdot P(x_i | \mu_k, \Sigma_k)}{\sum_{j=1}^K \pi_j \cdot P(x_i | \mu_j, \Sigma_j)}$$

$\gamma_1(x_1)$ $\gamma_1(x_2)$ $\gamma_1(x_3)$
 $\gamma_2(x_1)$ $\gamma_2(x_2)$ $\gamma_2(x_3)$

$$\gamma_1(x_1) = \frac{\pi_1 \cdot P(x_1 | \mu_1, \Sigma_1)}{\sum_{j=1}^K \pi_j \cdot P(x_1 | \mu_j, \Sigma_j)}$$

$$\rightarrow \pi_1 \cdot P(x_1 | \mu_1, \Sigma_1) + \pi_2 \cdot P(x_1 | \mu_2, \Sigma_2)$$

Maximization Step (M-Step)**Update Parameters:**

- Update the parameters for each Gaussian component based on the responsibilities calculated in the E-step.

Update Means (μ_k):

$$\mu_k = \frac{1}{N_k} \sum_{i=1}^N \gamma_k(x_i) \cdot x_i$$

- Where N_k is the effective number of data points for the k -th component, calculated as $N_k = \sum_{i=1}^N \gamma_k(x_i)$.

Update Covariances (Σ_k):

$$\Sigma_k = \frac{1}{N_k} \sum_{i=1}^N \gamma_k(x_i) \cdot (x_i - \mu_k)(x_i - \mu_k)^T$$

Update Mixture Weights (π_k):

$$\pi_k = \frac{N_k}{N}$$

- Where N is the total number of data points.

Convergence Check

- After each iteration, check if the change in parameters (means, covariances, weights) is below a certain threshold.
- Alternatively, check if the log-likelihood of the data given the model is not increasing significantly, which is a sign of convergence.

Output

- The final model parameters (means, covariances, weights) after convergence are used to represent the data as a mixture of Gaussian distributions.

Effective number of points

$$C_1 = N_1 = 1.58$$

$$C_2 = N_2 = 1.42$$

$$+ \quad \quad \quad$$

$$N_k = \sum_{i=1}^N \gamma_k(x_i)$$

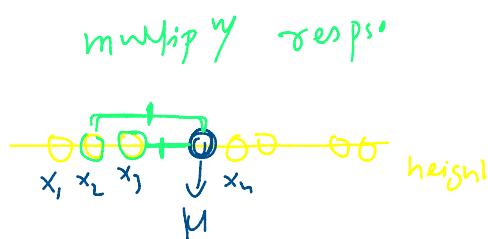
$$\downarrow \quad \quad \quad$$

$$\underline{N_1} \quad \underline{N_2} \quad \underline{N} = \gamma_1(x_1) + \gamma_1(x_2) + \gamma_1(x_3)$$

$$(50) + (30) + 60 \quad \mu_2$$

$$\begin{aligned} & \text{Update } \mu_2 = \frac{\gamma_1(x_1) + \gamma_2(x_2)}{N_1 + N_2} \\ & \text{green } \mu_2 = \frac{1.58}{3} = 0.5267 \\ & \text{red } \mu_2 = \frac{1.42}{3} = 0.4733 \\ & \text{Update weight } \pi_K \in \Pi_K \\ & \text{green } \pi_K = \frac{N_1}{N} = \frac{1}{3} \\ & \text{red } \pi_K = \frac{N_2}{N} = \frac{2}{3} \end{aligned}$$

$$\sum_{i=1}^n \frac{(x_i - \mu)^2}{N}$$



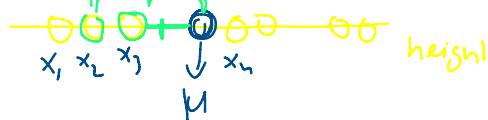
$$\text{Update weight } \pi_K = \frac{1.58}{3}$$

$$\pi_K = \frac{N_1}{N}$$

$$\pi_K = \frac{N_2}{N}$$

mult by resp.

$$\sum_{i=1}^n \frac{\gamma_K(x_i)(x_i - \mu)^2}{N_K}$$



update covariance matrix

$$\text{Update variance } \frac{1}{N} \sum_{i=1}^n \frac{\gamma_K(x_i)(x_i - \mu)^2}{N_K}$$

Python Code

05 January 2024 00:28

Covariance Type

05 January 2024 00:28

1. Spherical:

- In this type, each Gaussian component has its own single variance.
- All dimensions are assumed to have the same variance, which means that the component has a spherical shape in the feature space.
- This is the simplest covariance type, leading to equally sized clusters across all dimensions.

$$\text{cov} \rightarrow [a \ b] \rightarrow a_1 = a_2 = \dots = a_d$$

2. Diagonal:

- Each Gaussian component has its own diagonal covariance matrix.
- This means that the variances can differ across dimensions, but covariances between dimensions (off-diagonal elements) are assumed to be zero.
- The resulting clusters can have different sizes but are not rotated—i.e., they are aligned with the coordinate axes.

$$a+d \rightarrow [a \ b] \rightarrow a_i \neq a_j$$

3. Tied:

- All Gaussian components share the same general covariance matrix.
- This means that all components have the same shape, size, and orientation, but they can have different means (centers).
- This type can capture complex shapes but assumes that the same shape applies to all clusters in the data.

4. Full:

- Each Gaussian component has its own general covariance matrix.
- This is the most flexible type, as it allows each cluster to have its own shape, size, and orientation in the feature space.
- While it can model the data most accurately, it also requires the most parameters and can be prone to overfitting, especially in cases of limited data or high dimensionality.

$$\Sigma_1, \Sigma_2, \Sigma_3$$



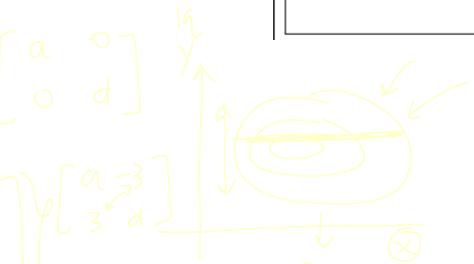
covariance PCA

$$\begin{bmatrix} a & 0 \\ 0 & d \end{bmatrix}$$

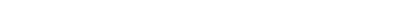
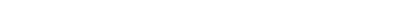
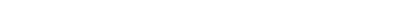
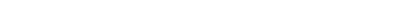
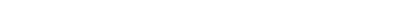
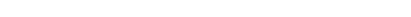


$$\left\{ \begin{bmatrix} a & b \\ c & d \end{bmatrix} \right\} \rightarrow \begin{bmatrix} a & 0 \\ 0 & d \end{bmatrix}$$

$$\Sigma = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{bmatrix}$$



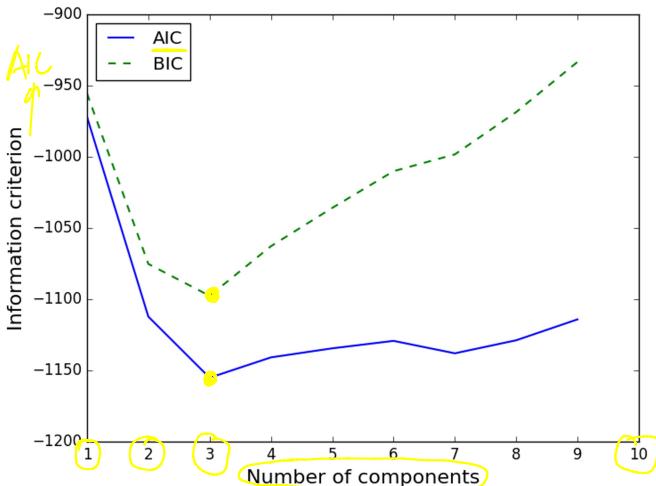
$$\Sigma = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$



How to decide $n_{\text{components}}$?

05 January 2024 12:02

$$(2) \text{ GP } 2d \rightarrow \mu_1 \quad \Sigma_1 \quad T_{11} \quad (6 \times 12) \quad 6 \times 3 \\ (2 \text{ mean}) \quad 3 \text{ item} \quad 1 \quad 18 \\ 2 + 3 + 1$$



Likelihood Formula for a Gaussian Mixture Model:

Given a GMM with K Gaussian components, each with its own parameters (mean μ_k , covariance Σ_k , and weight π_k), the likelihood for a dataset $\{x_1, x_2, \dots, x_N\}$ is:

$$L = \prod_{i=1}^N P(x_i)$$

where $P(x_i)$ is the probability of observing the data point x_i under the GMM, which is a weighted sum of the Gaussian components:

$$P(x_i) = \sum_{k=1}^K \pi_k \cdot \mathcal{N}(x_i | \mu_k, \Sigma_k)$$

Here:

- $\mathcal{N}(x_i | \mu_k, \Sigma_k)$ is the probability density of x_i under the k -th Gaussian distribution, which can be calculated using the multivariate normal density function.
- π_k is the weight of the k -th Gaussian component in the mixture.

Akaike Information Criterion (AIC):

The AIC is calculated using the following formula:

$$\text{AIC} = k \cdot 2 \ln(L)$$

where:

- k is the number of estimated parameters in the model.
- L is the maximum value of the likelihood function for the model.

The AIC balances the complexity of the model (the number of parameters) against the goodness of fit. A lower AIC value suggests a better model. When comparing models, the one with the lower AIC is generally preferred.

Bayesian Information Criterion (BIC):

The BIC is calculated using the formula:

$$\text{BIC} = \ln(n)k - 2 \ln(L)$$

where:

- n is the number of data points.
- k is the number of estimated parameters in the model.
- L is the maximum likelihood of the model.

Like the AIC, the BIC balances model complexity against fit, but it includes a penalty term for the number of data points, making it more stringent against complex models as the sample size increases. A lower BIC value indicates a better model.

PDF $1/2/3$

max likelihood

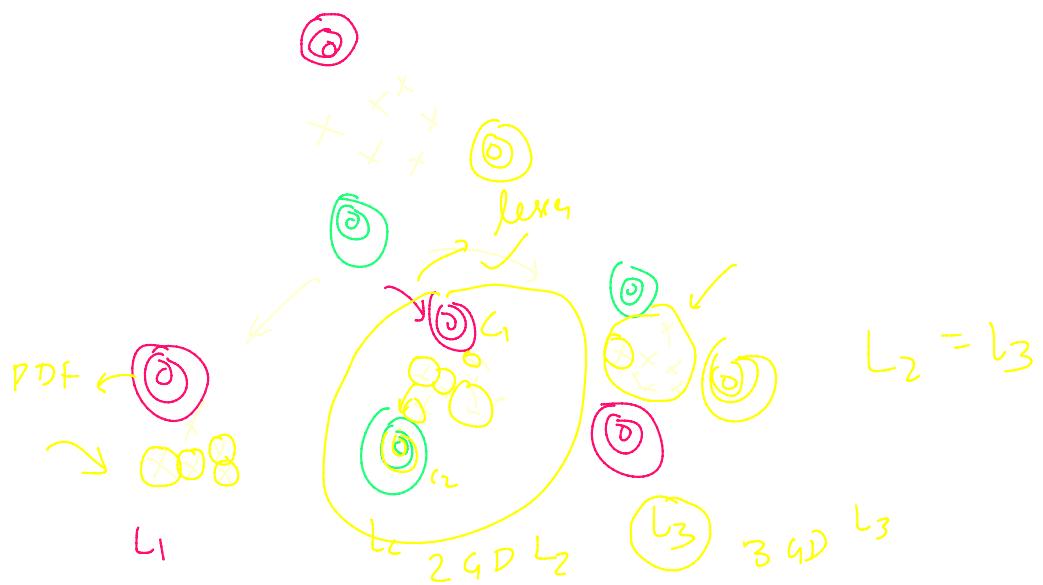
$\log \frac{L}{L_y}$

$L_q = P(x_1|q) \times P(x_2|q) \times P(x_3|q)$

$L_y = P(x_1|y) \times P(x_2|y) \times P(x_3|y)$

$\text{LR} = \frac{L_q}{L_y}$

$\text{LR} = \frac{\log \frac{L}{L_y}}{\log \frac{L_q}{L_y}}$



Why not Silhouette Score?

05 January 2024 14:31

Viz

05 January 2024 14:53

Assumptions

05 January 2024 14:28

1. Normality of Components:

- GMMs assume that each cluster in the data can be represented by a Gaussian (normal) distribution. This means that within each cluster, the data is expected to follow a bell-shaped curve characteristic of the Gaussian distribution.

2. Independence of Components:

- The model assumes that the individual Gaussian distributions (components) that make up the mixture are independent of each other. The presence or characteristics of one component do not affect the others.

3. Elliptical Shape of Clusters:

- Based on the Gaussian assumption, the shape of the clusters is expected to be elliptical. The exact shape can vary (spherical, elongated, rotated, etc.) depending on the covariance structure of each Gaussian component.

4. Soft Clustering:

- GMM assumes that data points can belong to more than one cluster, with varying degrees of membership. This is a departure from hard clustering methods like K-means, where each point is assigned to exactly one cluster.

5. Stationarity:

- GMMs often assume that the statistical properties of the clusters (mean, variance) do not change over time. This is important in applications involving time-series data.

Advantages & Disadvantages

05 January 2024 00:28

Advantages

1. **Soft Clustering:** Unlike K-means which assigns each data point to a single cluster (hard clustering), GMM provides the probability of each data point belonging to each cluster (soft clustering). This probabilistic clustering approach can offer more nuanced insights, especially when the data points could reasonably belong to more than one cluster.
2. **Flexibility in Cluster Shapes:** GMMs can accommodate clusters with different sizes and shapes, as each cluster is modelled with its own covariance matrix. This is in contrast to algorithms like K-means, which assume spherical clusters with similar sizes.
3. **Cluster Density Estimation:** GMMs not only assign data points to clusters but also provide a density estimation for each cluster. This is useful for understanding the distribution of data within each cluster.
4. **Accommodating Correlated Features:** Thanks to the covariance matrix in each Gaussian distribution, GMMs can handle clusters where features are correlated. This is more reflective of real-world data where such correlations are common.
5. **Model Complexity Control:** With GMMs, you can control the complexity of the model by limiting the shape of the covariance matrix (spherical, diagonal, tied, or full covariance). This control allows a balance between overfitting and underfitting.

Disadvantages

1. **Assumption of Gaussian Components:** GMMs assume that the data is generated from a mixture of Gaussian distributions. This assumption may not always be valid, especially for datasets with non-Gaussian or multimodal distributions.
2. **Sensitivity to Initialization:** GMMs can be sensitive to the initial choice of parameters. Poor initialization can lead to convergence to suboptimal solutions or local optima. This issue is often mitigated by using methods like k-means for initial parameter estimation.
3. **Choosing the Number of Components:** Determining the optimal number of Gaussian components in the mixture model can be challenging. It often requires

heuristic methods or information criteria like Akaike Information Criterion (AIC) or Bayesian Information Criterion (BIC), which can add complexity to the modelling process.

4. **Computational Complexity:** GMMs can be computationally intensive, especially for large datasets or when the number of components is high. This complexity arises from the calculations of the covariance matrices and the inversion of these matrices in the E-step of the EM algorithm.
5. **Difficulty in Interpretation:** The soft clustering nature of GMMs (where each point has a probability of belonging to multiple clusters) can sometimes make the interpretation of the results less straightforward compared to hard clustering methods like k-means.
6. **Handling High-Dimensional Data:** GMMs may not perform well with high-dimensional data due to the curse of dimensionality. In high dimensions, estimating the covariance matrices becomes difficult and less reliable.
7. **Covariance Structure:** The choice of covariance structure (spherical, diagonal, tied, full) impacts the model's flexibility and its ability to fit the data. However, choosing the appropriate structure is not always straightforward and can significantly affect the model's performance.

Kmeans Vs GMM

05 January 2024 14:21

1. Soft Clustering vs. Hard Clustering:

- GMM: Provides soft clustering, meaning each data point is assigned a probability (or weight) indicating its membership in each cluster. This is useful for understanding the uncertainty or the degree of membership of each point in multiple clusters.
- K-means: Implements hard clustering, where each data point is definitively assigned to one cluster. While simpler, this approach can be limiting in scenarios where the data naturally belongs to overlapping clusters.

2. Cluster Shape and Flexibility:

- GMM: Can accommodate clusters with different shapes and sizes due to its ability to model each cluster with its own covariance matrix. This allows GMM to fit more complex and diverse datasets effectively.
- K-means: Assumes clusters are spherical and of similar size. This can lead to suboptimal clustering when the true clusters are non-spherical or have different sizes.

3. Robustness to Initialization:

- GMM: Generally more robust to the initialization of cluster centers, especially when using methods like 'kmeans' for initialization.
- K-means: Can be quite sensitive to the initial placement of centroids, often requiring multiple runs with different initializations.

4. Probability Density Estimation:

- GMM: Not only clusters data but also provides a probability density estimation for the data distribution.
- K-means: Does not provide any probability density estimation.

DBSCAN Vs GMMs

05 January 2024 14:25

1. Probabilistic Cluster Assignment (Soft Clustering):

- GMM provides a probabilistic measure of membership for each data point in each cluster. This soft clustering approach is useful when the clusters overlap or when it's informative to know the degree of certainty about each point's cluster membership.
- DBSCAN, in contrast, assigns each point to exactly one cluster or as noise, which is a form of hard clustering.

2. Model-Based Approach:

- GMMs are based on a well-defined statistical model, which can be advantageous when you have some a priori knowledge about the data or when a probabilistic framework is desired for further statistical analysis.
- DBSCAN's density-based approach is more heuristic and doesn't provide a statistical model of the data.

3. Handling of Different Densities:

- While DBSCAN can struggle with data where clusters have different densities, GMMs can more effectively model such scenarios by adjusting the variances of the Gaussian components.

4. Parameter Interpretability:

- The parameters of a GMM (means, covariances, and mixture weights) have clear interpretations in terms of the data distribution. This can provide insights into the nature of the clusters.
- DBSCAN's parameters (`eps` and `min_samples`) are less intuitive and often require trial-and-error to set appropriately.

5. Quantitative Analysis:

- GMMs allow for a more quantitative analysis since they provide a likelihood function. You can use information criteria like AIC or BIC for model selection or to determine the number of components.
- DBSCAN doesn't provide a comparable quantitative measure for evaluating the quality of the clustering or for choosing its parameters.

Applications

05 January 2024 00:28

1. **Clustering:** GMMs are widely used for clustering when the data is assumed to come from a mixture of underlying Gaussian distributions. They provide more flexibility than K-means clustering by allowing for elliptical as well as spherical clusters.
2. **Density Estimation:** GMMs can model the probability distribution of a dataset, providing a parametric estimate of the density function. This is particularly useful in fields like statistics and data analysis for understanding the underlying distribution of data.
3. **Pattern Recognition:** In the field of pattern recognition, GMMs are used to identify patterns or regularities within data. This includes applications in image analysis and recognition, biometric identification (like fingerprint or face recognition), and speech and handwriting recognition.
4. **Image Segmentation:** In computer vision, GMMs are used to segment images into different regions based on color or texture. This application is beneficial in medical imaging, satellite image analysis, and automated image editing.
5. **Anomaly Detection:** GMMs can be used to detect unusual patterns or anomalies in data. By modeling the normal behavior of a system, anomalies can be identified as data points that have low probability under the fitted model.
6. **Financial Modeling:** In finance, GMMs are applied to model and analyze asset returns, risk modeling, and other financial time series data, given their ability to capture the multimodal nature of financial market movements.
7. **Speech Processing:** GMMs are employed in speech processing for speaker verification and identification, speech and sound recognition, and in the modeling of speech and audio signals.
8. **Bioinformatics:** In bioinformatics, GMMs are used for modeling gene expression data, protein folding patterns, and other biological processes that exhibit variability that can be captured effectively by mixture models.
9. **Astronomy:** Astronomers use GMMs to analyze the distribution of stars or galaxies, especially for clustering and identifying substructures within the universe.

10. **Market Research and Customer Segmentation:** GMMs can segment customers or markets based on purchasing behavior, preferences, or other measurable criteria, allowing for targeted marketing and product development.