Intelligent Systems: Reasoning and Recognition

James L. Crowley

MoSIG M1 Winter Semester 2021
Lesson 5 23 February 2021

Gaussian Mixture Models, K-Means and EM

Notation..2

Probability Density Functions...3
Bayes Rule with probability density functions 3
The Central Limit theorem and Normal Densities..........................4
Multivariate Normal Density Function 4

Gaussian Mixture Models 7
A Sum of Independent Sources...7
Estimating Gaussian Mixture models from Training Data 8

K-Means Clustering 9

The Expectation Maximization Algorithm (EM) 11
Convergence Criteria 13

Using Gaussian Mixture Models with Baye's Rule 14

Sources:

C. M. Bishop, "Pattern Recognition and Machine Learning", Springer Verlag, 2006.

Jeff Bilmes, A Gentle Tutorial of the EM Algorithm, Tech Report, Univ of Washington, 1998. (available for download from course website).

Notation

x	a variable
X	a random variable (unpredictable value)
$\vec{x}$	A vector of D variables.
$\vec{X}$	A vector of D random variables.
D	The number of dimensions for the vector $\vec{x}$ or $\vec{X}$
k	index for cluster, data source or GMM Mode
K	Total number of clusters, or sources, of events
M	Total number of sample events. $M = \sum_{k=1}^{K} M_k$
$\{\vec{X}_m\}$	A set of M Sample Observations (a training set)
$\{\vec{y}_m\}$	A set of indicator vectors for the training samples in $\{\vec{X}_m\}$ $\vec{y}_m$ indicates the source S_k for each training sample $\vec{X}_m$
Note that	$\vec{y}_m$ can be a binary vector with k rows (1 for S_k and 0 for others) or $\vec{y}_m$ can be the probability that $\vec{X}_m \in S_k$

$h(k,m) = \left(\vec{y}_1 \ \cdots \ \vec{y}_m\right)$ Indicator variables in matrix form. k rows, m columns

Eulers Number "e" e = 2.718281828…

Expected Value: $E\{X\} = \frac{1}{M}\sum_{m=1}^{M} X_m$

Mean: $\mu = E\{X\}$

Variance: $\sigma^2 = E\left\{(X-\mu)^2\right\} = E\left\{(X - E\{X\})^2\right\}$

Gaussian or Normal Density:

1-Dimension:

$$\mathcal{N}(X;\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(X-\mu)^2}{2\sigma^2}}$$

D-dimensions:

$$p(\vec{X}) = \mathcal{N}(\vec{X};\vec{\mu},\Sigma) = \frac{1}{(2\pi)^{\frac{D}{2}} \det(\Sigma)^{\frac{1}{2}}} e^{-\frac{1}{2}(\vec{X}-\vec{\mu})^T \Sigma^{-1} (\vec{X}-\vec{\mu})}$$

Probability Density Functions

A probability density function *p(X),* is a function of a continuous variable *X* such that

1) *X* is a continuous real valued random variable with values between $[-\infty, \infty]$
2) $\int_{-\infty}^{\infty} p(X) = 1$

Note that *p(X)* is NOT a number but a continuous function.

A probability density function defines the relatively likelihood for a specific value of *X*. Because *X* is continuous, the value of *p(X)* for a specific *X* is infinitely small. To obtain a probability we must integrate over some range of *X*.
To obtain a probability we must integrate over some range V of X.
In the case of D=1, the probability that X is within the interval [A, B] is

$$P(X \in [A,B]) = \int_A^B p(x)\,dx$$

This integral gives a number that can be used as a probability.

Note that we use upper case $P(X \in [A,B])$ to represent a probability value, and lower case *p(X)* to represent a probability density function.

Bayes Rule with probability density functions

Let ω_k represent the statement that a random variable is a member of class C_k: $\omega_k = X \in C_k$. Bayes Rule can be used to compute this probability as:

$$P(\omega_k \mid X) = \frac{p(X \mid \omega_k)}{p(X)} P(\omega_k) = \frac{p(X \mid \omega_k)P(\omega_k)}{\sum_{j=1}^{K} p(X \mid \omega_j)P(\omega_j)}$$

$\frac{p(X \mid \omega_k)}{p(X)}$ IS a number, provided that $p(X) = \sum_{k=1}^{K} p(X \mid \omega_k)P(\omega_k)$

This requires that the set of classes are disjoint and complete. Ever sample belongs to one and only one class C_k.

Probability density functions are easily generalized to vectors of random variables.
Let $\vec{X} \in R^D$, be a vector random variables.
A probability density function, $p(\vec{X})$, is a function of a vector of continuous variables

1) $\vec{X}$ is a vector of D real valued random variables with values between $[-\infty, \infty]$
2) $\int_{-\infty}^{\infty} p(\vec{X})d\vec{x} = 1$

The Central Limit theorem and Normal Densities.

The "Central Limit Theorem" tells us that whenever the features an observation are the result of a sequence of N independent random events, the probability density of the features will tend toward a Normal or Gaussian density.

The essence of the derivation is that repeated random events are modeled as repeated convolutions of density functions, and for any finite density function will tend asymptotically to a Gaussian (or normal) function. For any non-ideal density *p(X)* :

$$as\ M \rightarrow \infty \quad p(X)^{*M} \rightarrow \mathcal{N}(x;\ \mu,\ \sigma)$$

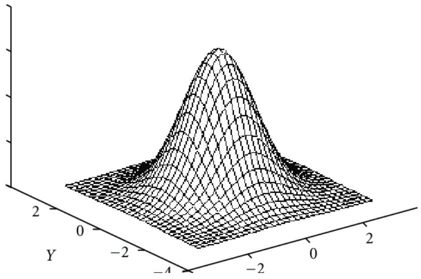
We can consider a sequence of random trials as a "source" of event

Source: $\boxed{S} \longrightarrow X$

The central limit theorem tells us that in this case, a normalized sum of many independent random variables will converge to a Normal or Gaussian density function:

$$p(\vec{X}) = \mathcal{N}(\vec{X};\vec{\mu},\Sigma)$$

Multivariate Normal Density Function



$$p(\vec{X}) = \mathcal{N}(\vec{X};\vec{\mu},\Sigma) = \frac{1}{(2\pi)^{\frac{D}{2}}\ \det(\Sigma)^{\frac{1}{2}}} e^{-\frac{1}{2}(\vec{X}-\vec{\mu})^T \Sigma^{-1}(\vec{X}-\vec{\mu})}$$

Where the parameters $\vec{\mu}$, Σ are the first and second moments of the density.

There are 3 parts to $\mathcal{N}(\vec{X};\vec{\mu},\ \Sigma)$:

(1) $\frac{1}{(2\pi)^{\frac{D}{2}}\ \det(\Sigma)^{\frac{1}{2}}}$, (2) e, and (3) $d(\vec{X},\vec{\mu};\Sigma)^2 = \frac{1}{2}\left(\vec{X}-\vec{\mu}\right)^T \Sigma^{-1}\left(\vec{X}-\vec{\mu}\right)$

1) $e = 2.7818281828\ldots$ Euler's Constant : $\int e^x dx = e^x$. Used to simplify the algebra.

2) The term $(2\pi)^{\frac{D}{2}} \det(\Sigma)^{\frac{1}{2}}$ is a normalization factor to assure an integral of 1.

$$(2\pi)^{\frac{D}{2}} \det(\Sigma)^{\frac{1}{2}} = \int\int\ldots\int e^{-\frac{1}{2}(\vec{X}-\vec{\mu})^T \Sigma^{-1} (\vec{X}-\vec{\mu})} dx_1\, dx_2 \ldots dx_D$$

det(Σ) is the determinant of Σ. This is a scalar value that can be computed from the elements of a square matrix and represents the volume of the linear transformation described by the matrix.

For 1-D $\det(a) = a$

For 2-D $\det\begin{pmatrix} a & b \\ c & d \end{pmatrix} = a \cdot d - b \cdot c$

For 3-D $\det\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} = a \cdot \det\begin{pmatrix} e & f \\ h & i \end{pmatrix} + b \cdot \det\begin{pmatrix} f & d \\ i & g \end{pmatrix} + c \cdot \det\begin{pmatrix} d & e \\ g & h \end{pmatrix}$

...

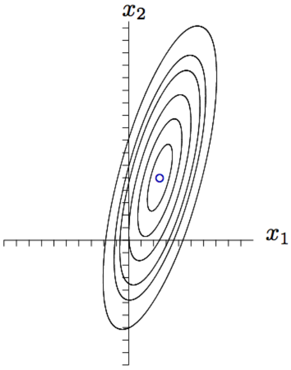
For a Normal density, the determinant represents the volume of the density function.

The mean is $\vec{\mu} = E\{\vec{X}\} = \begin{pmatrix} E\{X_1\} \\ E\{X_2\} \\ \ldots \\ E\{X_D\} \end{pmatrix} = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \ldots \\ \mu_D \end{pmatrix}$

and the Covariance is $\Sigma = E\{(\vec{X} - E\{\vec{X}\})(\vec{X} - E\{\vec{X}\})^T\} = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \ldots & \sigma_{1D} \\ \sigma_{21} & \sigma_{22} & \ldots & \sigma_{2D} \\ \ldots & \ldots & \ddots & \ldots \\ \sigma_{D1} & \sigma_{D2} & \ldots & \sigma_{DD} \end{pmatrix}$

where $\sigma_{ij} = \frac{1}{M} \sum_{m=1}^{M} (x_{mi} - \mu_i)(x_{mj} - \mu_j)$

The result can be visualized by looking at the equi-probable contours.



Ellipses for 99%, 95%, 90%, 75%, 50%, and 20% of the mass

If x_i and x_j are statistically independent, then σ_{ij} =0

For positive values of σ_{ij}, x_i and x_j vary together.
For negative values of σ_{ij}, x_i and x_j vary in opposite directions.

For example, consider features x_1 = height *(meters)* and x_2 = weight *(kg)*

In most people height and weight vary together and so σ_{12} would be positive

The exponent of the Normal is the Mahalanobis distance:

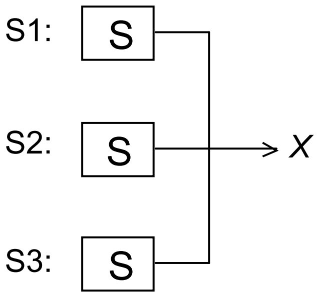
$$d(\vec{X},\vec{\mu};\Sigma) = \frac{1}{2}\left(\vec{X}-\vec{\mu}\right)^T \Sigma^{-1}\left(\vec{X}-\vec{\mu}\right)$$

This is the distance between $\vec{X}$ and the mean $\vec{\mu}$ normalized by the covariance, Σ.
The Mahalanobis distance provides a convenient distance metric when the individual components of $\vec{X}$ have incommensurate units, such a meters and kgs.

Gaussian Mixture Models

A Sum of Independent Sources

Sometimes a population will result from a set of K different sources, S_k, each with it own unique independent random variables and Normal Density function.



In this case, the probability density is better represented as a weighted sum of normal densities.

$$p(\vec{X}) = \sum_{k=1}^{K} \alpha_k \mathcal{N}(\vec{X}; \vec{\mu}_k, \Sigma_k)$$

Such a sum is referred to as a Gaussian Mixture Model (GMM).

A GMM can be used to represent density functions multiple sources. It can also be used to discover a set of subclasses within a global class.

Each normal density is considered to be produced from a different source, indicated by the coefficients α_k.
We can see the coefficients $\{\alpha_k\}$ as the relative frequencies (probabilities) for a set of independent "sources", S_k, for events. The α_k coefficients represent the relative probability that an event came from a source S_k.

For this to be a probability, we must assure that $\sum_{k=1}^{K} \alpha_k = 1$

Thus the α_k are form a probability Distribution: The probability of obtaining a sample from each Source.

Estimating Gaussian Mixture models from Training Data

Estimating a Gaussian mixture model from training data is equivalent to discovering the source for each sample, and to estimate the mean and covariance $(\vec{\mu}_k, \Sigma_k)$ for each source.

We will look at two possible algorithms for this: K-Means Clustering, and Expectation Maximization. In both cases, the algorithm will iteratively construct a table, *h(k,m)* that assigns each sample to one of K clusters or sources.

For K-Means, this will be a hard assignment,
with *h(k, m) = 1* if observation $\vec{X}_m$ is assigned to cluster S_k and 0 otherwise.

This can be seen as equivalent to the indicator variable $\vec{y}_m$

$$h(k,m) = \begin{cases} 1 & \text{if sample } \vec{X}_m \in S_k \\ 0 & \text{Otherwise} \end{cases}$$

h(k, m) = 1 if $\vec{X}_m$ is assigned to cluster k, 0 otherwise.

In the case of EM, this will be a soft assignment, in which *h(k,m)* represents the probability that sample $\vec{X}_m$ comes from source (or cluster), S_k.

$$h(k,m) = P(X_m \in S_k)$$

In either case we must initialize the estimated clusters: This can be initialized with,

$\vec{\mu}_k^{(0)} = k\vec{\mu}_0^{(0)}$, $\Sigma_k^{(0)} = I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ or any other convenient values.

A good initial estimate leads to faster convergence, so always use domain knowledge to initialize $\vec{\mu}_k^{(0)}$ and $\Sigma_k^{(0)}$ when possible.

K-means is sensitive to the starting point and can converge to a local minimum that is not the best estimate. EM is less sensitive and will generally converge to the global best estimate.

K-Means and EM can be used to discover the classes for each training sample, and are thus used for Unsupervised Learning.

They can also be used to estimate a multimodal density for a single class.

K-Means Clustering

Assume a set of M sample observations $\{\vec{X}_m\}$, with each observation drawn from one of K clusters S_k. Our problem is to discover an assignment table *h(k, m)* that assigns each observation, $\vec{X}_m$ in the sample set to the "best" cluster, S_k.

$$h(k,m) = \begin{cases} 1 & \text{if sample } \vec{X}_m \in S_k \\ 0 & \text{Otherwise} \end{cases}$$

Given an estimate of the mean, $\vec{\mu}_k$, and covariance Σ_k for each cluster, S_k. we can use the Mahalanobis Distance to determine the best cluster.

For each cluster we can then refine the estimate of the mean, $\vec{\mu}_k$, and covariance Σ_k.

This suggests an iterative process composed of two steps:

1) Expectation: For each sample, $\vec{X}_m$, determine the most likely cluster $S_{k.}$ using the distance to the current estimate of the mean, $\vec{\mu}_k$, and covariance Σ_k.

2) Maximization: For each cluster re-calcuate the mean, $\vec{\mu}_k$, and covariance Σ_k using sample assignments in *h(k,m)*.

We can initialize the process to any value. For example, $\vec{\mu}_k^{(0)} = k\vec{\mu}_0$, $\Sigma_k^{(0)} = I$

However, it IS possible for K-means to be stuck in a local minimum, and the closer we start to the best values, the faster the process converges.

We will seek to minimize a quality metric:
For K-Means this is the sum of the Mahalanobis distances.

$$Q^{(i)} = \sum_{m=1}^{M} \sum_{k=1}^{K} h^{(i)}(k,m)(\vec{X}_m - \vec{\mu}_k^{(i)})^T \Sigma_k^{(i)-1}(\vec{X}_m - \vec{\mu}_k^{(i)})$$

Initially $h^{(o)}(m, k) = 0, i=0$.
We can stop the process after a fixed number of iterations, or when the assignment table does not change or when $Q^{(i)}$ pass reaches a global minimum.

Expectation:

```
$i \leftarrow i+1$
$\forall m = 1, M$ :
    $\forall k = h^{(i)}(k,m) = 0$
    k $= \arg\text{-}\min_{k}\left\{(\vec{X}_m - \vec{\mu}_k)^T \Sigma_k^{-1} (\vec{X}_m - \vec{\mu}_k)\right\}$
    $h^{(i)}(k,m) \leftarrow 1$
```

Maximization

Mass: $M_k = \sum_{m=1}^{M} h^{(i)}(k,m)$ is the number of samples attributed to source k.

If $M_k \neq 0$:

Mean: $\mu_k^{(i)} = \frac{1}{M_k} \sum_{m=1}^{M} h^{(i)}(k,m) \cdot \vec{X}_m$

Covariance: $\Sigma_k^{(i)} = \frac{1}{M_k} \sum_{m=1}^{M} h^{(i)}(k,m) \cdot (\vec{X}_m - \vec{\mu}_k)(\vec{X}_m - \vec{\mu}_k)^T$

That is, for each component of the covariance, $\sigma_{ij}^{(i)}$:

$$\sigma_{ij}^{(i)} = \frac{1}{M_k} \sum_{m=1}^{M} h^{(i)}(k,m) \cdot (x_{mi} - \mu_{ki})(x_{mj} - \mu_{kj})$$

At the end of each cycle:

Quality: $Q^{(i)} = \sum_{m=1}^{M} \sum_{k=1}^{K} h^{(i)}(m,k)(\vec{X}_m - \vec{\mu}_k^{(i)})^T \Sigma_k^{(i)-1} (\vec{X}_m - \vec{\mu}_k^{(i)})$

The process stops after a fixed number of cycles, or when the sample assignment does not change or the quality metric does not change.

Each source can be interpreted as a separate class or as a mode in a Gaussian Mixture model, depending on the application.

The Expectation Maximization Algorithm (EM)

As before, assume a set of *M* sample observations $\{\vec{X}_m\}$, with each observation drawn from one of K sources S_k. Our problem is to discover an assignment table *h(k, m)* that assigns each observation, $\vec{X}_m$ in the sample set to the "best" cluster, S_k. For EM this will be a probability.

EM iteratively estimates the probability for the assignment of each observation to each source.

Expectation Maximization has many uses, including estimating the density functions for a Hidden Markov Model (HMM) as well as for estimating the parameters for a Gaussian Mixture model.

For a Gaussian Mixture model, a probability density is represented as a weighted sum of normal densities.

$$p(\vec{X}) = \sum_{k=1}^{K} \alpha_k \mathcal{N}(\vec{X};\vec{\mu}_k,\Sigma_k)$$

It is sometimes convenient to group the parameters for each source into a single vector:

$$\vec{v}_k = (\alpha_k, \vec{\mu}_k, \Sigma_k)$$

The complete set of parameters is a vector with K·P coefficients.
For a feature vector of D dimensions, $\vec{v}_k$ has P = 1 + D + D(D+1)/2 coefficients.

To estimate $\{\alpha_k, \vec{\mu}_k, \Sigma_k\}$ we need the assignment of samples to source, *h(k,m)*.
To estimate *h(k,m)* we need the parameters $\{\alpha_k, \vec{\mu}_k, \Sigma_k\}$

This leads to an iterative two-step process in which we alternately estimate *h(k,m)*. and then $\{\alpha_k, \vec{\mu}_k, \Sigma_k\}$.

The EM algorithms constructs a table, *h(k,m)*
Unlike K-Means, *h(k,m)* will contain probabilities.

$$h(k,m) = P(\vec{X}_m \in S_k)$$

Initialization:

Choose K (the number of sources). Use domain knowledge if possible.

set *i=0.*

Form an initial estimate for $\vec{v}^{(0)} = (\alpha_k^{(0)}, \vec{\mu}_k^{(0)}, \Sigma_k^{(0)})$ *for k = 1 to K.*

As with K-means, this can be initialized with $\alpha_k^{(0)} = \frac{1}{K}$, $\vec{\mu}_k^{(0)} = k\vec{\mu}_0$, $\Sigma_k^{(0)} = I$ or with any reasonable first estimation. The closer the initial estimate, the faster the algorithm converges. Domain knowledge is useful here.

Expectation step (E)

let $i \leftarrow i+1$

Calculate the table $h^{(i)}(k,m)$ using the training data and estimated parameters.

$$h^{(i)}(k,m) = P(\vec{X}_m \in S_k \mid \{X_m\}, \vec{v}^{(i-1)})$$

which gives :

$$h^{(i)}(k,m) \leftarrow \frac{\alpha_k^{(i-1)} \mathcal{N}(\vec{X}_m, \vec{\mu}_k^{(i-1)}, \Sigma_k^{(i-1)})}{\sum_{j=1}^{K} \alpha_j^{(i-1)} \mathcal{N}(\vec{X}_m, \vec{\mu}_j^{(i-1)}, \Sigma_j^{(i-1)})}$$

Maximization Step (M)

Estimate the parameters $\vec{v}^{(i)}$ using $h^{(i)}(k,m)$

Mass: $M_k^{(i)} \leftarrow \sum_{m=1}^{N} h^{(i)}(k,m)$ (Note: M_k is a real)

Probability: $\alpha_k^{(i)} \leftarrow \frac{M_k^{(i)}}{M} = \frac{1}{M}\sum_{m=1}^{M} h^{(i)}(k,m)$

Mean: $\vec{\mu}_k^{(i)} \leftarrow \frac{1}{M_k^{(i)}}\sum_{m=1}^{M} h^{(i)}(k,m)\vec{X}_m$

Covariance: $\Sigma_k^{(i)} \leftarrow \frac{1}{M_k^{(i)}}\sum_{m=1}^{M} h^{(i)}(k,m)(\vec{X}_m - \vec{\mu}_k^{(i)})(\vec{X}_m - \vec{\mu}_k^{(i)})^T$

Convergence Criteria

The quality metric is the Log-likelihood of the probability of obtaining the data given the parameters.

$$Q^{(i)} = \ln\{p(\{\vec{X}_n\} \mid \vec{v}^{(i)})\} = \sum_{m=1}^{M} \ln\left\{\sum_{j=1}^{K} \alpha_j^{(i)} \mathcal{N}(\vec{X}_m \mid \mu_j^{(i)}, \Sigma_j^{(i)})\right\}$$

It can be shown that, for EM, the log likelihood will converge to a stable maximum. The change in Q will monotonically decrease. This can be used to define a halting condition:

If $\Delta Q = Q^{(i)} - Q^{(i\text{-}1)}$ is less than a threshold, halt.

Using Gaussian Mixture Models with Baye's Rule

There are many ways that Gaussian Mixture Models can be used with Bayes Rule.

Supervised Learning:

Given a set of M Training samples $\left\{\vec{X}_m\right\}$ with ground-truth indicator variables $\{y_m\}$ telling the class for each sample, most classic techniques assumed that each class was represented by a single Normal (or Gaussian) density function. This is often an oversimplification that can lead to poor classification performance.

A Gaussian Mixture Model can be used to represent an arbitrarily complex density function for each class as a sum of Gaussians.

$$p(\vec{X};\vec{v}_k) = \sum_{n=1}^{N_k} \alpha_n \mathcal{N}(\vec{X};\vec{\mu}_n,\Sigma_n)$$

Where $\vec{v}_k = \{N,\alpha_1,\vec{\mu}_1,\Sigma_1,...,\alpha_N,\vec{\mu}_N,\Sigma_N)$ are the parameters for the density function for the kth class.

The most likely class for an observation x can be estimated using:

$$P(\omega_k \mid \vec{X}) = \frac{p(\vec{X} \mid \omega_k)}{p(\vec{X})} P(\omega_k) = \frac{p(\vec{X} \mid \omega_k) P(\omega_k)}{\sum_{j=1}^{K} p(\vec{X} \mid \omega_j) P(\omega_j)} = \frac{p(\vec{X};\vec{v}_k) P(\omega_k)}{\sum_{j=1}^{K} p(\vec{X};\vec{v}_j) P(\omega_j)} = \frac{\sum_{n=1}^{N_k} \alpha_n \mathcal{N}(\vec{X};\vec{\mu}_n,\Sigma_n) P(\omega_k)}{\sum_{j=1}^{K} \sum_{n=1}^{N_k} \alpha_n \mathcal{N}(\vec{X};\vec{\mu}_n,\Sigma_n) P(\omega_j)}$$

In this case EM (or K-means) make it possible to estimate the parameter vectors, $\vec{v}_k$ for each class.

Unsupervised Learning:

Given a set of M Training samples $\left\{\vec{X}_m\right\}$ without ground-truth indicator variables, then we can use EM (or K-means) to estimate the source for each sample. If we assume that each source is a independent class with a Normal density function, then we can use each component of the mixture model as a class.

In this case $p(\vec{X} \mid \omega_k) = \alpha_k \mathcal{N}(\vec{X};\vec{\mu}_k,\Sigma_k)$ and $P(\omega_k \mid \vec{X}) = \dfrac{p(\vec{X} \mid \omega_k)}{p(\vec{X})} P(\omega_k) = \dfrac{\alpha_k \mathcal{N}(\vec{X};\vec{\mu}_k,\Sigma_k)}{\sum_{j=1}^{K} \alpha_j \mathcal{N}(\vec{X};\vec{\mu}_j,\Sigma_j)}$