**Gaussian Mixture Model (GMM)**

This document explains GMM theory. Read the sklearn tutorial on using GMM first.

Probability density of the data point under each Gaussian component is given by the multivariate Gaussian distribution formula:

is the covariance matrix of the Gaussian component

|Σ| denotes the determinant of the covariance matrix Σ,

denotes the transpose operation, and

denotes the inverse operation.

is the data point,

is the mean vector of the Gaussian component,

is the number of dimensions (features) in your data,

To understand this, consider the case of n=1, then is a 1x1 matrix with the value of , the equation is essentially Gaussian distribution:

is standard deviation. A given describes a **range** around the mean .

In the case of n=2 and is 2x2 ***diagonal***, then Each value in is the of that dimension. A given set of describes an ellipse around the mean .

Let , , , the equation becomes:

In the case of n=2 and is a ***full matrix***, the ellipse is tilted.

Let the eigenvalue, eigenvector be , . The eigenvalues are standard deviations . are the two axes of the ellipse.

Assume , are sorted by large to small, then is the eigenvector corresponding to the largest eigenvector. The major axis of ellipse is tiled by :

To get the range of , simply let the covariance matrix be . This is because the axes of ellipse are scaled by 3, while and are not changed. According to eigen decomposition:

In the case of n=3, this would be an ellipsoid.

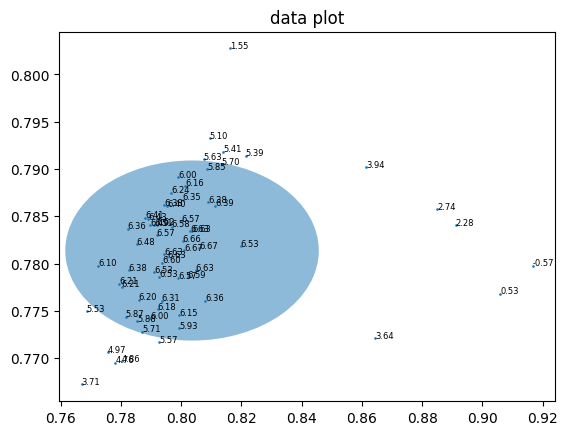
**Implementation**

Basic concepts using Iris dataset (3 classes, 0,1,2)

* n\_components: the number of classes.
* Prediction: classified results, represented by class id (int)
* Prediction Probability: the probability of a sample belongs to a class
  + 3 values corresponding to 3 classes.
* Sample Score: the log-likelihood of a sample
* Overall Score: the per-sample average log-likelihood of given data

If n\_components is 1. Something weird happens:

* Prediction is always 0, because only 1 class
* Prediction Probability is always 1: because no other classes compete with class 0
* Sample Score represents how close a point is to the mean of GMM, shown below.



But then, how to I determine if a point falls within sigma values?

* Predetermine the 3-sigma threshold using sigma values, this gives a range (1d), an ellipse (2d), or an ellipsoid (3d). Then check if a point X falls within such shapes.
* Alternatively, in a 1D setting, one can calculate Cumulative probability function (CDF), which looks like an S curve, and determine exactly how close a point X compared to mean , this gives the probability value between 0 and 1.
  + But the problem is CDF is cannot be expressed in terms of elementary functions, instead it introduces an error function erf(x). Numerical method or lookup table is required to calculate this function.
  + To expand into 2D and 3D, joint CDF is required, adding another layer of complexity.

Method to determine if a point falls within a ellipse.

**Implementation of rotation**

3D verification

Matlab has a function ellipsoid and a way to rotate it by axis. It can plot the points and ellipsoid easily.

To verify

Process to determine if a point falls within 3D GMM

GMM model (sklearn) -> mean and cov -> mean, sigma, rotation -> if X falls within such ellipsoid.

Matlab can calculate if a point falls within a 3D GMM of x sigma?

2D verification.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

