Unsupervised Machine Learning with Python

By Satish Reddy

Section 1: Contours of Normal Distribution PDF in 2 Dimensions

Recall that the normal distribution pdf is given by:

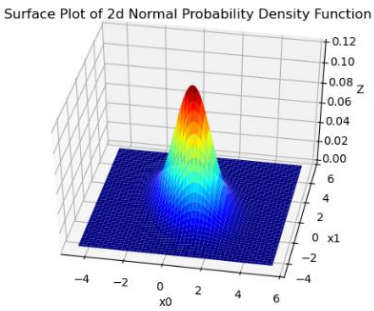
$$N(X, \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} e^{-\frac{1}{2}(X-\mu)^T \Sigma^{-1} (X-\mu)}$$

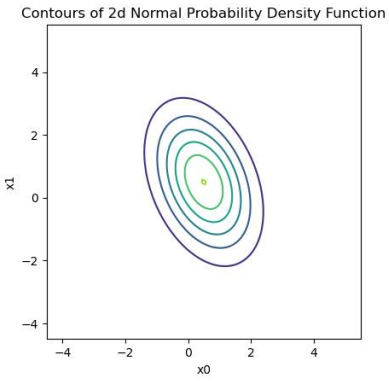
where μ is the mean, Σ is the covariance matrix, and $|\Sigma|$ is the determinant of the covariance matrix and d is the dimension.

Let's consider the 2-dimensional case, where:

$\mu = \begin{bmatrix} \mu_0 \\ \mu_1 \end{bmatrix}$ and $X = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix}$

The following figure shows the surface plot for N and the contours (curves of constant N which are ellipses) in the x0-x1 plane.





The goal of this section is to determine the formula for the contours. This information will be used to help visualize the Gaussian Mixture Model clustering algorithm.

Let us determine the contour (ellipse) where

$$N(X, \mu, \Sigma) = \frac{\phi}{\sqrt{(2\pi)^d |\Sigma|}} e^{-\frac{1}{2}(X-\mu)^T \Sigma^{-1} (X-\mu)} = c$$

Here ϕ is weight (constant) and for 2 dimensions d=2.

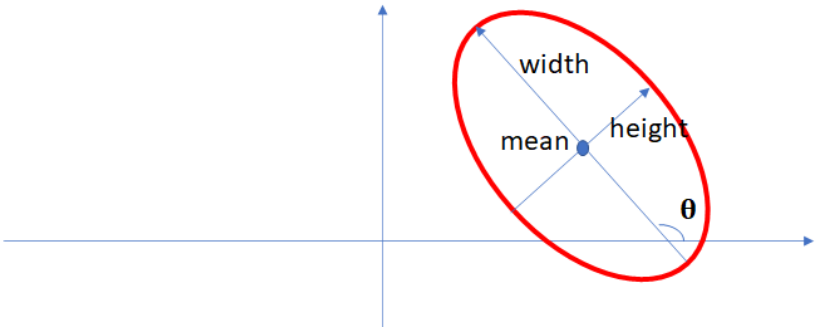
Solving for $(X-\mu)^T \Sigma^{-1} (X-\mu)$ we get:

$$(X-\mu)^T \Sigma^{-1} (X-\mu) = -2\log\left[c2\pi\sqrt{|\Sigma|}/\phi\right]$$

Since the covariance matrix is symmetric and is assumed to have positive eigenvalues, the eigenvalue decomposition and singular value decomposition of Σ are the same:

$$\Sigma = [u_0 \quad u_1]\begin{bmatrix}\sigma_0 & 0\\ 0 & \sigma_1\end{bmatrix}\begin{bmatrix}u_0^T\\ u_1^T\end{bmatrix}$$

Without going through all the details, it can be shown that the long dimension of the ellipse, denoted width in the figure below, is in the u_0 direction and the short dimension of the ellipse, denoted height in the figure below, is in the u_1 direction. Here θ is angle made by u_0 with horizontal axis.



For example, if we set $X = \alpha\sqrt{\sigma_0}u_0 + \mu$, where $\alpha\sqrt{\sigma_0}$ is the half width, then using the SVD formula to get inverse of Σ we have (using fact that $u_0^T u_0$ is length squared of u_0 which is 1 by definition):

$$(X-\mu)^T\Sigma^{-1}(X-\mu) = \alpha^2 = -2\log\left[c2\pi\sqrt{|\Sigma|}/\phi\right]$$

It follows that width axis is parallel to u_0 and

$$width = 2\alpha\sqrt{\sigma_0} = 2\sqrt{-2\log\left[c2\pi\sqrt{|\Sigma|}/\phi\right]}\sqrt{\sigma_0}$$

Similarly it can be shown that the height axis is parallel to u_1 and

$$height = 2\sqrt{-2\log\left[c2\pi\sqrt{|\Sigma|}/\phi\right]}\sqrt{\sigma_1}$$

Here θ is angle made by u_0 with horizontal axis. Hence:

$$u_0 = \begin{bmatrix}u_{00}\\ u_{10}\end{bmatrix} \; and \; \theta = \arctan\left(u_{10}/u_{00}\right)$$

The mean is

$$\mu = \begin{bmatrix}\mu_0\\ \mu_1\end{bmatrix}$$

The above formulas give us all the information required to determine and plot the contours of the normal distribution pdf in 2 dimensions.

Note that these ideas extend to higher dimensions, where the contours are multi-dimensional ellipsoids. The maximum extent of the ellipsoid in direction k is proportional to the square root of the k'th singular value. The direction k is parallel to the k'th column of U from the SVD.

Section 2: Expectation Maximization for the Gaussian Mixture Model

This section provides a derivation of the expectation maximization for the Gaussian Mixture Model. Assume data points $X_0,\ X_1,\ X_2, \ldots, X_{M-1}$ in d dimensions. Assume that there are K clusters, where cluster k denoted S_k, has mean μ_k, covariance matrix Σ_k, and weight ϕ_k. Note that weights satisfy $\phi_0 + \cdots + \phi_{K-1} = 1$. The probability density function for the mixture of Gaussians is:

$$P(X) = \sum_{k=0}^{K-1} \phi_k N(X, \mu_k, \Sigma_k)$$

The conditional probability that a data point is in cluster k given X is

$$P(S_k|X) = \frac{\phi_k N(X, \mu_k, \Sigma_k)}{\sum_{k=0}^{K-1} \phi_k N(X, \mu_k, \Sigma_k)}$$

The joint probability density function for $X_0, \ldots, X_{M-1}$ is given by likelihood function:

$$P(X_0, \ldots, X_{M-1}) = \prod_{i=0}^{M-1} P(X_i) = \prod_{i=0}^{M-1} \sum_{k=0}^{K-1} \phi_k N(X_i, \mu_k, \Sigma_k)$$

Maximum Likelihood Estimation attempts to find the parameters (values of means $\{\mu_k\}$, covariances $\{\Sigma_k\}$, and weights $\{\phi_k\}$) that has the maximum likelihood for the given data points. Following convention, we will do this by maximizing the log likelihood function:

$$L = \log P(X_0, \ldots, X_{M-1}) = \sum_{i=0}^{M-1} \log \left[\sum_{k=0}^{K-1} \phi_k N(X_i, \mu_k, \Sigma_k) \right]$$

subject to the constraint $\phi_0 + \cdots + \phi_{K-1} = 1$.

For the multi-dimensional case, one must maximize:

$$L' = \sum_{i=0}^{M-1} \log \left[\sum_{k=0}^{K-1} \phi_k N(X_i, \mu_k, \Sigma_k) \right] + \lambda(\phi_0 + \cdots + \phi_{K-1} - 1)$$

where λ is the Lagrange multiplier.

The multi-dimensional normal distribution probability density function in d dimensions is:

$$N(X, \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} e^{-\frac{1}{2}(X-\mu)^T \Sigma^{-1} (X-\mu)}$$

where X and μ are d-dimensional column vectors and the covariance Σ is dxd dimensional matrix. Note that |Σ| is the determinant.

Here are basic assumptions:

(A) The covariance matrix is symmetric

(B) The covariance matrix is invertible (determinant is non-zero)

Some basic matrix results for matrices:

(C) $(A^T)^{-1}$=$(A^{-1})^T = \mathrm{A}^{-T}$

Here are some results for derivatives (gradients) of functions of vectors and matrices. These results are taken from the Matrix Cookbook (see citation in the References section). Consider:

$$F = (X-\mu)^T\Sigma^{-1}(X-\mu)$$

First let's compute the gradient with respect to μ (see formula 86 in Matrix Cookbook):

$$\nabla_\mu F = -2\Sigma^{-1}(X-\mu)$$

Let's now compute the gradient with respect to the Σ (see formula 61 in Matrix Cookbook):

$$\nabla_\Sigma F = -\Sigma^{-T}(X-\mu)(X-\mu)^T\Sigma^{-T}$$

Let us define $G = |\Sigma|$ (determinant). The gradient is (see formula 49 in Matrix Cookbook):

$$\nabla_\Sigma G = |\Sigma|\Sigma^{-T}$$

Now we can compute the appropriate gradients of the normal pdf:

$$\nabla_\mu N(X,\mu,\Sigma) = \Sigma^{-1}(X-\mu)N(X,\mu,\Sigma)$$

$$\nabla_\Sigma N(X,\mu,\Sigma) = -\frac{1}{2}\Sigma^{-T}N(X,\mu,\Sigma) + \frac{1}{2}\Sigma^{-T}(X-\mu)(X-\mu)^T\Sigma^{-T}N(X,\mu,\Sigma)$$

We now have the tools to compute the gradients of L' and set them to 0. The equations are:

$$\nabla_{\mu_k} L' = 0 \quad k = 0,\dots,K-1$$

$$\nabla_{\Sigma_k} L' = 0 \quad k = 0,\dots,K-1$$

$$\frac{\partial L'}{\partial \phi_k} = 0 \quad k = 0,\dots,K-1$$

$$\frac{\partial L'}{\partial \lambda} = 0$$

Let's start with he gradient with respect to the means:

$$\nabla_{\mu_k} L' = \sum_{i=0}^{M-1} \frac{\phi_k \nabla_{\mu_k} N}{\sum_{k=0}^{K-1}\phi_k N(X_i,\mu_k,\Sigma_k)} = \sum_{i=0}^{M-1} \frac{\phi_k \Sigma^{-1}(X_i-\mu_k)N(X_i,\mu_k,\Sigma_k)}{\sum_{k=0}^{K-1}\phi_k N(X_i,\mu_k,\Sigma_k)} \quad k = 0,\dots,K-1$$

Let us define:

$$\gamma_{ki} = \frac{\phi_k N(X_i, \mu_k, \Sigma_k)}{\sum_{k=0}^{K-1} \phi_k N(X_i, \mu_k, \Sigma_k)} \; and \; M_k = \sum_{i=0}^{M-1} \gamma_{ki}$$

Rewriting the gradient and setting to 0:

$$\nabla_{\mu_k} L' = \sum_{i=0}^{M-1} \gamma_{ki} \, \Sigma_k^{-1} (X_i - \mu_k) = 0 \;\; k = 0, \dots, K-1$$

Multiplying by Σ_k and solving for μ_k, we have

$$\mu_k = \frac{\sum_{i=0}^{M-1} \gamma_{ki} X_i}{\sum_{i=0}^{M-1} \gamma_{ki}} = \frac{1}{M_k} \sum_{i=0}^{M-1} \gamma_{ki} X_i \quad k = 0, \dots, K-1$$

For the gradients with respect to the covariance matrices, we have:

$$\nabla_{\Sigma_k} L' = \sum_{i=0}^{M-1} \frac{\phi_k \nabla_{\Sigma_k} N}{\sum_{k=0}^{K-1} \phi_k N(X_i, \mu_k, \Sigma_k)} = \sum_{i=0}^{M-1} \gamma_{ki} \left(-\frac{1}{2} \Sigma_k^{-T} + \frac{1}{2} \Sigma_k^{-T} (X_i - \mu_k)(X_i - u_k)^T \Sigma_k^{-T}\right)$$

Setting the gradient to 0 and multiplying by $2\Sigma_k^T$, we have:

$$\sum_{i=0}^{M-1} \gamma_{ki} \left(-1 + (X_i - \mu_k)(X_i - u_k)^T \Sigma_k^{-T}\right) = 0 \quad k = 0, \dots, K-1$$

Since the covariance matrix is symmetric $\Sigma_k^{-T} = \Sigma_k^{-1}$. Solving this last equation for Σ_k, we have:

$$\Sigma_k = \frac{1}{M_k} \sum_{i=0}^{M-1} \gamma_{ki} (X_i - \mu_k)(X_i - \mu_k)^T \;\; k = 0, \dots, K-1$$

The derivative with respect to ϕ_k is:

$$\frac{\partial L'}{\partial \phi_k} = \sum_{i=0}^{M-1} \frac{N(X_i, \mu_k, \Sigma_k)}{\sum_{k=0}^{K-1} \phi_k N(X_i, \mu_k, \Sigma_k)} + \lambda = \frac{1}{\phi_k} \sum_{i=0}^{M-1} \frac{\phi_k N(X_i, \mu_k, \Sigma_k)}{\sum_{k=0}^{K-1} \phi_k N(X_i, \mu_k, \Sigma_k)} + \lambda$$

Setting the gradient to 0, we have:

$$\phi_k = -\frac{1}{\lambda} \sum_{i=0}^{M-1} \frac{\phi_k N(X_i, \mu_k, \Sigma_k)}{\sum_{k=0}^{K-1} \phi_k N(X_i, \mu_k, \Sigma_k)} = -\frac{1}{\lambda} \sum_{i=0}^{M-1} \gamma_{ki}$$

Setting $\phi_0 + \cdots + \phi_{K-1} = 1$, we have

$$-\frac{1}{\lambda} \sum_{k=0}^{K-1} \sum_{i=0}^{M-1} \gamma_{ki} = 1$$

Note that

$$\gamma_{ki} = \frac{\phi_k N(X_i, \mu_k, \Sigma_k)}{\sum_{k=0}^{K-1} \phi_k N(X_i, \mu_k, \Sigma_k)}$$

It is straightforward to show

$$\sum_{k=0}^{K-1} \gamma_{ki} = 1$$

Based on the definition, is relatively straightforward to show that

$$\sum_{k=0}^{K-1} \sum_{i=0}^{M-1} \gamma_{ki} = \sum_{i=0}^{M-1} \sum_{k=0}^{K-1} \gamma_{ki} = M$$

Hence, λ=-M and

$$\phi_k = \frac{1}{M} \sum_{i=0}^{M-1} \gamma_{ki} = \frac{M_k}{M}$$

Summarizing, we have for k = 0,…,K-1:

$$\gamma_{ki} = \frac{\phi_k N(X_i, \mu_k, \Sigma_k)}{\sum_{k=0}^{K-1} \phi_k N(X_i, \mu_k, \Sigma_k)} \; and \; M_k = \sum_{i=0}^{M-1} \gamma_{ki}$$

$$\phi_k = \frac{M_k}{M}$$

$$\mu_k = \frac{1}{M_k} \sum_{i=0}^{M-1} \gamma_{ki} X_i$$

$$\Sigma_k = \frac{1}{M_k} \sum_{i=0}^{M-1} \gamma_{ki} (X_i - \mu_k)(X_i - \mu_k)^T$$

We cannot solve the equations directly for μ_k, Σ_k and $\phi_k$because these variables appear on both sides of the equation. The Expectation Maximization algorithm attempts to solve the above equations using an iterative approach:

(1) Initialize: means $\{\mu_k\}$, variance matrices $\{\Sigma_k\}$, and weights $\{\phi_k\}$
One can randomly choose the means $\{\mu_k\}$ from among the data points randomly or using the K Means ++ approach. The weights are typically all set to the value ϕ_k=1/K, and the covariance matrices are all set to

$$\Sigma_k = \frac{1}{M} \sum_{i=0}^{M-1} (X_i - \mu)(X_i - \mu)^T$$

where

$$\mu = \frac{1}{M} \sum_{i=0}^{M-1} X_i$$

(2) Expectation Step: compute $\{\gamma_{ki}\}$
(3) Maximization Step: given $\{\gamma_{ki}\}$ determined in the Expectation Step, compute means $\{\mu_k\}$, variance matrices $\{\Sigma_k\}$, and weights $\{\phi_k\}$.

Steps (2) and (3) are repeated until convergence. Let $\mu_k^{(n)}$ denote the mean for cluster k at the n'th iteration. In this course, we say that convergence has occurred when the maximum distance between current and previous means is less than a specified tolerance ε.

$$max_k dist\left(\mu_k^{(n)}, \mu_k^{(n-1)}\right) < \varepsilon$$

Section 4: Gaussian Mixture Model Spherical Approach

For the spherical approach it is assumed that the covariance matrix for each component of the mixture is a constant diagonal (dxd) matrix of the form:

$$\Sigma_k = \begin{bmatrix} v_k & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & v_k \end{bmatrix},$$

where v_k is the variance.

For a constant diagonal covariance matrix, the normal distribution pdf simplifies to

$$N(X, \mu, \Sigma) = N(X, \mu, \mathrm{v}) = \frac{1}{\sqrt{(2\pi v)^d}} e^{-\frac{1}{2v}(X-\mu)^T(X-\mu)}$$

(Note the determinant of a constant diagonal dxd matrix is simply $|\Sigma| = v^d$.) It is more efficient to use this last formula than to actually input the constant diagonal covariance matrix into the general normal pdf formula.

The log likelihood function in this case is:

$$L = \log P(X_0, \ldots, X_{M-1}) = \sum_{i=0}^{M-1} \log\left[\sum_{k=0}^{K-1} \phi_k N(X_i, \mu_k, v_k)\right].$$

The goal is to maximize L subject to the constraint that weights satisfy $\phi_0 + \cdots + \phi_{K-1} = 1$

Without re-performing the derivation, here is a summary of the expectation maximization algorithm for the spherical case:

Initialization Step:

Choose means $\{\mu_k\}$ and weights $\{\phi_k\}$ using the same approach as in the full covariance matrix case. All the variances are set to:

$$v_k = \frac{1}{dM} \sum_{i=0}^{M-1} (X_i - \mu)^T (X_i - \mu)$$

where μ is the mean of the entire dataset

$$\mu = \frac{1}{M} \sum_{i=0}^{M-1} X_i$$

Expectation Step:

$$\gamma_{ki} = \frac{\phi_k N(X_i, \mu_k, v_k)}{\sum_{k=0}^{K-1} \phi_k N(X_i, \mu_k, v_k)} \; and \; M_k = \sum_{i=0}^{M-1} \gamma_{ki}$$

Maximization Step:

$$\phi_k = \frac{M_k}{M}$$

$$\mu_k = \frac{1}{M_k} \sum_{i=0}^{M-1} \gamma_{ki} X_i$$

$$v_k = \frac{1}{dM_k} \sum_{i=0}^{M-1} \gamma_{ki} (X_i - \mu_k)^T (X_i - \mu_k)$$

Section 4: References

Kaare Brandt Petersen and Michael Syskind Pedersen, *The Matrix Cookbook*, Version November 15, 2012. https://www.math.uwaterloo.ca/~hwolkowi/matrixcookbook.pdf