Gaussian Mixture Model

Boris Kodner, NDG UBI Team

Motivation


WANT TO HEAR MY PRESENTATION?
NOT MOTIVATED
memegenerator.net

K-Means Drawbacks

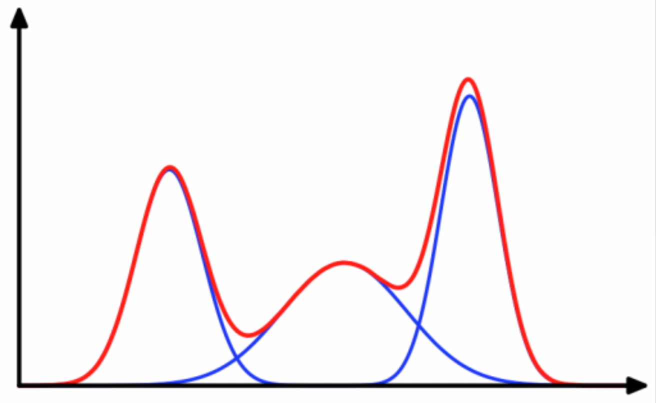
- Represents only spherical clusters
- Every point in the cluster has the same value no matter how far it is from the center
- Every point is classified, no outliers
- Iteration based process, may converge to local minimum

So why do everybody use it?

- Simple and Accessible

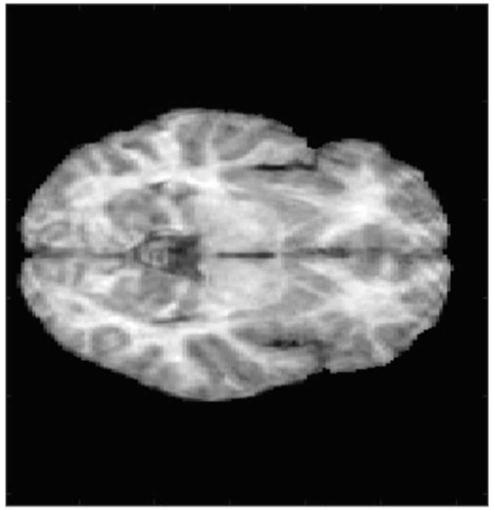
Gaussian Mixture Model

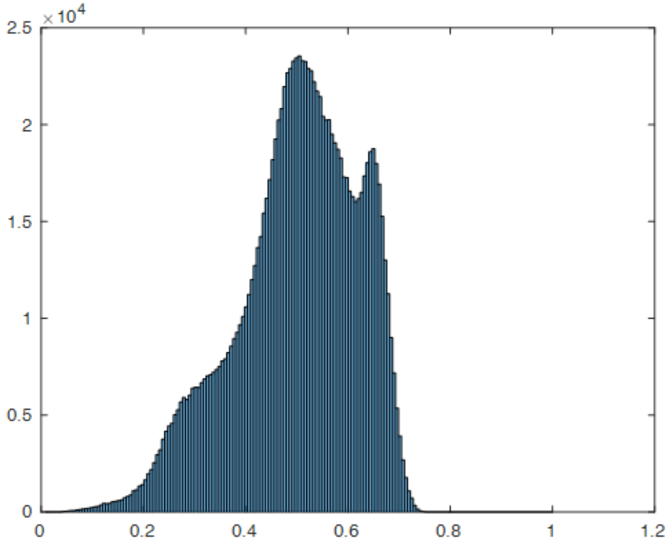
- Lets consider a probabilistic generative model that assumes the data can be represented as a linear combination of multivariate Gaussian distribution:
- $\mathrm{P}(\boldsymbol{x}) = \sum_{k=1}^{K_{max}} \omega_k \, P_{gauss}(\boldsymbol{x}|\boldsymbol{\mu_k}, \boldsymbol{\Sigma_k})$
  - $\boldsymbol{x} \in \mathbb{R}^D; \boldsymbol{x} = [x_1, x_2, \ldots, x_N]^T$
  - $\{\omega_k\}$ - mixture proportions
  - $\boldsymbol{\mu_k}, \boldsymbol{\Sigma_k}$- k^{th} component's mean and covariance matrix respectively

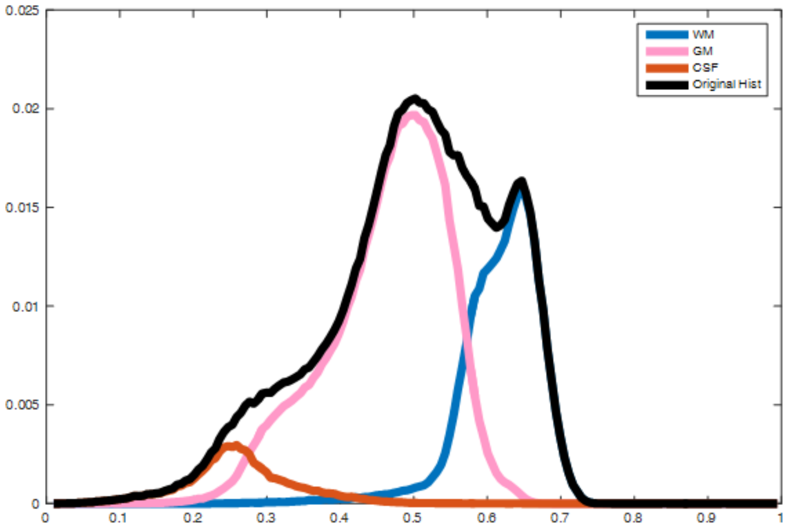


- Given a data set, $\{\boldsymbol{x}_l\}_{l=1}^{L}$, we would like to fit a GMM by estimating :
  - $\boldsymbol{\Theta} = \{\boldsymbol{\mu_1}, \ldots, \boldsymbol{\mu_k}, \boldsymbol{\Sigma_1}, \ldots, \boldsymbol{\Sigma_k}, \omega_1, \ldots, \omega_k\}$
  - Parameters estimation is done using expectation maximization (EM) algorithm.

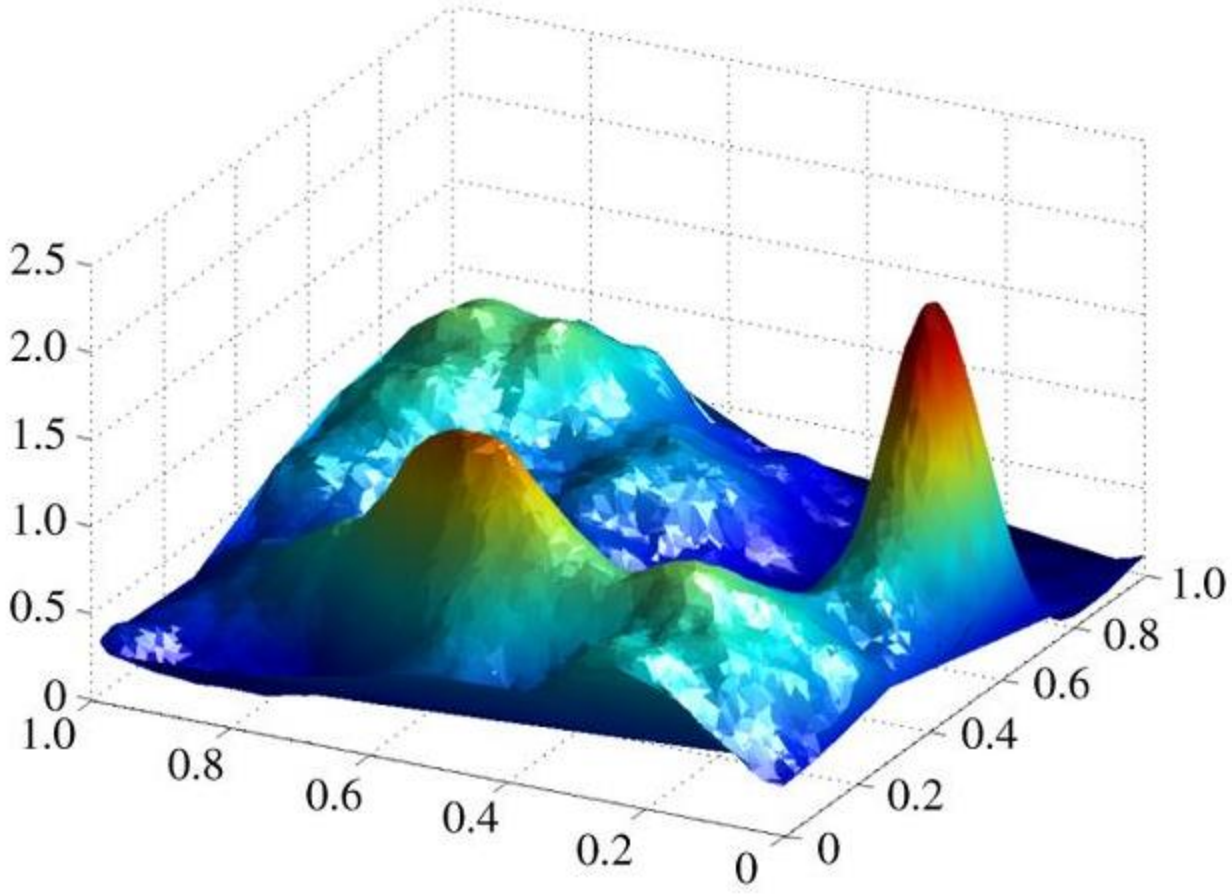
GMM 1D Example – MRI Data



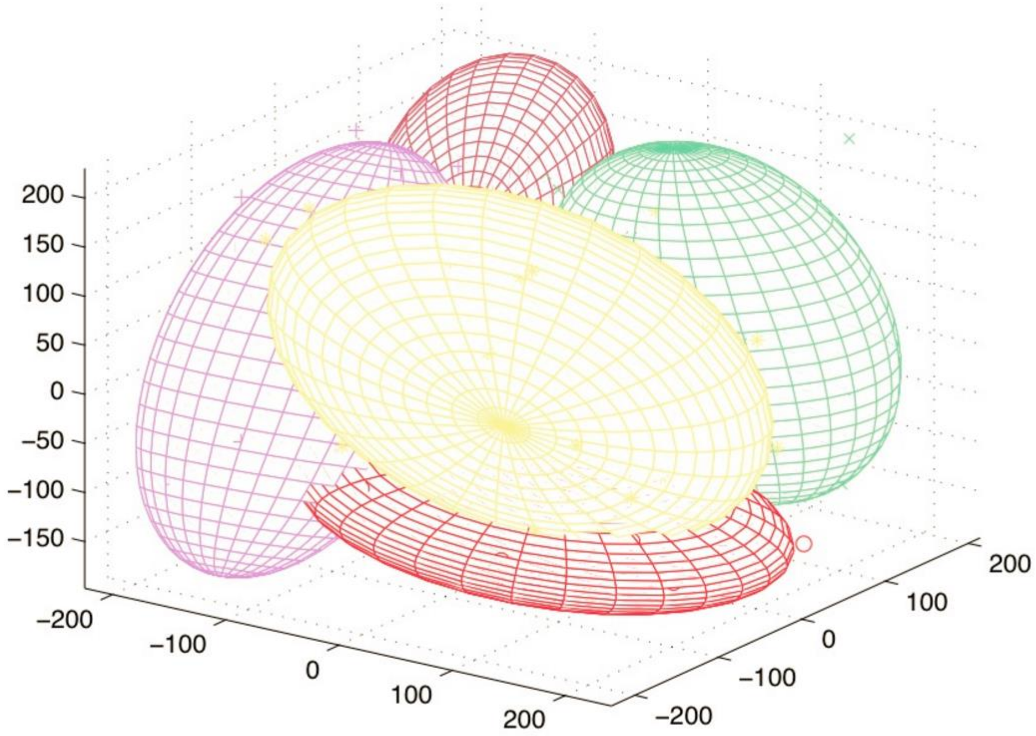




GMM 2D Example – Tsunami Simulator

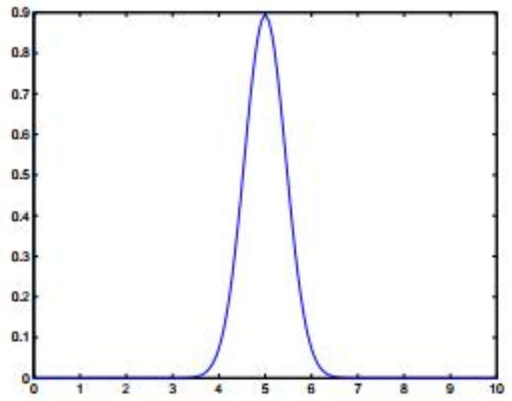


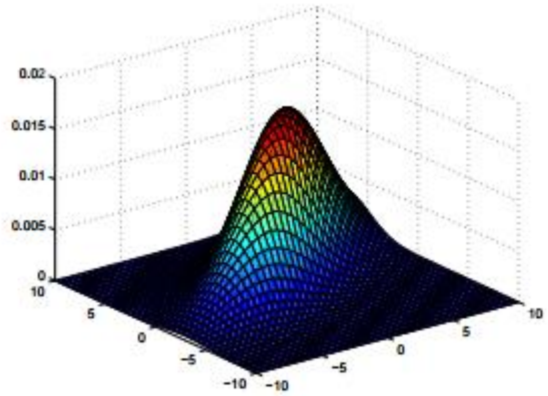
GMM 1D and 2D and 3D Examples



Gaussian Kernel – What is it good for?!

- $P_{gauss}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_\mathbf{k}) = \frac{1}{(2\pi|\boldsymbol{\Sigma}_k|)^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu_k)'\Sigma_\mathbf{k}^{-1}(\mathbf{x}-\mu_k)}$





- Easy to work with – just “log” it out.
- Can be represented using only the mean value and the variance.
- It’s useful to think that most of the noise in real world applications is modeled using Gaussian density function (Although it actually isn’t!!).

EM Algorithm Overview

The EM algorithm tries to find the Maximum Likelihood Estimator (MLE) of the marginal likelihood:

$$\widehat{\boldsymbol{\theta}} = \arg\max_{\theta} f(\boldsymbol{X}|\boldsymbol{\theta})$$

Under the missing data assumption:

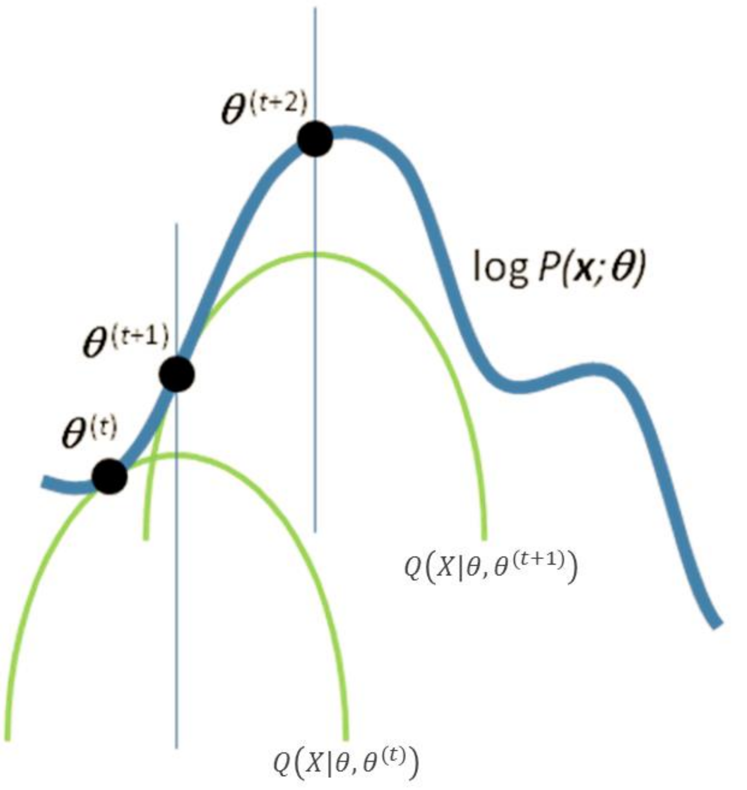
X- *visible data i.e. the samples*

Y- *latent unseen data i.e. random variable* $y_i = k$ *with probability* ω_k

EM Algorithm Logic

- *Initialization:* $\theta_0, \quad t = 1$
- *Expectation step (E step):*
  - *Calculate the expected value of the log likelihood:*
  - $Q\left(X|\theta, \theta^{(t+1)}\right) = E_{Y|X;\theta^{(t-1)}}\left[\log\left(P(X, Y|\theta)\right)\right]$
- *Maximization step (M step):*
  - *Find the parameter that maximizes this quantity:*
  - $\theta^{(t)} = \arg\max_{\theta} Q\left(X|\theta, \theta^{(t-1)}\right)$
- *If* $\left|\frac{\log\left(f\left(X|\theta^{(t)}\right)\right) - \log\left(f\left(X|\theta^{(t-1)}\right)\right)}{\log\left(f\left(X|\theta^{(t-1)}\right)\right)}\right| < \epsilon$ then stop

  else: $t = t + 1$ and go to E-Step



EM Algorithm For Convergence

- Given our current estimate of the parameters $\theta^{(t)}$:
  - E–step:
    - $Q(X|\theta,\theta^{(t-1)}) = E_{Y|X;\theta^{(t-1)}}[\log(P(X,Y|\theta))] = \sum_i \overbrace{P(y_i = k|x_i;\theta^{(t-1)})}^{T_{ik}^{(t)}} \log(P(x_i, y_i = k|\theta))$

$$= \sum_{i=1}^{n}\sum_{k=1}^{K} T_{ik}^{(t)} \left[\log(\omega_k) - \frac{1}{2}\log(|\Sigma_k|) - \frac{1}{2}(x_i - \mu_k)^T \Sigma_k^{-1} (x_i - \mu_k) - \frac{d}{2}\log(2\pi)\right]$$

  - M-step:
    - $\omega_k^{(t)} = \frac{1}{n}\sum_{i=1}^{n} T_{ik}^{(t-1)}$
    - $\mu_k^{(t)} = \frac{\sum_{i=1}^{n} T_{ik}^{(t-1)} x_i}{\sum_{i=1}^{n} T_{ik}^{(t-1)}}$
    - $\Sigma_k^{(t)} = \frac{\sum_{i=1}^{n} T_{ik}^{(t-1)} \left(x_i - \mu_k^{(t)}\right)\left(x_i - \mu_k^{(t)}\right)^T}{\sum_{i=1}^{n} T_{ik}^{(t-1)}}$

EM Algorithm E-Step Meaning

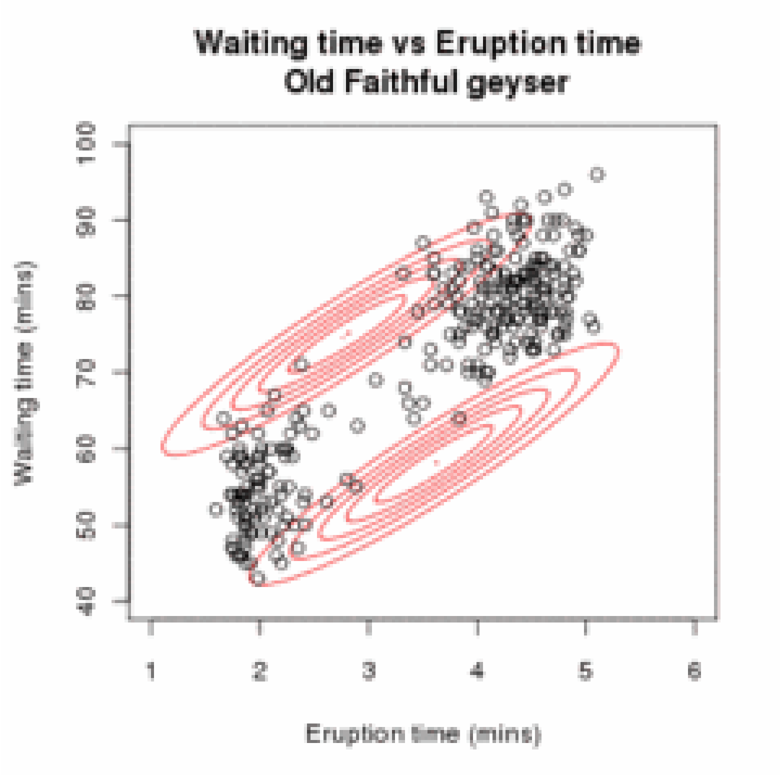
- $P(y_i = k|x_i;\theta) = \{bayes\} = \frac{P(y_i=k)P(x_i|y_i = k;\theta)}{\sum_k P(y_i=k)P(x_i|y_i = k;\theta)} = \frac{\omega_k P(x_i|y_i = k;\theta)}{\sum_k \omega_k P(x_i|y_i = k;\theta)} = T_{ik}^{(t)}$

- What is the meaning of $T_{ik}^{(t)}$?

- Unlike K-Means where “Hard Labels” are used, The GMM-EM algorithm provides a framework for “Soft Labels”.

- $T_{ik}^{(t)}$ Represent the “belonging” of each sample x_i to the Gaussian k.

EM Algorithm Example



Clustering After EM

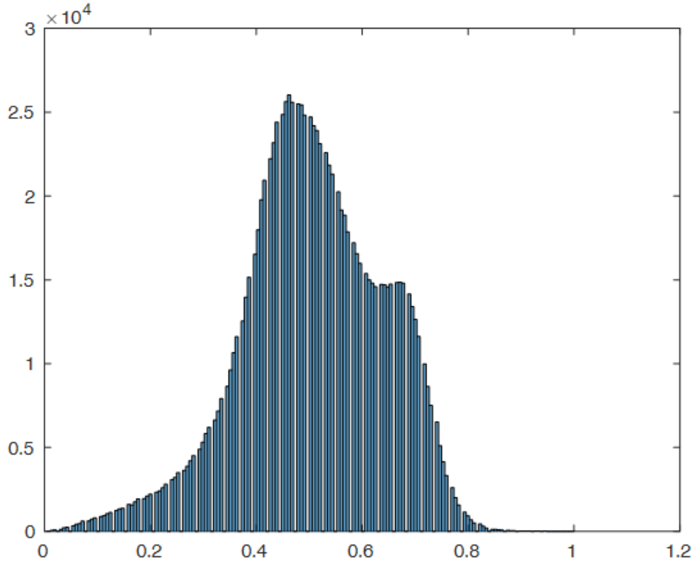
After the algorithm is converged, we can either cluster each point to a certain cluster by calculating:

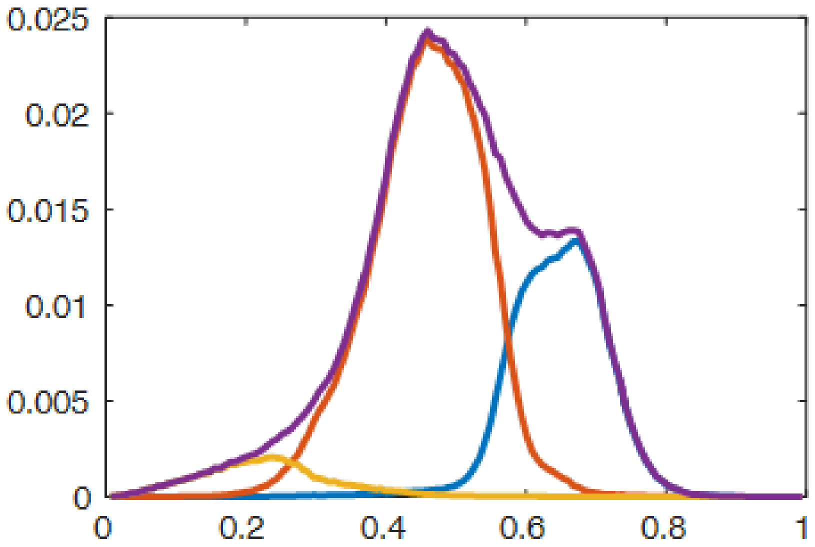
$$p(k|x_i) = \frac{\omega_k P(x_i|\theta)}{\sum_k \omega_k P(x_i|\theta)}$$

and find the cluster which gives the highest probability.

Or use the gain probabilities of each of the points belonging to each cluster according to our needs.

GMM Drawbacks





If you watch the purple histogram, you will not be able to detect the 3rd Gaussian. What can we do?

Goldberger, Jacob, and Hayit Greenspan. "Context-based segmentation of image sequences." *IEEE transactions on pattern analysis and machine intelligence* 28.3 (2006): 463-468.

MAP-EM

By "injecting" prior knowledge about the properties of the Gaussian, we can improve the EM algorithm and push it to converge in the right place.

Lets define a parameter β that will decide the "amount of injection" of the prior knowledge:

$$\beta = \alpha \cdot n$$
$$\beta_k = \omega_{k_{prior}} \cdot \beta$$

The augmented EM formulas are:

$$\omega_k^{(t)} = \frac{\sum_{i=1}^{n} T_{ik}^{(t-1)} + \beta_k}{n + \beta}$$

$$\mu_k^{(t)} = \frac{\sum_{i=1}^{n} T_{ik}^{(t-1)} x_i + \beta_k \mu_{k_{prior}}}{\sum_{i=1}^{n} T_{ik}^{(t-1)} + \beta_k}$$

$$\Sigma_k^{(t)} = \frac{\sum_{i=1}^{n} T_{ik}^{(t-1)} \left(x_i - \mu_k^{(t)}\right)\left(x_i - \mu_k^{(t)}\right)^T + \beta_k \left(\left(\mu_k^{(t)} - \mu_{k_{prior}}\right)^2 + \Sigma_{k_{prior}}\right)}{\sum_{i=1}^{n} T_{ik}^{(t-1)} + \beta_k}$$