GMM算法

一、E-step：更新后验概率

$$
\begin{aligned}
\gamma(z_k) \equiv p(z_k = 1|\mathbf{x}) &= \frac{p(z_k = 1)p(\mathbf{x}|z_k = 1)}{\sum_{j=1}^{K} p(z_j = 1)p(\mathbf{x}|z_j = 1)} \\
&= \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n|\mu_j, \Sigma_j)}
\end{aligned}
$$

```python
    def get_expectation(self, data):
        """
        更新posteriori(后验概率)
        :param data: 输入的数据点
        :return: 更新后的后验概率
        """
        for k in range(self.K):
            # 计算高斯分布
            self.posteriori[k] = multivariate_normal.pdf(
                data,
                mean = self.mu[k],
                cov = self.cov[k]
            )
        # ravel(): 将多维数组转化为一维数组， np.diag(): 将一维矩阵转化为对角矩阵
        self.posteriori = np.dot(np.diag(self.priori.ravel()), self.posteriori)
        self.posteriori /= np.sum(self.posteriori, axis=0)
```

二、M-step：更新均值、协方差、先验概率

1.先求出Nk

$$
N_k = \sum_{n=1}^{N} \gamma(z_{nk})
$$

get effective count 获得Nk的值

effecitve_count = np.sum(self.posteriori, axis=1)

2.更新GMM中的参数

$$
\begin{aligned}
\boldsymbol{\mu}_k^{\text{new}} &= \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \mathbf{x}_n \\
\boldsymbol{\Sigma}_k^{\text{new}} &= \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \left(\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}}\right) \left(\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}}\right)^{\mathrm{T}} \\
\pi_k^{\text{new}} &= \frac{N_k}{N}
\end{aligned}
$$

M-step: 更新GMM的参数

```python
# M-step: 更新GMM的参数
self.mu = np.asarray([np.dot(self.posteriori[k], data) / effecitve_count[k] for k in range(self.K)])
self.cov = np.asarray([np.dot((data - self.mu[k]).T, np.dot(np.diag(self.posteriori[k].ravel()), data - self.mu[k]))
self.priori = (effecitve_count / N).reshape((self.K, 1))
```