一. 模型

$$
\begin{aligned}
P(x|\theta) &= \sum_{i=1}^{K} p(x, z=k|\theta) \\
&= \sum_{k=1}^{K} p(x, |z=k, \theta) p(z=k|\theta)
\end{aligned}
$$

其中

$$
\begin{aligned}
p(z|\theta) &\overset{\Delta}{=} Cat(\pi) \\
p(x|z=k, \theta) &\overset{\Delta}{=} N(x|\mu_k, \Sigma_k)
\end{aligned}
$$

二. 参数估计

2.1 负对数似然

$$
\begin{aligned}
NLL(\theta) &= -\sum_{i=1}^{N} \log[\sum_{k=1}^{K} p(x|z=k, \theta) p(z=k|\theta)] \\
&= -\sum_{i=1}^{N} \log[\sum_{k=1}^{K} \pi_k N(\mu_k, \Sigma_k)]
\end{aligned}
$$

$$
\frac{\partial NLL(\theta)}{\partial \pi_k} = \sum_{i=1}^{N} \frac{N(\mu_k, \Sigma_k)}{\sum_{k=1}^{K} \pi_k N(\mu_k, \Sigma_k)}
$$

Intractable?

2.2 EM

从Variational Bayesian推出Exact EM:

VB中，目标是求后验分布 $p(h, \theta|x)$

ELBO (Evidence Lower BOuned)

$$
\begin{aligned}
\mathcal{L}(D, q, \theta) &= \log p(D) - KL(q(h, \theta)|p(h, \theta|D)) \\
&= E_q[p(D, h, \theta)] + H(q)
\end{aligned}
$$

使用Mean field, 并且限制 θ 的概率密度集中在一点。

$$
q(h, \theta) = \delta_{\theta=\theta^*}(\theta) \prod_i q_{h_i}(h_i)
$$

做Coordinate Descent

E Step:

$$
\begin{aligned}
\log q_{h_i}(h_i) &= E_{\delta(\theta)}[\log p(x_i, h_i, \theta)] + const \\
&= \log p(x_i, h_i, \theta^t) + const \\
&= \log p(h_i|x_i, \theta^t) + const
\end{aligned}
$$

M Step:

$$
\begin{aligned}
\theta^{t+1} &= \arg\max_{\theta} E_{q_h(h)}[\sum_{i=1}^{N} \log p(x_i, h_i, \theta)] \\
&= \arg\max_{\theta} E_{q_h(h)}[\sum_{i=1}^{N} \log p(x_i, h_i|\theta)] \\
&= \arg\max_{\theta} \sum_{i=1}^{N} E_{q_{h_i}(h_i)} \log p(x_i, h_i|\theta)
\end{aligned}
$$

从Variational Inference推出Exact EM

假设 θ 已知, 目标：求 $p(h|D;\theta)$

$$
\begin{aligned}
\mathcal{L}(q, \mathcal{D}; \theta) &= \log p(D;\theta) - KL(q(h)|p(h|D;\theta)) \\
&= \log p(D;\theta) - E_q[q(h)] + \sum_h q(h) \log p(h|D;\theta) \\
&= \sum_h q(h)[\log p(h|D;\theta) + \log p(D;\theta)] + H(q) \\
&= E_q[p(h, D; \theta)] + H(q)
\end{aligned}
$$

第t步， $\theta = \theta^t$

E Step:

$$
q(h) = p(h|D;\theta^t)
$$

M Step:

$$
\theta^{t+1} = \arg\max E_q[p(h, D; \theta)]
$$

证明每次迭代后， $p(D;\theta)$ 是上升的

$$
\begin{aligned}
p(D;\theta^{t+1}) &\geq \mathcal{L}(p(h|D;\theta^t), D, \theta^{t+1}) \\
&\geq \mathcal{L}(p(h|D;\theta^t), D, \theta^t) \\
&= p(D;\theta^t)
\end{aligned}
$$

2.3 EM for GMM

E Step

$$
\begin{aligned}
Q(\theta,\theta^t) &= \sum_{i=1}^{N} E_{z_i\sim p(z_i|x_i,\theta^t)}[\log p(x_i,z_i|\theta)] \\
&= \sum_{i=1}^{N}\sum_{k=1}^{K} p(z_i=k|x_i,\theta^t)\log p(x_i,z_i=k|\theta) \\
&= \sum_{i=1}^{N}\sum_{k=1}^{K} p(z_i=k|x_i,\theta^t)[\log p(x_i|z_i=k,\theta)+\log p(z_i=k|\theta)] \\
&= \sum_{i=1}^{N}\sum_{k=1}^{K} r_{ik}\log \pi_k + \sum_{i=1}^{N}\sum_{k=1}^{K} r_{ik}\log p(x_i|z_i=k,\theta)
\end{aligned}
$$

其中

$$
\begin{aligned}
r_{ik} &= \log p(z_i=k|x_i,\theta^t) = \log p(x_i,z_i=k|\theta^t) - \log p(x_i) \\
\pi_k &= p(z_i=k|\theta)
\end{aligned}
$$

对GMM,

$$
\begin{aligned}
\log p(x_i|z_i=k,\theta) &= \log[\frac{1}{2\pi^{D/2}\det(\Sigma_k)^{1/2}}\exp[-\frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k)]] \\
&= -\frac{1}{2}[D\log 2\pi + \log\det(\Sigma_k) + (x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k)]
\end{aligned}
$$

Note: 计算 $p(x_i|z_i=k,\theta)$ 时，需要处理Covariance矩阵的逆，下面想办法避免。

对 Σ 做Cholesky分解：

$$
\Sigma = LL^T
$$

则，精度矩阵

$$
\Lambda = \Sigma^{-1} = (LL^T)^{-1} = (L^{-T})(L^{-T})^T
$$

所以，精度矩阵可Cholesky分解为 AA^T , 其中 $A=L^{-T}$

计算A:

$$
\begin{aligned}
A &= L^{-T} \\
&= sovle(L, I_{identity}).T
\end{aligned}
$$

计算 $\log\det(\Sigma)$

$$
\begin{aligned}
\log\det(\Sigma) &= -\log\det(\Lambda) \\
&= -2\log det(A)
\end{aligned}
$$

计算 $(x-\mu)^T\Lambda(x-\mu)$

$$
(x-\mu)^T\Lambda(x-\mu) = [A^T(x-u)]^T[A^T(x-\mu)]
$$

矩阵化：（下面的 x_i 是向量）

$$\begin{bmatrix} (A^T x_1)^T \\ (A^T x_2)^T \\ \vdots \\ (A^T x_N)^T \end{bmatrix} = \begin{bmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_N^T \end{bmatrix} A = XA$$

M Step

求 π

Lagrangian:

$$L(\pi_1, ..., \pi_K, \lambda) = \sum_{i=1}^{N} \sum_{k=1}^{K} r_{ik} \log \pi_k - \lambda(\sum_{k=1}^{K} \pi_k - 1)$$

$$\pi_k = \frac{1}{N} \sum_{i=1}^{N} r_{ik}$$

求 μ

$$\begin{aligned} L(\mu_K, \Sigma_k) &= -\frac{1}{2} \sum_{i=1}^{N} r_{ik} [\log\det(\Sigma_k) + (x_i - \mu_k)^T \Sigma_k^{-1} (x_i - \mu_k)] \\ \nabla_{\mu_k} L &= \frac{1}{2} \sum_{i=1}^{N} r_{ik} (\Sigma^{-1} + \Sigma^{-T})(x_i - \mu_k) \\ &= \sum_{i=1}^{N} r_{ik} (\Sigma^{-1})(x_i - \mu_k) = 0 \end{aligned}$$

所以

$$\begin{aligned} \mu_k &= \frac{\sum_{i=1}^{N} r_{ik} x_i}{N_k} \\ \text{self.means\_} &= \begin{bmatrix} \mu_1^T \\ \mu_2^T \\ \vdots \\ \mu_K^T \end{bmatrix} = \begin{bmatrix} \sum_i r_{i1} x_1^T \\ \sum_i r_{i2} x_2^T \\ \vdots \\ \sum_i r_{iK} x_N^T \end{bmatrix} \\ &= \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1N} \\ r_{21} & r_{22} & \cdots & r_{2N} \\ \vdots & \vdots & \cdots & \vdots \\ r_{K1} & r_{K2} & \cdots & r_{KN} \end{bmatrix} \begin{bmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_N^T \end{bmatrix} \\ &= R^T X \end{aligned}$$

上面只是分子部分，处分母最后可利用numpy的broadcast

求 Σ

$$
\begin{aligned}
L(\mu_K, \Sigma_k) &= -\frac{1}{2}\sum_{i=1}^{N} r_{ik}[\log\det(\Sigma_k) + (x_i - \mu_k)^T \Sigma_k^{-1}(x_i - \mu_k)] \\
&= -\frac{1}{2}\sum_{i=1}^{N} r_{ik}[-\log\det(\Lambda_k) + (x_i - \mu_k)^T \Lambda_k (x_i - \mu_k)]
\end{aligned}
$$

注意

$$
\nabla_{\Lambda_k} \log\det(\Lambda_k) = \Lambda_k^{-T}
$$

所以

$$
\nabla_{\Lambda_k} L = -\frac{1}{2}\sum_{i=1}^{N} r_{ik}[-\Lambda_k^{-T} + (x_i - \mu_k)(x_i - \mu_k)^T] = 0
$$

则

$$
\begin{aligned}
\sum_{i=1}^{N} r_{ik}(x_i - \mu_k)(x_i - \mu_k)^T &= N_k \Lambda_k^{-T} \\
\Lambda_k^{-T} &= \frac{\sum_{i=1}^{N} r_{ik}(x_i - \mu_k)(x_i - \mu_k)^T}{N_k}
\end{aligned}
$$

注意协方差矩阵和精度矩阵都是对称的

$$
\Lambda_k^{-T} = \Lambda_k^{-1} = \Sigma_k
$$

implementation

$$
\begin{aligned}
&\sum_{i=1}^{N} r_{ik}(x_i - \mu_k)(x_i - \mu_k)^T \\
&= \begin{bmatrix} r_{1k}(x_1 - \mu_k) & r_{2k}(x_2 - \mu_k) & \cdots & r_{Nk}(x_N - \mu_K) \end{bmatrix} \begin{bmatrix} (x_1 - \mu_k)^T \\ (x_2 - \mu_k)^T \\ \vdots \\ (x_N - \mu_K)^T \end{bmatrix} \\
&= (R[:, k] * X^T) X
\end{aligned}
$$

三.一些函数

$\log p(z_i)$

```
@abstractmethod
def _log_z_prob(self):
    pass
```

返回的arr的shape应为 `(n_samples, n_components)`，`arr[i][k]` 表示 $\log p(z_i = k)$，下面类似。

$\log p(x_i | z_i, \theta)$

```
@abstractmethod
def _log_x_cond_z_prob(self, X):
    pass
```

$\log p(x_i, z_i | \theta)$

```
def _log_x_and_z_prob(self, X):
    return self._log_x_cond_z_prob(X) + self._log_z_prob()
```

$\log p(z_i | x_i, \theta)$

```
def _log_z_cond_x_prob(self, X):
    log_x_and_z = self._log_x_and_z_prob(X)
    log_x = logsumexp(log_x_and_z, axis=1)
    log_z_cond_x = log_x_and_z - log_x[:,np.newaxis]
    return log_x, log_z_cond_x
```

predict

$$\begin{aligned}
\text{y_pred}_i &= \arg\max_k p(z_i = k | x_i, \theta) \\
&= \arg\max_k p(z_i = k, x_i, \theta)
\end{aligned}$$

```
def predict(self, X):
    return self._log_x_and_z_prob(X).argmax(axis=1)
```

predict_prob

$$p(z_i | x_i, \theta) = \exp \log p(z_i | x_i, \theta)$$

```
def predict_proba(self, X):
    _, log_z_cond_x = self._log_z_cond_x_prob(X)
    return np.exp(log_z_cond_x)
```