特征维数问题：

* 分类错误率：

考虑二分类高斯分布：$p(x|w_i)\sim N(\mu_i,\Sigma)$，$p(w_1)=p(w_2)=\frac{1}{2}$，则

此时判别准则为：若 $|x-\mu_1|>|x-\mu_2|$，则判为 w_2，否则为 w_1

因此，Bayesian error：$p(error)=p(x\in R_1,w_2)+p(x\in R_2,w_1)$

$$=p(x\in R_1|w_2)p(w_2)+p(x\in R_2|w_1)p(w_1)$$

$$=p(|x-\mu_1|<|x-\mu_2|\,|w_2)\times\frac{1}{2}+p(|x-\mu_2|<|x-\mu_1|\,|w_1)\times\frac{1}{2}$$

$$=\frac{1}{2}\int_{\frac{r}{2}}^{\infty}e^{-\frac{u^2}{2}}du. \qquad r^2=(\mu_1-\mu_2)^t\Sigma^{-1}(\mu_1-\mu_2)$$

若 $\Sigma=diag\{\sigma_1^2\cdots,\sigma_d^2\}$.

则 $r^2=\sum_{i=1}^{d}(\frac{\mu_{i1}-\mu_{i2}}{\sigma_i})^2$，则显然，增加特征可以增大 r.

进一步降低 $p(error)$.

* 计算复杂度：对于 n 个 d 维特征的样本，假设类条件概率为高斯分布，

模型参数估计采用 MLE，则判别函数为：

$$g(x)=-\frac{1}{2}(x-\hat{\mu})^t\hat{\Sigma}^{-1}(x-\hat{\mu})-\frac{d}{2}\ln 2\pi-\frac{1}{2}\ln|\hat{\Sigma}|+\ln P(w)$$

参数估计：$\hat{\mu}=\frac{1}{n}\sum_{k=1}^{n}x_k$ $O(nd)$. $\hat{\Sigma}=\frac{1}{n}\sum_{k=1}^{n}(x_k-\hat{\mu})(x_k-\hat{\mu})^t$ $-O(nd^2)$

参数存储：$c(d+\frac{d(d+1)}{2})$

增加特征会增加计算复杂度

* 过拟合：特征维数高，训练样本少 ⇒ 过拟合

* 特征提取 / 特征变换

* 参数共享：共享协方差阵.

* 平滑：$\Sigma_i(\alpha)=\frac{(1-\alpha)n_i\Sigma_i+\alpha n\Sigma}{(1-\alpha)n_i+\alpha n}$.

EM算法：

使用场景：数据存在缺失下进行参数估计

$x_k = \{x_{kg}, x_{kb}\}$. $\quad D = \{x_1, \cdots, x_n\} = D_g \cup D_b$.

EM算法：在知道 θ^i 情况下重新估计 θ. 要求：

$$\max Q(\theta; \theta^i) = \mathbb{E}_{D_b}\{\ln p(D_g, D_b; \theta) \mid D_g; \theta^i\}.$$

即 Step 1. 初始化待估参数 θ^0

Step 2. 在知道 θ^0 的情况下计算

$$Q(\theta, \theta^i) = \mathbb{E}_{D_b}\{\ln p(D_g, D_b; \theta) \mid D_g; \theta^i\}.$$

Step 3. $\theta^{i+1} = \arg\max_\theta Q(\theta, \theta^i)$

Step 4. 不断迭代，直到 $Q(\theta^{i+1}, \theta^i) - Q(\theta^i, \theta^{i-1}) \leq T$.

EM算法可以保证收敛性.

例： $D = \{x_1, x_2, x_3, x_4\}. = \left\{\begin{pmatrix}0\\2\end{pmatrix}, \begin{pmatrix}1\\0\end{pmatrix}, \begin{pmatrix}2\\2\end{pmatrix}, \begin{pmatrix}*\\4\end{pmatrix}\right\}$.

2D Gaussian 有参数 $\theta = \begin{pmatrix}\mu_1\\\mu_2\\\sigma_1^2\\\sigma_2^2\end{pmatrix}$,

初始化参数 $\theta^0 = \begin{pmatrix}0\\0\\1\\1\end{pmatrix}$

则 θ^0 下有 $\quad p(x) = p(x_1|\theta) \cdot p(x_2|\theta) p(x_3|\theta) \cdot p(x_4|\theta)$

$$Q(\theta, \theta_0) = \int_{-\infty}^{+\infty} \left(\sum_{k=1}^{3} \ln p(x_k|\theta) + \ln p(x_4|\theta)\right) p(x_{41} \mid \theta^0; x_{42}=4)\, dx_{41}$$

$$= \sum_{k=1}^{3} \ln p(x_k|\theta) + \int_{-\infty}^{+\infty} \ln p(x_4|\theta) \cdot \frac{p(x_4|\theta^0)}{\int_{-\infty}^{+\infty} p(x_4|\theta^0)\, dx_{41}}\, dx_{41}$$

$$= \sum_{k=1}^{3} [\ln p(x_k|\theta)] + \frac{1}{K} \int_{-\infty}^{+\infty} \ln p(x_4|\theta) \frac{1}{2\pi} \exp\{x_{41}^2 + 4^2\}\, dx_{41}$$

$$= \sum_{k=1}^{3} \ln p(x_k|\theta) - \frac{1+\mu_1^2}{2\sigma_1^2} - \frac{(4-\mu_2)^2}{2\sigma_2^2} - \ln(2\pi\sigma_1\sigma_2)$$

从而可以求得： $\theta^1 = \arg\max_\theta Q(\theta, \theta^0)$.

EM算法求得 GM

GM模型： $p(x) = \sum_{k=1}^{K} \pi_k p(x|\theta_k)$, $\sum_{k=1}^{K} \pi_k = 1$.

即 K 个 Gauss 分布的凸组合.

使用MLE，有： $\prod_{n=1}^{N}\left[\sum_{k=1}^{K}\pi_k\, p(x_n|\theta_k)\right]$

$$\Rightarrow LL=\sum_{n=1}^{N}\log\sum_{k=1}^{K}\pi_k\, p(x_n|\theta_k)$$

拆分不开，求导无法得解析解。

使用EM算法：对给定的 μ_k、Σ_k、π_k，有

$$LL(x|\pi_k,\mu_k,\Sigma_k)=\sum_{n=1}^{N}\log\left(\sum_{k=1}^{K}\pi_k\,\mathcal{N}(x_n|\mu_k,\Sigma_k)\right)$$

对 μ_k 求导：

$$\nabla_{\mu_k}=-\sum_{n=1}^{N}\frac{\pi_k\,\mathcal{N}(x_n|\mu_k,\Sigma_k)}{\sum_{k=1}^{K}\pi_k\,\mathcal{N}(x_n|\mu_k,\Sigma_k)}\cdot\Sigma_k^{-1}(x_n-\mu_k)=0$$

记 $\gamma_{nk}=\pi_k\,\mathcal{N}(x_n|\mu_k,\Sigma_k)$，即 x_n 是从第k个Gauss采样的概率。$N_k=\sum_{n=1}^{N}\gamma_{nk}$，表示所有样本中属于第k个Gauss的平均数量。

$$\sum_{n=1}^{N}\gamma_{nk}x_n=\sum_{n=1}^{N}\gamma_{nk}\mu_k \Rightarrow \mu_k=\frac{\sum_{n=1}^{N}\gamma_{nk}x_n}{N_k}$$

同样，可求得 $\Sigma_k=\frac{1}{N_k}\sum_{n=1}^{N}\gamma_{nk}(x_n-\mu_k)(x_n-\mu_k)^T$

对于 π_k，要加入 Lagrange 乘子：

$$LL+\lambda\left(\sum_{k=1}^{K}\pi_k-1\right)$$

对 π_k 求梯度有： $\sum_{n=1}^{N}\frac{\mathcal{N}(x_n|\mu_k,\Sigma_k)}{\sum_{k=1}^{K}\pi_k\mathcal{N}(x_n|\mu_k,\Sigma_k)}+\lambda=0$

$$\Rightarrow \sum_{n=1}^{N}\frac{\mathcal{N}(x_n|\mu_k,\Sigma_k)\pi_k}{\sum_{k=1}^{K}\pi_k\mathcal{N}(x_n|\mu_k,\Sigma_k)}+\lambda\pi_k=0$$

$$\Rightarrow 0=N_k+\lambda\pi_k$$

$$\Rightarrow 0=\sum_{k=1}^{K}N_k+\lambda\sum_{k=1}^{K}\pi_k$$

$$\Rightarrow \lambda=-N \quad\Rightarrow \pi_k=\frac{N_k}{N}$$

隐马尔可夫模型（HMM）

* Markov Chain：

一阶马尔可夫链． $P(q_t=S_j|q_{t-1}=S_i,q_{t-2}=S_k,\cdots)=P(q_t=S_j|q_{t-1}=S_i)$

则 $p(q_1,\cdots,q_T)=p(q_1)p(q_2|q_1)\cdots p(q_T|q_{T-1})$．

转移概率． $a_{ij}\triangleq P(q_t=S_j|q_{t-1}=S_i)$

State duration: $O = \{\underset{1}{S_i}, \cdots, \underset{d}{S_i}, \underset{d+1}{S_j} \neq S_i\}$

$P(O \mid Model, q_1 = S_i) = (a_{ii})^{d-1}(1-a_{ii}) = p_i(d)$

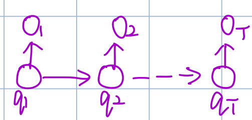
从 i d 步转出的概率.

Expected duration:

$$\bar{d}_i = \sum_{d=1}^{\infty} d\, p_i(d) = \sum_{d=1}^{\infty} (a_{ii})^{d-1}(1-a_{ii})\, d = \frac{1}{1-a_{ii}}$$

从i转出平均所需转移次数.

* Hidden Markov chain: (双嵌入 MC). $\lambda(A, B, \pi)$



N: 模型的状态空间状态数: $\{S_1, \cdots, S_N\}$.

M: 可以被观测到的信号类别数: $\{v_1, \cdots, v_M\}$.

$A = \{a_{ij}\}$: 状态转移概率: $a_{ij} = P(q_{t+1} = S_j \mid q_t = S_i)$

$B = \{b_j(k)\}$: 信号发射概率, $b_j(k) = P(v_k \text{ at } t \mid q_t = S_j)$

π_i: 初始状态. $\pi_i = P(q_1 = S_i)$

*Evaluation (如何快速计算观测概率 $p(O|\lambda)$)

$\lambda = (A, B, \pi)$, $O = O_1 O_2 \cdots O_T$

$$p(O|\lambda) = \sum_{q_1 \cdots q_T} \pi_{q_1} b_{q_1}(O_1) a_{q_1 q_2} b_{q_2}(O_2) \cdots a_{q_{T-1} q_T} b_{q_T}(O_T)$$

$$= \sum_{Q} p(O|Q,\lambda)\, p(Q|\lambda).$$

其中. $p(O|Q,\lambda) = b_{q_1}(O_1) b_{q_2}(O_2) \cdots b_{q_T}(O_T)$

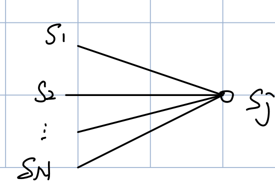
$p(Q|\lambda) = \pi_{q_1} a_{q_1 q_2} \cdots a_{q_{T-1}} a_{q_T}$

计算复杂度 $O(2TN^T)$

前向算法:

前向变量: $\alpha_t(i) = P(O_1 \cdots O_t, q_t = S_i \mid \lambda)$.

则 $\alpha_1(i) = \pi_i b_i(O_1)$

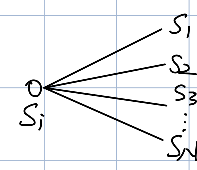


$$\alpha_{t+1}(j)=\left[\sum_{i=1}^{N}\alpha_t(i)a_{ij}\right]\cdot b_j(O_{t+1}) \qquad P(O|\lambda)=\sum_{i=1}^{N}\alpha_T(i)$$

计算复杂度： $O(TN^2)$

后向算法：

后向变量. $\beta_t(i)=P(O_{t+1}\cdots O_T \mid q_t=S_i,\lambda)$.



则 $\beta_T(i)=1$.

$$\beta_t(i)=\sum_{j=1}^{N}a_{ij}\,b_j(O_{t+1})\cdot\beta_{t+1}(j)$$

$$P(O|\lambda)=\sum_{i=1}^{N}\pi_i b_i(O_1)\beta_1(i)$$

计算复杂度： $O(TN^2)$.

* Decoding（如何根据观测序列选择概率最大的状态序列）

注意到： $\max_{q_1q_2\cdots q_T} P(q_1q_2\cdots q_T|O,\lambda)=\max_{q_1q_2\cdots q_T}P(q_1q_2\cdots,q_T,O|\lambda)$.

Viterbi 算法：

$$\delta_t(i)=\max_{q_1\cdots q_{t-1}}P(q_1q_2\cdots q_t=i,O_1O_2\cdots O_t|\lambda).$$

根据 Bellman 最优性条件：

$$\delta_{t+1}(j)=\left[\max_i \delta_t(i)a_{ij}\right]b_j(O_{t+1})$$

计算过程.

— 递归： $\delta_t(j)=\left[\max_i \delta_{t-1}(i)a_{ij}\right]b_j(O_t)$

$$\psi_t(j)=\arg\max_i \delta_{t-1}(i)a_{ij}$$

— 终止条件： $p^*=\max_i \delta_T(i) \qquad q_T^*=\arg\max_i \delta_T(i)$

— 回溯： $q_t^*=\psi_{t+1}(q_{t+1}^*)$

计算复杂度： $O(TN^2)$

* Training 目标： $\max_{A,B,\pi} P(O|\lambda)$.

* 监督学习下： $\hat{a}_{ij}=\dfrac{A_{ij}}{\sum_{j=1}A_{ij}}$（统计样本频率）.

$$\hat{b}_j(k) = \frac{B_{jk}}{\sum_{k=1}^{M} B_{jk}}.$$

* 无监督学习下：

Baum-Welch Algorithm（EM算法）.

假设隐数据 $I = (i_1, \cdots, i_T)$，则.

$$P(O|\lambda) = \sum_I P(O|I,\lambda) \times P(I|\lambda).$$ 使用 EM 算法.

E步： $Q(\lambda, \hat{\lambda}) = \sum_I P(O, I|\hat{\lambda}) \log P(O, I|\lambda).$（交叉熵）

$$P(O, I|\lambda) = \pi_{i_1} b_{i_1}(O_1) a_{i_1 i_2} b_{i_2}(O_2) \cdots a_{i_{T-1} i_T} b_{i_T}(O_T)$$

$$\Rightarrow Q(\lambda, \hat{\lambda}) = \sum_I \log \pi_{i_1} P(O, I|\hat{\lambda}) + \sum_I \left(\sum_{t=1}^{T-1} \log a_{i_t i_{t+1}}\right) P(O, I|\hat{\lambda}) + \sum_I \left(\sum_{t=1}^{T} b_{i_t}(O_t)\right) P(O, I|\hat{\lambda}).$$

M步： $\sum_I \log \pi_{i_1} P(O, I|\hat{\lambda}) = \sum_{i=1}^{N} \log \pi_i P(O, i_1 = i|\hat{\lambda}).$

而 $\sum_{i=1}^{N} \pi_i = 1$. 由 Lagrange 乘子法：

$L(\pi_i, \gamma) = \sum_{i=1}^{N} \log \pi_i \, P(O, i_1 = i|\hat{\lambda}) + \gamma\left(\sum_{i=1}^{N} \pi_i - 1\right)$ 求偏导得：

$\frac{\partial L(\pi_i, \gamma)}{\partial \pi_i} = 0 \iff P(O, i_1 = i|\hat{\lambda}) + \gamma \pi_i = 0$，由 $\sum \pi_i = 1$

$$\Rightarrow \gamma = -P(O|\hat{\lambda}). \quad \Rightarrow \pi_i = \frac{-P(O, i_1 = i|\hat{\lambda})}{P(O|\hat{\lambda})}.$$

可理可得： $a_{ij} = \frac{\sum_{t=1}^{T-1} P(O, i_t = i, i_{t+1} = j|\hat{\lambda})}{\sum_{t=1}^{T-1} P(O, i_t = i|\hat{\lambda})}. = \frac{\sum_{t=1}^{T-1} \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{t=1}^{T-1} \alpha_t(i) \beta_t(j)}$

$$b_j(k) = \frac{\sum_{t=1}^{T} P(O, i_t = j|\hat{\lambda}) I(O_t = v_k)}{\sum_{t=1}^{T} P(O, i_t = j|\hat{\lambda})} = \frac{\sum_{t=1, O_t = v_k}^{T} \alpha_t(i) \beta_t(j)}{\sum_{t=1}^{T} \alpha_t(i) \beta_t(j)}$$