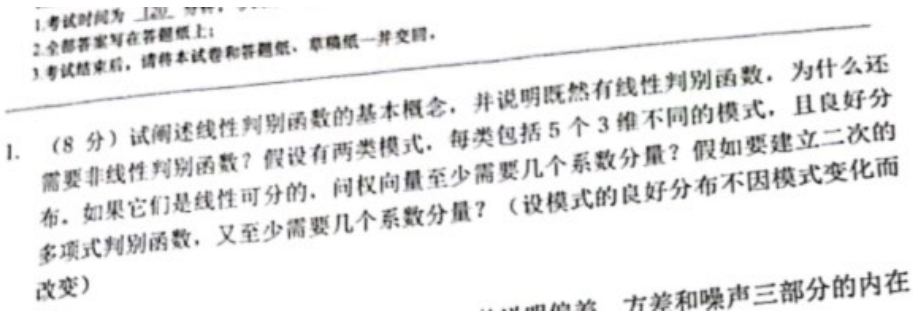

1.考试时间为 [illegible]
2.全部答案写在答题纸上；
3.考试结束后，请将本试卷和答题纸、草稿纸一并交回。

1. （8 分）试阐述线性判别函数的基本概念，并说明既然有线性判别函数，为什么还需要非线性判别函数？假设有两类模式，每类包括 5 个 3 维不同的模式，且良好分布，如果它们是线性可分的，问权向量至少需要几个系数分量？假如要建立二次的多项式判别函数，又至少需要几个系数分量？（设模式的良好分布不因模式变化而改变）

[illegible]，方差和噪声三部分的内在

i. 线性判别函数一般是$y = wx$　其中x是特征向量的增广形式，w 是权重系数。根据y的取值进行分类，这个函数在几何上一般表现为直线（高维空间的超平面），所以称之为线性判别函数。如果x是低纬向高维投影后的特征向量，那么就是广义线性判别，理论上广义线性判别可以模拟任意复杂的函数。

ii. 虽然广义线性判别可以拟合非线性，但是会面临参数爆炸问题，假如采用核技巧，虽然参数不会爆炸，但是面临 kernel 函数形式有限和无法度量哪个 kernel 函数更有效的问题，因此，假如经过先验知识，可以设计更合适的非线性的模型，是必要的。

iii. 参数数目

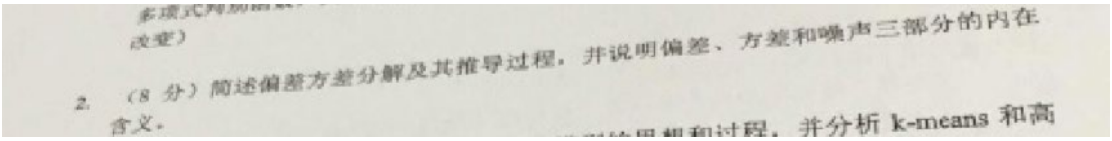
a) 线性需要（3+1）= 4

b) 二次需要（3(一次) + 3（二次） + 3(混合) + 1 （偏移） ）= 10

c) 或者直接用：公式$\frac{(n+r)!}{n!r!}$

i. 线性 $n = 3, r = 1$；$\frac{(n+r)!}{n!r!} = 4$

ii. 二次$n = 3, r = 2$；$\frac{(n+r)!}{n!r!} = 10$


改变）

2. （8 分）简述偏差方差分解及其推导过程，并说明偏差、方差和噪声三部分的内在含义。

[illegible]和过程，并分析 k-means 和高

i. 偏差-方差分解的推导过程

$$E(f_D, y_D) = E((f_D - y_D)^2) = E((f_D - f + f - y_D)^2)$$
$$= E((f_D - f)^2) + E((f - y_D)^2) + E(2(f_D - f)(f - y_D))$$

由于$E(f_D - f) = 0$,因此第三项为 0

$$= E((f_D - f)^2) + E((f - y + f - y_D)^2)$$
$$= E((f_D - f)^2) + E((f - y)^2) + E((y - y_D)^2) + E(2(f - y)(y - y_D))$$

由于$E(y - y_D) = 0$,因此第四项为 0

$$= E((f_D - f)^2) + E((f - y)^2) + E((y - y_D)^2)$$

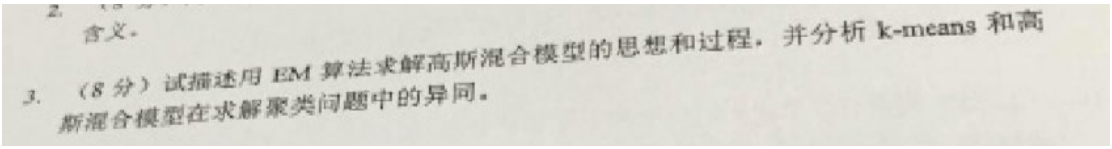
$E((f_D - f)^2)$ 是方差

$E((f - y)^2)$ 是偏差

$E((y - y_D)^2)$ 是误差

ii. 内在含义

a) 偏差：偏差是训练所使用的模型和模式之间的差异导致的错误
b) 方差：方差是相同模型在不同采样数据下训练带来的误差。
c) 噪音：是采样过程中采样误差（噪声）导致的训练结果的错误。


含义。

3. （8 分）试描述用 EM 算法求解高斯混合模型的思想和过程，并分析 k-means 和高斯混合模型在求解聚类问题中的异同。

i. EM 算法求解高斯混合模型
   a) 思想： 假定存在 M 个独立的高斯分布，数据x_i按照π_i 的概率从第i个高斯进行采样获取的。
   b) 过程
      i. 首先初始化混合高斯的参数$\theta = \{\pi, \delta, \mu\}$
      ii. 迭代直到收敛
         1. E 步骤，计算$\gamma(z_i^j) = p(y_i|x_i, \theta) = \frac{p(y_i|x_i,\theta)}{\sum_{i=1}^{M} p(y_i|x_i,\theta)}$
         2. M 步骤，更新

$$\theta = maxarg_\theta \log(p(x, y|\theta))$$
$$= maxarg_\theta \sum_{i=1}^{N} \log\left(\sum_{j=1}^{M} p(x_i, y_i^j | \theta)\right)$$

参数迭代公式（背过吧~）

$$\pi_k = \frac{\sum_i \gamma(z_i^k)}{N}$$
$$\mu_k = \frac{\sum_i \gamma(z_i^k) x_i}{\sum_i \gamma(z_i^k)}$$
$$\Sigma_k = \frac{\sum_i \gamma(z_i^k)(x_i - \mu_k)(x_i - \mu_k)^T}{\sum_i \gamma(z_i^k)}$$

ii. k-means 和高斯混合的异同
   a) 不同：
      i. k-means 的损失函数是最小平方距离， 混合高斯是负对数似然函数
      ii. k-means 是硬划分，混合高斯是软化分
      iii. k-maans 假设类别是概率相同且是球簇，混合高斯可以处理非球形，类别概率不同
   b) 相同：
      i. 混合高斯的 E 步骤其实就是软化分的 k-means
      ii. 当类别概率相同，且$\Sigma = \sigma I, \sigma \to 0, r_{i,j} \to \{0,1\}$ 的时候，混合高斯退化为 k-means。

---

4. （10分）用下列势函数

$$K(x,x_k)=e^{-\|x-x_k\|^2}$$

求解以下模式的分类问题

ω_1: $\{(0\ 1)^T,\ (0\ -1)^T\}$

ω_2: $\{(1\ 0)^T,\ (-1\ 0)^T\}$

迭代直到全部可以分类:

$$K_0(\boldsymbol{x})=0\ ;$$
$$K_{i+1}(\boldsymbol{x})=\ K(\boldsymbol{x})+K(\boldsymbol{x},\boldsymbol{x_i})\ \ \boldsymbol{if}\ K(\boldsymbol{x_i},\boldsymbol{x})\le\boldsymbol{0}\ \&\&\ \boldsymbol{w_i}=\boldsymbol{1}$$
$$K_{i+1}(\boldsymbol{x})=\ K(\boldsymbol{x})-K(\boldsymbol{x},\boldsymbol{x_i})\ \ \boldsymbol{if}\ K(\boldsymbol{x_i},\boldsymbol{x})\ge\boldsymbol{0}\ \&\&\ \boldsymbol{w_i}=-\boldsymbol{1}$$
$$K(x)=\exp\left(-(\boldsymbol{x}-(0,1)^T)^2\right)+\ \exp\left(-(\boldsymbol{x}-(0,-1)^T)^2\right)$$
$$-\exp\left(-(\boldsymbol{x}-(1,0)^T)^2\right)-\exp\left(-(\boldsymbol{x}-(-1,0)^T)^2\right)$$

刚好四个点都需要添加进去。

---

5. （10分）试述K-L变换的基本原理，并将如下两类样本集的特征维数降到一维，同时画出样本在该空间中的位置。

ω_1: $\{(-5\ -5)^T,\ (-5\ -4)^T,\ (-4\ -5)^T,\ (-5\ -6)^T,\ (-6\ -5)^T\}$

ω_2: $\{(5\ 5)^T,\ (5\ 6)^T,\ (6\ 5)^T,\ (5\ 4)^T,\ (4\ 5)^T\}$,

其中假设其先验概率相等，即 $P(\omega_1)=P(\omega_2)=0.5$。

i. K-L 变换的基本原理:
   a) K-L 的关注问题是在均方误差最小的条件下获得最佳降维变换。
   b) 算法步骤是:
      i. 将特征减去均值$X=X-E[X]$
      ii. 计算协方差矩阵$C=XX^T$
      iii. C进行特征值分解，获得的特征向量按照特征值大小排序，取其前k个作为转移矩阵W
      iv. W^TX 就是降维后的特征

ii. 对样本进行降维
   a) $E(X)=(0,0)^T$ 符合最佳 K-L 变换需求
   b) $C=X^TX=\begin{Bmatrix}254 & 250\\ 250 & 254\end{Bmatrix}$
   c) $(C-\lambda I)x=0\ \rightarrow\begin{vmatrix}254-\lambda & 250\\ 250 & 254-\lambda\end{vmatrix}=0$

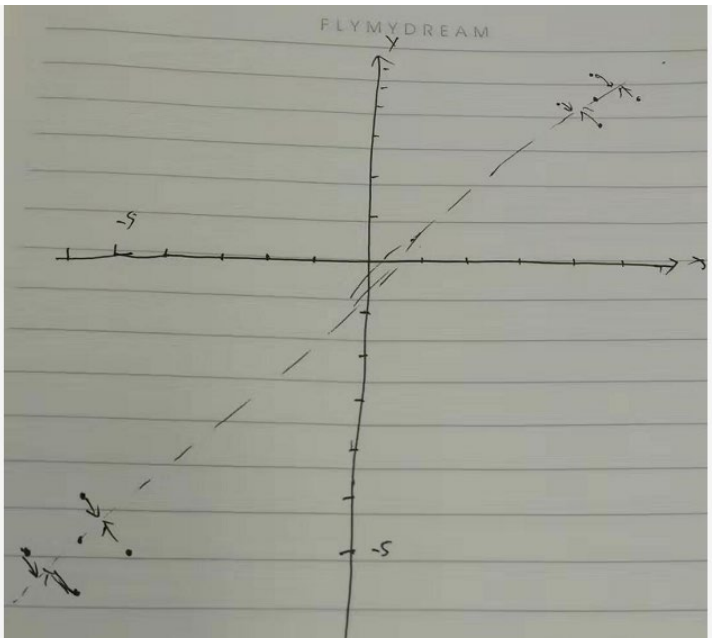
$$\rightarrow\begin{vmatrix}4-\lambda & -4+\lambda\\ 250 & 254-\lambda\end{vmatrix}=0$$
$$\rightarrow(4-\lambda)(504-\lambda)=0$$

      i. $\rightarrow\lambda_1=504\rightarrow$ 特征向量$w=(\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}})^T$

      降维: $w=(\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}})^T$

ii. $W^T X = \{-\frac{10}{\sqrt{2}}, -\frac{9}{\sqrt{2}}, -\frac{9}{\sqrt{2}}, -\frac{11}{\sqrt{2}}, -\frac{11}{\sqrt{2}}, \frac{10}{\sqrt{2}}, \frac{11}{\sqrt{2}}, \frac{11}{\sqrt{2}}, \frac{9}{\sqrt{2}}, \frac{9}{\sqrt{2}}\}$

d)



---

6. （10 分）详细描述 AdaBoost 算法，并解释为什么 AdaBoost 经常可以在训练误差为 0 后继续训练还可能带来测试误差的继续下降。

i. AdaBoost 算法

a) 预设样本为$(x_i, y_i), i = 1,2, \dots N$； 第 m 个弱分类器为$\phi_m(x)$，分类器性能为$\varepsilon_m$，分类器权重为$\alpha_m$，第 m 轮样本权重为$w_{m,i}$

b) 初始化样本权重$w_{1,i} = \frac{1}{N}$

c) 迭代 $m = 1{:}M$

a) 根据当前权重$w_{m,i}$ 训练弱分类器$\phi_m(x)$，要求性能优于随即猜想

b) 计算$\varepsilon_m = \sum_{i=1}^{N} w_{m,i}\, \mathbb{I}(y_i! = \phi_m(x_i))$，$\alpha_m = \frac{1}{2}\log((1-\varepsilon_m)/\varepsilon_m)$

c) 更新$w_{m+1,i} = \frac{w_{m,i}\exp(-\alpha_m y_i\, \phi_m(x_i))}{Z_m}$ 其中Z_m是归一化因子

d) 最终的分类器是$sgn(\sum_m(\alpha_m \phi_m(x)))$

ii. 继续训练类似于增大 margin，虽然训练正确率 100%，但是泛化能力增加。

---

7. （10 分）描述感知机（Perceptron）模型，并给出其权值学习算法。在此基础上，以仅有一个隐含层的三层神经网络为例，形式化描述 Back-Propagation（BP）算法中是如何对隐层神经元与输出层神经元之间的连接权值进行调整的。

i. 感知器模型:

a) 描述: 感知器模型是一种赏罚模型，分类正确就不处罚，分类错误就处罚，直到全部样本都分类正确为止。

b) 公式解释: $y = w^T x$ 期望对于所有x 有 $(y = w^T x) > 0$ ， 因此其损失函数为$J(w) = \sum_{y \in Y} y = \sum_{y_i \in Y} w^T x_i$ 其中Y 是全部错分$(y \leq 0)$ 样本。 因此其权重更新公式为 $w_{i+1} = w_i + \eta \sum_{y_i \in Y} x_i$ ，其中η是超参数"步长"。

ii. BP 算法需要误差σ 的反向传播。预先定义: 每一层的输入向量分别是 x, y, z，真实标签是t ，每一层的输出是$f_{net}(y)$, $f_{net}(z)$ ，转移函数是f_{net} ， 权重是 $w_{i,h}$ 和$w_{\mathrm{h,g}}$

a) 在输出层，误差$\sigma_z = t - f_{net}(z)$ ，对应的输入的误差是$\sigma_{z_in} = \sigma_z f_{net}(z)'$

i. 隐含层→ 输出层的连接权值更新:

$$\Delta w_{\mathrm{h,g}} = {\sigma_{z\_in}}^g * \eta x_{\mathrm{h,g}} = (t^g - f_{net}(z)^g) f_{net}(z)^{g'} \eta x_{\mathrm{h,g}}$$

(如果题干要输入→ 隐含层的权重更新)

b) 在隐含层，输出的误差是输出层输入的加权求和 $\sigma_y = \sum_g w_{\mathrm{h,g}} {\sigma_{z\_in}}^g$

i. 对应的输入的误差是$\sigma_{y_in} = \sigma_y f_{net}(y)'$

ii. 对于输入层→ 隐含层的权重更新$\Delta w_{\mathrm{i,h}} = {\sigma_{y_in}}^g * \eta x_{\mathrm{i,h}}$

---

8. （12 分）已知正例点$\mathbf{x}_1 = (3,3)^{\mathrm{T}}, \mathbf{x}_2 = (4,3)^{\mathrm{T}}$，负例点$\mathbf{x}_3 = (1,1)^{\mathrm{T}}$，试用线性支持向量机的对偶算法求最大间隔分离超平面和分类决策函数，并在图中画出分离超平面、间隔边界及支持向量。

i. 原问题

$$min_{w,b} \ \frac{1}{2}||w||_2^2$$

$$st \ \ y^i\left(w^T x^i + b\right) \geq 1$$

ii. 对偶问题算法

$$\max \sum_{i=1}^{n} \alpha^i - \frac{1}{2} \sum_{i,j=0}^{n} \alpha^i \alpha^j y^i y^j (x^i)^T x^j$$

$$st \ \ \alpha^i \geq 0, i = 1, 2, \ldots n$$

$$\sum_{i=0}^{n} \alpha^i y^i = 0$$

这三个点 显然支持向量是x^1, x^3，因此$\alpha^1 = \alpha^3, \alpha^2 = 0$;

带入$\sum_{i=1}^{n} \alpha^i - \frac{1}{2} \sum_{i,j=0}^{n} \alpha^i \alpha^j y^i y^j (x^i)^T x^j$ 得到 $2\alpha^1 - 4(\alpha^1)^2$ ，求导为 0 得到$\alpha^1 = \frac{1}{4}$

$$\alpha^1 = \frac{1}{4}, \alpha^2 = 0, \alpha^3 = \frac{1}{4}$$

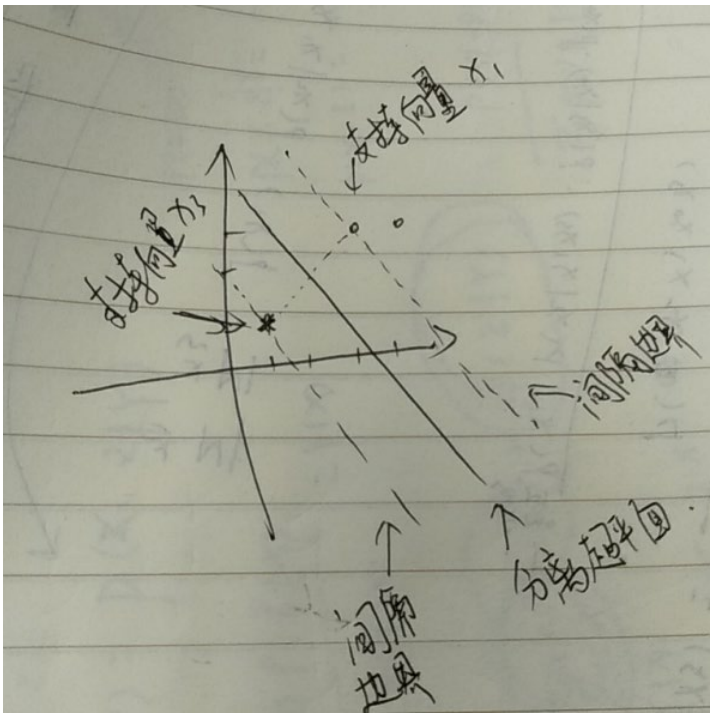
$$w = \sum_{i=0}^{n} \alpha^i y^i x^i = (\frac{1}{2}, \frac{1}{2})^T$$

支持向量是x^1, x^3 ,因此 $w^T x^1 + b = 3 + b = 1; \rightarrow b = -2$

同理$w^T x^3 + b = 1 + b = -1 \rightarrow b = -2$

$(\frac{1}{2},\frac{1}{2})^T x - 2 = 0$ ← 分类面。二维坐标系下 ： $x + y - 4 = 0$

iii. 草图



间隔边界及支持向量。

9. （12 分）假定对一类特定人群进行某种疾病检查，正常人以 ω_1类代表，患病者以 ω_2类代表。设被检查的人中正常者和患病者的先验概率分别为

正常人：$P(\omega_1)=0.9$

患病者：$P(\omega_2)=0.1$

现有一被检查者，其观察值为 x，从类条件概率密度分布曲线上查得

$P(x|\omega_1)=0.2$，$P(x|\omega_2)=0.4$

同时已知风险损失函数为

$$\begin{pmatrix}\lambda_{11} & \lambda_{12}\\ \lambda_{21} & \lambda_{22}\end{pmatrix}=\begin{pmatrix}0 & 6\\ 1 & 0\end{pmatrix}$$

其中 λ_{ij}表示将本应属于第 j 类的模式判为属于第 i 类所带来的风险损失。试对该被检查者用以下两种方法进行分类：

（1） 基于最小错误率的贝叶斯决策，并写出其判别函数和决策面方程；

（2） 基于最小风险的贝叶斯决策，并写出其判别函数和决策面方程。

i. 最小错误率贝叶斯:

a) 判别函数 ：

$$if\ p(x|w_1) * p(w_1) > p(x|w_2) * p(w_2) \rightarrow w_1;$$
$$else\ if\ p(x|w_1) * p(w_1) < p(x|w_2) * p(w_2) \rightarrow w_2$$

b) 决策面方程

$$p(x|w_1) * p(w_1) - p(x|w_2) * p(w_2) = 0$$

c) 决策

$$p(x|w_1) * p(w_1) = 0.18$$

$$p(x|w_2) * p(w_2) = 0.04$$

判决属于w_1。

ii. 最小风险贝叶斯:

a) 判别函数

$$if\ p(x|w_2) * p(w_2)\lambda_{22} + p(x|w_1) * p(w_1)\ \lambda_{21}$$
$$< p(x|w_1) * p(w_1)\ \lambda_{11} + p(x|w_2) * p(w_2)\lambda_{12} \rightarrow w_2$$
$$if\ p(x|w_2) * p(w_2)\lambda_{22} + p(x|w_1) * p(w_1)\ \lambda_{21}$$
$$> p(x|w_1) * p(w_1)\ \lambda_{11} + p(x|w_2) * p(w_2)\lambda_{12} \rightarrow w_1$$

b) 决策面方程

$$p(x|w_2) * p(w_2)\ (\lambda_{22} - \lambda_{12}) + p(x|w_1) * p(w_1)\ (\lambda_{11} - \lambda_{21}) = 0$$

c) 决策:

$$p(x|w_2) * p(w_2)\lambda_{22} + p(x|w_1) * p(w_1)\ \lambda_{21} = 0.18$$
$$p(x|w_1) * p(w_1)\ \lambda_{11} + p(x|w_2) * p(w_2)\lambda_{12} = 0.24$$

故判决属于w_2

---

(2) 基于最小风险的

10. （12 分）假设有 3 个盒子，每个盒子里都装有红、白两种颜色的球。按照下面的方法抽球，产生一个球的颜色的观测序列：开始，以概率 π 随机选取 1 个盒子，从这个盒子里以概率 B 随机抽出 1 个球，记录其颜色后，放回；然后，从当前盒子以概率 A 随机转移到下一个盒子，再从这个盒子里以概率 B 随机抽出一个球，记录其颜色，放回；如此重复进行 3 次，得到一个球的颜色观测序列：O =（红，白，红）。请计算生成该序列的概率P(O|{A, B, π})。

提示：假设状态集合是{盒子 1，盒子 2，盒子 3}，观测的集合是{红，白}，本题中已知状态转移概率分布、观测概率分布和初始概率分布分别为：

$$A = \begin{matrix} \\ 盒子1 \\ 盒子2 \\ 盒子3 \end{matrix} \begin{matrix} 盒子1 & 盒子2 & 盒子3 \end{matrix} \begin{bmatrix} 0.5 & 0.2 & 0.3 \\ 0.3 & 0.5 & 0.2 \\ 0.2 & 0.3 & 0.5 \end{bmatrix}$$

$$B = \begin{matrix} \\ 盒子1 \\ 盒子2 \\ 盒子3 \end{matrix} \begin{matrix} 红 & 白 \end{matrix} \begin{bmatrix} 0.5 & 0.5 \\ 0.4 & 0.6 \\ 0.7 & 0.3 \end{bmatrix}, \pi = [0{,}2, 0.4, 0.4]^T。$$

i. 前向计算需要的公式

$$\alpha(t+1) = \sum_{y_t} \alpha(t)\ A_{y_t, y_t+1}\ B_{y_t+1, x}$$
$$= B_{y_t+1, x} \sum_{y_t} \alpha(t)\ A_{y_t, y_t+1}$$

ii. 计算

$$\alpha(y_t = 1\ , t = 1) = \pi_1 * p(x = 红 | y = 1) = 0.1$$

$$\alpha(y_t = 2\ , t = 1) = 0.4 * 0.4 = 0.16$$
$$\alpha(y_t = 3, t = 1) = 0.4 * 0.7 = 0.28$$

$$\alpha(y_t = 1, t = 2) = 0.5 * (0.1 * 0.5 + 0.16 * 0.3 + 0.28 * 0.2)$$
$$= 0.5 * (0.05 + 0.048 + 0.056) = 0.077$$
$$\alpha(y_t = 2, t = 2) = 0.6 * (0.1 * 0.2 + 0.16 * 0.5 + 0.28 * 0.3)$$
$$= 0.1104$$
$$\alpha(y_t = 3, t = 2) = 0.3 * (0.1 * 0.3 + 0.16 * 0.2 + 0.28 * 0.5) = 0.0606$$

$$\alpha(y_t = 1, t = 3) = 0.5 * (0.077 * 0.5 + 0.1104 * 0.3 + 0.0606 * 0.2)$$
$$= 0.5 * (0.0385 + 0.03312 + 0.01212) = 0.04187$$
$$\alpha(y_t = 2, t = 2) = 0.4 * (0.077 * 0.2 + 0.1104 * 0.5 + 0.0606 * 0.3)$$
$$= 0.4 * (0.0154 + 0.0552 + 0.01818) = 0.035512$$
$$\alpha(y_t = 3, t = 3) = 0.7 * (0.077 * 0.3 + 0.1104 * 0.2 + 0.0606 * 0.5)$$
$$= 0.7 * (0.0231 + 0.02208 + 0.0303) = 0.05283$$

$$p(x) = \sum_{y_t} \alpha(y_t, t = 3) = 0.13021$$

因为没有仔细检查，上述计算可能存在错误！关键是知道前向计算的迭代公式！