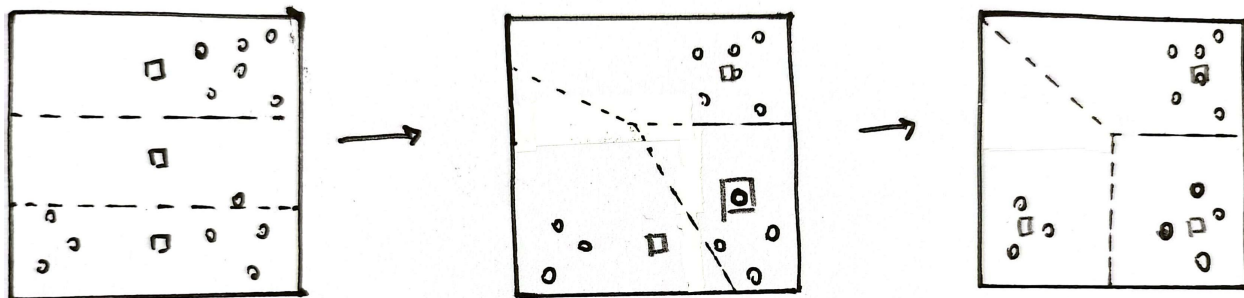


第五章作业

6. 解答. 只需三次迭代即可得到稳定解, 如下图所示



8. 解答. 求解结果及 Python 代码如下:

迭代次数	A 正面期望	A 背面期望	B 正面期望	B 背面期望	A 正面概率	B 正面概率
1	6.82	18.19	17.18	7.81	0.27	0.69
2	5.82	17.2	18.18	8.8	0.25	0.67
3	4.99	16.32	19.01	9.68	0.23	0.66
4	4.27	15.53	19.73	10.47	0.22	0.65
5	3.59	14.75	20.41	11.25	0.2	0.64
6	2.92	13.95	21.08	12.05	0.17	0.64
7	2.21	13.04	21.79	12.96	0.14	0.63
8	1.39	11.96	22.61	14.04	0.1	0.62
9	0.52	10.75	23.48	15.25	0.05	0.61
10	0.03	10.04	23.97	15.96	0	0.6

```

1 import numpy as np
2 import pandas as pd
3
4 # 绘制表格参数
5 cols = ['A 正面期望', 'A 背面期望', 'B 正面期望', 'B 背面期望', 'A 正面概率', 'B 正面
  ↪ 概率']
6 df = pd.DataFrame(columns=cols) # 绘制表格

```

```

7
8 T = 10 # 总迭代次数
9 prA, prB = 0.3, 0.7 # 硬币 A,B 正面朝上的概率
10 samples = [4, 6, 0, 9, 5] # 每个样本中正面朝上的个数
11 for _ in range(T):
12     expectA, expectB = np.zeros(2), np.zeros(2) # 硬币 A,B 的期望
13     for i in range(len(samples)):
14         tmp1 = np.power(prA, samples[i]) * np.power(1 - prA, 10 - samples[i])
15         tmp2 = np.power(prB, samples[i]) * np.power(1 - prB, 10 - samples[i])
16         chooseA = tmp1 / (tmp1 + tmp2) # 选择硬币 A 的概率
17         chooseB = 1 - chooseA # 选择硬币 B 的概率
18         expectA += np.array([samples[i] * chooseA, (10 - samples[i]) * chooseA])
19         expectB += np.array([samples[i] * chooseB, (10 - samples[i]) * chooseB])
20     prA = expectA[0] / np.sum(expectA)
21     prB = expectB[0] / np.sum(expectB)
22     tmp = pd.DataFrame(
23         np.concatenate((expectA, expectB, np.array([prA]),
24             ↪ np.array([prB])))).reshape([1, -1]),
25         columns=cols)
26     df = pd.concat([df, tmp])
27 df = df.reset_index(drop=True)
28 df.index += 1
29 df.index.name = '迭代次数'
30 print(df)
31 df.round(2).to_excel('ans8.xlsx')

```

9. 解答. K 均值聚类算法可视为一种特殊的 EM 算法，聚类质心视为隐变量，求期望步骤中，通过计算欧氏距离判断每个样本点属于哪个聚类质心；在期望最大化中，通过计算均值，更新聚类质心位置，从而减小样本点到聚类质心的方差，增大属于该聚类质心的期望。