一. (30 分) 软间隔 SVM

教材 6.4 节介绍了软间隔概念，用来解决线性不可分情况下的 SVM 问题，同时也可缓解 SVM 训练的过拟合问题。定义松弛变量 $\xi = \{\xi_i\}_{i=1}^m$，其中 $\xi_i > 0$ 表示样本 x_i 对应的间隔约束不满足的程度。软间隔 SVM 问题可以表示为：

$$\max_{w,b} \rho$$

$$\text{s.t. } \frac{y_i(w^\top x_i + b)}{\|w\|_2} \geq \rho, \quad \forall i \in [m].$$

该式显式地表示了分类器的间隔 ρ。基于这种约束形式的表示，可以定义两种形式的软间隔。

- 第一种是绝对软间隔：

$$\frac{y_i({w_i}^\top x_i + b)}{\|w\|_2} \geq \rho - \xi_i.$$

- 第二种是相对软间隔：

$$\frac{y_i({w_i}^\top x_i + b)}{\|w\|_2} \geq \rho(1 - \xi_i).$$

这两种软间隔分别使用 ξ_i 和 $\rho\xi_i$ 衡量错分样本在间隔上的违背程度。在优化问题中加入惩罚项 $C\sum_{i=0}^m \xi_i^p$ （其中 $C > 0, p \geq 1$），使得不满足约束的样本数量尽量小（即让 $\xi_i \to 0$）。

问题：

1. **(10 分)** 软间隔 SVM 通常采用相对软间隔，写出其原问题的形式（要求不包含 ρ）。
2. **(10 分)** 写出采用绝对软间隔的 SVM 原问题（不包含 ρ），并说明为什么一般不使用绝对软间隔来构建 SVM 问题。
3. **(10 分)** 写出 $p = 1$ 情况下软间隔 SVM 的对偶问题。

解:

（1）相对软间隔的 SVM 原问题

相对软间隔 SVM 的原问题在不包含 ρ 的情况下可以表示为:

$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i^p, \quad y_i(w^\top x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad \forall i \in [m]$$

（2）绝对软间隔的 SVM 原问题

绝对软间隔 SVM 的原问题在不包含 ρ 的情况下可以表示为:

$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i^p, \quad y_i(w^\top x_i + b) \geq 1 - \|w\|\xi_i, \qquad \xi_i \geq 0, \quad \forall i \in [m]$$

采用绝对软间隔时，约束条件中出现了 $\|w\|\xi_i$ 项，使得约束条件对 w 和 ξ_i 是非凸的，这导致优化问题变为非凸优化问题，无法保证找到全局最优解，也不能直接应用标准的二次规划方法求解。

而采用相对软间隔时，约束条件为：$y_i(w^\top x_i + b) \geq (1 - \xi_i)$。此时，约束条件对 w 和 ξ_i 都是线性的，目标函数也是凸的，因此整个优化问题是一个凸二次规划问题，可以高效地找到全局最优解。

（3）$p = 1$ 情况下软间隔 SVM 的对偶问题

软间隔支持向量机（SVM）在 $p = 1$ 时的原始优化问题为:

$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i, \quad y_i(w^\top x_i + b) \geq 1 - \xi_i, \quad i = 1, 2, \ldots, m, \quad \xi_i \geq 0, \quad i = 1, 2, \ldots, m$$

构建拉格朗日函数:

$$\mathcal{L}(w, b, \xi; \alpha, \beta) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i - \sum_{i=1}^{m}\alpha_i[y_i(w^\top x_i + b) - (1 - \xi_i)] - \sum_{i=1}^{m}\beta_i\xi_i$$

其对偶问题为:

$$\begin{aligned}
\max_{\alpha,\beta} \min_{w,b,\xi} \quad & \mathcal{L}(w, b, \xi; \alpha, \beta) \\
\text{s.t.} \quad & \alpha_i \geq 0, \quad i = 1, 2, \ldots m \\
& \beta_i \geq 0, \quad i = 1, 2, \ldots m
\end{aligned}$$

上式内层对 w, b, ξ 的优化属于无约束优化问题，则令偏导为零:

$$\begin{aligned}
&\frac{\partial \mathcal{L}}{\partial w} = 0 \implies w = \sum_{i=1}^{m}\alpha_i y_i x_i \\
&\frac{\partial \mathcal{L}}{\partial b} = 0 \implies \sum_{i=1}^{m}\alpha_i y_i = 0 \\
&\frac{\partial \mathcal{L}}{\partial \xi} = 0 \implies \beta_i = C - \alpha_i
\end{aligned}$$

代入得：

$$\begin{aligned}\mathcal{L}(w,b,\xi;\alpha,\beta) &= \frac{1}{2}\|w\|^2 - \sum_{i=1}^{m}\alpha_i y_i w^\top x_i - b\sum_{i=1}^{m}\alpha_i y_i + \sum_{i=1}^{m}(C-\alpha_i-\beta_i)\xi_i + \sum_{i=1}^{m}\alpha_i \\ &= \frac{1}{2}\left(\sum_{i=1}^{m}\alpha_i y_i x_i\right)^\top \left(\sum_{j=1}^{m}\alpha_j y_j x_j\right) + \sum_{i=1}^{m}\alpha_i - \sum_{i=1}^{m}\alpha_i y_i w^\top x_i \\ &= \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j x_i^\top x_j\end{aligned}$$

因此对偶问题为：

$$\begin{aligned}\max_{\alpha} \quad & \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j x_i^\top x_j \\ \text{s.t.} \quad & \sum_{i=1}^{m}\alpha_i y_i = 0 \\ & 0 \le \alpha_i \le C, \quad i = 1,2,\ldots,m\end{aligned}$$

二. (20 分) SVM 编程

设想你正在进行一项客户数据的分类任务，目标是通过支持向量机（SVM）构建一个模型，准确地区分两类客户。以下是你的任务要求：

已知数据集

我们有一个二维数据集，其中包含两个类别的点，数据如下：

数据点编号	x_1	x_2	类别
1	1.0	2.0	1
2	2.0	3.5	1
3	1.5	1.0	1
4	3.0	3.0	1
5	2.0	1.5	1
6	8.0	8.5	2
7	9.0	10.0	2
8	8.5	9.5	2
9	7.0	7.5	2
10	6.5	9.0	2

任务要求

1. **(10 分)** 用 Python 训练一个支持向量机分类模型，使用 `scikit-learn` 中的 `SVC` 来分类上表中的数据。要求：
   (a) 训练一个非线性核（如 RBF 核）的支持向量机模型。
   (b) 输出支持向量，并绘制分类边界。

   请给出 SVM 模型训练过程的完整代码以及实验结果的截图。

2. **(10 分)** 假设你希望提高模型的泛化能力，请完成以下任务：

   通过网格搜索优化 SVM 的**惩罚参数** C 和**核系数** γ 。请尝试 C 取值 $[0.1, 1, 10, 100]$ 和 γ 取值 $[0.1, 1, 10]$，找出最佳参数组合，并在优化后输出训练准确率和支持向量。同时，总结惩罚参数 C 和核系数 γ 是如何影响分类效果和模型的泛化能力的。

   提示： 网格搜索是一种用于调优模型超参数的简单方法。它会在给定的参数范围内尝试所有可能的参数组合，选择效果最好的组合。

解:

(1) 使用 Python 训练 SVM 分类模型

```python
import numpy as np
import matplotlib.pyplot as plt
from sklearn.svm import SVC
from sklearn.model_selection import GridSearchCV

X = np.array([
    [1.0, 2.0],
    [2.0, 3.5],
    [1.5, 1.0],
    [3.0, 3.0],
    [2.0, 1.5],
    [8.0, 8.5],
    [9.0, 10.0],
    [8.5, 9.5],
    [7.0, 7.5],
    [6.5, 9.0]
])
y = np.array([1, 1, 1, 1, 1, 2, 2, 2, 2, 2])

model = SVC(kernel='rbf')
model.fit(X, y)

print("支持向量：")
print(model.support_vectors_)

plt.scatter(X[:, 0], X[:, 1], c=y, cmap='coolwarm', s=100, label='Data Points')
plt.scatter(model.support_vectors_[:, 0], model.support_vectors_[:, 1], s=150, facecolors='none', edgecolors='k', label='Support Vectors')
plt.legend()

xx, yy = np.meshgrid(np.linspace(0, 10, 500), np.linspace(0, 12, 500))
Z = model.decision_function(np.c_[xx.ravel(), yy.ravel()])
Z = Z.reshape(xx.shape)

plt.contourf(xx, yy, Z, levels=[-1, 0, 1], alpha=0.1, colors=['blue', 'red', 'purple'])
plt.contour(xx, yy, Z, levels=[0], linewidths=2, colors='black')
plt.xlabel('$x_1$')
plt.ylabel('$x_2$')
plt.title("SVM Decision Boundary and Support Vectors")
plt.show()
```

Listing 1: RBF 核的 SVM 分类模型

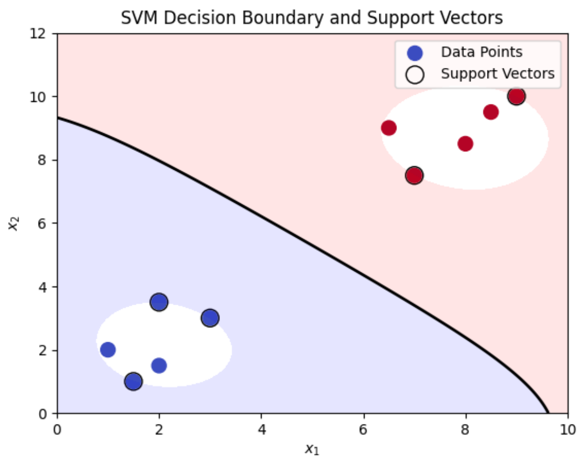

图 1: SVM 分类边界

```
支持向量:
[[ 2.   3.5]
 [ 1.5  1. ]
 [ 3.   3. ]
 [ 9.  10. ]
 [ 7.   7.5]]
```

图 2: 支持向量

(2) 使用网格搜索优化 SVM 模型

```python
import numpy as np
import matplotlib.pyplot as plt
from sklearn.svm import SVC
from sklearn.model_selection import GridSearchCV

X = np.array([
    [1.0, 2.0],
    [2.0, 3.5],
    [1.5, 1.0],
    [3.0, 3.0],
    [2.0, 1.5],
    [8.0, 8.5],
    [9.0, 10.0],
    [8.5, 9.5],
    [7.0, 7.5],
    [6.5, 9.0]
])
y = np.array([1, 1, 1, 1, 1, 2, 2, 2, 2, 2])

param_grid = {
    'C': [0.1, 1, 10, 100],
    'gamma': [0.1, 1, 10]
}

grid_search = GridSearchCV(SVC(kernel='rbf'), param_grid, cv=5)
grid_search.fit(X, y)

print("最佳参数组合: ", grid_search.best_params_)
print("最佳训练准确率: ", grid_search.best_score_)

best_model = grid_search.best_estimator_
```

```python

print("优化后的支持向量：")
print(best_model.support_vectors_)

plt.scatter(X[:, 0], X[:, 1], c=y, cmap='coolwarm', s=100, label='Data Points')
plt.scatter(best_model.support_vectors_[:, 0], best_model.support_vectors_[:, 1], s=150, facecolors='none
    ', edgecolors='k', label='Support Vectors')
plt.legend()

xx, yy = np.meshgrid(np.linspace(0, 10, 500), np.linspace(0, 12, 500))
Z = best_model.decision_function(np.c_[xx.ravel(), yy.ravel()])
Z = Z.reshape(xx.shape)

plt.contourf(xx, yy, Z, levels=[-1, 0, 1], alpha=0.1, colors=['blue', 'red', 'purple'])
plt.contour(xx, yy, Z, levels=[0], linewidths=2, colors='black')
plt.xlabel('$x_1$')
plt.ylabel('$x_2$')
plt.title("SVM Decision Boundary with Optimized Parameters")
plt.show()
```

Listing 2: 网格搜索优化的 SVM 分类模型

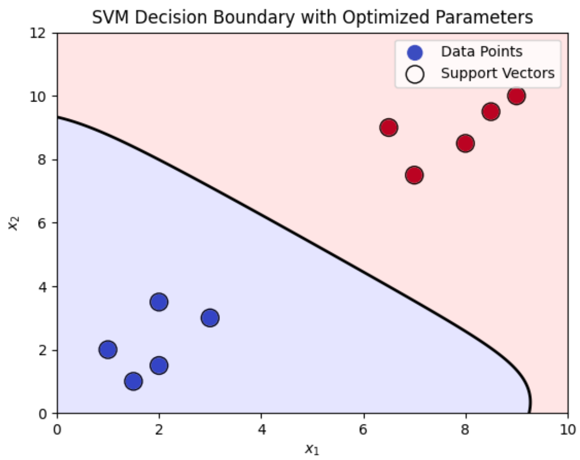

图 3: 优化后的 SVM 分类边界

```
最佳参数组合： {'C': 0.1, 'gamma': 0.1}
最佳训练准确率： 1.0
优化后的支持向量：
[[ 1.   2. ]
 [ 2.   3.5]
 [ 1.5  1. ]
 [ 3.   3. ]
 [ 2.   1.5]
 [ 8.   8.5]
 [ 9.  10. ]
 [ 8.5  9.5]
 [ 7.   7.5]
 [ 6.5  9. ]]
```

图 4: 优化后的支持向量

惩罚参数 C 可以控制模型对误分类样本的容忍度。较大的 C 值倾向于对误分类样本更敏感，从而获得更低的训练误差，但可能会导致模型复杂化，泛化能力降低。较小的 C 值允许模型在训练时容忍一些误分类，以便获得更好的泛化能力。

核系数 γ 可以控制单个支持向量的影响范围。较高的 γ 值会使模型变得更复杂，容易过拟合；较低的 γ 值会使得支持向量的影响范围更广，从而提升泛化能力，但也可能导致欠拟合。

三. (20 分) 朴素贝叶斯分类器

一家电商公司希望通过用户评论的关键词来预测评论的情感（**正面**或 **负面**)。假设已经收集了一小部分用户评论，并从中提取出以下五个关键词作为特征：**“good”**、**“bad”**、**“quality”**、**“price”** 和 **“recommend”**。每条评论可以被标记为“正面”或“负面”。

评论情感	good 出现次数	bad 出现次数	quality 出现次数	price 出现次数	recommend 出现次数
正面评论	50	5	45	20	60
负面评论	10	30	5	25	2

假设正面评论和负面评论的先验概率分别为 $P(\text{正面评论}) = 0.7$ 和 $P(\text{负面评论}) = 0.3$。

问题

1. **(8 分)** 基于上述数据，使用**拉普拉斯修正**（$\alpha = 1$）计算以下条件概率：
   - $P(\text{good}|\text{正面评论})$
   - $P(\text{bad}|\text{正面评论})$
   - $P(\text{quality}|\text{正面评论})$
   - $P(\text{price}|\text{正面评论})$
   - $P(\text{recommend}|\text{正面评论})$

   同时，计算上述特征在负面评论下的条件概率。

2. **(12 分)** 假设我们有一条新评论 $X = \{\text{good}, \text{quality}, \text{price}\}$，请使用朴素贝叶斯分类器计算该评论属于**正面评论**和**负面评论**的后验概率 $P(\text{正面评论}|X)$ 和 $P(\text{负面评论}|X)$，并根据计算结果确定该评论的情感类别。

提示：

- 本题的答案请以分式或者小数点后两位的形式给出，比如 P=0.67.
- 在计算条件概率时，请注意考虑拉普拉斯修正后的分母变化。
- 最终的后验概率可以使用贝叶斯公式结合条件独立性假设：

  $$P(y|X) = \frac{P(y) \cdot P(X|y)}{P(X)}$$

  因为 $P(X)$ 是相同的常数项，比较 $P(y) \cdot P(X|y)$ 即可。

解:

（1）使用拉普拉斯修正计算条件概率

对于正面评论：$|D| = 180$ ，$N = 5$

$$P(\text{good}|\text{正面评论}) = \frac{50+1}{185} = 0.28$$

$$P(\text{bad}|\text{正面评论}) = \frac{5+1}{185} = 0.03$$

$$P(\text{quality}|\text{正面评论}) = \frac{45+1}{185} = 0.25$$

$$P(\text{price}|\text{正面评论}) = \frac{20+1}{185} = 0.11$$

$$P(\text{recommend}|\text{正面评论}) = \frac{60+1}{185} = 0.33$$

对于负面评论：$|D| = 72$ ，$N = 5$

$$P(\text{good}|\text{负面评论}) = \frac{10+1}{77} = 0.14$$

$$P(\text{bad}|\text{负面评论}) = \frac{30+1}{77} = 0.40$$

$$P(\text{quality}|\text{负面评论}) = \frac{5+1}{77} = 0.08$$

$$P(\text{price}|\text{负面评论}) = \frac{25+1}{77} = 0.34$$

$$P(\text{recommend}|\text{负面评论}) = \frac{2+1}{77} = 0.04$$

（2）计算后验概率

对于正面评论：

$$\begin{aligned} P(\text{正面评论}|X) &= \frac{P(\text{正面评论})}{P(X)} \cdot P(\text{good}|\text{正面评论}) \cdot P(\text{quality}|\text{正面评论}) \cdot P(\text{price}|\text{正面评论}) \\ &\approx 0.00545 \cdot \frac{1}{P(X)} \end{aligned}$$

对于负面评论：

$$\begin{aligned} P(\text{负面评论}|X) &= \frac{P(\text{负面评论})}{P(X)} \cdot P(\text{good}|\text{负面评论}) \cdot P(\text{quality}|\text{负面评论}) \cdot P(\text{price}|\text{负面评论}) \\ &\approx 0.00113 \cdot \frac{1}{P(X)} \end{aligned}$$

$P(\text{正面评论}|X) > P(\text{负面评论}|X)$，因此，这条新评论 X 属于正面评论。

四. (30 分) EM 算法

假设有一个包含 6 次硬币抛掷结果的数据集，记录了每次抛掷是否得到“正面”：

$X = \{$正面, 正面, 反面, 正面, 反面, 反面$\}$

假设这些结果是由两枚硬币 A 和 B 生成的，每次抛掷时选择使用硬币 A 或 B 的概率均为 0.5。然而，具体每次抛掷使用的是哪一枚硬币是未知的。硬币 A 和 B 的正面概率分别为 θ_A 和 θ_B。我们的目标是通过 EM 算法估计这两枚硬币的正面概率 θ_A 和 θ_B。

已知：1. 初始参数：硬币 A 的正面概率 $\theta_A^{(0)} = 0.6$ 和硬币 B 的正面概率 $\theta_B^{(0)} = 0.5$。2. 每次抛掷使用硬币 A 和硬币 B 的概率均为 0.5，即 $P(A) = 0.5$ 和 $P(B) = 0.5$。

请通过一轮 EM 算法的迭代步骤，估计硬币 A 和 B 的正面概率 θ_A 和 θ_B。本题的答案请以分式或者小数点后两位的形式给出，比如 P=0.67.

问题：

1. **E 步**（15 分）：对于每一次抛掷结果，使用当前的参数估计值（$\theta_A^{(0)}$ 和 $\theta_B^{(0)}$），计算该结果由硬币 A 和硬币 B 生成的后验概率，即每次抛掷属于硬币 A 和硬币 B 的“软分配”概率。

   请计算以下内容：

   - 在第 1 次到第 6 次抛掷中，每个结果（正面或反面）由硬币 A 生成的概率 $P(A|x_i)$。
   - 每个结果由硬币 B 生成的概率 $P(B|x_i)$。

2. **M 步**（15 分）：基于 E 步计算出的“软分配”概率，计算硬币 A 和 B 的正面和反面出现的期望次数，并更新硬币 A 和 B 的正面概率 θ_A 和 θ_B。

   请计算以下内容：

   - 硬币 A 的正面和反面期望出现次数，并据此更新硬币 A 的正面概率 $\theta_A^{(1)}$。
   - 硬币 B 的正面和反面期望出现次数，并据此更新硬币 B 的正面概率 $\theta_B^{(1)}$。

解:

(1) E 步

计算似然函数:

- 对于正面(H): $P(H|A)=\theta_A^{(0)}=0.6$, $P(H|B)=\theta_B^{(0)}=0.5$
- 对于反面(T): $P(T|A)=1-\theta_A^{(0)}=0.4$, $P(T|B)=1-\theta_B^{(0)}=0.5$

$$P(A|x_i)=\frac{P(x_i|A)\cdot P(A)}{P(x_i)}=\frac{P(x_i|A)}{P(x_i|A)+P(x_i|B)}$$
$$P(B|x_i)=\frac{P(x_i|B)\cdot P(B)}{P(x_i)}=\frac{P(x_i|B)}{P(x_i|A)+P(x_i|B)}$$

第 1 次抛掷(正面):

$$P(A|x_1)=\frac{0.6\times 0.5}{0.6\times 0.5+0.5\times 0.5}=\frac{6}{11}\approx 0.55$$
$$P(B|x_1)=1-P(A|x_1)=\frac{5}{11}\approx 0.45$$

第 2 次抛掷(正面):

$$P(A|x_2)=\frac{6}{11}\approx 0.55,\quad P(B|x_2)=\frac{5}{11}\approx 0.45$$

第 3 次抛掷(反面):

$$P(A|x_3)=\frac{0.4\times 0.5}{0.4\times 0.5+0.5\times 0.5}=\frac{4}{9}\approx 0.44$$
$$P(B|x_3)=1-P(A|x_3)=\frac{5}{9}\approx 0.56$$

第 4 次抛掷(正面):

$$P(A|x_4)=\frac{6}{11}\approx 0.55,\quad P(B|x_4)=\frac{5}{11}\approx 0.45$$

第 5 次抛掷(反面):

$$P(A|x_5)=\frac{4}{9}\approx 0.44,\quad P(B|x_5)=\frac{5}{9}\approx 0.56$$

第 6 次抛掷(反面):

$$P(A|x_6)=\frac{4}{9}\approx 0.44,\quad P(B|x_6)=\frac{5}{9}\approx 0.56$$

综上,

- 第 1 次抛掷(正面): $P(A|x_1)=\frac{6}{11}\approx 0.55$, $P(B|x_1)=\frac{5}{11}\approx 0.45$
- 第 2 次抛掷(正面): $P(A|x_2)=\frac{6}{11}\approx 0.55$, $P(B|x_2)=\frac{5}{11}\approx 0.45$
- 第 3 次抛掷(反面): $P(A|x_3)=\frac{4}{9}\approx 0.44$, $P(B|x_3)=\frac{5}{9}\approx 0.56$
- 第 4 次抛掷(正面): $P(A|x_4)=\frac{6}{11}\approx 0.55$, $P(B|x_4)=\frac{5}{11}\approx 0.45$
- 第 5 次抛掷(反面): $P(A|x_5)=\frac{4}{9}\approx 0.44$, $P(B|x_5)=\frac{5}{9}\approx 0.56$
- 第 6 次抛掷(反面): $P(A|x_6)=\frac{4}{9}\approx 0.44$, $P(B|x_6)=\frac{5}{9}\approx 0.56$

(2) M 步

- 对于硬币 A：
  - 正面期望次数 E_H^A：

$$E_H^A = E_{H,1}^A + E_{H,2}^A + E_{H,3}^A + E_{H,4}^A + E_{H,5}^A + E_{H,6}^A = \frac{6}{11} + \frac{6}{11} + 0 + \frac{6}{11} + 0 + 0 = \frac{18}{11} \approx 1.64$$

  - 反面期望次数 E_T^A：

$$E_T^A = 0 + 0 + \frac{4}{9} + 0 + \frac{4}{9} + \frac{4}{9} = \frac{12}{9} = \frac{4}{3} \approx 1.33$$

- 对于硬币 B：
  - 正面期望次数：

$$E_H^B = \frac{5}{11} + \frac{5}{11} + 0 + \frac{5}{11} + 0 + 0 = \frac{15}{11} \approx 1.37$$

  - 反面期望次数：

$$E_T^B = 0 + 0 + \frac{5}{9} + 0 + \frac{5}{9} + \frac{5}{9} = \frac{15}{9} = \frac{5}{3} \approx 1.67$$

使用期望出现次数来更新 $\theta_A^{(1)}$ 和 $\theta_B^{(1)}$：

$$\theta_A^{(1)} = \frac{E_H^A}{E_H^A + E_T^A} = \frac{27}{49} \approx 0.55$$

$$\theta_B^{(1)} = \frac{E_H^B}{E_H^B + E_T^B} = \frac{9}{20} = 0.45$$