第一部分：简述题

1.1

adaboost 算法的设计思想为：给定训练集，寻找比较粗糙的分类规则（弱分类器）要比寻找精确的分类规则要简单得多。所以从弱学习算法出发，反复学习，得到一系列弱分类器；然后组合这些弱分类器，构成一个强分类器。

adaboost 算法的主要计算步骤是：

(1) 初始化训练数据的权值分布为均匀分布

(2) 使用具有权值分布的训练数据，学习弱分类器

(3) 计算步骤 (2) 中得到分类器的加权分类错误率

(4) 由步骤 (3) 中的分类错误率得到弱分类器的贡献值

(5) 更新训练数据集的权重分布，使本次错分的训练数据得到更高的权重

(6) 不断重复 (2)-(5) 步，得到 M 个弱分类器

(7) 将 M 个弱分类器按照各自的贡献值线性加权组合，得到最终分类器

1.2

给定一组样本，总共有 c 个类别，各类符合高斯分布，均值和协方差矩阵都未知，通过最大似然估计法对这些参数进行估计。可以得出第 i 类的均值为各样本数据的加权和，样本权重为该样本属于第 i 类的概率。同理第 i 类的协方差矩阵的计算也需要乘样本的权重。K-Means 聚类法可以看作是简化的混合高斯密度函数估计，它把聚类中的每一个样本都以 $0,1$ 概率分配给某一个混合成分（也就是说，样本属于且只属于一类分布），且各混合成分协方差相等，均为对角矩阵 $\sigma^2 I$。算法步骤是：

(1) 给定聚类数，初始化聚类中心

(2) 将样本点根据与聚类中心的距离分类

(3) 将每一类的聚类中心更新为该类样本点的均值

(4) 重复 (2)、(3) 步直到聚类中心不发生变化

预设聚类个数、初始化类别中心的方法、距离度量方法，会对 K-Means 聚类算法的性能产生影响。

1.3

谱聚类算法建立在图论的谱图理论基础之上，其本质是将聚类问题转化为一个图上的关于顶点划分的最优问题。经典算法的步骤是：

(1) 利用点对之间的相似性构建亲和度矩阵

(2) 构建拉普拉斯矩阵

(3) 求解一组拉普拉斯矩阵最小特征值对应的特征向量（舍弃零特征值对应的特征向量分量全相等的特征向量）(4) 用这组特征向量对样本点降维

(5) 使用 K-Means 等聚类方法完成最终聚类

谱聚类算法中，拉普拉斯矩阵构造过程中边与边之间权重的定义方法、子图的切割的定义方法、算法分类个数的超参，最终聚类算法的选择及参数设定会对性能产生影响。

第二部分：编程题

具体程序请见 k-means.py 文件，数据集在 data.npy 文件中存储。

(1) 初始聚类中心为随机取 5 个样本点，最终得到的聚类中心为 (1.02, -0.98)、(5.47, -4.38)、(0.92, 3.93)、(6.03, 4.46)、(8.97, -0.19)。经过 5 次迭代后聚类中心不发生改变，类内方差和为 1952.38，计算中心与真实均值之间的误差（MSE）为 0.065。最终聚类结果如图 1 所示。

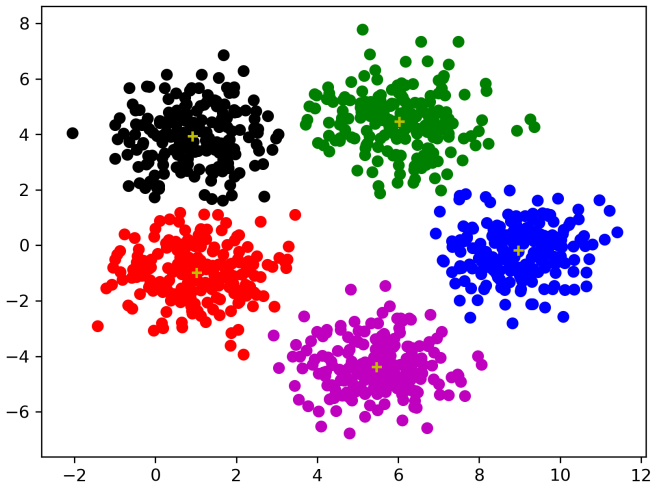

图 1: 随机选择中心

(2) 初始聚类中心为从第一类样本点中取 5 个点，最终得到的聚类中心为 (1.02, -0.98)、(5.47, -4.38)、(0.92, 3.93)、(6.03, 4.46)、(8.97, -0.19)。经过 8 次迭代后聚类中心不发生改变，类内方差和为 1952.39，计算中心与真实均值之间的误差（MSE）为 0.065。最终聚类结果如图 2 所示。

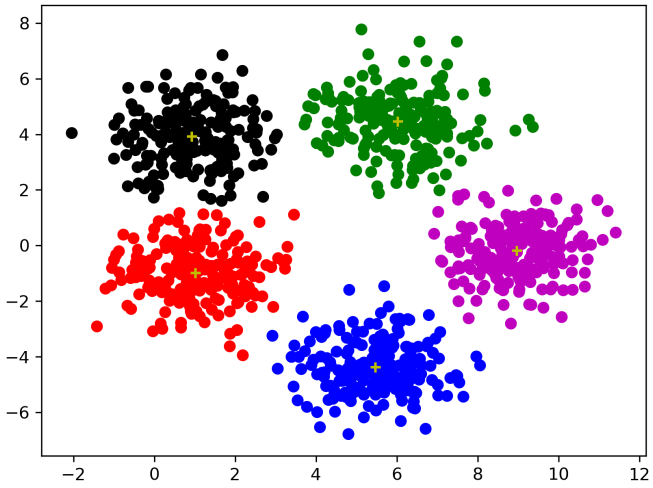

图 2: 初始中心均为第一类样本