机器学习第五次作业

姓名：刘文晨　　　　　学号：202222280328

题目 1

题面：

从网上下载或自己编程实现 TSVM 算法 选择两个 UCI 数据集，将其中 30%的样例用作测试样本，10%的样例用作有标记样本，60%的样例用作无标记样本。分别训练出利用无标记样本的 TSVM 以及仅利用有标记样本的 SVM，并比较其性能。

解：

选择最常用的 iris 数据集，将数据集标准化之后，将其中 30 个样例用作测试样本，10 个样例用作有标记样本，60 个样例用作无标记样本。以 sklearn 的 SVM 算法为基础建立 TSVM。模型训练好后，输出经过有标记的样本训练后对未标记样本的预测正确率、经过 TSVM 训练后，对未标记样本的预测正确率和经过 TSVM 训练后对测试样本的预测正确率。最后绘制散点图和分别由 SVM 和 TSVM 得到的超平面。代码如下：

```python
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from sklearn import svm
from sklearn.datasets import load_iris
from sklearn.preprocessing import MinMaxScaler


def create_data():
    iris = load_iris()
    df = pd.DataFrame(iris.data, columns=iris.feature_names)
    df['label'] = iris.target
    df.columns = ['sepal length', 'sepal width', 'petal length', 'petal width', 'label']
    data = np.array(df.iloc[:100, [0, 1, -1]])
    # 对数据进行标准化处理
    sc = MinMaxScaler()
    sc.fit(data)
    data = sc.transform(data)
```

```
        data[:, -1] = data[:, -1] * 2 - 1
        # 30 个测试样本
        test = np.vstack((data[:15], data[50:65]))
        # 10 个有标记样本
        labeled_sample = np.vstack((data[15:20], data[65:70]))
        # 60 个无标记样本
        unlabeled_sample = np.vstack((data[20:50], data[70:]))
        return test, labeled_sample, unlabeled_sample


    test, labeled, unlabeled = create_data()
    clf = svm.SVC(C=1, kernel='linear')
    # 有标记的样本训练 SVM
    clf.fit(labeled[:, :2], labeled[:, -1])
    positive_labeled = labeled[5:]
    negative_labeled = labeled[:5]
    plt.scatter(labeled[:5, :2][:, 0], labeled[:5, :2][:, 1], color='red', s=40, label=-1)
    plt.scatter(labeled[5:, :2][:, 0], labeled[5:, :2][:, 1], color='blue', s=40, label=1)
    x_points = np.linspace(0, 1, 10)
    y_points = -(clf.coef_[0][0] * x_points + clf.intercept_) / clf.coef_[0][1]
    plt.plot(x_points, y_points, color='green')
    plt.legend()
    # 伪标记
    fake_label = clf.predict(unlabeled[:, :2])
    unlabeled_positive_x = []
    unlabeled_positive_y = []
    unlabeled_negative_x = []
    unlabeled_negative_y = []
    for i in range(len(unlabeled)):
        if int(fake_label[i]) == 1:
            unlabeled_positive_x.append(unlabeled[i, 0])
            unlabeled_positive_y.append(unlabeled[i, 1])
        else:
            unlabeled_negative_x.append(unlabeled[i, 0])
            unlabeled_negative_y.append(unlabeled[i, 1])

    plt.scatter(unlabeled_positive_x, unlabeled_positive_y, color='red', s=15)
    plt.scatter(unlabeled_negative_x, unlabeled_negative_y, color='blue', s=15)

    print('经过有标记的样本训练后，对未标记样本的预测正确率为{}'.format(clf.score(unlabeled[:, :2], unlabeled[:, -1])))
```

```

    Cu = 0.1
    Cl = 1  # 初始化 Cu,Cl
    weight = np.ones(len(labeled) + len(unlabeled))
    # 样本权重
    weight[len(unlabeled):] = Cu
    # 用于训练有标记与无标记样本集合
    train_sample = np.vstack((labeled[:, :2], unlabeled[:, :2]))
    # 用于训练的标记集合
    train_label = np.hstack((labeled[:, -1], fake_label))
    unlabeled_id = np.arange(len(unlabeled))

    while Cu < Cl:
        clf.fit(train_sample, train_label, sample_weight=weight)
        while True:
            # 通过训练得到的预测标记
            predicted_y = clf.decision_function(unlabeled[:, :2])
            # 伪标记，这里为与预测的区分开，写为 real_y
            real_y = fake_label
            epsilon = 1 - predicted_y * real_y
            positive_set, positive_id = epsilon[real_y > 0], unlabeled_id[real_y > 0]
            negative_set, negative_id = epsilon[real_y < 0], unlabeled_id[real_y < 0]
            positive_max_id = positive_id[np.argmax(positive_set)]
            negative_max_id = negative_id[np.argmax(negative_set)]
            epsilon1, epsilon2 = epsilon[positive_max_id], epsilon[negative_max_id]
            if epsilon1 > 0 and epsilon2 > 0 and round(epsilon1 + epsilon2, 3) >= 2:
                fake_label[positive_max_id] = -fake_label[positive_max_id]
                fake_label[negative_max_id] = -fake_label[negative_max_id]
                train_label = np.hstack((labeled[:, -1], fake_label))
                clf.fit(train_sample, train_label, sample_weight=weight)
            else:
                break
        # 更新 Cu
        Cu = min(2 * Cu, Cl)
        # 更新样本权重
        weight[len(unlabeled):] = Cu
    # 绘图
    x_points = np.linspace(0, 1, 10)
    y_points = -(clf.coef_[0][0] * x_points + clf.intercept_) / clf.coef_[0][1]
```

```
    plt.plot(x_points, y_points, color='yellow')
    plt.savefig('运行结果.jpg')
    plt.show()
    # 打印结果
    print('经过 TSVM 训练后，对未标记样本的预测正确率为
{}'.format(clf.score(unlabeled[:, :2], unlabeled[:, -1])))
    print('经过 TSVM 训练后，对测试样本的预测正确率为
{}'.format(clf.score(test[:, :2], test[:, -1])))
```

运行程序后，我们得到：经过有标记的样本训练后，对未标记样本的预测正确率为 0.9666666666666667，经过 TSVM 训练后，对未标记样本的预测正确率为 0.9833333333333333，经过 TSVM 训练后，对测试样本的预测正确率为 1.0。

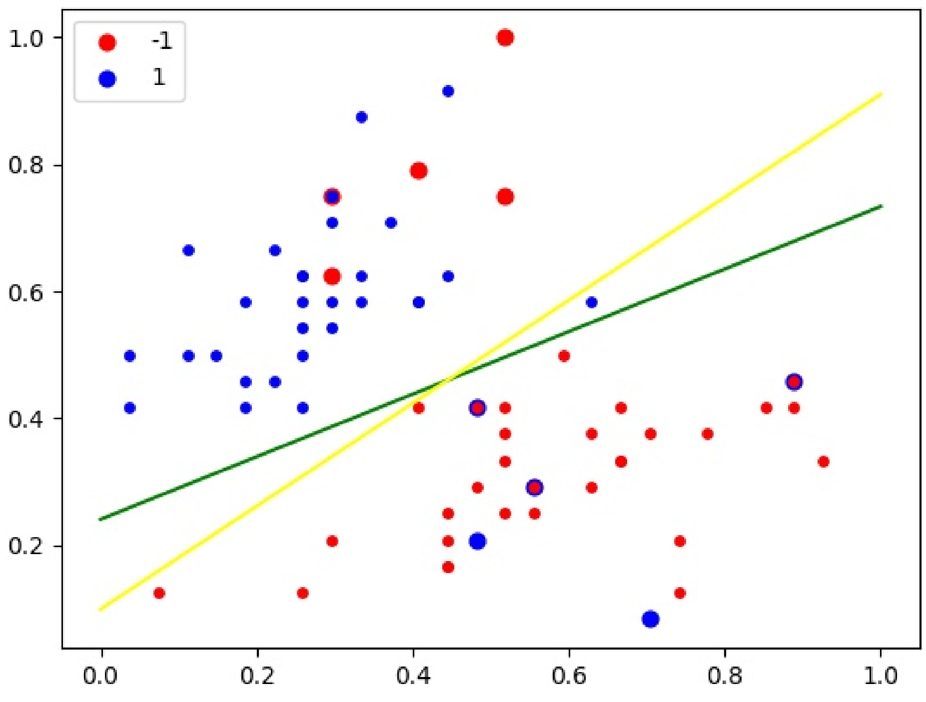

图 1 散点图与超平面

程序绘制的散点图和超平面如图 1 所示。其中，绿线是仅利用有标记样本的 SVM 得到的超平面，黄线是利用无标记样本的 TSVM 模型得到的超平面。

由实验结果可知，对于 iris 数据集，TSVM 通过利用未标记数据能提高最终分类的准确率，从 SVM 的 96.67%提高到了 TSVM 的 98.33%，并且预测标记与测试集的真实标记一致，预测正确率为 100%。

题目 2

题面：

假设数据由混合专家（mixture of experts）模型生成，即数据是基于k个成分混合而得的概率密度生成：$p(x|\theta)=\sum_{i=1}^{k}\alpha_i\cdot p(x|\theta_i)$，其中，$\theta=\{\theta_1,\theta_2,\dots,\theta_k\}$是模型参数，$p(x|\theta_i)$是第$i$个混合成分的概率密度，混合系数$\alpha_i\geq 0$，$\sum_{i=1}^{k}\alpha_i=1$。假设每个混合成分对应一个类别，但每个类别可能包含多个混合成分。试推导相应的生成式半监督学习算法。

解：

首先，我们假定：

- 数据X包含$M=l+u$个样本：$X=\{\mathbf{x}_j\},j=1,\dots,M$
- 所有样本中共有$|\mathcal{C}|$个类别：c_j表示样本的类别，$c_j\in\mathcal{C}$
- 混合模型含有N个混合成分，$\{m_j=i\},i=1,\dots,N$表示样本$\mathbf{x}_j$可能的混合成分，θ_i表示对应混合成分的模型参数，则相应模型可以表示为$f(\mathbf{x}_j\mid\theta_i)=p(\mathbf{x}_j\mid m_j=i,\theta_i)=p(\mathbf{x}_j\mid\theta_i)$

在此处：

$$
\begin{aligned}
LL(D_l\cup D_u)&=\sum_{(\mathbf{x}_i,c_j)\in D_l}\ln p(\mathbf{x}_j,c_j\mid\theta)+\sum_{\mathbf{x}_i\in D_u}\ln p(\mathbf{x}_j\mid\theta)\\
&=\sum_{(\mathbf{x}_i,c_j)\in D_l}\ln\sum_{i=1}^{N}\alpha_i p(c_j\mid\mathbf{x}_j,m_j=i,\theta_i)p(\mathbf{x}_j\mid m_j=i,\theta_i)+\sum_{\mathbf{x}_i\in D_u}\ln\sum_{i=1}^{N}\alpha_i p(\mathbf{x}_j\mid m_j=i,\theta_i)\\
&=\sum_{(\mathbf{x}_i,c_j)\in D_l}\ln\sum_{i=1}^{N}\alpha_i p(c_j\mid\mathbf{x}_j,m_j=i,\theta_i)f(\mathbf{x}_j\mid\theta_i)+\sum_{\mathbf{x}_i\in D_u}\ln\sum_{i=1}^{N}\alpha_i f(\mathbf{x}_j\mid\theta_i)
\end{aligned}
$$

我们采用广义混合模型（The Generalized Mixture Model, GM），采用高斯分布作为混合成分，来推导 EM 算法的更新参数。

显然，此时：

$$f(\mathbf{x}_j\mid\theta_i)=p(\mathbf{x}_j\mid\theta_i)=p(\mathbf{x}_j\mid\mu_i,\Sigma_i)$$

则第一个式子变为：

$$LL(D_l\cup D_u)=\sum_{(\mathbf{x}_i,c_j)\in D_l}\ln\sum_{i=1}^{N}\alpha_i p(c_j\mid\mathbf{x}_j,m_j=i,\mu_i,\Sigma_i)p(\mathbf{x}_j\mid\mu_i,\Sigma_i)+\sum_{\mathbf{x}_i\in D_u}\ln\sum_{i=1}^{N}\alpha_i p(\mathbf{x}_j\mid\mu_i,\Sigma_i)$$

代入 GM 公式，得：

$$LL(D_l\cup D_u)=\sum_{(\mathbf{x}_i,c_j)\in D_l}\ln\sum_{i=1}^{N}\alpha_i\beta_{c_j|i}p(\mathbf{x}_j\mid\mu_i,\Sigma_i)+\sum_{\mathbf{x}_i\in D_u}\ln\sum_{i=1}^{N}\alpha_i p(\mathbf{x}_j\mid\mu_i,\Sigma_i)$$

我们的目的是要求得最优的$\alpha_i, \beta_{c_j|i}, \mu_i, \sum_i$使上式取得最大值。

对于混合系数α_i，除了要最大化$LL(D_l \cup D_u)$，还应满足隐含条件：$\alpha_i \geq 0$，$\sum_{i=1}^{N} \alpha_i = 1$，因此考虑对$LL(D_l \cup D_u)$使用拉格朗日乘子法，变为优化：$LL(D_l \cup D_u) + \lambda(\sum_{i=1}^{N} \alpha_i - 1)$，代入$LL(D_l \cup D_u)$的计算式，得到：

$$\frac{\partial LL(D_l \cup D_u)}{\partial \alpha_i} = \sum_{\mathbf{x}_j \in D_l} \frac{\beta_{c_j|i} \cdot p(\mathbf{x}_j \mid \mu_i, \Sigma_i)}{\sum_{i=1}^{N} \alpha_i \cdot \beta_{c_j|i} \cdot p(\mathbf{x}_j \mid \mu_i, \Sigma_i)} + \sum_{\mathbf{x}_j \in D_u} \frac{p(\mathbf{x}_j \mid \mu_i, \Sigma_i)}{\sum_{i=1}^{N} \alpha_i \cdot p(\mathbf{x}_j \mid \mu_i, \Sigma_i)} + \lambda = 0$$

令：

$$p(m_j = i \mid c_j, \mathbf{x}_j, \mu_i, \Sigma_i) = \frac{\alpha_i \cdot \beta_{c_j|i} \cdot p(\mathbf{x}_j \mid \mu_i, \Sigma_i)}{\sum_{i=1}^{N} \alpha_i \cdot \beta_{c_j|i} \cdot p(\mathbf{x}_j \mid \mu_i, \Sigma_i)}$$

同时，将高斯模型代入方程，得：

$$p(m_j = i \mid \mathbf{x}_j, \mu_i, \Sigma_i) = \frac{\alpha_i \cdot p(\mathbf{x}_j \mid \mu_i, \Sigma_i)}{\sum_{i=1}^{N} \alpha_i \cdot p(\mathbf{x}_j \mid \mu_i, \Sigma_i)}$$

$$0 = \sum_{\mathbf{x}_j \in D_l} p(m_j = i \mid c_j, \mathbf{x}_j, \mu_i, \Sigma_i) + \sum_{\mathbf{x}_j \in D_u} p(m_j = i \mid \mathbf{x}_j, \mu_i, \Sigma_i) + \alpha_i \cdot \lambda$$

令上式对所有高斯混合成分求和：

$$\begin{aligned} 0 &= \sum_{\mathbf{x}_j \in D_l} \sum_{i=1}^{N} p(m_j = i \mid c_j, \mathbf{x}_j, \mu_i, \Sigma_i) + \sum_{\mathbf{x}_j \in D_u} \sum_{i=1}^{N} p(m_j = i \mid \mathbf{x}_j, \mu_i, \Sigma_i) + \alpha_i \cdot \lambda \\ &= \sum_{\mathbf{x}_j \in D_l} 1 + \sum_{\mathbf{x}_j \in D_u} 1 + \lambda \\ &= M + \lambda \end{aligned}$$

令$\lambda = -M$，将其带入上式，得：

$$\alpha_i = \frac{1}{M} \cdot \left(\sum_{\mathbf{x}_j \in D_l} p(m_j = i \mid c_j, \mathbf{x}_j, \mu_i, \Sigma_i) + \sum_{\mathbf{x}_j \in D_u} p(m_j = i \mid \mathbf{x}_j, \mu_i, \Sigma_i) \right)$$

对于高斯分布，其偏导具有如下性质：

$$\frac{\partial p(\mathbf{x} \mid \mu_i, \Sigma_i)}{\partial \mu_i} = p(\mathbf{x} \mid \mu_i, \Sigma_i) \cdot \Sigma_i^{-1} \cdot (\mu_i - \mathbf{x})$$
$$\frac{\partial p(\mathbf{x} \mid \mu_i, \Sigma_i)}{\partial \Sigma_i} = p(\mathbf{x} \mid \mu_i, \Sigma_i) \cdot \Sigma_i^{-2} \cdot ((\mathbf{x} - \mu_i)(\mathbf{x} - \mu_i)^\top - \Sigma_i)$$

求偏导，得：

$$\begin{aligned}\frac{\partial LL(D_l \cup D_u)}{\partial \mu_i} &= \sum_{\mathbf{x}_j \in D_l} \frac{\alpha_i \cdot \beta_{c_j|i} \cdot p(\mathbf{x}_j \mid \mu_i, \Sigma_i)}{\sum_{i=1}^{N} \alpha_i \cdot \beta_{c_j|i} \cdot p(\mathbf{x}_j \mid \mu_i, \Sigma_i)} \cdot \Sigma_i^{-1} \cdot (\mu_i - \mathbf{x}_j) + \sum_{\mathbf{x}_j \in D_u} \frac{\alpha_i \cdot p(\mathbf{x}_j \mid \mu_i, \Sigma_i)}{\sum_{i=1}^{N} \alpha_i \cdot p(\mathbf{x}_j \mid \mu_i, \Sigma_i)} \cdot \Sigma_i^{-1} \cdot (\mu_i - \mathbf{x}_j) \\ &= \sum_{\mathbf{x}_j \in D_l} p(m_j = i \mid c_j, \mathbf{x}_j, \mu_i, \Sigma_i) \cdot \Sigma_i^{-1} \cdot (\mu_i - \mathbf{x}_j) + \sum_{\mathbf{x}_j \in D_u} p(m_j = i \mid \mathbf{x}_j, \mu_i, \Sigma_i) \cdot \Sigma_i^{-1} \cdot (\mu_i - \mathbf{x}_j) \\ &= \Sigma_i^{-1} \cdot \left(\sum_{\mathbf{x}_j \in D_l} p(m_j = i \mid c_j, \mathbf{x}_j, \mu_i, \Sigma_i) \cdot (\mu_i - \mathbf{x}_j) + \sum_{\mathbf{x}_j \in D_u} p(m_j = i \mid \mathbf{x}_j, \mu_i, \Sigma_i) \cdot (\mu_i - \mathbf{x}_j) \right)\end{aligned}$$

继续计算：

$$\mu_i = \frac{1}{M\alpha_i} \cdot \left(\sum_{\mathbf{x}_j \in D_l} \mathbf{x}_j \cdot p(m_j = i \mid c_j, \mathbf{x}_j, \mu_i, \Sigma_i) + \sum_{\mathbf{x}_j \in D_u} \mathbf{x}_j \cdot p(m_j = i \mid \mathbf{x}_j, \mu_i, \Sigma_i) \right)$$

$$\begin{aligned}\frac{\partial LL(D_l \cup D_u)}{\partial \Sigma_i} &= \sum_{\mathbf{x}_j \in D_l} \frac{\alpha_i \cdot \beta_{c_j|i} \cdot p(\mathbf{x}_j \mid \mu_i, \Sigma_i)}{\sum_{i=1}^{N} \alpha_i \cdot \beta_{c_j|i} \cdot p(\mathbf{x}_j \mid \mu_i, \Sigma_i)} \cdot \Sigma_i^{-2} \cdot ((\mathbf{x}_j - \mu_i)(\mathbf{x}_j - \mu_i)^\top - \Sigma_i) \\ &\quad + \sum_{\mathbf{x}_j \in D_u} \frac{\alpha_i \cdot p(\mathbf{x}_j \mid \mu_i, \Sigma_i)}{\sum_{i=1}^{N} \alpha_i \cdot p(\mathbf{x}_j \mid \mu_i, \Sigma_i)} \cdot \Sigma_i^{-2} \cdot ((\mathbf{x}_j - \mu_i)(\mathbf{x}_j - \mu_i)^\top - \Sigma_i) \\ &= \sum_{\mathbf{x}_j \in D_l} p(m_j = i \mid c_j, \mathbf{x}_j, \mu_i, \Sigma_i) \cdot \Sigma_i^{-2} \cdot ((\mathbf{x}_j - \mu_i)(\mathbf{x}_j - \mu_i)^\top - \Sigma_i) \\ &\quad + \sum_{\mathbf{x}_j \in D_u} p(m_j = i \mid \mathbf{x}_j, \mu_i, \Sigma_i) \cdot \Sigma_i^{-2} \cdot ((\mathbf{x}_j - \mu_i)(\mathbf{x}_j - \mu_i)^\top - \Sigma_i)\end{aligned}$$

$$\begin{aligned}\Sigma_i = \frac{1}{M\alpha_i} \cdot \Bigg(&\sum_{\mathbf{x}_j \in D_l} p(m_j = i \mid c_j, \mathbf{x}_j, \mu_i, \Sigma_i) \cdot ((\mathbf{x}_j - \mu_i)(\mathbf{x}_j - \mu_i)^\top) \\ &+ \sum_{\mathbf{x}_j \in D_u} p(m_j = i \mid \mathbf{x}_j, \mu_i, \Sigma_i) \cdot ((\mathbf{x}_j - \mu_i)(\mathbf{x}_j - \mu_i)^\top) \Bigg)\end{aligned}$$

对于混合系数$\beta_{k|i}$，除了要最大化$LL(D_l \cup D_u)$，还应满足隐含条件：$\beta_{k|i} \geq 0$，$\sum_{k=1}^{|C|} \beta_{k|i} = 1$，因此考虑对$LL(D_l \cup D_u)$使用拉格朗日乘子法，变为优化：$LL(D_l \cup D_u) + \lambda(\sum_{k=1}^{|C|} \beta_{k|i} - 1)$，代入$LL(D_l \cup D_u)$的计算式，得到：

$$\frac{\partial LL(D_l \cup D_u)}{\partial \beta_{k|i}} = \sum_{\mathbf{x}_j \in D_l \wedge c_j = k} \frac{\alpha_i \cdot p(\mathbf{x}_j \mid \mu_i, \Sigma_i)}{\sum_{i=1}^{N} \alpha_i \cdot \beta_{c_j|i} \cdot p(\mathbf{x}_j \mid \mu_i, \Sigma_i)} + \lambda = 0$$

$$\begin{aligned} 0 &= \sum_{\mathbf{x}_j \in D_l \wedge c_j = k} \frac{\alpha_i \cdot \beta_{k|i} \cdot p(\mathbf{x}_j \mid \mu_i, \Sigma_i)}{\sum_{i=1}^{N} \alpha_i \cdot \beta_{c_j|i} \cdot p(\mathbf{x}_j \mid \mu_i, \Sigma_i)} + \beta_{k|i} \cdot \lambda \\ &= \sum_{\mathbf{x}_j \in D_l \wedge c_j = k} p(m_j = i \mid c_j, \mathbf{x}_j, \mu_i, \Sigma_i) + \beta_{k|i} \cdot \lambda \end{aligned}$$

$$\begin{aligned} 0 &= \sum_{k=1}^{|\mathcal{C}|} \sum_{\mathbf{x}_j \in D_l \wedge c_j = k} p(m_j = i \mid c_j, \mathbf{x}_j, \mu_i, \Sigma_i) + \sum_{k=1}^{|\mathcal{C}|} \beta_{k|i} \cdot \lambda \\ &= \sum_{\mathbf{x}_j \in D_l} p(m_j = i \mid c_j, \mathbf{x}_j, \mu_i, \Sigma_i) + \lambda \end{aligned}$$

$$\lambda = -\sum_{\mathbf{x}_j \in D_l} p(m_j = i \mid c_j, \mathbf{x}_j, \mu_i, \Sigma_i)$$

最终得到：

$$\beta_{k|i} = \frac{\sum_{\mathbf{x}_j \in D_l \wedge c_j = k} p(m_j = i \mid c_j, \mathbf{x}_j, \mu_i, \Sigma_i)}{\sum_{\mathbf{x}_j \in D_l} p(m_j = i \mid c_j, \mathbf{x}_j, \mu_i, \Sigma_i)}$$