机器学习-第2次作业

张志博 2017211416

2019 年 11 月 22 日

1 《统计学习方法》

习题7.1、7.2

1.1 习题7.1

比较感知机的对偶形式与线性可分支持向量机的对偶形式。

1.1.1 对偶形式

对偶形式的基本思想是，将w和b表示为实例x_i和标记y_i的线性组合的形式，通过求解其系数而求得w和b，对误分类点(x_i, y_i)通过

$$\begin{cases} w \leftarrow w + \mu y_i x_i \\ b \leftarrow b + \mu y_i \end{cases}$$

逐步修改w, b，设修改n次，则w, b关于(x_i, y_i)的增量分别是$\alpha_i y_i x_i$和$\alpha_i y_i$，这里$\alpha_i = n_i \mu$。最后学习到的w, b可以分别表示为

$$\begin{cases} w = \sum\limits_{i=1}^{N} \alpha_i y_i x_i \\ b = \sum\limits_{i=1}^{N} \alpha_i y_i \end{cases}$$

1.1.2 感知机学习算法的对偶形式

输入：线性可分的数据集$T = \{(x_1, y_1), (x_2, y_2), \ldots (x_N, y_N))\}$，其中$x_i \in R^n$，$y_i \in -1, +1$，$i = 1, 2, \ldots, N$；学习率$\mu (0 < \mu \leq 1)$

输出：α，b；感知机模型$f(x) = sign(\sum_{j=1}^{N} \alpha_j y_j x_j x+b)$。其中$\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_N)^T$

(1)

$$\begin{cases} \alpha \leftarrow 0 \\ b \leftarrow 0 \end{cases}$$

(2) 在训练集中选取数据(x_i, y_i)

(3) 如果$(\sum_{j=1}^{N} \alpha_j x_j y_j x_i + b) \leq 0$

$$\begin{cases} \alpha_i \leftarrow \alpha_i + \mu \\ b \leftarrow b + \mu y_i \end{cases}$$

(4) 转至（**2**）直至没有误分类数据

w, b实质是将其表示为x_i, y_i的线性组合形式：

$$\begin{cases} w = \sum_{i=1}^{N} \alpha_i^* j_i x_i \\ b = \sum_{i=1}^{N} \alpha_i^* j_i \end{cases}$$

1.1.3 支持向量机学习算法的对偶形式

原始问题的对偶问题是

$$\begin{cases} \min\limits_{\alpha} \frac{1}{2} \sum\limits_{i=1}^{N} \sum\limits_{j=1}^{N} \alpha^i \alpha^j y^i y^j x_i x_j - \sum\limits_{i=1}^{N} \alpha_i \\ s.t. \ \sum\limits_{i=1}^{N} \alpha_i y_i = 0 \\ 0 \leq \alpha_i \leq C, \ i = 1, 2, \ldots, N \end{cases}$$

求解对偶问题后得到的w, b实质是将其表示为x_i, y_i的线性组合形式：

$$\begin{cases} w^* = \sum\limits_{i=1}^{N} \alpha_i^* j_i x_i \\ b^* = y_j - \sum\limits_{i=1}^{N} \alpha_i^* x_i j_i \end{cases}$$

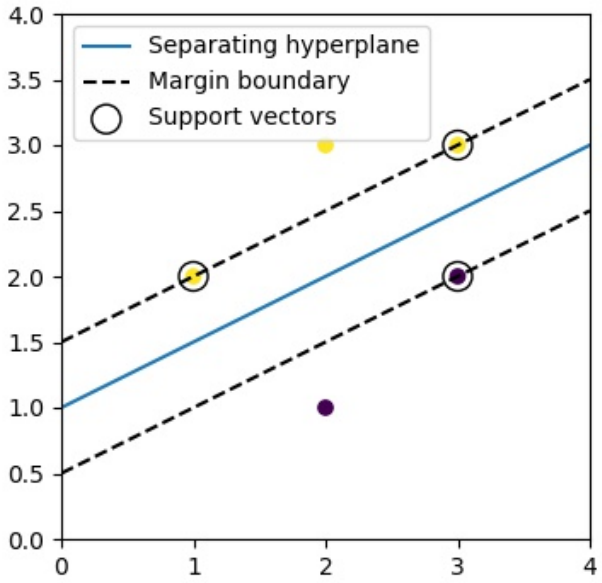
1.2 习题7.2

试求最大间隔分离超平面和分类决策函数，并在图上画出分离超平面、间隔边界及支持向量。

```
w1: -1.0000000000000004
w2: 2.0000000000000013
b: -2.0000000000000004

Process finished with exit code 0
```

使用了**Python**的库**numpy**、**matplotlib.pyplot**、**sklearn.svm**，采用线性核的**SVM**，惩罚系数**C=10**（因为数据集过小，所以采用了很高的惩罚系数，这样结果和手动计算得到的系数更加接近），计算后得到结果，最大间隔分离超平面：$-x+2y-2=0$，分类决策函数为$f(x)=sign(-x+2y-2=0)$



分离超平面、间隔边界及支持向量如上图

2 《机器学习》

习题6.3

2.1 习题6.3

选择两个UCI数据集，分别用线性核和高斯核训练一个SVM，与C4.5决策树进行实验比较

训练集、测试集的划分使用了sklearn.model_selection.train_test_split，具体参数如下图，训练集和测试集比例为7：3，让random_state=1使几次训练集和测试集的划分一致，减少无关变量。

使用了Python的库sklearn，计算后得到结果

```
iris
linear-SVM score: 1.0
The number of support vectors is:  [ 3 10  9]
rbf-SVM score: 0.9777777777777777
The number of support vectors is:  [ 4 15 18]
------------------------------------------------
wine
linear-SVM score: 0.9629629629629629
The number of support vectors is:  [4 7 7]
rbf-SVM score: 0.35185185185185186
The number of support vectors is:  [36 52 36]

Process finished with exit code 0
```

对于数据集**iris**，用线性核训练的**SVM**，总共有**22**个支持向量，在测试集上的准确率是**100%**，由此可见，**iris**数据集是线性可分的；用高斯核训练的**SVM**，总共有**37**个支持向量，在测试集上的准确率是**97.8%**，效果不如线性核；

对于数据集**wine**，用线性核训练的**SVM**，总共有**18**个支持向量，在测试集上的准确率是**96.3%**；用高斯核训练的**SVM**，总共有**124**个支持向量，然而数据大小才只有**178**，在测试集上的准确率只有**35.2%**，可能是由于特征过多，也可能是因为使用默认参数，导致发生了过拟合。

```
---------------------------------IRIS---------------
未剪枝C4.5决策树在测试数据集上的分类正确率为: 84.44%

后剪枝C4.5决策树在测试数据集上的分类正确率为: 95.55%

预剪枝C4.5决策树在测试数据集上的分类正确率为: 95.55%

---------------------------------WINE---------------
未剪枝C4.5决策树在测试数据集上的分类正确率为: 96.29%

后剪枝C4.5决策树在测试数据集上的分类正确率为: 96.29%

预剪枝C4.5决策树在测试数据集上的分类正确率为: 96.29%

Process finished with exit code 0
```

对于数据集**iris**，用后剪枝的**C4.5**决策树在测试集上的准确率是**95.6%**，低于**SVM**方法；对于数据集**wine**，用**C4.5**决策树在测试集上的准确率是**96.3%**，

低于线性核训练的SVM，高于高斯核训练的SVM。