LB2 机器学习概论

PB19151769　　马宇骁

目录

1 实验要求

完成类 ‘SVM1‘ 和 ‘SVM2‘，并且使用不同的算法去寻找支持向量机的解。更具体地说，因为解决支持向量机的关键在于解决书本上的二次规划问题（6.6），你只需要使用两种不同的方法去解决（6.6）。剩下的部分，比如预测，内容可以相同。

在完成了类方法的部分之后，你需要测试你代码的效率。比较应当包含以下内容：

1. 正确率，
2. 计算（训练）的时间消耗。

禁止使用 “ sklearn“ 或者其他的机器学习库，你只被允许使用 ‘numpy‘, ‘pandas‘, ‘matplotlib‘, 和 Standard Library, 你需要从头开始编写这个项目。

2 实验原理

支持向量机（support vector machines, SVM）是一种二分类模型，它的基本模型是定义在特征空间上的间隔最大的线性分类器，间隔最大使它有别于感知机；SVM 还包括核技巧，这使它成为实质上的非线性分类器。SVM 的的学习策略就是间隔最大化，可形式化为一个求解凸二次规划的问题，也等价于正则化的合页损失函数的最小化问题。SVM 的的学习算法就是求解凸二次规划的最优化算法。

2.1 SVM 算法原理

SVM 学习的基本想法是求解能够正确划分训练数据集并且几何间隔最大的分离超平面。即为分离超平面，对于线性可分的数据集来说，这样的超平面有无穷多个（即感知机），但是几何间隔最大的分离超平面却是唯一的。

假设我们要寻找的最优分割平面：

$$\mathrm{w}^* \cdot \mathrm{x} + \mathrm{b}^* = 0$$

把一个新的特征带入等号左边，如果大于 0，说明为正类，小于 0，说明为负类。

2.2 SMO 描述算法

SVM 的基本思路确定我们要优化的目标函数，即最大化几何间隔。在优化时我们取倒数，所以就是最小化目标函数。通过拉格朗日乘数法来对目标函数进行优化，拉格朗日乘数法需要引入新的参数集 α.

$$\mathrm{L}(\mathrm{w}, \mathrm{b}, \mathrm{a}) = \frac{1}{2}\|\mathrm{w}\|^2 - \sum_{\mathrm{i}=1}^{\mathrm{N}} \alpha_{\mathrm{i}} \mathrm{y}_{\mathrm{i}} (\mathrm{w} \cdot \mathrm{x}_{\mathrm{i}} + \mathrm{b}) + \sum_{\mathrm{i}=1}^{\mathrm{N}} \alpha_{\mathrm{i}}$$

原始问题：

$$\min_{\mathrm{w},\mathrm{b}} \max_{\alpha} \mathrm{L}(\mathrm{w}, \mathrm{b}, \mathrm{a})$$

对偶问题，可以转换成拉格朗日对偶问题从而变成极大极小问题：

$$\max_{\alpha} \min_{\mathrm{w,b}} \mathrm{L(w, b, a)}$$

3 实验实现

3.1 算法流程

1. 初始化 α 参数集。数量等于 example 的数量。每个 α 与 x 、y 对应；

2. 选择两个 α，分别记为 α_1 α_2（第二个我用随机取）；

3. 使用 α_1，α_2 计算误差 E1，E2；

$$\begin{gathered}\mathrm{g(x)} = \sum_{\mathrm{i}=1}^{\mathrm{N}} \alpha_{\mathrm{i}} \mathrm{y_i} \mathrm{K}\left(\mathrm{x_i}, \mathrm{x}\right) + \mathrm{b} \\ \mathrm{E_i} = \mathrm{g}\left(\mathrm{x_i}\right) - \mathrm{y_i}\end{gathered}$$

4. 计算范围，$\mathrm{y_1 != y_2}$:

$$\begin{gathered}\mathrm{L} = \max\left(0, \alpha_2^{\mathrm{old}} - \alpha_1^{\mathrm{old}}\right) \\ \mathrm{H} = \min\left(\mathrm{C}, \mathrm{C} + \alpha_2^{\mathrm{old}} - \alpha_1^{\mathrm{old}}\right)\end{gathered}$$

$\mathrm{y_1 == y_2}$:

$$\begin{gathered}\mathrm{L} = \max\left(0, \alpha_2^{\mathrm{old}} + \alpha_1^{\mathrm{old}} - \mathrm{C}\right) \\ \mathrm{H} = \min\left(\mathrm{C}, \alpha_2^{\mathrm{old}} + \alpha_1^{\mathrm{old}}\right)\end{gathered}$$

5. 计算 η：

$$\eta = \mathrm{K_{11}} + \mathrm{K_{22}} - 2\ \mathrm{K_{12}}$$

6. 计算 α_2^{new}，并且限制在 H,L 区间内

$$\begin{gathered}\alpha_2^{\mathrm{new}} = \alpha_2^{\mathrm{old}} + \frac{\mathrm{y_2(E_1 - E_2)}}{\eta} \\ \alpha_2^{\mathrm{new}} = \mathrm{clip}\left(\alpha_2^{\mathrm{new}}, \mathrm{L}, \mathrm{H}\right)\end{gathered}$$

$$\alpha_1^{\mathrm{new}} = \alpha_1^{\mathrm{old}} + \mathrm{y_1 y_2}\left(\alpha_2^{\mathrm{old}} - \alpha_2^{\mathrm{new}}\right)$$

7. 对于 b 的更新：

$$\begin{gathered}\mathrm{b_1^{new}} = -\mathrm{E_1} - \mathrm{y_1}\ \mathrm{K_{11}}\left(\alpha_1^{\mathrm{new}} - \alpha_1^{\mathrm{old}}\right) - \mathrm{y_2}\ \mathrm{K_{21}}\left(\alpha_2^{\mathrm{new}} - \alpha_2^{\mathrm{old}}\right) + \mathrm{b^{old}} \\ \mathrm{b_2^{new}} = -\mathrm{E_2} - \mathrm{y_1}\ \mathrm{K_{12}}\left(\alpha_1^{\mathrm{new}} - \alpha_1^{\mathrm{old}}\right) - \mathrm{y_2}\ \mathrm{K_{22}}\left(\alpha_2^{\mathrm{new}} - \alpha_2^{\mathrm{old}}\right) + \mathrm{b^{old}} \\ \mathrm{b} = \left(\mathrm{b_1^{new}} + \mathrm{b_2^{new}}\right)/2\end{gathered}$$

8. 转到第 2 步，直到收敛。

最后训练好的参数预测时使用 sign() 函数。

3.2 实验结果

由于沿用了之前的数据集（5,100），样本较少，两种算法只是采用的核函数不同：线性与多项式（**2）。训练时间如下：

用时 1.454028 秒
用时 1.553001 秒

训练曲线：

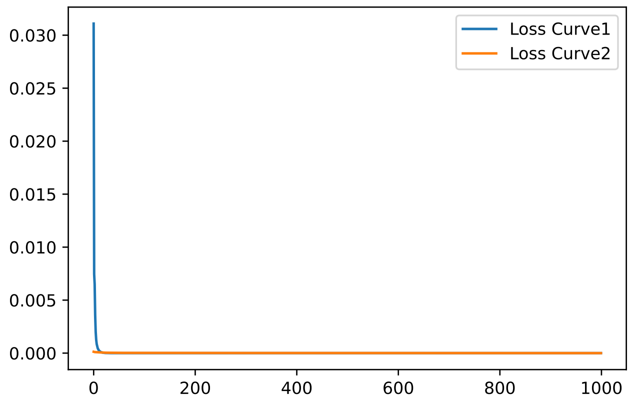

图 1: Loss Curve

训练预测准确率如下：

0.7721518987341772
0.620253164556962