Author: 陆星宇 Date: 2021/2/27

第7章 支持向量机(SVM)

一、交流讨论

问题1：7.3.3中介绍的多项式，高斯核函数，字符串核函数各自有什么样的应用场景？

多项式核函数：低维空间线性不可分的模式通过非线性映射到高维特征空间则可能实现线性可分，而多项式核函数避免了高维展开的维度灾难。

高斯核函数：指数的计算量大，参数的挑选十分重要，在特征线性可分性较差的情况下，效果较好。

问题2：支持向量机的优点与缺点各有那些？

优点：

1. SVM利用内积核函数代替向高维空间的非线性映射；
2. 支持向量在SVM分类决策中起决定作用，支持向量较少，可以预先存储维度较低的Gram矩阵，避免唯独灾难，因此分类的计算复杂度较低

缺点：

1. 大规模训练样本的训练过程中，需要计算样本数量阶的矩阵，时间空间复杂度较高
2. 由于分离超平面的优劣只由离其最近的少数点决定，因此对噪声十分敏感。
3. 传统的SVM不适合多分类

问题3：感知机与支持向量机的相同点与不同点有哪些？

相同点：

1. 都是以超平面分割特征空间实现二分类器

不同点：

1. 感知机平等对待所有样本数据；支持向量机只关注支持向量。
2. 感知机采用梯度下降法，收敛迅速；支持向量机算法较为复杂。
3. 感知机只注重解并不唯一；支持向量机求解唯一的间隔最大的分离超平面。
4. 感知机没有核方法的概念，无法用于非线性可分的模式。

问题4：SVM的对偶形式为什么能够降低计算量？

1. 对于特征向量具有较高纬度的情况，原问题维数较高，复杂度较大。
2. 对偶问题的子问题具有解析解，收敛较快。

二、内容概要

2.1 线性可分支持向量机与硬间隔最大化

2.1.1 线性可分支持向量机

分类决策函数： $f(x) = sign(w \cdot x + b)$

分离超平面： $w \cdot x + b = 0$ 分离超平面是间隔最大的超平面，具有唯一性。

2.1.2 间隔

函数间隔： $\gamma = min_{i=1,2,...,N}[y_i(w \cdot x_i + b)]$ 既可表示距离也能表示是否正确

几何间隔： $\gamma = min_{i=1,2,...,N}[y_i \cdot \frac{w \cdot x_i + b}{|w|}]$ 考虑到对超平面来说系数具有可缩放性，规范化更加合理

2.1.3 间隔最大化

间隔最大化等价于下列凸二次规划问题：

$$min \ \frac{1}{2}||w||^2$$
$$s.t. \ \ 1 - y_i(w \cdot x_i + b) \leq 0, \ i = 1, 2, \ldots, N$$

2.1.4 对偶算法

利用KKT条件可解得：

$$w = \sum_{i=1}^{N} \alpha_i y_i x_i$$
$$\sum_{i=1}^{N} \alpha_i y_i = 0$$

带入得等价对偶最优化问题：

$$min \ \frac{1}{2} \sum_{i=1}^{N} \sum_{i=1}^{N} \alpha_i \alpha_i y_i y_j (x_i \cdot x_j) - \sum_{i=1}^{N} \alpha_i$$
$$s.t. \ \ \sum_{i=1}^{N} \alpha_i y_i = 0$$
$$\alpha_i \geq 0, \ i = 1, 2, \ldots, N$$

还可证明

$$b = y_j - \sum_{i=1}^{N} \alpha_i y_i (x_i \cdot x_j)$$

2.2 线性支持向量机与软间隔最大化

2.2.1 线性支持向量机

对于并不能完全线性可分的数据集，添加松弛变量，优化问题修改为：

$$min \ \frac{1}{2}||w||^2 + C \sum_{i=1}^{N} \xi_i$$
$$s.t. \ \ y_i(w \cdot x_i + b) \geq 1 - \xi_i, \ i = 1, 2, \ldots, N$$
$$\xi_i \geq 0, \ i = 1, 2, \ldots, N$$

可以证明w的解是唯一的，b的解则存在一个区间。

2.2.2 对偶算法

原始问题等价的对偶问题是

$$min\ \frac{1}{2}\sum_{i=1}^{N}\sum_{i=1}^{N}\alpha_i\alpha_i y_i y_j(x_i\cdot x_j)-\sum_{i=1}^{N}\alpha_i$$

$$s.t.\ \ \sum_{i=1}^{N}\alpha_i y_i=0$$

$$0\le\alpha_i\le C,\ i=1,2,\dots,N$$

2.2.3 合页损失函数

线性支持向量机学习还有另外一种解释，就是最小化以下目标函数：

$$\sum_{i=1}^{N}[1-y_i(w\cdot x_i+b)]_+ +\lambda||w||^2$$

其中$L(y_(w\cdot x+b))=[1-y(w\cdot x+b)]_+$被称为合页损失函数

其中

$$[z]_+=\begin{cases}z, & z>0\\ 0, & z\le 0\end{cases}$$

最小化合页损失函数与原问题目标函数等价，但对学习有更高的要求。

2.3 非线性支持向量机与核函数

2.3.1 (正定)核函数

$$K(x,z)=\phi(x)\cdot\phi(z)$$

用$\phi(x)$代替x,则可将对偶问题目标函数与分类决策函数分别修改为

$$W(\alpha)=\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j K(x_i,x_j)-\sum_{i=1}^{N}\alpha_i$$

$$f(x)=sign(g(x))=sign(\sum_{i=1}^{N}a_i y_i K(x_i,x)+b)$$

正定核的充要条件是：

$K(x,z)$对应的Gram矩阵是半正定矩阵

$$[K(x,z)]_{gram}=[K(x_i,x_j)]_{N\times N}$$

2.3.2 常用核函数

多项式核函数：$K(x,z)=(x\cdot z+1)^p$

高斯核函数：$K(x,z)=exp(-\frac{||x-z||^2}{2\sigma^2})$

字符串核函数

2.4 序列最小最优化算法(SMO)

2.4.1 两个变量二次规划的求解方法

$$\alpha_2^{new,unc}=\alpha_2^{old}+\frac{y_2[(g(x_1)-y_1)-(g(x_2)-y_2)]}{K_{11}+K_{22}-2K12}$$

$$\alpha_2^{new} = \begin{cases} L, & \alpha_2^{new,unc} < H \\ \alpha_2^{new,unc}, & L < \alpha_2^{new,unc} < H \\ H, & \alpha_2^{new,unc} > H \end{cases}$$

其中

$$L = \begin{cases} max(0, \alpha_2^{old} + \alpha_1^{old} - C), y_1 = y_2 \\ max(0, \alpha_2^{old} - \alpha_1^{old}), y_1 \neq y_2 2 \end{cases}$$

$$H = \begin{cases} max(0, \alpha_2^{old} + \alpha_1^{old}), y_1 = y_2 \\ max(0, \alpha_2^{old} - \alpha_1^{old} + C), y_1 \neq y_2 2 \end{cases}$$

2.4.2 变量的选择方法

第一个变量选取违反KKT条件最严重的点；

第二个变量选择能有足够大变化的点($max(|E_1 - E_2|)$)，有时此方法不能使目标函数下降足够，遍历数据集，若仍不够，重新选择第一个变量。

遍历顺序一般采取先支持向量再其他向量额顺序。

每次更新完两个参数后，还需更新b与E(存为列表减小重复计算)