开课吧数据竞赛第三课-钟老师-20191103

笔记本： 开课吧-小钟讲课
创建时间： 2019/10/28 星期一 22:17 **更新时间：** 2019/11/3 星期日 1:09
作者： 你看起来好像很好吃n_n
URL： https://www.cnblogs.com/liweiwei1419/p/9870034.html

开课吧-数据竞赛及相关问题从小工到专家

时间：2019-11-03

1. 二分类的评价指标

大家都习惯了天气预报带来的便捷服务，但是否知道目前天气预报的准确率多高？

气象预报包括航空、气候 、空气质量、水资源、火险天气、海洋、数字服务、观测等方面的内容。

1.1准确率

准确率是一个非常好理解的评价指标:

$$准确率 = \frac{预测正确的分类的个数}{模型输出需要预测的所有个数}$$

但是准确率在数据类别不平衡的时候单纯使用这个评价指标是不能非常客观的评论算法的优劣的。

比如我们需要对10000个文件夹进行病毒检测，可能病毒就潜伏在几个文件夹里面。这里假设有病毒潜伏在5个文件夹中。
那么我们就设计了一个“不太好的算法”，预测这10000个文件夹都没有病毒，那么根据准确的定义。只有5个文件夹预测错了，其余9995个都预测准确，按照准确率的定义：

$$准确率=\frac{预测正确的分类的个数}{模型输出需要预测的所有个数}=\frac{9995}{10000}=99.95\%$$

然后假如我们认为这样的准确率很高，非常有效的话，将会对电脑是致命的，所以我们就需要考虑其他的评价指标。

1.2 混淆矩阵

首先定义：
TN：算法预测为负例（0N），实际上也是负例（0N）的个数，模型预测对了（True）
FP：算法预测为正例（1P），实际上是负例（0N）的个数，模型预测错了（False）
FN：算法预测为负例（0N），实际上是正例（1P）的个数，模型预测错了（False）
TP：算法预测为正例（1P），实际上也是正例（1P）的个数，模型预测对了（True）
通过TP,FP,FN,TP组成混淆矩阵

	预测为0	预测为1
真实为0	TN	FP
真实为1	FN	TP

精准率（precision）的定义：

$$precision=\frac{TP}{TP+FP}$$

精准率：预测值为正（1）中预测正确的个数
召回率（recall）的定义：

$$recall=\frac{TP}{TP+FN}$$

召回率：真实值为正（1）中的预测正确的个数。
股票我们希望的是精准率
电脑病毒我们希望的是召回率，上面的例子召回率为0.
兼顾召回率和精准率：

1.3 F1 score

$$\frac{1}{F1}=\frac{1}{2}(\frac{1}{precision}+\frac{1}{recall})$$

通过上式推导得到：

$$F1=\frac{2\cdot precision\cdot recall}{precision+recall}$$

1.4 AUC

假正率（False positive rate， FPR），预测为正但实际为负的样本占所有负例样本的比例：

$$FPT=\frac{FP}{TN+FP}$$

真正率（True positive rate， TPR），真实值为正（1）中的预测正确的个数：

$$TPR=\frac{TP}{TP+FN}$$

假正率为横坐标，真正率为纵坐标，组成的roc曲线下的面积，就是AUC。

2. SVM（主要配合-李航统计学习方法（第一版）和第二版，和《西瓜书》）

支持向量：支持或支撑平面上把两类类别划分开来的超平面的向量点。
“机”：这里的“机（machine，机器）”便是一个算法。
目的：找到一个超平面，使得它能够尽可能多的将两类数据点正确的分开，同时使分开的两类数据点距离分类面最远。
支持向量机（support vector machines，SVM）是一种二类分类模型。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。
支持向量机的学习算法是求解凸二次规划的最优化
算法。
SVM包括：

- 线性可分支持向量机（linear support vector machine in linearly separable case）。当训练数据线性可分时，通过硬间隔最大化（hardmargin maximization），学习一个线性的分类器，即线性可分支持向量机；
- 线性支持向量机（linear support vector machine）。当训练数据近似线性可分时，通过软间隔最大化（soft marginmaximization），也学习一个线性的分类器，即线性支持向量机，又称为软间隔支持向量机；

- 非线性支持向量机（non-linear support vector machine）当训练数据线性不可分时，通过使用核技巧（kernel trick）及软间隔最大化，学习非线性支持向量机。
- 当训练数据是线性可分的，能够找到一个平面可以完全的的将数据分类正确，此时训练数据到超平面的距离就叫做硬间隔；
- 如果训练数据近似线性可分，同样可以找到一个超平面能够将绝大部分的数据分类正确，但是还是有少数的样本分类错误，如果能容忍这种错误的存在（且设置了容忍系数），此时训练数据到超平面的距离就是软间隔。

2.1 线性可分支持向量机

每个样本表示：

$$(x_1, y_1), (x_2, y_2), \cdots (x_n, y_n)$$

$$x_i \in \Re^d, \quad y_i \in \{-1, 1\}$$

$$y_i = 1 \text{ 表示 } \quad x_i \in \omega_1; \quad y_i = -1 \text{ 表示 } \quad x_i \in \omega_2$$

$$w * x + b = 0$$

一个线性分类的边界面形式为：
如果xa和xb 是 两个分界面上的点：

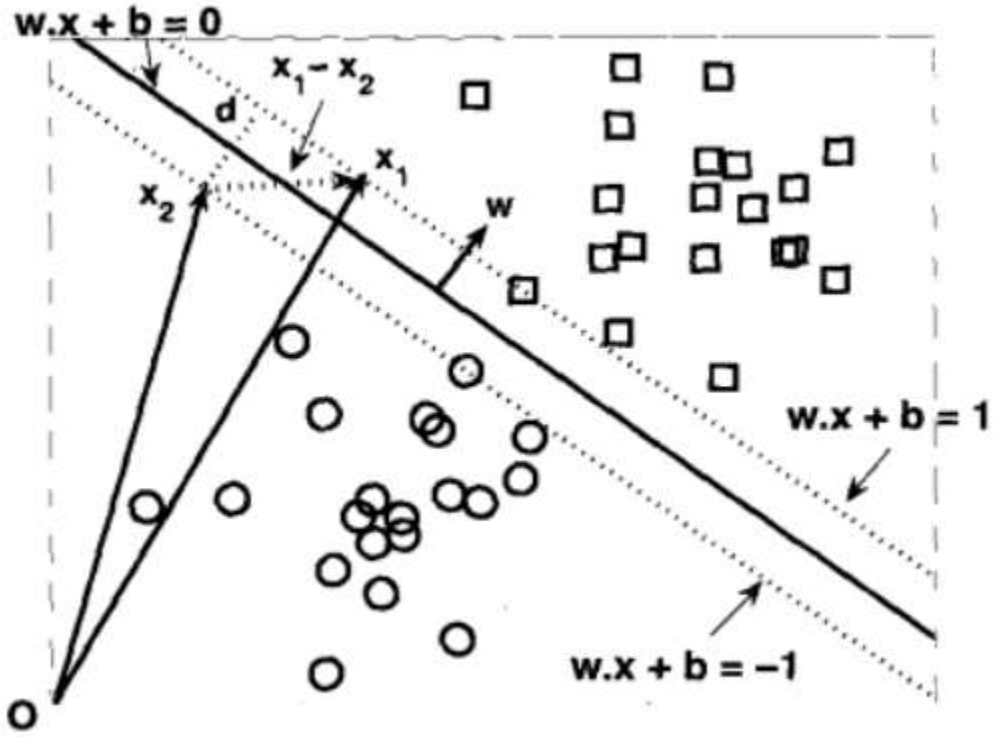
则：$w * x_a + b = 0$

$$w * x_b + b = 0$$

两个相减：

$$w * (x_a - x_b) + b = 0$$

如图所示：



标记：

方块那条线为：+1

圆形那条线为: -1

则有：

$$y=\begin{cases}1如果w*x+b>0\\-1如果w*x+b<0\end{cases}$$

则有两条边界线：

$$w*x+b=1$$

$$w*x+b=-1$$

假如a,b分别属于上面两个式子的点，则有：

$$w*x_a+b=1$$

$$w*x_b+b=-1$$

两式相减：

$$w*(x_a-x_b)=2$$

$$\|w\|\times d=2$$

最终：

$$d = \frac{2}{\| w \|}$$

$$d = \frac{2}{\| w \|}$$

希望求

最大则等价于：

$$\min_{w} \frac{\| w \|^2}{2}$$

其中：

$$st. y_i (w * x_i + b) \geq 1.$$

解得最优解：

$$w^*, b^*$$

由此得超平面：

$$w^* \bullet x + b^* = 0$$

分类决策函数：

$$f(x) = sign(w^* \cdot x + b^*)$$

学习的对偶性：

$$L(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{m} \alpha_i y_i (w x_i + b) + \sum_{i=1}^{m} \alpha_i$$

将拉格朗日函数$L(w, b, \alpha)$分别对w,b求偏导数并令其等于0。

$$\nabla_w L(w, b, \alpha) = w - \sum_{i=1}^{m} \alpha_i y_i x_i = 0$$

$$\nabla_b L(w, b, \alpha) = -\sum_{i=1}^{m} \alpha_i y_i = 0$$

得

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

$$\sum_{i=1}^{m} \alpha_i y_i = 0$$

将上面两个式代入拉格朗日函数：

$$L(w,b,\alpha) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j (x_i \cdot x_j) - \sum_{i=1}^{N}\alpha_i y_i \left(\left(\sum_{j=1}^{N}\alpha_j y_j x_j\right)\cdot x_i + b\right) + \sum_{i=1}^{N}\alpha_i$$

$$L(w,b,\alpha) = -\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j x_i\ x_j + \sum_{i=1}^{m}\alpha_i$$

$$\min L(w,b,\alpha) = -\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j x_i^T x_j + \sum_{i=1}^{m}\alpha_i$$

对α的极大。
则有：

$$\max_{\alpha} -\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j x_i\ x_j + \sum_{i=1}^{m}\alpha_i$$

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

$$\alpha_i \geq 0, i = 1, 2, \ldots, m$$

转化为极小：

$$\min_{\alpha} \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j x_i\ x_j - \sum_{i=1}^{m}\alpha_i$$

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

$$\alpha_i \geq 0, i = 1, 2, \ldots, m$$

定理：

上式子最优解：

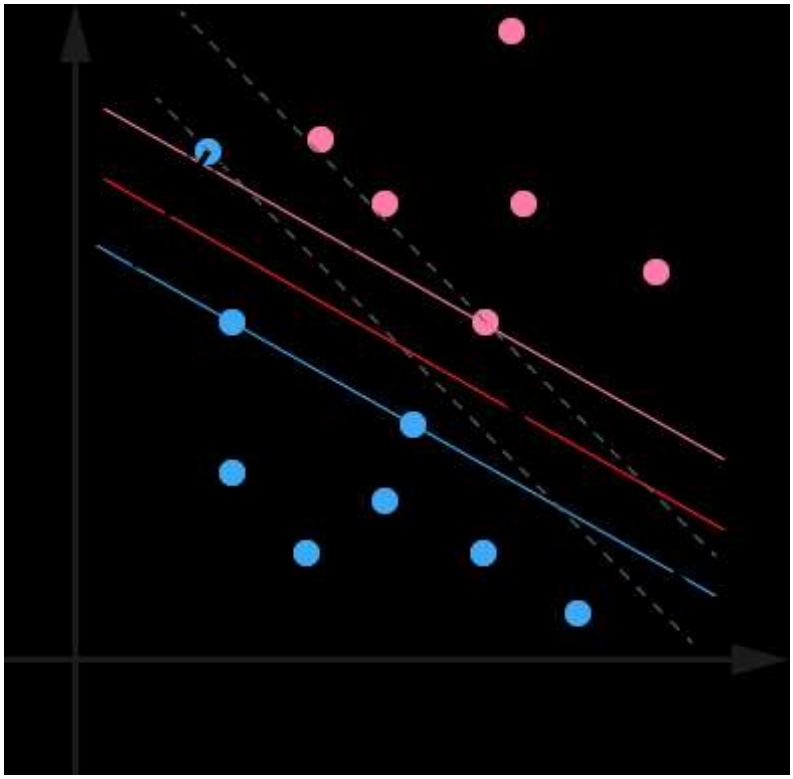
$$w^* = \sum_{i=1}^{m} \alpha_i^* y_i x_i$$

$$b^* = y_j - \sum_{i=1}^{m} \alpha_i^* y_i x_i \; x_j$$

作业：
理解第一版《统计学习方法》:例7.2（107页）

2.2 线性支持向量机与软间隔最大化

线性可分支持向量机的学习方法对线性不可分的训练数据是不适用的。线性不可分意味着某些样本点(xi,yi)(xi,yi)不能满足函数间隔大于等于1的条件，那么可以引入一个松弛变量ξi≥0,



约束条件为：

$$y_i(wx_i + b) \geq 1 - \xi_i$$

加入 $\frac{1}{2}|w||^2$ 和误分类的个数尽可能小，加入调和系数C。这样目标函数为：

$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i$$

$$s.t. \quad y_i(wx_i + b) \geq 1 - \xi_i, i = 1, \ldots, m$$

$$\xi_i \geq 0, i = 1, \ldots, m$$

转化为拉格朗日函数为：

$$L(w, b, \xi, \alpha, \mu) = \frac{1}{2}|w|^2 + C\sum_{i=1}^{m}\xi_i - \sum_{i}^{m}\alpha_i(y_i(wx_i + b) - 1 + \xi_i) - \sum_{i=1}^{m}\mu_i\xi_i$$

分别对w、b、ξw、b、ξ求偏导
对偶问题为：

$$\min_{\alpha} \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j x_i\ x_j - \sum_{i=1}^{m}\alpha_i$$

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

$$0 \leq \alpha_i \leq C, i = 1, 2, \ldots, m$$

$$w^* = \sum_{i=1}^{m}\alpha_i^* y_i x_i$$

$$b^* = y_j - \sum_{i=1}^{m}\alpha_i^* y_i x_i\ x_j$$

同样得到最优解：

，其中：$0 \leq \alpha_i \leq C,$

2.3非线性支持向量机与核函数

非线性分类问题，显然无法用一个线性分离超平面来把不同的类别的数据点分开：

1. 使用一个变换z=ϕ(x)（核函数）将非线性特征空间映射到线性特征空间z；
2. 在新的特征空间z中使用线性分类SVM学习分类从而训练数据获得模型。

2.4 SMO算法

smo需要解决的对偶问题是：

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

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

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

其中，目标函数的变量为$(\alpha_1, \alpha_2, ..., \alpha_N)$，每一个实例$(x_i, y_i)$对应一个$\alpha_i$。

假设α1,α2为两个变量，固定α3,α4,...,αN。

SMO基本思路：如果所有变量的解都满足此最优化问题的KKT条件，那么这个最优化问题的解就得到了。因此KKT条件是该最优化问题的充分必要条件。SMO算法的目标是求解满足KKT条件的(α1,α2,...,αN)。方法是选择两个变量，固定其他变量，针对这两个变量构建一个二次规划的子问题。这个二次规划子问题关于这两个变量的解更接近原始二次规划问题的解，因为这会使得原始二次规划问题的目标函数值变得更小。二次规划子问题有两个变量，一个是主动变量，其为违反KKT条件最严重的那一个，另一个被动变量，由主动变量依据约束条件自动确定。如此，SMO算法将原问题不断分解为子问题并对子问题求解，进而达到求解原问题的目的。

SMO优化子问题：

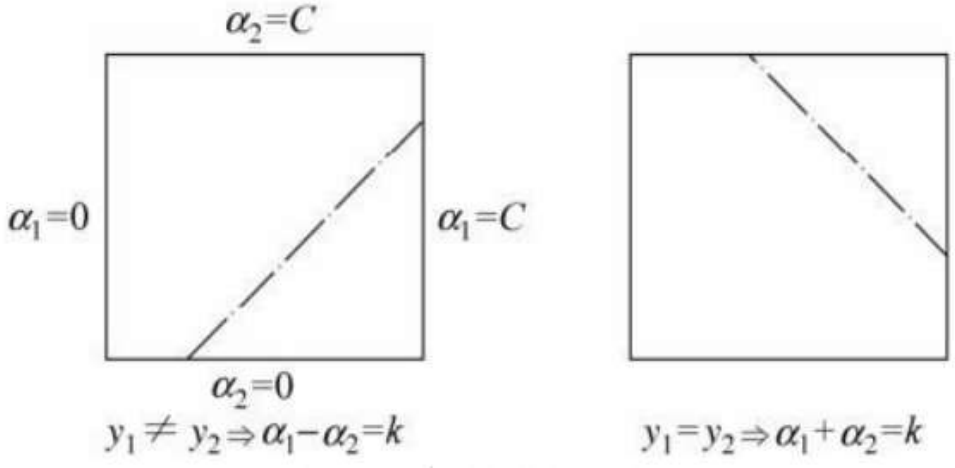
$$\min_{\alpha_1,\alpha_2} \quad W(\alpha_1,\alpha_2)=\frac{1}{2}K_{11}\alpha_1^2+\frac{1}{2}K_{22}\alpha_2^2+y_1y_2K_{12}\alpha_1\alpha_2-(\alpha_1+\alpha_2)+y_1\alpha_1\sum_{i=3}^{N}y_i\alpha_iK_{i1}+y_2\alpha_2\sum_{i=3}^{N}y_i\alpha_iK_{i2}$$

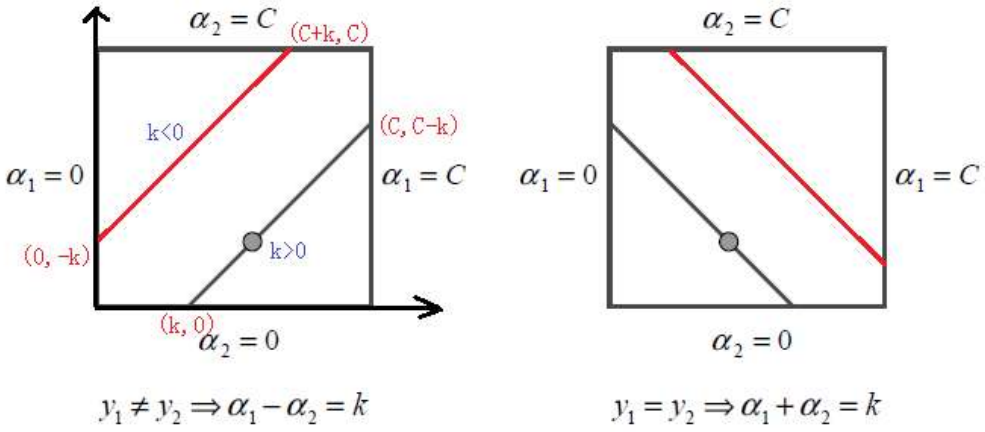
$$s.t. \quad \alpha_1y_1+\alpha_2y_2=-\sum_{i=3}^{N}\alpha_iy_i=\zeta$$

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

其中，Kij=K(xi,xj),i,j=1,2,...,N，ζ为常数。

两个变量（α1，α2）的约束条件：





α 1,α 2都在[0,C]范围内，并且α1,α2满足的直线平行于盒子范围【0，C】X【0，C】的对角线。

假设问题 (7.101)~(7.103) 的初始可行解为 $\alpha_1^{\text{old}}, \alpha_2^{\text{old}}$，最优解为 $\alpha_1^{\text{new}}, \alpha_2^{\text{new}}$，并且假设在沿着约束方向未经剪辑时 α_2 的最优解为 $\alpha_2^{\text{new,unc}}$。

由于 α_2^{new} 需满足不等式约束 (7.103)，所以最优值 α_2^{new} 的取值范围必须满足条件

$$L \leqslant \alpha_2^{\text{new}} \leqslant H$$

其中，L与 H是 α_2^{new} 所在的对角线段端点的界。如果 $y_1 \neq y_2$（如图 7.8(a) 所示），则

$$L = \max(0, \alpha_2^{\text{old}} - \alpha_1^{\text{old}}), \quad H = \min(C, C + \alpha_2^{\text{old}} - \alpha_1^{\text{old}})$$

如果 $y_1 = y_2$（如图 7.8(b) 所示），则

$$L = \max(0, \alpha_2^{\text{old}} + \alpha_1^{\text{old}} - C), \quad H = \min(C, \alpha_2^{\text{old}} + \alpha_1^{\text{old}})$$

下面，首先求沿着约束方向未经剪辑即未考虑不等式约束 (7.103) 时 α_2 的最优解 $\alpha_2^{\text{new,unc}}$；然后再求剪辑后 α_2 的解 α_2^{new}。我们用定理来叙述这个结果。为了叙述简单，记

$$g(x) = \sum_{i=1}^{N} \alpha_i y_i K(x_i, x) + b \tag{7.104}$$

令

$$E_i = g(x_i) - y_i = \left(\sum_{j=1}^{N} \alpha_j y_j K(x_j, x_i) + b \right) - y_i, \quad i = 1, 2 \tag{7.105}$$

当 $i = 1, 2$ 时，E_i 为函数 $g(x)$ 对输入 x_i 的预测值与真实输出 y_i 之差。

定理 7.6 最优化问题 (7.101)~(7.103) 沿着约束方向未经剪辑时的解是

$$\alpha_2^{\text{new,unc}} = \alpha_2^{\text{old}} + \frac{y_2(E_1 - E_2)}{\eta} \tag{7.106}$$

其中，

$$\eta = K_{11} + K_{22} - 2K_{12} = \|\Phi(x_1) - \Phi(x_2)\|^2 \tag{7.107}$$

$\Phi(x)$ 是输入空间到特征空间的映射，E_i，$i = 1, 2$，由式 (7.105) 给出。

经剪辑后 α_2 的解是

$$\alpha_2^{\text{new}} = \begin{cases} H, & \alpha_2^{\text{new,unc}} > H \\ \alpha_2^{\text{new,unc}}, & L \leqslant \alpha_2^{\text{new,unc}} \leqslant H \\ L, & \alpha_2^{\text{new,unc}} < L \end{cases} \tag{7.108}$$

由 α_2^{new} 求得 α_1^{new} 是

$$\alpha_1^{\text{new}} = \alpha_1^{\text{old}} + y_1 y_2 (\alpha_2^{\text{old}} - \alpha_2^{\text{new}}) \tag{7.109}$$

3. 数据竞赛

1. 「二分类算法」提供银行精准营销解决方案-新加特征工程
2. 员工离职预测训练赛-LR
3. kaggle 欺诈信用卡预测