A INTRO TO SVM

支持向量机（II）

龙飞宇
2019.7
DSSC

04 非线性与核技巧

非线性SVM的主要思想

用线性分类方法求解非线性分类问题，分两步

- 首先使用一个变换将原空间的数据映射到新空间;
- 然后在新空间里用线性分类学习方法从训练数据中学习分类模型。

一.升维操作

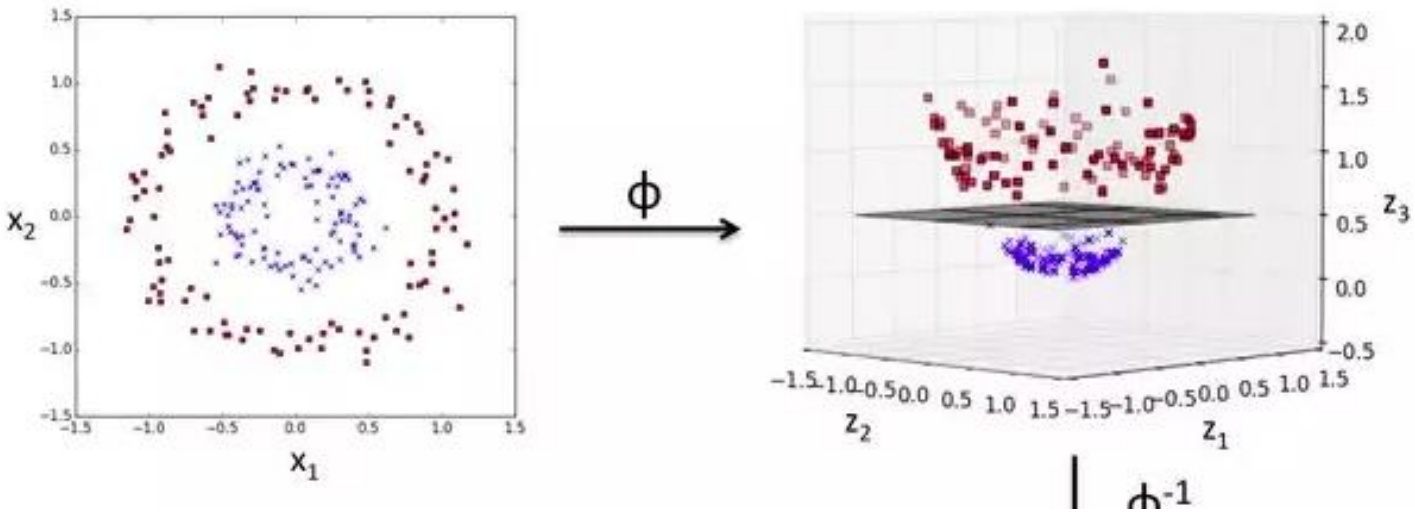
1. 动机

数据集线性不可分，软间隔的支持向量机误分类点过多，无法取得很好效果的情况下，考虑提升维度来将非线性分类转化成线性分类。

2. 可行性

Cover定理：“假设空间不是稠密分布的，将复杂的模式分类问题非线性地投射到高维空间将比投射到低维空间更可能将其变为线性可分的。”

3. 操作过程



上图中特征空间只有两维。我们设定横坐标为x_1,纵坐标为x_2，那我们可以得到这条分隔线的函数为:

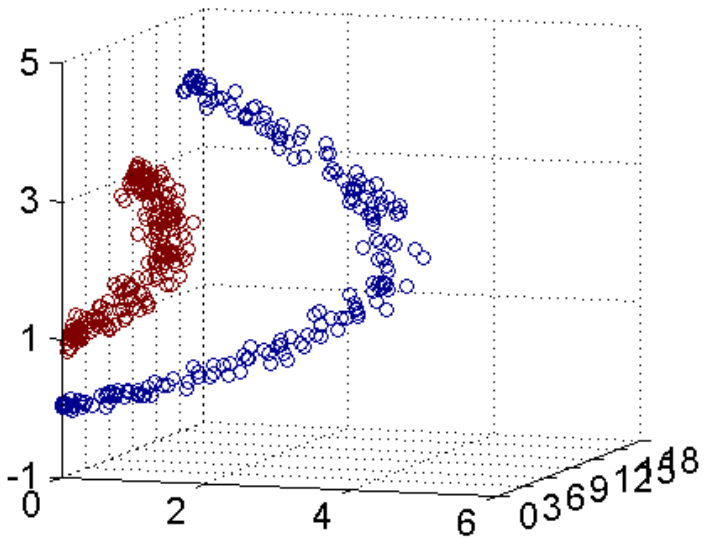
$$f(x_1,x_2)=\alpha_1 x_1+\alpha_2 x_2+\alpha_3 x_1^2+\alpha_4 x_2^2+\alpha_5 x_1 x_2+\alpha_6$$

这里的a为系数。那我们现在可以令 $Z=[x_1,x_2,x_1^2,x_2^2,x_1x_2]$ 并且由Z构建新的坐标系的坐标轴.那我们得到一个五维的映射空间。在这新的五维空间中，我们可以看出这个椭圆形分隔线函数为:

$$f(z_1,z_2,z_3,z_4,z_5)=a_1z_1+a_2z_2+a_3z_3+a_4z_4+a_5z_5+a_6$$

这是一个分割超平面的方程，是一个线性方程。那么我们便通过这一次由x到z的映射，成功使用升维的办法将非线性分类问题转化成线性分类问题。

升维其实是一种解决办法，但是我们很容易发现其致命缺点——维度灾难。



二.核技巧（Kernel Trick)

我们将定义成核函数$K(x,z)=\Phi(x)\cdot\Phi(z)$，定义为$\Phi(x)$映射函数。

核技巧的思想是：在学习和预测中只使用$K(x,z)$，而并不使用或者显式计算$\Phi(x)\cdot\Phi(z)$。因为通常计算$K(x,z)$比较容易。

下面用简单例子来说明核函数和映射函数的关系：

- 假设输入空间是R^2，核函数是$K(x,z)=(x\cdot z)^2$，试找出其相关的特征空间H和映射 $\phi(x):\mathbf{R}^2\to\mathcal{H}$
- 解：

  取特征空间 $\mathcal{H}=\mathbf{R}^3$，记 $x=(x^{(1)},x^{(2)})^T$，$z=(z^{(1)},z^{(2)})^T$

$$(x\cdot z)^2=(x^{(1)}z^{(1)}+x^{(2)}z^{(2)})^2=(x^{(1)}z^{(1)})^2+2x^{(1)}z^{(1)}x^{(2)}z^{(2)}+(x^{(2)}z^{(2)})^2$$

- 可以取：$\phi(x)=((x^{(1)})^2,\sqrt{2}x^{(1)}x^{(2)},(x^{(2)})^2)^T$

$$\phi(x)=\frac{1}{\sqrt{2}}((x^{(1)})^2-(x^{(2)})^2,2x^{(1)}x^{(2)},(x^{(1)})^2+(x^{(2)})^2)^T$$

$$\phi(x)=((x^{(1)})^2,x^{(1)}x^{(2)},x^{(1)}x^{(2)},(x^{(2)})^2)^T$$

- 容易验证：$\phi(x)\cdot\phi(z)=(x\cdot z)^2=K(x,z)$ 都满足条件。

非线性支持向量机算法

输入：线性不可分训练数据集 $T=\{(x_1,y_1),(x_2,y_2),\cdots,(x_N,y_N)\}$

$x_i \in \mathcal{X}=\mathbf{R}^n \qquad y_i \in \mathcal{Y}=\{-1,+1\}, \quad i=1,2,\cdots,N$

输出：分类决策函数

1、选取适当的核函数和参数C，构造最优化问题：

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

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

$$0\leqslant\alpha_i\leqslant C, \quad i=1,2,\cdots,N$$

5

求得最优解：$\alpha^*=(\alpha_1^*,\alpha_2^*,\cdots,\alpha_N^*)^T$

非线性支持向量机算法

2、并选择α*，适合条件 $0<\alpha_j^*<C$ ，计算

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

3、构造决策函数

$$f(x) = \text{sign}\left(\sum_{i=1}^{N} \alpha_i^* y_i K(x \cdot x_i) + b^* \right)$$

提问：

采用什么样的核函数才能使得式5有解？（凸二次优化问题解存在）

三.正定核

假设$K(x,z)$是定义在$\mathcal{X}\times\mathcal{X}$上的对称函数，并且对任意的$x_i\in\mathcal{X}$,i=1,2,…，m,关于$K(x,z)$的Gram矩阵是半正定的，**则称$K(x,z)$为正定核**。可以根据函数$K(x,z)$构成一个希尔波特空间。

其步骤是(1)首先定义映射，并构成向量空间S；(2)在S上定义内积构成内积空间；(3)最后将S完备化构成希尔伯特空间。

正定核的充要条件

$K(x,z)$为核函数<=>对任意$K(x,z)$对应的Gram矩阵是半正定的。

这一定义在构造核函数时很有用。但对于一个具体函数$K(x,z)$来说，检验它是否为核函数并不容易。因为要求对任意有限输入集验证K对应的Gram矩阵是否为半正定的。在实际问题中往往应用**已有的核函数**。

$$Gram\quad K=\begin{bmatrix} K(x_1,x_1) & \cdots & K(x_1,x_j) & \cdots & K(x_1,x_m) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ K(x_i,x_1) & \cdots & K(x_i,x_j) & \cdots & K(x_i,x_m) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ K(x_m,x_1) & \cdots & K(x_m,x_j) & \cdots & K(x_m,x_m) \end{bmatrix}$$

常用核函数

名称	表达式	参数
线性核	$\kappa(x_i, x_j) = x_i^T x_j$	无
多项式核	$\kappa(x_i, x_j) = (x_i^T x_j)^d$	$d \geq 1$为多项式次数
高斯核	$\kappa(x_i, x_j) = exp(-\frac{\|\|x_i - x_j\|\|^2}{2\sigma^2})$	$\sigma > 0$为高斯核带宽
拉普拉斯核	$\kappa(x_i, x_j) = exp(-\frac{\|\|x_i - x_j\|\|^2}{\sigma})$	$\sigma > 0$
$Sigmoid$核	$\kappa(x_i, x_j) = tanh(\beta x_i^T x_j + \theta)$	$tanh$为双曲正切函数，$\beta > 0, \theta < 0$

如果上表不够用，还可以采取组合的方式

- 若k_1和 k_2为核函数，则对于任意正数 $\gamma_1\gamma_2$,线性组合 $\gamma_1 k_1 + \gamma_2 k_2$也是核函数。
- 若k_1和 k_2为核函数，则函数的直积$k_1 \otimes k_2 = k_1 k_2$也是核函数。
- 若k_1和 k_2为核函数，则对于任意函数$g(\mathrm{x}), k(x_1, x_2) = g(x_1)k_1(x_1, x_2)g(x_2)$也是核函数。

选择与调参

对于核函数如何选择的问题，吴恩达教授是这么说的：
1.如果Feature的数量很大，跟样本数量差不多，这时候选用LR或者是Linear Kernel的SVM
2.如果Feature的数量比较小，样本数量不算大也不算小，选用SVM+Gaussian Kernel(RBF)
3.如果Feature的数量比较小，而样本数量很多，需要手工添加一些feature变成第一种情况

速度

主要由模型参数多少决定，线性核肯定是最快的

普遍做法

试所有的核函数，看哪个效果好。就说自己的数据是适合哪种核函数的数据分布特性的。

05 SMO算法

1. 动机

高效实现支持向量机学习，优化凸二次规划的对偶问题中的拉普拉斯参数。

2. 序列最小最优化（SMO）算法

- 是一种**启发式算法**，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，那么得到解；
  - 否则，选择两个变量，固定其它变量，针对这两个变量构建一个二次规划问题，称为子问题，可通过解析方法求解，提高计算速度。
  - 子问题的两个变量：一个是违反KKT条件最严重的那个，另一个由约束条件自动确定。

3. 操作过程

$$\sum_{i=1}^{N} a_i y_i = 0$$

SMO算法包括两个部分：

- **求解两个变量二次规划的解析方法**
- **选择变量的启发式方法**

两个变量二次规划的求解过程

假设我们选取的两个需要优化的参数为α_1，α_2, 剩下的α则固定，作为常数处理。则SMO的5的子问题:

$$\min_{\alpha_1,\alpha_2} \quad W(\alpha_1,\alpha_2)=\frac{1}{2}K_{11}\alpha_1^2+\frac{1}{2}K_{22}\alpha_2^2+y_1y_2K_{12}\alpha_1\alpha_2$$
$$-(\alpha_1+\alpha_2)+y_1\alpha_1\sum_{i=3}^{N}y_i\alpha_iK_{i1}+y_2\alpha_2\sum_{i=3}^{N}y_i\alpha_iK_{i2} \qquad (6)$$
$$\text{s.t.} \quad \alpha_1y_1+\alpha_2y_2=-\sum_{i=3}^{N}y_i\alpha_i=\varsigma \qquad 0\le\alpha_i\le C$$

1.获得没有修剪的原始解

用α_2来表示α_1, 带入式6得到：W(α_2)，然后求极值，并用更新前的α_2^{old}来表示α_2^{new}，得:

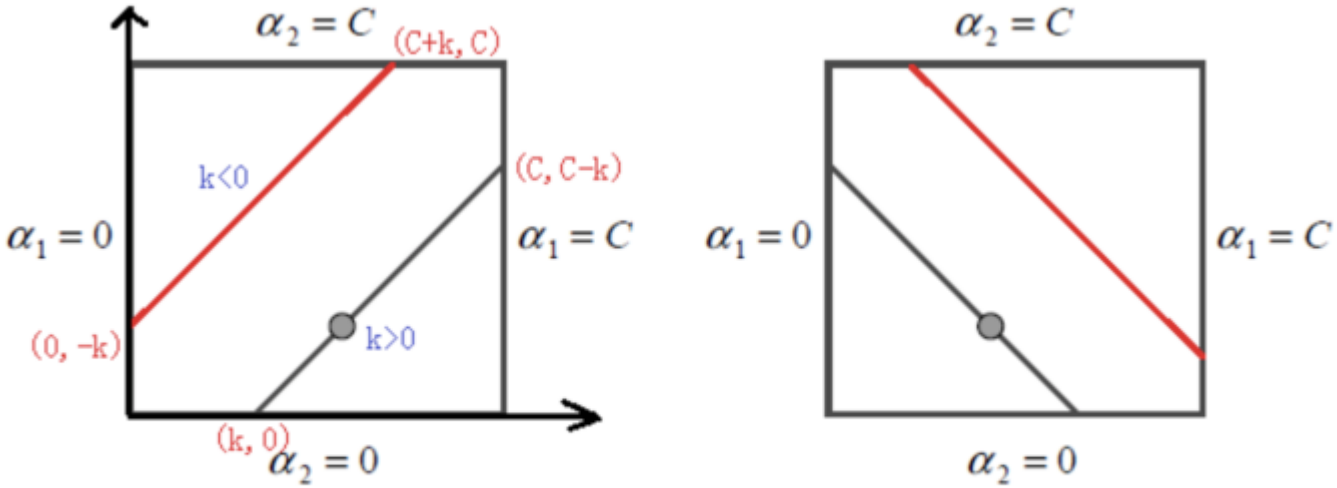
$$a_2^{new,unc}=a_2^{old}+\frac{y_2(E_1-E_2)}{\eta}$$

其中E_i为为SVM预测值与真实值的误差:$E_i=f(x_i)-y_i$,$\mu=K_{11}+K_{22}-2K_{12}$

2.对原始解进行修剪

在SVM中α_i是有约束的即：$\alpha_1 y_1 + \alpha_2 y_2 = -\sum_{i=3}^{N} \alpha_i y_i = \zeta \qquad 0 \le \alpha_i \le C$

在二维平面中我们可以看到这是个限制在方形区域中的直线（见下图）。



$y_1 \neq y_2 \Rightarrow \alpha_1 - \alpha_2 = k$　　$y_1 = y_2 \Rightarrow \alpha_1 + \alpha_2 = k$

当$y_1 \neq y_2$时(左图)，线性限制条件可以写成:$a_1 - a_2 = k$，根据k的正负可以得到不同的上下界，表示成:

下界: $L = \max(0, \alpha_2^{old} - \alpha_1^{old})$

上界: $H = \min(C, C + \alpha_2^{old} - \alpha_1^{old})$

05 SMO算法

当$y_1 = y_2$时(右图)，线性限制条件可以写成:$a_1 + a_2 = k$，上下界表示成:

下界: $L = \max(0, \alpha_1^{old} + \alpha_2^{old} - C)$

上界: $H = \min(C, \alpha_2^{old} + \alpha_1^{old})$

根据得到的上下界，我们可以得到修剪后的α_2^{new}:

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

得到α_2^{new}后可$\alpha_1^{\text{new}}\mathrm{y}_1 + \alpha_2^{\text{new}}\mathrm{y}_2 = \alpha_1^{old}\mathrm{y}_1 + \alpha_2^{old}\mathrm{y}_2$,得到$\alpha_1^{\text{new}}$:

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

于是就得到了二次最优化问题的解。

变量的选择过程

1.第一个变量的选择：外循环

违反KKT最严重的样本点，

检验样本点是否满足KKT条件：

先检查 ➡

$$\alpha_i = 0 \Leftrightarrow y_i g(x_i) \geqslant 1$$

$$0 < \alpha_i < C \Leftrightarrow y_i g(x_i) = 1$$

$$\alpha_i = C \Leftrightarrow y_i g(x_i) \leqslant 1$$

$$g(x_i) = \sum_{j=1}^{N} \alpha_j y_j K(x_i, x_j) + b$$

2.第二个变量的检查: 内循环

选择的标准是希望能使目标函数有足够大的变化

即对应$|E1 - E2|$最大，比如E1，E2 的符号相反，差异最大

如果内循环通过上述方法找到的点不能使目标函数有足够的下降

则：遍历间隔边界上的样本点，测试目标函数下降

如果下降不大，则遍历所有样本点

如果依然下降不大，则丢弃外循环点，重新选择

变量更新后重新计算阈值b

当α_1^{new}不在边界，即$0 < \alpha_1^{\text{new}} < C$, 根据KKT条件可知相应的数据点为支持向量，满足$y_1(w^T + b) = 1$, 两边同时乘上y_1得到 , 进而得到b_1^{new}的值:

$$b_1^{new} = y_1 - \sum_{i=3}^{N} \alpha_i y_i K_{i,1} - \alpha_1^{new} y_1 K_{1,1} - \alpha_2^{new} y_2 K_{2,1}$$

其中上式的前两项可以写成:

$$y_1 - \sum_{i=3}^{N} \alpha_i y_i K_{i,1} = -E_1 + \alpha_1^{old} y_1 K_{1,1} + \alpha_2^{old} y_2 K_{2,1} + b^{old}$$

当$0 < \alpha_2^{\text{new}} < C$, 可以得到的表达式(推导同上):

$$b_2^{new} = -E_2 - y_1 K_{1,2}(\alpha_1^{new} - \alpha_1^{old}) - y_2 K_{2,2}(\alpha_2^{new} - \alpha_2^{old}) + b^{old}$$

当b_1和b_2都有效的时候b_1^{new}与b_2^{new}是相等的 。
当两个乘子都在边界上，且$L \neq H$时， b_1, b_2之间的值就是和KKT条件一直的阈值。SMO选择他们的中点作为新的阈值:

$$b^{new} = \frac{b_1^{new} + b_2^{new}}{2}$$

序列最小最优化（SMO）算法总结

输入：训练数据集 $T=\{(x_1,y_1),(x_2,y_2),\cdots,(x_N,y_N)\}$

$x_i\in\mathcal{X}=\mathbf{R}^n$　$y_i\in\mathcal{Y}=\{-1,+1\}$，$i=1,2,\cdots,N$，精度$\varepsilon$

输出：近似解 α

（1）取初值$\alpha^{(0)}=0$，令$k=0$

（2）选取优化变量$\alpha_1^{(k)},\alpha_2^{(k)}$，解析求解两个变量的最优化问题

求得最优解$\alpha_1^{(k+1)},\alpha_2^{(k+1)}$，更新$\alpha$为$\alpha^{(k+1)}$；

（3）若在精度ε范围内满足停机条件

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

$$0\leqslant\alpha_i\leqslant C,\quad i=1,2,\cdots,N$$

$$y_i\cdot g(x_i)=\begin{cases}\geqslant 1, & \{x_i\mid\alpha_i=0\}\\ =1, & \{x_i\mid 0<\alpha_i<C\}\\ \leqslant 1, & \{x_i\mid\alpha_i=C\}\end{cases}$$

$$g(x_i)=\sum_{j=1}^{N}\alpha_j y_j K(x_j,x_i)+b$$

则转 (4)；否则令$k=k+1$，转 (2)；

（4）取$\hat{\alpha}=\alpha^{(k+1)}$

后续——SVM算法包

- Joachims **SVM light**：http://svmlight.joachims.org
- **LIBSVM**：http://www.csie.ntu.edu.tw/~cjlin/libsvm