法律声明

- 课程详情请咨询
  - 微信公众号：北风教育
  - 官方网址：*http://www.ibeifeng.com/*

人工智能之机器学习

SVM支持向量机

主讲人：Gerry

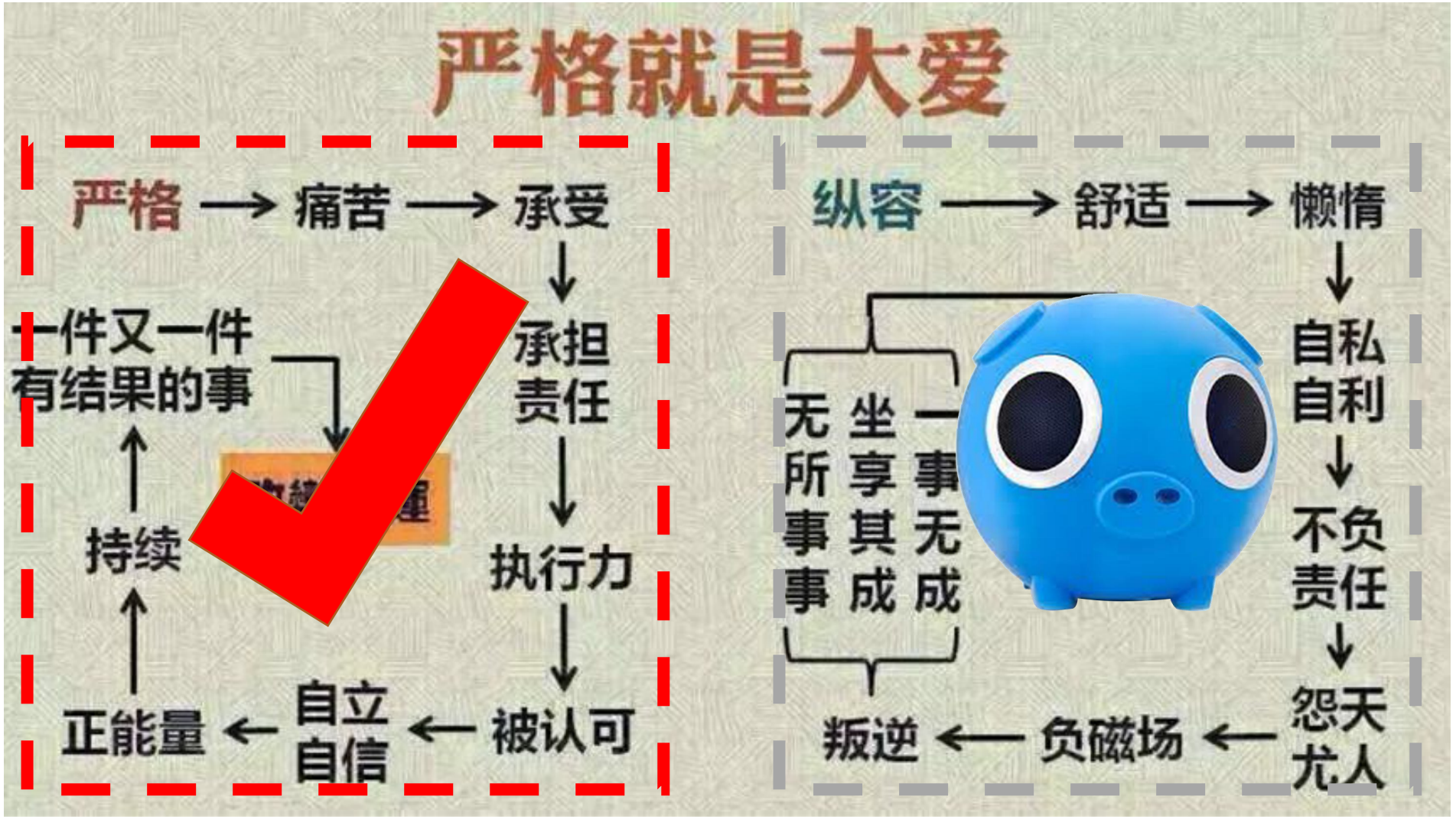
上海育创网络科技有限公司

北风网 IBEIFENG.COM

课程要求

- 课上课下“九字”真言
  - 认真听，善摘录，勤思考
  - **多温故，乐实践**，再发散
- 四不原则
  - 不懒散惰性，不迟到早退
  - 不请假旷课，不拖延作业
- 一点注意事项
  - 违反“四不原则”，不包就业和推荐就业

严格是大爱


严格就是大爱
严格
痛苦
承受
承担责任
执行力
被认可
自立自信
正能量
持续
一件又一件有结果的事
纵容
舒适
懒惰
自私自利
不负责任
怨天尤人
负磁场
叛逆
一事无成
坐享其成
无所事事

寄语

做别人不愿做的事，

做别人不敢做的事，

做别人做不到的事。

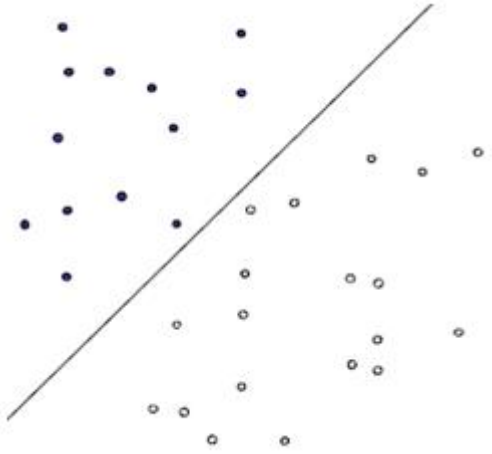
SVM

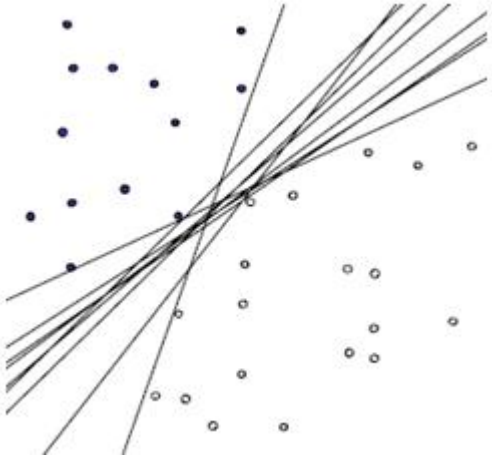
- 线性可分
- 线性不可分
- SMO
- 核函数

线性分类器

一、线性分类器：

首先给出一个非常非常简单的分类问题（线性可分），我们要用一条直线，将下图中黑色的点和白色的点分开，很显然，图上的这条直线就是我们要求的直线之一（可以有无数条这样的直线）





线性分类器

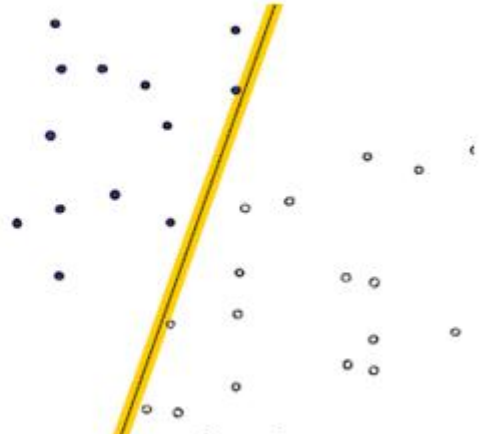
假如说，我们令黑色的点 = -1， 白色的点 = +1，直线f(x) = w.x + b

当向量x的维度=2的时候，f(x) 表示二维空间中的一条直线， 当x的维度=3的时候，f(x) 表示3维空间中的一个平面，当x的维度=n > 3的时候，表示n维空间中的超平面

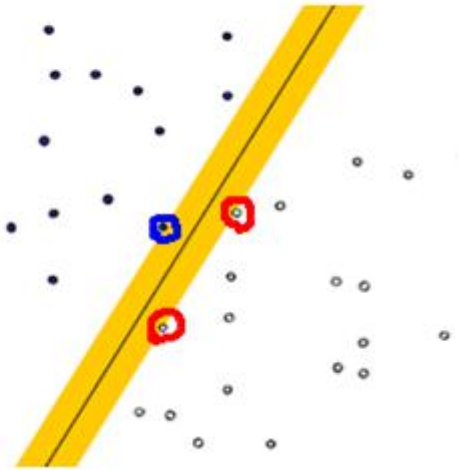
一个很直观的感受是，让这条直线到给定样本中最近的点最远，这句话读起来比较拗口，下面给出几个图，来说明一下：

线性分类器

第一种分法：



第二种分法：



这两种分法哪个更好呢？

支持向量

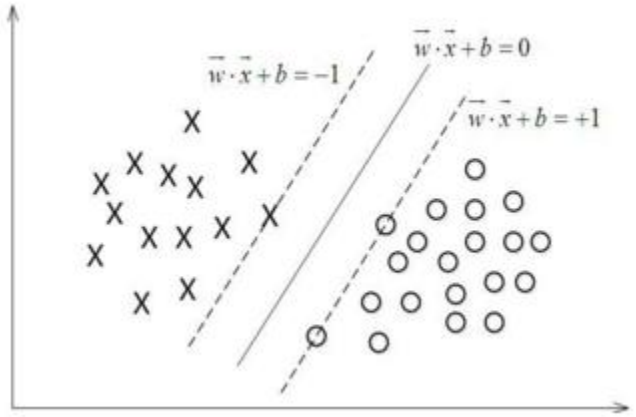
就像我们平时判断一个人是男还是女，就是很难出现分错的情况，这就是男、女两个类别之间的间隙非常的大导致的，让我们可以更准确的进行分类。在SVM中，称为Maximum Marginal，是SVM的一个理论基础之一。

选择使得间隙最大的函数作为分割平面是有很多道理的，比如说从概率的角度上来说，就是使得置信度最小的点置信度最大（听起来很拗口），从实践的角度来说，这样的效果非常好。

上图被红色和蓝色的线圈出来的点就是所谓的支持向量(support vector)。

线性可分支持向量机

- 针对线性可区分，求超平面推导



超平面的公式可以定义为 ： W * X + b = 0 W表示权重向量 W={w1,w2,w3,w4.....,wn},n为特征值的个数 ， X为训练实例， b表示偏移量

线性可分支持向量机

- 在这里假设二维特征向量X=（x1，x2）

做另外一个假设就是把b看作是另外一个weight，那么超平面就可以更新为：

$b+ w1 * x1 +w2 * x2 = 0$

所有超平面右上方的点满足：　　$b + w1 * x1 +w2 * x2 > 0$

所有超平面左下方的点满足：　　$b+ w1 * x1 +w2 * x2 < 0$

数学输入

- 给定一个特征空间上的训练数据集

  T={(x1,y1),(x2,y2),(x3,y3).....(xn,yn)}

  其中，$x1 \in R^n$，$yi \in \{+1,-1\}$,i=1,2,3,4.....N

  xi为第i个实例(若 n>1,xi为向量)

  yi为xi的类标记

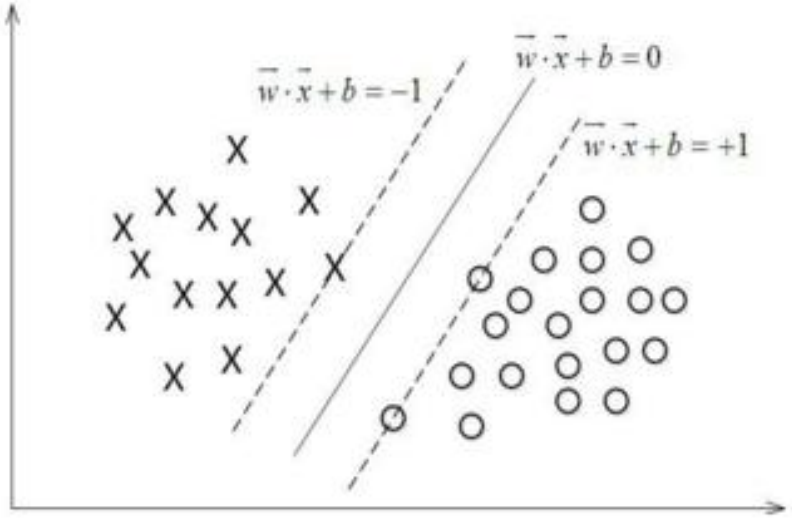
  当yi=+1时，称xi为正例

  当yi=-1时，称xi为负例

  (xi,yi)称为样本点

线性可分支持向量机

- 对平面中的点进行等比例缩小或者放大，总有一个比例，使得支持向量点到超平面的距离为1



目标函数

- 把w1和w2看成一个向量，x1,x2也是一个向量则有

$$y(x)=w^T\Phi(x)+b$$

$$\begin{cases} y(x_i)>0 \Leftrightarrow y_i=+1 \\ y(x_i)<0 \Leftrightarrow y_i=-1 \end{cases} \Rightarrow y_i \cdot y(x_i)>0$$

最大间隔分离超平面

- w,b等比例缩放，等价于点到超平面的距离缩放，从而

$$\frac{y_i \cdot y(x_i)}{\|w\|} = \frac{y_i \cdot \left(w^T \cdot \Phi(x_i) + b\right)}{\|w\|}$$

- 前面已经提到过，让这条直线到给定样本中最近的点最远,即目标函数是

$$\arg\max_{w,b} \left\{ \frac{1}{\|w\|} \min_i \left[y_i \cdot \left(w^T \cdot \Phi(x_i) + b\right) \right] \right\}$$

目标函数

- 因为这里的约束条件是支持向量点到超平面的距离为1

即

$$y_i \cdot \left(w^T \cdot \Phi(x_i) + b\right) \geq 1$$

- 到这里，我们前面的问题转化为了求

$$\arg\max_{w,b} \frac{1}{\|w\|}$$

$$s.t. \quad y_i\left(w^T \cdot \Phi(x_i) + b\right) \geq 1, \quad i = 1,2,\cdots,n$$

目标函数

- ||w||的意思是w的二范数，跟 $\sqrt{w.w}$ 表达式是一个意思，之前得到，求SVM的关键就是，最大化这个式子等价于最小化||w||, 另外由于||w||是一个单调函数，我们可以对其加入平方，和前面的系数。

$$\max \frac{1}{\|w\|} \rightarrow \min \frac{1}{2}\|w\|^2$$

$$s.t. \quad y_i\left(w^T \cdot \Phi(x_i)+b\right) \geq 1, \quad i=1,2,\cdots,n$$

拉格朗日乘子法

- 转化为对偶问题，并优化求解:

这个优化问题可以用拉格朗日乘子法去解，使用了KKT条件的理论，这里直接作出这个式子的拉格朗日目标函数：

$$L(w,b,a)=\frac{1}{2}\|w\|^2-\sum_{i=1}^{n}\alpha_i(y_i(w^Tx_i+b)-1)$$

求解这个式子的过程需要拉格朗日对偶性的相关知识

拉格朗日乘子法

- 原问题是极小极大问题

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

- 问题的对偶问题，是极大极小问题

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

拉格朗日函数

首先让L关于w，b最小化，分别令L关于w，b的偏导数为0，得到关于原问题的一个表达式

$$\frac{\partial L}{\partial w}=0\Rightarrow w=\sum_{i=1}^{n}\alpha_i y_i \Phi(x_n)$$

$$\frac{\partial L}{\partial b}=0\Rightarrow 0=\sum_{i=1}^{n}\alpha_i y_i$$

拉格朗日函数

- 推导过程

$$\mathcal{L}(\mathrm{w},\mathrm{b},\alpha)=\frac{1}{2}\|w\|^2-\sum_{i=1}\alpha_i\left[y^{(i)}\left(w^Tx^{(i)}+b\right)-1\right]$$

$$=\frac{1}{2}\mathrm{w}^T\mathrm{w}-\sum_{i=1}^{m}\alpha_iy^{(i)}w^Tx^{(i)}-\sum_{i=1}^{m}\alpha_iy^{(i)}b+\sum_{i=1}^{m}\alpha_i$$

$$=\frac{1}{2}\mathrm{w}^T\sum_{i=1}^{m}\alpha_iy^{(i)}x^{(i)}-\sum_{i=1}^{m}\alpha_iy^{(i)}w^Tx^{(i)}-\sum_{i=1}^{m}\alpha_iy^{(i)}b+\sum_{i=1}^{m}\alpha_i$$

$$=\frac{1}{2}\mathrm{w}^T\sum_{i=1}^{m}\alpha_iy^{(i)}x^{(i)}-w^T\sum_{i=1}^{m}\alpha_iy^{(i)}x^{(i)}-\sum_{i=1}^{m}\alpha_iy^{(i)}b+\sum_{i=1}^{m}\alpha_i$$

$$=-\frac{1}{2}\mathrm{w}^T\sum_{i=1}^{m}\alpha_iy^{(i)}x^{(i)}-\sum_{i=1}^{m}\alpha_iy^{(i)}b+\sum_{i=1}^{m}\alpha_i$$

$$=-\frac{1}{2}\mathrm{w}^T\sum_{i=1}^{m}\alpha_iy^{(i)}x^{(i)}-b\sum_{i=1}^{m}\alpha_iy^{(i)}+\sum_{i=1}^{m}\alpha_i$$

$$=-\frac{1}{2}\left(\sum_{i=1}^{m}\alpha_iy^{(i)}x^{(i)}\right)^T\sum_{i=1}^{m}\alpha_iy^{(i)}x^{(i)}-b\sum_{i=1}^{m}\alpha_iy^{(i)}+\sum_{i=1}^{m}\alpha_i$$

拉格朗日函数

- 结果加个负号得到

$$\min_{\alpha} \quad \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j\left(\Phi(x_i)\cdot\Phi(x_j)\right)-\sum_{i=1}^{n}\alpha_i$$

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

$$\alpha_i \geq 0, \quad i = 1,2,\ldots,n$$

线性可分支持向量机

- 假设有最优解 α^*
- 计算

$$w^* = \sum_{i=1}^{N} \alpha_i^* y_i \Phi(x_i)$$

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

- 求得分离超平面

$$w^* \Phi(x) + b^* = 0$$

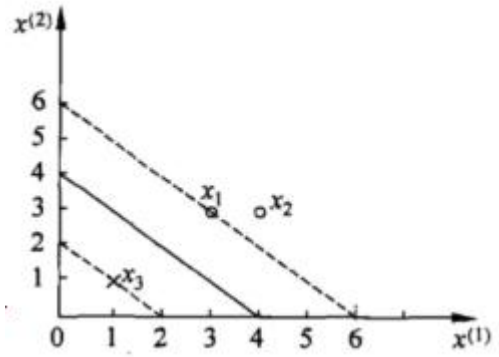
- 分类决策函数

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

例子

- 举例：

给定三个数据点：正例点x1=(3,3),x2=(4,3),负例点x3=(1,1),求线性可分支持向量机。



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

$$= \frac{1}{2}\left(18\alpha_1^2 + 25\alpha_2^2 + 2\alpha_3^2 + 42\alpha_1\alpha_2 - 12\alpha_1\alpha_3 - 14\alpha_2\alpha_3\right) - \alpha_1 - \alpha_2 - \alpha_3$$

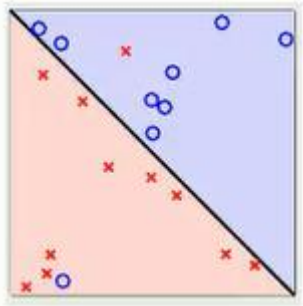
$$s.t. \ \ \alpha_1 + \alpha_2 - \alpha_3 = 0$$

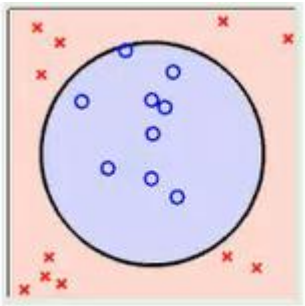
$$\alpha_i \geq 0, \quad i = 1,2,3$$

线性支持向量机

- 线性可区分（linear separable）和线性不可区分（linear inseparable）

上面是线性可区分的，很容易找到一个超平面将数据分割成两类，但如果不能直接划分出来呢？





线性支持向量机

- 线性支持向量机

若是存在离群点呢？

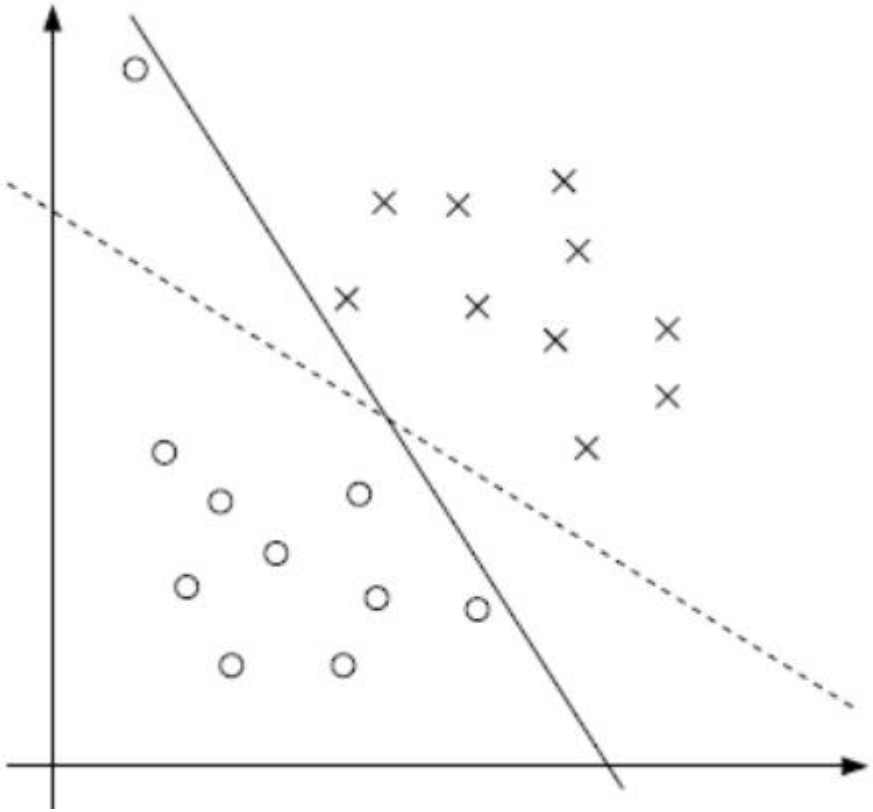
如果使用实线来划分，拟合的精度会高一些

但是模型的泛化能力会很差

即测试的时候出现误差过大，即过拟合

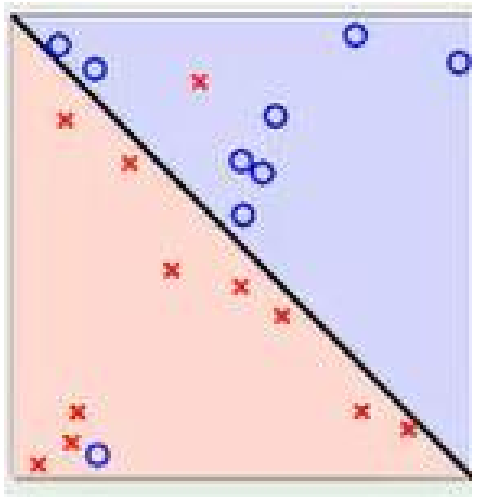
这时候我们就需要适当的允许一些离群的点可

边界，就是图中的虚线。



线性不可分

- 我们知道所有数都不那么干净，不能100%线性可分。我们可以通过引入松弛变量，来允许有些数据点可以处于分割面的错误的一侧。



$$y_i(w \cdot x_i + b) \geq 1 - \xi_i$$

- 需要加一个松弛因子 $\xi_i \geq 0$

目标函数

- 目标函数：

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

$$s.t. \quad y_i\left(w\cdot x_i + b\right) \geq 1-\xi_i, \quad i=1,2,\cdots,n$$

$$\xi_i \geq 0, \quad i=1,2,\cdots,n$$

- 拉格朗日函数：

$$L(w,b,\xi,\alpha,\mu) \equiv \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i\left(y_i\left(w\cdot x_i+b\right)-1+\xi_i\right) - \sum_{i=1}^{n}\mu_i\xi_i$$

C代表惩罚系数，即如果某个x是属于某一类，但是它偏离了该类，跑到边界上后者其他类的地方去了，C越大表明越不想放弃这个点，边界就会缩小（sklearn里面默认是1）

目标函数

- 接下来就是同样的，求解一个拉格朗日对偶问题，得到一个原问题的对偶问题的表达式：

对 w,b,ξ 求偏导

$$\frac{\partial L}{\partial w}=0\Rightarrow w=\sum_{i=1}^{n}\alpha_i y_i\phi(x_n)$$

$$\frac{\partial L}{\partial b}=0\Rightarrow 0=\sum_{i=1}^{n}\alpha_i y_i$$

$$\frac{\partial L}{\partial \xi}=0\Rightarrow C-\alpha_i-\mu_i=0$$

目标函数

- 带入得：

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

求极大得：

$$\max_{\alpha} \; -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j (x_i \cdot x_j) + \sum_{i=1}^{n}\alpha_i$$

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

$$C - \alpha_i - \mu_i = 0$$

$$\alpha_i \geq 0$$

$$\mu_i \geq 0, \quad i = 1,2,\ldots,n$$

$$0 \leq \alpha_i \leq C$$

目标函数

- 整理得：

$$\min_{\alpha} \quad \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j\left(x_i \cdot x_j\right)-\sum_{i=1}^{n}\alpha_i$$

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

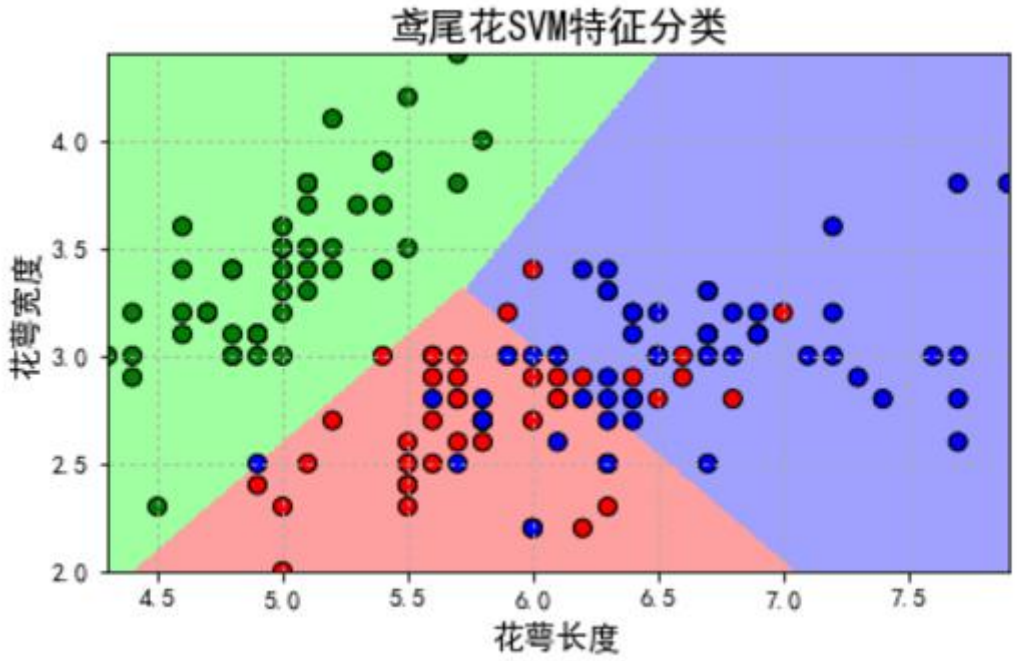
$$0 \le \alpha_i \le C, \quad i = 1,2,\ldots,n$$

α_i的取值范围是与线性可分的对偶问题表达式的不同之处。在线性不可分情况下得到的对偶问题，不同的地方就是α的范围从[0, +∞)，变为了[0, C]，增加的惩罚ε没有为对偶问题增加什么复杂度。

案例一：鸢尾花数据分类

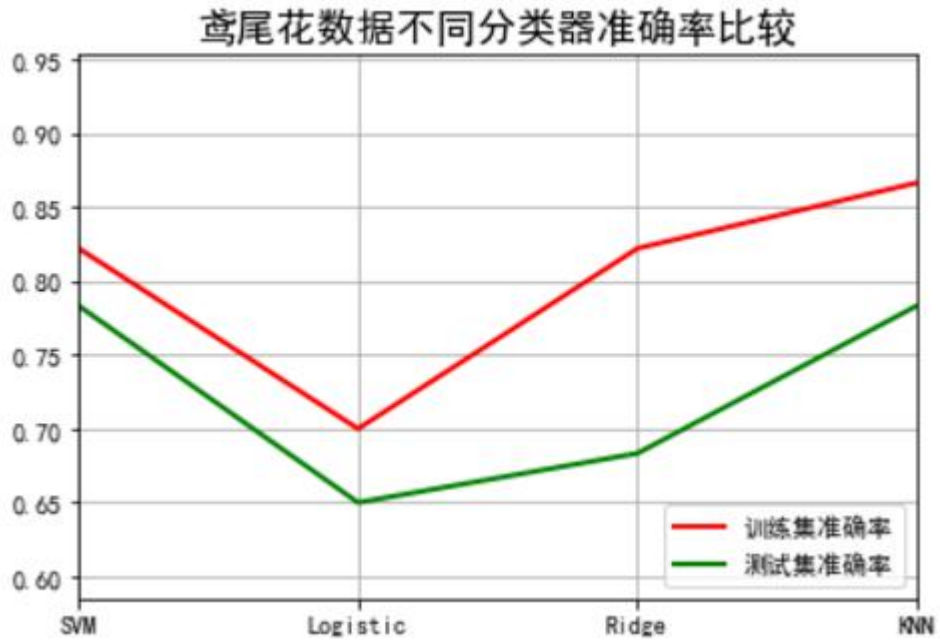
- 使用线性SVM对鸢尾花数据进行分类操作

```
clf = svm.SVC(C=1, kernel='linear')
clf.fit(x_train, y_train)
```

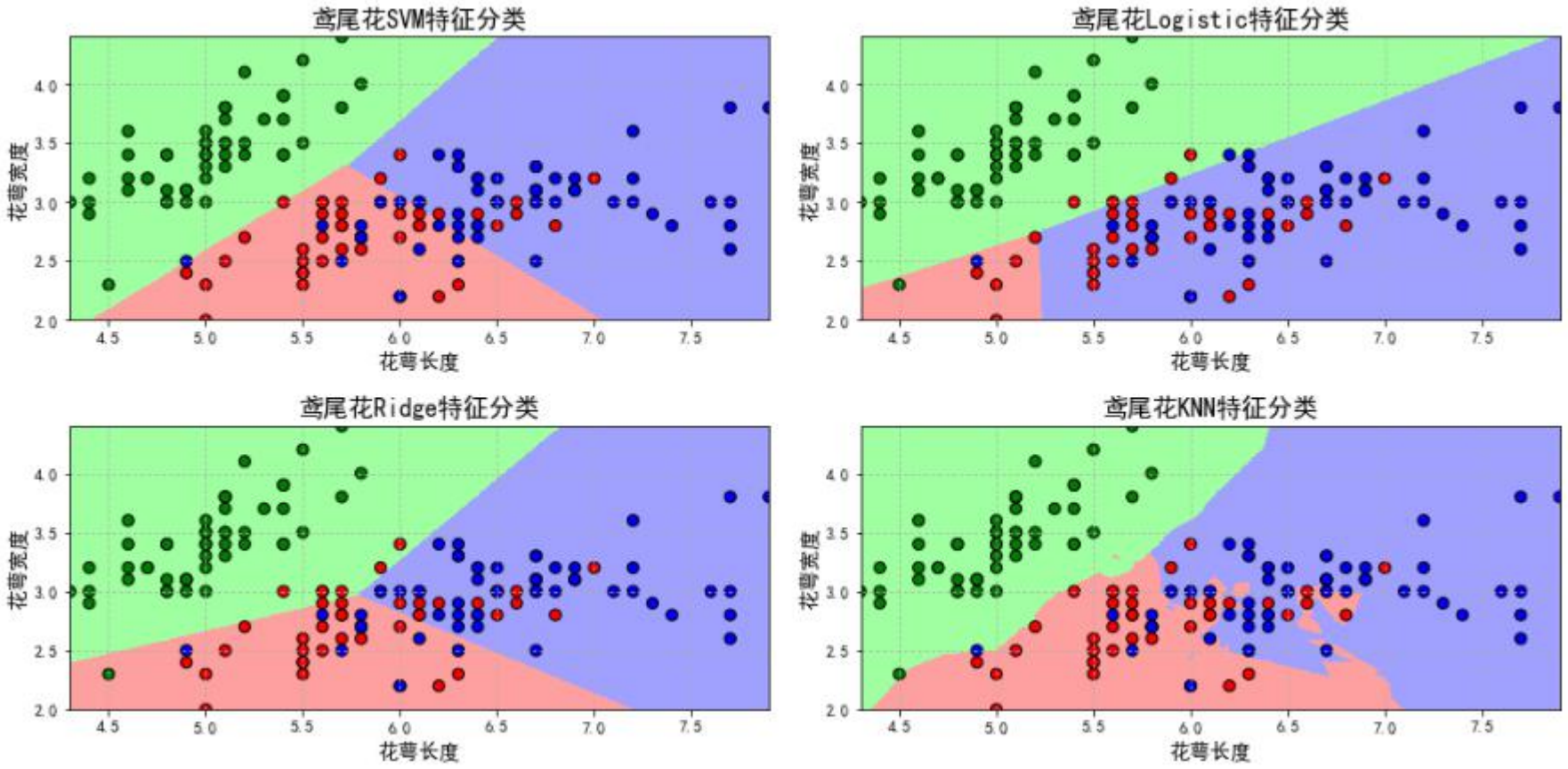


案例二：鸢尾花数据不同分类器效果比较

- 使用线性SVM、 LogisticRegression、RidgeClassifier、 knn对鸢尾花数据进行分类效果对比



案例二：鸢尾花数据不同分类器效果比较



SMO

- SVM 序列最小化方法SMO
- 简单的优化问题----两个拉格朗日乘子优化问题的解 ，假设 $\alpha 1$ 和 $\alpha 2$ 是变量，其他是定值：

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

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

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

SMO

$$\min_{\alpha_1,\alpha_2} W(\alpha_1,\alpha_2)$$

$$= \frac{1}{2}\kappa_{11}\alpha_1^2 + \frac{1}{2}\kappa_{22}\alpha_2^2 + y_1y_2\alpha_1\alpha_2\kappa_{12} - (\alpha_1+\alpha_2) \qquad s.t.\quad \alpha_1y_1+\alpha_2y_2 = -\sum_{i=3}^{N} y_i\alpha_i = \zeta$$

$$+ y_1\alpha_1\sum_{i=3}^{N} y_i\alpha_i\kappa_{i1} + y_2\alpha_2\sum_{i=3}^{N} y_i\alpha_i\kappa_{i2} \qquad 0 \le \alpha_i \le C$$

SMO

- 迭代公式 ：

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

$$\eta=\kappa(x_1, x_1)+\kappa(x_2, x_2)-2\kappa(x_1, x_2)=\|\Phi(x_1)-\Phi(x_2)\|^2$$

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

$$\alpha_j^{new}=\alpha_j^{old}+\frac{y_j(E_i-E_j)}{\eta}$$

SMO

- 退出条件：

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

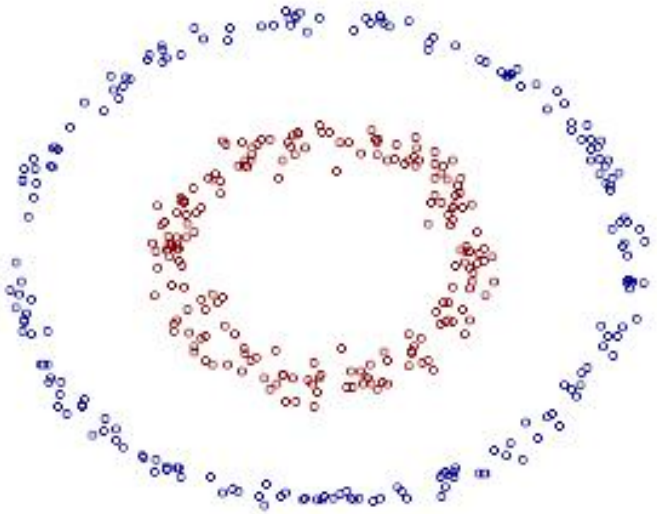
$$0 \le \alpha_i \le C, \quad i = 1,2,\ldots n$$

$$y_i \cdot g(x_i) = \begin{cases} \ge 1, & \{x_i | \alpha_i = 0\} \text{// 落在边界外} \\ = 1, & \{x_i | 0 < \alpha_i < C\} \text{// 落在边界上} \\ \le 1, & \{x_i | \alpha_i = C\} \text{// 落在边界内} \end{cases}$$

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

核函数

- 如图所示的两类数据，分别分布为两个圆圈的形状，这样的数据本身就是线性不可分的，此时咱们该如何把这两类数据分开呢？由我们的经验可以得到，一个理想的分界应该是一个“圆圈”而不是一条线（超平面）。如果用 X1 和 X2 来表示这个二维平面的两个坐标的话， 我们知道一条二次曲线（圆圈是二次曲线的一种特殊情况）的方程可以写作这样的形式：



$$a_1X_1 + a_2X_1^2 + a_3X_2 + a_4X_2^2 + a_5X_1X_2 + a_6 = 0$$

课程内容

- 注意上面的形式，如果我们构造另外一个五维的空间，其中五个坐标的值分别为 Z1=X1, Z2=X1^2, Z3=X2, Z4=X2^2, Z5=X1X2，那么显然，上面的方程在新的坐标系下可以写作：

$$\sum_{i=1}^{5} a_i Z_i + a_6 = 0$$

关于新的坐标 Z ，这正是一个超平面(hyper plane) 的方程！也就是说，如果我们做一个映射 ϕ:R2→R5 ，将 X 按照上面的规则映射为 Z ，那么在新的空间中原来的数据将变成线性可分的，从而使用之前我们推导的线性分类算法就可以进行处理了。这正是 Kernel 方法处理非线性问题的基本思想。

课程内容

- 这样一来问题就解决了吗？似乎是的：拿到非线性数据，就找一个映射，然后一股脑把原来的数据映射到新空间中，再做线性 SVM 即可。不过事实上没有这么简单！其实刚才的方法稍想一下就会发现有问题：在最初的例子里，我们对一个二维空间做映射，选择的新空间是原始空间的所有一阶和二阶的组合，得到了五个维度；如果原始空间是三维，那么我们会得到 19 维的新空间，这个数目是呈爆炸性增长的，这给计算带来了非常大的困难，而且如果遇到无穷维的情况，就根本无从计算。

课程内容

- 不妨还是从最开始的简单例子出发，设两个向量 $x_1 = (\eta_1, \eta_2)^T$ 和 $x_2 = (\xi_1, \xi_2)^T$，而即是到前面说的五维空间的映射，因此映射过后的内积为：

$$\langle \phi(x_1), \phi(x_2) \rangle = \eta_1\xi_1 + \eta_1^2\xi_1^2 + \eta_2\xi_2 + \eta_2^2\xi_2^2 + \eta_1\eta_2\xi_1\xi_2$$

另外，我们又注意到：

$$(\langle x_1, x_2 \rangle + 1)^2 = 2\eta_1\xi_1 + \eta_1^2\xi_1^2 + 2\eta_2\xi_2 + \eta_2^2\xi_2^2 + 2\eta_1\eta_2\xi_1\xi_2 + 1$$

二者有很多相似的地方，实际上，我们只要把某几个维度线性缩放一下，然后再加上一个常数维度就是一样的了。

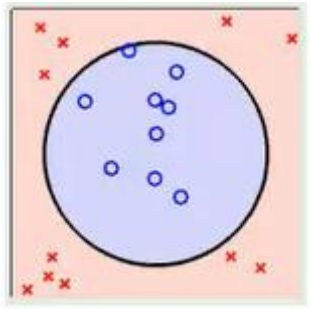
核函数

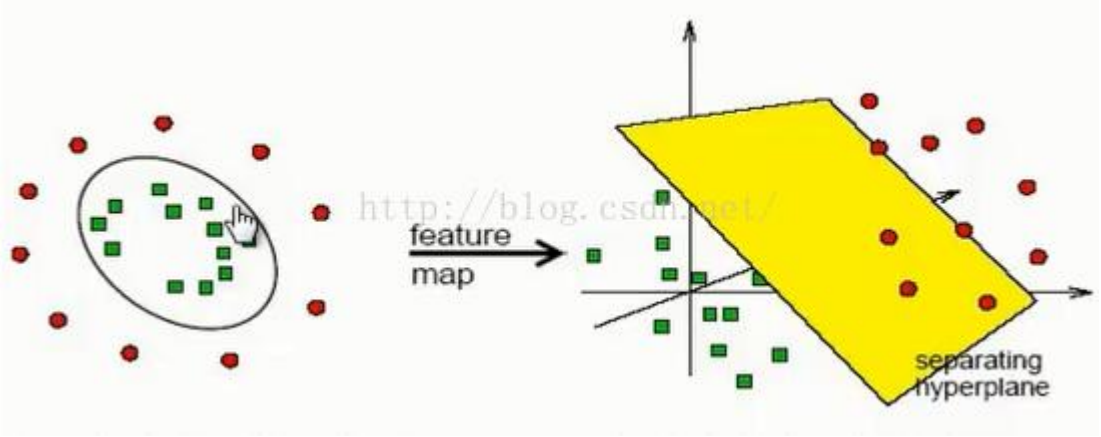
核函数Kernel

事实上，大部分时候数据并不是线性可分的，这个时候满足这样条件的超平面就根本不存在。在上文中，我们已经了解到了SVM处理线性可分的情况，那对于非线性的数据SVM如何处理呢？对于非线性的情况，SVM 的处理方法是选择一个核函数 κ，通过将数据映射到高维空间，来解决在原始空间中线性不可分的问题。

具体来说，在线性不可分的情况下，支持向量机首先在低维空间中完成计算，然后通过核函数将输入空间映射到高维特征空间，最终在高维特征空间中构造出最优分离超平面，从而把平面上本身不好分的非线性数据分开。

核函数





核函数

- 核函数与之前计算的区别：

1、一个是映射到高维空间中，然后再根据内积的公式进行计算；

2、而另一个则直接在原来的低维空间中进行计算，而不需要显式地写出映射后的结果。

我们把这里的计算两个向量在隐式映射过后的空间中的内积的函数叫做核函数(Kernel Function)，例如，在刚才的例子中，我们的核函数为：

$$\kappa(x_1, x_2) = (\langle x_1, x_2 \rangle + 1)^2$$

核函数

- 常用的几个核函数：

多项式核：$\kappa(x_1, x_2) = (\langle x_1, x_2 \rangle + R)^d$

高斯核函数 RBF：$\kappa(x_1, x_2) = \exp\left(-\frac{\|x_1 - x_2\|^2}{2\sigma^2}\right)$

线性核：$\kappa(x_1, x_2) = \langle x_1, x_2 \rangle$

核函数

- 核函数的本质：

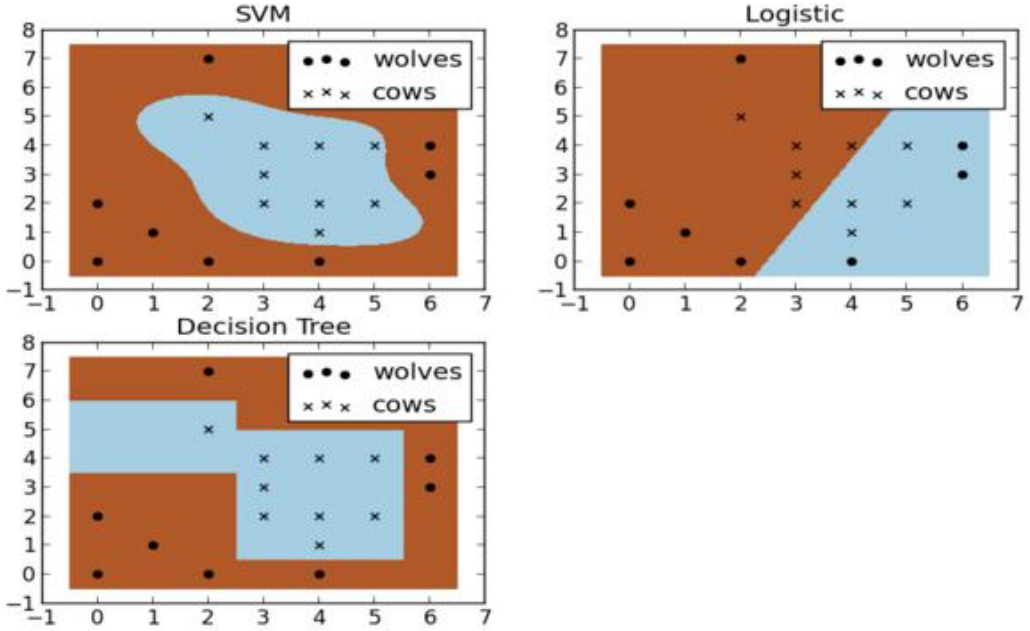
实际中，我们会经常遇到线性不可分的样例，此时，我们的常用做法是把样例特征映射到高维空间中去(如上文的那幅图所示，映射到高维空间后，相关特征便被分开了，也就达到了分类的目的）。

但进一步，如果凡是遇到线性不可分的样例，一律映射到高维空间，那么这个维度大小是会很高。

此时，就需要核函数，核函数的价值在于它虽然也是将特征进行从低维到高维的转换，但核函数它事先在低维上进行计算，而将实质上的分类效果表现在了高维上，也就如上文所说的避免了直接在高维空间中的复杂计算。

核函数

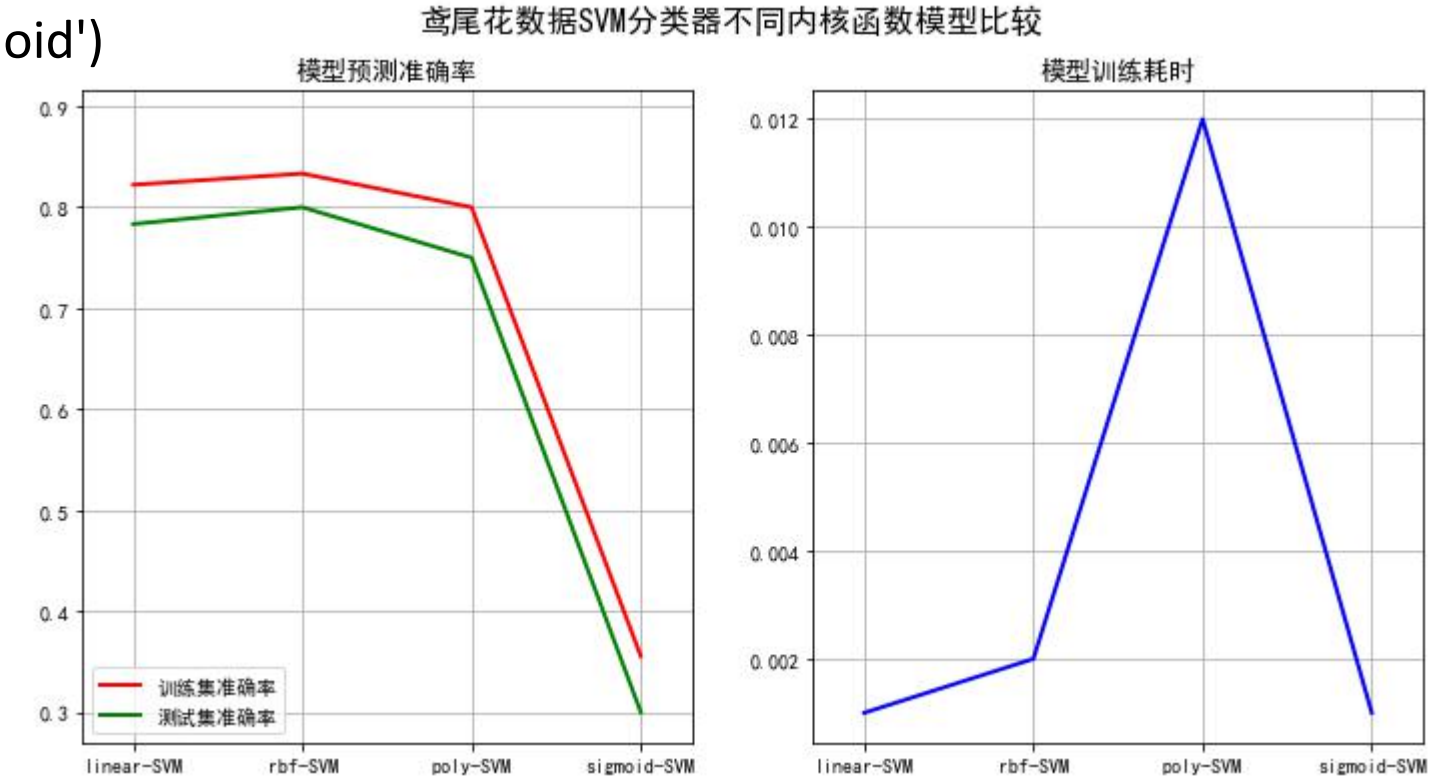
- 最后引用这里的一个例子举例说明下核函数解决非线性问题的直观效果。
- 假设现在你是一个农场主，圈养了一批羊群，但为预防狼群袭击羊群，你需要搭建一个篱笆来把羊群围起来。但是篱笆应该建在哪里？你很可能需要依据牛群和狼群的位置建立一个“分类器”，比较图这几种不同的分类器，我们可以看到SVM完成了一个很完美的解决方案。



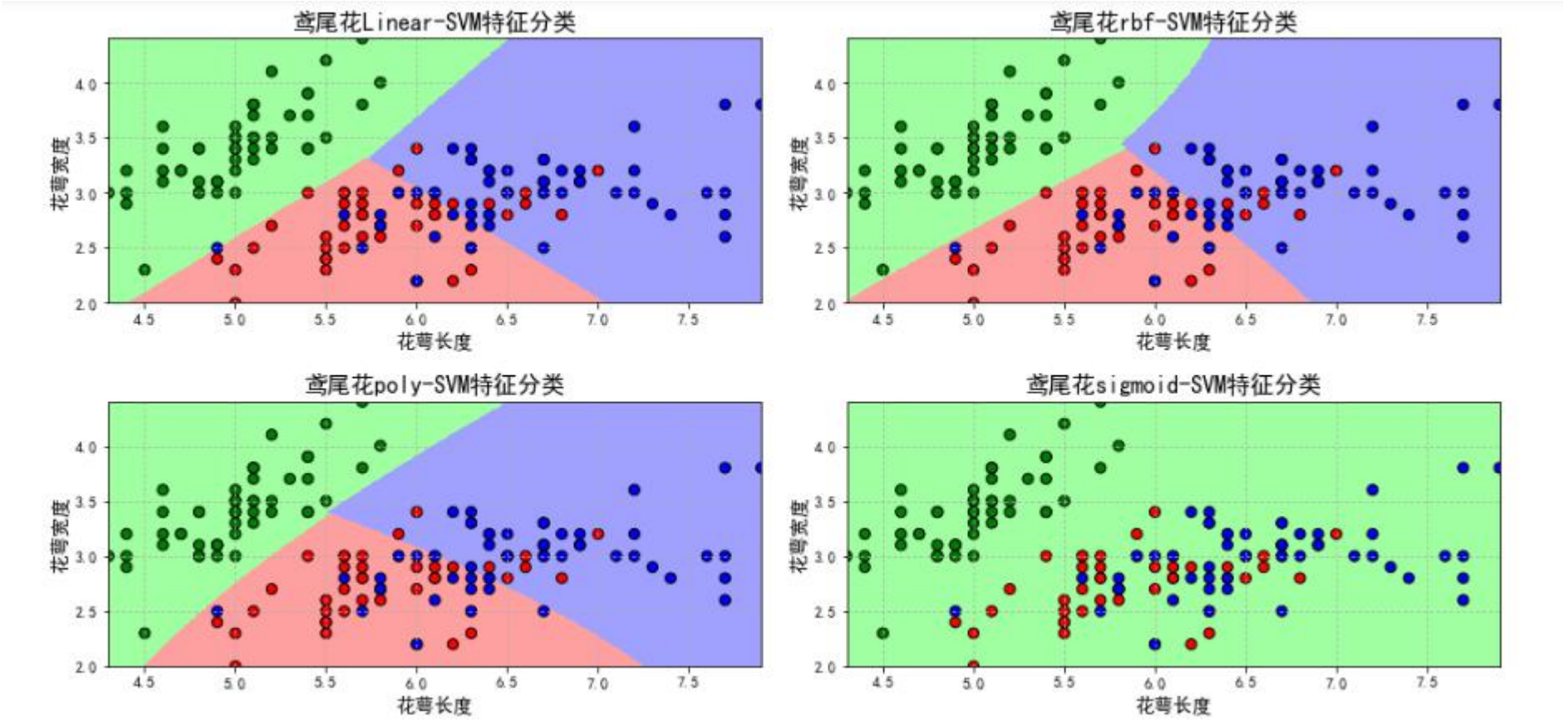
案例三：不同SVM核函数效果比较

- 使用SVM不同核函数对鸢尾花数据进行分类操作

```
svm1 = SVC(C=1, kernel='linear')
svm2 = SVC(C=1, kernel='rbf')
svm3 = SVC(C=1, kernel='poly')
svm4 = SVC(C=1, kernel='sigmoid')
```



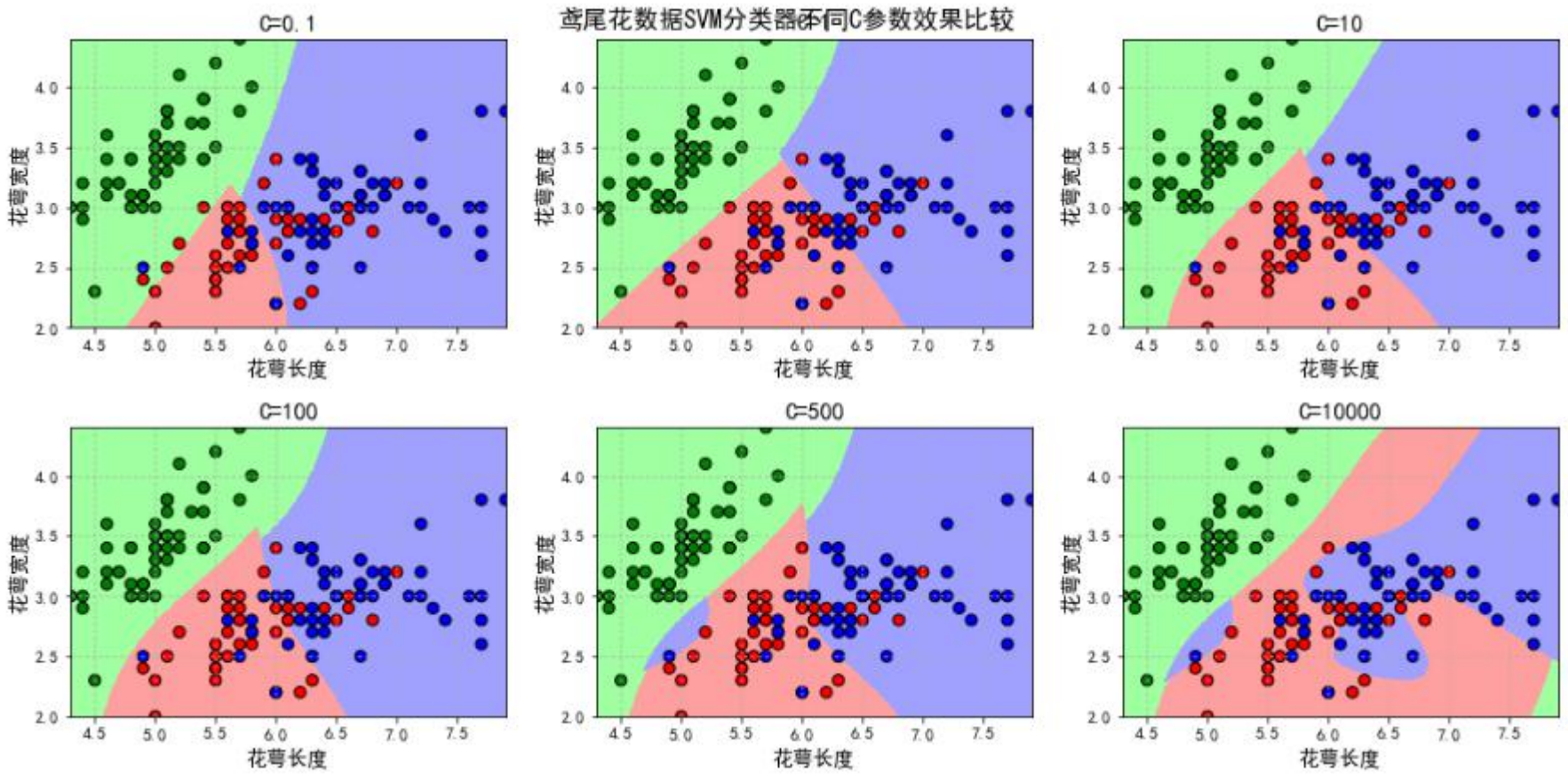
案例三：不同SVM核函数效果比较



案例四：不同SVM惩罚参数C值不同效果比较

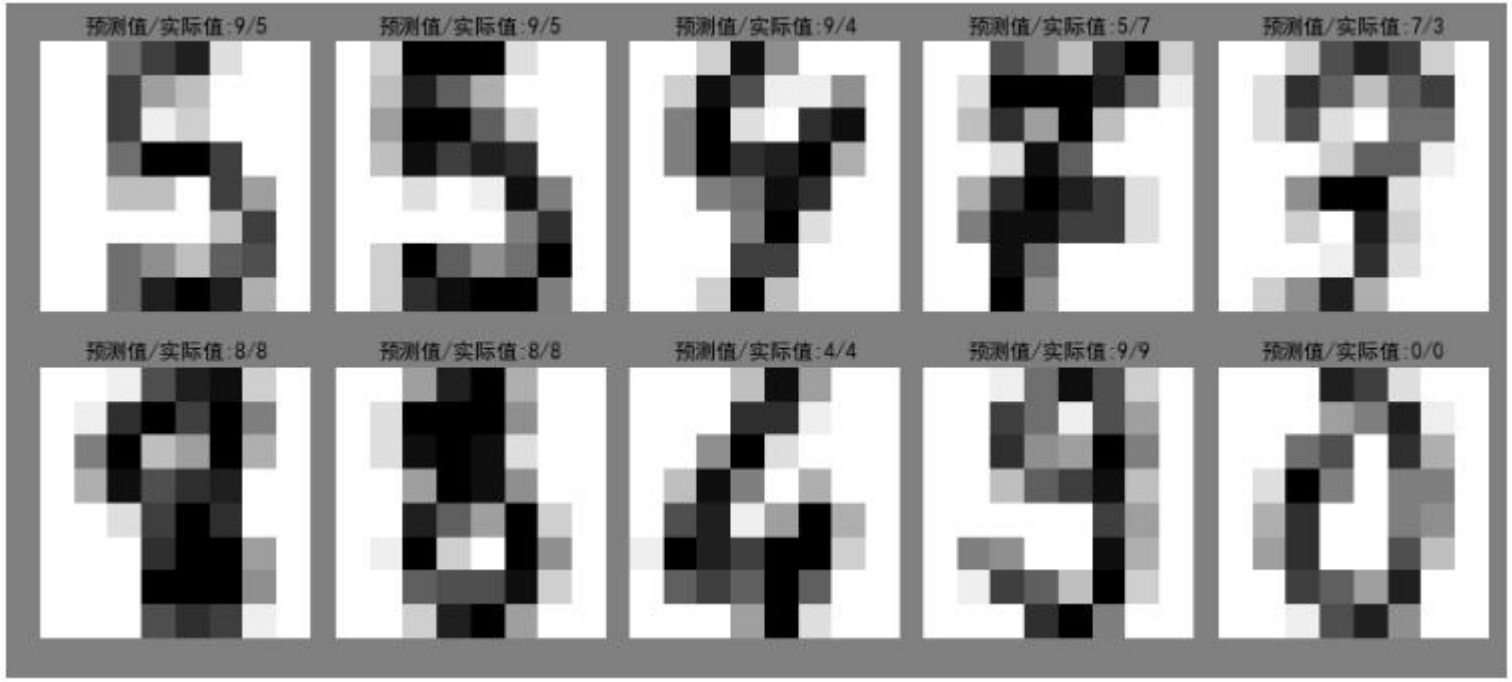
- 使用SVM惩罚参数C值效果对比

```
svm1 = SVC(C=0.1,
kernel='rbf')
svm2 = SVC(C=1,
kernel='rbf')
svm3 = SVC(C=10,
kernel='rbf')
svm4 = SVC(C=100,
kernel='rbf')
svm5 = SVC(C=500,
kernel='rbf')
svm6 = SVC(C=100000,
kernel='rbf')
```



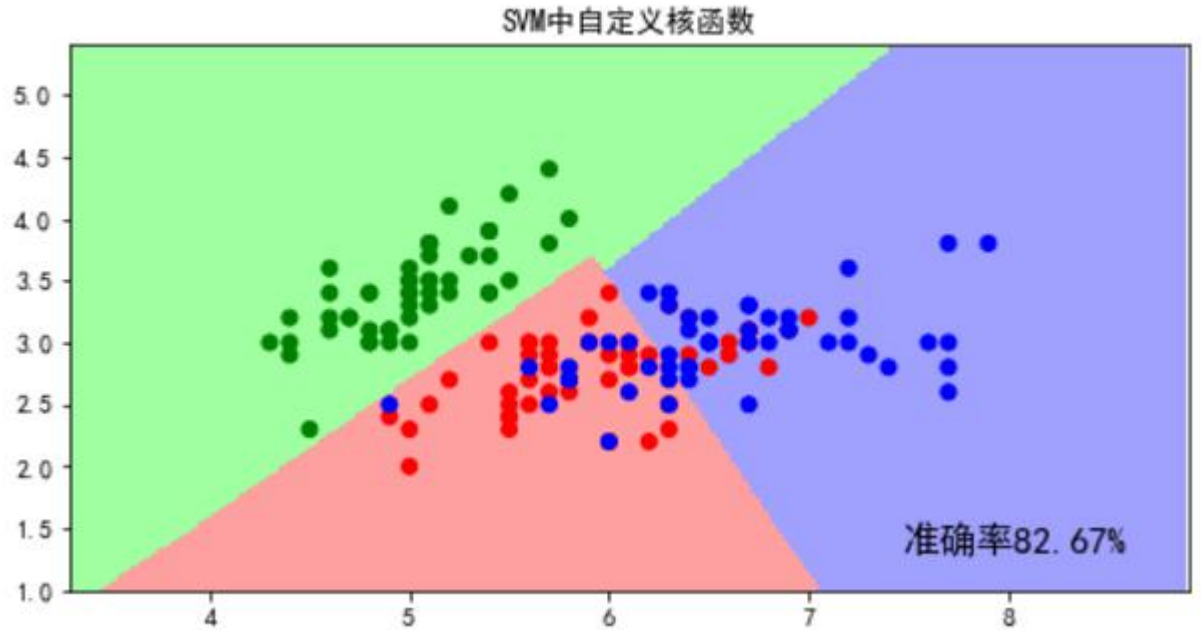
综合案例1：手写数字识别

- 使用SVM对手写数字的识别



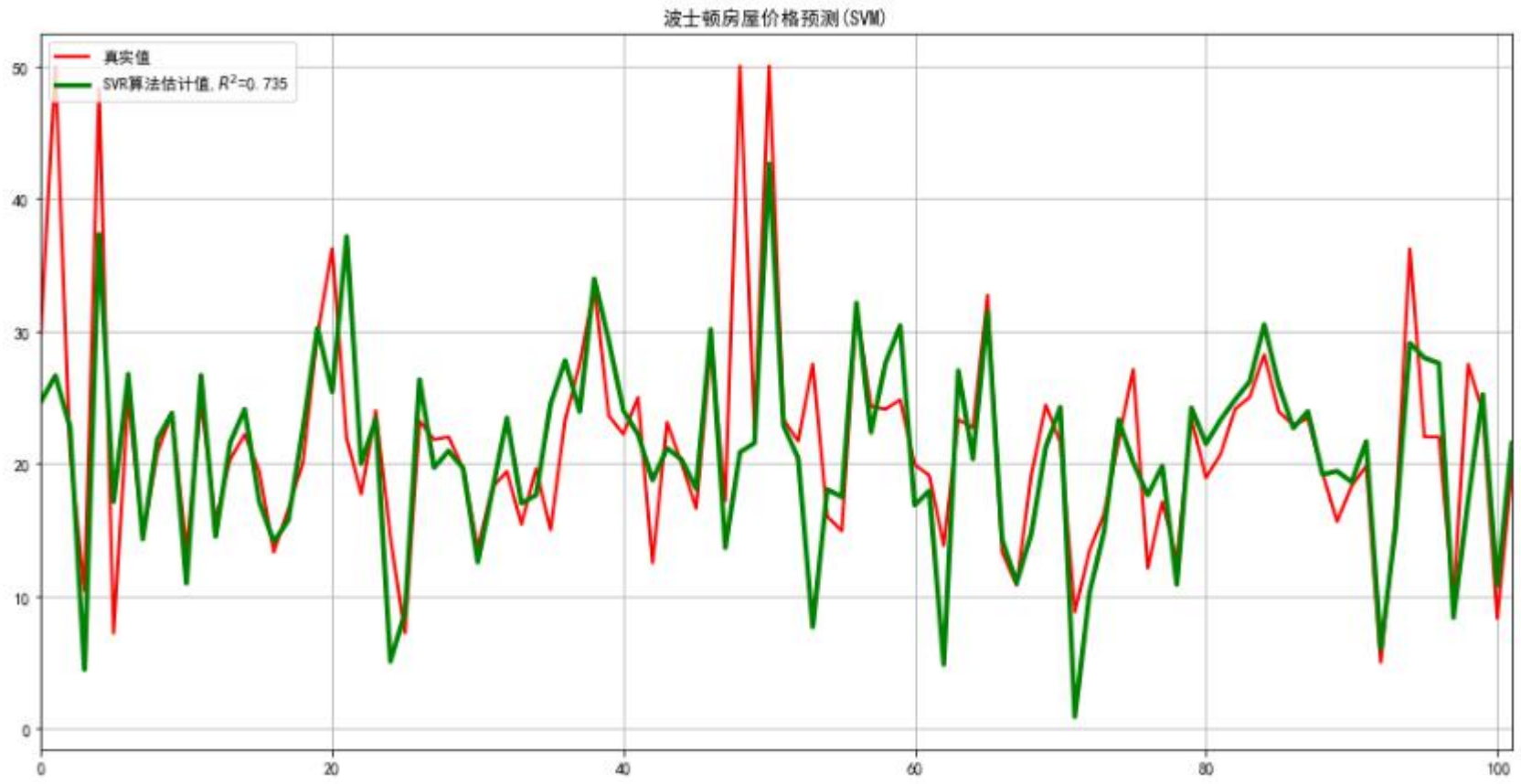
综合案例2：自定义SVM内部核函数

- 练习SVM自定义函数的应用



综合案例3：使用SVM预测波士顿房价

- 使用SVM中SVR对波士顿房价预测，熟悉SVM的回归问题应用



THANK YOU

上海育创网络科技有限公司