支持向量机

18 人智 于松松

问题描述：

在进行感知机，以及逻辑斯蒂回归问题的讨论中，我们都以二分类问题为例。其中对于线性不可分的数据集，感知机和逻辑斯蒂回归已然失效，逻辑斯蒂回归对感知机进行的优化，是进行了一种思维上的“偷懒”，不再武断地判断预测数据属于哪个类，而是“圆滑”地告诉我们该数据属于哪一个的概率多大，通过对极大似然函数的参数估计，实现了模型的学习；通过极大似然估计，实现了数据类别的预测。但是这种优化并没有对线性不可分数据集提出解决方法，支持向量机，运用了几何间隔最大化（硬间隔或软间隔）和核技巧，将问题通过拉格朗日对偶转化，“投靠”了更高维的空间，即将线性不可分的数据集通过核技巧映射到高维空间(事实上，支持向量机并没有直接将空间进行映射，而是通过核技巧将内积进行从欧氏空间到希尔伯特空间的转换)，企图从高维空间找到一个超平面能将数据集正确分类。

直观思路：

该思路的比较直观的解释，如果我们利用有限的特征不能将事物分类，那么我们可以将这些有限的特征通过某些关系组合构成新的特征从而识别出某类事物。如图 1 是原始数据集在某平面的映射，可见它是线性不可分的，我们通过

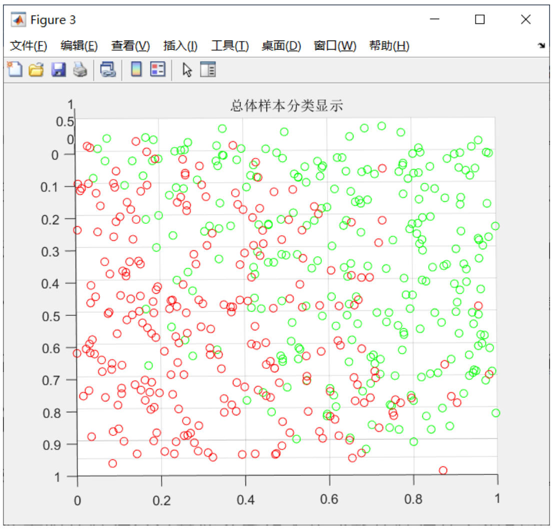


图 1 拥有两个特征的数据集

核技巧可以将其“映射”到高维空间，我们猜想在高维空间这个数据集是线性可分的，于是得到图 2，重新映射后，发现可以进行线性分类。

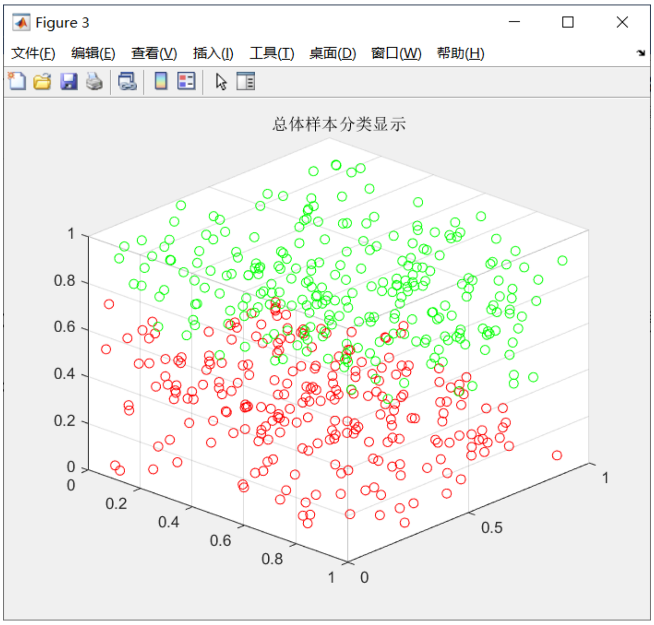


图 2 三维空间的映射

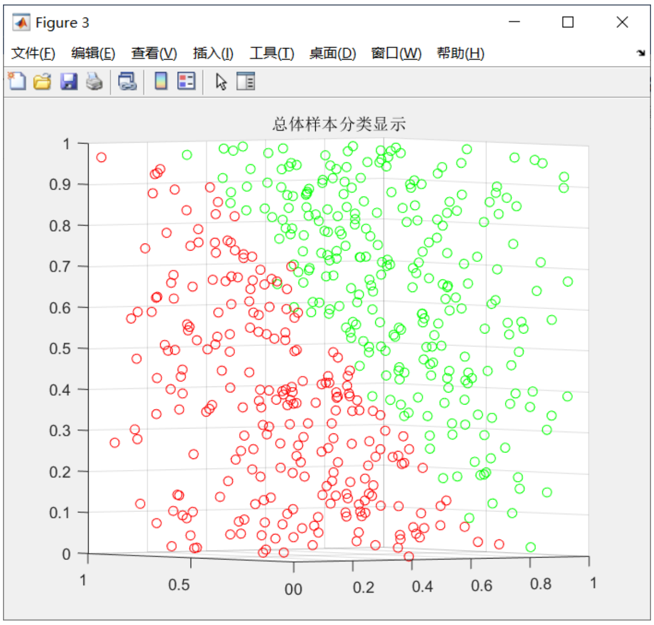


图 3 重新找到一个平面进行映射

适用范围：

二分类模型，属于非线性分类器，策略为间隔最大化。

对于数据集不同的情况，采取的学习策略也不同。

表一

训练数据集类型	学习策略
线性可分	硬间隔最大化
线性近似可分	软间隔最大化
线性不可分	核技巧+软间隔最大化

输入输出：

输入：

训练集（线性可分 or 线性近似可分 or 线性不可分）$T=\{(x_1,y_1),(x_2,y_2),\cdots,(x_N,y_N)\}$，

其中 $x_i \in \chi = R^n$， $y_i \in Y=\{+1,-1\}, i=1,2,\cdots,N$

输出：分离超平面和分类决策函数

函数关系：

支持向量 x_i 位置	拉格朗日乘子 $0 \le \alpha_i^* \le C$	松弛变量 $0 \le \xi_i$
间隔边界上	$\alpha_i^* < C$	$\xi_i = 0$
间隔边界与超平面之间且正确分类	$\alpha_i^* = C$	$0 < \xi_i < 1$
在分离超平面上	$\alpha_i^* = C$	$\xi_i = 1$
间隔边界与超平面之间且错误分类	$\alpha_i^* = C$	$\xi_i > 1$

C 惩罚参数；$|wx+b| = 1$ 为间隔边界；$|wx+b| = 0$ 为分离超平面

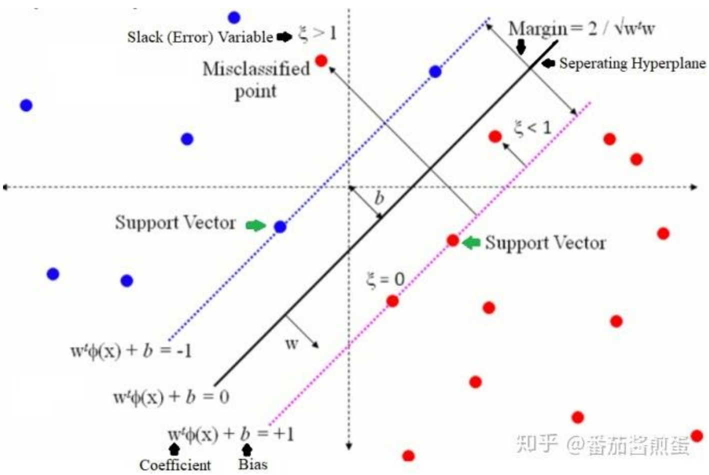


图 4 超平面与支持向量

直观理解：如果一个样例点靠近我们的超平面，我们将它对应的 α_i^* 增大，类似于惩罚，为了使松弛变量发挥作用，设置一个惩罚参数，作为上界，α_i^* 如果大于这个参数则均取得这个参数，以便于让松弛项发挥作用。当松弛变量 $\xi_i = 0$，则说明不需要进行软约束既满足正确分类且满足硬间隔最大化，样例在间隔边界上；当松弛变量增大，说明该样例已经进入了两个间隔之间，松弛变量的数值越大，则说明我们需要对

这个样采取更放松的条件，当放松的条件超过某个值，即ξ_i>1，说明该样例已经被目前的超平面错误分类。

待续。。。

概念补充：

关于“凸”

一、几何体的向量表示

在介绍凸集等概念之前，首先介绍一下空间几何体的向量表示，下面在定义凸集概念时使用到了线段的线段表示。先通过一个例子来认识一下如何使用向量表示线段

已知二维平面上两定点A(5, 1)、B(2, 3)，给出线段AB的方程表示如下：

$$\begin{cases} x_1 = \theta * 5 + (1-\theta) * 2 \\ x_2 = \theta * 1 + (1-\theta) * 3 \end{cases} \qquad \theta \in [0,\ 1]$$

如果将点A看成向量a，点B看成向量b，则线段AB的向量表示为：

$$\overrightarrow{x} = \theta\overrightarrow{a} + (1-\theta) * \overrightarrow{b} \qquad \theta \in [0,\ 1]$$

而直线的向量表示是：

$$\overrightarrow{x} = \theta\overrightarrow{a} + (1-\theta) * \overrightarrow{b} \qquad \theta \in R$$

由此衍生推广到高维，可得以下几何体的向量表示，三角形的向量表示：

$$\overrightarrow{x} = \theta_1\overrightarrow{a}_1 + \theta_2\overrightarrow{a}_2 + \theta_3\overrightarrow{a}_3 \qquad \theta_i \in [0,\ 1] \text{ and } \sum \theta_i = 1$$

三维平面的向量表示：

$$\overrightarrow{x} = \theta_1\overrightarrow{a}_1 + \theta_2\overrightarrow{a}_2 + \theta_3\overrightarrow{a}_3 \qquad \theta_i \in R \text{ and } \sum \theta_i = 1$$

超几何体的向量表示：

$$\overrightarrow{x} = \theta_1\overrightarrow{a}_1 + \theta_2\overrightarrow{a}_2 + \ldots + \theta_k\overrightarrow{a}_k \qquad \theta_i \in [0,\ 1] \text{ and } \sum \theta_i = 1$$

超平面的向量表示：

$$\overrightarrow{x} = \theta_1\overrightarrow{a}_1 + \theta_2\overrightarrow{a}_2 + \ldots + \theta_k\overrightarrow{a}_k \qquad \theta_i \in R \text{ and } \sum \theta_i = 1$$

1、凸集

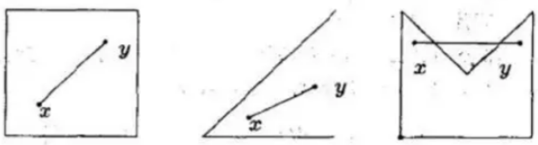
集合C内任意两点间的线段也均在集合C内，则称集合C为凸集，数学定义为：

$$\forall x_1, x_2 \in C, \forall \theta \in [0,1], \text{则} x = \theta \cdot x_1 + (1-\theta) \cdot x_2 \in C$$

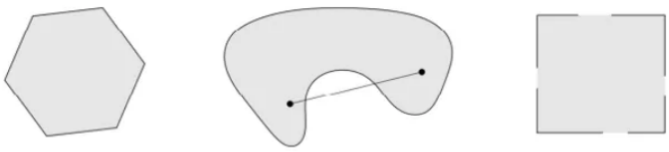
k个点的版本：

$$\forall x_1, x_2, \cdots x_k \in C, \theta_i \in [0,1] \text{且} \sum_{i=1}^{k} \theta_i = 1, \text{则} x = \sum_{i=1}^{k} \theta_i x_i \in C$$

上面凸集定义中使用到了线段的向量表示，含义是如果点x1和点x2在集合C内，则线段x1x2上所有点都在集合c内，凸集的交集仍是凸集，下面展示几个凸集示例：



(a) 有界凸集 (b) 无界凸集 (c) 非凸集



凸集 非凸集 非凸集

2、凸函数

凸函数定义为：

$$f : C \subseteq R^n -> R^1, \ C.x_1, x_2 \in C, :$$

$$f(\alpha_1 x_1 + \alpha_2 x_2) <= \alpha_1 f(x_1) + \alpha_2 f(x_2) \quad \sum \alpha_i = 1, \alpha_i >= 0$$

则成 f(x) 为定义在凸集C上的凸函数

严格凸函数定义：设 **f ⊆ Rn–> R1，C是凸集，对于x1, x2∈C** 都有：

$$f(\alpha_1 x_1 + \alpha_2 x_2) < \alpha_1 f(x_1) + \alpha_2 f(x_2) \quad \sum \alpha_i = 1, \alpha_i > 0$$

则成 **f(x)** 为定义在凸集C上的严格凸函数

凸函数的等价定义：设**f ⊆ Rn–> R1，C是凸集，对于x1, x2, x3∈C且x1<x2<x3**，下式成立则 **f(x)** 为凸函数：

$$\frac{f(x_2) - f(x_1)}{x_2 - x_1} <= \frac{f(x_3) - f(x_1)}{x_3 - x_1} <= \frac{f(x_3) - f(x_2)}{x_3 - x_2}$$

3、凸函数定义几何解释

对于凸函数公式描述：

$$f(\alpha_1 x_1 + \alpha_2 x_2) <= \alpha_1 f(x_1) + \alpha_2 f(x_2) \quad \sum \alpha_i = 1, \alpha_i >= 0$$

如下图所示，设A1、A2是凸函数曲线上的两个点，他们对应的横坐标x1<x2，且x∈(x1, x2)，则存在 α1， α2>0且 α1+ α2=1，使得x= α1x1+ α2x2，过点x做x轴的垂线交函数于A，交直线A1A2于B点，则上式左端即为A的纵坐标，右端即为B的纵坐标：

$$y_A = f(\alpha_1 x_1 + \alpha_2 x_2)$$

$$y_B = \alpha_1 f(x_1) + \alpha_2 f(x_2)$$

因此，凸函数的几何含义是：函数任意两点A1和A2之间的部分位于弦A1A2的下方或曲线任一点切线上方，不严谨一个说法：割线始终位于两点间函数曲线的上方

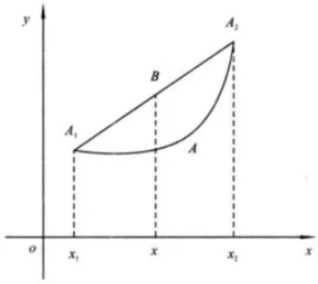


图 1 凸函数的几何性质

关于仿射：

从$\mathbb{R}^n$到$\mathbb{R}^m$的映射$x \mapsto Ax + b$，称为仿射变换（affine transform）或仿射映射（affine map），其中 A 是一个$m \times n$矩阵，b 是一个 m 维向量。当$m = 1$时，称上述仿射变换为仿射函数。

仿射函数即由 1 阶多项式构成的函数，一般形式为 f(x)=Ax+b，这里，A 是一个 m×k 矩阵，x 是一个 k 向量，b 是一个 m 向量，实际上反映了一种从 k 维到 m 维的空间映射关系，只有 b=0 时，仿射函数才可以叫“线性函数”（“正比例”关系）。

一般称线性组合 $p_1x_1+p_2x_2+\ldots+p_nx_n$，其中 $p_1+p_2+\ldots+p_n=1$ 为仿射组合；一般称所有 $p_i \geq 0$ 的仿射组合为凸组合。其实一般意义上的仿射函数是一个矩阵函数，如果构成一个类似 LMI 的不等式，可以成为仿射矩阵不等式。

实例运行：

预先准备了数据集，数据为随机生成，并且根据预设的函数关系（为验证 SVM 的有效性，采用了非线性的函数关系），进行了分类。

使用了线性可分以及线性不可分的数据集对不同的核函数进行测试和验证，以下为测试结果。

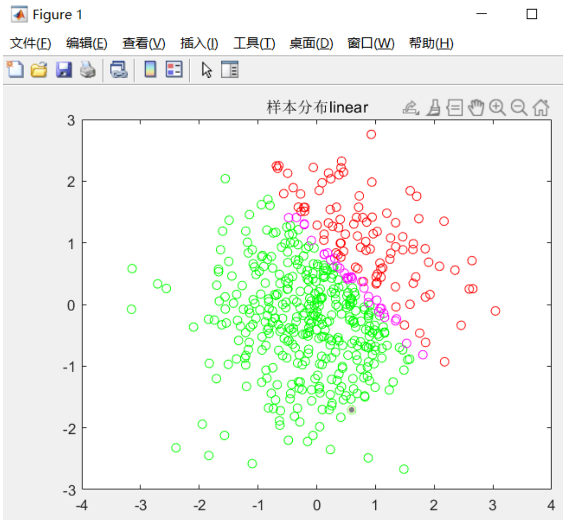


图 5 线性核函数

训练完成!

应用模型：SVM 支持向量机

优化算法:interior-point-convex

核函数：linear

测试集识别率为：0.998000

图 6 线性核函数准确率

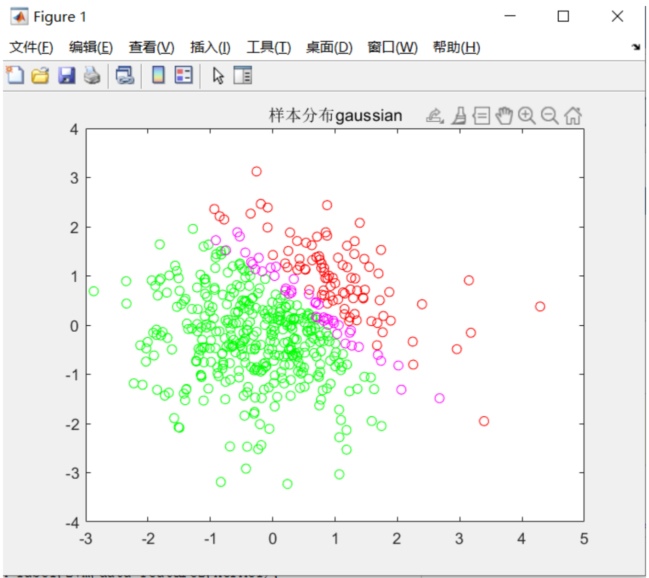


图 7 高斯核函数

```
训练完成!
应用模型：SVM 支持向量机
优化算法:interior-point-convex
核函数：gaussian
测试集识别率为：0.992000
>>
```

图 8 高斯核函数准确率

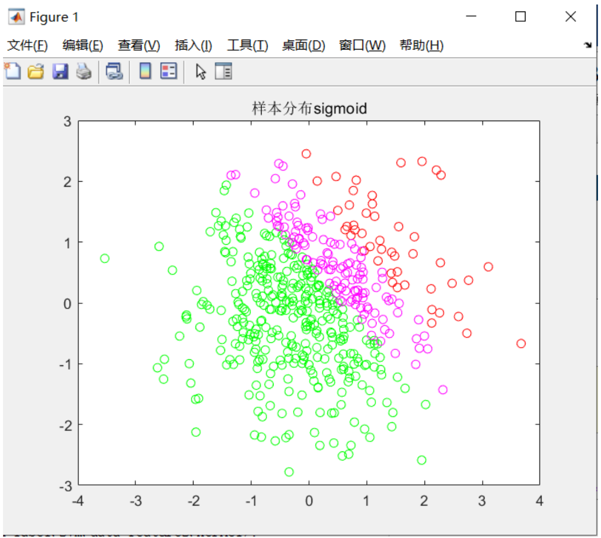


图 9 sigmoid 核函数

```
训练完成!
应用模型：SVM 支持向量机
优化算法:interior-point-convex
核函数：sigmoid
测试集识别率为：0.988000
```

图 10 sigmoid 核函数准确率

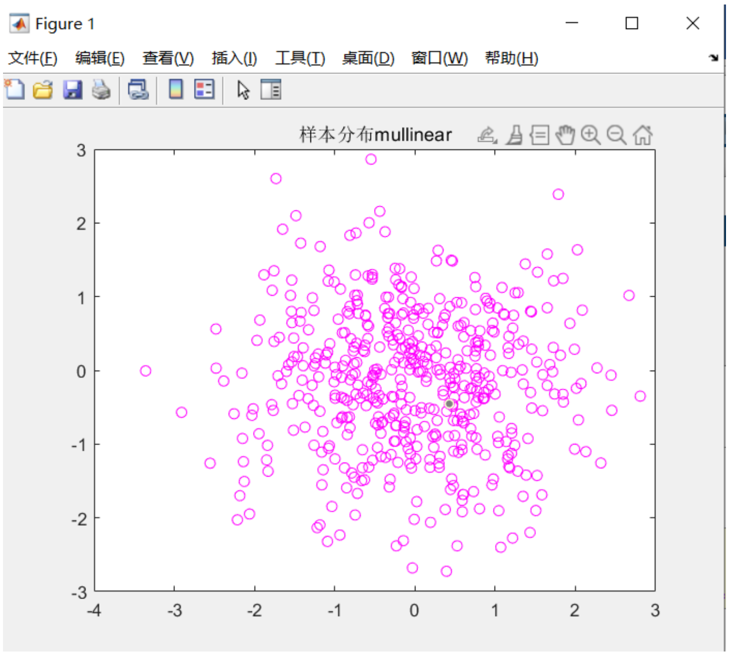


图 11 多项式核函数

```
训练完成!
应用模型：SVM 支持向量机
优化算法:interior-point-convex
核函数：mullinear
测试集识别率为：0.766001
```

图 12 多项式核函数准确率

线性不可分数据集:

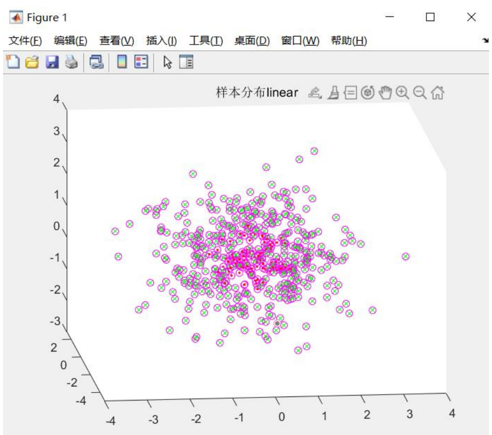


图 13 线性核函数

训练完成!
应用模型：SVM 支持向量机
优化算法:interior-point-convex
核函数：linear
测试集识别率为：0.784001

图 14 线性核函数准确率

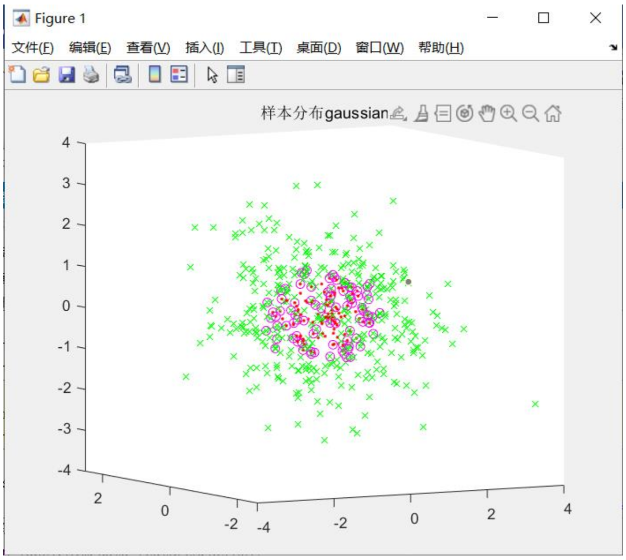


图 15 高斯核函数

训练完成!
应用模型：SVM 支持向量机
优化算法:interior-point-convex
核函数：gaussian
测试集识别率为：0.978000

图 16 高斯核函数准确率

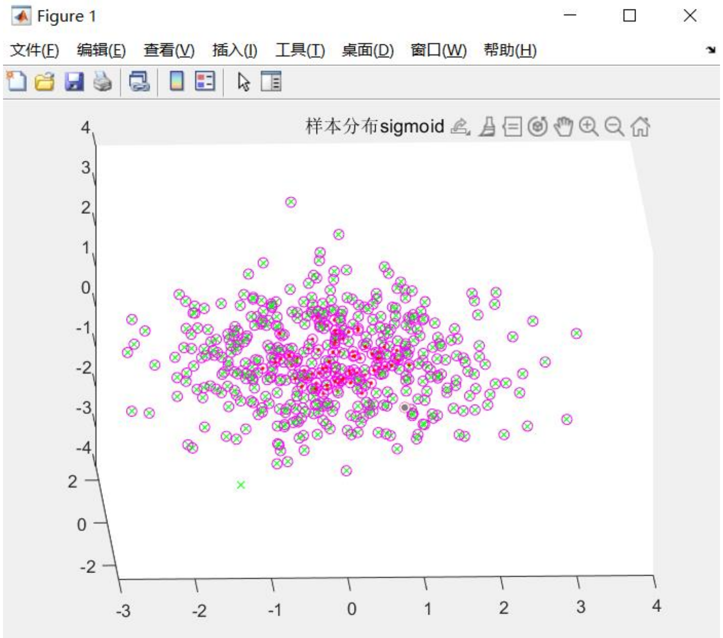


图 17 sigmoid 核函数

训练完成!
应用模型：SVM 支持向量机
优化算法:interior-point-convex
核函数：sigmoid
测试集识别率为：0.812001

图 18 sigmoid 核函数准确率

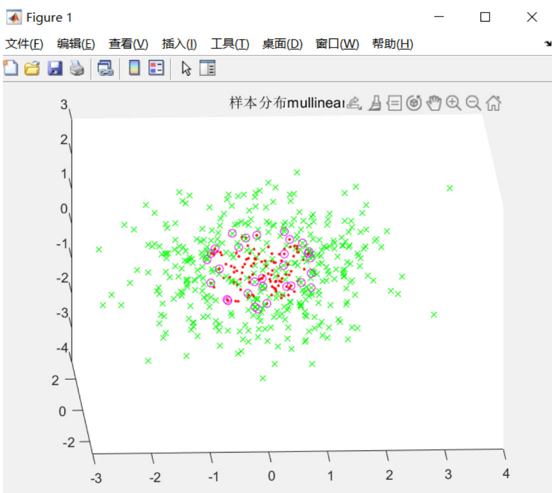


图 19 多项式核函数

```
训练完成!
应用模型：SVM 支持向量机
优化算法:interior-point-convex
核函数：mullinear
测试集识别率为：0.994000
```

图 20 多项式核函数准确率

结论：

由上不难看出，对于线性可分的数据集，线性核函数表现良好，但多项式核函数表现欠佳；但对于线性不可分数据集，则情况相反。高斯核函数对于两种情况表现都比较好，这说明高斯核函数对于空间映射更具有普适性。

基于支持向量机的多分类问题：

方法一：One-Versus-Rest（一对多）

训练时依次把某个类别的样本归为一类,其他剩余的样本归为另一类，这样 k 个类别的样本就构造出了 k 个 SVM。分类时将未知样本分类为具有最大分类函数值的那类。

优点：分类速度快，分离超平面个数为数据类别数。缺点：存在分类重叠与不可分类现象。

分类重叠现象：在进行测试时，对于每一类 label,svm 模型都会选择说测试数据属于这一类，导致都想“争抢”测试样本；此类问题比较容易解决，选择一个置信度最高的类即可（几何间隔最大）。

不可分类现象：在进行测试时，对于每一类 label，svm 都会说不属于自己这一

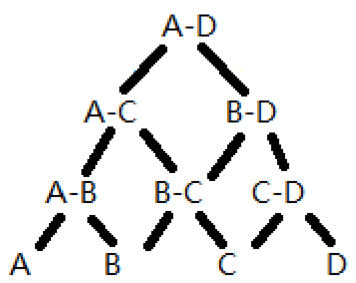
类，导致都选择“排斥”测试样本，这导致多分类 svm 失效。

数据集偏斜现象：训练集本身，对于每一类样本数据个数不对等，相差较多的情况，这样学得的超平面，为了保证间隔最小，会导致让小数据类别更多地被误分类，这是不难理解的，因为牺牲小部分样例误分类如果可以让大数目样本更多的被正确分类，这种牺牲对于损失函数下降是有意义的，从而导致在测试时本属于小数目样本的数据可能被误分类为大数目的样本类别中。One-Versus-Rest 方案正是人为的造成了数据集偏斜。

方法二：One-Versus-One（一对一）

具体做法是在任意两类样本之间设计一个 SVM，因此 k 个类别的样本就需要设计 k(k-1)/2 个 SVM。当对一个未知样本进行分类时，最后得票最多的类别即为该未知样本的类别。在训练时往往比一对多花的时间少，并且由于采用了投票表决的方式，一定程度上解决了样本不可分问题，但是分类的类别多的时候，会使 svm 数目爆炸。

方法三：Directed Acyclic Graph SVM（有向无环图）DAG SVM



DAG 存在的一个问题是单分类器的误判代价较高。如上图若根分类器判断错误，输入真实类别为 A，却被错分至右边，这次误判将直接导致输入不会被正确分类。故而在选择根分类器时需选择鲁棒性较强的一个二分类器，子树的根分类器同理。

方法四：层次支持向量机

待续。。。

参考：

机器学习中对核函数的理解:

https://link.zhihu.com/?target=http%3A//mp.weixin.qq.com/s%3F__biz%3DMzIxNDIwMTk2OQ%3D%3D%26mid%3D2649077019%26idx%3D1%26sn%3De0c4a6c502e3668e1dc410f21e531cfd%26scene%3D0%23wechat_redirect

核函数的定义和作用是什么？：

https://www.zhihu.com/question/24627666

机器学习概念篇：一文详解凸函数和凸优化，干货满满

https://mp.weixin.qq.com/s?src=11&timestamp=1619016438&ver=3022&signature=vq2-rqI8DivdbDhKDqFMHs3Pc7uV-eZr3yWYxWxlMdktd4m4tTrA5q0X8I8GwZ*bo8QcFZKWWU5X1sakYzBPFviW7siF48zmFxf-ww-R*F4AX5eaINFwM2tRxmQS*FfJ&new=1

拉格朗日对偶性

https://zhuanlan.zhihu.com/p/45431511

线性可分支持向量机推广到线性不可分（四）

https://zhuanlan.zhihu.com/p/42773288

机器学习-白板推导系列(七)-核方法（Kernel Method）笔记

https://zhuanlan.zhihu.com/p/337498283

支持向量机 SVM—实现多分类问题的解决方案

https://blog.csdn.net/weixin_44822636/article/details/110820652

收获：

·在面向过程编程时，也可以借助对象，将参数结构体化，让参数列表看起来简洁（不可乱用。。。）