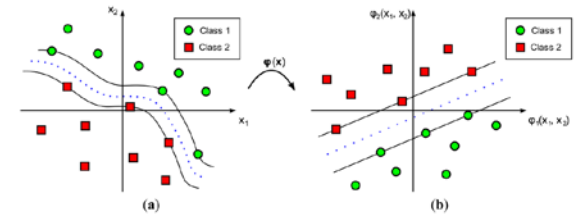


王颖 201722090323
黄孙培 201722090330



Support Vector Machine in Classification

Wang Ying

Outline :

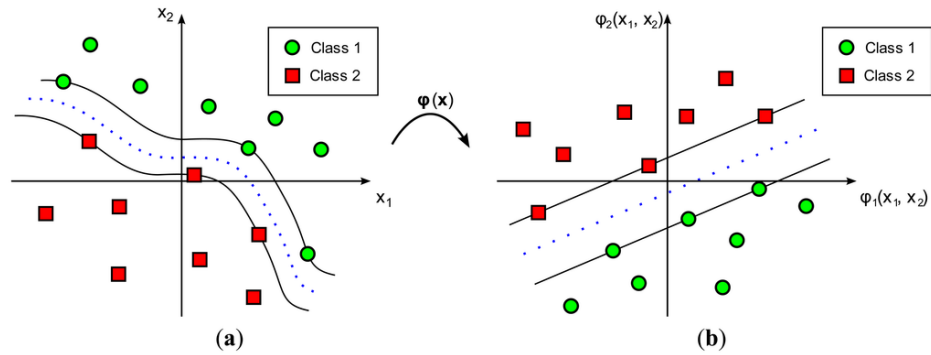
- ▶ ~~Face Recognition~~
- ▶ Data Visualization of 2 Dimension Data





~~Face Recognition~~

Face Recognition



~~Face Recognition~~

optimization finished, #iter = 757

nu = 0.000016

obj = -83.736190, rho = 0.881717

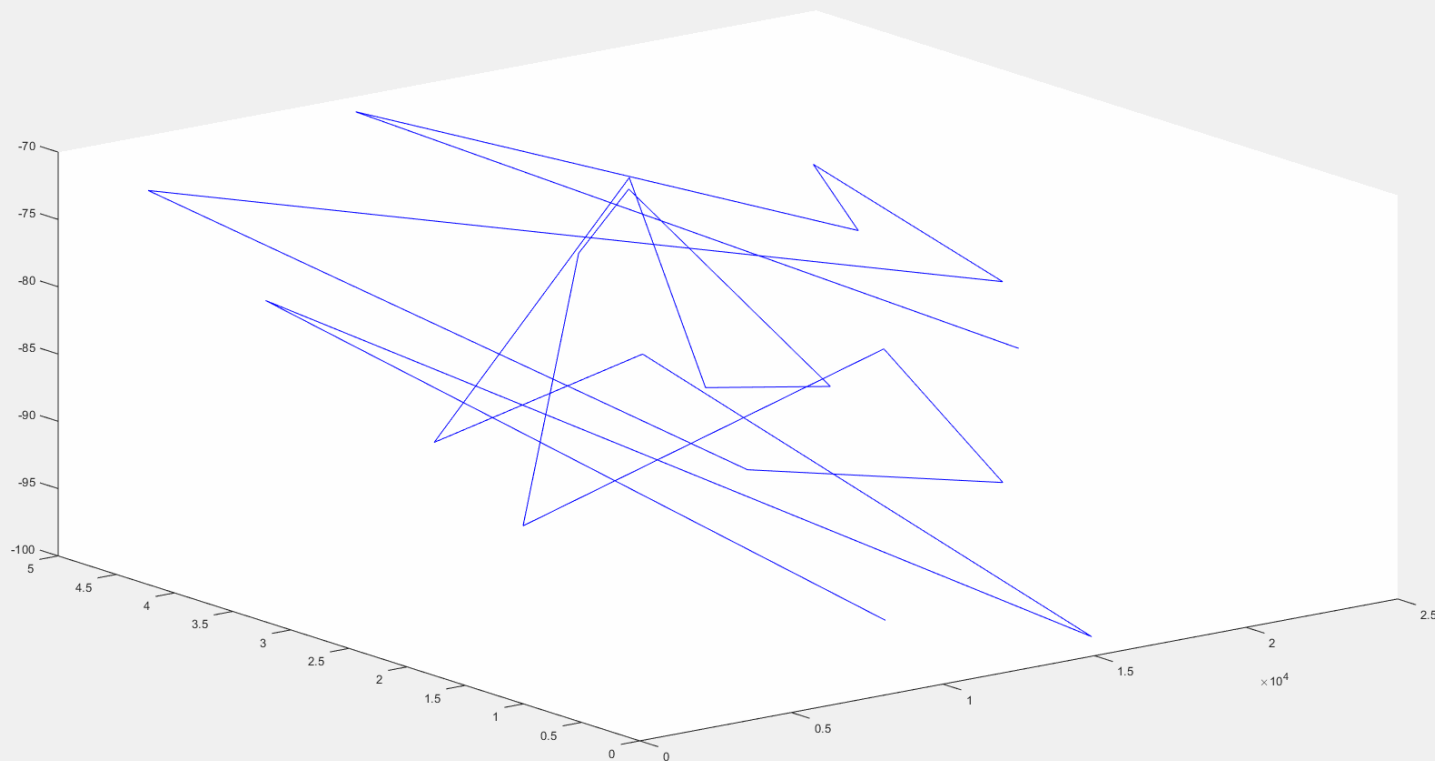
nSV = 324, nBSV = 0

Total nSV = 324

Cross Validation Accuracy = 99.625%

gbest_cost =

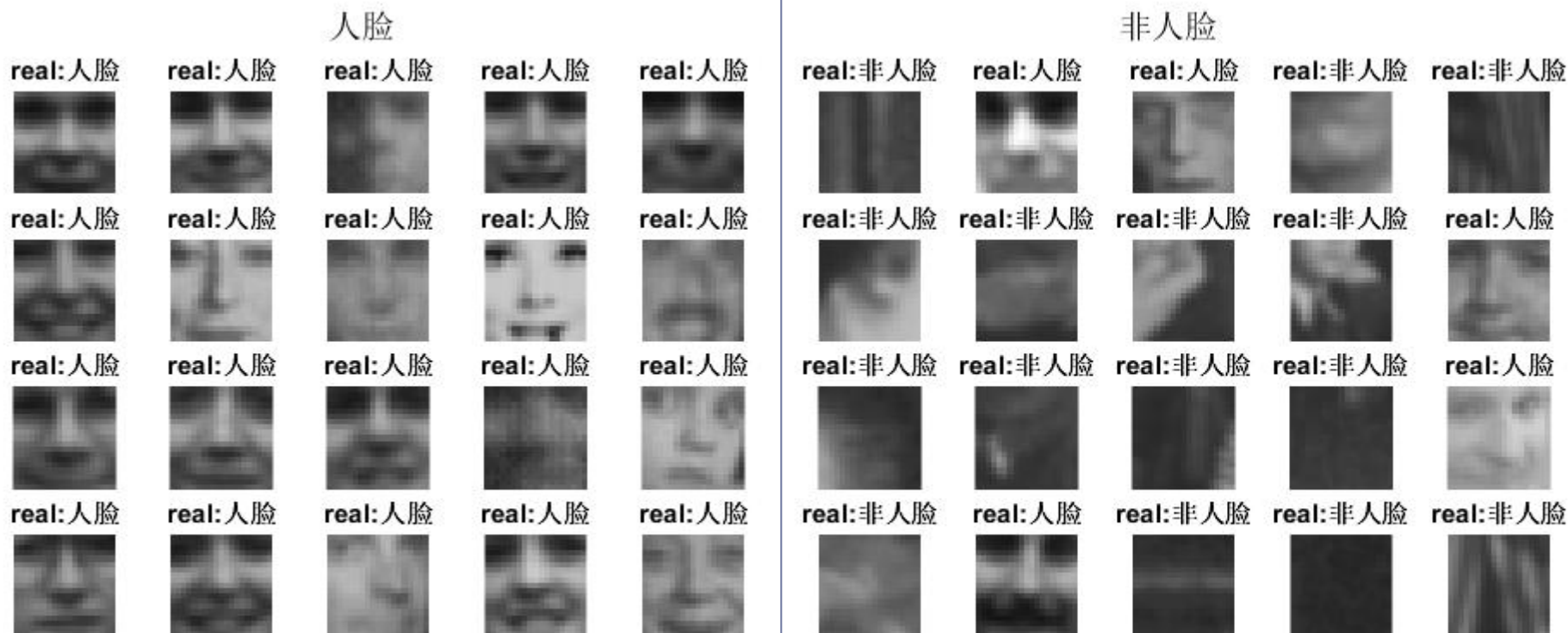
-99.6250



随便设了20个粒子20代

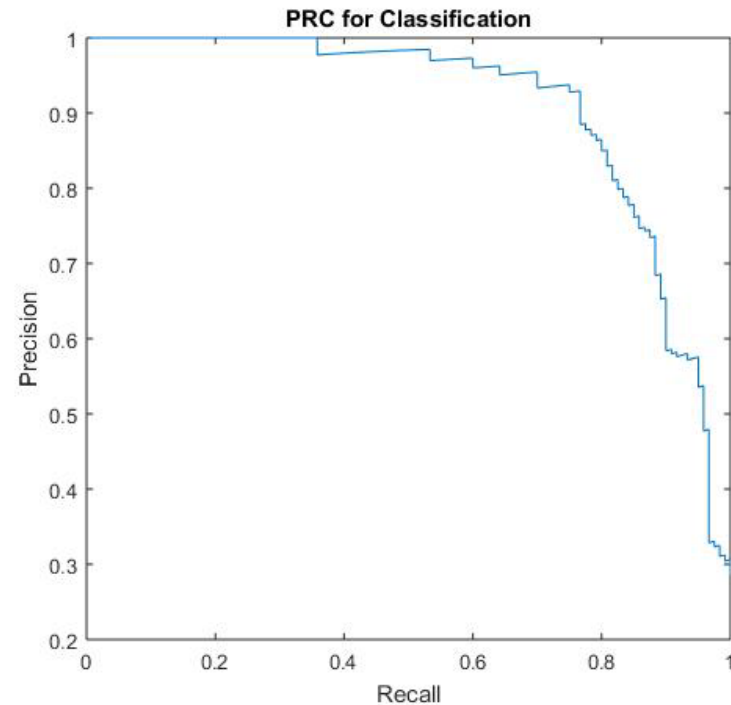
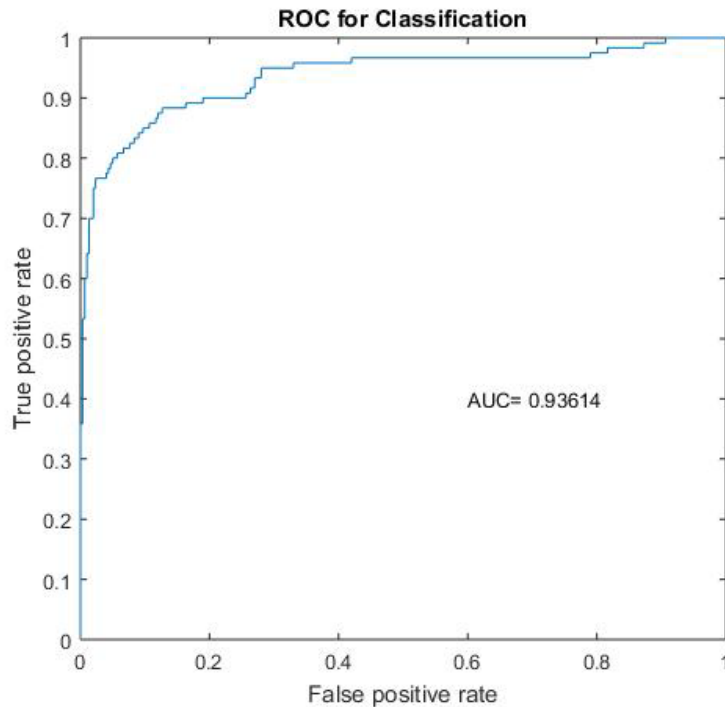
这里用20个粒子运动了20次找到了一个比较优的参数，对比网格寻优全局搜索，由于代数较少所以PSO只是停在了相对最优的位置，也没能画出等高线。

Result



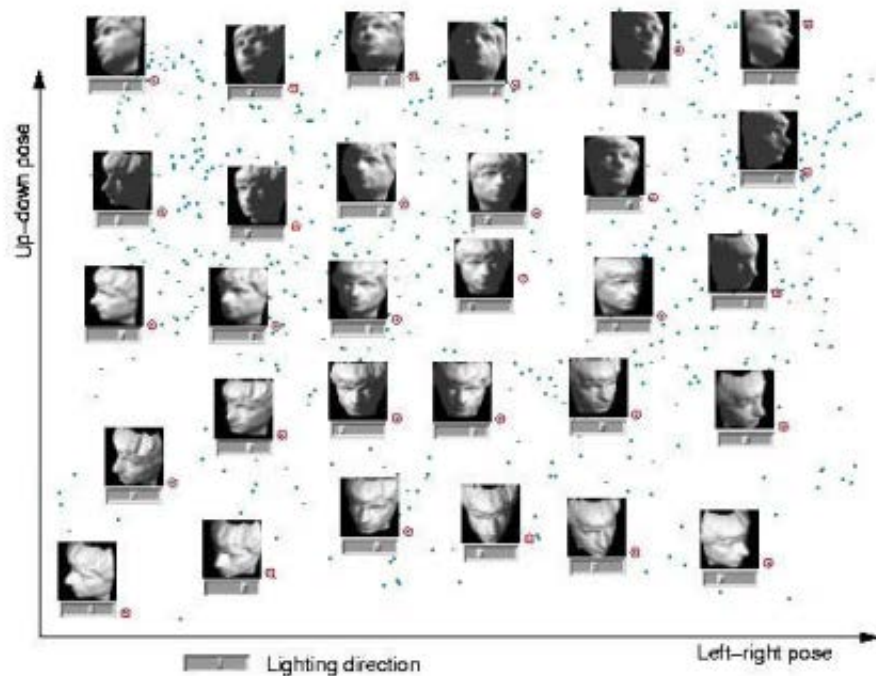
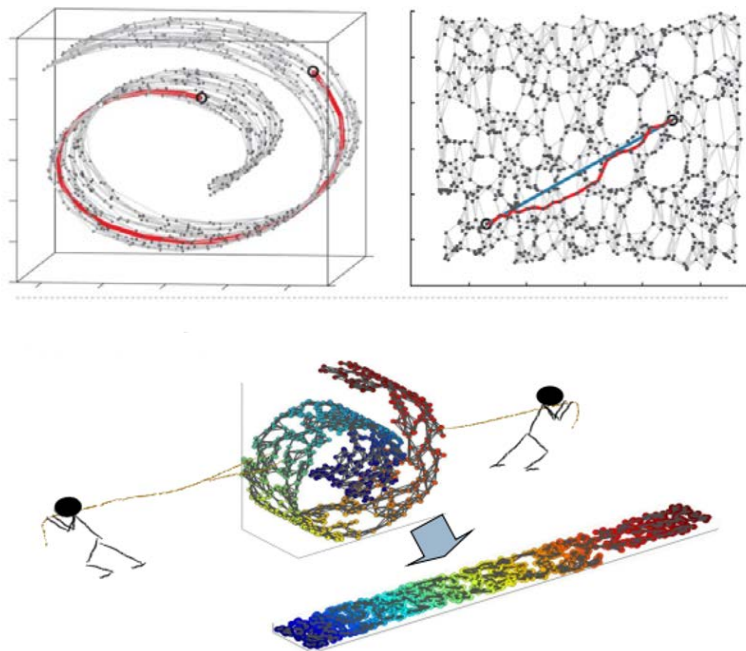
核函数将人脸数据从361维投射到了某个几乎线性可分的空间，对数据进行了分类。可以看到其没有足够的样本训练，所以对非正面的人脸或受光线影响较大的人脸还是会错分。

Performance of SVM in ~~Face Recognition~~



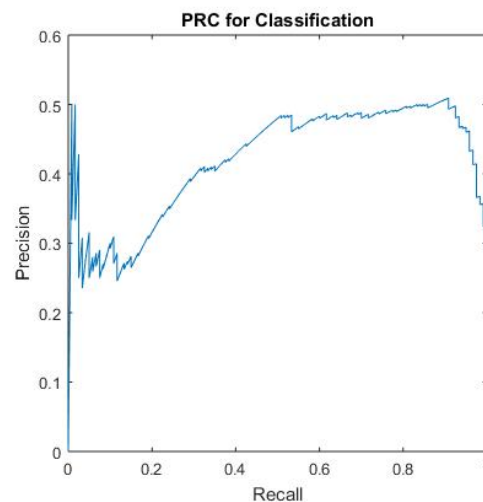
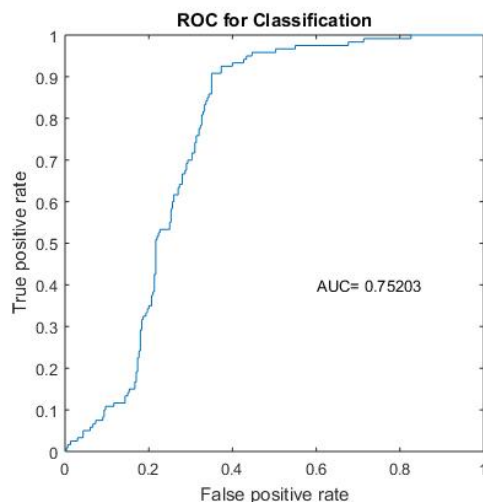
```
optimization finished, #iter = 277
nu = 0.759336
obj = -391.184092, rho = 3.371824
nSV = 490, nBSV = 483
Total nSV = 490
Cross Validation Accuracy = 84.875%
Accuracy = 85.4762% (359/420) (classification)
```

Preprocessing

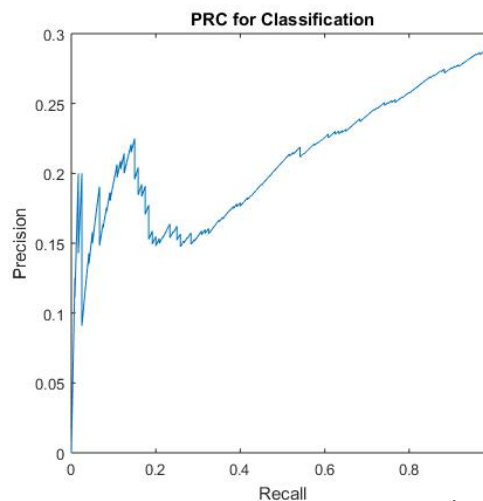
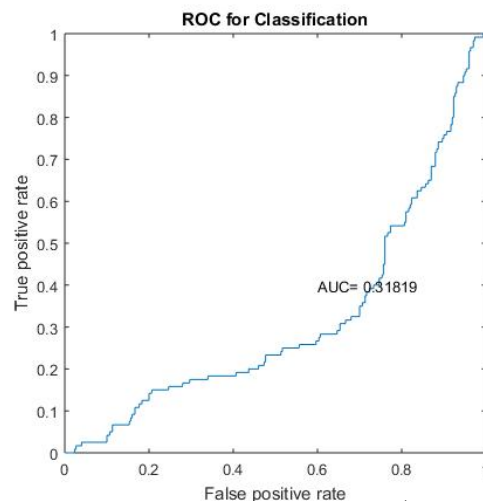


如果人脸数量足够多的话，可以通过ISOMAP，LLE等非线性降维找到人脸数据分布的流形，找到固定的模式，利用预处理后的特征进行分类。

Performance after dimension reduction



降到12维的性能曲线



降到3维的性能曲线

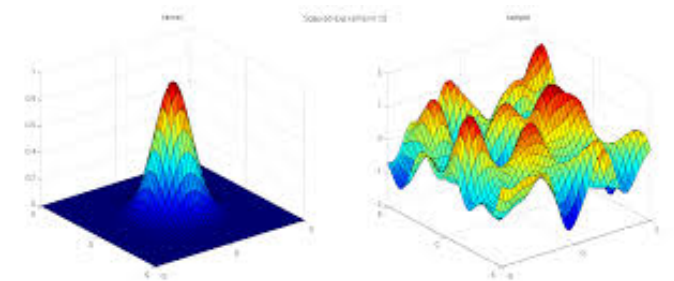
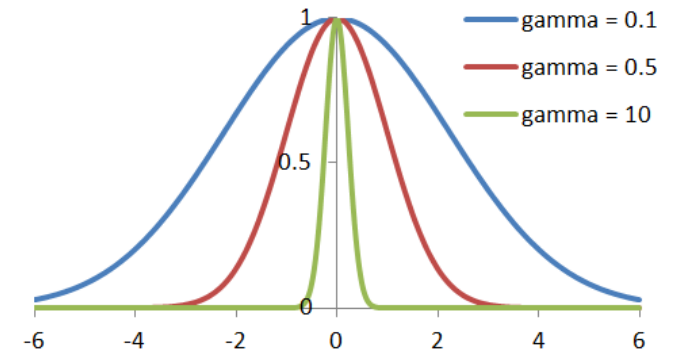
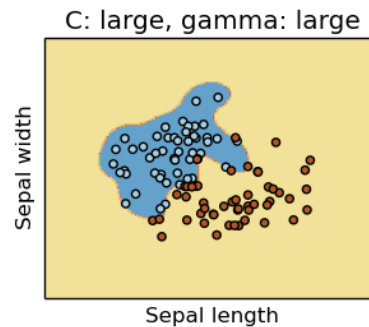
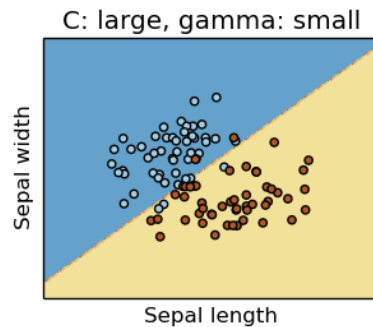
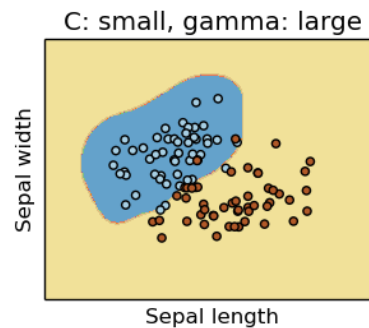
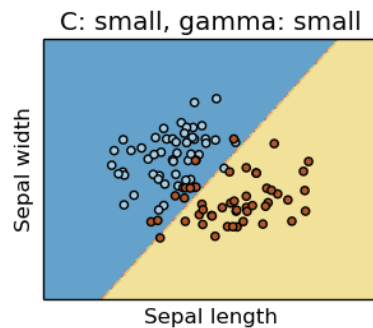
因为数据量较少，所以试利用算全局测地距的ISOMAP进行降维。先用MLE估计了下流形展开到12维降比较合适。然后也试着展开到3维看分类结果怎么样。



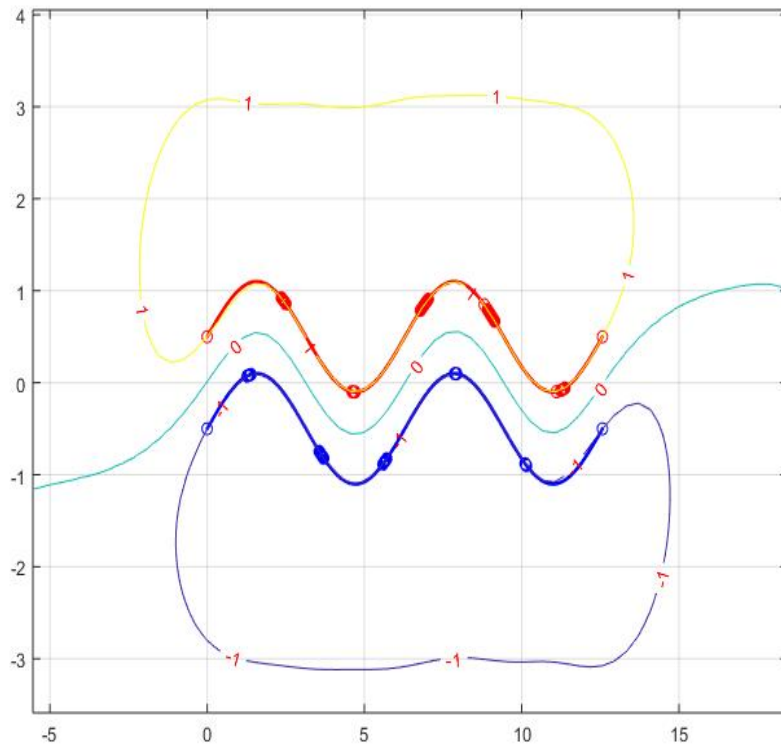
Data Visualization of 2 Dimension Data

Effect of Parameters in RBF-kernel

C & gamma

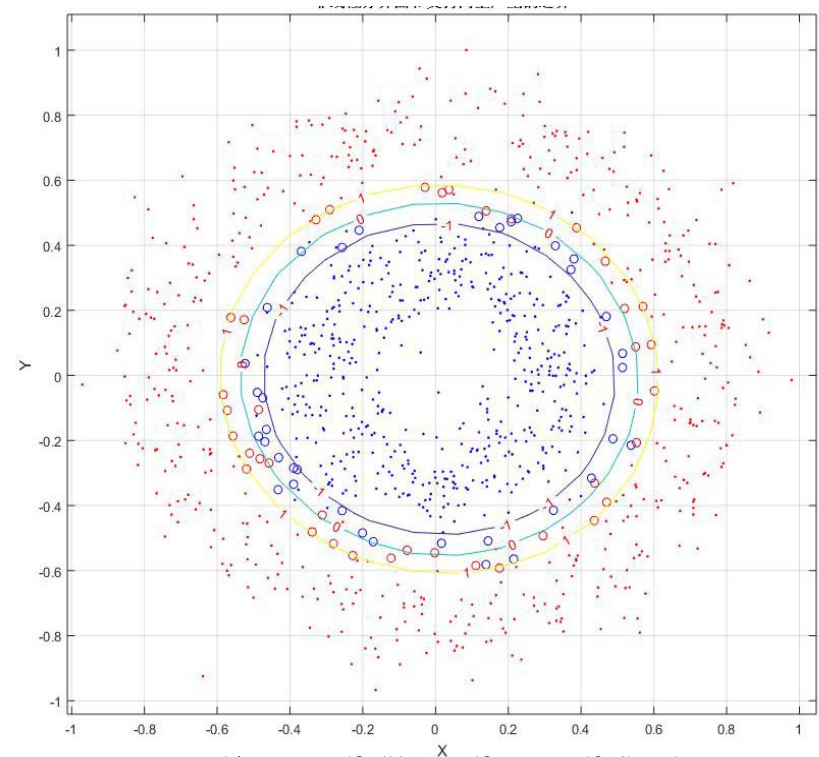


Data Visualization of 2 Dimension Data



$$y_1 = 0.5 \sin x + 0.25$$

$$y_2 = 0.5 \sin x - 0.25$$



$$x_1 = (4 + \text{norm}(0,1)) \cdot \cos(2 \cdot \text{norm}(0,1) \cdot \pi)$$

$$y_1 = (4 + \text{norm}(0,1)) \cdot \sin(2 \cdot \text{norm}(0,1) \cdot \pi)$$

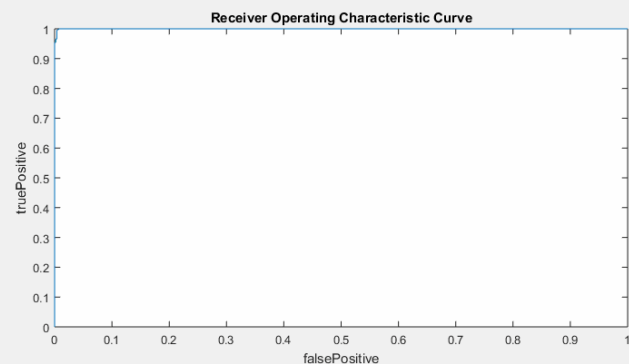
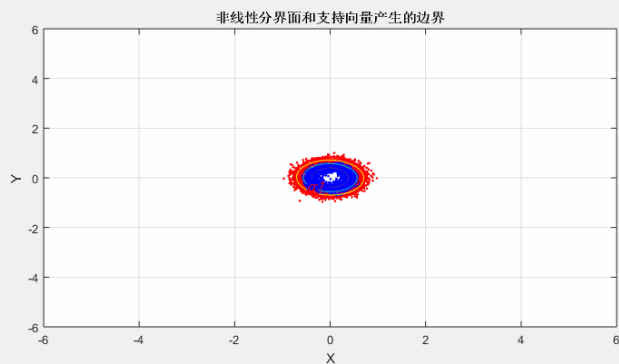
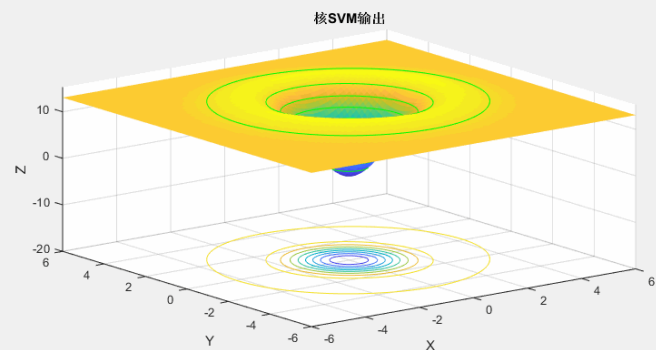
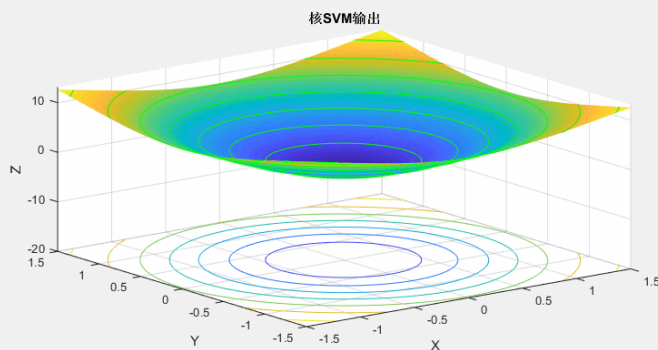
$$x_2 = (9 + \text{norm}(0,1)) \cdot \cos(2 \cdot \text{norm}(0,1) \cdot \pi)$$

$$y_2 = (9 + \text{norm}(0,1)) \cdot \sin(2 \cdot \text{norm}(0,1) \cdot \pi)$$

后面还是几张小动图怎么改变的

仅仅是为了
看看调参对RBF-SVM输出到底改变了些什么

C的变化在Circle数据分类上的效果



$$C=0.5^{(21)} \sim 0.5^{(-19)};$$

C是惩罚系数，即对误差的宽容度。C越高，说明越不能容忍出现误差,容易过拟合。

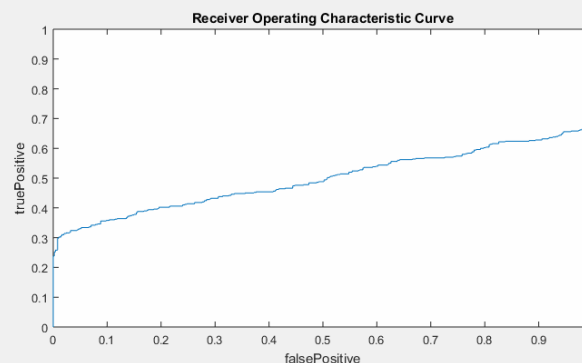
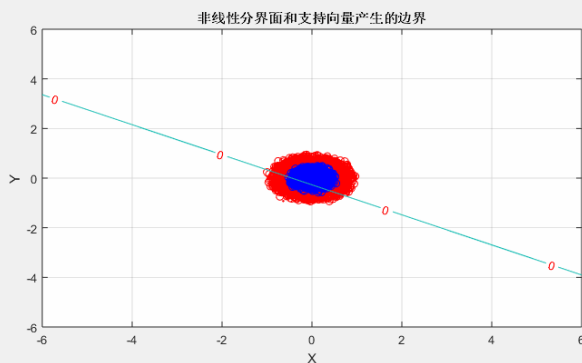
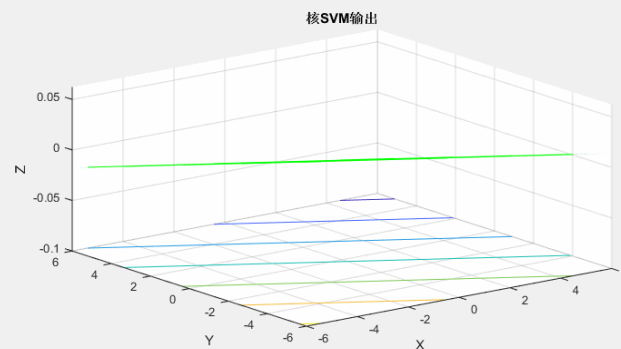
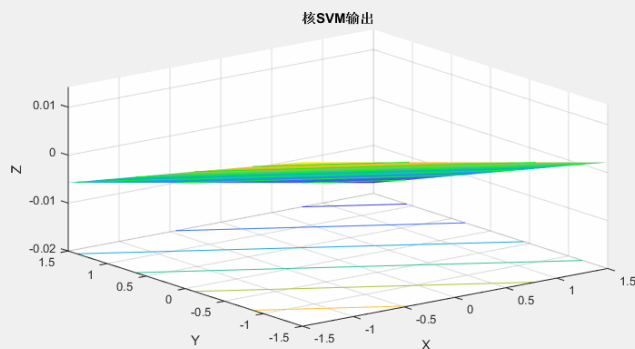
C越小，容易欠拟合。

C过大或过小，泛化能力变差

σ 为类比高斯滤波器的带宽

Gamma的变化

$$k(x,z) = \exp\left(-\frac{d(x,z)^2}{2 \cdot \sigma^2}\right) = \exp(-\text{gamma} \cdot d(x,z)^2) \Rightarrow \text{gamma} = \frac{1}{2 \cdot \sigma^2}$$

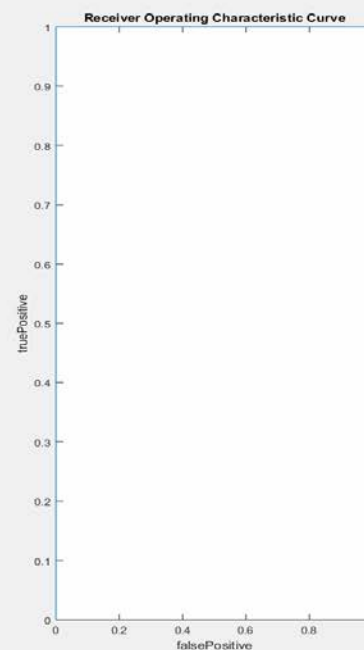
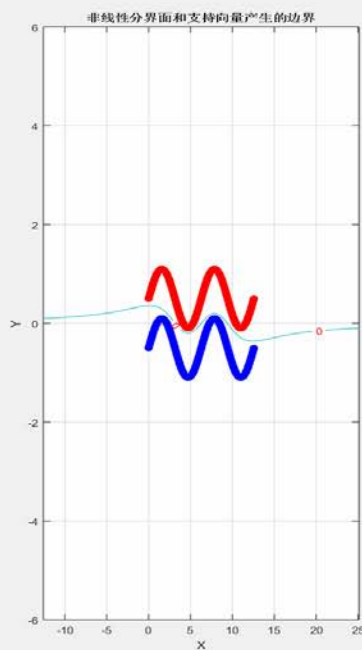
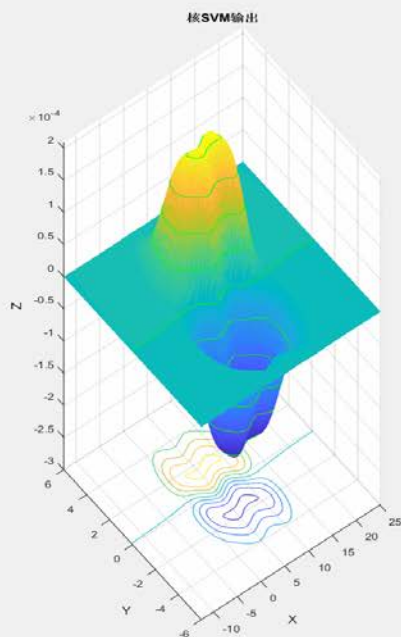


$$\text{Gamma} = 0.5^{(21)} \sim 0.5^{(-19)};$$

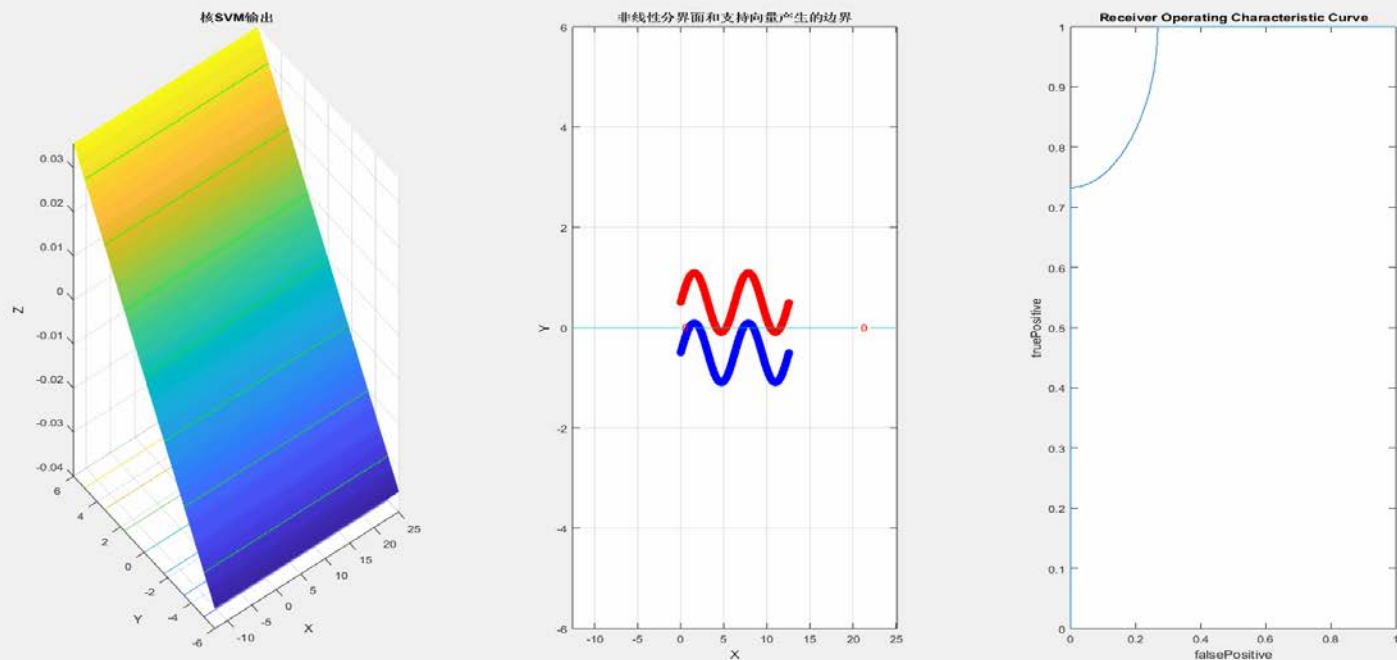
Gamma决定了数据映射到新的特征空间后的分布，gamma越大，支持向量越少，gamma值越小，支持向量越多。

Gamma设的太大， σ 会很小， σ 很小的高斯分布长得又高又瘦，会造成只会作用于支持向量样本附近，对于未知样本分类效果很差，存在训练准确率可以很高，(如果 σ 让无穷小，则理论上，高斯核的SVM可以拟合任何非线性数据，但容易过拟合)而测试准确率不高的可能，就是通常说的过训练；而如果设的过小，则会造成平滑效应太大，无法在训练集上得到特别高的准确率，也会影响测试集的准确率。

C变化在sin数据分类上的效果



Gamma变化在sin数据分类上的效果



从中间的图上可以看到，开始的支持向量数量较少，随着gamma的增大，支持向量开始越来越多。

当gamma很大时所有样本轨迹都变成了支持向量，所以在训练样本上可以达到100%的准确率，但是如果训练样本中没出现过的就会出错，泛化能力变得很差。

XOR 用SVM的分类效果可视化

