实验四　利用 SVM 实现分类实验

实验目标：理解 SVM 的分类原理；

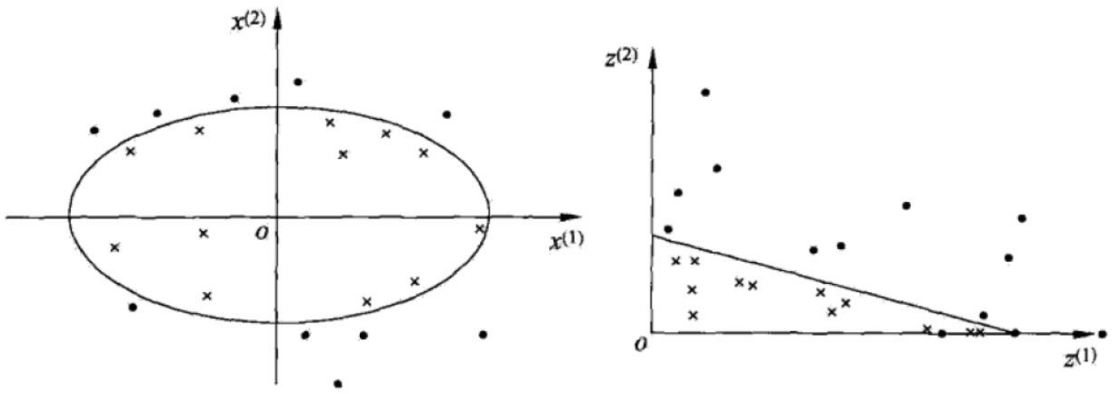
能根据数据集设计合理的 SVM 分类方法；

准确评估分类器精度。

实验工具：LIBSVM，Matlab

实验步骤：

一、SVM 分类原理:

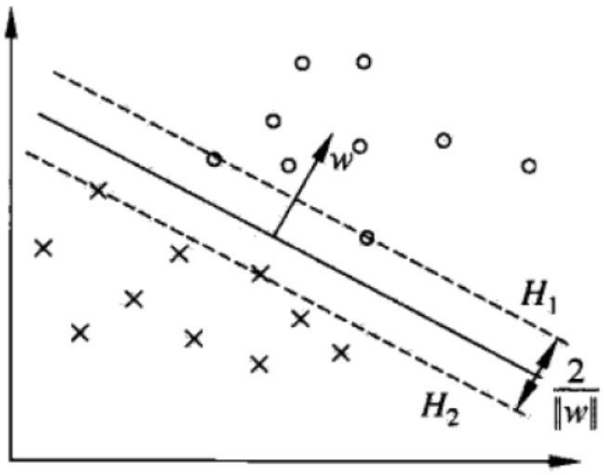


非线性分类，如上图左侧，直线无法（线性模型）将正负例正确分开，需要一条椭圆曲线（非线性模型）。核技巧应用到支持向量机，其基本思想即通过一个非线性变换将输入空间（欧氏空间或离散集合）对应于一个特征空间（希尔伯特空间），使原有的超曲面模型对应于特征空间的超平面模型。

划分超平面可通过线性方程来描述：$g(x) = w^T x + b = 0$

在线性可分的情况下，训练数据集的样本点中与分离超平面距离最近的样本点的实例称为支持向量，如图，H_1 H_2 上的点。支持向量需要满足下面的约束条件：

$$y_i(\omega \cdot x_i + b) - 1 = 0$$



SVM 使分开的两个类别有最大间隔，即分隔面最近的数据点具有最大距离。需要找到两个超平面，与分类平面平行：

H1: y = wTx + b=+1　　H2: y = wTx + b=-1

H1 与 H2 的间隔为 2/||w||，要最大化这个间隔，等价于最小化||w||，进一步最小化 $\frac{1}{2}\|w\|^2$。

假设超平面能将样本正确分类，则 $\begin{cases} w^T x_i + b \geq +1, & y_i = +1 \\ w^T x_i + b \leq -1, & y_i = -1 \end{cases}$，即：$y_i(w^T x_i + b) \geq 1$

为目标函数的约束条件。

对于上述的最优化问题，先要构造拉格朗日函数：

$$\mathscr{L}(\mathbf{w}, b, \boldsymbol{\alpha}) \equiv \frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{i=1}^{N} \alpha_i y_i (\mathbf{w} \cdot \mathbf{x}_i - b)$$

分别对 w 和 b 求导，再代入拉格朗日函数后得到原问题的对偶问题：

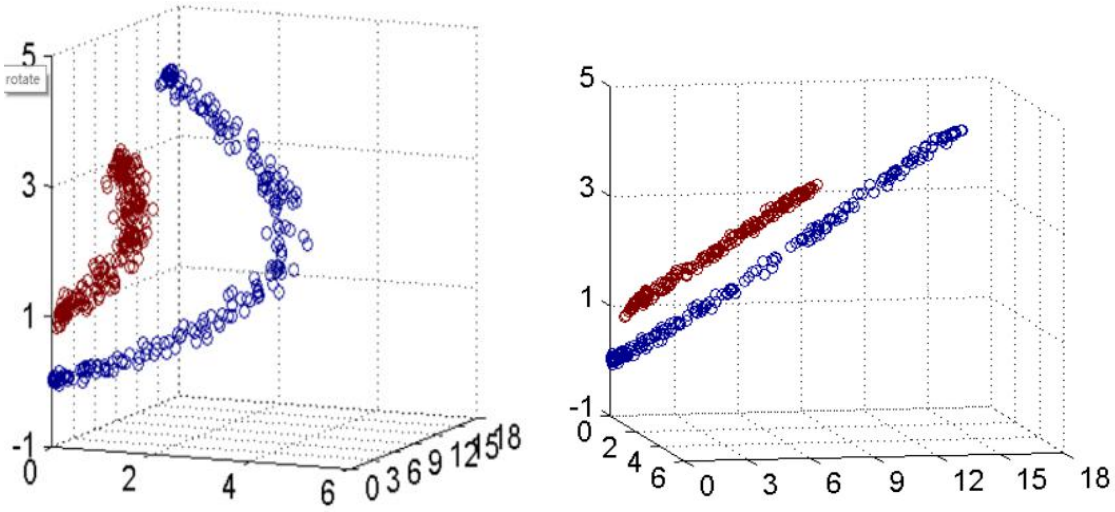
$$\max.\ W(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1,j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

由上面计算得到的 w，带入 g(x)得到：$g(x) = \sum_{i=1}^{N} a_i y_i \langle x_i, x \rangle + b$

要对一个数据点分类是，只需要把待分类的数据点带入 g(x)中，把结果和正负号对比。

对 x 的预测只需要求它与训练点的内积，这是用 kernal 进行线性推广的基本前提。训练点只需要用到支持向量，非支持向量的系数 a 为 0。

对线性不可分的样本集，低维的样本集映射到高维则可以变成线性可分。



寻找一个函数，使得在低维空间中进行计算的结果和映射到高维空间中计算内积<Φ(x-1)，Φ(x2)>的结果相同，避开直接在高维空间中进行计算。分类函数如下：

$$f(x) = \sum_{i=1}^{n} a_i y_i \kappa(x_1, x_2) + b$$

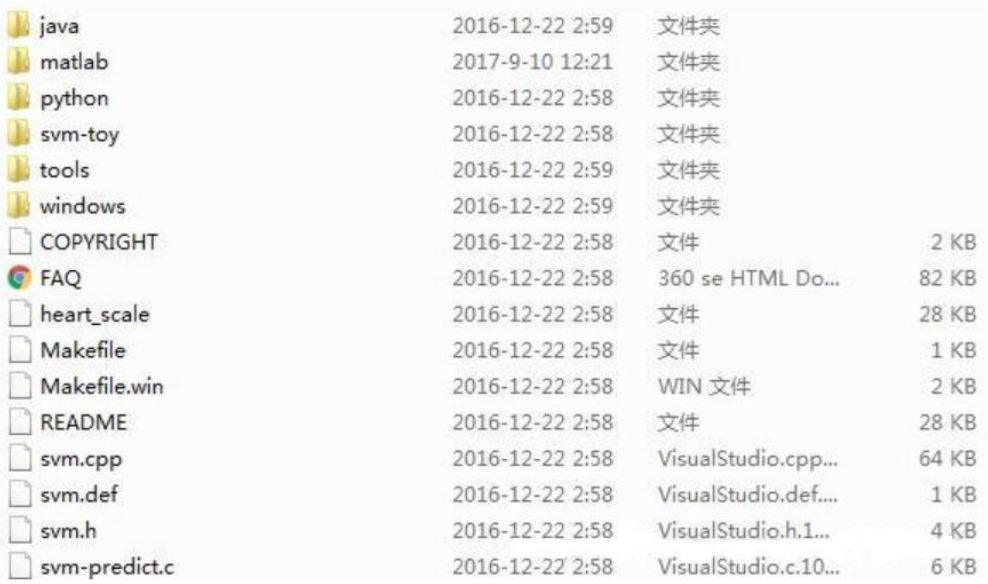
其中 k 就是核函数。

任何将计算表示为数据点内积的方法都可以用核方法进行非线性扩展。

（参考：https://blog.csdn.net/u012581541/article/details/51181041）

- **线性 SVM 分类**

下载 LIBSVM 库 http://www.csie.ntu.edu.tw/~cjlin/libsvm/， 下载后解压后如下图：



java	2016-12-22 2:59	文件夹	
matlab	2017-9-10 12:21	文件夹	
python	2016-12-22 2:58	文件夹	
svm-toy	2016-12-22 2:58	文件夹	
tools	2016-12-22 2:59	文件夹	
windows	2016-12-22 2:59	文件夹	
COPYRIGHT	2016-12-22 2:58	文件	2 KB
FAQ	2016-12-22 2:58	360 se HTML Do...	82 KB
heart_scale	2016-12-22 2:58	文件	28 KB
Makefile	2016-12-22 2:58	文件	1 KB
Makefile.win	2016-12-22 2:58	WIN 文件	2 KB
README	2016-12-22 2:58	文件	28 KB
svm.cpp	2016-12-22 2:58	VisualStudio.cpp...	64 KB
svm.def	2016-12-22 2:58	VisualStudio.def....	1 KB
svm.h	2016-12-22 2:58	VisualStudio.h.1...	4 KB
svm-predict.c	2016-12-22 2:58	VisualStudio.c.10...	6 KB

数据下载地址：

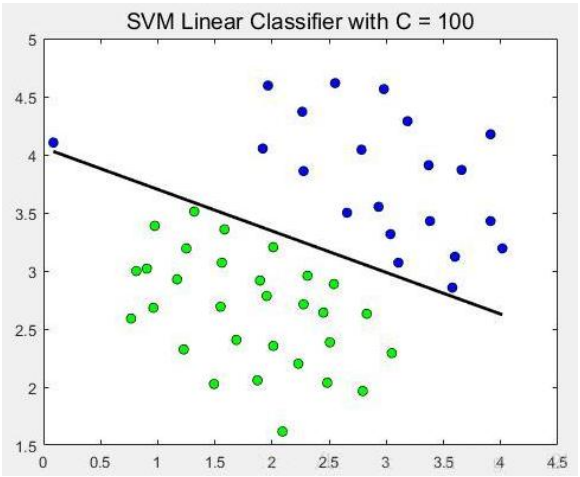
http://openclassroom.stanford.edu/MainFolder/courses/MachineLearning/exercises/ex7materials/ex7Data.zip

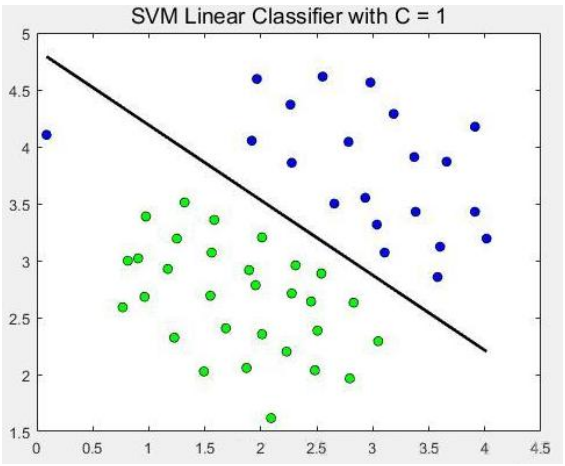
代码如下：

```
% SVM Email text classification

clear all; close all; clc

% Load training features and labels

[train_y, train_x] = libsvmread('email_train-50.txt');

% Train the model and get the primal variables w, b from the model

% Libsvm options

% -t 0 : linear kernel

% Leave other options as their defaults

% model = svmtrain(train_y, train_x, '-t 0');

% w = model.SVs' * model.sv_coef;
```

```
% b = -model.rho;

% if (model.Label(1) == -1)

% w = -w; b = -b;

% end

model = svmtrain(train_y, train_x, sprintf('-s 0 -t 0'));

% Load testing features and labels

[test_y, test_x] = libsvmread('email_test.txt');

[predicted_label, accuracy, decision_values] = svmpredict(test_y, test_x, model);

% After running svmpredict, the accuracy should be printed to the matlab

% console
```

调节 C 可以调节分类面的 Margin，C 越大，Margin 越小正确率也越高





选择不同的训练集的规模来做比较，如 50、100、400 规模的训练集，分别查看准确度。

- **非线性 SVM 分类**

数据集下载：

http://openclassroom.stanford.edu/MainFolder/courses/MachineLearning/exercises/ex

8materials/ex8Data.zip

我们使用的核函数是 RBF（高斯核）

The RBF kernel

In this exercise, you will use the Radial Basis Function (RBF) kernel in LIBSVM. This kernel has the formula

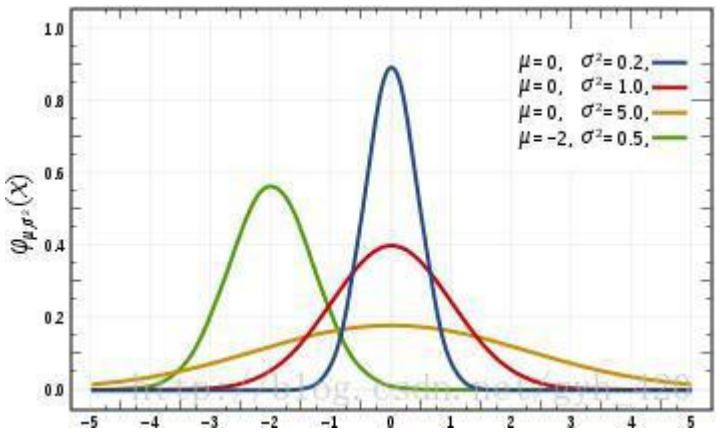
$$\begin{aligned} K\left(x^{(i)}, x^{(j)}\right) &= \phi(x^{(i)})^T \phi(x^{(j)}) \\ &= \exp\left(-\gamma \left\|x^{(i)} - x^{(j)}\right\|^2\right), \qquad \gamma > 0 \end{aligned}$$

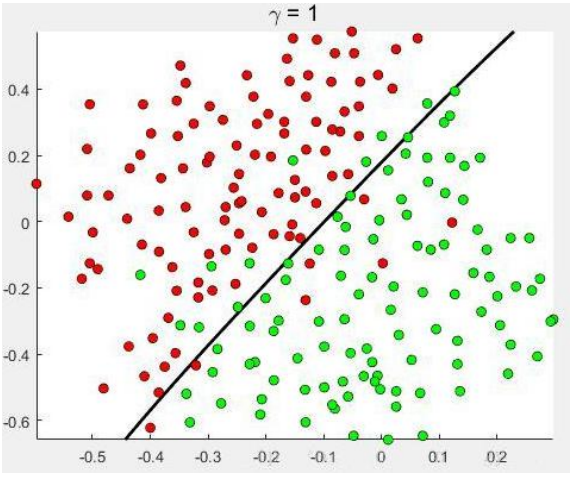
使用 LIBSVM 中的选择高斯核来训练 model，代码如下：

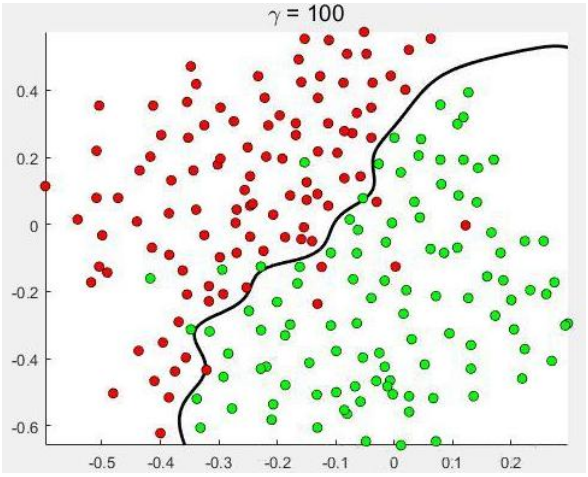
```
clear all; close all; clc

% Load training features and labels[y, x] = libsvmread('ex8a.txt');

gamma = 100;

% Libsvm options

% -s 0 : classification

% -t 2 : RBF kernel

% -g : gamma in the RBF kernel

model = svmtrain(y, x, sprintf('-s 0 -t 2 -g %g', gamma));

% Display training accuracy[predicted_label, accuracy, decision_values] = svmpredict(y, x, model);

% Plot training data and decision boundary

plotboundary(y, x, model);

title(sprintf('\\gamma = %g', gamma), 'FontSize', 14);
```

通过选择合适的γ值来调整 Margin，γ越大越容易过拟合。

$$g(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}.$$







在使用 SVM 时要选择合适的 C 和γ值，思考怎么选择合适的参数？

（参考：

http://openclassroom.stanford.edu/MainFolder/DocumentPage.php?course=MachineLearning&doc=exercises/ex7/ex7.html ）

实验要求：

- 参考以上两段代码，在 Matlab 中，以链接中给出的数据集为例，分别设置合适的 C\γ值，查看分类结果，统计精度。
- 以 sklearn 中的 Iris 数据集为例，用 LIBSVM 结合 Matlab，构建分类器，查看结果。
- 思考怎样设置合适的参数。

实验报告要求：

- 实验结果课上检查。当节课未完成的同学，请下次实验课找我检查。