

作业四

关文聪 2016060601008

1. 下载并安装 libsvm, <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>, 在西瓜数据集 3.0a 上分别用线性核训练一个 SVM。用正类 1-6 和负类 9-14 作为训练集, 其余作为测试集。C 取不同的值, 其它参数设为默认值。作出测试正确率随 C 取值变化的图, $C=[1\ 100\ 10000\ 10^6\ 10^8]$ 。
2. 换成高斯核(宽度设为 1), 重复上题的步骤。

解答:

下载并解压 libsvm 包, 按照文档给定的格式输入数据并保存。

```
52 The format of training and testing data files is:
53
54 <label> <index1>:<value1> <index2>:<value2> ...
55
```

训练集数据 trainingset.txt 如下所示:

测试集数据 testset.txt 如下所示:

```
1 1 1:0.697 2:0.460
2 1 1:0.774 2:0.376
3 1 1:0.634 2:0.264
4 1 1:0.608 2:0.318
5 1 1:0.556 2:0.215
6 1 1:0.403 2:0.237
7 0 1:0.666 2:0.091
8 0 1:0.243 2:0.267
9 0 1:0.245 2:0.057
10 0 1:0.343 2:0.099
11 0 1:0.639 2:0.161
12 0 1:0.657 2:0.198
```

```
1 1 1:0.481 2:0.149
2 1 1:0.437 2:0.211
3 0 1:0.36 2:0.37
4 0 1:0.593 2:0.042
5 0 1:0.719 2:0.103
```

调用 svm-train.exe 进行训练, 参数如下所示:

```
Usage: svm-train [options] training_set_file [model_file]
options:
-s svm_type : set type of SVM (default 0)
  0 -- C-SVC          (multi-class classification)
  1 -- nu-SVC         (multi-class classification)
  2 -- one-class SVM
  3 -- epsilon-SVR    (regression)
  4 -- nu-SVR         (regression)
-t kernel_type : set type of kernel function (default 2)
  0 -- linear: u'*v
  1 -- polynomial: (gamma*u'*v + coef0)^degree
  2 -- radial basis function: exp(-gamma*|u-v|^2)
  3 -- sigmoid: tanh(gamma*u'*v + coef0)
  4 -- precomputed kernel (kernel values in training_set_file)
-d degree : set degree in kernel function (default 3)
-g gamma : set gamma in kernel function (default 1/num_features)
-r coef0 : set coef0 in kernel function (default 0)
-c cost : set the parameter C of C-SVC, epsilon-SVR, and nu-SVR (default 1)
-nu nu : set the parameter nu of nu-SVC, one-class SVM, and nu-SVR (default 0.5)
-p epsilon : set the epsilon in loss function of epsilon-SVR (default 0.1)
-m cachesize : set cache memory size in MB (default 100)
-e epsilon : set tolerance of termination criterion (default 0.001)
-h shrinking : whether to use the shrinking heuristics, 0 or 1 (default 1)
-b probability_estimates : whether to train a SVC or SVR model for probability estimates, 0 or 1 (default 0)
-wi weight : set the parameter C of class i to weight*C, for C-SVC (default 1)
-v n: n-fold cross validation mode
-q : quiet mode (no outputs)
```

使用线性核训练，指定核为线性核，指定参数 C 的值，进行训练得到模型。调用 `svm-predict.exe`，使用训练出的支持向量机模型对测试集进行测试，得到正确率。

```
C:\Users\Eternity-Myth\Desktop>svm-train.exe -t 0 -c 1 trainingset.txt trainingmodel
*
optimization finished, #iter = 6
mu = 1.000000
obj = -11.116675, rho = 0.671734
nSV = 12, nBSV = 12
Total nSV = 12

C:\Users\Eternity-Myth\Desktop>svm-predict.exe testset.txt trainingmodel output
Accuracy = 20% (1/5) (classification)

C:\Users\Eternity-Myth\Desktop>svm-train.exe -t 0 -c 100 trainingset.txt trainingmodel
.*
optimization finished, #iter = 22
mu = 0.502573
obj = -546.130886, rho = 4.627020
nSV = 8, nBSV = 5
Total nSV = 8

C:\Users\Eternity-Myth\Desktop>svm-predict.exe testset.txt trainingmodel output
Accuracy = 40% (2/5) (classification)

C:\Users\Eternity-Myth\Desktop>svm-train.exe -t 0 -c 10000 trainingset.txt trainingmodel
.....*
optimization finished, #iter = 395
mu = 0.333333
obj = -39636.727854, rho = 9.653994
nSV = 5, nBSV = 3
Total nSV = 5

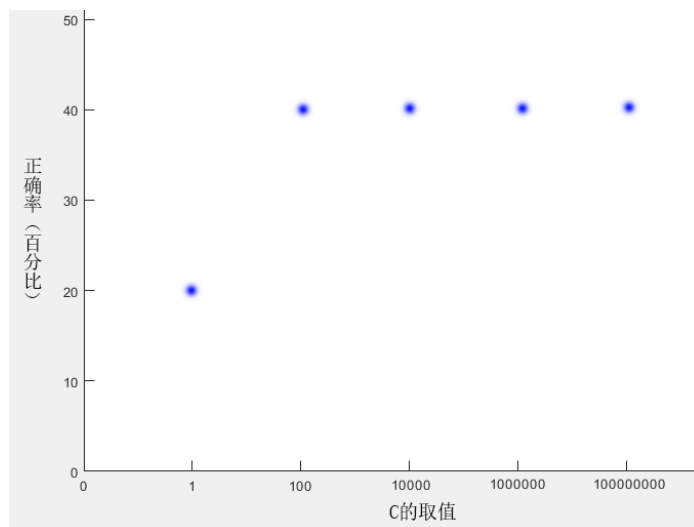
C:\Users\Eternity-Myth\Desktop>svm-predict.exe testset.txt trainingmodel output
Accuracy = 40% (2/5) (classification)

optimization finished, #iter = 38211
mu = 0.319038
obj = -3834778.699307, rho = 21.723516
nSV = 5, nBSV = 2
Total nSV = 5

C:\Users\Eternity-Myth\Desktop>svm-predict.exe testset.txt trainingmodel output
Accuracy = 40% (2/5) (classification)

optimization finished, #iter = 5158711
mu = 0.301188
obj = -469983343.758039, rho = 11.052479
nSV = 6, nBSV = 2
Total nSV = 6

C:\Users\Eternity-Myth\Desktop>svm-predict.exe testset.txt trainingmodel output
Accuracy = 40% (2/5) (classification)
```



正确率随 C 变化的图像如图：

再使用高斯核函数，用同样的方法训练并测试，结果如下：

```
C:\Users\Eternity-Myth\Desktop>svm-train.exe -t 2 -c 1 trainingset.txt trainingmodel
*
optimization finished, #iter = 6
nu = 1.000000
obj = -11.176997, rho = 0.034297
nSV = 12, nBSV = 12
Total nSV = 12

C:\Users\Eternity-Myth\Desktop>svm-predict.exe testset.txt trainingmodel output
Accuracy = 20% (1/5) (classification)

C:\Users\Eternity-Myth\Desktop>svm-train.exe -t 2 -c 100 trainingset.txt trainingmodel
*
optimization finished, #iter = 14
nu = 0.500000
obj = -521.674270, rho = 4.200510
nSV = 7, nBSV = 5
Total nSV = 7

C:\Users\Eternity-Myth\Desktop>svm-predict.exe testset.txt trainingmodel output
Accuracy = 40% (2/5) (classification)

C:\Users\Eternity-Myth\Desktop>svm-train.exe -t 2 -c 10000 trainingset.txt trainingmodel
.....*.....*
optimization finished, #iter = 269
nu = 0.049993
obj = -3000.157602, rho = 38.700830
nSV = 4, nBSV = 0
Total nSV = 4

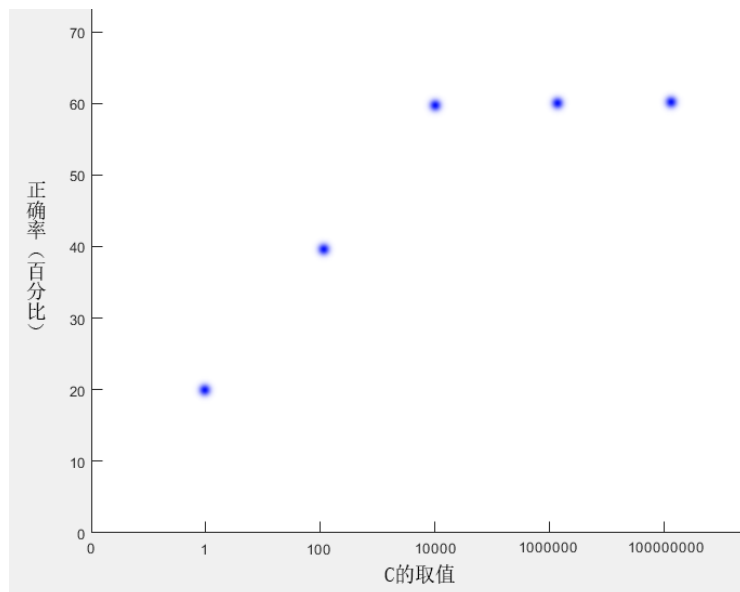
C:\Users\Eternity-Myth\Desktop>svm-predict.exe testset.txt trainingmodel output
Accuracy = 60% (3/5) (classification)

C:\Users\Eternity-Myth\Desktop>svm-train.exe -t 2 -c 1000000 trainingset.txt trainingmodel
.....*.....*
optimization finished, #iter = 269
nu = 0.000500
obj = -3000.157602, rho = 38.700830
nSV = 4, nBSV = 0
Total nSV = 4

C:\Users\Eternity-Myth\Desktop>svm-predict.exe testset.txt trainingmodel output
Accuracy = 60% (3/5) (classification)

C:\Users\Eternity-Myth\Desktop>svm-train.exe -t 2 -c 100000000 trainingset.txt trainingmodel
.....*.....*
optimization finished, #iter = 269
nu = 0.000005
obj = -3000.157602, rho = 38.700830
nSV = 4, nBSV = 0
Total nSV = 4

C:\Users\Eternity-Myth\Desktop>svm-predict.exe testset.txt trainingmodel output
Accuracy = 60% (3/5) (classification)
```



正确率随 C 变化的图像如图：