CSCI964 Computational Intelligence: Assignment #1

Mei Wangzhihui 2019124044

Task 1

Data1

Test Data
We set the latter half of the data as test data.
Classifier
The **two-spiral problem** is a two class problem, which is non-linear. So the kernel function must be non-linear. As the feature number is far smaller than number of sample. In this dataset, I applied RBF kernel C-SVC SVM.
Parameter
There are 2 critical parameters:

1. C: the cost function factor. If C is too little, it is likely to underfit, if C is too large, it is likely to overfit. we set C to 2048 to get better generalization.

2. *gamma*: *gamma* is a parameter that comes with this function after selecting the RBF function as the kernel. It implicitly determines the distribution of the data after it is mapped to a new feature space. The gamma decreases, and the fewer support vectors, the smaller the gamma value, the more support vectors. The number of support vectors affects the speed of training and prediction.

Accuracy
97.92%
Printout

```
param.svm_type = C_SVC; // -s svm类型: SVM设置类型(默认0)0 -- C-SVC 1 --v-SVC
                        // 2 - 一类SVM 3 -- e-SVR 4 -- v-SVR
param.kernel_type = RBF;
//-t 核函数类型: 核函数设置类型(默认2) 0 - 线性: u'v 1 -
//多项式: (r*u'v + coef0)^degree 2 - RBF函数: exp(-gamma|u-v|^2) 3
//-sigmoid: tanh(r*u'v + coef0)
param.degree = 3;       You, 3 days ago • SVM code basic
//    -d degree: 核函数中的degree设置(针对多项式核函数)(默认3)

param.gamma = 8.0;
// -g
// r(gama): 核函数中的gamma函数设置(针对多项式/rbf/sigmoid核函数)(默认1/n_features)

param.coef0 =
    0; //    -r coef0: 核函数中的coef0设置(针对多项式/sigmoid核函数)((默认0)

param.nu = 0.5; //    -n nu: 设置v-SVC, 一类SVM和v- SVR的参数(默认0.5)

param.cache_size =
    100; //    -m cachesize: 设置cache内存大小, 以MB为单位(默认40)

param.C = 2048; //    -c cost: 设置C-SVC, e -SVR和v-SVR的参数(损失函数)(默认1)

param.eps = 1e-3; //    -e eps: 设置允许的终止判据(默认0.001)

param.p = 0.1; //    -p p: 设置e -SVR 中损失函数p的值(默认0.1)

param.shrinking = 1; //    -h shrinking: 是否使用启发式, 0或1(默认1)

param.probability =
    0; //-b 概率估计: 是否计算SVC或SVR的概率估计, 可选值0 或1, 默认0;
```

Figure 1: two-spiral problem-parameter

```
optimization finished, #iter = 832068
nu = 0.179704
obj = -43223.748971, rho = -0.000264
nSV = 136, nBSV = 16
Total nSV = 136
acc=0.979167
```

Figure 2: two-spiral problem-output

Data2

Test Data
We set the former 3677 samples as training set, the latter 500 samples as test set.
Classifier
The **Abalone Age Problem** is regression problem. In this dataset, I applied Epsilon-SVR.
Parameter
SVM Type: Epsilon-SVR
Kernel: RBF
Cost: 4
C is the penalty coefficient, that is, tolerance for errors. The higher the c, the more intolerable the error is, and it is easy to overfit. The smaller C, the easier to underfit. C is too large or too small, the generalization ability becomes poor
gamma: 1
Gamma is a parameter that comes with this function after selecting the RBF function as the kernel. It implicitly determines the distribution of the data after mapping into a new feature space. For gamma replacement, the fewer support vectors, the smaller the gamma value and the more support vectors. The number of support vectors affects the speed of training and prediction.

```
param.svm_type = EPSILON_SVR; // -s svm类型: SVM设置类型(默认0)0
-- C-SVC 1
                              // --v-SVC 2 - 一类SVM 3 -- e-SVR
                              4 -- v-SVR

param.C = 4; //    -c cost: 设置C-SVC, e -SVR和v-SVR的参数(损失函
数)(默认1
param.gamma = 1;
// -g
// r(gama): 核函数中的gamma函数设置(针对多项式/rbf/sigmoid核函数)
(默认1/n_features)

param.kernel_type = RBF;
//-t 核函数类型: 核函数设置类型(默认2) 0 - 线性: u'v 1 -
//多项式: (r*u'v + coef0)^degree 2 - RBF函数: exp(-gamma|u-v|^2)
3
//-sigmoid: tanh(r*u'v + coef0)
param.degree = 3;
//    -d degree: 核函数中的degree设置(针对多项式核函数)(默认3)

param.gamma = 1;
// -g
// r(gama): 核函数中的gamma函数设置(针对多项式/rbf/sigmoid核函数)
(默认1/n_features)
```

Figure 3: Abalone Age Problem parameter

Accuracy acc=0.706
Output

```
optimization finished, #iter = 8370
nu = 0.986354
obj = -1114.853448, rho = -0.817244
nSV = 3677, nBSV = 3580
index:0 predict: 0.609684 actual: 0.578947
index:1 predict: 0.629204 actual: 0.526316
index:2 predict: 0.499436 actual: 0.526316
```

Figure 4: Abalone Age Problem parameter

```
index:497 predict: 0.542221 actual: 0.473684
index:498 predict: 0.484455 actual: 0.526316
index:499 predict: 0.603441 actual: 0.631579
acc=0.706
```

Figure 5: Abalone Age Problem parameter

Tuning

It should be noted that the physical meaning of gamma. We mentioned that the width of many RBFs will affect the range of Gaussian corresponding to each support vector, thereby affecting the generalization performance. My understanding: If the gamma is set too large, the variance will be small, and the Gaussian distribution with small variance will be tall and thin, which will only act near the support vector samples, and it has a poor classification effect for unknown samples. There is training The accuracy rate can be very high, (if the variance is infinitesimally small, theoretically, the SVM of the Gaussian kernel can fit any nonlinear data, but it is easy to overfit) and the possibility that the test accuracy rate is not high is usually overtraining; If the setting is too small, the smoothing effect will be too large to obtain a particularly high accuracy rate on the training set, and it will also affect the accuracy rate of the test set.

Data3

Test Data

There are 349 samples, first 299 as training data, last 50 as training data.

Classifier

C-SVC and RBF as kernel function. The penalty coefficient C is the coefficient of the slack variable that we talked about in the previous chapter. In the optimization function, it mainly balances the relationship between the complexity of the model and the misclassification rate, which can be understood as the regularization coefficient. When C is larger, our loss function will also be larger, which means that we are reluctant to give up farther outliers. In this way, we will have fewer support vectors, which means that the support vector and hyperplane models will become more complicated and easy to overfit. On the contrary, when C is relatively small, it means that we do n't want to deal with those outliers and will choose more samples as support vectors. The final support vectors and hyperplane models will also be simple.

Another hyperparameter is the parameter gamma of the RBF kernel function. gamma mainly defines the influence of a single sample on the entire classification hyperplane. When gamma is relatively small, the influence distance of a single sample on the entire classification hyperplane is relatively long, and it is easy to be selected as a support vector. On the contrary, when gamma is relatively large, a single sample affects the entire The influence distance of the classification hyperplane is relatively short, and it is not easy to be selected as the support vector, or the support vector of the entire model will be less, and the model will become more complicated.

Parameter

C = 2

gamma = 2
PrintOut

```
optimization finished, #iter = 805
nu = 0.289092
obj = -86.459119, rho = -0.571984
nSV = 257, nBSV = 0
Total nSV = 257
```

Figure 6: SPECT Heart Diagnosis Problem

```
index:47 predict: 1 actual: 1
index:48 predict: 0 actual: 0
index:49 predict: 0 actual: 0
acc=1
```

Figure 7: SPECT Heart Diagnosis Problem

Task 2

Steps

1. Initialize various variables: read datafile, set parameter
2. Initialize population: gene is the visiting order, each individual is one visiting order.
3. Implement Fitness function: $f = 1/distance$
4. Implement Select function: Roulette
5. Implement Crossover function: crossover visiting order by Probability P_m, keep the first and the last one city, reorder the middle ones.
6. Implement Mutation function: Randomly mutate 2 point of the visiting order
7. Implement main loop

Detail

1. Encoding method: The general encoding methods are binary encoding method, Gray code encoding method, floating point encoding method, symbol encoding method, and real number encoding used directly this time
2. Group composition: Elite retention There are four ways to form a new generation group composition method:
    - N: all-new-generation inter-generational updates that replace the previous generation groups

- G: update the partial update method of some individuals in the group according to a certain proportion (or generation gap method, the extreme of this case is to delete only the worst way of death for each uncomfortable individual).
- E: The best retention (elitist) group construction method that keeps a best parent string. Method.
- B: Select the best group formation form of the individual from the children and parents. Generally speaking, the global search performance of N mode is the best, but the convergence speed is the slowest; the convergence speed of B mode is the fastest, but the global search performance is the worst; the performance of E mode and G mode is between N mode and B mode. In the application of the solution to the salesman problem, the E method is mostly used.

For performance considerations, the E method was adopted in this comprehensive training. In the selection of the first generation, the selection crossover and mutation of each generation used the ideas retained by the elite.

How can we allow more elite individuals to survive, and still maintain the richness of the guaranteed species, thereby reducing more algorithm running time? We first divide the group into two different groups, that is, to survive in two regions with different environments to form geographical isolation. The purpose of this is to increase the richness of the population. After that, we changed the living environment of two different groups into evil strategies (that is, two different and more demanding and complex selection criteria). The purpose of this is to allow more elite individuals to survive. When the two populations have multiplied to a certain standard, they will be merged into a single population, and eventually a global optimal solution will be generated more quickly through a certain standard.

Train

```
step=258000
best path:
    0   96   48   19   54   89   77   80   63   99   41   94   91   45   58   55   47   81   53   22
    60   59   28   62   65   70   46   39   35   75   42   82   27   98   36   93    5   56   31   66
    85    1   38   25   86   52   74   17   14   26   71   61   37   57   16   40   73   43   24   33
    92   88   64   20   10   34   83   11   69    2   51   30   29   49   13   90   23    8   84    3
    78   21   87   79   97   18   50   32   67    9    6   95    7   72   15   76   44    4   68   12    0
best path cost: 6386.973983
step=260000
best path:
    0   96   48   19   54   89   77   80   63   99   41   94   91   45   58   55   47   81   53   22
    60   59   28   62   65   70   46   39   35   75   42   82   27   98   36   93    5   56   31   66
    85    1   38   25   86   52   74   17   14   26   71   61   37   57   16   40   73   43   24   33
    92   88   64   20   10   34   83   11   69    2   51   30   29   49   13   90   23    8   84    3
    78   21   87   79   97   18   50   32   67    9    6   95    7   72   15   76   44    4   68   12    0
best path cost: 6386.973983
step=262000
best path:
    0   96   48   19   54   89   77   80   63   99   41   94   91   45   58   55   47   81   53   22
    60   59   28   62   65   70   46   39   35   75   42   82   27   98   36   93    5   56   31   66
    85    1   38   25   86   52   74   17   14   26   71   61   37   57   16   40   73   43   24   33
    92   88   64   20   10   34   83   11   69    2   51   30   29   49   13   90   23    8   84    3
    78   21   87   79   97   18   50   32   67    9    6   95    7   72   15   76   44    4   68   12    0
best path cost: 6386.973983
```

Figure 8: The Training Outcome (100 citys)

```
step=150000
best path:
   0  47  30 115  86   1  69  49 161  43  38 103 138 198 144 126   9  42 137  12
  28  58 149 111 158 183  37 125 162   5 166 164  80 117   4   7 185  87 184  24
  36   2 123  11 188 109 135  40 119 133  90 104 182  94 163 134 116  26 110  33
 120 140 139  83 141  70  73 191 101 100  50 129  66 121 136  71  93  68  19  84
  21  59  55 130 197  82  18 179  45  44 165  91 192 174 124 195 146  61  64 108
 170 196  65 159  34 193  16  99  41  10  48  22  81 113 157 160 118  78  25  14
  56  13 180 154 142   8  35 106  52 153 194 148  27  54  79 112 128  15 107 151
 145 178 199 190 114 156   6 173  92 122  88  72 189  95 150 105 175 176 143  96
  31  17   3 177 147  39 172 181 132  46  97  32  75  57  67 186  74 171  51 167
  29 102  85 127 187  23  89  62  76 155 169  77  63 168  53 152  60  20 131  98   0
best path cost: 9844.306903
step=152000
best path:
   0  47  30 115  86   1  69  49 161  43  38 103 138 198 144 126   9  42 137  12
  28  58 149 111 158 183  37 125 162   5 166 164  80 117   4   7 185  87 184  24
  36   2 123  11 188 109 135  40 119 133  90 104 182  94 163 134 116  26 110  33
 120 140 139  83 141  70  73 191 101 100  50 129  66 121 136  71  93  68  19  84
  21  59  55 130 197  82  18 179  45  44 165  91 192 174 124 195 146  61  64 108
 170 196  65 159  34 193  16  99  41  10  48  22  81 113 157 160 118  78  25  14
  56  13 180 154 142   8  35 106  52 153 194 148  27  54  79 112 128  15 107 151
 145 178 199 190 114 156   6 173  92 122  88  72 189  95 150 105 175 176 143  96
  31  17   3 177 147  39 172 181 132  46  97  32  75  57  67 186  74 171  51 167
  29 102  85 127 187  23  89  62  76 155 169  77  63 168  53 152  60  20 131  98   0
best path cost: 9844.306903
```

Figure 9: The Training Outcome (200 citys)

```
step=258000
best path:
   0  96  48  19  54  89  77  80  63  99  41  94  91  45  58  55  47  81  53  22
  60  59  28  62  65  70  46  39  35  75  42  82  27  98  36  93   5  56  31  66
  85   1  38  25  86  52  74  17  14  26  71  61  37  57  16  40  73  43  24  33
  92  88  64  20  10  34  83  11  69   2  51  30  29  49  13  90  23   8  84   3
  78  21  87  79  97  18  50  32  67   9   6  95   7  72  15  76  44   4  68  12   0
best path cost: 6386.973983
step=260000
best path:
   0  96  48  19  54  89  77  80  63  99  41  94  91  45  58  55  47  81  53  22
  60  59  28  62  65  70  46  39  35  75  42  82  27  98  36  93   5  56  31  66
  85   1  38  25  86  52  74  17  14  26  71  61  37  57  16  40  73  43  24  33
  92  88  64  20  10  34  83  11  69   2  51  30  29  49  13  90  23   8  84   3
  78  21  87  79  97  18  50  32  67   9   6  95   7  72  15  76  44   4  68  12   0
best path cost: 6386.973983
step=262000
best path:
   0  96  48  19  54  89  77  80  63  99  41  94  91  45  58  55  47  81  53  22
  60  59  28  62  65  70  46  39  35  75  42  82  27  98  36  93   5  56  31  66
  85   1  38  25  86  52  74  17  14  26  71  61  37  57  16  40  73  43  24  33
  92  88  64  20  10  34  83  11  69   2  51  30  29  49  13  90  23   8  84   3
  78  21  87  79  97  18  50  32  67   9   6  95   7  72  15  76  44   4  68  12   0
best path cost: 6386.973983
```

Figure 10: The Training Outcome (300 citys)

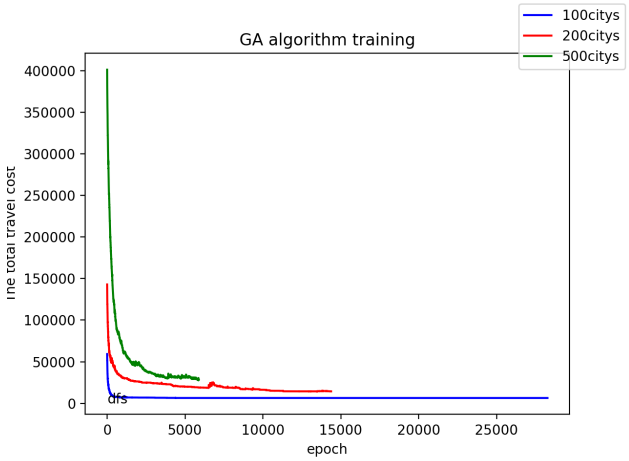

Figure 11: The Training Convergence