SVM Project

Kai Lin 11610205
School of Computer Science and Engineering
Southern University of Science and Technology
Email: 11610205@mail.sustc.edu.cn

1. Preliminaries

1.1. Problem Description

In machine learning,support vector machines(SVM) are supervised learning models that using algorithm to analyze data used for classication and regression analysis.In this project we are aim to take a ten domain data for train and try to analyze the new test data for two category 1 and -1 that is the problem we need to slave.[1]

1.2. Problem Application

SVM can slave many difficult problems. In this project we are aim to train for two category. So I use two way that SMO and Pegasos to try and solve this problem.

2. Methodology

2.1. Notation

- **Pegasos**
  - dataSet: the train data .
  - labels: the result of train data .
  - lam: the parameter for gradient calculation .
- **SMO**
  - dataSet: the train data .
  - labels: the result of train data .
  - C: the tolerance for data.
  - toler: the tolerance for result.

2.2. Software and data structure

In the project, I used pycharm IDE and python 3.6 to implement my project, in addition, the Numpy is the library I used in the project. In programming,I main use mat to calculate for the matrix.

2.3. Model design

2.3.1. Formulate the problem.

For SVM has many ways to solve in this project I try pegasos and SMO two method.

2.3.2. Method to solve the problem.

For Pegasos,each loop I random for a sample to gradient descent,and for certain number of loops,then get the nal result .[2]

For SMO ,each loop I choose a pair of alpha and update alpha matrix until the iteration times enough or the model convergence .

2.3.3. Architecture.

Here are the important functions in Python file Pegasos with pesudocode format.

- **Pegasos**: Each loop I random for a sample to gradient descent,and for certain number of loops,then get the nal result .

Then is other important functions in Python file SMO with pesudocode format.

- **SMO**: Each loop I choose a pair of alpha and update alpha matrix until the iteration times enough or the model convergence .

- **selectJ**: This is the second choice for the max change of alpha

- **updateEk**: Get the error and save

- **innerLoop**: The SMO inner loop for calculate new alpha and updata

2.3.4. Detail of algorithm.

First talk the Pegasos the main way is to get a good value for random gradient descent

Algorithm 1 Pegasos: One method for SVM.

```
Input: $mat$ dataSet ,$mat$ labels,$int$ lam,$int$Tl
Output: $array$ w
 1: function PEGASOS($dataSet, labels, lam, T$)
 2:     $m, n \leftarrow shape(dataSet)$
 3:     $w \leftarrow zeros(n)$
 4:     for t in range(T+1) do
 5:         $i \leftarrow random(m)$
 6:         $eta \leftarrow 1/(lam * t)$
 7:         $p \leftarrow w * dataSet[i]$
 8:         if labels[i]*p¡1 then
 9:             w=(1-1/t)*t+
10:             eta*labels[i]*dataSet[i]
11:         end if
12:         if labels[i]*p¿=1 then
13:             w=(1-1/t)*t
14:         end if
15:     end for
16:     return $w$
17: end function
```

Second talk the SMO the main way is to get a good hyperplane to divide to two part.

Algorithm 2 SMO: One method for SVM.

```
Input: $mat$ dataSet ,$mat$ labels,$float$ C,$float$ toler,$int$
    maxIterl
Output: $int$ b,$array$ alphas
 1: function SMO($dataSet, labels, C, toler, maxIter$)
 2:     $iter \leftarrow 0$
 3:     $entireSet \leftarrow 1$
 4:     $alph\_c \leftarrow 0$
 5:     for $iter < maxIter$ do
 6:         if $alph\_c <= 0$ then
 7:             break
 8:         end if
 9:         if entireSet then:
10:             for i in range(len(dataSte[0])) do
11:                 alph_c+=innerL(i,dataSet,labels)
12:             end for
13:         end if
14:
15:         if !entireSet then:
16:             $ran \leftarrow alphas.A > 0 * alphas.A < c[0]$
17:             for i in range(ran) do
18:                 alph_c+=innerL(i,dataSet,labels)
19:             end for
20:         end if
21:         entireSet=!entireSet
22:         return $b, alphas$
23:
```

3. Empirical Verification

3.1. Data set

I set the train_data.txt for the mainly use dataset.For the one train_data that I take 9 and 1 divide for train and test .For whole data I random choice 90 percent by random in train model and other 10 percent is to test the model

3.2. Test environment

The test environment for this experiment is pycharm community version 2018.2, python version is 3.70, then, the hardware part is, the host is Intel(R) Core i5-7300HQ 2.50GHz only one number of Process.

3.3. Performance measure

For take data to random get 90 percent train model and other 10 test for Pegasos is faster then SMO.

For pegasos take one time use about 0.6s and the accuracy rate isabout 97.86%

For SMO take one time use about 71s and the accuracy rate isabout 90.24%

3.4. Hype-parameters

- **Pegasos**
  - lam: 2.0 .
  - loop: 1000 .
- **SMO**
  - C :0.6
  - toler :0.001

4. Conclusion

4.1. Result

For my train_data I find it is two likely shapes as two Ball type that one over another that 1 and -1 can get much difference . For this type data use pegasos can get better accuracy .We can find that pegasos can do better in split for data that have clear divide . What's more the Pegasos can use less time . But as teacher said the SMO can do better for data that hard to divide and many domains.

4.2. Analysis

For divide data by SVM the difficult points is the speed and accuracy .But for different data the Pegasos and SMO can do better in on part. So maybe no a best algorithm . What we should do is use right algorithm in right time

5. Conclusion

I would like to thank my teacher Yao Zhao who wish to teach me more and more useful konwledge whatever study and life . Read and discuss the parper which the hard to understand part and hard lesson preparation .Last I would like to thank SA who will check my codes and reports.

References

[1] understand SVM(one)_ realize SVM and write (2018).Retrieved from https://blog.csdn.net/puqutogether/ article/details/39894835

[2] python write SMO algorithm_(2018)Retrieved from https://blog.csdn.net/qq_27015119/article/details/80816975