HW2

Machine Learning, 2016 Fall

R05921075, 電機一, 鄭凱文

● Method1: Logistic Regression Function

1. Read train data

There are 4001 57-dimension features.

2. Normalize train data

Normalize each element to normal Gaussian distribution(μ=0, σ=1).

3. Logistic regression train

Because logistic regression function has two part: weights and bias, my method is merge bias into weights vector(*W*). The weights vector *W* now is dimension 58 including bias at last dimension. Features should fit it, so extend feature into dimension 58 with last dimension value equals to 1.

Input: train feature X and corresponding Y
Output: regression weight W (58-dimension include bias at last)

W^0={$W|W_i = 0$, for each i}
$\eta = 1$
iteration $i = 0$
while *true* **do**
  $i = i + 1$
  $\nabla W = \frac{1}{4001}\sum_{n=1}^{4001}(\theta(W^T X_n) - Y_n)X_n$
  $W^{i+1} = W^i - \eta\nabla W$
  if $i\%100 == 0$ **then**
    if $|\nabla W| < 0.001$ **then**
      $W = W^i$
      break
    end
  end
end
return W

4. Testing

Normalize testing features and extend to dimension 58. Calculate the inner product of logistic weight and testing feature, and then feed into sigmoid function. If the value is greater than 0.5, output 1, and else, output 0.

● Mothod2: sklearn package

Use “sklearn” package, and “pickle” package. Adapt data format to its requirement. Then, use its API of logisitic regression.

```
model=LogisticRegression()
model=model.fit(Xtrain, Y)
predicted = model.predict(Xtest)
```

The work is done.

- Discussion

	My logistic regression function	sklearn logistic
Training time	about 5 minutes	about 20 seconds
Kaggle score	0.92333	0.93

My method only use feature normalization and gradient descent. No matter in time or score, the “sklearn” package all wins. To my surprise, the training time has big gap between two method. However, the source code of “sklearn” is hard to read. I cannot understand it in a short time. It really give me a lesson. If we want to do logistic regression in the future, using package is more convenient and more accurary.