CS 5350/6350: Machine Leariniing Fall 2017

Yucheng Yang
Homework 5

Handed out: Thursday November 16th, 2017
Due date: Saturday December 2nd, 2017

General Instructions

- You are welcome to talk to other members of the class about the homework. I am more concerned that you understand the underlying concepts. However, you should write down your own solution. Please keep the class collaboration policy in mind.
- Feel free discuss the homework with the instructor or the TAs.
- Your written solutions should be brief and clear. You need to show your work, not just the final answer, but you do *not* need to write it in gory detail. Your assignment should be **no more than 10 pages**. Every extra page will cost a point.
- Handwritten solutions will not be accepted.
- The homework is due by midnight of the due date. Please submit the homework on Canvas.

1 Logistic Regression

We looked Maximum A Posteriori (MAP) learning of the logisitic regression classifier in class. In particular, we showed that learning the classifier is equivalent to the following optimization problem:

$$\min_{\mathbf{w}} \left\{ \sum_{i=1}^{m} \log(1+\exp(-y_i \mathbf{w}^T \mathbf{x}_i)) + \frac{1}{\sigma^2}\mathbf{w}^T\mathbf{w} \right\}$$

In this question, you will derive the stochastic gradient descent algorithm for the logistic regression classifier.

1. [5 points]

$$g(w) = \frac{1}{(1+\exp(-y_i \mathbf{w}^T \mathbf{x}_i))} * \exp(-y_i \mathbf{w}^T \mathbf{x}_i) * (-y_i \mathbf{x}_i) = \frac{-y_i \mathbf{x}_i}{1+\exp(y_i \mathbf{w}^T \mathbf{x}_i)}$$

2. [5 points]

$$\nabla J(w) = \frac{1}{(1+\exp(-y_i \mathbf{w}^T \mathbf{x}_i))} * \exp(-y_i \mathbf{w}^T \mathbf{x}_i) * (-y_i \mathbf{x}_i) + \frac{2w}{\sigma^2} = \frac{-y_i \mathbf{x}_i}{1+\exp(y_i \mathbf{w}^T \mathbf{x}_i)} + \frac{2w}{\sigma^2}$$

3. [10 points] Given a training set $S = \{(x_i, y_i)\}, x \in^n, y \in \{-1, 1\}$

```
(a) Initialize $w = 0 \in^n$
(b) For epoch = 1...T:
    i. Shuffle the training set
    ii. For each training example$(x_i, y_i) \in S$:
        $w \Leftarrow w - \gamma_t * (\frac{-y_i \mathbf{x}_i}{1 + \exp(y_i \mathbf{w}^T \mathbf{x}_i)} + \frac{2w}{\sigma^2})$
(c) Return w
```

2 Experiments

For this question, you will have to implement and compare six different learning strategies: SVM, logistic regression (from your answer to the previous question), the naive Bayes classifier, bagging and two ensembles over decision trees.

2.1 The task and data

2.2 Implementation Notes

2.3 Algorithms to Compare

1. [15 points] **Support Vector Machine**

   Solution: Base on the slide, I set the gamma(learning-rate) decreases $\gamma = \frac{\gamma}{1+t}$, t is the number of trained data elements. The data-set only store the indexes of features which are 1, therefore reform each data element to a $1 * 70000$ vector, in the for-loop, fill the data element in a $1 * 70000$ vector, then calculate the the sub-gradient to update the weight vector which size is $70000 * 1$. The way of calculate loss is Hinge loss.

2. [15 points] **Logistic regression**

   Solution: Base on the algorithm on question 1.3, I set the gamma(learning-rate) decreases $\gamma = \frac{\gamma}{1+t}$, t is the number of trained data elements. The data-set only store the indexes of features which are 1, therefore reform each data element to a $1 * 70000$ vector, in the for-loop, fill the data element in a $1*70000$ vector, then calculate the the sub-gradient to update the weight vector which size is $70000 * 1$. The way of calculate loss is logistic regression.

3. [15 points] **Naive Bayes**

   Solution: Base on the slide, I count the number elements of label =1 and the number elements of label =-1. Meanwhile, I count all features: the number elements of features = 1 and label = 1, the number elements of features = 0 and label = 1, the number elements of features = 1 and label = 0,the number elements of features = 0 and label

$= 0$. I put all of counted numbers in a $4 * 70000$ table as I reform data to a $n * 70000$ table. There reform this table by the smoothing term. Then the predict of Naive Bayes is to calculate the sum of all features' $log(probability)$, compare the probability of label = 1 and the probability of label = -1, then choose the higher probability label as the result of predict.

4. [15 points] **Bagged Forests**

   Solution: Reform the ID3depth function to fit the data set. Then I random choose 100 elements in training data to form a new data table. Call ID3depth to create a tree which depth is 3 based on the new data table. Repeat 1000 times, then I get 1000 trees. Create a new visit-Node function. In this predict function, each element in data set go through 1000 trees, if above 500 trees vote 1, then the prediction of this element is 1, otherwise prediction of this element is -1.

5. [10 points] **SVM over trees**

   Solution: Base on the Bagged Forests above, create 1000 trees. Base on these 1000 trees, then using the data set to collect the predictions of 1000 trees to a $1*1000$ vector, combine all these vector, we have a new data table. Throw this new table into the SVM training function before, we have a new weight vector. Then the predict function needs imputs: 1000 trees, test data and weight vector. before predicting, reform the test data based on 1000 trees, get a new prediction table. Use this prediction table combine with the weight vector to get the prediction. After getting the prediction, compare the prediction to the label in the test data.

6. [10 points] **Logistic regression over trees**

   Solution: Base on the Bagged Forests above, create 1000 trees. Base on these 1000 trees, then using the data set to collect the predictions of 1000 trees to a $1 * 1000$ vector, combine all these vector, we have a new data table. Throw this new table into the logistic regression training function before, we have a new weight vector. Then the predict function needs imputs: 1000 trees, test data and weight vector. before predicting, reform the test data based on 1000 trees, get a new prediction table. Use this prediction table combine with the weight vector to get the prediction. After getting the prediction, compare the prediction to the label in the test data.

7. [10 points] **Extra Credit**

   Solution: There are too many feature in our data set. what I set is 70000. therefore in the recursive function(ID3), when the depth is 200, it will recursive about $2^200timesandpickoneofaround70000featureswhichhasthehighestinformationgains.Notonlythetin$

2.4 What to report

1.

	Best hyper-parameters	Average cross-validation accuracy	Training accuracy	Test Accuracy
SVM	$\gamma = 1; C = 10$	0.8616	0.93896	0.85213
Logistic regression	$\gamma = 1; \sigma^2 = 10$	0.80376	0.83251	0.79468
Naive Bayes	$\lambda = 1$	0.52913	0.50106	0.5
Bagged Forests	NAN	NAN	0.75905	0.74894
SVM over trees	$\gamma = 1; C = 10$	0.8332	0.89922	0.81702
Logistic regression over trees	$\gamma = 1; \sigma^2 = 1$	0.8073	0.81299	0.78298

Table 1: Result table

Experiment Submission Guidelines

1. The report should detail your experiments. For each step, explain in no more than a paragraph or so how your implementation works.

2. *Your code should run on the CADE machines.* You should include a shell script, `run.sh`, that will execute your code in the CADE environment. Your code should produce similar output to what you include in your report.

   You are responsible for ensuring that the grader can execute the code using only the included script. If you are using an esoteric programming language, you should make sure that its runtime is available on CADE.

3. Please do not hand in binary files! We will *not* grade binary submissions.