# Your Name Weiwei Yan

# Your Andrew ID wyan

# Homework 5

**0. Statement of Assurance**

You must certify that all of the material that you submit is original work that was done only by you. If your report does not have this statement, it will not be graded. I certify that all of the material that I submit is original work that was done only by myself.

# 1. Performance evaluation (60%)

## 1.1 P@10, NDCG@10 and MAP (10%)

Write down the formula of each metric, and give one example for each of them to show that single metric is not enough to guarantee the ranking quality.

P@10 =

If two ranked lists having the same number of relevant documents within the first ten documents, P@10 gives the same quality while the relevant documents may have different positions across two ranked lists, saying for list A, five relevant documents are placed at rank 1,2,3,4,5 compared to list B, five relevant documents are placed at rank 5,6,7,8,9,10.

NDCG@10 = 

If ranked list A has 2 relevant documents at the top while ranked list B has 3 relevant documents at the top, NDCG will treat them equally even though the ranked list B should be better.

MAP = (Precision at 1st position of the relevant document + Precision at 2nd position of the relevant document + Precision at the last position of the relevant document)/total number of relevant documents

Since MAP depends on the rank of the positions, it may give more errors on the ranked list.

For example, if list A has 1 relevant document at rank 1, and 2 relevant documents at rank 9,10 versus a list B has 3 relevant documents at rank 4,5,6. The first list has higher MAP but also higher errors.

## 1.2 Performance and time cost table (20%)

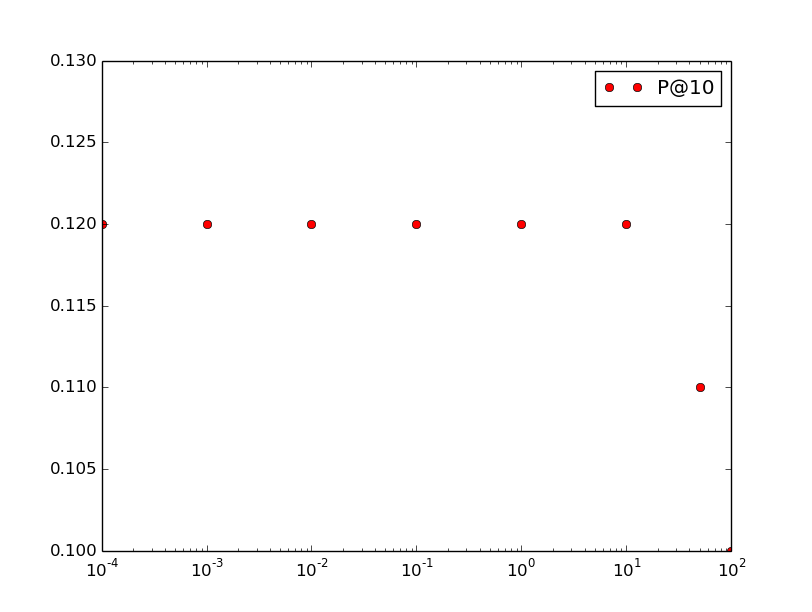
Fill in the table with your experiment results.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| C | Logistic Regression | | | SVM | | |
| P@10 | NDCG@10 | MAP | P@10 | NDCG@10 | MAP |
| 0.0001 | 0.12 | 0.23 | 0.19 | 0.09 | 0.12 | 0.07 |
| 0.001 | 0.12 | 0.23 | 0.19 | 0.09 | 0.14 | 0.088 |
| 0.01 | 0.12 | 0.23 | 0.19 | 0.11 | 0.19 | 0.17 |
| 0.1 | 0.12 | 0.23 | 0.19 | 0.14 | 0.24 | 0.19 |
| 1 | 0.12 | 0.23 | 0.19 | 0.16 | 0.28 | 0.22 |
| 10 | 0.12 | 0.22 | 0.18 | 0.18 | 0.34 | 0.27 |
| 50 | 0.11 | 0.19 | 0.16 | 0.16 | 0.32 | 0.26 |
| 100 | 0.1 | 0.17 | 0.12 | 0.15 | 0.31 | 0.26 |

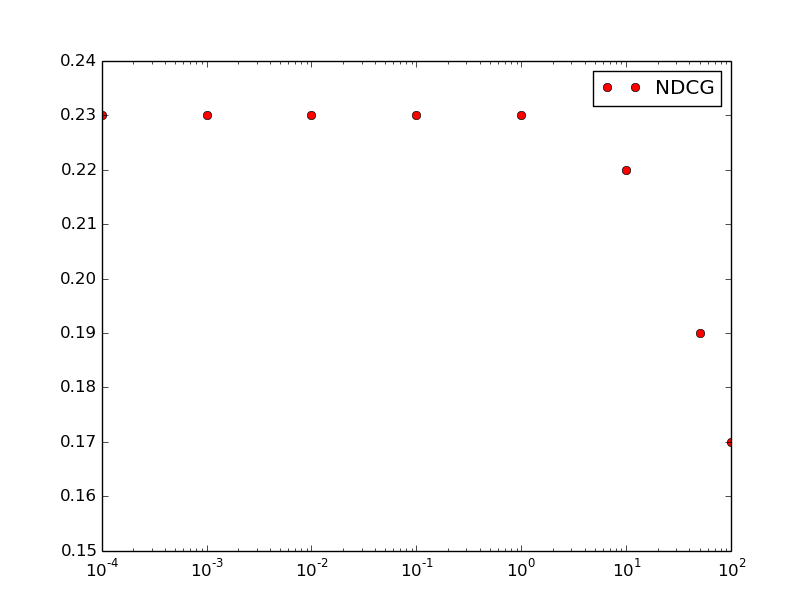
|  |  |  |
| --- | --- | --- |
| C | Time | |
| Logistic Regression (s) | SVM (s) |
| 0.0001 | 1m22s | 12m39s |
| 0.001 | 1m21s | 11m |
| 0.01 | 1m20s | 8m12s |
| 0.1 | 1m21s | 4m19s |
| 1 | 43s | 1m50s |
| 10 | 40s | 41s |
| 50 | 20s | 50s |
| 100 | 13s | 58s |

## 1.3 Plots (10%)

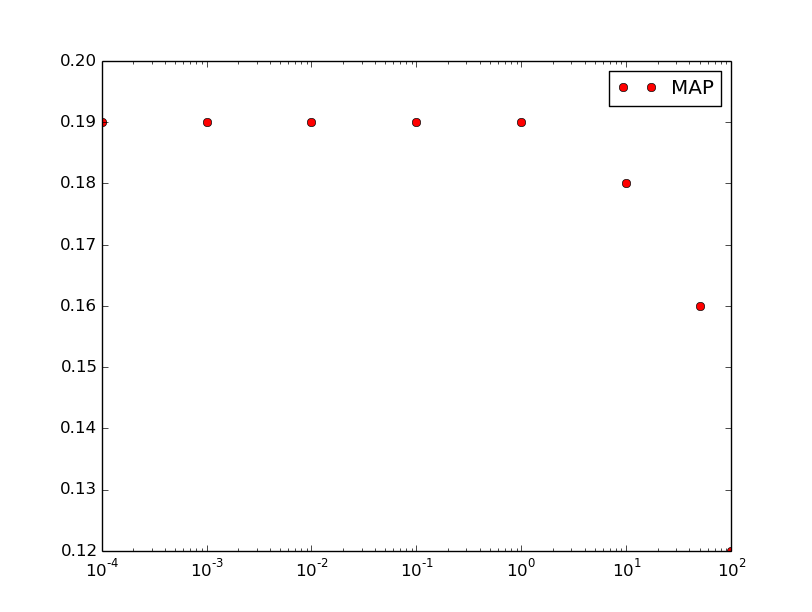
Plots graphs for each metric of Logistic Regression and SVM. You should have six graphs in total.



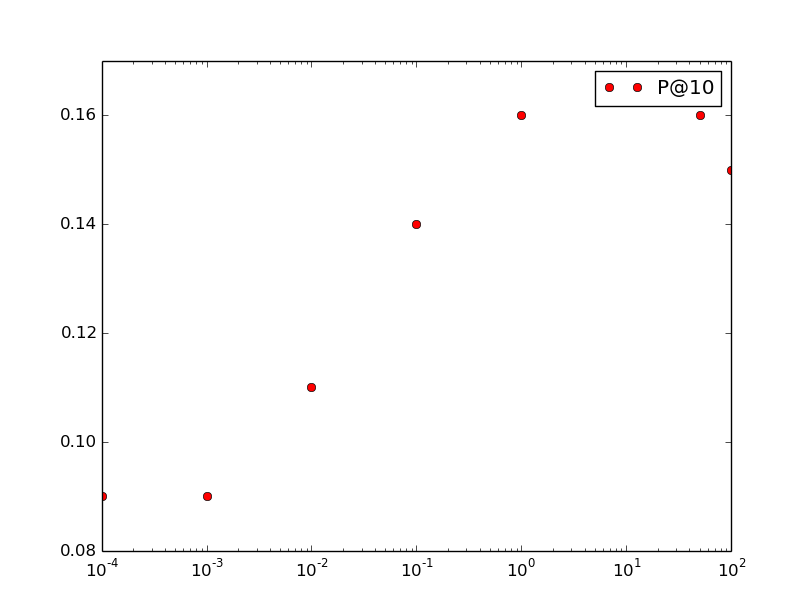
logistic regression



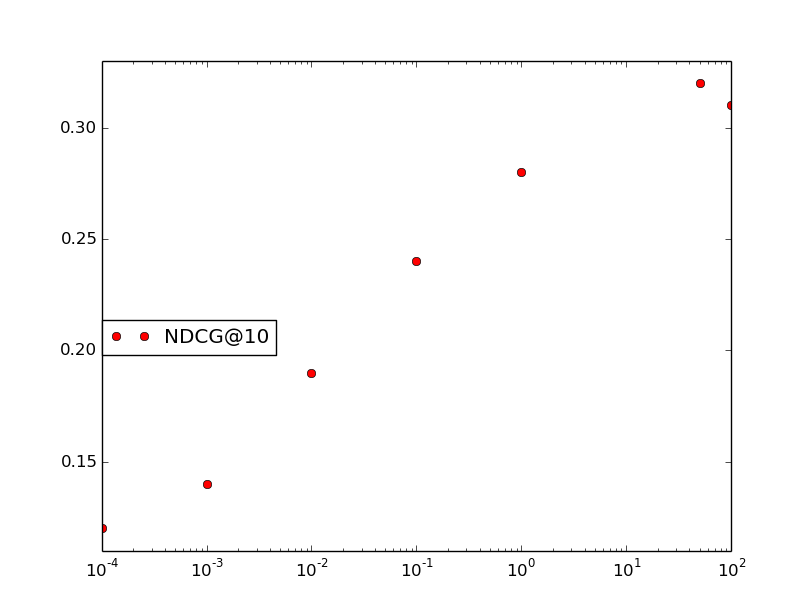
logistic regression



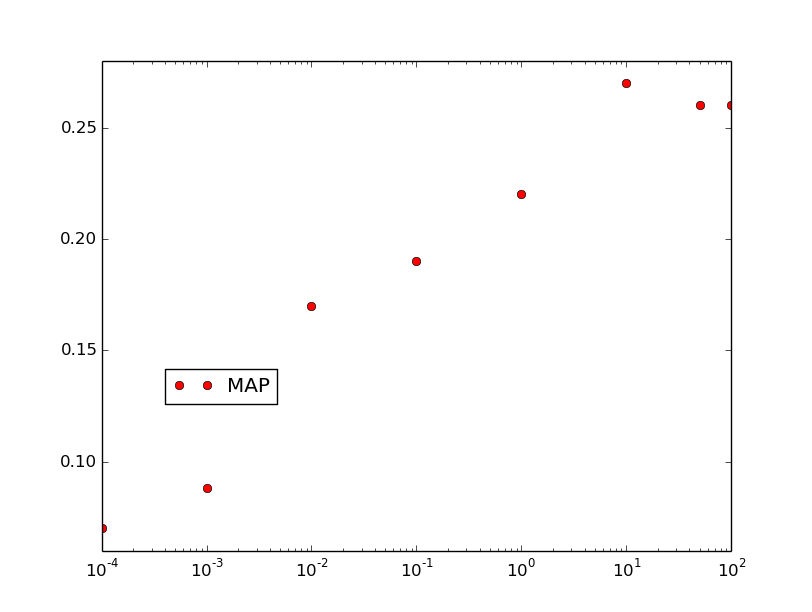
logistic regression



SVM



SVM



SVM

## 1.4 Analysis (20%)

Analysis based on your experiment results (performance and time), and some conclusions (What did you learn from the results? What are the advantages and disadvantages of using Logistic Regression and SVM? etc)

Logistic regression’s performance does not change much when C grows. Only when C is larger than 10, the logistic regression’s performance seemed to be a little bit worse. The regularization term has little impact on the LR’s performance. But overall, when C grows, the time to compute is much less. The logistic regression achieves the best performance when C is 0.1. The evaluation metrics gave consistent results. When P@10 dropped, MAP and NDCG@10 also dropped.

SVM’s performance improved dramatically when C grows from 0.0001 to 10. The time to compute also dropped when the performance is better. The regularization term successfully prevented SVM from over fitting. SVM achieves the best performance when C is 10. The evaluation metrics also gave consistent results. When P@10 increased, MAP and NDCG@10 also increased.

Advantage of using logistic regression is the computing time for logistic regression does not vary much across different Cs. Disadvantage of using logistic regression is the performance does not improve using different Cs.

Advantage of using SVM is it achieved improved performance when introducing different Cs to avoid over fitting. Disadvantage of using SVM is the time to compute when C is smaller is really slow.

# 2. Feature Design (25%)

Add three new features that you think might work, and test it. Give some analysis and conclusions based on the experiment results.

The new features I added are :

1. product of BM25 score of the anchor and BM25 score of the title

2. product of HITS authority and HITS hub

3. product of tfidf body and tfidf anchor

|  |  |  |  |
| --- | --- | --- | --- |
|  | P@10 | NDCG@10 | MAP |
| feature 1 | 0.13 | 0.284 | 0.2 |
| feature 2 | 0.12 | 0.23 | 0.19 |
| feature 3 | 0.12 | 0.23 | 0.19 |
| feature1, feature2 | 0.13 | 0.284 | 0.2 |
| feature1,feature3 | 0.13 | 0.284 | 0.2 |
| feature2,Feature3 | 0.12 | 0.23 | 0.19 |
| feature1,feature2,feature3 | 0.13 | 0.284 | 0.2 |

Only adding the feature1 to the original model has improved the performance of logistic regression. Adding this feature seems to make the object function more convex and can speedily converge to the local maximum. Adding other two features do not help to improve the performance at all and combing three features together gives the same performance as the feature 1.

# 3. The software implementation (5%)

Description of your code including any data structures used, design considerations etc. In addition, please describe any changes you made to your system after HW4.

Programming language: Python 2.7

Libraries: Numpy, Scipy, Scikit-learn(for Cartesian and L2-norm)

Data structure used:

a. Initially use list to store the labels and the features that are retrieved from the training file

b. Use ndarray to store the feature matrix and compute gradient descent

Description of code

Logistic regression:

1. def read\_data():

read the training file and construct the new feature matrix and the matrix of the labels, perform L2 normalization on the feature matrix

2. def initialize\_models():

initialize the parameter values and set them as all zeros

3. def gradient\_descent():

Do the gradient descent until convergence, return the final model’s parameter values

4. def max\_function():

Compute the log value of the objective function and compare the value of the previous iteration. Return true if the log value of the current objective function – log value of the previous object function is less than 4

5. def red\_test\_data():

Read the test data and construct the feature matrix for the testing set

6. def main():

main function

SVM:

1. read\_data()

Read the training set and construct the new feature matrix

2. write\_to\_data()

Write the new feature matrix for the training set into a txt file

3. read\_test\_data()

Remove the unnecessary query ids and the annotation, and write the format-clean test set into a txt file

4. def main():

main function

Design decision:

The stopping criteria for doing further gradient descent are that if the log value of the current objective function – log value of the previous object function is less than 4. I found if I set the stopping criteria stricter, the performance of logistic regression does not improve and it takes more time. The learning rate is 0.001. I found that if the learning rate is larger than this number, it stops immediately while it will end up with endless loops if the learning rate is smaller.