Program Documentation

Yin Yumeng
A0105408N

1 Introduction Of program

In this assignment, I use a Java Program to perform ontext- sensitive spelling correction.
I implemented a logistic regression classifier for context-sensitive spelling correction, I specially use stochastic gradient ascent learning algorithm.

2 Architecture of the program

2.1 API for the program

java sctrain word1 word2 train_file model_file | train the program
java sctest word1 word2 test_file model_file answer_file | test the program
java CheckTestAccuracy answer_file answer_file_predicted | check the accuracy of the test result

2.2 Classes of program

- FeatureExtraction
  This class will extract all features from either the training file or testing file.
- LogisticRegression
  This class will handle all training and testing process. The logistic regression with stochastic gradient ascent is used to classify data.
- CheckTestAccuracy
  This class will handling comparing the predicted data with correct data and output the accuracy of the testing result.
- sctrain
  This class will read input from user and use the FeatureExtraction and LogisticRegression to run the whole training.
- sctest
  This class will read input from user and use the FeatureExtraction and LogisticRegression to run the whole testing.

3 Implementing details

3.1 Feature extraction Class

3.1.1 Actual features used for training and testing

- Surrounding words: All word types contained in the training file with stop words removed.

- Collocations: The collocations words used for the program are all two word phrases before and after the target word.

3.1.2 Variables and functions:

- FeatureList: Vector<String>
  store all words types and the $C_{-2,-1}$ and $C_{1,2}$ phrase types contained in the training file.(when extract data for testing data, new types will be ignored)
- dataList: List<List<Integer>>
  Training :
  - Column: 1: label (0 for word0 and 1 for word1) ;
  - Column 2: ID;
  - Following columns: count of the number of word or phrases contained for the sentence in featureList)

  Testing :
  - Column 1: ID
  - Following columns: count of the number of word or phrases contained in featureList (word or phrase will be ignored if not contained in featureList))

3.2 LogisticRegression Class

3.2.1 Parameters for logistic regression classifier:

- learningRate: 0.1
- maxIterations: 20000
- threshold:0.002

3.2.2 Details when implementing logistic regression with gradient ascent:

- For each iterations ran in program, program will go through all training samples. Here stochastic gradient ascent is used that only one sample is used to update weights array. To train each sample of data, function to update weight value at a specific place is listed as following.

```
for (int i=0; i<x.size(); j++) {
    weights[i] = weights[i] + learningRate * (label - predicted) *
x.get(i);
}
```

Hera formula used is listed as following:

$$w_{i(k+1)} \leftarrow w_{i(k)} + \alpha \cdot x_i^j \cdot (y^j - \frac{1}{1 + e^{-w_k \cdot x^j}})$$

And the $||\nabla l(w)||$ is computed everytime after all trainings are complete for an iterations.
The iterations will stop only when either the maxIterations is met or when $||\nabla l(w)||$ is less than a predefined threshold.

- Predicted label is computed by using formula.

$$P(y = 1|x) = \frac{1}{1 + e^{-w \cdot x}}$$

$$w \cdot x = w_0 x_0 + w_1 x_1 + \cdots + w_{n-1} x_{n-1}$$

In the program, the function `classify(List<Integer> x)` will take the training data for a sample as input and using current weight array to compute predicted label.