Report of Assignment1

Author: Zheming Kang; Date: Jan.30th

1. Instructions to compiling and running codes

There is only one file of my code. At the beginning of my code, there is a variable called filepath. `filePath="../data/covtype.data"` Here you can change the address of the covtype.data file or put the covtype.data into the data folder.

The environment requirements are: python: 3.8, numpy: 1.24.

2. Document of my programs

At the beginning of my file, there was a function called readData(). This is the function that we used to read the covtype.data file from our disk, and store the data in the memory. I used np.loadtxt function to store the data in a numpy 2D array.

2.1 Band Join

In the function named bandJoin(), I implemented an optimized band join function to calculate result.

First, I sorted the first column of the data.

```
sorted_indices=np.argsort(data[:, 0])
sorted_data=data[sorted_indices]
```

Then, I used a dictionary to count the number of each value in the first column of the data.

```
valueCount = {}
for i in range(len(dataInUse)):
    if dataInUse[i] in valueCount:
        valueCount[dataInUse[i]] += 1
```

```
    else:
        valueCount[dataInUse[i]] = 1
```

After that, we can make a reference and calculate them directly.

First, turn the dictionary into a list. Since the data is sorted, and we read the data line by line, this dictionary is also ordered, which will lead to a ordered list.

Then, we read them one by one. The number m represents how many records are there in the data with the value equals to what we are considering. n is m-1.

For a value, there must be (1+n)*n/2 records with the different = 0. This is simple mathematics. Then, for i+1, i+2,... if the differenct is smaller than k, we know there are m * t records also fits the requirement, where t is the number of how many records are there in the data with the value equals to i+j th element.

```
values = list(valueCount.keys())
result = 0
for i in range(len(values)):
    n = valueCount[values[i]]-1
    m = valueCount[values[i]]
    result += (1+n)*n/2
    j = 1
    while i+j < len(values) and np.abs(values[i] - values[i+j]) <= k:
        result += m*valueCount[values[i+j]]
        j += 1
```

Finally, we add the results together, and we'll get the correct answer.

2.2 Similarity

I implemented 2 functions to answer question 2: normalize, and similarity. The normalize function reads the data, and normalize the first 10 columns by `data[:, i] = (data[:, i] -mincol)/(maxcol-mincol)` . The similarity function calculates the similarity between each pair of datapoints.

For the first 10 columns, the similarity is:

```
for k in range(10):
    di = np.abs(data[i][k] - data[j][k])
    si = 1/(1+di)
    delta += 1
    similarity += si
```

and for the next several columns, the similarity is:

```
for k in range(10, collen):
    if data[i][k] == data[j][k] and data[i][k] == 1:
        similarity += 1
        delta += 1
    if data[i][k] != data[j][k]:
        delta += 1
```

which follows the idea from our lecture slides.

After calculating the sum similarity of one pair of data points, we still need to let the sum divided by delta, to get the pair similarity. We add the result to averageSimilarity and update the max, min values. After the calculation of all the pairs of records, we calculate the average similarity by `averageSimilarity/(rowlen*(rowlen-1)/2)`

The following is the sample codes:

```
... For each pairs of records:
        similarity = similarity/delta
        averageSimilarity += similarity
        minimumSimilarity = min(minimumSimilarity, similarity)
        maximumSimilarity = max(maximumSimilarity, similarity)
# After all pairs are done
averageSimilarity = averageSimilarity/(rowlen*(rowlen-1)/2)
```

Finally, I implemented a main function to help user select the functions they are looking for. For question2, you can choose "1" to select the similarity for all the data, or choose "2" to see the similarities from each types of data.

Results

Results for question 1

K = 0

```
This is the solution of assignment 1.
Please input a NUMBER to choose solution.
1. Band Join
2. The similarity function
Please input a number: 1
You are choosing Band Join.
It will take a integer as parameter, please input a integer.
0
Reading data from file...
Reading data - Finished
Sorting data - Finished
Calculating result...
Scanning Data for All Values...
Scanning Data for All Values - Finished
Calculating result...
Calculating result - Finished
The result is:
199138468.0
```

K = 1

```
This is the solution of assignment 1.
Please input a NUMBER to choose solution.
1. Band Join
2. The similarity function
Please input a number: 1
You are choosing Band Join.
It will take a integer as parameter, please input a integer.
1
Reading data from file...
Reading data - Finished
Sorting data - Finished
Calculating result...
Scanning Data for All Values...
Scanning Data for All Values - Finished
Calculating result...
Calculating result - Finished
The result is:
568355289.0
```

K = 2

```
This is the solution of assignment 1.
Please input a NUMBER to choose solution.
1. Band Join
2. The similarity function
Please input a number: 1
You are choosing Band Join.
It will take a integer as parameter, please input a integer.
2
Reading data from file...
Reading data - Finished
Sorting data - Finished
Calculating result...
Scanning Data for All Values...
Scanning Data for All Values - Finished
Calculating result...
Calculating result - Finished
The result is:
937077919.0
```

K = 3

```
This is the solution of assignment 1.
Please input a NUMBER to choose solution.
1. Band Join
2. The similarity function
Please input a number: 1
You are choosing Band Join.
It will take a integer as parameter, please input a integer.
3
Reading data from file...
Reading data - Finished
Sorting data - Finished
Calculating result...
Scanning Data for All Values...
Scanning Data for All Values - Finished
Calculating result...
Calculating result - Finished
The result is:
1327371987.0
```

Result for question 2

Similarity between all the data

```
PS C:\Users\sygra\Documents\GitHub\COMP7107\Assignment1\src> & C:/Users/sygra/.conda/envs/
.py
This is the solution of assignment 1.
Please input a NUMBER to choose solution.
1. Band Join
2. The similarity function
Please input a number: 2
You are choosing The similarity function.
Reading data from file...
Reading data - Finished
Normalizing data...
Normalizing data - Finished
Please input a number to choose random saple from: 1. all data; 2. each type of the data
1
You are choosing random sample from all data.
data shape is:  (1000, 55)
The number of columns is:  54
The number of rows is:  1000
Calculating similarity...
Calculating similarity - Finished
The minimum similarity is:
0.48863842525702145
The maximum similarity is:
0.9948972334528499
The average similarity is:
0.6865057901929233
```

Similarities between each type of the data

```
Type  1  min similarity  0.5290499920055479  max similarity  0.9935424047744296  average similarity  0.6966093731792534
Type  2  min similarity  0.5170408288291768  max similarity  0.9975705612885463  average similarity  0.6959200821725856
Type  3  min similarity  0.5283733327176204  max similarity  0.9974772926046279  average similarity  0.7211175580785919
Type  4  min similarity  0.6769976933991922  max similarity  0.9983191732252698  average similarity  0.8006721807064157
Type  5  min similarity  0.5357593036753752  max similarity  0.9983596134988796  average similarity  0.7145975797871182
Type  6  min similarity  0.540475029198333  max similarity  0.9981144899367028  average similarity  0.7350819112861223
Type  7  min similarity  0.5130522491972078  max similarity  0.9990646595714732  average similarity  0.7150515149084069
```

Observations

Observations for Q1

- When k is from 0 to 3, the answer is 1.99, 5.68, 9.37, 13.27 (* 10^8). We can see each time when k increase 1, there will be 4 * 10^8 more records.

Observations for Q2

- From question 2, we could see the minimum similarity for all the data is exteremely smaller than each type. This is reasonable since data in one type should be more similar than each other rather than other types.
- The maximum simialrity seems not very far from the specific types. This is also reasonable because we might choose some data from one type of data.

- The average is slightly lower than those from each type of data. Even though the sample volume is high, which might let the difference not high, it is still lower than the similarity from one type.