CS 839 Spring 2019, Project Stage 3

PROJECT REPORT

TEAM MEMBERS

1. Karthik Chandrashekar - karthik.chandrashekar@wisc.edu
2. Kartik Sreenivasan - ksreenivasa2@wisc.edu
3. Niveditha Hariharan - nhariharan@wisc.edu

Filtering Candidate set:

The size of our candidate set is 524 (greater than 500). So we we randomly sampled 50 tuples and manually labeled them. We got a density of 0.44 as shown in the screenshot below of the iPython notebook code. Hence we did not write blocking rules to reduce our candidate set.

Stage 3 Work

In [8]:
```
import pandas as pd
candidate_set_df = pd.read_csv('candidate_set')
table_a_df = pd.read_csv('table_a', )
table_a_df = table_a_df.rename(index=str, columns={"_id": "A_id"})
table_b_df = pd.read_csv('table_b')
table_b_df = table_b_df.rename(index=str, columns={"_id": "B_id"})
first_join = pd.merge(candidate_set_df, table_a_df, on='A_id')
result = pd.merge(first_join, table_b_df, on='B_id')

# Extract a sample of 50 tuples randomly and label them
sample1 = result.sample(n=50, random_state=4)

# Read the labeled file back and compute the density
labeled_sample1 = pd.read_csv('sample1_labeled')
np = len(labeled_sample1[labeled_sample1.label == True])
density = np*1.0/len(labeled_sample1)
print 'Density =', density
```

```
Density = 0.44
```

Estimating Accuracy:

Now that we have a density of 0.44 which is greater than 0.2, we sampled 350 more items from the same set (400 in total using the same seed as before) as shown in the screenshot below of the iPython notebook code.

In [9]:
```
# Since our density is more than 0.2 we can break here
# Randomly samepl 350 more tuples. If we use the same seed we will get the first 50 elements we got b
efore.
final_sample = result.sample(n=400, random_state=4)
```

In [10]:
```
# read the labeled pairs file, i.e. the file with the labels
labeled_pairs = pd.read_csv('final_labeled')
print(estimate_PR(labeled_pairs, dfc, dfp))
```

```
((0.9690025475829874, 0.9995013894248866), (0.9559581611784236, 0.9820263349456073))
```

Results:

The precision and recall we obtained are as follows:

Precision : (0.9559581611784236, 0.9820263349456073)
Recall : (0.9690025475829874, 0.9995013894248866)