CS 784 Project Stage 3

Walter Cai {wcai@cs.wisc.edu}
Guangshan Chen {gchen9@wisc.edu}

October 26, 2015

Blocker Choice

After considering multiple blockers, we have opted to use a Sorted Neighborhood over the movie release-year attribute. The canopies will consist of movie pairs whose release years are within 1. Another idea was an index over tokens derived from the movie titles.

Development

Because sorted neighborhood has not yet been implemented within `Magellan`, we independently developed the simple neighborhoods using the blackbox blocker implementation that is already in place. Nevertheless, we suspect there are a variety of optimization that can and should be implementable. For example, we suspect that the blackbox does not perform a sort before generating candidate pairs.

The resulting `tableC.csv` contains 1148823 candidate pairs; a reduction of 87.23% from the potential $3000^2 = 90000000$ candidate pairs that would have been considered without the movie year blocking.

Problems Faced

One problem we faced before blocking could even begin was reading the data into `Magellan`. This is because several of the extracted movie titles contained commas which were interfering with the structure of the `csv` format for `tableA.csv` and `tableB.csv`.In retrospect, it would have behooved us to create a placeholder variable to take the place of he comma in the acquisition phase. We also noticed the existence of empty tuples: "`[ID#],,,,,`" which needed to be removed as they presented no hope of future matching. Both of these anomalous types were removed from both original tables via a preliminary python cleaning script. In total, less that 400 of the original 20000+ extracted tuples needed to be discarded for these reasons.

Similarly, movies that failed to list a year were also discarded from the resulting blocks.

Another challenge was the fact that sorted neighborhood has not yet been implemented. We believe this is a crucial next addition and are considering implementing it as the final stage of this project. A resulting challenge was understanding the desired syntax for the blackbox blocker constructor although this was quickly understood as well.

Before running the blocking script on any large amount of data, we first tested the ad hoc sorted neighborhood blocker on simply 10-tuple tables. After manually checking the results, we began the actual blocking compilation. This led to the final bottleneck: runtime for the construction of candidate pairs in `tableC.csv`. At first we were running the blocker on the two 10K size tables, however, after a while we opted to instead only try and run the blocker on the first 3K tuples from either tuple. With this modification the runtime was approximately 1600 seconds, or just under half an hour.