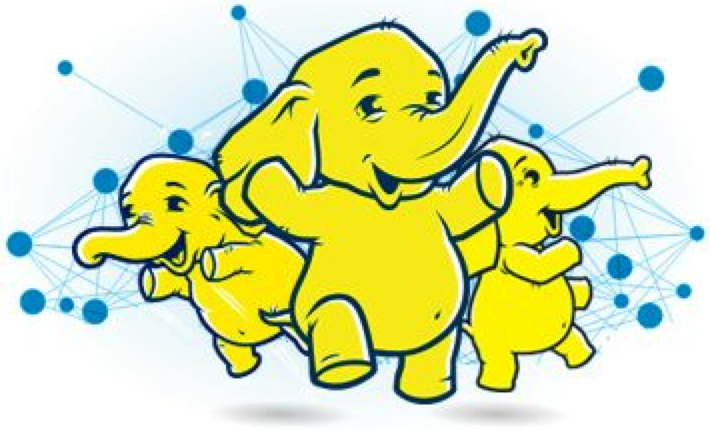
LSU
CSC 4740 (Big Data Technology)



Determining Accuracy of Yelp Ratings Based on Review Text

May 5, 2015

Taylor Lapeyre & Andrew Bergeron

Table of Contents

1. Introduction

Our project was a Hadoop mapreduce program for Yelp's academic dataset which ran sentiment analysis on reviews. The goal of this project is to accurately predict the rating of a business based on the language that reviewers use. To give us something to go off of, we used Pittsburg University's Subjectivity Lexicon as a database of words to their respective sentiment values.

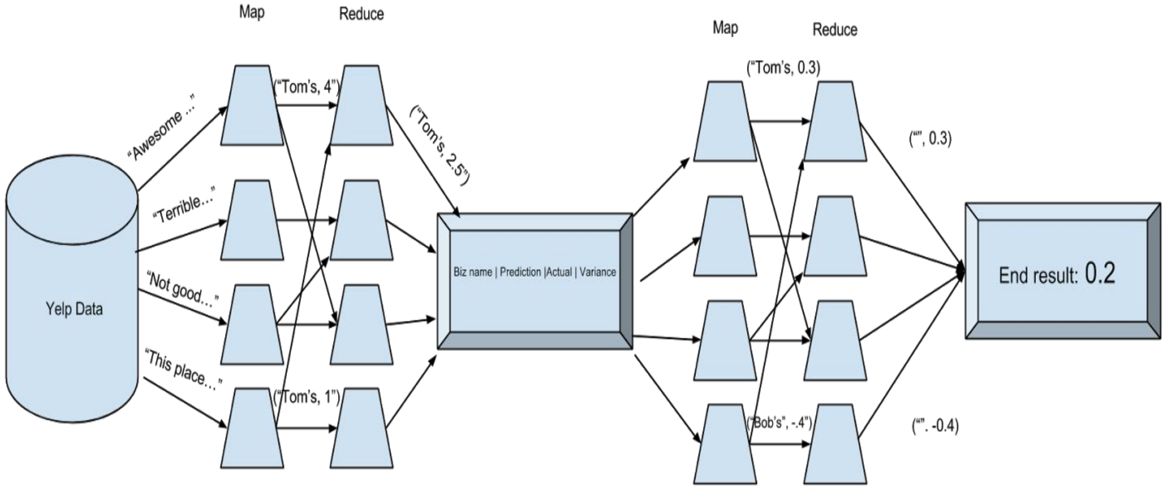
2. System Overview

Our project consists of two mapreduce jobs. The first job receives as input the entire academic database from yelp. This job will read each business and get a few important pieces of data which includes the “id” of the business and the current rating of the business. The job will then generate a rating prediction based on the text of each review. The output of this first job is the average of all of these predictions for each business.

A second job will be used to average the difference between our predictions and the true ratings (stars) of all businesses analysed. We will adjust our algorithm until the variance between our estimate and the truth is as close to 0 as possible.

3. Design Overview

JOB 1   *JOB 2*



4. Related Works

Improving Restaurants by Extracting Subtopics from Yelp Reviews
by James Huang, Stephanie Rogers, Eunkwang Joo
http://www.yelp.com/html/pdf/YelpDatasetChallengeWinner_ImprovingRestaurants.pdf

Hidden Factors and Hidden Topics: Understanding Rating Dimensions with Review Text
by Julian McAuley, Jure Leskovec
http://www.yelp.com/html/pdf/YelpDatasetChallengeWinner_HiddenFactors.pdf

5. System Architectural Design

We will be utilizing the MapReduce architecture which the Hadoop system offers. The Map procedure performs the filtering and sorting of data while the Reduce procedure performs a summary operation. This system orchestrates the processing by marshalling the distributed servers, running the various tasks in parallel, managing all communications and data transfers between the various parts of the system, and providing for redundancy and fault tolerance.

6. Detailed Description of Components

1. Mapper Components
    a. Sentiment Mapper()
        i. Converts review json object to a java object and calculates a review object based on the sentiment of the review.
    b. Variance Mapper()
        i. Extracts the true rating and rating prediction from the output of the first job, calcualtes the variance between the two, and pases it to the reducer.
2. Reducer Components
    a. Rating Reducer()
        i. Averages the predicted ratings for a particular business.
    b. Accuracy Rating Reducer()
        i. Averages the variances between the real rating and our predicted rating.
3. Review
    a. Represents a single view in the data set
    b. Contains the logic for how to calculate the sentiment value based on the review text.
4. **Sentiment Evaluator**
    a. Reads the “Pittsburg University Subjectivity Lexicon” data set and converts it into a java object that we can query.

i. For example the line,
   type=weaksubj len=1 word1=acceptable pos1=adj stemmed1=n priorpolarity=positive
ii. Data used: **type**, **word**, and **priorpolarity**.
    1. The type is either "*weaksubjective*" or "*strongsubjective*" where we have different values for each.
    2. The **priorpolarity** tells us whether the word is a positive of negative sentiment.
iii. Will be parsed and inserted into SentimentEvaluator as a key/value.

7. Evaluation and Testing

We used rudimentary testing with JUnit to verify that core functions (such as Review.java's caculateSentimentValue method) were working correctly. Besides this, we were able to confirm that our mappers and reducers were functioning correctly based on their output.

8. Conclusion

The end result of all computation done in this project was a single number: the average variance between our predictions and the true ratings of all businesses analysed. We call this number our accuracy rating.

The value of this number is of critical importance: the closer to zero it becomes, the more accurate our algorithm was. If positive, the conclusion is that users generally rate more negatively than they think. Conversely, a negative value indicates that users tend to inflate their rating.

The algorithm we generated resulted in an accuracy rating of **1.0963274215972127**. This result implies that the language that users write in their reviews is not as critical of the business as the "star" rating which the user gave. Therefore, we conclude that there is a skew in the two methods that users have to rate businesses. Yelp may want to consider making their "star" rating system more complex than just a simple average to alleviate this issue.