Yelp Recommendation

A Comparision of Cluster and User-to-User Algorithm

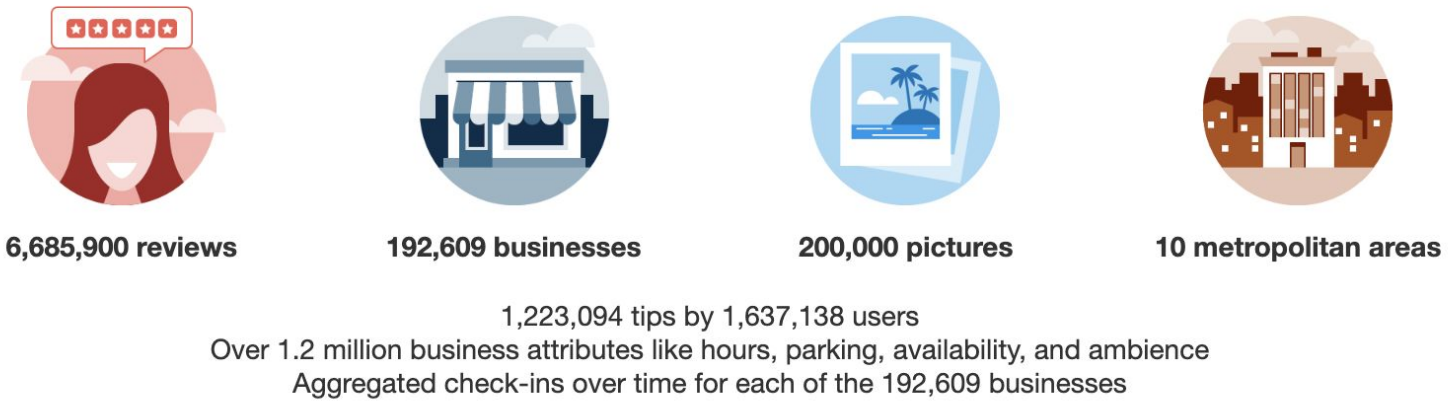
Ellen Hsieh, Fulin Guo, Jiaxu Han, Ying Sun
Computational Social Science
University of Chicago

Content

- Dataset Description
- Hypotheses
- Algorithms
- Big Data Approach - MapReduce
- Challenges
- Results
- References

Description of Dataset

- Original Yelp Dataset (about 8GB)



6,685,900 reviews

192,609 businesses

200,000 pictures

10 metropolitan areas

1,223,094 tips by 1,637,138 users
Over 1.2 million business attributes like hours, parking, availability, and ambience
Aggregated check-ins over time for each of the 192,609 businesses

- What we actually used:
  - 1 million reviews from review.json

Hypotheses

1. Some people have similar taste in choosing restaurants.

2. User-to-user algorithm has a higher prediction accuracy than clustering algorithm for recommending the businesses for the user.

Algorithms

1. **User-to-User**: Calculate the similarity scores for every two users. Then for each user, select 5 users / 3 users that have the highest similarity scores with respect to this user. We also calculate the prediction accuracy and select parameters using the test set.

2. **Cluster Algorithm**: Classify users based on whether they “like” same restaurants. The parameters include the threshold of ratings, the threshold of times visiting the same restaurant, and the threshold of times two users exist in the same sub-cluster. We use the same criteria as the user-to-user algorithm to calculate the prediction accuracy in the test set.

BIG DATA APPROACH: MAPREDUCE

- Cluster Algorithm: 3-Step MapReduce Implementation
  - **1st step**: Get the tuples (ratings, visit times) for all the customers for each business. We used the business as the key and the tuples (rating, visit times) as the value.
  - **2nd step**: Use the random variable generated by python to decide whether the business be put in the training set or the test set. If the business set is in the training set, we decide which users should be classified in the same subclassification. If the business set is in the test set, we calculate whether the two users are truly similar in selecting this restaurant. This step yields subclassifications of users (if the business is in the training set) and the number of successful predictions for user pairs visiting this restaurant (if the business is in the test set).
  - **3rd step:** For the training set, we combined subclassifications to generate the final classification of users based on the times threshold. For the test set, we calculated the prediction accuracy rates.

BIG DATA APPROACH: MAPREDUCE

- User-to-User Algorithm: 4-Step MapReduce Implemention
  - **1st step:** Get the tuples (rating, visiting times) for all the customers for each business (key: business id, value: the tuple)
  - **2nd step:** Giving different weights for ratings and visit times to construct similarity scores for each user pairs (training/test = 0.7/0.3). (key: user pair, value: similarity score)
  - **3rd step:** Find the top 5/top 3 most similar users
  - **4th step:** Compute the accuracy scores to evaluate the model (select the parameter to determine the best weights split between ratings and visiting times.)

- Friend recommendation: the highest -similarity-score user

CHALLENGES

- *Solved (or kind of solved):*
  - Unzip big data file and store it
  - Convert large json file to csv file
  - Using mrjob
  - Algorithm
  - calculate prediction accuracy
  - perparameter tuning and run time: 2~3 hours for each user-to-user model, ~30 minutes for each cluster model
  - MapReduce + Google cloud VM instance (16 CPU, 6 cores, 50 GB disk) + Dataproc, can only run 1 million data (Permission denied problem while running MrJob code)
- *Still have question:*
  - Combining two results from MapReduce
  - Using iterative mapreduce to implement k-means method

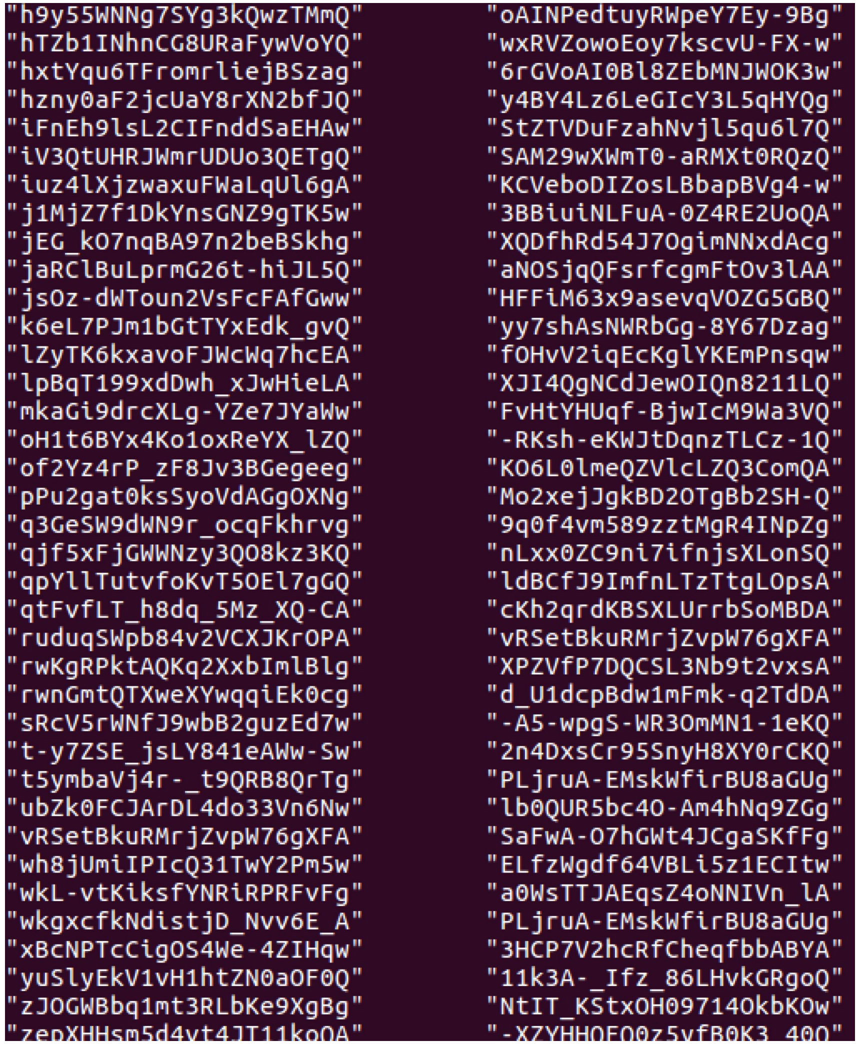
results -Model Selection

Model	User-to-User	User-to-User	User-to-User	User-to-User
Parameter	Frequency weight: 0.5 Rating weight: 0.5 Top 5 similar users	Frequency weight: 0.7 Rating weight: 0.3 Top 5 similar users	Frequency weight: 0.7 Rating weight: 0.3 Top 3 similar users	Frequency weight: 0.9 Rating weight: 0.1 Top 3 similar users
Prediction Accuracy	Prior: 0.6667481608020919 Post: 0.6639369244020407	Prior: 0.6666976368553207 Post: 0.6737991618310767	Prior: 0.6667046212301068, Post: 0.6769557999584976	Prior: 0.6666575201470023, Post: 0.6800968220504412
Model	Cluster	Cluster	Cluster	Cluster
Parameter	Frequency threshold: 1 Rating threshold: 2 Times threshold: 1	Frequency threshold: 1 Rating threshold: 3 Times threshold: 1	Frequency threshold: 1 Rating threshold: 3 Times threshold: 2	Frequency threshold: 1 Rating threshold: 4 Times threshold: 1
Prediction Accuracy	Prior: 0.6685598510496469 Post: 0.7611386138613861	Prior: 0.673182786646351 Post: 0.75	Prior: 0.6885730674465395 Post: 1.0	Prior: 0.6739148305405427 Post: 0.7765726681127982

Results-Friend recommendation

- apply the model on the whole dataset

- a user and his/her most
  similar user (recommend friend)



```
"h9y55WNNg7SYg3kQwzTMmQ"        "oAINPedtuyRWpeY7Ey-9Bg"
"hTZb1INhnCG8URaFywVoYQ"        "wxRVZowoEoy7kscvU-FX-w"
"hxtYqu6TFromrliejBSzag"        "6rGVoAI0Bl8ZEbMNJWOK3w"
"hzny0aF2jcUaY8rXN2bfJQ"        "y4BY4Lz6LeGIcY3L5qHYQg"
"iFnEh9lsL2CIFnddSaEHAw"        "StZTVDuFzahNvjl5qu6l7Q"
"iV3QtUHRJWmrUDUo3QETgQ"        "SAM29wXWmT0-aRMXt0RQzQ"
"iuz4lXjzwaxuFWaLqUl6gA"        "KCVeboDIZosLBbapBVg4-w"
"j1MjZ7f1DkYnsGNZ9gTK5w"        "3BBiuiNLFuA-0Z4RE2UoQA"
"jEG_kO7nqBA97n2beBSkhg"        "XQDfhRd54J7OgimNNxdAcg"
"jaRClBuLprmG26t-hiJL5Q"        "aNOSjqQFsrfcgmFtOv3lAA"
"jsOz-dWToun2VsFcFAfGww"        "HFFiM63x9asevqVOZG5GBQ"
"k6eL7PJm1bGtTYxEdk_gvQ"        "yy7shAsNWRbGg-8Y67Dzag"
"lZyTK6kxavoFJWcWq7hcEA"        "fOHvV2iqEcKglYKEmPnsqw"
"lpBqT199xdDwh_xJwHieLA"        "XJI4QgNCdJewOIQn8211LQ"
"mkaGi9drcXLg-YZe7JYaWw"        "FvHtYHUqf-BjwIcM9Wa3VQ"
"oH1t6BYx4Ko1oxReYX_lZQ"        "-RKsh-eKWJtDqnzTLCz-1Q"
"of2Yz4rP_zF8Jv3BGegeeg"        "KO6L0lmeQZVlcLZQ3ComQA"
"pPu2gat0ksSyoVdAGgOXNg"        "Mo2xejJgkBD2OTgBb2SH-Q"
"q3GeSW9dWN9r_ocqFkhrvg"        "9q0f4vm589zztMgR4INpZg"
"qjf5xFjGWWNzy3QO8kz3KQ"        "nLxx0ZC9ni7ifnjsXLonSQ"
"qpYllTutvfoKvT5OEl7gGQ"        "ldBCfJ9ImfnLTzTtgLOpsA"
"qtFvfLT_h8dq_5Mz_XQ-CA"        "cKh2qrdKBSXLUrrbSoMBDA"
"ruduqSWpb84v2VCXJKrOPA"        "vRSetBkuRMrjZvpW76gXFA"
"rwKgRPktAQKq2XxbImlBlg"        "XPZVfP7DQCSL3Nb9t2vxsA"
"rwnGmtQTXweXYwqqiEk0cg"        "d_U1dcpBdw1mFmk-q2TdDA"
"sRcV5rWNfJ9wbB2guzEd7w"        "-A5-wpgS-WR3OmMN1-1eKQ"
"t-y7ZSE_jsLY841eAWw-Sw"        "2n4DxsCr95SnyH8XY0rCKQ"
"t5ymbaVj4r-_t9QRB8QrTg"        "PLjruA-EMskWfirBU8aGUg"
"ubZk0FCJArDL4do33Vn6Nw"        "lb0QUR5bc4O-Am4hNq9ZGg"
"vRSetBkuRMrjZvpW76gXFA"        "SaFwA-O7hGWt4JCgaSKfFg"
"wh8jUmiIPIcQ31TwY2Pm5w"        "ELfzWgdf64VBLi5z1ECItw"
"wkL-vtKiksfYNRiRPRFvFg"        "a0WsTTJAEqsZ4oNNIVn_lA"
"wkgxcfkNdistjD_Nvv6E_A"        "PLjruA-EMskWfirBU8aGUg"
"xBcNPTcCigOS4We-4ZIHqw"        "3HCP7V2hcRfCheqfbbABYA"
"yuSlyEkV1vH1htZN0aOF0Q"        "11k3A-_Ifz_86LHvkGRgoQ"
"zJOGWBbq1mt3RLbKe9XgBg"        "NtIT_KStxOH09714OkbKOw"
"zepXHHsm5d4yt4JT11koQA"        "-XZYHHOFQ0z5vfB0K3_40Q"
```

REFERENCES

- http://aimotion.blogspot.com/2012/08/introduction-to-recommendations-with.html
- Bodoia, Max. "Map-Reduce Algorithms for k-means Clustering." accessed online at: https://stanford. edu/~ rezab/classes/cme323 S 16.