Scoring Dimensions of User Reviews

Savla | Thakkar | Vilankar

Table of Contents

Scoring Dimensions of User Reviews

Introduction and Motivation

Yelp is a way for users to rate & review local businesses. Businesses organize their own listings while users rate the business from 1-5 stars and write their own text reviews. Though star ratings provide a generic score for a business, reviews contain more detailed information about their strengths and weakness. However, over time the number of reviews grow exponentially, making it difficult for potential users to read all of them to make a well-informed decision. Our project aims to provide business owners and users valuable insights by summarizing reviews and predicting personalized ratings for every user for the various restaurants.

Problem Definition

Businesses constantly evaluate new reviews to improve their service. Therefore, it would be very useful to have reviews summarized by identifying and scoring various features. For instance, review-features for restaurants would include food, quality, taste, service, ambience etc. These features also help users evaluate a business quickly and make a choice based on personal preferences. As an example, a user going on a date might weigh ambience higher than a student who might value price. In addition, the system also predicts the rating a user would give to a business based on users past ratings. This helps users to quickly find businesses he is more probable to like.

Survey

Our literature survey involved studying techniques to mine reviews and ratings.

Most of the papers studied ratings [1, 2] and review-text in isolation. [1] and [2] describe the matrix-factorization algorithms that use ratings but fail to incorporate valuable user-preference information from reviews.

Few works combine ratings and reviews. The most similar line of work is aspect discovery [3, 4 and 5]. Aspects discover dimensions along which ratings and reviews vary. These dimensions do not necessarily explain the variation present across entire review corpora. For example, individual users assign different weights to aspects such as 'smell' and 'taste' when reviewing food. However, this does not explain why one user may love the smell of a dish, while another does not.

The goal of Latent Dirichlet Allocation (LDA) [6] is to discover hidden dimensions in text. But, the problem is that the dimensions discovered from reviews are not necessarily correlated with ratings. However, a variant of LDA called supervised LDA (sLDA) [7] has dimensions related to output variables (such as ratings). sLDA can predict movie ratings from reviews but still can not combine both ratings and review text together.

Hidden Factors and Topics (HFT) [8] associates latent rating dimensions (such as those of latent-factor recommender systems) with latent review topics (such as those learned by LDA).

One way to summarize reviews is to identify product features for which people have expressed their opinions [9]. This paper provides us a base of how we can generate frequent features using association rule mining [10] and prune unnecessary ones. A major drawback of this paper is that it can only identify explicit features and not implicit ones.

Features can be even temporal and user-centric [11]. This paper enlists the techniques for feature selection. Though the paper infers future business attention, the potential uses are limited.

Once you have a list of product features you need to determine people's opinion on those features. By determining the sentiment polarity of opinions you can summarize reviews by the count of positive and negative opinions for each product feature [12]. This paper provides us with a baseline to determine the sentiments associated with product features. The major drawback of this paper is that it just gives +ve/-ve count (polarity) for features whereas we predict scores on a scale of 0-10 for all features.

We aggregate sentiment of opinion words to predict quantitative score for dimensions and use Sentiwordnet to determine the sentiment of individual words [13]. A major drawback of Sentiwordnet is that many words have 0 values. To overcome this, we create our own dictionary that includes sentiments of all opinion words on a scale of 0-4.

Our Approach

Intuition - List of Innovations

Our approach is better than existing review summarization and feature score prediction techniques because:

- ➔ Determines sentiment score instead of sentiment orientation.
- ➔ Supports crowd based feature identification. Discovering features from user reviews gives more insight than the businesses explicitly stating them because it shows what customers think is important and identifies weaknesses businesses would never mention.
- ➔ Tag cloud visualization of review features reduces the time and effort a user has to take to decide if a business matches their expectation.
- ➔ The visualization allows business owners to can gain insight into:
  - ◆ how well users think they are performing on different dimensions
  - ◆ compare performance with competitors to plan future business strategy.

Description

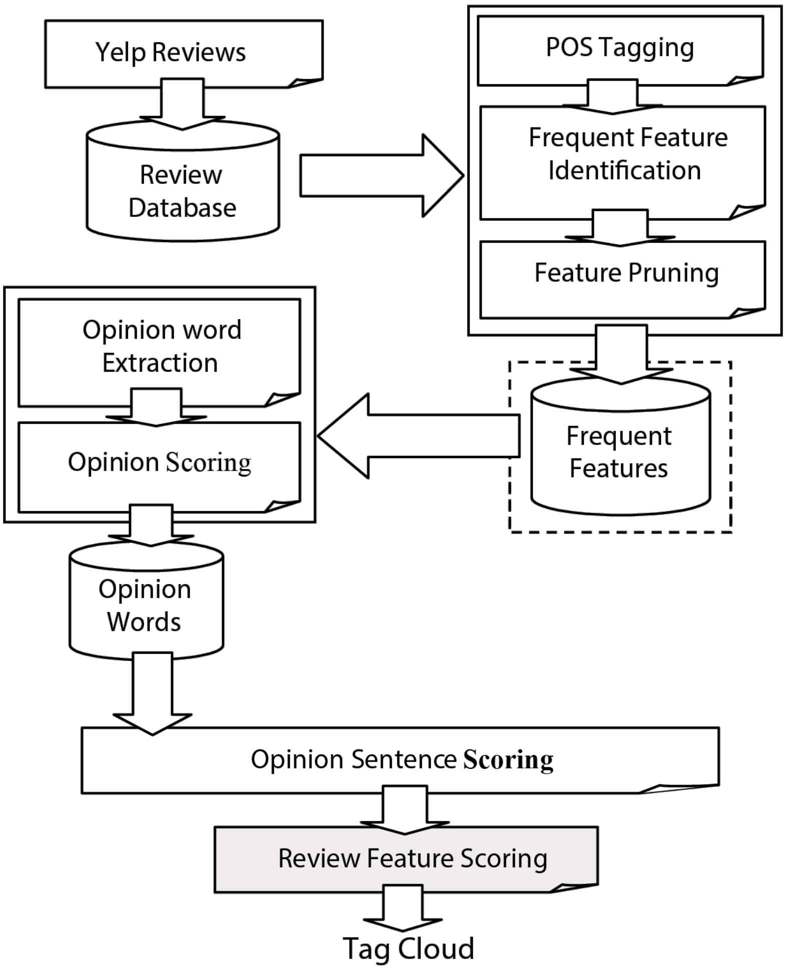


Figure: Feature -based opinion summari zation

1. Review Feature Scoring

The figure on the left explains our process and the details are explained below:

1. We first mine the Yelp data for restaurant reviews and cluster all the reviews by restaurant. We then select the Top 50 restaurants based on review count as our review database.
2. We then run the Part Of Speech(POS) Tagger on the database to identify Noun Phrases. This constitutes the candidate feature set. We then use Apriori algorithm for frequent feature extraction and prune redundant ones based on p-support. This gives us a list of frequent features for each restaurant.
3. We then extract the opinion words (Adjectives) associated with these features and remove invalid words using a English dictionary. We create our own sentiment dictionary that gives a score on a scale of 0-4 for each of these opinion words. Our dictionary is constructed by combining sentiment scores from the below 4 sentiment dictionaries:
   a. A list of positive and negative opinion words(around 6800 words) by Bing Liu [12]
   b. SentiWordnet [13]
   c. Stanford Sentiment Tool [14]
   d. SentiWordnet+ [15]
4. We then compute the sentiment of a sentence for a particular feature as the weighted average of all the opinion words in that sentence where the effective opinion has weight 2 whereas all other words have weight 1. Effective opinion is the word that is closest to the feature and hence considered to have more impact. Also, if a "not" precedes the opinion word then we reverse the sentiment of the opinion word for that sentence.
5. Finally we compute the sentiment score of features as the average of the sentiment of sentences in which the feature appears for that restaurant. The total no of such sentences is defined as the opinion count of the feature. The sentiment scores are then normalized in the range of 0-10 to get the final sentiment value.

6. Resultant feature scores are visualized as a tag cloud.

2. User Rating Prediction for Restaurants using Matrix Factorization

Latent Matrix Factorization tries to explain the ratings by uncovering the latent features of both users and the restaurant.

Formalization:

1. Map both users and restaurant to a joint latent factor space of dimensionality k.
2. Each item i is associated with a vector qi and each user u is associated with a vector pu. qi measures the extent to which the item possesses those factors. p_u measure the extent of interest the user has in items.
3. The resulting dot product $q_i^T p_u$ captures the rating interaction r_{ui} between user and item.

$$\hat{r}_{ui} = q_i^T p_u.$$

We use matrix factorization to produce two matrices P(nxf) and Q(mxf) that represent the latent factors of users and business respectively. To handle the high portion of missing values caused by sparseness in the user-business rating-matrix, we learn the vectors p_u and q_i only on the set of known ratings:

$$\min_{q^*,p^*} \sum_{(u,i)\in\kappa} (r_{ui} - q_i^T p_u)^2 + \lambda(\| q_i \|^2 + \| p_u \|^2)$$

We use Stochastic Gradient Descent (SGD) and Alternating Least Squares (ALS) to perform the above minimization. ALS operates by starting with a random guess for the users (or business) and then computes a least square step to compute the business (or users) alternating between the two. SGD works by starting with a random guess and computing the gradient of the cost function and making a step in the direction of the gradient.

$$q_i \leftarrow q_i + \gamma \cdot (e_{ui} \cdot p_u - \lambda \cdot q_i)$$
$$p_u \leftarrow p_u + \gamma \cdot (e_{ui} \cdot q_i - \lambda \cdot p_u)$$

Typically, collaborative filtering data exhibits large systematic tendencies for some users to give higher ratings than others and for some items to receive higher ratings than others. These are called the user-bias b_u and item-bias b_i. They are approximated as follows:

$$b_{ui} = \mu + b_u + b_i$$

The estimation becomes:

$$\hat{r}_{ui} = \mu + b_i + b_u + q_i^T p_u$$

The system learns by minimizing the squared error function

$$\min_{b_*,q_*,p_*} \sum_{(u,i)\in\mathcal{K}} (r_{ui} - \mu - b_i - b_u - q_i^T p_u)^2 + \lambda_4 (b_i^2 + b_u^2 + \|q_i\|^2 + \|p_u\|^2)$$

This is called Biased-SGD and the gradient descent rule becomes as follows:

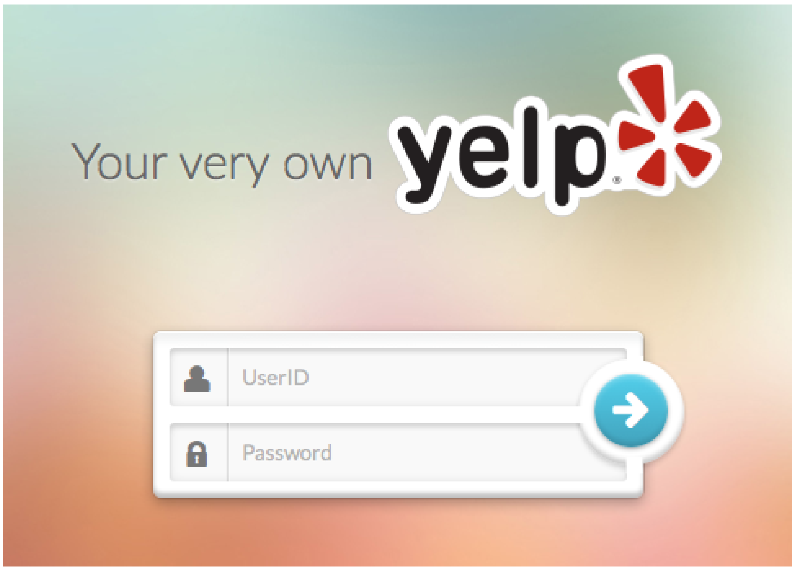
$$
\begin{aligned}
b_u &\leftarrow b_u + \gamma \cdot (e_{ui} - \lambda_4 \cdot b_u) \\
b_i &\leftarrow b_i + \gamma \cdot (e_{ui} - \lambda_4 \cdot b_i) \\
q_i &\leftarrow q_i + \gamma \cdot (e_{ui} \cdot p_u - \lambda_4 \cdot q_i) \\
p_u &\leftarrow p_u + \gamma \cdot (e_{ui} \cdot q_i - \lambda_4 \cdot p_u)
\end{aligned}
$$

When we combine Bias-SGD with a neighbourhood Model(one that finds the neighbourhood of users/items using some similarity measure), we get SVD++. Its minimization equation is as follows:

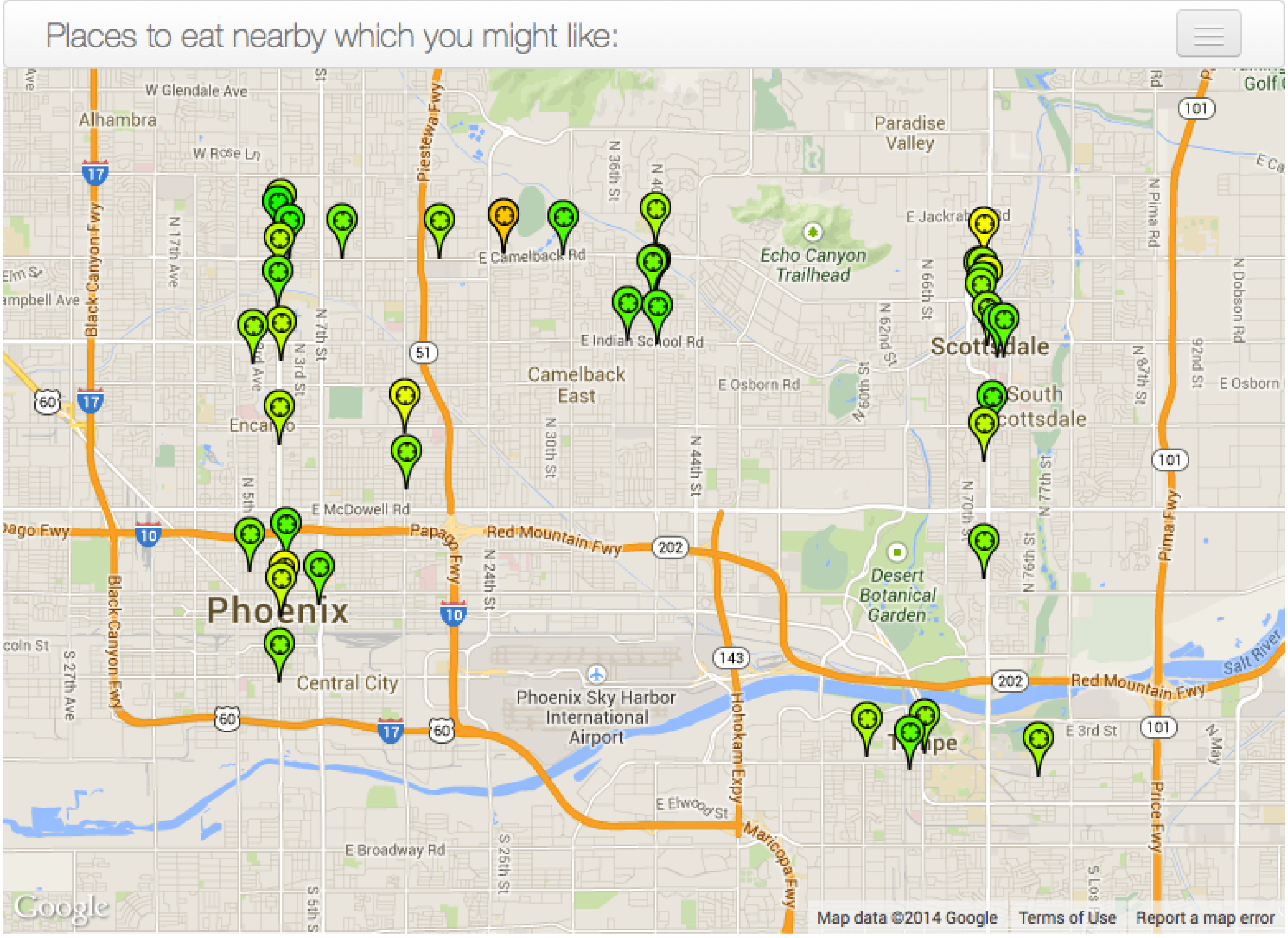
$$
\hat{r}_{ui} = \mu + b_u + b_i + q_i^T \left(p_u + |\mathrm{N}(u)|^{-\frac{1}{2}} \sum_{j \in \mathrm{N}(u)} y_j \right)
$$

$$
+ |\mathrm{R}^k(i;u)|^{-\frac{1}{2}} \sum_{j \in \mathrm{R}^k(i;u)} (r_{uj} - b_{uj}) w_{ij} + |\mathrm{N}^k(i;u)|^{-\frac{1}{2}} \sum_{j \in \mathrm{N}^k(i;u)} c_{ij}
$$

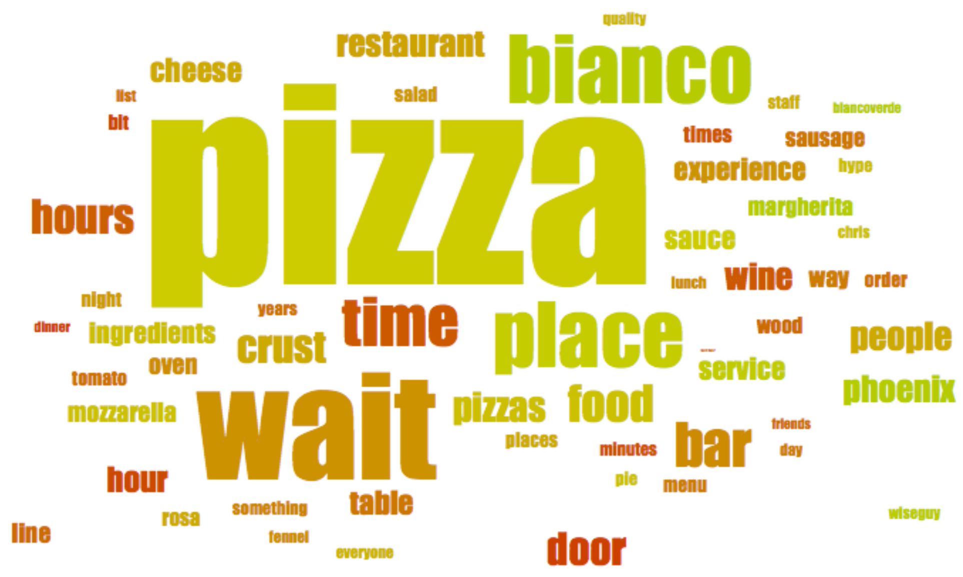
3. User Interface



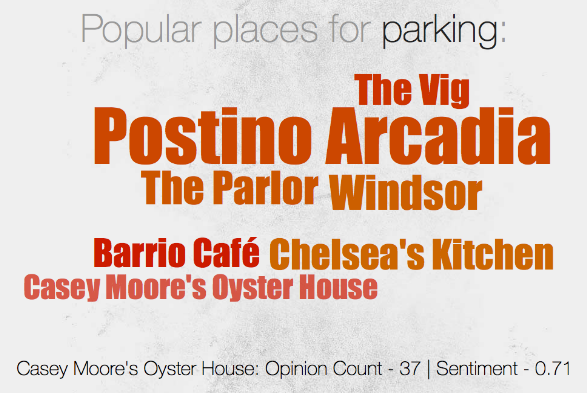
Since we want to show the user a meaningful way to interact with our results, we combined our various analysis to build a user-personalised portal of yelp. Once a user logs in, based on his current location, we show them nearby restaurants that we recommend. Based on users previous restaurant ratings, our algorithm predicts a rating for each restaurant that are shown on the map to the user. The color of the marker indicates the rating so the user can quickly see which restaurant he is probable to like and which restaurants are a big no-no for him.The color scale ranges from red to green where red denotes 0 and green denotes 5 star rating.



Places to eat nearby which you might like:



Once the user selects a particular restaurant, we show them a feature cloud pertaining to that restaurant. The opinion count determines the size of the features, so bigger words are the most talked about features. The sentiment score determines the color of the feature, so user can quickly see the best and worst features of a restaurant. The color scale ranges from red to green where red denotes 0 and green denotes 10. The user can hover over any feature to see the exact sentiment score and opinion count. The user can also navigate in the cloud to find places popular about a feature and vice versa.


Popular places for parking:

The Vig
Postino Arcadia
The Parlor Windsor
Barrio Café Chelsea's Kitchen
Casey Moore's Oyster House

Casey Moore's Oyster House: Opinion Count - 37 | Sentiment - 0.71

Evaluation

Testbed

- ➔ Data
    - Yelp Dataset from the greater Phoenix, AZ metropolitan area including:
        - 15,585 businesses (5556 Restaurants)
        - 70,817 users
        - 335,022 reviews
    - Total Size - 353 MB JSON file
    - Top 50 Reviewed Restaurants with total 26,746 reviews
    - 535 Average Reviews/Restaurant

- ➔ System Environment
    - Sentiment analysis was done on Windows v7 Platform with Java v7 as the primary runtime engine
    - Matrix Factorization was done on Ubuntu 13.10- 64 bit using the Graphlab Collaborative Filtering Toolkit
    - The Visualization is modeled to work with HTML5, CSS3, and JavaScript 1.8, making it compatible with recent versions of Mozilla Firefox, and Chromium.

Research Question

1. What is the best minimum support for Apriori for our dataset.
2. What sentiment values to use for opinion words.
3. How can reviews be summarized effectively.
4. Improving Prediction Accuracy.

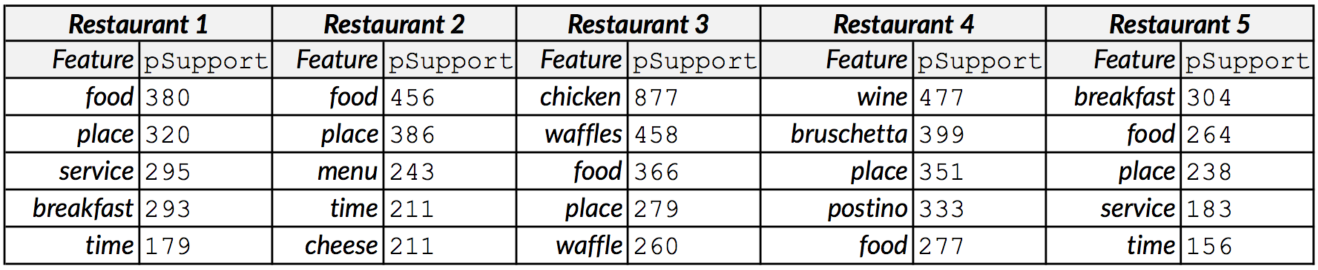
Experiments

1. Determining minimum support for Apriori

We used Apriori algorithm to get the Frequent features. We tried different minimum support values as shown in below table. We decided to set Minimum support to 10% of all reviews of that restaurant as this generates a feature set with a reasonable # of unigrams & bigrams.

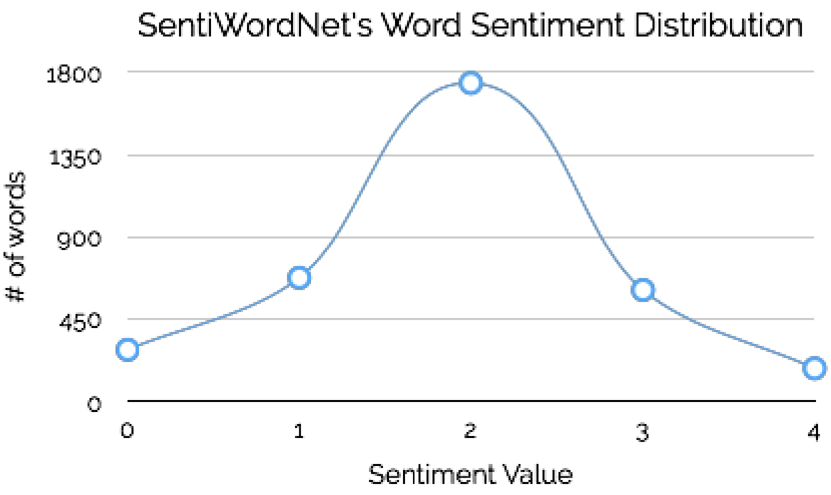
Min _ Support	5%	10%	20%
Unigram Avg	121.5	52.24	17.3
Unigram Std Dev	19.2637	10.15	4.428318
Bigram Avg	8.26	2.5	0.76
Bigram Std Dev	4.581746	2.03224	0.788923

We finally prune out redundant features using pure-support of 10% which gives us an average of 51.88 features per restaurant with a 9.79 std dev. The below table shows the Top 5 features of 5 restaurants with their p-support values. As seen food and place features appear in all 5.



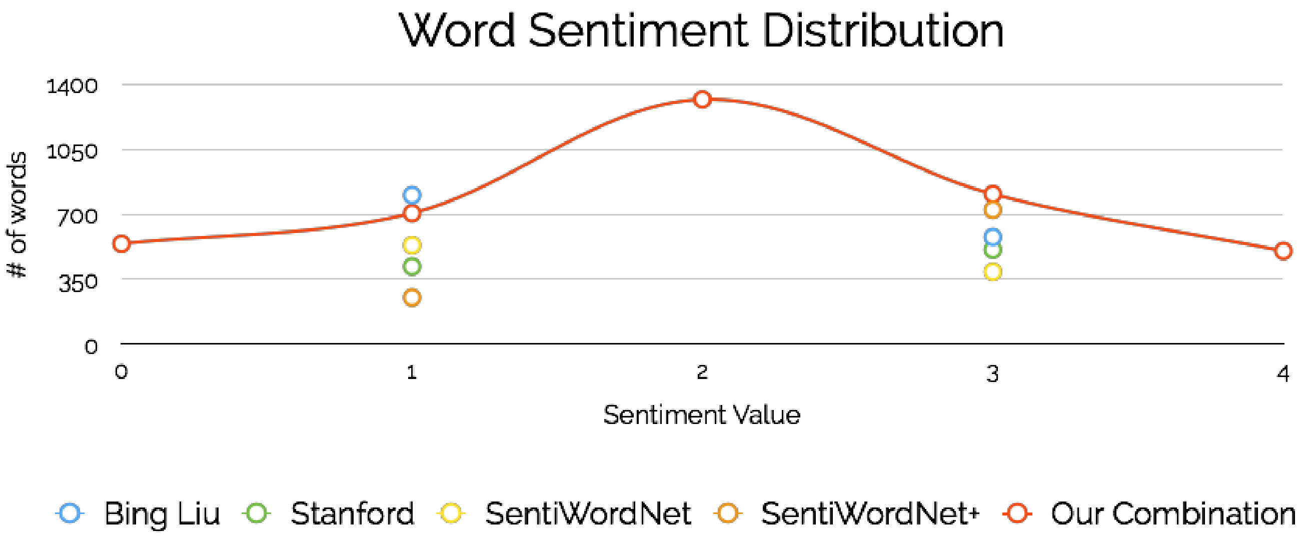
Restaurant 1		Restaurant 2		Restaurant 3		Restaurant 4		Restaurant 5	
Feature	pSupport	Feature	pSupport	Feature	pSupport	Feature	pSupport	Feature	pSupport
food	380	food	456	chicken	877	wine	477	breakfast	304
place	320	place	386	waffles	458	bruschetta	399	food	264
service	295	menu	243	food	366	place	351	place	238
breakfast	293	time	211	place	279	postino	333	service	183
time	179	cheese	211	waffle	260	food	277	time	156

2. Determining Opinion Word Strength

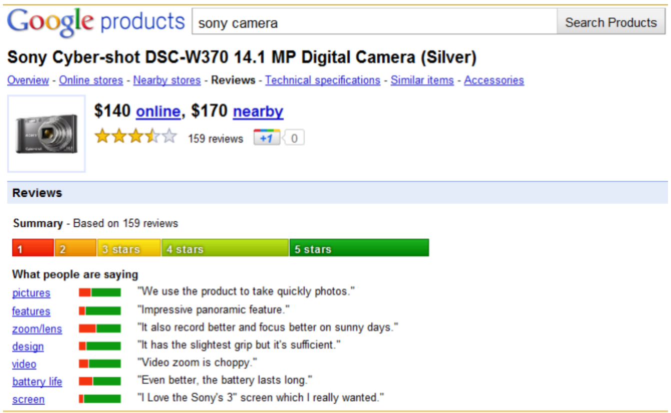


We initially tried using the sentiment values from only SentiWordnet. We basically classified the values into 5 classes as shown in the graph on the right. However, we were not getting good results and there were several mis-classifications.

If we used only one of of the existing sentiment dictionaries we were getting on average sentiment of just 1053.25 opinion words. So we decided to combine the sentiment polarity from 4 existing dictionaries If a word has polarity more than 0.3 in the Sentiwordnet or Sentiwordnet+ dictionary then it is considered to be strongly positive/negative and is used to strengthen the opinion strength. That is, if a word is positive in just one dictionary then it is scored 3, but if it is positive in two or more dictionaries then it is scored 4. Similarly, if a word is negative in just one dictionary then it is scored 1, but if it is negative in two or more dictionaries then it is scored 0. A word not found in any of these dictionaries is considered to be neutral and scored 2. This way we constructed our own sentiment dictionary of 3882 words that gives sentiment scores not just sentiment orientation.



3. Effectiveness of Review Summarization Visualization



Google products | sony camera | Search Products

Sony Cyber-shot DSC-W370 14.1 MP Digital Camera (Silver)

Overview - Online stores - Nearby stores - **Reviews** - Technical specifications - Similar items - Accessories

$140 online, $170 nearby

★★★½☆ 159 reviews +1 0

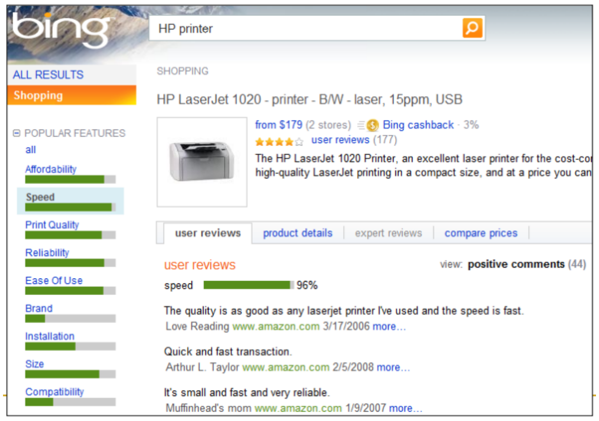
Reviews

Summary - Based on 159 reviews

1 | 2 | 3 stars | 4 stars | 5 stars

What people are saying

pictures	"We use the product to take quickly photos."
features	"Impressive panoramic feature."
zoom/lens	"It also record better and focus better on sunny days."
design	"It has the slightest grip but it's sufficient."
video	"Video zoom is choppy."
battery life	"Even better, the battery lasts long."
screen	"I Love the Sony's 3" screen which I really wanted."



HP printer

ALL RESULTS

Shopping

POPULAR FEATURES

all

Affordability

Speed

Print Quality

Reliability

Ease Of Use

Brand

Installation

Size

Compatibility

SHOPPING

HP LaserJet 1020 - printer - B/W - laser, 15ppm, USB

from $179 (2 stores) Bing cashback · 3%

★★★★☆ user reviews (177)

The HP LaserJet 1020 Printer, an excellent laser printer for the cost-co
high-quality LaserJet printing in a compact size, and at a price you can

user reviews | product details | expert reviews | compare prices

user reviews view: **positive comments** (44)

speed 96%

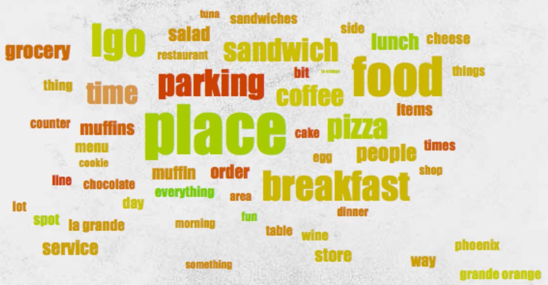
The quality is as good as any laserjet printer I've used and the speed is fast.
Love Reading www.amazon.com 3/17/2006 more...

Quick and fast transaction.
Arthur L. Taylor www.amazon.com 2/5/2008 more...

It's small and fast and very reliable.
Muffinhead's mom www.amazon.com 1/9/2007 more...

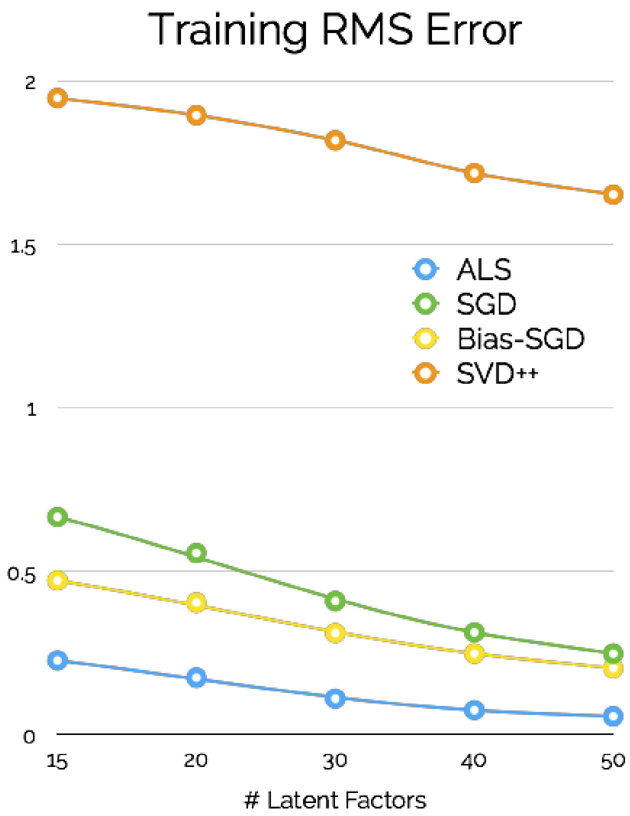
We compare our tag cloud visualization with existing review summarization techniques from Top companies. Our visualization conveys more information because we provide opinion strength and not just opinion polarity. Also because opinion count determines the size of features in tag cloud, user can quickly see most talked about features.

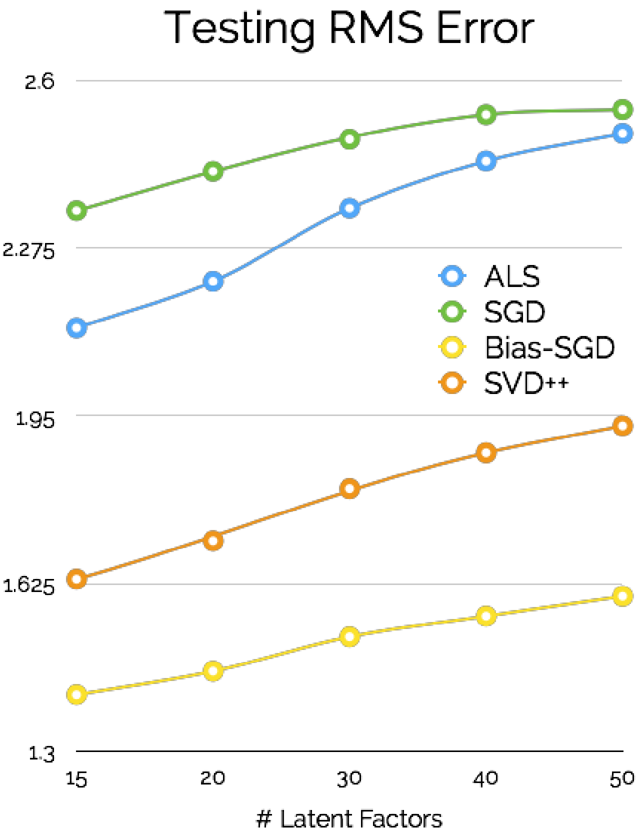
Salient features about La Grande Orange Grocery:



time: Opinion Count - 144 | Sentiment - 3.17

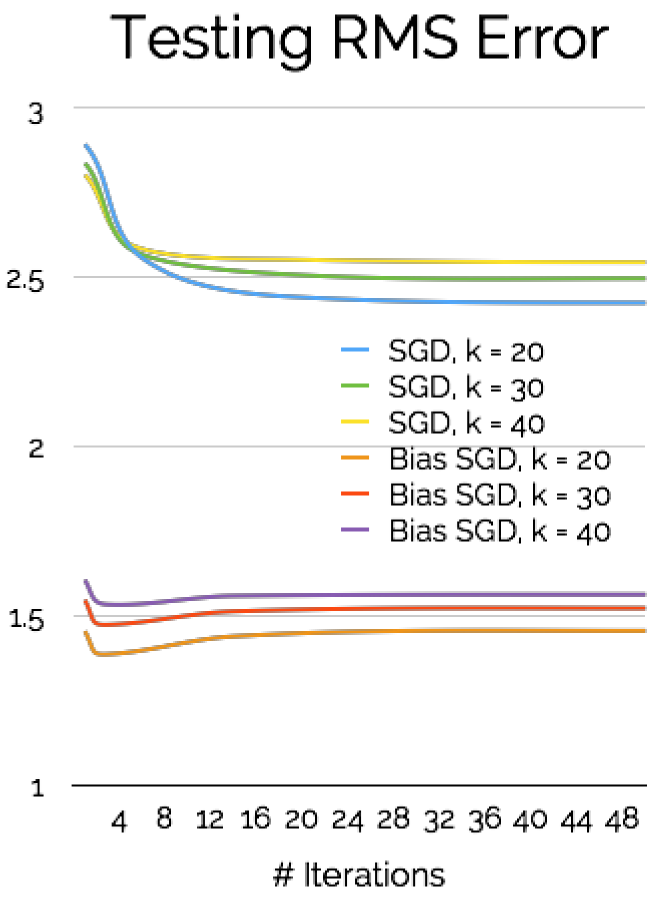
4. Improving Prediction Accuracy

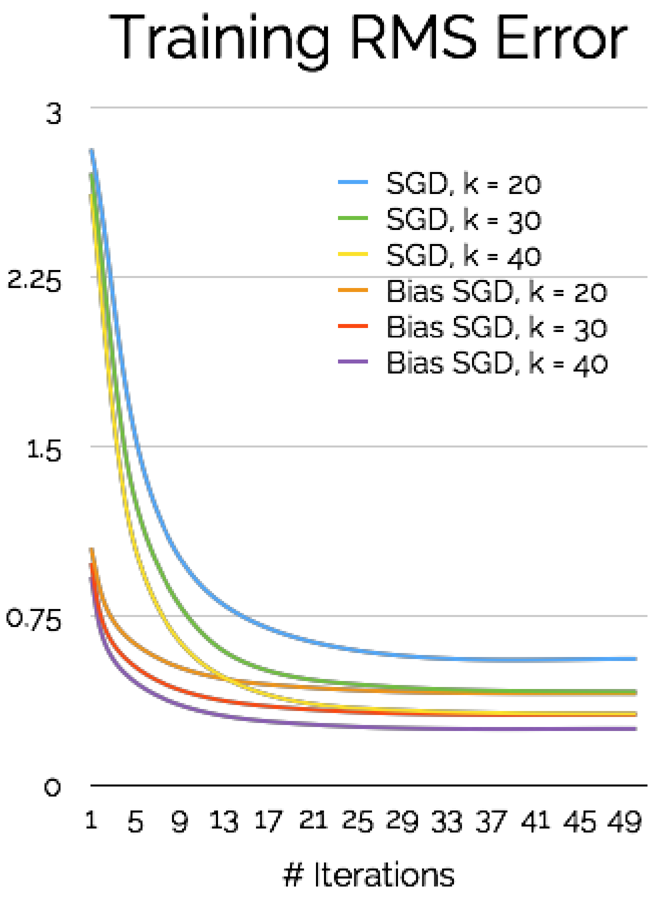




From the above diagram, we can see that Bias-SGD achieves much better results than SGD and ALS. This is so because, Bias-SGD models the user-bias and item-bias as a part of its factorization step. So, it runs gradient descent on not just the latent-factors, but also the biases, resulting in better fit of the underlying sparse data. The SVD++ has been proven to be more accurate method than others once tuned. But in our case, we got lower results than Bias-SGD. SVD++ uses implicit information to improve its performance over that of the other algorithms. The lack of this information for our datasets could be one of the reasons of it performing worse than Bias-SGD although it did manage to beat SGD by large margin.

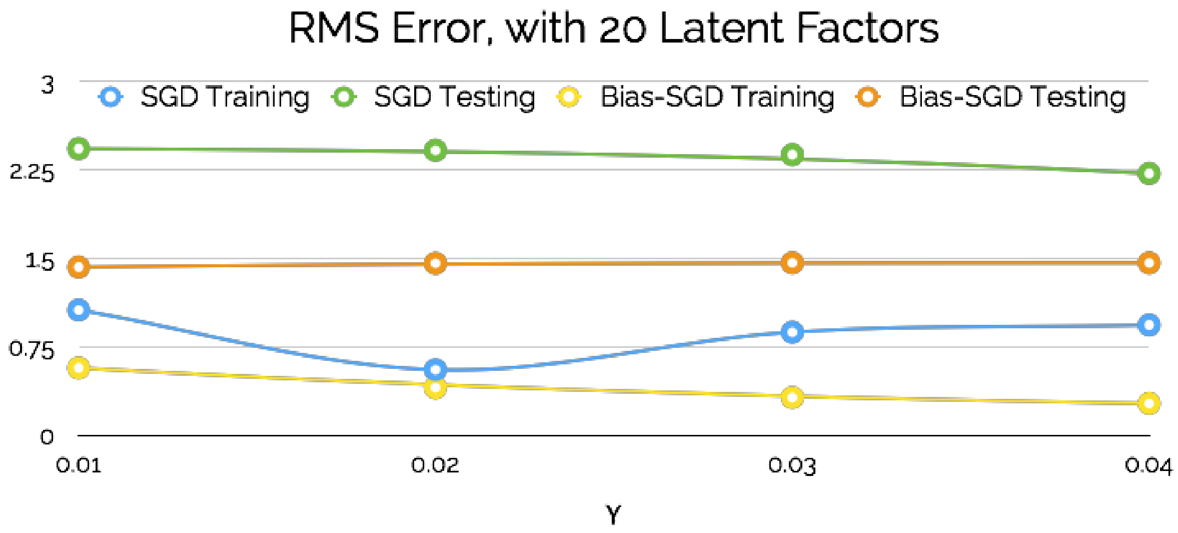
Some common trends for all the algorithms.
1. As the number of Latent factors increases, the training RMSE decreases. This seemed natural since more latent factors can model the data better.
2. However, the testing error seemed to decrease on increasing the number of latent factors. This seemed strange because more latent factors lead to better prediction accuracy on the test set.





Some observations from the above diagram:

1. Testing and the training RMSE errors drop quickly and then plateau out as the number of iterations increase.
2. The fall in the training error is much sharper than testing error.
3. After a certain number of iterations, there is a minor increase in the testing error while the training error flattens out. We believe that this is the best stopping point for all methods.



The main problem with Bias-SGD, SGD and SVD++ is that they don't used the closed form equations for performing gradient descent like ALS. So, they require more iterations than

ALS to get to the same error value. But, they require less computation overhead per iteration, so end up having lesser running-times than ALS.

The learning rate γ plays a very important role in final results of all algorithms. Selecting a large γ resulted in ups and downs in the training error for SGD while a fairly stable decrease in training error for Bias-SGD. Selecting a small γ resulted in the training error not going down enough even after several iterations. We found the value of $\gamma = 0.02$ to be the best for both SGD and Bias-SGD. Very large γ values caused numerical errors in Bias-SGD and SGD while SVD++ seemed immune to these errors.

Conclusions and Discussion

Given more time, we would have liked to improve the sentiment analysis component to account for comparative opinions. For sentence *"Burgers are much better than pizzas"*, our algorithm associates the sentiment of opinion word better to both burgers and pizzas. The ideal way would be associate more positivity to burgers as compared to pizzas. Also our current algorithm cannot identify implicit features. For sentence *"It easily fits in pocket"*, the algorithm cannot interpret that user is talking about size. This a very hard problem in NLP with no successful solution as yet.

Overall, the review summarization technique of this project not only makes it easy for users to quickly gain insight from other users experiences but also helps businesses learn and improve from user comments without the need to have separate feedback mechanisms to capture this information.

Bibliography

[1] Y. Koren and R. Bell. Advances in collaborative filtering. In Recommender Systems Handbook. Springer, 2011.

[2] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. Computer, 2009.

[3] I. Titov and R. McDonald. A joint model of text and aspect ratings for sentiment summarization. In ACL, 2008.

[4] W. Zhao, J. Jiang, H. Yan, and X. Li. Jointly modeling aspects and opinions with a MaxEnt-LDA hybrid. In EMNLP, 2010.

[5] G. Ganu, N. Elhadad, and A. Marian. Beyond the stars: Improving rating predictions using review text content. In WebDB, 2009.

[6] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. JMLR, 2003.

[7] D. Blei and J. McAuliffe. Supervised topic models. In NIPS,2007.

[8] Julian McAuley and Jure Leskovec. Hidden Factors and Hidden Topics: Understanding Rating Dimensions with Review Text in ACM RecSys '13.

[9] Minqing Hu and Bing Liu. Mining Opinion Features in Customer Reviews. In AAAI'04 Proceedings of the 19th national conference on Artifical intelligence.

[10] Agrawal, R. & Srikant, R. 1994. Fast algorithm for mining association rules. VLDB'94, 1994.

[11] Bryan Hood, Victor Hwang and Jennifer King. Inferring Future Business Attention.

[12] Minqing Hu and Bing Liu. Mining and Summarizing Customer Reviews. In KDD '04 Proceedings of the tenth ACM SIGKDD.

[13] Esuli, Andrea, and Fabrizio Sebastiani. "Sentiwordnet: A publicly available lexical resource for opinion mining." Proceedings of LREC. Vol. 6. 2006.

[14] Socher, Perelygin, Wu, Chuang, Manning, Ng, Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In EMNLP, 2013.

[15] Li, Thakkar, Wang, Riedl. Data-Driven Alibi Storytelling for Social Believability . In FDG Workshop,2014