1 Formulation

I employ the LASSO regularization. Namely, I solve this problem

$$\min_{\beta} \| X\beta - y \|^2_2 + \lambda \| \beta \|_1$$

where $X$ is a $m \times n$ data matrix (with each row a specific data point or sample, each column a specific feature, or term(s)), $y$ is a $m-$dimensional response vector, and $\lambda > 0$ is a parameter.

The $l_1$-norm penalty encourages the regression coefficient vector $\beta$ to be sparse, bringing interpretability to the result. If $\lambda$ is large, then the optimal $\beta$ will be driven to be very sparse, and the LASSO model tends to allow only a few (nonzero) features that are the best predictors of the response vector.

I use gradient descent to solve for an exact solution for feature selection each time.

2 Data

At the moment people decided to purchase books, they had been biased towards giving a higher rate; namely, they spent money on the books because they were supposed to like them.

Out of a total of 975, 194 Amazon book reviews, 633, 280 (64.94%) are rated at 5, 218, 015 (22.36%) rated at 4, 55, 975 (5.74%) rated at 2, and 67, 924 (6.97%) rated at 1. In other words, 87.3% of the reviews are rated as positive, while only 12.7% are rated as negative. This can potentially lead to a false positive problem in prediction.

This is a rich dataset with very detailed XML-tags. Particularly, I re-tokenized the texts from the un-processed dataset, and specifically reserved the tags on

- reviewer name
- reviewer geographical location
- helpful index
- time-stamp, and
- book title,

other than just “rating” versus “reviewer texts” which are required by the homework.
The remainder of my write-up is structured as follows: in Section 3, I examine the relationship between rating and possible-sentiment words from reviewer texts, and verify the quality of the model by performing a 10-fold cross validation. In Section 4, I investigate and try to calibrate the relationship between rating and particular reviewers, books, geographical locations. I visualize the over-time evolution of representative sentiment words for each rating in Section 4. Finally, I conclude in Section 5.

3 Model Selection and Validation

I build up four models and verify their performances.

1. unigram, without stop-words removal
2. unigram, with stop-words removal
3. unigram, without stop-words removal, but with TF-IDF re-scaling
4. unigram, with TF-IDF re-scaling, and then with stop-words removal.

Additionally, I calculated the top-weighted words associated to each rating, based on the 4th model:

Rating 5: great, excellent, best, wonderful, must, love, read, highly, loved, life, easy, books, anyone, recommend, ever, amazing, beautiful, favorite, well
Rating 4:
good, bit, overall, interesting, however, liked, pretty, somewhat, nice, enjoyed, stars, enjoyable, complaint, although, ending, fairly, though, nevertheless, solid, slightly

Rating 2:
disappointing, boring, disappointed, disappointment, ok, poorly, unfortunately, mediocre, seemed, predictable, tedious, didn’t, weak, nothing, hoping, dull, confusing, okay, expected, lacking, lacks, annoying

Rating 1:
waste, money, boring, worst, poorly, bad, awful, disappointing, useless, nothing, terrible, garbage, disappointment, disappointed, horrible, trash, poor, save, stupid, bother, zero, worthless

Although top-weighted words make sense intuitively subject to each rating, the unbalanced data causes a higher false positive rate. The size of the positive class to the negative class is $87.3 : 12.7 \approx 6.9 : 1$. Hence, when it comes to a 10-fold cross-validation, it’s not possible to build a balanced training model no matter how we carefully select the training samples.

4 In-depth look into the dataset

Some people tend to be harsh when rating products, while other always being sweet. Therefore, when it comes to model selection, identifying the reviewer can be an important first step every time in the prediction. Using the 5-star reviews as a positive class verses the rest, I identify these top-3 sweetest reviewers. The figures inside each parenthesis indicate the number of 5-star reviews over the number of total reviews made by that reviewer:

- midwest book review ($8,030/8,039 = 99.89\%$), marina kushner author of the truth about caffeine ($443/443 = 100\%$), john matlock gunny ($1,743/1,746 = 99.83\%$).

The same scenario applies to other rating, too. Hence, we may put a heavier weight depending on the reviewer giving very unbalanced reviews, to improve accuracy.

In terms of geographical distribution of ratings, I identify top locations from which the reviews tend to receive higher scores:

- oregon wi usa ($8,022/8,032 = 99.88\%$), scr books ($476/476 = 100\%$), new york ($3,790/5,696 = 66.54\%$).
We can also use the same model to single out the products which tend to receive higher/lower rates. 5-star ones:

- a walk to remember books nicholas sparks, in cold blood books truman capote, a short history of nearly everything books bill bryson.

Finally, there is an interesting “helpful” XML-tag in the dataset. I obtain these “helpful” scores for the 5-star:

- 3 of 3, 4 of 4, 5 of 5, 6 of 6, 3 of 4, 7 of 7, 8 of 8.

The scores for the 1-star are:

- 1 of 19, 0 of 15, 1 of 18, 3 of 23, 3 of 19, 1 of 15, 2 of 18, 1 of 27, 1 of 16, 3 of 29, 0 of 20.

The “helpful” scores are strongly positively correlated to the rating scores and may be used as a feature to predict sentiments.

5 Visualizing the evolution of sentiment words

The usage of language evolves over time. This includes the words people use to express sentiments in Amazon book reviews. According to the dataset on the XML-date tag, the reviews span from June 20, 1995 to April 28, 2007.

I set a window size to be 3-month, and calculated top 10 words for each window. I select top two words from each window to form a final list. Then, I sort the list by their first appearances over time, to visualize the temporal evolution of sentiment words coming in. Finally, we draw the heat map with the block intensity based on the strength of association of every particular word to that rating. The sorting yields an upper-triangle heat map.

The visualization is coded up using D3.js - a Javascript visualization library developed by Stanford HCI group.

URL links to the online interactive version:

http://atticus.berkeley.edu/guanchengli/showcase/Amazon_Rating_5/
http://atticus.berkeley.edu/guanchengli/showcase/Amazon_Rating_4/
http://atticus.berkeley.edu/guanchengli/showcase/Amazon_Rating_2/
http://atticus.berkeley.edu/guanchengli/showcase/Amazon_Rating_1/
Figure 1: Amazon Review Rating 5: top weight words

Figure 2: Amazon Review Rating 4: top weight words
Figure 3: Amazon Review Rating 2: top weight words

Figure 4: Amazon Review Rating 1: top weight words
6 Conclusion

- I wanted to build a bigram model but haven’t been able to work out. Re-tokenizing and re-building the huge matrix for bigram features takes forever. My hardware suffers, too. I hope to discover efficient ways of doing this by MapReduce later in the semester.

- The top weight words used for the 5-star reviews are stable – they happened to recur a lot of times, including “best”, “great”, “wonderful”, “excellent”, “highly”.

- The top weight words used for the 1-star reviews are less stable – the most recurring ones are “waste”, “boring”, “terrible”, “money”.

- The top weight words used for the 2-star and 4-star reviews are basically in-and-out – they change over time. For the 4-star, the recurring ones are only “good”, “bit”, “overall”. The 2-star contains “disappointing”, “disappointed”. Other than this, the top words are very product-specific. Namely, people are more likely to point out product names in 2/4-star, while expressing strong sentiments in 1/5 star.

- Future work requires elaborating the parameter usage of timestamp, geographical location of the reviewer, the cynical level of the reviewer, and helpful scores towards a more accurate model for sentiment analysis.