

Efficient Estimation of Aspect Weights

Jon Parker, Andrew Yates, Nazli Goharian
Department of Computer Science
Georgetown University

{jon, andrew, nazli}@ir.cs.georgetown.edu

Wai Gen Yee
Orbitz Worldwide
Chicago IL, 60661

waigen.yee@orbitz.com

ABSTRACT

Many websites encourage people to submit reviews of various products and services. We present and evaluate a novel approach to efficiently model and analyze the text within user reviews to estimate how much reviewers care about different aspects of a product (i.e., amenities, food, location, room, etc. of a hotel). Our approach performs statistically quite similar to the best existing method. However, our method for computing aspect weights is a linear time method while the current state of the art solution requires cubic time at best.

Categories and Subject Descriptors

I.2.7 [Natural Language Processing]: Text Analysis

General Terms

Algorithms, Performance, Experimentation

Keywords

Aspect ranking, opinion and sentiment analysis, review mining.

1. INTRODUCTION

The proliferation of websites that collect user reviews has increased the value of a methodology that can mine these reviews for customer preference data. The Latent Aspect Rating Analysis (LARA) method introduced in [1] and expanded in [2] estimates aspect weights, (i.e. customer preferences) from user reviews. This is the only work we are aware of, besides the work of Xu et al. [3], which tries to determine what customers think about aspects. LARA assumes a review's overall star rating is a linear combination of the review's aspect star ratings. The coefficients in this linear combination represent the weights a reviewer places on the aspects. Given this assumption, a latent rating regression is performed to simultaneously estimate each individual review's aspect ratings as well as the underlying aspect weights using just the individual review's text and overall rating.

We present a significantly more efficient method for estimating aspect weights that generates very similar results. The Force of Commentary (FoC) method introduced here models the impetus that prompts people to write the text portion of a review. We hypothesize that this impetus is well modeled as a mixture of what users are surprised about and what users care about. FoC estimates customer preference using solely the text within reviews. This enables FoC to be used when ordinal rating data (e.g., star rating) is not reliable or not available. While we perform our experiments on a hotel review data set, our method is not domain-specific and should work for other types of reviews.

2. METHODOLOGY

2.1 The Force of Commentary Curve

We assume people write about aspects they find important and/or surprising. This assumption suggests a generic functional form for what we call the Force of Commentary (FoC) curve. For each aspect we have the generic equation:

$$E[n_i] = w_i \cdot \int Prob(x_i) \cdot Surprise(x_i) dx_i$$

where $E[n_i]$ is the expected value of n_i , the number of comments written about aspect i , w_i is the importance of aspect i , $Prob(x_i)$ is the probability a hotel provides an x_i level of service, and $Surprise(x_i)$ is a function that specifies how much a customer notices receiving an x_i level of service. For convenience, we require $Surprise$ to be decreasing from $-\infty$ to 0, increasing from 0 to ∞ , and equal to 0 at 0. This requirement ensures that the above integral can be cleanly split into one integral (from 0 to ∞) that corresponds to positive comments and a second integral (from $-\infty$ to 0) that corresponds to negative comments.

Assuming for simplicity that the performance level x_i is Normal(μ_i , 1) and $Surprise$ is $abs(x_i)$, the equation becomes:

$$E[n_i] = w_i \cdot \frac{1}{\sqrt{2\pi}} \cdot \int e^{-\frac{(x-\mu_i)^2}{2}} \cdot |x_i| dx_i$$

2.2 Processing the Hotel Review Data

The Orbitz hotel review dataset contained 609,884 individual human authored reviews that rated 30,621 hotels. Each review contained an overall experience "star rating", multiple aspect "star ratings", and a review text. The review text typically contains 1 to 5 sentences that detail the customer's thoughts about a hotel property.

Processing the reviews begins by gathering the text portion from every review of a specific hotel property into one large macro-review. Macro-reviews that do not contain 100 sentences are thrown out because they do not contain enough information to estimate our 6 selected aspect weights well.

We selected 6 aspects which we believe are sufficient to characterize hotels and whose themes occur frequently in the reviews. They are food, room, cleanliness, amenities, value, and location. These aspects correspond closely with the aspect ratings available in our dataset as well as the aspects used by Wang et al [1].

We used the χ^2 bootstrapping method described in [1] to expand a list of seed words used to determine which aspect each sentence in a macro-review is addressing. Once a sentence is associated with an aspect its polarity is determined using SentiWordNet[4]. The sentence's polarity is the polarity of the majority of its words. We discard neutral sentences.

To compute the aspects' weights, we begin by finding the μ_i values for which the ratio between the expected number of

positive comments and the expected number of negative comments matches the ratio between the observed number of positive comments and the observed number of negative comments. Next, we find the vector of w_i values, for which the relative distribution between the computed $E[n_i]$ values matches the observed distribution of comments. The computed weight vector contains a valuable synthesis of information about the average hotel customer's preferences and expectations as well as what the hotel actually provides.

Notice, the method described herein does not make use of either the overall "star rating" or any of the aspect "star ratings". Relying on ratings presents two problems. First, a non-trivial fraction of reviewers seem to misinterpret the star rating scale. These reviewers write scathing remarks and then leave 5-star ratings, and vice versa. Second, humans vary on how generously they award stars. One reviewer's 5-star experience could be another reviewer's 3-star experience. This bias is well-known in user review research [5]. Ignoring human assigned ratings also allows this method to be deployed where numeric or ordinal ratings are unavailable.

3. RESULTS & EVALUATION

The aspect weight vector was computed twice for each of the 111 hotels with a macro-review of at least 100 sentences. The first weight vector was computed using the Force of Commentary method outlined in Section 2. The second weight vector was computed using the LARA method from Wang et al. [1,2] as the state-of-the-art.

Table 1. Average Aspect Weight Vector by Method

| | Food | Room | Cle. | Ame. | Val. | Loc. |
|------|------|------|------|------|------|------|
| FoC | 0.06 | 0.34 | 0.36 | 0.11 | 0.02 | 0.11 |
| LARA | 0.11 | 0.27 | 0.23 | 0.14 | 0.11 | 0.14 |

The average aspect weight vector for all 111 hotels is shown in Table 1. This table suggests the aspect weights computed using FoC are similar to the aspect weights computed using LARA. The correlation coefficients shown in Table 2 confirm a relationship does indeed exist. In fact, the median Spearman's rank correlation coefficient is .829 and the mean Spearman's rank correlation is .699. Given these correlations coefficients, it is clear that these methods are generating answers that are well within the same ballpark.

Table 2. Pearson Correlation Coefficients Between FoC and LARA Aspect Weights

| Food | Room | Cle. | Ame. | Val. | Loc. |
|------|------|------|------|------|------|
| .614 | .567 | .534 | .617 | .217 | .733 |

All 111 hotels were clustered into 6 groups according to their FoC aspect weights. The goal of this clustering was to determine if hotels with similar customer populations were assigned to the same cluster. After all, if FoC aspect weights do indeed characterize customer preferences then this regularity should appear. As expected, 4 of the 6 clusters had clear hotel tier effects. For example, one cluster contained high-end hotels like the ARIA and Wynn (both of Las Vegas) while another cluster contained Quality Suites, Holiday Inn, Ramada, and Best Western

chains. The 2 remaining clusters contained hotels with a significant number of cleanliness complaints and location praise respectively.

A human evaluation of the FoC clusters was also performed. However, there was very little agreement among the evaluator comments. We believe this is due to the subjectivity and high dimensionality of the task.

4. EFFICIENCY ANALYSIS

LARA includes a non-linear optimization problem that is solved using a conjugate gradient interior point method. Applying this method to a linear programming problem requires $O(n^3/\log(n))$ bit operations. LARA is actually a non-linear optimization problem, thus its computational complexity is higher. We ignore this fact because we can show that FoC is more efficient than the linear optimization problem.

FoC consists of three separate stages performed in succession: sentence aspect assignment, sentence polarity assignment, and weight vector computation. All three rely on lookup tables that can be computed offline. We do not include the cost of computing these tables when determining the final computational complexity.

Both the sentence aspect assignment and sentence polarity assignment stages of FoC require $O(n)$ time (where n is number of words in all the macro reviews) due to the table lookups within each stage. The final weight computation stage is linear in number of macro-views. Thus, the $O(n)$ stages of FoC dominate the computation complexity and FoC is $O(n)$.

Computing the FoC weights listed in the paper took less than 5 minutes while computing the LARA weights took 36 hours.

5. CONCLUSION

We demonstrated that the FoC aspects weights are: (1) vastly more efficient to compute than LARA weights, with FoC's computation taking linear time and LARA's computation taking cubic time (2) statistically very similar to LARA weights and (3) accurate enough to generate hotel clusters that appear natural.

6. REFERENCES

- [1] H. Wang, Y. Lu, and C. Zhai. 2010. Latent aspect rating analysis on review text data: a rating regression approach. In KDD '10. ACM, New York, NY, USA, 783-792.
- [2] H. Wang, Y. Lu, and C. Zhai. 2011. Latent aspect rating analysis without aspect keyword supervision. In KDD '11. ACM, New York, NY, USA, 618-626.
- [3] J. Xu, Z. Zha, M. Wang, and T. Chua. 2011. Aspect ranking: identifying important product aspects from on-line consumer reviews. In *ACL '11*. ACL, Stroudsburg, PA, USA, 1496-1505.
- [4] S. Baccianella, A. Esuli, and F. Sebastiani. 2010. SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In LREC'10. ELRA, Paris, France, 2200-2204.
- [5] Koren, Y. 2009. Collaborative filtering with temporal dynamics. In KDD '09. ACM, New York, NY, USA, 447-456.