An Entropy based Product Ranking Algorithm using Reviews and Q&A Data

Bushra Anjum
Amazon Inc., 1194 Pacific St., San Luis Obispo
CA 93401, USA
banjum@amazon.com

Chaman Lal Sabharwal
Missouri University of Science and Technology, Rolla
MO 63128, USA
chaman@mst.edu

Abstract — **Amazon.com, along with several other commercial websites for products and services, provides a platform for consumers to share their opinions by providing reviews and answering product related questions (QA data). These opinions can be quantitative, qualitative or a combination of both. Owing to the large corpus of such data available, there are several learning and classification approaches available to scrutinize them e.g., those based on Entropy measures, machine learning, stochastic, and natural language processing etc. In this paper, we review some of the prominent techniques and explore a hybrid approach, involving Entropy, Bilinear and statistical measures, to use heterogeneous consumer data and simultaneously analyze and rank products for customers. With experimental results, we show that our approach effectively ranks products using (1) text reviews (2) QA data and (3) star rating of products. We also make a case that the ranks calculated are more relevant to the customers and can enable better prediction on the products sale for the sellers.**

Keywords-product reviews, product ranking, similarity, classification

I. Introduction

Gaining insights from product reviews has emerged as a novel field of research and has valuable implications in the real world. Many e-commerce websites, such as Amazon.com, provide a platform for consumers to share their opinions. Unbiased reviews by other consumers build confidence of a customer to go ahead with a transaction [1]. The reviews are generally quantitative in the form of star rating, or qualitative in the form of comments written in plain text. Due to the large corpus of review data available and limited customer time, researchers are not only focusing on means to ascertain the quality, authenticity, and usefulness of reviews but also on ranking products based on the available data.

It has been ascertained that the product rating with stars (or numerical scale 0-5 etc.) alone does not provide enough semantics information about customer's sentiment and that a text based review is more revealing in that context [2]. Reviews are subjective opinions and judgment about a product or the service. Hence, nowadays the trend is to use both star based ranking and text reviews in all types of surveys pertaining to products and services. However, it is also important to note that though it is quick and easy to process quantitative ratings, producing qualitative semantic information is a challenging problem because the deciphering and evaluation of text reviews is both time-consuming and technically complex due to an unstructured form of natural language text [1].

As mentioned earlier, it is important and necessary to classify the relevance of product related information, for the sake of consumer's premium time [3]. Reviews are a major source of such information. Another interesting area of research, though not a focus of this work, is the availability of search and evaluation tools that helps a customer distinguish between forged and genuine reviews from their implied reputation [4]. In this work, we are assuming that the reviews provided are genuine with varying degree of relevance

There are several approaches used to ascertain the quality, authenticity, and usefulness of the reviews such as Entropy (H), Decision trees (DT), Singular Value Decomposition (SVD), support vector machines (SVM), machine learning (ML), stochastic probability, and natural language processing (NLP) etc. These approaches are explained further in the related work section and help in evaluating the helpfulness of a review, rank products based on relevance from the reviews, enable prediction on the products, and reduce the search time.

DOI reference number: 10.18293/DMS2016-024

In this paper, we will explore a hybrid analysis approach to use heterogeneous product review data (star ranking, text based reviews, question/answer data) to simultaneously analyze and evaluate the reviews. We show that adapting the ranking algorithm by simultaneously using Entropy and Bilinear Similarity measures (explained later in the paper) yields more accurate evaluation than using them in isolation and on restricted data types. With experimental results, we learn that our approach is effective in ranking products based on interest, and relevance and also accurately relies on review text and question/answer data to produce the final ranking of the product.

The paper is organized as follows: Section II describes the preliminary background and literature review, section III describes our contributions in detail, section IV is on experiments and section V lists conclusions followed by references.

II. Background and Related Work

A. Preliminaries

Here we describe all the *standard* terms used in this wok and discussion. Data instance/object is in the form of a vector. We assume all vectors are *column* vectors. A matrix represents the aggregate of all data instances. A *row* vector is the *transpose* of a column vector. A *unit* vector is a vector of length one. A *weight* vector is a vector with positive components whose sum is *unity*. The *dot* product of two vectors is a scalar, *Hadamard* product of two vectors is a vector of point-wise products of the corresponding components of the two vectors. The *similarity* between two *unit* vectors can be defined by *simple* dot product, *weighted* dot product, and *complex* measure Okapi *BM25 (Best Matching 25)* [5], see section II.C. The data matrix format is also called *TF-IDF* format, meaning term-frequency by inverse-document-frequency. *Entropy* is the measure of uncertainty in the classification, where the smaller the value of uncertainty, the better is the resulting classification. Accuracy is measured as: *Precision*, how accurate prediction is over only positive instances, *Recall,* how accurate positive prediction is for only relevant instances, and *F-measure*, which is the *weighted Harmonic mean* of Precision and Recall.

B. Related work

The field of machine learning is vast, challenging and, is increasingly being utilized by non-technical consumers, many of which do not have a basic understanding of the foundational principles. In our case, the task is to make a recommendation based on product reviews. At a high level, there are two kinds of tasks frequently seen in machine learning: *classification*, and *ranking*. There are several tools used to accomplish it, entropy and regression being two of them. For instance, a student in the class gets "A" grade, the letter grade is *classification*, and the numeric grade is determined by *an algorithm* for predicting the letter grade from multiple scores of tests, home works, and quizzes. The relative *ranking* of students in the same grade is determined from scores where the raw score lies in the scores range.

Also, it is a de facto reality that more complex machine learning metric is not necessarily an improved evaluation technology; especially when data is limited. Simple techniques frequently outperform more complex ones [3]. In order to evaluate the usefulness of reviews of a product, we want to build upon the available resources.

Several researchers have been working on sentiment classification using online product reviews with the goal to determine whether the consumers find the product useful. The machine learning for classification can be performed in two ways: supervised learning [6] to predict the semantics of adjectives, unsupervised learning [2] to classify reviews as helpful or no helpful by analyzing the semantic orientation of reviews. We use supervised machine learning here because there is a classification attribute in the data. The quality and helpfulness of the reviews is an active area of research.

For supervised learning, [7] developed a method using support vector machine (SVM) to automate the review helpfulness evaluation, [8] used entropy-based approach to explore the online review helpfulness and [5] used bilinear approach for classification of Amazon data.

The related area of research is the *effect* of online product reviews on the *product sales*. The consumers consider not only the reviews but also the reputation of the reviewer [9]. Finding a best data-mining tool is itself a data mining problem with no resolution.

C. Helpfulness Metrics for Supervised Learning

First, we briefly discuss the Entropy measure [8] and bilinear similarity [5] measures for relevance ranking of reviews for products. We will be using these measures in a hybrid fashion later in this paper. Our approach is unique as it uses the heuristics from singular value decomposition (SVD) and best of bilinear and entropy. The resulting metric may be used to determine relevance, classification and ranking simultaneously.

An online review consists of words, including opinion words, product features, product parts, and other words. The importance of each word to relevance and helpfulness of a review is calculated from the training dataset that has the consumer voting data information.

1) Helpfulness Model

Let P be the set of products, $p \in P$ be a product; R be the set of reviews and $r \in R$ be a review. Let r_h be the number of customers who vote for the review r as helpful. Let $r_{\bar{h}}$ be the number of customers who vote against the review r i.e. the review r is "not helpful". The review's helpfulness measure is defined by

$$H = \frac{r_h}{r_h + r_{\bar{h}}}.$$

Let $\tau = 0.66$, say, be the threshold to indicate that the *review is helpful* if $H > \tau$, i.e. two thirds of the consumers liked it. This is a simplistic view of the metric for classifying helpfulness of online reviews. Two of the metrics, we use for our algorithm are information Entropy [8] and Bilinear similarity [5], both help in the overall ranking of the products.

2) Entropy Model

First, we derive formulation for general *n* categories and single item classification. Then we adapt it to a review consisting of several words for two categories: "helpful" and "not helpful". Let $C = \{c_1, c_2, c_3, \ldots, c_n\}$ be the set of categories for products in the review space. Then information Entropy [10] needed to classify a review *r* is

$$\mathrm{H(C)} = -\sum_{i=1}^{n} \mathrm{P}_r(c_i) \log \mathrm{P}_r(c_i)$$

H stands in honor of Boltzmann's H-Theorem. Non-uniform entropy is normalized by the maximum entropy -$\log(\frac{1}{n}) = \log(n)$, the normalized entropy is

$$\mathrm{H(C)}= - \sum_{i=1}^{n} \frac{\mathrm{P}_r(c_i) \log \mathrm{P}_r(c_i)}{\log n}$$

The amount of information contributed by a term *t* (or word relevant to a document) is

$$\mathrm{H(C|t)} = -\sum_{i=1}^{n} \frac{\mathrm{P}_r(c_i|t) \log \mathrm{P}_r(c_i|t)}{\log n}$$

Information gain (entropy reduction by knowledge of *t*) is then

$$\mathrm{G(t) = H(C) - H(C|t)}$$

Higher the gain, higher the ability to classify, lower the uncertainty in the ability to classify.

This is the standard metric used to classify unknown (new items, not in the training set) related items. To get a better estimate for prediction, we include the occurrence and non-occurrence of t

$$\mathrm{G(t) = H(C)} - \mathrm{P}_r\mathrm{(t)\, H(C|t)} - \mathrm{P_r}(\bar{t})\, \mathrm{H}(\mathrm{C}|\bar{t})$$

contributes to the prediction ability. Larger the value of G(t), better predictor by knowledge of *t*.

Note. In this case, "helpful" and "not helpful" are two categories. Let c_1 be a category of "helpful" and c_2 be "not helpful". In order to differentiate prediction ability for two categories, G(t) upgraded to

$$\mathrm{Gain(t)} = \begin{Bmatrix} G(t) \;if\; P(c_1|t) > P(c_2|t) \\ -\,G(t) \quad otherwise \end{Bmatrix}$$

3) Prediction Computation

If there are *n* words in the review, then score is for a review is sum of the gain of each term

$$Score(r_i) = \sum_{k=1}^{N} Gain(t_k) * f(r_i, t_k)$$

where Gain(t_k) is k-th word's gain value, *N* is the number of words in review r_i and f(r_i, t_k) ,the Heaviside function,

$$f(r_i, t_k) = \begin{cases} 1 \;\; if\; term\; t_k occurs\; in\; r_i \\ \quad 0 \;\; otherwise \end{cases}$$

The helpfulness of product reviews is modeled by Gain and their scores, higher the rank of review, more the helpful information.

D. Bilinear Model

Similarity learning is another area of supervised machine learning in artificial intelligence. Several attempts have been made to determine the relevance of a query *q* to a document *d*, e.g., similarity of two documents to detect plagiarism. Similarity property must satisfy sim(d,d)=1, sim(d,d_1)=sim(d_1,d), where *d* and d_1 are documents. This model originates from simple cosine similarity or Euclidean dot product

$$cos(q,d) = \frac{q \bullet d}{|q||d|}$$

Closer cos(q,d) is to 1 more similar they are, closer cos(q,d) to zero, more dissimilar they are. Cosine similarity has one problem, that common irrelevant words can dominate the ranking. It is resolved by using the weighted cosine measure [11]

$$cos_w(q,d) = \frac{(q \odot d) \bullet w}{|q||d|}$$

where *w* is the weight vector and ⊙ is Hadamard product operator. This definition violates the similarity property sim(d,d) = 1. Since the weight vector has positive components, we take the point wise positive square root of components, and get $w = \sqrt{w} \odot \sqrt{w}$. Now the accurate weighted cosine is accurately defined as

$$cos_w(q,d) = cos\,(q_w,d_w) = \frac{q_w \bullet d_w}{|q_w||d_w|}$$

where $q_w = q \odot \sqrt{w}$ and $d_w = d \odot \sqrt{w}$

It could not detect words that appear many times in a selected document, but which are rare among other documents. A new metric BM25 [5] was introduced to resolve this.

$$bm25(q,d) = \sum_{i=1}^{n} \frac{IDF(q_i) \bullet f(q_i,d) \bullet (k_1 + 1)}{f(q_i,d) + k_1 \bullet (1 - b + b \bullet \frac{|d|}{avgdl})}$$

where parameters q_i is a word in the query *q*, *d* is a document, *f* is term frequency of q_i, IDF inverse document frequency, '*avgdl*' is the average document length, *b* and k_1 are weight parameters; see [12, 13, 14] for further detail.

The problem still persisted. These could not resolve query and document to be of different lengths and another problem was with the similar words. Bilinear measure uses the concepts in these measures and is able to overcome their deficiencies. The bilinear model can compute the similarity of objects of different dimensions.

TF-IDF like measures resolve some of the similarity issues: (1) irrelevant words issue of cosine measure via weighted cosine and (2) words frequent/important in one and rare in other documents issue via bm25 measure. But still there *are other issues (1) different length documents, a query, and a document are mostly of different lengths, (2) concerning synonyms, i.e., different words being used to refer to the same concept, e.g*. bright, sharp. *(3) the same word may have different meanings, e.g*. it is my "fan" versus he is my "fan". Such problems are not resolved by the above-mentioned similarity measures. Since queries/reviews are done by different people, it is important to differentiate whether the questions and reviews are tangentially related and may be drawn from very different vocabularies [15], [8]. Thus, one needs to learn whether a word (say burning) is used for "fire" or "heat" in a given scenario. Bilinear models [16, 17] can help to address this issue by learning complex mappings between words in one corpus and words in another (or more generally between arbitrary feature spaces). A common approach for learning similarity is to learn correlation matrix M so that the similarity metric for q and d of possibly different lengths becomes q^TMd. The correlation matrix is also suitable for streaming data. The matrix M can be quickly updated by a simple matrix addition. Thus, the compatibility between a query and a document is given by

$$q^T M d = \sum_{i=1,n;j=1,n} M_{ij}\, q_i\, d_j$$

where M_{ij} encodes the relationship between a term q_i in the query and a term d_j in document d. It incorporates the relation between different words while Cosine handles the relation between the identical lengths query and document. This is a flexible metric because the dimensions of q and d can be different. In practice, M is very high-dimensional (square of the size of the vocabulary) sparse matrix. Base on the size of vocabulary as compared to use in a query or document, we assume that it is a low rank matrix. Any matrix M can be decomposed as $M = USV^T$ where S is a diagonal matrix with a few non-zero diagonal entries (eigenvalues of M), U and V are orthogonal matrices. They can be reduced depending on the rank of M. This is known as singular value decomposition (SVD) of M. For a related article see [18]. Now the similarity measure becomes

$$q^TMd = q^TUSV^Td = q^TUS(d^TV)^T$$
$$= (U^Tq)^{T}S\,(V^Td)$$

Intuitively, U^Tq and V^Td are projections of q and d on smaller spaces than the query and the document dimensions.

These projections provide useful information about the dominant words. An interesting consequence of projections is that synonyms in q and d are projected to nearby words resulting higher inner product. Also, it optimizes the calculation over q^TMd by replacing M by new M using reduced dimensions of U, S, and V.

Note. In [5] the matrix M is decomposed into A and B such that $M=AB^T$. It does not ensure that A and B are orthogonal matrices. This decomposition is *not unique* because there are several QR type decompositions, for example, we can have A=U, B=SV, or A=US, B=V or A=U√S and B=√SV etc.

It is not beneficial to apply data mining algorithm before preprocessing the data. Bilinear similarity implicitly cleans up the data in resolving several issues mentioned earlier. Now that data has been reduced, we can apply entropy to the dataset with reduced dimensions of q and d. his yields a hybrid algorithm. In the previous section, we used Entropy on raw data. Overall we have hybrid data QA and text, hybrid algorithm using entropy, and bilinear similarity.

In the experiments section, we use entropy and bilinear similarity of q and d. Next step will be to compare the outcomes by using compressed U^Tq and V^Td instead of q and d. The hybrid algorithm will give even better decision power than the application of standalone algorithms.

III. Contributions of this Paper

A. *Hybrid Study*

This study began with a wider goal to develop novel algorithms, not just selectively use available algorithms, for better online classification and ranking of products and services. We ended up defining a hybrid approach to take advantage of the best features related to review data and available classifiers.

Amazon review data for various products is both solicited and unsolicited. Unsolicited data ends up at the blogs and solicited data is in the form of queries, query by form (QBF). Unsolicited data is naturally unstructured. The QBF data is somewhat structured and is used in our research study to evaluate the effectiveness of the proposed algorithm. The raw data that we have used is available freely for use at [20] and is discussed in detail in sub-section C.

For our study reported in this paper, we are using two types of records. First category of record type is in the form of plain text and contains nine attributes, they are: nominal (titleOfReview, reviewerID, reviewerName, productID, reviewText in the form of text comments), ordinal (productRating 0 to 5, ratingFrequency [helpful ratings, total ratings], interval (unixReviewTime, reviewDate). Here five of the attributes are irrelevant to the product review. The second category of data type is in the form of questions and answers. Each record contains seven attributes: productID, questionType (y/n or open-ended), answerType, questionText, answerText, unixTime, date. Again three of the attributes are not relevant to the analysis.

We start with two categories of opinion based data, text reviews and QA, and calculate similarity score for data of each type uniformly. We then take the results and use the weighted contribution of each similar to the approach taken in [19]. We propose the weighting of each may be calculated from the datasets as follows. If the r_T is the number of reviews in text form and r_Q is the number of reviews in question/answer form, their contribution is weighted with weight vector $\frac{(r_T, r_Q)}{r_T + r_Q}$.

The value of such a model is relevance, classification, and ranking parameters are learned simultaneously. This allows individual learners to focus on classifying only those instances that are 'relevant,' without considering the irrelevant instances.

B. Ranking Algorithm

We take advantage of many of the techniques surveyed in section II into a single adaptive hybrid algorithm to rank products. We use the following heterogeneous information related to a product as inputs to our approach:

1. Text of the review
2. Question/Answer data
3. Rating of the product
4. Rating of the reviews

The QA data is further divided into the following categories:

1. Normalized ratio of asked questions to answered questions, where 0 would indicate no answers and 1 would indicate all questions are answered.

2. Normalized relevance rating of the answer, where 0 would indicate the answer is not relevant and 1 would indicate the answer is completely relevant.

Overall the product is ranked on weighted QA rank, weighted text based review rank, and normalized rating using the adaptive hybrid algorithm defined in Figure 1. The weighting function of reviews vs QA ranks is calculated from the number of related dataset records. For example, if the r_T is the number of records in text form reviews and r_Q is the number of records in question/answer form, then the weight vector (γ,δ) is calculated as:

$$(\gamma, \delta) = \left(\frac{r_T}{r_T + r_Q}, \frac{r_Q}{r_T + r_Q}\right)$$

Algorithm: A product '*i*''s final rank is given by:

$$productRank_i = \alpha\,(productRank_i) + \beta\left(\gamma\left(\frac{{productRating_{,i}}^{review_{rank,i}}}{2}\right) + \delta * QA_{rank,i}\right)$$

where,

- α and β are regression coefficients, α+ β=1
- γ and δ are review data size weight factors, γ + δ=1
- *productRating$_{,i}$*: The normalized over [0,2] average of all product '*i*' ratings received from reviewers. The initial value is 1.
- $review_{rank,i} = \begin{cases} 0.5\,entropyRating_i + 0.5\,\frac{helpfulRating_{count,i}}{totalRating_{count,i}}, & if\ totalRating_{count,i} > 0 \\ entropyRating_i, & otherwise \end{cases}$
- $QA_{rank,i} = 0.5\,answered_{ratio,i} + 0.5\,bilinearSimilarity_{score,i}$
- *answered$_{ratio,i}$*: the ratio of answered questions to asked questions
- *bilineaerSimilarity$_{score,i}$*: normalized relevance rating of the answers using bilinear similarity

Figure 1. Product ranking algorithm

The algorithm has the following desirable properties:

1. The coefficients α and β can be fine-tuned to influence the adaptability of the rank to new information presented (in terms of reviews and Q&A). If the rank is α heavy, i.e., α > 0.5, the rank is more stable whereas if it is β heavy, i.e., β > 0.5, then the algorithm is more reactive to new information

2. *review*$_{rank}$ is calculated by blending both user helpfulness rating and also the entropy-based machine learning rating, see sections II.C.1 and II.C.2. This removes any implicit or explicit bias in a particular product's reviews and uses the information gathered as the collective wisdom to make the rating more neutral.

3. Helpful reviews fused with higher *productRating* (greater than 1) score improves the overall product rank. Whereas, helpful reviews combined with the lower *productRating* (less than 1) score, pulls it down further. See Fig. 2 for illustration.

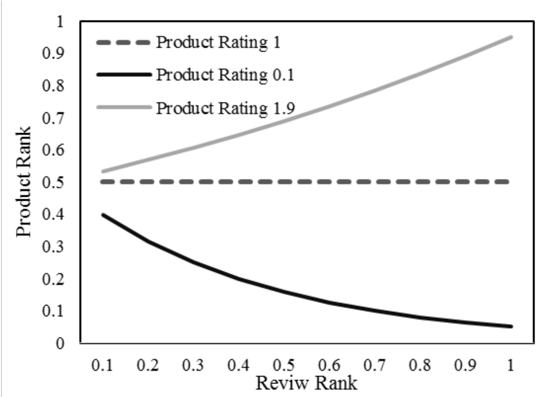

Figure 2. Impact of Review Rank on overall Product Rank for given Product Rating

C. Dataset

We used Amazon product review data for our analysis, particularly, the star ranking, reviews, and related Q&A. The raw data is available freely for use at [20] and contains product reviews and metadata from Amazon, including 142.8 million reviews spanning May 1996 - July 2014. This dataset includes reviews (ratings, text, helpfulness votes), product metadata (descriptions, category information, price, brand, and image features), and links (also viewed/also bought graphs).and has been used in related research, such as [21].

As mentioned above, we are particularly interested in star ranking, review text, and Q&A text. For our analysis, we utilized the k-core subsets of Musical Instruments (500,000 records), Electronics (800,000 records), and Health and Personal Care (900,000 records) categories. Sample records are presented below:

```
{
    "reviewerID": "A3LA5EHF2WWNFK",
    "asin": "B000000545",
    "reviewerName": "Philip Y.",
    "helpful": [1, 1],
    "reviewText": "One of the best albums I've ever heard. B-Legit's great delivery and lyrics along with many fly guest featured over this tight production - it just makes a perfect album with the bay area flava. C-Bo, Kurupt, E40, Celly Cell and many more join the Savage in one of the west coast best and most underrated albums. Get that fly sh*t!!!",
    "overall": 5.0,
    "summary": "OFF THE HEEZY!!!!",
    "unixReviewTime": 906681600,
    "reviewTime": "09 25, 1998"
}
```

where

- *reviewerID* - ID of the reviewer
- *asin* - ID of the product
- *reviewerName* - name of the reviewer
- *helpful* - helpfulness rating of the review as [helpful ratings, total ratings]
- *reviewText* - text of the review
- *overall* - rating of the product out of [0, 5]
- *summary* – summary/title of the review
- *unixReviewTime* - time of the review (unix time)
- *reviewTime* - time of the review (raw)

Review data record in the form of QA:

```
{
    'questionType': 'yes/no',
    'asin': '9792372326',
    'answerTime': 'Jan 21, 2015',
    'unixTime': 1421827200,
    'question': 'Will the K10 be damaged if its XLR output is connected to a mixing board and somebody accidentally pushes the +48V phantom power button on the board?',
    'answerType': 'N',
    'answer': 'no'
}
```

where

- *questionType* - type of question. Could be 'yes/no' or 'open-ended'
- *asin* - ID of the product
- *answerType* - type of answer. Could be 'Y', 'N', or '?' (if the polarity of the answer could not be predicted). Only present for yes/no questions.
- *answerTime* - raw answer timestamp
- *unixTime* - answer timestamp converted to unix time
- *question* - question text
- *answer* - answer text

IV. Evaluation and Discussion

We start by calculating the entropy measure of product reviews. Using the entropy based classification model as explained in section II B, we calculate the $Score_i$ for the reviews per product '*i*'. We have used k-fold cross validation, k=10, to optimize the model parameters to make the model fit the training data as best as possible.

We have used Amazon's Elastic Map Reduce (EMR) service, using Java, to sift through both the review and the QA files, see description of dataset in the next section. Below is the brief description of the two phases used, also depicted in Figure 3.

Phase 1: The mappers map the review and QA data to a product ASIN, whereas the Reducers output the data per product in the form of a string array.

Phase 2: The mappers calculate the stats, entropy and bilinearity, per product, whereas the Reducers take all the relevant stats per product and using algorithm given in Figure 1, output the final rank.

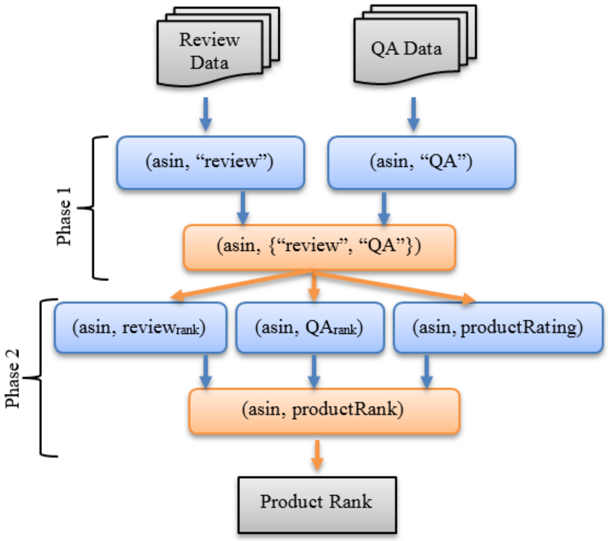

Figure 3. MapReduce Phases for calculating the Product Rank. Blue boxes represent Mappers and orange boxes represent Reducers.

The entropyRating$_i$ measure, as used in the algorithm presented above, is calculated as:

$$entropyRating_i = \frac{Score_i}{Score_{max}}$$

$$Score_{max} = \max \forall Score_i$$

Hence entropyRating will be 1 for the product with maximum score, and will be distributed in the unit interval [0, 1] for all other products.

Table I and II present the precision, recall and F-measures for the entropy-based classifications of reviews as 'helpful' and 'unhelpful'

TABLE I. Precision, Recall, and F-Measure for reviews classified as 'Helpful'

	Precision	Recall	F-Measure
Musical Instruments	0.78	0.76	0.76987
Health and Personal Care	0.76	0.77	0.764967
Electronics	0.72	0.79	0.753377

TABLE II. Precision, Recall and F-Measure for reviews classified as 'Not Helpful'

	Precision	Recall	F-Measure
Musical Instruments	0.79	0.75	0.769480519
Health and Personal Care	0.75	0.78	0.764705882
Electronics	0.74	0.77	0.754701987

In Table III we present some product ranking examples. The product rank is calculated by both the old method, which is simply the average of all the product ratings received by the users and the new method, which is based on our algorithm. We would like to elaborate on the interesting contrasts in the two methods. If you look at examples 1-3, the old ranking is the same for all three cases, as the product rating is the same. However, the new ranking is quite different for the three cases. It is all the more interesting is to see how the new product ranking accurately reflects the review ranks and QA ranks, with higher values reflected in higher ranks and vice versa.

TABLE III. Product Rank as Calculated by Old and New Algorithm

	Product Rating	Review Rank	QA Rank	Product rank (new)	Product rank (old)
1	1.9	0.5	0.5	0.5946012	0.95
2	1.9	0.8	0.8	0.8177754	0.95
3	1.9	0.3	0.3	0.4530861	0.95
4	1.1	0.8	0.8	0.6698076	0.55
5	1.1	0.8	0.4	0.4698076	0.55
6	0.5	0.8	0.8	0.5435873	0.25
7	0.5	0.3	0.8	0.6030631	0.25

Examples 4 and 5 show the effect of different QA rank on the overall product ranking where the product rating and review rank is the same. Where the old product rank is solely dependent on product rating, the new rank reflects the QA ranking proportionally. Similarly, examples 6 and 7 reflect the effects of different review rank when product rating and QA rank are the same. It is, however, important to note that here the higher review rank resulted in lower overall product rank. This is one of the key characteristics of our algorithm, i.e., helpful reviews on a product increase the authenticity of the lower rating. Hence, two products with same low product rating but one having more 'helpful' reviews is an indication of a strong negative response towards the product. Hence in the overall ranking, this product falls further below the rank list.

V. Conclusion and Future Work

Several commercial websites for products and services provide platforms for the consumers to share their opinions. In this paper, we developed a hybrid adaptive algorithm that uses heterogeneous product review data (star ranking, text based reviews, question/answer data) to simultaneously analyze and evaluate the reviews in order to rank similar products. We show that adapting the ranking algorithm by simultaneously using Entropy and Bilinear Similarity measures yields more accurate evaluation than using them in isolation and on restricted data types. For experimentation and evaluation we used three categories of Amazon.com review data: k-core subsets of Musical Instruments, Health and Personal Care and Electronics to simultaneously analyze and evaluate the reviews for product ranking. The results of experiments are displayed using Gold standard metrics with the help of tables. We showed that our hybrid approach is effective and yields results which make not only intuitive sense but are mathematically sound.

In this work we have used weighted scores of Entropy and Bilinearity breadth first techniques. In future, we would like to explore the potential for depth first hybrid algorithm. Also, we would like to further expand this work to enable product related predictions for both consumers and sellers.

Acknowledgment

Use of Amazon product review data, available freely for research [20]

References

[1] G. Lackermair, D. Kailer and K. Kanmaz, "Importance of Online Product Reviews from a Consumer's Perspective". Advances in Economics and Business Vol. 1(1) 2013. Pages 1-5

[2] P. D. Turney, "Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews". Proceeding ACL 2002 Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. Pages 417-424

[3] Aria Haghighi, How to approach machine learning as a non-technical person. Crunch Network, Apr 2, 2016. http://techcrunch.com/2016/04/02/how-to-approach-machine-learning-as-a-non-technical-person/

[4] D.H. Park, J. Lee, and I. Han, "The Effect of On-Line Consumer Reviews on Consumer Purchasing Intention: The Moderating Role of Involvement". International Journal of Electronic Commerce Vol. 11, No. 4 (Summer, 2007). Pages 125-148

[5] L. Zhang, D. Agarwal and B. C. Chen, "Generalizing matrix factorization through flexible regression priors". In Proceedings of the fifth ACM conference on Recommender systems 2011, Pages 13-20

[6] H. Yu and V. Hatzivassiloglou. "Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences". Proceeding of EMNLP '03 Proceedings of the 2003 conference on Empirical methods in natural language processing. Pages 129-136

[7] D. H. Park, H. D. Kim, C. Zhai and L. Guo, "Retrieval of Relevant Opinion Sentences for New Products". SIGIR 2015 Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval. Pages 393-402

[8] R. Zhang and T. Thomas, "An Entropy-Based Model for Discovering the Usefulness of Online Product Reviews". In Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web. Pages 759-762

[9] M. Hu and B. Liu, "Mining and summarizing customer reviews". KDD 2004 Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. Pages 168-177

[10] C. E. Shannon and W. Weaver, "The Mathematical Theory of Communication". University of Illionoins Press

[11] J. McAuley and A. Yang, "Addressing Complex and Subjective Product-Related Queries with Customer Reviews". WWW 2016 Proceedings of the 25th International Conference on World Wide Web. Pages 625-635

[12] K. S. Jones, S. Walker and S. Robertson, "A probabilistic model of information retrieval: development and comparative experiments". Journal of Information Processing and Management: an International Journal, Volume 36 Issue 6, 2000. Pages 779-808

[13] J. Jeon, W. B. Croft and J. H. Lee. "Finding similar questions in large question and answer archives". CIKM 2005 Proceedings of the 14th ACM international conference on Information and knowledge management. Pages 84-90

[14] C. D. Manning, P. Raghavan, and H. Schültze, An Introduction to Information Retrieval. Cambridge University Press, 2009.

[15] A. Berger, R. Caruana, D. Cohn, D. Freitag, and V. Mittal, "Bridging the lexical chasm: statistical approaches to answer finding". SIGIR 2000 Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval. Pages 192-199

[16] W. Chu and S.-T. Park, "Personalized recommendation on dynamic content using predictive bilinear models". WWW 2009 Proceedings of the 18th international conference on World wide web. Pages 691-700

[17] W. T. Freeman and J. Tenenbaum, "Learning bilinear models for two-factor problems in vision". IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1997. Pages 554-560

[18] C. L. Sabharwal and B. Anjum, "Principal Component Analysis as an Integral Part of Data Mining in Health Informatics". Proceedings of 31st International Conference on Computers and Their Applications, CATA 2016. Pages 251-256.

[19] R. A. Jacobs, M. I. Jordan, S. J. Nowlan and G. E. Hinton, "Adaptive mixtures of local experts". Journal of Neural Computation. Volume 3 Issue 1, Spring 1991. Pages 79-87

[20] Amazon Review Data: jmcauley.ucsd.edu/data/amazon

[21] J. McAuley, R. Pandey and J. Leskovec, "Inferring Networks of Substitutable and Complementary Products". KDD 2015 Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Pages 785-794