See the Invisible: Mining Market Knowledge from Online Reviews using ‘TEM’

Author:  
Ran Wang

Supervisor:  
Prof. dr. H.P. Boswijk  
dr. A.J. van Es

August 26, 2015
See the Invisible

Mining Market Knowledge from Online Reviews using ‘TEM’

Abstract
With the proliferation of the Internet, interest as to extract market related information from various online information source, especially online reviews, has grown substantially. While numerous methods have been proposed in order to extract important product features, there is no existing model that offers the functionality of mining the latent evaluation, which we define as the structural relationship in terms of how numerous features collectively determine the overall sentiment level. This knowledge is desirable, however, as it both facilitates a deeper understanding of consumer decision-making processes as well as a straight-forward presentation to the managerial board. To this end, we proposed TEM (Transform, Extract and Mine).

The general flow of ‘TEM’ can be summarized as follows. In the ‘Transform’ Step, we extracted nouns as candidate feature, then match the polarity of adjacent adjectives as its sentiment orientation. In the ‘Extract’ step, we adapted the model from Nonlinear Component Analysis while adding a penalty to model complexity, in order to extract latent evaluations of individual review. This information is then applied to ‘Mine’ market related knowledge, such as consumer preferences, market trajectory and consumer heterogeneity etc.

In general, our new approach is computationally efficient, satisfactory in accuracy and insightful on presentation. In this fashion, this study contributes both as a new perspective on online information mining and a building block for model-building in the “big data” age.
## Contents

1. Introduction 3  
2. Literature Review 6  
2.1. Internet and Marketing 6  
2.2. Reviews of Existing Methods 7  
2.3. Research Gap 8  
3. Methodology 10  
3.1. Transform 10  
3.2. Extraction 11  
3.3. Mining 11  
4. Details of the ‘Extraction’ Step 13  
4.1. Data Description and Notation 13  
4.2. Model Description and Identification Issues 13  
4.3. General Description of the Algorithm 15  
4.4. Details of the Algorithm 16  
4.5. General Comments 22  
5. Monte Carlo Study 24  
5.1. Introduction 24  
5.2. Data Generation Processes 24  
5.3. Monte Carlo Procedure 25  
5.4. Results 25  
6. Empirical Data Analysis 29  
6.1. Data Overview 29  
6.2. Macro Marketing Analysis 30  
6.3. Micro Marketing Analysis 33  
7. Conclusion: Contributions and Future Studies 39  
7.1. Contributions 39  
7.2. Future Studies 39  
8. Appendix 1: Likelihood Based Approach 41  
8.1. Introduction 41  
8.2. Data 41  
8.3. Model and Likelihood 41  
8.4. Estimation Routine 42  
8.5. Details of the Routine 42  
8.6. Problems of the Routine 50  
9. Appendix 2: Parallel Computing with C++11 Thread Library 51  
9.1. Introduction 51  
9.2. Introduction to Parallel Programming 51  
9.3. MPI and its Limitation 52  
9.4. C++11 Thread Library 53  
9.5. Performance Test 54  
References 56
1. Introduction

It is widely agreed that consumer preferences are what lies at the heart of marketing theory and practice. Before the emergence of the Internet, consumer preferences had mostly been inferred via experiment-based methods such as Focus Group Talk [15] and Pre-Purchase Test [28], or survey-based methods such as Conjoint Analysis [24] and Discrete Choice Analysis [37]. It appears that two pitfalls are common to all of these methods. Firstly, the implementation costs of these methods tend to escalate with the scale of targeted consumers, and consequently, marketeers are often forced to limit the scope of their surveys to niche markets. More importantly, these tests typically place consumers in an environment rudimentarily different from a real-life scenario, and this artificiality is known to significantly alter the behaviour of the subjects (see [49] for review). While in theory such drawbacks can be avoided by analyzing empirical purchase data, the inaccessibility to pre- or post-purchase behavioural details has limited the power of this alternative method. Fortunately, the recent proliferation of the Internet has resulted in a plethora of user-generated content in the form of blogs, fora and review websites. As a result, consumer word-of-mouth has moved from small and private conversations to large-scale online and public networks, where consumers freely express and exchange their experience, evaluation of and preference for certain products [33]. These alternative data sources can be exploited considerably in order to provide novel insight on consumer decision-making process.

One of the frequently studied forms of online consumer-generated content is online consumer reviews, which often take on two forms: (1) review score — a number that usually ranges from 0 to 10 and which reflects the overall preferences of consumers toward certain products; or (2) review text — the unstructured texts posted by the reviewers and which contain a more detailed elaboration on the reasons why a particular score was given. These forms of data have been shown to contain substantial amounts of relevant information with regard to sale trajectory, market structure, and most importantly, consumer preferences. More specifically, sales prediction can be achieved by utilizing review scores, see e.g. [12], [17], [18], [19] and [35], as could also be the forecast of sentiment evolution (see e.g. [13], [22], [39] and [40]). Consumer preferences in regard to different product features, as well as their potential implications on product performance, can be also thoroughly investigated by studying these reviews (see e.g. [20], [57], [31], [44], [27] and [61]).

Although these studies have no doubt enriched the toolbox of the marketeers, it appears that a crucial gap still remains to be filled. In particular, although various methods have been proposed in order to identify various important product features ([5], [16], [27], [62] etc.), the structural relationship in terms of how numerous features collectively determine the overall sentiment level and which is referred to as latent consumer preferences is still unknown. Nevertheless, for various reasons, this knowledge is arguably
desirable. For example, consider the following review obtained from Metacritic.com concerning the game Assassin’s Creed IV: Black Flag\(^1\) (Score 10/10)

The most fun I’ve had with a game in a long time. The story isn’t the most strong in the series, but there’s so much fun content here that the story is almost the side show. I rarely write my impressions on a game, but I just absolutely had to share how impressed I am with this one.

Black Flag’s naval combat is stellar. You’ve got so many options on how to raid, board and decimate enemy ships that it doesn’t get boring. Great stealth sections, lots of choice on how to tackle missions, optional outer animus sections, great voice acting and even better music, this is the definitive AC experience; and when playing on ps4 it’s even more easy to appreciate the amount of work that has gone into the graphics\(^2\).

While many features are on display in this example, it can be argued that these features center around two main aspects, namely “story” and “gameplay”. Furthermore, it is also logical to assume that the “gameplay” bears more importance than the “story”, for otherwise the game would not have been rated 10 out of 10. In this case, the model that consumers first develop their evaluations of the product based upon several latent attributes, which they then elaborate in review texts, is more appropriate than models that simply employ a uni-dimensional collection of significant features. Unfortunately, the former model appears to be absent in the literature, possibly due to the fact that as the number of latent attributes increases, so does the number of parameters. This, in combination with the innate complexity of natural language, is bound to result in an overwhelmingly high-dimensional model. In such a case, traditional estimation procedures have been proven to be unstable and inefficient in a number of difficult settings (\cite{6, 52, 54}).

Fortunately, the progress in high-dimensional data analysis has allowed us to tailor an estimation paradigm to suit this purpose. Specifically, we propose a 3-step approach called TEM (Transform, Extract and Mine) that would efficiently help extract market knowledge from online review data. The details are as follows. In the first step, we adapt methods from \cite{57} in order to transform unstructured texts into an analyzable data array. Specifically, we match extracted features with a pre-determined lexicon by \cite{45}, which allows us to determine the sentiment orientation for each feature. In the second step, we model extracted features in the latent feature space using a model adapted from Categorical Component Analysis \cite{34} while incorporating both the review scores and the texts. Different from \cite{54}, a penalty has been introduced to shrink the number of extracted features into a desirable range. Shrinkage estimators of such kind have been proven to be efficient in various high-dimensional settings (\cite{9, 52}). Finally, the extracted latent evaluations are used to extract a rich collection of marketing information.

\(^2\)http://www.metacritic.com/game/playstation-4/assassins-creed-iv-black-flag/user-reviews
e.g. consumer preferences, market structure, sentiment evolution and latent consumer clusters etc.

When compared to existing methods, the presented approach displays the following advantages. Firstly, our approach applies to a broad range of review website design. More specifically, our approach does not require reviews to be pre-classified into the “pros” and the “cons”, as is the case in epinions.com. Secondly, it is possible to uncover latent evaluations of individual reviews, which would successfully facilitate both individual targeting and consumer latent class analysis. Thirdly, our model takes into account the sparse nature of the data. This would likely not only result in more accurate estimation, but also significantly improve the interpretability of our results. Fourthly, the results of this approach can be presented in a fairly straight-forward yet insightful fashion, accessible to both marketing scholars and practitioners. Lastly, our approach is computationally efficient, which allows us to apply this approach to possibly much larger data sets.

This study contributes to the existing literature as follows. From a marketing perspective, we are the first to extract the latent structure of product attributes based on the review data. In this manner, we provide novel insights into how consumers trade off between different features, and also pragmatic guidelines for product development. From a text mining perspective, we provide an alternative model which simultaneously selects important features and portrays their relationships, thus providing different perspectives to existing text mining techniques. From a methodological perspective, we advocate “big data” techniques, which are highly valued in the transformation of marketing science in the information age. In effect, our approach takes into account both the heavy computational burden as well as the high dimensionality — the two obstacles commonly present in analyzing online data.

The rest of our paper is organized as follows. The paper starts with the implications of the Internet on the marketing practice, followed by a review of existing methods for marketing knowledge extraction and demonstration of the needs for new models. We then present the details of our approach, followed by a Monte Carlo study that demonstrates the validity of our method. After this we will apply our model to game review data collected from MetaCritic.com, and demonstrate the possible application of our methodology. Finally, we are going to conclude the implications of this approach for future works.

3http://www.epinions.com/
2. Literature Review

2.1. Internet and Marketing. One of the most important game changers in the commercial world is the Internet. Since its inception, the Internet has amassed more than three billion users,\(^4\) 80% of whom have shopped at least once\(^4\) through different online channels. Contributing to this enormous scale of online trade is the incredible transform volume. For example, more than 400 transactions are issued per second on amazon.com alone, the number of which continues to grow daily.\(^6\) The significance of the Internet is not only in the number of business opportunities it creates, but also on how it reshapes the purchasing process. For example, instead of relying on a street-corner bookstore manager for product information, customers might first consult the twitter account of a book recommender\(^7\) for new hits, then place an order on Amazon\(^8\) and finally share their reviews on Google Books\(^9\). In general, these changes have facilitated both chances and challenges for marketeers.

One of such changes is the emergence of online review websites, e.g. epinions.com\(^10\), metacritic.com\(^11\) and buzzillions.com\(^12\) etc. In addition to these general review websites, many online sellers, e.g. ebay.com\(^13\), amazon.com\(^14\) and booking.com\(^15\) also allow buyers to post their reviews on the purchased products or service. These reviews usually contain the following information: (1) online review score, or rating — a number that represents the general evaluation of the product; (2) review texts — the unstructured texts where consumers freely express their evaluations, sentiments, and feedbacks of the product or service. These forms of information are commonly known as (online) word-of-mouth, and are similar to their offline counterparts in that they are closely related to consumer buying behaviours (\([12, 13, 17, 18, 19, 22, 35, 39\) and \([40])\).

Unlike offline word-of-mouth, however, online reviews are generally freely accessible, which creates an opportunity for marketeers to extract market information. More specifically, with the aid of online review data, consumer preference inference can be implemented such that it is free from potential problems associated with classical market analysis tool kits, especially in situations where word-of-mouth is absent. Indeed, while the importance of the word-of-mouth has long been recognized by both scholars and practitioners, this information is mostly confined to private context often inaccessible by

\(^4\)http://www.internetlivestats.com/internet-users/
\(^5\)http://www.cpcstrategy.com/blog/2013/08/ecommerce-infographic/
\(^7\)https://twitter.com/brecommend
\(^8\)https://www.amazon.com/gp/gw/ajax/s.html
\(^9\)https://books.google.nl/?hl=en
\(^10\)http://www.epinions.com/
\(^11\)http://www.metacritic.com
\(^12\)http://www.buzzillions.com/
\(^13\)http://www.ebay.com/
\(^14\)http://www.amazon.com/
\(^15\)http://www.booking.com/
marketeers. As a consequence, market analysis is usually based on alternative data sources that are considered to contain information similar to that of the word-of-mouth. Two candidates for such sources are marketing surveys and consumer interviews. While data collected through these channels has been proven to significantly improve the efficiency and accuracy of market decisions, there are some intrinsic drawbacks in these data sources, and we will elaborate on the deficiencies in the subsequent paragraphs.

One of the alternative data sources is marketing survey, where consumers are asked to complete pre-designed questionnaires, whose data is later analysed by the marketeers. Unfortunately, three pitfalls seem to pertain to this method. Firstly, handing out and collecting questionnaires can prove to be costly, especially in case of a high non-response rate. Consequently, analyses are usually confined to a small scale, thus creating potential bias in the results. Additionally, the success of the method hinges on the questionnaire designs. In cases where the prior knowledge is inaccurate, the collected data can also be less informative than desired originally, and resulting conclusions may misguide marketing decisions. Lastly, this method places consumers in a situation remote from a real-life scenario, which could create potential difference in consumer behaviours.

Another common data source is consumer interviews, which are usually carried out in a focus-group fashion. While this method suffers less from design bias as they can be carried out in a less guided fashion, the associated cost is often non-trivial. Moreover, the interviews can prove to be even more artificial, due to the presence of a company representative, which may too alter the behavior of consumers. While numerous attempts have been made to alleviate these drawbacks, the artificiality of these data collections makes it unlikely that these obstacles can be overcome completely as analysing techniques advance.

These intrinsic deficiencies usually do not present themselves in online review data for the following reasons. Firstly, online reviews are freely accessible, and hence data collection cost is almost negligible. This is especially convenient for large global enterprises since multinationally scaled surveys can prove expensive. Secondly, online review data is contributed by the consumers voluntarily; and hence, the potential bias caused by a survey or an experiment design can be reasonably avoided. In addition to preventing these deficiencies, online review data are also known to contain extensive knowledge complementary to collectible information from empirical purchasing data. For these reasons, both the scholars and the marketing practitioners have embarked on proposing methods to ‘mine’ market knowledge from such data.

2.2. Reviews of Existing Methods. From an academic standpoint, the existing methods for ‘mining’ marketing knowledge can be classified into two streams of literature, namely natural language processing (NLP) and marketing. While both aim at accurate extraction of meaningful information, the two schools maintain a different focus. Methods proposed in NLP literature mostly focus on a systematic approach in document summarization, while marketing literature tends to center on extraction of market knowledge using a variety of models.
The attempt at ‘mining’ online reviews is first made in the seminal work of [27], where the authors proposed PFE methods to extract product features. The steps of the PFE approach can be summarized as follows: (1) identify nouns or noun phrases as candidate features; (2) apply association algorithms to the above candidate features ([2] and [50]); (3) prune infrequent and redundant features, which results in the frequent feature set; (4) associate adjectives that are adjacent to words in the frequent feature set as opinions. By implementing these methods, [27] successfully ‘mined’ important product features based on review text data alone. Many alternatives have been proposed after [27], aiming at improving the selection accuracy by introducing alternative models and/or additional data sources. For example, [44] introduced KnowItAll which includes predetermined patterns to improve the extraction accuracy, [61] introduced training samples and applied supervised learning to determine the association and occurrence of different features. [57] introduces the General Inquirer to determine the sentiment orientation of adjectives. Although substantial difference exists in the aforementioned methods, they are in general aimed at a proper summarization of the review texts.

Different from NLP literature, marketing literature follows a more volatile path in online review mining, in that the proposed approaches are not necessarily based on summarization of review texts. For example, [32] transformed each presented word in a large 0-1 array based on its presence, and applied a clustering analysis and correspondence analysis. In this fashion, they are able to cluster consumers into different groups. Furthermore, [11] based their analysis on the concurrency of brand/product in certain reviews, and then build a model based on a Markov chain to reveal the market structure. [55] make a “tag cloud” plot based on the frequency of present features, which allows for a direct presentation of the importance of product features. [5] regressed the manually extracted features on sale data, thus selecting features that determines the overall sentiment level. Similar effects are achieved by [16], although with review scores replaced by sales data. These approaches complement those of NLP literature and provide a different possible angle of solving the problems.

2.3. Research Gap. Although all the aforementioned studies have successfully extracted important features, a related unanswered question is the structural relationship of these features in terms of their relationship to the overall sentiment level, usually represented by the user review scores ([16] and [17]). Generally speaking, there is a substantial amount of growing evidence showing that instead of evaluating on all the possible features of a certain produce or service to purchase, consumers only consider a few of them due to the limitations of their cognitive ability (for a review see [21]). On the other hand, a large number of extractable features are often present in reviews [16]. These facts imply that a relatively small number of latent features, based on which consumers determine the overall desirability of a certain product/service, might determine the “visible” features in their reviews. For this reason, it is desirable to propose a model that not only extracts significant visible features, but also uncovers the latent ones underlying the actual decision processes.
The need for such models also bears pragmatic considerations. In essence, the extraction of latent evaluations are used to improve marketing decisions, especially how to trade off between different possible improvements of product features. This creates a dilemma for model builders: on the one hand, if the underlying model selects a large number of features [16], simultaneously improvement on all of them in product design can prove costly; on the other hand, if modellers force a parsimonious model by selecting only a small subset of features (e.g. [27] and [57]), the model might not adequately reflect the complexity of the consumer decision processes. One solution of the aforementioned dilemma is to realize that evaluation of different extractable features is not isolated, but rather arguably determined by underlying latent evaluations. Uncovering these latent evaluations will not only lead naturally to more parsimonious and interpretable models, but also to the discovery of how consumers trade off between different attributes. Hence such models are highly desirable.

Unfortunately, as far as we are concerned, no such model or approach exists in the literature. For this reason, a new approach is proposed in our study, the details of which are presented in subsequent sections.
3. Methodology

In order to achieve our goals, we propose the following TEM (Transform, Extraction and Mine) approach, as presented in figure 1.

![Figure 1. The General Flow of TEM](image)

### 3.1. Transform

Before statistical analysis can be performed, review texts have to be transformed into numerical arrays. To this end, we modified [57], the content of which can be summarized five steps. For a better illustration, we consider the hypothetical example: “The games is good, although graphics is bad. Story? I don’t know” and summarize the steps as follows.

1. **Lemmatization.** In this step we “clean” the review texts for subsequent analysis, the tasks of which include spelling correction, noun singularization, upper-case to lower-case transformation among others (for a complete list, see [1]). In our example, the text will be transformed into “the game be good, although graphics be bad. story. I not know”.

2. **Tagging and Noun Extraction.** In this step, we “tag” the syntactic structure of each word present in each reviews, which allows us to identify nouns as potential features. It should be noted that this is accomplished via a Part-Of-Speech fashion, meaning that the syntactic function of the words depend on their grammatical position. Upon finishing this step, we then extract nouns as candidate features. In our example, the extracted features are “game”, “graphics”, “story”.

3. **Frequency Based Feature Pruning.** In order to achieve computational efficiency, we prune the noun features identified in the last step. In particular, if a feature is not present frequently enough in reviews, this feature is deleted from the candidate feature list.
(4) Concurrency Based Feature Pruning. Although a feature might appear frequently in the review, it does not necessarily appear frequently with an attitudinal adjective. To this end, we also prune feature(s) that do not enjoy adjacent with any adjectives.

(5) Sentiment Orientation Extraction. In this step, we match the adjectives in the review with a predetermined dictionary by [45], which classifies adjective into positive and negative groups. The content of this lexicon allows us to determine the sentiment orientation of each adjective present in the review. We then match these orientations to the adjacent feature(s) to determine whether a reviewer regards the specific feature(s) as positive or negative. In our example, the feature “game” is accompanied by the word “good”, the latter of which is deemed as positive, hence we deduce the evaluation of the feature “game” to be positive. On the other hand, “story” is not accompanied with any adjective in the same sentence, hence we consider the evaluation of this feature as non-present.

All the above steps are implemented using the Stanford NLP toolkit [36], a leading comprehensive Java library developed by Stanford University to handle the various needs of natural language analysis.

3.2. Extraction. As far as we are concerned, there is no model directly applicable for our needs. Were it possible to obtain a continuous evaluation on each extractable feature, one could directly apply Principal Component Analysis [51] or its sparse variation [64]. Unfortunately, the lexicon in this study only allows to determine an ordinal measure of sentiment orientation. To solve this problem, we assume that each level of the evaluation corresponds to certain continuous, fixed albeit unknown number, similar to [34]. Thus, instead of applying Principal Component Analysis or its variants directly to the original data set, we apply the method to a transformed data set. We deem our transformation optimal as it minimizes the prediction loss of transformed data set and the latent evaluations associated with it. In reality, analytical solutions are not available, and model estimation is achieved by iterating between transformation, latent score evaluation and score loading estimation/selection. Due to the additional computational burden required to deduce the transformation, we also propose a new method to implement the latent score evaluation and score loading estimation and selection. In addition, in order to maximize the efficiency of our approach, an appropriate number of non-zero coefficients in score loadings needs to be determined appropriately. This is achieved by a modified cross-validation. Finally, we choose the number of latent evaluations to be two for ease of presentation.

The details of this algorithm are presented in section 4. Due to the fact that our methods are new, we also conduct a Monte Carlo study, the results of which are presented in section 5. In general, it turns out that our method yields satisfactory computational efficiency and estimation accuracy.

3.3. Mining. After the extraction step has been implemented, a rich collection of market knowledge can be extracted.
As a starting point, we view the reviewers as homogeneous, which allows us to make inference on the macro market structure. In this perspective, our approach not only allows us to select significant features, but also to reveal the structural relationship within the selected feature set. This is accomplished by projecting a large collection of features onto a low-dimensional space, thus presenting straightforwardly how several features collectively determine a certain aspect of consumer preference. Furthermore, our approach also allows us to monitor the market trajectory — how consumer preferences evolve across time, as well as market structure — how one specific product is evaluated compared to others. This information will allow firm decision makers to further understand the nature of targeted consumers and existing competitors.

In addition, our approach can also significantly facilitate the exploration of micro-level market knowledge. Specifically, it is straightforward to discover consumers with lower satisfaction level on either overall or specific evaluations. Furthermore, one can also cluster consumers into different groups with different evaluations. Implementing these analyses will allow enterprises to apply individual based marketing strategies to improve customer retentions, and to tailor specifically designed marketing strategy to each niche market.
4. Details of the ‘Extraction’ Step

In this section, we present the details of our algorithm. We begin by describing the data and notation, then move on to the problem statement. After this, we present the outline of our algorithm and a short comparison with other existing methods that could be adjusted to suit our needs, albeit with difficulties. Finally, we conclude this section by elaborating on the computational detail.

4.1. Data Description and Notation. Here and throughout the rest of the text, we assume accessibility to training/validation data sets consisting of \( N_1/N_2 \) observations of reviews. With a slight abuse of notation, we use \( N \) to denote either \( N_1 \) or \( N_2 \), the exact content of which will be clear by the context. The review score for review \( i \) is denoted as \( y_i \), a quantity ranging from 0 to 10. Furthermore, we assume the existence of \( K \) latent features in either data set, and we denote the evaluation for feature \( k \) and review \( i \) as \( x_{ik} \) with the following coding:

\[
x_{ik} = \begin{cases} 
-1 & \text{If the evaluation is negative} \\
0 & \text{If the feature is not present} \\
1 & \text{If the evaluation is positive}
\end{cases}
\]

In order to facilitate our presentation, we also adopt the following notation. We use capital letters to denote matrices and lower-case letters to denote vectors or scalars, the dimensions of which should be clear from the context. For an arbitrary matrix \( V \), we denote \( v_{(i)} \) and \( v_{(k)} \) its \( i \)’th row \( k \)’th column respectively. We use \( \|v\|_2 \) to denote the \( L_2 \) norm of a vector — \( \|v\|_2 = \sqrt{v^T v} \), and \( \|V\|_F \) to denote the Frobenius-norm — \( \|V\|_F = \sqrt{\text{tr}(V^T V)} \).

4.2. Model Description and Identification Issues.

4.2.1. Model Description. We aim to minimize the following quantity, subject to conditions presented in the next subsection:

\[
\frac{1}{N} \|\tilde{y} - a_{00} z_0 - a_{01} z_1\|_2^2 + \frac{1}{N} \sum_{k=1}^{K} \|q_k - a_{k0} z_0 - a_{k1} z_1\|_2^2 + \lambda \sum_{k=1}^{K} (|a_{k0}| + |a_{k1}|)
\]

(1)

Here and throughout the text, \( \tilde{y} \) denotes the ”\( N \)-normalized” \( y \), i.e., \( y \) normalized in such a way that the arithmetic mean is 0 and the variance is \( N \). Furthermore, \( z_0 \) and \( z_1 \) denote the latent scores, while \( a_{k0} \) and \( a_{k1} \), for \( k \in \{0, 1, 2, \ldots, K\} \), denote the score/component loadings. Finally, \( q_{ik} \) is an unknown monotone transform of \( x_{ik} \), i.e., \( q_{ik} = f_k(x_{ik}) \) where \( f_k(\cdot) \) is unknown. Since \( x_{ik} \) only takes values \( \{-1, 0, 1\} \), we can equivalently define \( q_{ik} \) as follows:

\[
q_{ik} = \begin{cases} 
\alpha_{k0} & \text{if } x_{ik} = -1 \\
\alpha_{k0} + \alpha_{k1} & \text{if } x_{ik} = 0 \\
\alpha_{k0} + \alpha_{k1} + \alpha_{k2} & \text{if } x_{ik} = 1
\end{cases}
\]
where we require that $\alpha_{k1} \geq 0$ and $\alpha_{k2} \geq 0$ for all $k$'s.

Intuitively the first element in equation (1) corresponds to the prediction loss of the review scores, the second element serves as the prediction loss of the extracted features, and the last element represents a penalty for model complexity. If the $x_{ik}$ is continuous, transformations to $q_{ik}$ are not needed. However, $x_{ik}$ is discrete and it can be argued that a higher value of $x_{ik}$ represents a better evaluation, hence we introduce an unknown transformation to account for the possible non-linearity of $x_{ik}$ while maintaining the order of measurements. Lastly, the penalty is introduced to reach a more parsimonious solution. Note that the coefficients with respect to $y$'s are not penalized.

The above model can be also formulated in the following convenient fashion. Let $Q$ be an $N \times (K + 1)$ matrix, with $q_{(0)} = \tilde{y}_0$ and $q_{(k)} = q_k$, for $k \in \{1, 2, \ldots, K\}$, $Z$ an $N \times 2$ matrix with $z_{(0)} = z_0$ and $z_{(1)} = z_1$, and $A$ an $(K + 1) \times 2$ matrix with entries $a_{ik}$. Expression (1) can then be conveniently represented as:

$$\|Q - ZA^T\|_F^2 + \lambda \sum_{k=1}^{K} (|a_{k0}| + |a_{k1}|)$$  \quad (2)

4.2.2. Identification Issues. In general the model is not identified without further restrictions, which is clear as setting $Q$, $Z$, $A$ to zero matrices will result in perfect yet degenerate solutions. To avoid this situation, we apply the following restrictions:

1. $Q$ is N-normalized, i.e., each column of $Q$ has mean 0 and variance $\frac{1}{N}$

2. $Z^T Z = NI$, where $I$ is the identity matrix.

With the above restrictions, the above model is not identified given $Q$ when $\lambda = 0$, because by rotating both $Z$ and $A$ appropriately one could arrive at the same minimum. In classical Principal Component Analysis, this is resolved by exploiting the Singular Value Decomposition [51]. Generally, the Singular Value Decomposition takes the following form. For a $d_1 \times d_2$ matrix $C$ with $d_1 > d_2$, there exists an unique decomposition such that

$$C = K\Lambda L^T$$

where $K$ is a $d_1 \times d_2$ matrix, $\Lambda$ is a $d_2 \times d_2$ diagonal matrix, and $L$ is a $d_2 \times d_2$ matrix. Furthermore, both $K^T K = NI$ and $L^T L = \frac{1}{N}$. $Z$ and $A$ can be chosen as the first 2 columns of $K$ and $L\Lambda$ respectively. This decomposition provides an important benchmark as initial values for testing the validity of certain algorithms.

In case $\lambda > 0$, although unidentification by rotation is no longer an issue, some additional restrictions are still required to ensure proper identification. Specifically, one can change the sign of either column of $Z$ and $A$ simultaneously to arrive at the same minimum. Furthermore, exchanging the column of $Z$ and $A$ will also result in the same minimum. Although these unidentifications result only in different interpretation of latent scores in empirical analysis, normalization is needed such that we can compare the results in Monte Carlo studies. To this end, we require $a_{00} > a_{01} \geq 0$. 

4.3. General Description of the Algorithm. Since an analytical solution is not known, we rely on the following general algorithms, the details of which will be presented in the following section.

**Algorithm 1 General Algorithm**

1. Start with $\lambda = 0$
2. Initialize $Q$, $Z$, $A$. Repeat the following steps until $A$ converges.
   a. Given $Z,A$, minimize w.r.t. $Q$.
   b. Given $Q,A$, minimize w.r.t. $Z$.
   c. Given $Q,Z$, minimize w.r.t. $A$.
3. Increase $\lambda$ by a small predetermined value.
4. REPEAT step 2 and 3 until $\lambda$ reaches a predetermined threshold.
5. Choose the appropriate value of $\lambda$ by a modified version of cross-validation.

4.3.1. Alternative Approaches. Before we present the details of our algorithms, we briefly dedicate a section to other existing methods that achieve a shrinkage estimation of the score loadings. Although all these approaches are only directly applicable to continuous data sets, it is theoretically possible to adapt them to suit our situation. The purpose of this subsection is, however, not to compare the validity of these approaches, but rather to briefly outline several difficulties in adjusting these methods to tame our problem.

One possible approach, without invoking the L-1 penalty, is to set the estimated component loadings smaller than a certain threshold to zero. This can be achieved in two ways: either one can rely on a method similar to hypothesis testing using the bootstrap [34], or one can set parameter values to 0 for a suitably determined threshold [29]. The former algorithm could prove computationally expensive even if a moderate number of bootstrap replications are used, and the latter algorithm will require substantial improvement to adjust to our settings since it is not clear how to select the threshold in our nonlinear setting.

If one were to use shrinkage penalties, there are still two approaches available. As a natural start point, one might attempt to adapt a penalized likelihood scheme, similar to [42]. We have implemented one variation of this algorithm, the details of which can be found in appendix 1, albeit with different methods to estimate the latent scores. Unfortunately, the performance is less ideal, possibly due to the special properties of our data set. To illustrate the reason, consider the ordered Probit model, a standard econometric model for ordered data [11]. Denote the evaluation of feature $k$ for review $i$ as $x_{ik}$, the ordered Probit model takes the following form,

$$
\begin{align*}
x_{ik} \text{ is } \begin{cases} 
\text{negative} & \text{if } \sum_{l=1}^{L} \eta_{il} \gamma_k + \epsilon_{ik} < -\xi^2 \\
\text{not present} & \text{if } -\xi^2 \leq \sum_{l=1}^{L} \eta_{il} \gamma_k + \epsilon_{ik} \leq \xi^2 \\
\text{positive} & \text{if } \sum_{l=1}^{L} \eta_{il} \gamma_k + \epsilon_{ik} > \xi^2 
\end{cases}
\end{align*}
$$

where $L$ is the dimension of latent scores, $\eta_{il}$ is the latent score, $\gamma_k$ is the score loadings, $\xi_{ik}$ is threshold parameter, and $\epsilon_{ik}$ is noise term, assumed to follow a normal distribution\(^{16}\).

\(^{16}\)One alternative is to assume a logistic distribution. Empirically, the difference between these models are minimal [11].
Ideally, one would apply the L-1 penalty to $\gamma$’s in hope of reaching a parsimonious model. Unfortunately, even if the latent scores were known, the estimation would still turn out to be difficult, as we discovered in Appendix 1. One possible explanation is that our data set is sparse, in that the evaluations of a large number of features are not present in individual review. For this to happen, either $\sum_{l=1}^{L} \eta_{l}^{i} \gamma_{k}$ needs to be close to 0, or $\xi_{ik}^{2}$ needs to be large. As a consequence, a global minimum is only achieved when the initial values are considerably close to the truth, and a slight deviation from the truth will result in a local minimum. Consequently, this approach is difficult to implement in our settings.

Finally, one can apply an L-1 penalty to the our settings. In [64] a similar approach is implemented, the data set of which is continuous. Different from our methods though, [64] first computes the SVD decomposition and then regress the latent evaluations on the original data set with an L-1 penalty. It is possible to adapt this approach to our settings by alternatively transform the data and apply the proposed approach. However, since the transformation of data will both affect the results of SVD decomposition as well as the regressors, it is not clear whether this approach will result in an ideal convergence. Furthermore, since SVD is computationally costly, the computational efforts may prove demanding. Consequently, substantial adjustment is required.

4.4. Details of the Algorithm.

4.4.1. Initialization. Case: $\lambda = 0$

In order to speed up the convergence as well as to avoid finding a local minimum, appropriate initial values need to be chosen. More specifically, we choose the initial value of $Q$ by normalizing the original data matrix, i.e., we set $q_{ik} = x_{ik}$ and then re-normalize the matrix $Q$. Once $Q$ is set, we exploit the Singular Value Decomposition of $Q$ to obtain an appropriate initial value of $Z$ and $A$. Assuming the Singular Value Decomposition of $Q$ is given by $Q = SUV^T$, we set $Z$ and $A$ to be the first 2 columns of $S$ and $VU$.

Case: $\lambda > 0$

When $\lambda > 0$, one can not rely on the Singular Value Decomposition even if $Q$ is known, and analytical solutions are not available. Instead, since in each iteration we propose a new $\lambda$ by increasing its value by a small step, we use solutions of $Q, A$ and $Z$ from previous iteration.

The validity of this initialization strategy is tested using a Monte Carlo analysis, the results of which can be found in the next section.

4.4.2. Given $Z, A$, minimize w.r.t. $Q$. Before we present the details of this step, it should be noted that the target quantity only relies on $ZA^T$, and hence can be calculated prior to any iteration. We denote this quantity to be $T$, and the minimization problem can re-formulated as follows:

$$\min_{q_{(1)}, q_{(2)}, \ldots, q_{(K)}} \sum_{k=1}^{K} \| q_{(k)} - t_{(k)} \|_2^2$$

subject to,
\[
\sum_i q_{ik} = 0
\]

and
\[
\sum_i q_{ik}^2 = N^2
\]

We make the following observations. First, the optimization of \( q_k \) does not depend on \( q_{k'} \) for \( k \neq k' \). Therefore, we can drop sub-index \( k \) for simplicity. Second, since we have assumed a monotone transformation of \( x_{ik} \)'s, it is equivalent to solve the optimization problem w.r.t. \( \alpha_0, \alpha_1 \) and \( \alpha_2 \), where
\[
q_i = \begin{cases} 
\alpha_0 & \text{if } x_i = -1 \\
\alpha_0 + \alpha_1 & \text{if } x_i = 0 \\
\alpha_0 + \alpha_1 + \alpha_2 & \text{if } x_i = 1 
\end{cases}
\]

with \( \alpha_1 \geq 0 \) and \( \alpha_2 \geq 0 \).

In this notation, the restriction becomes,
\[
(N - n_1 - n_2)\alpha_0 + n_1(\alpha_0 + \alpha_1) + n_2(\alpha_0 + \alpha_1 + \alpha_2) = 0
\]

and
\[
(N - n_1 - n_2)\alpha_0^2 + n_1(\alpha_0 + \alpha_1)^2 + n_2(\alpha_0 + \alpha_1 + \alpha_2)^2 = N^2
\]

where \( n_1/n_2 \) denotes the number of 0’s/1’s respectively.

The solution to the above two equations w.r.t. \( \alpha_0 \) is:
\[
\begin{align*}
\alpha_1 &= -\frac{Na_0+2n_1\sqrt{(N-n_1+n_2)+(Nn_1+Nn_2)/(n_1n_2)}}{n_1+n_2} \\
\alpha_2 &= \frac{N(\alpha_0^2(N-n_1+n_2)+Nn_1+Nn_2)/(n_1n_2)}{(n_1+n_2)}
\end{align*}
\]

For an admissible solution to exist we need the following restriction,
\[
-Nn_2/(N-n_2) \leq \alpha_0 \leq -\sqrt{N(n_1+n_2)/(-N+n_1+n_2)}
\]

It can be shown that the r.h.s of the inequality is always larger than the left side, and therefore solutions always exist.

With these bounds and equalities, one can adopt Golden Section Search to minimize the target quantity.

4.4.3. Given \( Q, A \), minimize w.r.t. \( Z \). First we reformulate the problem in the following form:
\[
\min_{z_{(0)}, z_{(1)}, \ldots, z_{(K)}} \frac{1}{N} \sum_{k=0}^{K} \left\| q_{(k)} - a_k z_{(0)} - a_k z_{(1)} \right\|_2
\]

subject to
\[
\begin{align*}
z_{(0)}^T z_{(0)} &= N \\
z_{(0)}^T z_{(1)} &= 0
\end{align*}
\]
\[ z_{(1)}^T z_{(1)} = N \]  
(8)

The natural starting point is to apply a Lagrangian Multiplier scheme in hope of obtaining analytical solutions. By straightforward algebra, the first order conditions are,

\[
\begin{cases}
-\frac{2}{N} \sum_{k=0}^{K} (q_{(k)} - a_{k0} z_{(0)} - a_{k1} z_{(1)}) a_{k0} + 2\mu_1 z_{(0)} + \mu_2 z_{(1)} = 0 \\
-\frac{2}{N} \sum_{k=0}^{K} (q_{(k)} - a_{k0} z_{(0)} - a_{k1} z_{(1)}) a_{k1} + 2\mu_3 z_{(1)} + \mu_2 z_{(0)} = 0
\end{cases}
\]
(9)

where we have denoted the Lagrangian Multiplier by \( \mu \)'s.

Multiplying the first equation by \( z_{(0)} \), \( z_{(1)} \) the second by \( z_{(1)} \) and using restrictions (6), (7) and (8), we can solve for \( \mu \)'s given \( z \)'s. Specifically,

\[
\mu_1 = \frac{1}{N^2} \sum_{k=0}^{K} (q_{(k)}^T z_{(0)} - Na_{k0}) a_{k0}
\]
\[
\mu_2 = \frac{2}{N^2} \sum_{k=0}^{K} (q_{(k)}^T z_{(1)} - Na_{k1}) a_{k0}
\]
\[
\mu_3 = \frac{1}{N^2} \sum_{k=0}^{K} (q_{(k)}^T z_{(1)} - Na_{k1}) a_{k1}
\]

Unfortunately, substituting these solutions into the original equations will lead to nonlinear equations for the \( z \)'s, which can only be solved numerically. Since the number of unknown parameter is large (\( N \times 2 \)), it is likely that numerical algorithms may converge only to a local minimum and may be computationally inefficient. To deal with this problem, we rely on the following general routine of duality [7].

Let us first introduce general settings of solutions by duality in constrained minimization problems. In general form, suppose we have a scalar target function \( f(x) \), which we want to minimize subject to \( g_i(x) \leq 0 \). The following scheme can be applied,

1. Create the Lagrangian: \( L(x, u) = f(x) + u^T g(x) \)
2. Create the dual function: \( L^*(u) = \min_x (f(x) + u^T g(x)) \)
3. Maximize the dual function subject to conditions that \( u \geq 0 \)

The above scheme only applies to inequality constraints, and the success of this algorithm relies on the fact that \( L^*(u) \) is easily solvable. While the latter requirement is satisfied in this setting, the former one is problematic if the duality routine is directly applied. In effect, if the duality routine is applied directly, instead of solving the target problem, we solve:

\[
\min_{z_{(0)}, z_{(1)}, \ldots, z_{(K)}} \frac{1}{N} \sum_{k=0}^{K} \|q_{(k)} - a_{k0} z_{(0)} - a_{k1} z_{(1)}\|_2^2
\]

subject to

\[
z_{(0)}^T z_{(0)} \leq N
\]
In general, the extremum will not be attained with equality.  
In order to tailor this algorithm to fit our problem, we reformulate the 
minimization problem as,
\[
\begin{align*}
\min_{z(0), z(1), \ldots, z(K)} & \frac{1}{N} \left\| q(k) - a_{k0} z(0) - a_{k1} z(1) \right\|_2 \\
& + p_1 (N - z_T^{(0)} z(0)) + p_2 (z_T^{(0)} z(1)) + p_3 (N - z_T^{(1)} z(1)) 
\end{align*}
\]  
subject to
\[
\begin{align*}
z_T^{(0)} z(0) & \leq N 
\end{align*}
\]
\[
\begin{align*}
z_T^{(0)} z(1) & \leq 0 
\end{align*}
\]
\[
\begin{align*}
z_T^{(1)} z(1) & \leq N
\end{align*}
\]
where \( p_1, p_2, p_3 \) are large positive numbers.  
The rationale of the algorithm is as follows. The extra terms in expression
\[(10)\] represent the penalty for violation of equality constrained \((6), (7)\) and
\((8)\). Suppose one would set \( z_T^{(0)} z(0) < N \), then \( p_1 (N - z_T^{(0)} z(0)) \) will be
large and this solution will no longer minimize expression \((10)\). Hence by
applying penalties, one “forces” conditions \((11), (12)\) and \((13)\) to be actual
equalities.

To carry out the algorithm, one needs to derive a convenient expression
for the dual function, to this end, first note that the Lagrangian is:
\[
\frac{1}{N} \sum_{k=0}^{K} \left\| q(k) - a_{k0} z(0) - a_{k1} z(1) \right\|_2^2 \\
+ (p_1 - \mu_1)(N - z_T^{(0)} z(0)) + (p_2 - \mu_2)(z_T^{(0)} z(1)) + (p_3 - \mu_3)(N - z_T^{(1)} z(1))
\]
And the first order conditions are
\[
\begin{align*}
-2 \frac{1}{N} \sum_{k=0}^{K} (q(k) - a_{k0} z(0) - a_{k1} z(1)) a_{k0} - 2(p_1 - \mu_1) z_T^{(0)} (0) - (p_2 - \mu_2) z_T^{(1)} (1) = 0 \\
-2 \frac{1}{N} \sum_{k=0}^{K} (q(k) - a_{k0} z(0) - a_{k1} z(1)) a_{k1} - 2(p_3 - \mu_3) z_T^{(1)} (1) - (p_2 - \mu_2) z_T^{(0)} (0) = 0
\end{align*}
\]
Re-arranging the terms, the above equations can be conveniently
expressed in the following form
\[
ZP = V
\]  
where
\[ V = \left[ \frac{2}{N} \sum_{k=0}^{K} q(k) a_{k0}, \frac{2}{N} \sum_{k=0}^{K} q(k) a_{k1} \right] \] (15)

and

\[ P = \left[ \frac{2}{N} \sum_{k=0}^{K} a_{k0}^2 + \mu_1 - p_1, \frac{2}{N} \sum_{k=0}^{K} a_{k0} a_{k1} + \mu_2 - p_2 \right] \]

It should be noted that the evaluation of \( V \) does not depend on values of the \( \mu \)'s, hence it can be calculated prior to any iteration. Furthermore, the dual function is equal to

\[ \frac{1}{N} \sum_{k=0}^{K} \left\| q(k) - VP^{-1} a(k) \right\| \] (16) + \( (p_1 - \mu_1)(N - z_{(0)}^T z_{(0)}) + (p_2 - \mu_2)(-z_{(0)}^T z_{(1)}) + (p_3 - \mu_3)(N - z_{(1)}^T z_{(1)}) \) (17)

where \( z_{(0)}, z_{(1)} \) is obtained from equation 14.

We test the validity of this subroutine as follows. We first generate 10000 \( Q \) randomly from a standard normal distribution, normalize \( Q \), and then apply the Singular Value Decomposition to \( Q \) to obtain \( A \). Based on this \( Q \) and \( A \), we apply our subroutine to compute \( Z \) and compare to the values obtained by Singular Value Decomposition. The subroutine is rather satisfactory as it converges to a global minimum in every instance with random initial values.

4.4.4. Given \( Q, Z \), minimize w.r.t. \( A \). Let us present the problem as follows,

\[ \min_{a_{k0}, a_{k1}, \ldots, a_{kK}} \sum_{i} (q_{ik} - a_{k0} z_{i0} - a_{k1} z_{i1})^2 + \lambda \sum_{k=1}^{K} (|a_{k0}| + |a_{k1}|) \] (18)

Since the penalty is only applied to the \( a \)'s with \( k \geq 1 \), we present the solutions for \( k = 0 \) and \( k \geq 1 \) separately.

**Case** \( k = 0 \) The problem is

\[ \min_{a_{k0}} \frac{1}{N} \sum_{i} (q_{i0} - a_{k0} z_{i0} - a_{k1} z_{i1})^2 \] (19)

Recall that we have required that \( a_{k0} > a_{k1} \geq 0 \).

The first order condition for \( a_{k0} \) is

\[ \frac{2}{N} \sum_{i} (q_{i0} - a_{k0} z_{i0} - a_{k1} z_{i1}) z_{i1} = 0 \]

Using the fact that \( \sum_{i} z_{i0}^2 = N \) and \( \sum_{i} z_{i1} z_{i0} = 0 \) we see that

\[ a_{k0} = \frac{1}{N} \sum_{i} q_{i0} z_{i1} \]

If \( a_{k0} \geq 0 \), we keep this solution. Otherwise we set \( a_{k0} \) to be 0.
Similarly, we can derive that

\[ a_{00} = \frac{1}{N} \sum_i q_i z_i^0 \]

and we keep the solution if \( a_{00} \geq a_{01} + \epsilon \), where \( \epsilon \) is a prefixed small number. Otherwise we set \( a_{00} \) equal to \( a_{01} + \epsilon \).

**Cases:** \( k \geq 1 \)

Before we start the exposition, it should be noted that the minimization w.r.t. \( a_{(k)} \) is independent of \( a_{(k')} \), and we drop the index \( k \) for simplicity. The minimization problem is

\[
\min_{a_{0}, a_{1}} \frac{1}{N} \sum_i (q_i - a_0 z_i^0 - a_1 z_i^1)^2 + \lambda (|a_0| + |a_1|) \tag{20}
\]

Again, by exploiting the condition \( Z^T Z = NI \), we can derive that the minimization is equivalent to computing

\[
\min_{a_{0}, a_{1}} -\frac{2}{N} \left( \sum_i q_i z_i^0 \right) + \lambda |a_0| + a_0^2 - \frac{2}{N} \left( \sum_i q_i z_i^1 \right) + \lambda |a_1| + a_1^2 \tag{21}
\]

Hence the minimization w.r.t. \( a_0 \) is independent of that w.r.t. \( a_0 \). For simplicity, we only present the details for \( a_0 \). By straightforward algebra, the solutions for \( a_0 \) can take three values

\[
\begin{cases} 
\frac{2}{N} \sum_i q_i z_i^0 + \lambda & \text{if } a_0 \geq 0 \\
0 & \text{if } a_0 = 0 \\
\frac{2}{N} \sum_i q_i z_i^0 - \lambda & \text{if } a_0 \leq 0 
\end{cases} \tag{22}
\]

Therefore it is sufficient to compare the function values evaluated at the above solutions.

4.4.5. **Choice of \( \lambda \)**. In general, we have adopted a cross-validation scheme to determine the appropriate value of \( \lambda \). Ideally, we would like to obtain the estimate of \( Q \) and \( A \) by minimizing

\[
\frac{1}{N} \left\| \tilde{y} - a_{00} z_{(0)} - a_{01} z_{(-1)} \right\|_2^2 + \frac{1}{N} \sum_{k=1}^K \left\| q_{(k)} - a_{k0} z_{(0)} - a_{k1} z_{(-1)} \right\|_2^2 + \lambda \sum_{k=1}^K (|a_{k0}| + |a_{k1}|) \tag{23}
\]

Let the solutions for \( A \) be denoted by \( \hat{A} \). We then evaluate the following quantities,

\[
\frac{1}{N} \left\| \tilde{y}' - \hat{a}_{00} z_{(0)}' - \hat{a}_{01} z_{(-1)}' \right\|_2^2 + \frac{1}{N} \sum_{k=1}^K \left\| q_{(k)}' - \hat{a}_{k0} z_{(0)}' - \hat{a}_{k1} z_{(-1)}' \right\|_2^2 \tag{24}
\]

where we differentiate the training and validation data set by denoting the latter with “’”.

While it is possible to obtain \( Q' \) based on the transformation obtained from the training set, it is only possible to obtain the values of latent evaluations by using the data from the validation data set. This creates an
obstacle as ideally we would like to avoid training the parameters on the validation set to avoid over-fitting. Specifically, as $Z$ includes a large number of parameters, a more complex model, demonstrated as more non-zero coefficient in $A$'s, will offer more “opportunities” for $Z$ to adjust itself to reach a smaller value of expression (24). This is indeed the case if we were to estimate the $Z'$ with each $\lambda$ and corresponding $\hat{A}$, as is seen in figure 4.4.5.

To tackle this problem, we propose the following modified version of cross-validation. Through the rest of the text, we assume that $Q$ and $A$ have been estimated on a grid of values of $\lambda$. The framework of the algorithm can be summarized as follows:

**Algorithm 2 Modified Cross-Validation**

1: Initialize: Start with $\lambda$ equal to the largest available values for which the model is estimated; Calculate $Z'$.  
2: Decrease $\lambda$ by a small step. Obtain $\hat{A}$ and $Q'$ based on the new $\lambda$  
3: Calculate expression (24).  
4: Update $Z'$ based on the new $\lambda$.  
5: Repeat step 2-3 until $\lambda$ reaches 0.

As an illustration, we present a specific realization of this cross-validation strategy, the related result is shown in figure 4.4.5, the settings of which can be found in next section. In this example, the shrinkage parameter is chosen as 1.2 while the optimal value is 1.8. Unfortunately this tendency of under-selection of shrinkage parameter turns out to be a robust finding in our Monte Carlo study, although the difference in performance is acceptable. Some authors (e.g. [38]) propose adding a constant to the shrinkage parameter for optimality, possibly obtained from bootstrapping. We do not adopt these approach for two reasons: Firstly, bootstrapping is computationally demanding; Secondly, we discovered in our Monte Carlo studies that the appropriate adjustment depends on the level of sparsity of the true model, which is unknown to us in empirical data analysis.

4.5. General Comments. Before we present the results of Monte Carlo study, we would like to make the following general comments.

We begin our comments on the convergence property of our algorithm. Ideally, one would like to prove a newly proposed algorithm to converge to a global minimum via a mathematical rigorous method. Unfortunately, since our method deviates significantly from the standard convex settings [25], we are not able to do so, nor are we able to find a proof in a similar setting. However, we would like make the following two remarks. Firstly, since in each iteration we minimize the target function, our algorithms will not converge to a global maximum. Secondly, in generally, we do find that the algorithms converge to local minimum or fixed point. In order to assess significance of this problem, we perform a Monte Carlo study to test the stability of the algorithm, the details of which can be found in next section. In general, the algorithm demonstrates satisfactory stability.

17The reason why we only provide a specific reason is that the extreme point is “averaged” away if an averaged version is presented.
Next we would like to comment on the computational advantage of our algorithm. Firstly, apart from the initialization strategy, our algorithm only involves matrix/vector products, which is considered of order $N^2$, less than the algorithms involving matrix factorization, which is usually of order $N^3$. Especially, we have avoided repeatedly using Singular Value Decomposition in order to reach a significant performance gain. Secondly, we have implemented our algorithm in a parallel computing fashion. More specifically, while the complete algorithm is of sequential nature, the major steps of each step can be computed concurrently. In fact, a large proportion our routine can be computed in a perfect parallel fashion, in the sense that the computing is implemented completely independently across multiple processors. This is clear as when optimizing w.r.t. $Q$ and $A$, the evaluation of different $k$'s is independent. Furthermore, while optimization w.r.t. $Z$ is not perfectly concurrent, the evaluation of the sums in expression 16, which is relatively computationally intense, can be evaluated again perfectly concurrently. The implementation of our routine is based on the C++11 thread library, a short introduction of which, as well as a performance test of non-parallel version and parallel version, can be found in Appendix 2.
5. Monte Carlo Study

5.1. Introduction. In this part we present the settings and the results of our Monte Carlo study, which serves a benchmark for the performance of our model. We begin by details of our data generating processes, and then move to the results of Monte Carlo study.

5.2. Data Generation Processes.

5.2.1. General Settings. Before we present the details, we present a short, self-complete introduction to the ordered Probit model. The ordered Probit model is a standard model used to model ordinal discrete choice, and serves as building blocks for more complex models.

In its most general form, the ordered Probit model can be formulated as follows. Suppose we have an dependent variable $y_i$ which takes categorical values $\{0, 1, 2, \ldots, P\}$ for observation $i$, and dependent variables $x_{ik} \in \mathbb{R}$. An ordered Probit model assumes the existence of latent variable $y_i^* = x_i^T \beta + \epsilon_i$, which is linked to $y_i$ via the following relationships:

$$
y_i = \begin{cases} 
0 & \text{if } y_i^* \leq \alpha_0 \\
1 & \text{if } \alpha_0 < y_i^* \leq \alpha_1 \\
\vdots & \vdots \\
p & \text{if } y_i^* > \alpha_p
\end{cases}
$$

Assuming $\epsilon_i$ follows a normal distribution, we arrive at the ordered Probit model.

In our settings, the regressors are unknown. Instead we assume the existence of latent variables $\eta_{il}$, which are related to the evaluations of features in the following way

$$
x_{ik} = \begin{cases} 
-1 & \text{if } \sum_{l=1}^L \eta_{il} \gamma_k + \epsilon_{ik} < -\xi^2 \\
0 & \text{if } -\xi^2 \leq \sum_{l=1}^L \eta_{il} \gamma_k + \epsilon_{ik} \leq \xi^2 \\
1 & \text{if } \sum_{l=1}^L \eta_{il} \gamma_k + \epsilon_{ik} > \xi^2
\end{cases}
$$

Finally, we assumed that the review score $y_i$ is linearly related with the $\eta$'s.

It should be noted that our data generation process is different from our underlying statistical model. The reason for this choice lies in the non-parametric nature of our approach. Indeed, we have not assumed any probability structure in our model. While we could have attempted to base our Monte Carlo study on a setting closer to our statistical model, it would be desirable to test the validity of our model based on standard probabilistic models.

5.2.2. Choice of Parameter. Due to the limitation of computational power, we tried to choose parameters in such a way that the choice is as representative as possible. Specifically, we choose a parameter value such that the generated data is as close as possible to the data set that we are going to analyse when possible. If not possible, we randomly choose the parameters in each Monte Carlo replication.
To begin with, we generated the $\eta$'s in the following way. We first generate $\eta$'s from independent standard normal distribution, and then apply the Gram-Schmidt procedure to ortho-normalize the latent evaluations in accordance with our model. Note that the scale and dispersion is not important due to the ortho-normalization, and the latent evaluation always has variance one.

The noise is chosen from normal distributions with expectation 0 and standard deviation $\delta$. In our Monte Carlo study, 3 situations are included: (1) low noise setting ($\delta = 0.1$); (2) mid noise setting ($\delta = 0.3$) and (3) high noise setting ($\delta = 0.5$). Note that these numbers should not be interpreted as noise to signal value since errors present in each of the latent evaluations, and hence even a small fluctuation will result in considerate changes in the values of $x_{ik}$, which has to be accounted for by the algorithm.

The number of observations is 600 and the number of latent evaluations is 62, a ratio chosen to resemble the real data. Furthermore, the threshold parameter $\xi$ is chosen in such a way that the 0's presented in the generated data set is as close to the real data set as possible.

Lastly, we choose the $\gamma$'s in the following way. We first choose the sparsity rate, a number between 0.1 and 0.2 as the fraction of non-zero $\gamma$'s. Then the values of these $\gamma$'s are chosen from a uniform $[-1, 1]$ distribution, while the rest is set to 0. In our simulation, we make sure that at least one parameter for each latent evaluation will be non-zero. Otherwise one could discard the latent evaluation with 0 loadings.

5.3. Monte Carlo Procedure. 1000 Monte Carlo samples are generated for each setting (low/medium/high noise) and the models are estimated according to the methods in section 4. In order to test whether the estimates converge to a global minimum, after determining the shrinkage parameter, we also estimate the model again using the true parameter values as a benchmark by comparing the parameter estimate with that obtained from our initialization strategy.

5.4. Results. In general, our algorithm is satisfactory in demonstrating decent stability and recovery rate.

We start our exposition on the stability of the algorithm, which refers to the ability to reach global minimum with appropriate initializing strategies. To study the stability, one needs the global minimum as benchmarks. To this end, we first perform a Monte Carlo study that tests whether initialization with the true parameter will lead to a global minimum. Specifically, 100 Monte Carlo samples are generated, each estimated with 100 random initial values. We found no initialization strategy that leads to a better estimate than those initialized with the truth. Hence we use the parameter values obtained by initializing from the truth as the benchmark.

In summary, the algorithm converges to a global minimum in most cases despite a few anomalies. Specifically, 14/32/68 of the estimations do not converge to the global minimum in low/medium/high noise settings respectively. While this number is potentially small, it does testify the necessity

---

18The number is chosen mostly due to computational consideration
to apply different initialization strategy for a robustness check in empirical data analysis.

We next present the results of the parameter recovery rate, starting with the recovery rate of latent evaluations. Figure 3 presents the correlation between the estimated latent evaluations with the truth, with the results of non-global-convergent results deleted for ease of presentation. The algorithm yields satisfactory recovery in the low/medium noise settings and a decent result in the high noise settings.

**Figure 3. Recovery Rate of Latent Evaluation:** In all of the panels, the left box denotes the Pearson-correlation of the first latent evaluation with the truth, while the right one denotes the second.

We next present the recovery result of the score loadings. Since the scale in our algorithm and the data generating process is different, it is not possible
to directly compare the magnitudes of the estimates. Instead, we compute the results of correlations between true parameter values that are non-zero with those estimated from our approach. Note that this approach is also empirically reasonable since only the relative difference in score loadings affect interpretation. The results are presented in figure 5.4 and in all of the three settings the recovery rate is satisfactory.

Figure 4. Recovery Rate of Score Loadings: From left to right lies the Pearson correlation between the estimated score loadings with the truth in small/medium/large noise setting.

19The reason why we do not present the correlation of all the score loadings is because a large number of 0’s are present, hence including all the coefficients will artificially boost the correlation.
Finally, we present the correlation selection rate of the zero/nonzero parameters, as is seen in table 1. In general, the algorithm over selects non-zero elements, albeit only in a moderate extent. Furthermore, it is noteworthy that the selection rates are similar in all of noise settings. Thus the different in recovery rate more of a result of error in estimation magnitude of score loadings.

<table>
<thead>
<tr>
<th>Settings</th>
<th>Rates of Correctly Selecting Non-Zero Coefficient</th>
<th>Rates of Correctly Selecting Zero Coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low Noise</td>
<td>96.63%</td>
<td>82.20%</td>
</tr>
<tr>
<td>Medium Noise</td>
<td>92.65%</td>
<td>79.21%</td>
</tr>
<tr>
<td>High Noise</td>
<td>91.59%</td>
<td>76.93%</td>
</tr>
</tbody>
</table>

Table 1. Correct Selection Rate
6. Empirical Data Analysis

6.1. Data Overview. In this section, we present the result of applying our approach to an empirical data set extracted from the website metacritic.com. Founded in 2001, metacritic.com is one of the largest online review websites featuring more than 3 million user reviews, the subject of which covers games, movies, actors/actress, among many others. In this website, the users are allowed to post a number between 0 and 10 to represent the general favoribility, and a detailed text review up to 3000 words. These online review scores and texts act as an important source of consumer online word-of-mouth, and contain valuable information for the marketeers.

Our empirical data set is based on reviews of first-person shooter games, a popular genre where players experience the combat systems based on guns from a first-person perspective. We focus on a particular genre since elements that determine user preferences in one genre might not even be present in another one. For example, while “weapons” may be an important feature in the game Modern Warfares, the word “weapons” do not present itself at all in the game Simcity. For this reason, mixing data from different genre may lead to undesirable results. It should be noted, however, although our results are based on data from a specific genre, our approach is applicable to others without need of any modification.

The data set is collected from 3rd June 2008 to 23rd July 2013, including 120 titles and 4902 reviews. Among these data, we further select titles with more than 30 user reviews, leaving 26 titles and 3198 reviews. The average review is 155 words in length, which is substantial considering the fact that reviewers are not compensated. In these reviews, the total number of extractable features is 1608 before pruning, and 126 after pruning by frequency. Since it is not possible to determine the opinion sentiment for reviews with none of these features present, 296 reviews are further deleted since all the evaluations for selected features are nonpresent, leaving finally 2903 reviews to analyze. We split the training and validation samples according to a 50:50 scheme. The results are presented by merging the training and validation sample set. As a robustness check, we also adapt 20 randomly selected initialization values, although none has resulted in a better estimate than the initialization proposed in section 4.

In general, the fitting of the model is satisfactory, as the Spearman correlation between the first and second latent attribute is 0.67 and 0.53 respectively, representing a decent explanation power of the latent attributes towards the general sentiment level.

---

20http://www.metacritic.com/
21http://www.metacritic.com/about-metacritic
22The games in this genre is pre-classified by metacritic.com
23https://www.callofduty.com/mw3
24http://www.simcity.com/
25For Robustness check, a 70:30 split is also applied, resulting in essentially identical results
6.2. Macro Marketing Analysis. We begin by examining the score loadings of latent evaluations, which serve as the foundation for further interpretation such as marketing structure analysis. Out of 126 features, 29 are selected, with the results presented in Table 2. It is worth commenting that the selected features are not necessarily most frequently presented, a finding that coincides with [16]. Another notable finding is that several features related to the quality of storyline has non-zero coefficients, such as “story”, “character”, and “acting”. This confirms that apart from playability, a carefully designed storyline can also help the success of the game, even when the game is action-based.

We next present our interpretation of the latent attributes. In addition to features related with the quality of the story, a significant number of features related with playability shows positive loadings on first latent attribute, such as “weapon”, “system”, “customizability”, “map”. For this reason, we interpret the first latent attribute as the playability and storyline. For the second latent attribute, we note that several features related to the quality of the sound and graphics have positive loadings on and only on the second attribute, such as “sound”, “design”, “visual”, “effects” and “textures”. For this reason, we interpret the second latent attribute as sound and graphic quality.

<table>
<thead>
<tr>
<th>Feature</th>
<th>L1</th>
<th>L2</th>
<th>Feature</th>
<th>L1</th>
<th>L2</th>
</tr>
</thead>
<tbody>
<tr>
<td>games</td>
<td>19.17</td>
<td>11.61</td>
<td>gun</td>
<td>12.49</td>
<td>0</td>
</tr>
<tr>
<td>graphic</td>
<td>16.81</td>
<td>14.01</td>
<td>design</td>
<td>0</td>
<td>11.92</td>
</tr>
<tr>
<td>story</td>
<td>11.73</td>
<td>0</td>
<td>visuals</td>
<td>0</td>
<td>15.78</td>
</tr>
<tr>
<td>gameplay</td>
<td>16.03</td>
<td>17.38</td>
<td>levels</td>
<td>17.74</td>
<td>17.07</td>
</tr>
<tr>
<td>multiplayer</td>
<td>11.48</td>
<td>0</td>
<td>effects</td>
<td>0</td>
<td>22.44</td>
</tr>
<tr>
<td>player</td>
<td>14.85</td>
<td>0</td>
<td>customization</td>
<td>14.72</td>
<td>-11.47</td>
</tr>
<tr>
<td>fun</td>
<td>18.11</td>
<td>12.97</td>
<td>voice</td>
<td>0</td>
<td>31.71</td>
</tr>
<tr>
<td>maps</td>
<td>17.39</td>
<td>17.54</td>
<td>portal</td>
<td>11.47</td>
<td>0</td>
</tr>
<tr>
<td>weapons</td>
<td>15.64</td>
<td>12.78</td>
<td>puzzles</td>
<td>-12.64</td>
<td>0</td>
</tr>
<tr>
<td>system</td>
<td>14.77</td>
<td>0</td>
<td>textures</td>
<td>0</td>
<td>18.59</td>
</tr>
<tr>
<td>controls</td>
<td>15.31</td>
<td>0</td>
<td>points</td>
<td>16.77</td>
<td>20.09</td>
</tr>
<tr>
<td>online</td>
<td>0</td>
<td>11.78</td>
<td>acting</td>
<td>32.07</td>
<td>0</td>
</tr>
<tr>
<td>character</td>
<td>11.78</td>
<td>0</td>
<td>map</td>
<td>16.665</td>
<td>12.04</td>
</tr>
<tr>
<td>experience</td>
<td>13.38</td>
<td>21.11</td>
<td>modes</td>
<td>0</td>
<td>12.16</td>
</tr>
<tr>
<td>sound</td>
<td>0</td>
<td>17.87</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2. Score Loadings: L1/L2 represents loadings on first/second dimension

We move on to examining the Scatter plot and the Contour plot of latent evaluations in figure 5, aggregated across the games. From the Contour Plot, it can be concluded that the evaluations are mostly homogeneous, i.e., no subgroups are present. On the other hand, a careful inspection of the scatter plot reveals that significant dispersion exists along both latent attributes. Furthermore, in general, extreme positive evaluation of second latent attributes are not as common as the first one, possibly indicating...
a general unfavorability of the performance of the targeted games on the second latent attributes.

Figure 5. Scatter Plot and Contour Plot of Latent Evaluations

The product information can be retrieved from a centroid plot (figure 6). Each centroid in the plot is computed as the arithmetic mean of the associated evaluations. For ease of presentation, we have marked the centroid with the id of the titles, and the content can be found in table 3.
Figure 6. Centroid Plot of Products

<table>
<thead>
<tr>
<th>ID</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Aliens: Colonial Marines</td>
</tr>
<tr>
<td>2</td>
<td>Battlefield: Bad Company</td>
</tr>
<tr>
<td>3</td>
<td>Battlefield: Bad Company 2</td>
</tr>
<tr>
<td>4</td>
<td>Borderlands</td>
</tr>
<tr>
<td>5</td>
<td>Brink</td>
</tr>
<tr>
<td>6</td>
<td>Call of Duty: Black Ops</td>
</tr>
<tr>
<td>7</td>
<td>Call of Duty: Modern Warfare 2</td>
</tr>
<tr>
<td>8</td>
<td>Call of Duty: World at War</td>
</tr>
<tr>
<td>9</td>
<td>Crisys 2</td>
</tr>
<tr>
<td>10</td>
<td>Dead Island: Riptide</td>
</tr>
<tr>
<td>11</td>
<td>Deus Ex: Human Revolution</td>
</tr>
<tr>
<td>12</td>
<td>Dust 514</td>
</tr>
<tr>
<td>13</td>
<td>F.E.A.R 3</td>
</tr>
<tr>
<td>14</td>
<td>Homefront</td>
</tr>
<tr>
<td>15</td>
<td>Hyperdimension Neptunia Victory</td>
</tr>
<tr>
<td>16</td>
<td>MAG</td>
</tr>
<tr>
<td>17</td>
<td>Metro: Last Light</td>
</tr>
<tr>
<td>18</td>
<td>Mirror’s Edge</td>
</tr>
<tr>
<td>19</td>
<td>Portal 2</td>
</tr>
<tr>
<td>20</td>
<td>Rage</td>
</tr>
<tr>
<td>21</td>
<td>Resistance 2</td>
</tr>
<tr>
<td>22</td>
<td>Resistance 3</td>
</tr>
<tr>
<td>23</td>
<td>Resistance: Fall of Man</td>
</tr>
<tr>
<td>24</td>
<td>Super Motherload</td>
</tr>
<tr>
<td>25</td>
<td>The Amazing Spiderman</td>
</tr>
<tr>
<td>26</td>
<td>Unreal Tournament 3</td>
</tr>
</tbody>
</table>

Table 3. Game id and Game Title
In order to demonstrate the possible usage of this plot, consider for example game 2 (Battle Field: Bad Company) and game 6 (Call of Duty: Black Ops), the former of which is considered more successful than the latter. The two games enjoy similar scores on the first latent attribute, while differing significantly from the second attribute. For these considerations, it could be seen that the difference in reception majorly comes from the lack of impressive sound and graphic quality in Call of Duty: Black Ops. As another example, while the performance of Portal 2 (id 19) seems to be average in terms of its sound and graphics quality, the playability and the story seems to be widely acclaimed.

6.3. **Micro Marketing Analysis.** In this subsection, we focus on the micro-level analysis of aforementioned games, *Battle Field: Bad Company* and *Call of Duty: Black Obs*. As is demonstrated in last section, the market position of the two titles differ significantly, and we try to identify the reasons behind the difference in this section.

Starting from the game *Battle Field: Bad Company*, we first apply our methods to plot the market trajectory, as can be seen in figure 6.3. The upper left side of panel demonstrates the scatter plot of latent evaluations in both dimensions. Similar to the findings in figure 6.2, while most of reviews tend to be positive, there are substantial numbers of less satisfactory reviews. Based on this figure, marketeers could easily identify these reviewers and learn their reason of being less satisfied to facilitate further marketing decisions. Furthermore, the other three panels show the evolution of review score, first and second latent evaluations in time. We observe substantial fluctuations. This further testifies the need for an efficient means of monitoring the marketing for timely decisions.

On the other hand, the market monitor results for *Call of Duty: Black Obs* shows a distinctive trend, as presented in figure 6.3. First of all, from the scatter plot in the upper left panel, it is clear that the variations in evaluations are more significant in *Call of Duty: Black Obs*, indicating the title has sparked controversial receptions. Secondly, the evolving of review scores does not demonstrate a downward trend as in [33], but instead shows a seemingly upward result. However, a downward trend is observed in the first latent attributes. Moreover, the periodical trend in the review score seems to be more driven by the second latent attribute, confirming our findings that the difference between the two titles lie mostly in the evaluation of second latent attribute.
Figure 7. Market Monitoring of Battle Field: Bad Company
Figure 8. Market Monitoring of *Call of Duty: Black Ops*
In addition, our information can be also applied to understand the heterogeneity of consumers. Specifically, based on the latent evaluations, we first perform a hierarchical clustering analysis on *Battle Field: Bad Company*. The result is shown in the upper panel of figure 6.3. From the Dendrogram, it seems that the reviewers are heterogeneous, and different marketing strategy should be applied to different groups for maximal effect. To identify the group, we further perform a k-means clustering analysis, the result of which is presented in the lower panel of figure 6.3 with group number equal to three. It seems that the groups represent the different level of satisfaction of the reviewers, and this result again can be used to target specific group.

Turning to *Call of Duty: Black Obs*, a different pattern is again observed in figure 6.3. In accordance with the large variation in latent evaluation, the Dendrogram is indicates more heterogeneity among reviewers. A K-means clustering is once again performed, although the results are clearly influenced by reviews that strongly favour/disavour the title. This further testifies the need to apply different strategies to different target groups.
Figure 9. Consumer Heterogeneity Analysis of *Battle Field: Bad Company*
Figure 10. Consumer Heterogeneity Analysis of *Call of Duty: Black Ops*
7. Conclusion: Contributions and Future Studies

7.1. Contributions. As the Internet is becoming increasingly widespread, the interest in the possibility of extracting market information from various forms of online data sources has grown considerably. One such data source is online review, considered to be an important part of consumer word-of-mouth and containing valuable information necessary when it comes to boosting marketing decisions. For this reason, consistent efforts have been devoted to this end, both in marketing literature and in the one on natural language processing.

This study contributes to improvement on existing methods. More specifically, the approach not only enables us to extract product features that determines the overall consumer sentiment level, but also allows us to reveal the structural relationships of them. In this fashion, extensive information can be extracted and presented to marketeers both straightforwardly and insightfully. Through implementing this method, complex and rich data is turned into powerful decision-making and supporting materials which could significantly improve the efficiency and accuracy of marketing decisions.

In additional to the aforementioned utility of this approach, it also significantly contributes to the development of the methodologies used in extracting information from the Internet. Specifically, one of the main obstacles in these analyses lies in the fact that while the Internet contains a plethora of information, the latter usually has to be extracted from an amassment of unstructured and complex messages. In this regard, a successful application needs to both take into account the possibly sparse nature of online data to achieve accurate and stable estimation, as well as to be computationally efficient. Both such requirements are fulfilled through using our algorithm, and hence our approach not only serves as a valuable tool for information retrieval from online reviews, but also as the next building block for further development of model building in the information age.

7.2. Future Studies. Three possibilities present themselves for future study. To begin with, our study has mostly relied on a pre-coded lexicon of adjectives with sentiment orientation. While this method is efficient and sufficiently accurate, other elements could also be utilized for information retrieval from the natural language. For example, adjectives are not only oriented, but can also possibly possess different magnitudes of emotional intensity [60]. Yet another example, the order of words in a specific sentence, and more generally its grammatical structure may also prove informative [43]. While it is possible to apply these elements in such a way as to build a more powerful model, the main difficulty lies in integrating different forms of possibly model-specific information.

One other possible research opportunity is to further model the heterogeneity in the consumer evaluations, particularly in terms of how lenient the consumers are in evaluation. In our approach, the assumption has been made that the transformation of each evaluation corresponds to the same unknown quantities even in case of different consumers, which is not necessarily true since different individuals may evaluate differently. Evidence has shown that consumer evaluation varies in leniency, which is determined by a
number of individual or situational factors \[53\]. This provides the possibility of assuming different levels of transformations in cases where the evaluations are the same, such as in \[10\]. In order to successfully implement this proposal, however, modellers may have to consider all of the extra parameters this approach may create.

Lastly, while this study focuses primarily on online reviews, our approach is applicable to different settings. In principle, this methodology can be applied to any source of information that includes evaluations of target products, including results retrieved from search engines. The inclusion of such data sources would not only cover a broader aspect of online sentiment evolution, but also might provide additional information such as demographical factors, which could, in turn, facilitate marketing decision making. Unfortunately, the inclusion will likely result in an even larger amount of data that could be stored in distributed file systems such as Hadoop \[58\]. For this reason, substantial adjustments in our methods have to be made to combat these impediments, possibly through aggregating a collection of our models using model averaging techniques (e.g. \[14\], \[48\] and \[56\]).
8. Appendix 1: Likelihood Based Approach

8.1. Introduction. In this appendix we present the likelihood based model we originally developed. Unfortunately, due to numerical issues, the estimation of the models turn out to be unsatisfactory, as well as the computational speed.

8.2. Data. Here and throughout the rest of the text, we assume that we have training/validation data sets consisting of $N_1/N_2$ observations. With a slight abuse of notation, we use $N$ to denote either $N_1$ or $N_2$, the exact content of which will be clear by the contexts. In each data set, we assume the existence of $K$ latent features. The review score for review $i$ is denoted as $y_i$, a quantity that ranges from 0 to 10. Furthermore, we denote the evaluation for feature $k$ and review $i$ as $x_{ik}$, and the coding is as follows:

$$x_{ik} = \begin{cases} 
-1 & \text{If the evaluation is negative} \\
0 & \text{If the feature is not present} \\
1 & \text{If the evaluation is positive}
\end{cases}$$

8.3. Model and Likelihood. The starting point for the model is the ordered logit regression. In general, the logit regression can be summarized as follows. Suppose we have an dependent variable $y_i$ which takes categorical values $\{0, 1, 2, \ldots, P\}$ for observation $i$, and dependent variables $x_{ik} \in R$. An ordered logit regression assumes the existence of latent variable $y^*_i = x_{ik}^T \beta + \epsilon_i$, which is linked to $y_i$ via the following relationships

$$y_i = \begin{cases} 
0 & \text{if } y^*_i \leq \alpha_0 \\
1 & \text{if } \alpha_0 < y^*_i \leq \alpha_1 \\
\ldots & \ldots \\
p & \text{if } y^*_i > \alpha_p
\end{cases}$$

If $\epsilon_i$ follows a logistic distribution, then it can be shown that

$$P(y_i = p) = \begin{cases} 
\psi(\alpha_0 - x_{ik}^T \beta) & \text{if } y_i = 0 \\
\psi(\alpha_p - x_{ik}^T \beta) - \psi(\alpha_{p-1} - x_{ik}^T \beta) & \text{if } y_i \in \{1, 2, \ldots, P - 1\} \\
1 - \psi(\alpha_0 - x_{ik}^T \beta) & \text{if } y_i = P
\end{cases}$$

where $\psi(z) = \frac{e^z}{1 + e^z}$. In our model, the regressors are not known and have to be inferred. Instead, we assume the existence of latent evaluations $\eta_{id}$ and that conditional on $\eta_i$

$$P(y_i = -1|\eta_i) = \psi(\alpha_0 - \sum_{l=1}^L \beta_l \eta_{il})$$

$$P(y_i = 0|\eta_i) = \psi(\alpha_1 - \sum_{l=1}^L \beta_l \eta_{il}) - \psi(\alpha_0 - \sum_{l=1}^L \beta_l \eta_{il})$$

$$P(y_i = 1|\eta_i) = 1 - \psi(\alpha_1 - \sum_{l=1}^L \beta_l \eta_{il})$$
Here and throughout the texts, to ensure \( P(y_i = 0 | \eta_i) > 0 \), we reparametrize \( \alpha_1 = \alpha_0 + \alpha^2 \).

Similarly, we also have

\[
P(x_{ik} = 0 | \eta_i) = \psi(\delta^2 - \sum_{l=1}^{L} \gamma_{lk} \eta_{il}) - \psi(-\delta^2 - \sum_{l=1}^{L} \gamma_{lk} \eta_{il})
\]

\[
P(x_{ik} = 1 | \eta_i) = 1 - \psi(\delta^2 - \sum_{l=1}^{L} \gamma_{lk} \eta_{il})
\]

If conditional independence is assumed, the likelihood takes the product form, namely,

\[
P(y_i, x_{i1}, \ldots, x_{iK} | \eta_i) = P(y_i | \eta_i) \prod_{k=1}^{K} P(x_{ik} | \eta_i)
\]

8.4. Estimation Routine. Let us suppose we split the posteriors of the \( \eta \)'s into training and testing samples of size \( N_1 \) and \( N_2 \). The estimation framework can be summarized as follows,

1. Given \( \lambda \) and the posterior of the training samples, estimate the \( \{ \beta, \gamma \} \)'s and the \( \{ \alpha_0, \alpha, \delta \} \) by minimizing the minus penalized (log) integrated likelihood, with penalty parameter \( \lambda \), namely:

\[
- \sum_{i=0}^{N_1-1} \log \int L(Y_i, X_i | \eta_i) d\Pi(\eta_i) + \lambda \sum_{l \in \{1,2\}} \sum_{k} |\gamma_{lk}| \tag{25}
\]

where \( L(\cdot) \) is the rescaled likelihood, defined in next session.

2. Given the posterior of the \( \eta \)'s, evaluate the (log)-integrated likelihood, and propose a new point of \( \lambda \) by the Golden Section Method. Specifically, the (log)-integrated likelihood is,

\[
- \sum_{i=0}^{N_2-1} \log \int L(Y_i, X_i | \eta_i) d\Pi(\eta_i) \tag{26}
\]

3. Given parameter estimates of the \( \{ \beta, \gamma \} \)'s and the \( \{ \alpha_0, \alpha, \delta \} \)'s, estimate posteriors the \( \eta \)'s from both training and testing samples by methods similar to iterated laplace methods (INLA) [46].

The algorithm is completed by iterating the above steps until convergence.

8.5. Details of the Routine. Throughout this subsection, we consider the case where \( L = 2 \), i.e. we assume the existence of two latent attributes.
8.5.1. Step 1. Before we present a detailed treatment of the estimation routine, we would like to make the following notes.

1. For identification purpose, we have normalized \( \gamma_{11} = 1 \) and \( \gamma_{22} = 1 \).

2. \( \int L(Y_i, X_i | \eta_i) d\Pi(\eta_i) \) is approximated as \( \sum_{r=0}^{R-1} L(Y_i, X_i | \eta_{ir}) p(\eta_{ir}) \), where \( p(\eta_{ir}) \) is the probability density derived from step 3.

3. Although the L-1 penalty is not differentiable, it can be successively approximated by a ridge penalty. Specifically, assume the current estimation of in iterations of Quasi-Newton methods is denoted by \( \gamma^{(k)} \), then it can be shown that minimizing expression (1) is the same as minimizing \( \frac{(\gamma^{(k+1)})^2}{(\gamma^{(k)})^2} \) w.r.t. \( \gamma^{(k+1)} \).

4. Throughout this appendix, we minimize the rescaled (conditional) likelihood instead of the original likelihood for the following reason. Since the likelihood takes a product form, and each of the term in the product is less or equal than 1, the product can be small, and serious round-off errors may occur. To combat this problem, we define \( L \) to be \( L(Y_i, X_i | \eta_i) = scale \times P(Y_i | \eta_i) \prod_{k=1}^{K} (scale \times P(X_{ik} | \eta_i)) \).

To carry out the above minimization, we need to evaluate function values and gradients efficiently. Below we outline the implementation.

8.5.1.A. Evaluations of Function Values
To evaluate the function values and subsequent derivatives, first create an auxiliary 3d-matrix of dimension \( N_1 \times R \times (K + 1) \) (call it \( \text{EXP} \)), where

\[
\text{EXP}_{i,r,0} = e^{\beta_1 \eta_{1r} + \beta_2 \eta_{2r}}
\]

and

\[
\text{EXP}_{i,r,k} = e^{\gamma_{1k} \eta_{1r} + \gamma_{2k} \eta_{2r}}
\]

for \( k \in \{1, \ldots, K\} \), \( i \in \{0, \ldots, N_1 - 1\} \) and \( r \in \{0, \ldots, R - 1\} \).[26]

Next, compute the auxiliary following quantities

\[
\varepsilon_{a_0} := e^{\alpha_0}
\]

\[
\varepsilon_{a_1} := e^{\alpha_0 + \alpha^2}
\]

\[
\varepsilon_{\alpha} := e^{\alpha^2}
\]

\[
\varepsilon_{\delta} := e^{\delta^2}
\]

In order to evaluate the function value, we proceed to compute the following \( N_1 \times R \) matrix \( A \) and \( N_1 \) vector \( C \). Specifically, each \( A_{ir} \) equals \( L(Y_i, X_i | \eta_{ir}) p(\eta_{ir}) \), and is evaluated using a FOR-loop to calculate the product of following quantities.

\[
\text{scale} \times P(y_i | \eta_i) = \text{scale} \times \begin{cases} e_{a_0} & \text{if } y_i = -1 \\ \frac{\text{EXP}_{i,r,0} + e_{a_0}}{\text{EXP}_{i,r,0} + e_{a_1}} & \text{if } y_i = 0 \\ \\ \frac{\text{EXP}_{i,r,0} + e_{a_0}}{\text{EXP}_{i,r,0} + e_{a_1}} & \text{if } y_i = 1 \end{cases}
\]

[26] The reason of starting from 0 instead of 1 is to coincide with the tradition of \( C++ \) indexing.
and for $k \in \{1, \ldots, K\}$

$$scale \times P(x_{ik}|\eta_i) = scale \times \begin{cases} 
\frac{1}{e_{\delta} \times \text{EXP}_{i,r,k} + 1} & \text{if } x_{ik} = -1 \\
\frac{\text{EXP}_{i,r,k} + e_{\delta}}{\text{EXP}_{i,r,k}} & \text{if } x_{ik} = 0 \\
\frac{1}{e_{\delta} \times \text{EXP}_{i,r,k} + 1} & \text{if } x_{ik} = 1
\end{cases}$$

and

$$A_{ir} = p(\eta_{ir}) \prod_k V_k^{(i,r)}$$

Upon updating $A$, the entries of vector $C$ are

$$C_i = \sum_{r=0}^{R-1} A_{ir}$$

and finally the (log)-integrated likelihood can be obtained from the following expression

$$\sum_{i=0}^{N_1-1} \log C_i$$

8.5.1.A. Evaluations of Gradients
8.5.1.A.a. Gradient w.r.t. $\delta$

Note that $L(Y_i, X_i|\eta_{ir})$ is a product function of $\delta$. In order to evaluate the quantity, one can apply function recursions. Specifically, we follow the following steps.

First, calculate $scale \times P(Y_i|\eta_{ir})$ according to the previous routine, also remember that $e_\delta = e^{\delta}$ is already calculated. Now define a $K \times 1$ vector $D$ where $D_k' = \prod_{k=0}^{k'} scale \times P(X_{ik}|\eta_{ir})$, which can be done recursively. Now define function $F(k'), k \in \{1, \ldots, K\}$ to be the derivatives w.r.t. $\delta^2$ for the first $k'$th product, it can be proved that $F(\cdot)$ follows the following recursion,

$$F(k') := \frac{d}{d\delta^2} \prod_{k=1}^{k'} (scale \times P(X_{ik}|\eta_{ir})) = scale \times P(X_{ik'}|\eta_{ir}) F(k' - 1) + scale \times D_{k'-1} \times \frac{d}{d\delta^2} P(X_{ik'}|\eta_{ir})$$

where

$$\frac{d}{d\delta^2} P(X_{ik}|\eta_{ir}) = \begin{cases} 
- \frac{\text{EXP}_{i,r,k}/e_{\delta}}{(\text{EXP}_{i,r,k}/e_{\delta} + 1)^2} & \text{if } X_{ik} = -1 \\
\frac{\text{EXP}_{i,r,k}/e_{\delta}}{(\text{EXP}_{i,r,k}/e_{\delta} + 1)^2} + \frac{\text{EXP}_{i,r,k}/e_{\delta}}{(\text{EXP}_{i,r,k}/e_{\delta} + 1)^2} & \text{if } X_{ik} = 0 \\
\frac{\text{EXP}_{i,r,k}/e_{\delta}}{(\text{EXP}_{i,r,k}/e_{\delta} + 1)^2} & \text{if } X_{ik} = 1
\end{cases}$$

and finally the individual gradient for specific $i$ and $r$ can be evaluated as,

$$GRAD_i^\delta = \sum_{r=1}^{R} GRAD_{i,r}^\delta p(\eta_{ir}) \frac{C_i}{C_i}$$

where
\[ GRAD_{i,r}^{\delta} = 2\delta \times \text{scale} \times P(Y_i|\eta_{ir}) \times F(K) \]

8.5.1.A.b. Gradients w.r.t. \( \alpha_0, \alpha, \beta_1, \beta_2 \)

Note that all the above quantities appear only in \( P(Y_i|\eta_{ir}) \), and we have the following equation (denote \( \theta \) to be any of the four above parameter),

\[
\frac{\partial}{\partial \theta} \log \sum_{r=0}^{R-1} L(Y_i, X_i|\eta_{ir}) = \frac{\sum_{r=0}^{R-1} \frac{\partial}{\partial \theta} (\log P(Y_i|\eta_{ir})) \times A_{ir}}{C_i}
\]

Hence it suffices to work out expression for \( \frac{\partial}{\partial \theta} (\log P(Y_i|\eta_{ir})) \)

By straight forward computation, we have

when \( y_i = -1 \) calculate \( a_{i,r}^{\text{temp}} = \frac{\text{EXP}_{i,r,0}}{e_{\alpha_0} + \text{EXP}_{i,r,0}} \)

\[
\frac{\partial}{\partial \alpha_0} (\cdot) = a_{i,r}^{\text{temp}}
\]

\[
\frac{\partial}{\partial \alpha} (\cdot) = 0
\]

\[
\frac{\partial}{\partial \beta_1} (\cdot) = -\eta_{i,r,1} a_{i,r}^{\text{temp}}
\]

\[
\frac{\partial}{\partial \beta_2} (\cdot) = -\eta_{i,r,2} a_{i,r}^{\text{temp}}
\]

when \( y_i = 0 \), calculate \( b_{i,r}^{\text{temp}} = \frac{e_{\alpha_0} e_{\alpha_1} - \text{EXP}^2_{i,r,0}}{(e_{\alpha_0} + \text{EXP}_{i,r,0})(e_{\alpha_1} + \text{EXP}_{i,r,0})} \), then

\[
\frac{\partial}{\partial \alpha_0} (\cdot) = b_{i,r}^{\text{temp}}
\]

\[
\frac{\partial}{\partial \alpha} (\cdot) = \frac{e_{\alpha}(e_{\alpha_0} + \text{EXP}_{i,r,0})}{(e_{\alpha} - 1)(\text{EXP}_{i,r,0} + e_{\alpha_1})} \times 2\alpha
\]

\[
\frac{\partial}{\partial \beta_1} (\cdot) = \eta_{i,r,1} b_{i,r}^{\text{temp}}
\]

\[
\frac{\partial}{\partial \beta_2} (\cdot) = \eta_{i,r,2} b_{i,r}^{\text{temp}}
\]

when \( y_i = 1 \), calculate \( c_{i,r}^{\text{temp}} = \frac{e_{\alpha_1}}{\text{EXP}_{i,r,0} + e_{\alpha_1}} \), then

\[
\frac{\partial}{\partial \alpha_0} (\cdot) = -c_{i,r}^{\text{temp}}
\]

\[
\frac{\partial}{\partial \alpha_1} (\cdot) = -c_{i,r}^{\text{temp}} \times 2\alpha
\]

\[
\frac{\partial}{\partial \beta_1} (\cdot) = \eta_{i,r,1} c_{i,r}^{\text{temp}}
\]

\[
\frac{\partial}{\partial \beta_2} (\cdot) = \eta_{i,r,2} c_{i,r}^{\text{temp}}
\]

8.5.1.A.c. Gradients w.r.t. \( \gamma \)'s
Note that the gradients w.r.t. $\gamma$’s can be also obtained similarly. However, one should first note that in approximating the L-1 loss, when a certain estimate of $|\gamma|$ is smaller than a predetermined-threshold, then the gradients should be set to 0. Hence we only need to calculate the gradients of $\gamma$’s which is not 0 in the last iteration of conjugate gradient method. Furthermore, the gradients for $\gamma_{11}$ and $\gamma_{22}$ are also set to 0.

Adopting similar tricks, we obtain that individual partial derivative contribution to the likelihood is

$$
\frac{\partial}{\partial \gamma_{l,k}} \log \sum_{r=0}^{R-1} L(Y_i, X_i|\eta_{ir}) = \sum_{r=0}^{R-1} \frac{\partial}{\partial \gamma_{l,k}} (\log P(X_{ik}|\eta_{ir})) \times A_{ir}/C_i
$$

and now we have

When $X_{ik} = -1$ calculate $d_{i,r}^{\text{temp}} = \frac{e^\delta \text{EXP}_{i,r,k}}{\text{EXP}_{i,r,k} e^\delta + 1}$, and then

$$
\frac{\partial}{\partial \gamma_{1,k}} (\cdot) = -\eta_{i,r,1} d_{i,r}^{\text{temp}}
$$

$$
\frac{\partial}{\partial \gamma_{2,k}} (\cdot) = -\eta_{i,r,2} d_{i,r}^{\text{temp}}
$$

When $X_{ik} = 0$ calculate $f_{i,r}^{\text{temp}} = \frac{\text{EXP}_{i,r,k}^2 - 1}{\text{EXP}_{i,r,k}^2 \text{EXP}_{i,r,k} e^\delta + 1}$, and then

$$
\frac{\partial}{\partial \gamma_{1,k}} (\cdot) = \eta_{i,r,1} f_{i,r}^{\text{temp}}
$$

$$
\frac{\partial}{\partial \gamma_{2,k}} (\cdot) = \eta_{i,r,2} f_{i,r}^{\text{temp}}
$$

Lastly, when $X_{ik} = 1$ calculate $g_{i,r}^{\text{temp}} = \frac{e^\delta}{e^\delta + \text{EXP}_{i,r,k}}$, and then

$$
\frac{\partial}{\partial \gamma_{1,k}} (\cdot) = \eta_{i,r,1} g_{i,r}^{\text{temp}}
$$

$$
\frac{\partial}{\partial \gamma_{2,k}} (\cdot) = \eta_{i,r,2} g_{i,r}^{\text{temp}}
$$

8.5.2. Step 2. We omit the details for implementation of step 2 due to similarity to step 1.

8.5.3. Step 3. In this section, we outline the estimation routine for the evaluation of latent evaluations. Throughout this section, we assume the parameters $\{\alpha’s, \delta, \beta’s, \gamma’s\}$ are given, while $\{\eta_{1i}, \eta_{2i}\}$’s are unknown. Furthermore, we also assume $\eta_{1i} = \Psi(\xi_{1i})$ and $\eta_{2i} = \Psi(\xi_{2i})$, and $\Psi(z) = 1 - \frac{2}{1+e^z}$. Finally, note that given the parameter estimate, hence we omit the subscript $i$ in the remainder of this section.

Note that the latent evaluations are computed based on a fixed number of grid, which we chose to be 121. Although one can use a fixed grid for each iteration, the accuracy might be not ideal. Instead, we follow the steps as in Iterated Laplace Procedure (INLA), where we first match the mode and the variance with a Gaussian distribution using methods similar to Quasi-Newton. Assume the corresponding mean is $\mu_i$ and variance $\nu_i$, then we layout a fixed grid centered at $\mu_i$ and spread as far as 1.96*\nu_i. After this the
latent evaluations can be easily obtained by a straightforward application of Bayes Formula.

In the next few paragraphs, we outline the details of the matching part. **8.5.3.A. Details** Specifically, the following step is proposed by [46] to match the mode and the variance. Let us give ξ’s a prior of joint normal, with precision matrix

\[ Q = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix} \]

and define

\[ g(\xi) = \log P(Y, X | \xi) \]

then define b to be the first partial derivative, and C to be the second order derivative of \( g(\cdot) \) w.r.t. \( \xi \). Then the following iterative procedure is proposed.

Step 1: Given b and C, solve for \( \xi \) from the equation

\[ (Q - C)\xi = b \]

Step 2: Given \( \xi \), obtain new estimates of b and C based on Taylor expansion.

In this regard, we need to compute 1st and 2nd order partial derivatives of log-likelihood w.r.t. ξ’s. A direct expression is available, albeit lengthy and computationally inefficient. To facilitate our derivation, we utilize the following chain rules.

\[
\frac{\partial \log(\cdot)}{\partial \xi_l} = \frac{\partial \log(\cdot)}{\partial \eta_l} \cdot \frac{\partial \eta_l}{\partial \xi_l}
\]

\[
\frac{\partial^2 \log(\cdot)}{\partial \xi_l^2} = \frac{\partial^2 \log(\cdot)}{\partial \eta_l^2} \left( \frac{\partial \eta_l}{\partial \xi_l} \right)^2 + \frac{\partial \log(\cdot)}{\partial \eta_l} \cdot \frac{\partial^2 \eta_l}{\partial \xi_l^2}
\]

\[
\frac{\partial^2 \log(\cdot)}{\partial \xi_l \partial \xi_2} = \frac{\partial^2 \log(\cdot)}{\partial \eta_l \partial \eta_2} \cdot \frac{\partial \eta_l}{\partial \xi_1} \cdot \frac{\partial \eta_2}{\partial \xi_2}
\]

where \( l = 1, 2 \).

Below is the detailed step for computing the partial derivatives of \( \log P(Y, X | \xi) \) w.r.t. ξ’s.

First (before any loop) calculate

\[ e_{a0} := e^{a_0} \]

\[ e_{a1} := e^{a_1} \]

\[ e_{\delta} := e^{\delta^2} \]

Next for each \( i \) and each iteration, calculate

\[ e_{\xi_i} := e^{\xi_i}, e_{\xi_2} := e^{\xi_2} \]

\[ \eta_1 = 1 - \frac{2}{1 + e_{\xi_1}}, \eta_2 = 1 - \frac{2}{1 + e_{\xi_2}} \]
\[ \exp = e^{\beta_1 \eta_1 + \beta_2 \eta_2} \]

\[ \exp^{(k)} = e^{\gamma_1 \eta_1 + \gamma_2 \eta_2} \]

\[ p_{\xi_1} = \frac{e^{\xi_1}}{(e^{\xi_1} + 1)^2}, p_{\xi_2} = \frac{e^{\xi_2}}{(e^{\xi_2} + 1)^2} \]

\[ p_{\xi_1}^{(2)} = -\frac{e^{\xi_1} (e^{\xi_1} - 1)}{(e^{\xi_1} + 1)^3}, p_{\xi_2}^{(2)} = -\frac{e^{\xi_2} (e^{\xi_2} - 1)}{(e^{\xi_2} + 1)^3} \]

We first calculate the partial derivatives of \( y \)-term when \( y = -1 \)

\[ a_{\text{temp}} := \frac{\exp}{e_{\alpha_0} + \exp} \]

\[ b_{\text{temp}} := \frac{e_{\alpha_0} a_{\text{temp}}}{e_{\alpha_0} + \exp} \]

\[ \frac{\partial}{\partial \xi_1} (\cdot) = -\beta_1 a_{\text{temp}} p_{\xi_1} \]

\[ \frac{\partial}{\partial \xi_2} (\cdot) = -\beta_2 a_{\text{temp}} p_{\xi_2} \]

\[ \frac{\partial^2 (\cdot)}{\partial \xi_1^2} = -\beta_1^2 b_{\text{temp}} p_{\xi_1}^2 - \beta_1 a_{\text{temp}} p_{\xi_1}^{(2)} \]

\[ \frac{\partial^2 (\cdot)}{\partial \xi_2^2} = -\beta_2^2 b_{\text{temp}} p_{\xi_2}^2 - \beta_2 a_{\text{temp}} p_{\xi_2}^{(2)} \]

\[ \frac{\partial^2 (\cdot)}{\partial \xi_1 \partial \xi_2} = -\beta_1 \beta_2 b_{\text{temp}} p_{\xi_1} p_{\xi_2} \]

when \( y = 0 \)

\[ c_{\text{temp}} := \frac{e_{\alpha_0} e_{\alpha_1} - \exp^2}{(e_{\alpha_0} + \exp)(e_{\alpha_1} + \exp)} \]

\[ d_{\text{temp}} := -\frac{(e_{\alpha_0} e_{\alpha_1})^2 + e_{\alpha_1} e_{\alpha_0}^2 + (e_{\alpha_0} + e_{\alpha_1}) \exp^2 + 4 e_{\alpha_0} e_{\alpha_1} \exp \exp}{(e_{\alpha_0} + \exp)^2(e_{\alpha_1} + \exp)^2} \]

\[ \frac{\partial}{\partial \xi_1} (\cdot) = \beta_1 c_{\text{temp}} p_{\xi_1} \]

\[ \frac{\partial}{\partial \xi_2} (\cdot) = \beta_2 c_{\text{temp}} p_{\xi_2} \]

\[ \frac{\partial^2 (\cdot)}{\partial \xi_1^2} = \beta_1^2 d_{\text{temp}} p_{\xi_1}^2 \]

\[ \frac{\partial^2 (\cdot)}{\partial \xi_2^2} = \beta_2^2 d_{\text{temp}} p_{\xi_2}^2 \]

\[ \frac{\partial^2 (\cdot)}{\partial \xi_1 \partial \xi_2} = \beta_1 \beta_2 d_{\text{temp}} p_{\xi_1} p_{\xi_2} \]
\[
\frac{\partial^2 (\cdot)}{\partial \xi_1 \partial \xi_2} = \beta_1 \beta_2 t^\text{temp} p_{\xi_1} p_{\xi_2}
\]

Finally, when \( y = 1 \)

\[
 f^\text{temp} := \frac{e_{\alpha_1}}{e_{\alpha_1} + \text{EXP}}
\]
\[
 g^\text{temp} := -\frac{\text{EXP} e_{\alpha_1}}{(\text{EXP} + e_{\alpha_1})^2}
\]
\[
 \frac{\partial}{\partial \xi_1} (\cdot) = \beta_1 t^\text{temp} p_{\xi_1}
\]
\[
 \frac{\partial}{\partial \xi_2} (\cdot) = \beta_2 t^\text{temp} p_{\xi_2}
\]
\[
 \frac{\partial^2 (\cdot)}{\partial \xi_1^2} = \beta_1^2 t^\text{temp} p_{\xi_1}^2 + \beta_1 f^\text{temp} p_{\xi_1}^{(2)}
\]
\[
 \frac{\partial^2 (\cdot)}{\partial \xi_2^2} = \beta_2^2 t^\text{temp} p_{\xi_2}^2 + \beta_2 f^\text{temp} p_{\xi_2}^{(2)}
\]
\[
 \frac{\partial^2 (\cdot)}{\partial \xi_1 \partial \xi_2} = \beta_1 \beta_2 t^\text{temp} p_{\xi_1} p_{\xi_2}
\]

Next we proceed to calculate the partial derivatives of \( X_k \) term. When \( X_k = -1 \)

\[
 a^\text{temp} := -\frac{e^{\delta} \text{EXP}^{(k)}}{e^{\delta} \text{EXP}^{(k)} + 1}
\]
\[
 b^\text{temp} := \frac{a^\text{temp}}{e^{\delta} \text{EXP}^{(k)} + 1}
\]
\[
 \frac{\partial}{\partial \xi_1} (\cdot) = \gamma_{1k} a^\text{temp} p_{\xi_1}
\]
\[
 \frac{\partial}{\partial \xi_2} (\cdot) = \gamma_{2k} a^\text{temp} p_{\xi_2}
\]
\[
 \frac{\partial^2 (\cdot)}{\partial \xi_1^2} = \gamma_{1k}^2 b^\text{temp} p_{\xi_1}^2 + \gamma_{1k} a^\text{temp} p_{\xi_1}^{(2)}
\]
\[
 \frac{\partial^2 (\cdot)}{\partial \xi_2^2} = \gamma_{2k}^2 b^\text{temp} p_{\xi_2}^2 + \gamma_{2k} a^\text{temp} p_{\xi_2}^{(2)}
\]
\[
 \frac{\partial^2 (\cdot)}{\partial \xi_1 \partial \xi_2} = \gamma_{1k} \gamma_{2k} b^\text{temp} p_{\xi_1} p_{\xi_2}
\]

When \( X_k = 0 \)

\[
 c^\text{temp} := -\frac{(\text{EXP}^2 - 1) * e^{\delta}}{e^{\delta} (1 + \text{EXP}^2 + e^{\delta} \text{EXP}) + \text{EXP}}
\]
\[
 d^\text{temp} := -\frac{(e^{2\delta} + \text{EXP}^2 + 4e^{\delta} \text{EXP} + (e^{\delta} \text{EXP})^2 + 1)e^{\delta} \text{EXP}}{(e^{\delta} + \text{EXP})^2 e^{\delta} \text{EXP} + 1)^2}
\]
\[
\frac{\partial}{\partial \xi_1}(\cdot) = \gamma_1 c_{\text{temp}} p_{\xi_1}
\]
\[
\frac{\partial}{\partial \xi_2}(\cdot) = \gamma_2 c_{\text{temp}} p_{\xi_2}
\]
\[
\frac{\partial^2}{\partial \xi_1^2}(\cdot) = \gamma_1^2 d_{\text{temp}} p_{\xi_1}^2 + \gamma_1 c_{\text{temp}} p_{\xi_1}^{(2)}
\]
\[
\frac{\partial^2}{\partial \xi_2^2}(\cdot) = \gamma_2^2 d_{\text{temp}} p_{\xi_2}^2 + \gamma_2 c_{\text{temp}} p_{\xi_2}^{(2)}
\]
\[
\frac{\partial^2}{\partial \xi_1 \partial \xi_2}(\cdot) = \gamma_1 \gamma_2 d_{\text{temp}} p_{\xi_1} p_{\xi_2}
\]

When \(X_k = 1\)

\[
f_{\text{temp}} := \frac{e_\delta}{\text{EXP}(k) + e_\delta}
\]
\[
g_{\text{temp}} := -\frac{\text{EXP}(k) e_\delta}{(\text{EXP}(k) + e_\delta)^2}
\]
\[
\frac{\partial}{\partial \xi_1}(\cdot) = \gamma_1 f_{\text{temp}} p_{\xi_1}
\]
\[
\frac{\partial}{\partial \xi_2}(\cdot) = \gamma_2 f_{\text{temp}} p_{\xi_2}
\]
\[
\frac{\partial^2}{\partial \xi_1^2}(\cdot) = \gamma_1^2 g_{\text{temp}} p_{\xi_1}^2 + \gamma_1 f_{\text{temp}} p_{\xi_1}^{(2)}
\]
\[
\frac{\partial^2}{\partial \xi_2^2}(\cdot) = \gamma_2^2 g_{\text{temp}} p_{\xi_2}^2 + \gamma_2 f_{\text{temp}} p_{\xi_2}^{(2)}
\]
\[
\frac{\partial^2}{\partial \xi_1 \partial \xi_2}(\cdot) = \gamma_1 \gamma_2 g_{\text{temp}} p_{\xi_1} p_{\xi_2}
\]

8.6. Problems of the Routine. In this subsection, we briefly outline the problems we encounter while implementing the above routine. Heuristically, our data set is sparse in the sense that many \(x_{ik}\) take value zero. In our model, this can be achieved either by setting \(\gamma_{ik}\) to be small, or by setting \(\delta\) to be large. Consequently, the algorithm suffers from local minima and a global convergence is only observed when the initial value is set close to the truth, even if \(\eta_i\) is known. Note that this is not an identification issue, the function value at these local minimums is still far larger than that of the global minimum.
9. Appendix 2: Parallel Computing with C++11 Thread Library

9.1. Introduction. In this section, we present a short introduction to parallel computing with C++ thread library. This library is a generic part of the C++11 standards, which allows shared-memory based concurrent programming. The main goal of this introduction is to explain the logic structure of how concurrent algorithms in our research are implemented, without involving language features of C++. More specifically, we present a short analysis of the advances of C++ thread library compared to the classical Message Passing Interface (MPI), as well as some implementation obstacles in our model. In the end of this introduction, the results of a short performance test are presented.

9.2. Introduction to Parallel Programming. In general, parallel algorithms, or concurrent algorithms, are defined as the computations performed (partly) independently and (logically) simultaneous by different processors. As an illustration, consider evaluating the following sum:

\[ \sum_{i=1}^{N} a_i \]

Without further instructions, the above quantity is computed sequentially, even in a computer with multiple processors. For example, the computer will first compute \( a_1 + a_2 \), save the value in a temporary variable \( a \), then computed \( a + a_2 \) etc. On the other hand, a concurrent algorithm with two processors might take following routine: while the first processor is evaluating \( \sum_{i=1}^{\lfloor N/2 \rfloor} a_i \), the second one is calculating \( \sum_{i=\lfloor N/2 \rfloor+1}^{N} a_i \). Since the evaluation of the first half of the sum does not depend on the evaluation of the second half, the performance gain is achieved by minimizing the unnecessary waiting time of the idle processors. In fact, this problem is known as perfectly/naturally parallel/concurrent [8], since the evaluations of different parts of the sums are completely independent, and the theoretical performance gain is proportional to the number of available processors.

In estimating statistical models, concurrent programming can be a powerful tool to accelerate the computations. In many statistical models, the most computational demanding part involves computing quantities related to observations. For example, when applying a Quasi-Newton scheme to estimate a likelihood-based model, one needs to evaluate the likelihoods and gradients based on current parameter estimate. These quantities are usually sums of functions of observations, and when the number of observations is large, the evaluations of these quantities can be expensive. A strategy to speed up the computation is to compute the partial sums of these quantities using different CPU’s, and then aggregate these sums. As we have argued before, the theoretical efficiency gain is proportional to the number of CPU’s. This fact makes it rather desirable to include concurrent algorithms when estimating statistical models.

It should be noted that we have not considered implementation issues when considering the performance gain of using parallel computing. In effect, while the theoretical performance gain may be substantial, inherent
overheads are also commonly present in actual design of concurrent algorithms. We briefly outline these issues with respect to two commonly used frameworks in parallel computing, the MPI standard and the C++11 Thread Library in the next sections.

9.3. **MPI and its Limitation.** One of the classical schemes to implement concurrent algorithm is MPI. During the time when MPI was developed, multi-processor computers were unavailable to the majority of researchers. For this reason, MPI is built on the distributed memory model, which means each computer in the cluster has its own memory pool. Since the memory pool of a specific computer is not directly available to others, data has to be ‘passed’ to other computers before it can be used for further computations, hence the name “message passing interface”. The framework is implemented in a variety of libraries, notably MPICH\(^{27}\) OpenMPI\(^{28}\) and has proven to be powerful in a variety of settings [30].

In statistical applications, however, one potential problem could significantly impede the performance of MPI-based concurrent algorithms. This problem lies in the fact that sample data has to be transported in MPI settings, even in a standalone multi-core computer. In fact, MPI is designed in such a way that identical source code is compilable for both stand-alone machines or clusters, and in the latter case, virtual machines with disjoint memory segment are created to mimic the behavior of a computer cluster. If the sample size is large, the cost of serializing and transferring the data can prove to be horrendous, rendering the concurrent algorithms even less efficient than the sequential algorithm. In a worse case, multiple copies of original data can consumer large space in the memory pool, making the algorithm impossible to implement.

Due to these concerns, the shared memory model is proposed. As the name suggests, the data in the shared memory pool is accessible to all the CPU’s to read from or write to, and no additional copies are needed. In this case, the message passing is unnecessary and the efficiency gain is substantial if the volume of the data is large.

Apart from computational efficiency, there is another drawback of the MPI framework. As the name suggests, the MPI is a low level framework where sending and receiving messages has to be explicitly programmed. While this makes it conceptually easier to develop concurrent algorithms since one can explicitly specify the order of the algorithm, it makes it difficult to adapt sequential codes into concurrent ones. Specifically, in order to test the correctness of the algorithm, it is usually recommended to compose a sequential version which will later be transformed into concurrent ones [8]. This transformation usually entails completely rewritten of existing routines if MPI is chosen, which is error prone and less efficient.

\(^{27}\)https://www.mpich.org/

\(^{28}\)http://www.open-mpi.org/
9.4. C++11 Thread Library. In order to overcome the aforementioned short-comings of MPI, a number of alternative frameworks have been proposed, including the C++11 thread library\(^{29}\). Compared to MPI, the C++ thread library is shared-memory based, making it ideal for standalone computation of statistical applications. Furthermore, the thread library is constructed on a high abstract level, allowing minimal changes to the existing sequential code.

However, the C++ thread library is not without problems. The first problem is the possibility of data race. To illustrate the point, consider the simple application where two threads are adding 1 to an integer \(a\). In a single thread application, the CPU needs to first read the value of \(a\) into register, perform the addition, and then write the new value (\(a = 2\)) back to the memory. When \(a\) is still “in” the register, the \(a\) in the memory pool still holds value 1. If during this time another CPU reads the value from the memory and perform the same addition, the resulting value will be 2 instead of 3 after both operations finish, as is illustrated in table 4. It should be noted that this illustration is instance specific, which means while the data race problem may occur in certain runs of the program, in others it might not occur at all. This adds to the difficulty in debugging concurrent programs.

<table>
<thead>
<tr>
<th>Time Order</th>
<th>Thread 1</th>
<th>Thread 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Read (a) into register ((a = 1))</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>Read (a) into register ((a = 1))</td>
</tr>
<tr>
<td>3</td>
<td>Perform the addition ((a = 2))</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>Perform the addition ((a = 2))</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>Write back to memory ((a = 2))</td>
</tr>
<tr>
<td>6</td>
<td>Write back to memory ((a = 2))</td>
<td></td>
</tr>
</tbody>
</table>

Table 4. An Illustration of Data Race Problem

The data race problem, which is caused by the fact that multiple processors are trying to access the same area of memory, can be only solved by a properly designed synchronization strategy. Unfortunately, synchronizing using C++ thread libraries is not less straightforward than the MPI framework. Specifically, while in MPI synchronization can be easily achieved by specifying the task upon receiving information, there is no similar mechanism in C++ thread library and one cannot directly specify the working order of existing thread. This is designed on purpose to avoid the idioms in MPI framework.

The standard model to synchronize threads is the *thread pool*, or *producer-consumer* model\(^{29}\). Heuristically, the thread pool model can be described in the following way. Assume there exists a queue on which the required work is placed, and local threads (consumers) take turns to inspect the queue. If there is a job on the queue, one of the threads will take the job and start to work on it. Otherwise if the queue is empty, the scheduler will temporarily hibernate the local threads. The synchronization is achieved by

\(^{29}\)Technically speaking, the C++11 library is a specific implementation of thread based concurrent programming idioms. We do not differentiate these differences here
a global thread (producer) which pushes jobs unto the queue in appropriate time.

The major advantage of a thread pool is the ability to reuse idle threads. Furthermore, once the thread pool is constructed, it can be easily mingled into existing sequential code. Unfortunately, writing thread pools is considered difficult even by C++ experts\(^3\) since the whole mechanism is highly abstract and is subject to a variety of difficulties due to the language feature of C++. Due to the limit of scope, we refer the interested readers to [59] for implementation details.

9.5. **Performance Test.** In this section, we present briefly the results of comparing the sequential version with the parallel version of our algorithm. The results are calculated based on 500 Monte Carlo replications, each calculated with using either the sequential version or parallel version. The data generating process is identical to our Monte Carlo study in Section 5, although with a random draw \(\lambda\) from uniform \([0.5, 2]\) and initialization with the true parameter. Furthermore, we distinguish three settings: small sample size \((N = 500)\), medium sample size \((N = 1000)\) and large sample size \((N = 1500)\). In all the settings the number of features is 50. Finally, the Monte Carlo study is performed on a 4-core computer.

The results are presented in figure 9.5. It should be noted that there is substantial performance gain in all the cases, and the magnitude of efficiency gain increases with the sample size. However, the performance gain does not reach the theoretical threshold (in our case, four times faster), the reasons of which include the intrinsic overhead in submitting the tasks via the producer-consumer model, as well as the computational cost in the non-parallel part. However, this example does demonstrate the benefit of implementing a parallel version in statistical applications.

\(^3\)In fact, the author of [3] recommends rewriting the whole library from the scratch instead of C++11 thread library
Figure 11. Performance test of Non-parallel and Parallel Algorithm
References


Shirani, A. Artificiality: the tension between internal and external validity in economic experiments. *Journal of Economic Methodology* 12, 2 (2005), 225–237.


