Leveraging Online Reviews with Dimension Reduction Techniques for Mobile App Management

Shawn Mankad[1], Shengli Hu[1], and Anandasivam Gopal[2]

[1]Cornell University

[2]University of Maryland

Abstract

Mobile apps are one of the building blocks of the mobile digital economy. A differentiating feature of mobile apps to traditional enterprise software is online reviews, which are available on app marketplaces and represent a valuable source of consumer feedback on the app. We create a supervised topic modeling approach for app developers to use mobile reviews as useful sources of quality and customer feedback, thereby complementing traditional software testing. The approach is based on a constrained matrix factorization that leverages the relationship between term frequency and a given response variable in addition to co-occurrences between terms to recover topics that are both predictive of consumer sentiment and useful for understanding the underlying textual themes. The factorization can provide guidance on a single app's performance as well as systematically compare different apps over time for benchmarking of features and consumer sentiment. We apply our approach using a dataset of over 81,000 mobile reviews over several years for two of the most reviewed online travel agent apps from the iOS and Google Play marketplaces.

1 Introduction

Mobile commerce is expected to reach $250 billion by 2020 (MobileBusinessInsights, 2016), and through the increasing prevalence of smartphones, has already started to significantly influence all forms of economic activity. Increasingly, the mobile ecosystem is gaining significant attention from enterprises that are porting many of their standardized enterprise-based software functionalities to mobile platforms (Serrano et al., 2013). The rise of tablets and smartphones, combined with the corresponding drop in PC-based traffic on the Internet (ABIresearch, 2012), suggests that most enterprises will need to consider "mobile" as an important part of their service portfolio. A central part of this move to the mobile ecosystem is, of course, the *mobile app*.

Mobile apps are software products that are typically embedded in the native operating system of the mobile device, link to various wireless telecommunication protocols for communication, and offer specific forms of services to the consumer (Wasserman, 2010; Krishnan et al., 2000). One critical issue faced by all software development teams is that of software quality (Pressman, 2005), leading to the quality of experience for the user (Kan et al., 1994). The issue of quality of experience, based on the underlying functionality provided by the mobile app, is of particular importance in the mobile context (Ickin et al., 2012), especially as service industries increase their presence in this sphere. Poor quality of experience on the mobile app can damage the underlying brand (Anthes, 2011), alienate rewards customers and increase defections to competitors for more casual users, thus reducing revenues. These issues are also faced in enterprise software development contexts, where quality and the customer experience are particularly critical. To meet these requirements, firms spend considerable time and effort in surveying customers and developing theoretical models of software quality and customer requirements before-hand (Parasuraman et al., 1988; Pressman, 2005).

In contrast to these organizational efforts to manage quality and customer requirements, however, the mobile developer has access to a significant quantity of feedback on the quality of experience from the app through the channel of *online reviews*. Online reviews provide the development team with ready and easily

accessible feedback on the quality of experience from using the app, while also influencing other potential customers' download decisions. Moreover, useful information in such reviews are often found in the text, rather than simply the overall rating for the app. Thus, an arguably easy approach to understanding user-perceived quality and satisfaction with a mobile app may be to simply manually read the related online reviews and incorporate this understanding into the app development process. However, this approach poses several challenges. First, online reviews are characterized by high volume and diversity of opinions, making it harder to parse out the truly important feedback from non-diagnostic information (Godes and Mayzlin, 2004). Second, they are driven by significant individual biases and idiosyncracies (Li and Hitt, 2008; Chen and Lurie, 2013; Chen et al., 2014). Finally, reading and absorbing all reviews associated with an app is infeasible simply due to volume, given the number of apps that are available on the marketplace, the number of reviews that are generated per app, and the rate at which new reviews are added, which is at an increasing rate (Lim et al., 2015).

Researchers at the intersection of software engineering and unstructured data analysis have developed methodologies to help the app development teams tap into this useful source of collective information to extract specific insights that may guide future development work on the app (see Bavota (2016) for a comprehensive survey). For example, Chen et al. (2014) developed a decision support tool to automatically filter and rank informative reviews that leverages topic modeling techniques, sentiment, and classification algorithms. Iacob and Harrison (2013); Panichella et al. (2015) and Maalej and Nabil (2015) use a combination of linguistic pattern matching rules and classification algorithms to classify reviews into different categories, like feature requests and problem discovery, that developers can use to filter for informative reviews. Galvis Carreño and Winbladh (2013) applied topic modeling to app store reviews to capture the underlying consumer sentiment at a given moment in time. Fu et al. (2013) perform regularized regression with word frequencies as covariates to identify terms with the strong sentiment and guide subsequent topic modeling of app reviews.

In this work we extend this literature to help understand the evolution of con-

sumer sentiment over time while benchmarking apps against their competitors by systematically incorporating time and the competitive landscape into a supervised topic modeling framework that estimates the impact of certain discussion themes on the customer experience. Our data contains online reviews from the iTunes and Google Play marketplaces for two firms at the heart of the travel ecosystem in the United States, namely Kayak and TripAdvisor. Both these apps are free, and are aimed at frequent travelers, with functionality for search, managing reservations, accessing promotions, logging into travel accounts, reviewing travel activities, and so on.

Figure 1 shows that the time-series of average star ratings for each of the apps evolves over time as new versions are released. As an illustrative example, important issues for the Kayak's managerial and development teams in 2014 (if not sooner) would be to understand *why* ratings have trended downwards and how consumer discussion compares to competing firms, so that appropriate remedial action can be taken to improve their positioning in the mobile marketplace.

The main idea behind our approach is that features can be derived from the text not only by considering the co-occurrences between terms in reviews, but also with the observed association between term usage and star ratings – the response variable of interest. Thus, by using a constrained matrix factorization approach we leverage the relationship between terms and the response variable to recover topics that are predictive of the outcome of interest in addition to being useful for understanding the underlying textual themes. The model is flexible enough to analyze multiple apps around common topics with evolving regression coefficients as new app versions are released to the public. These are important and novel extensions with respect to the topic modeling literature, since they allows managers and development teams to go beyond a static summary of the review corpus of the app to systematically compare different apps over time for benchmarking of features and consumer sentiment. By pinpointing the causes of user dissatisfaction, a manager or development team can steer future development appropriately while ensuring a match between the user experience and the appropriate development effort by the development team.

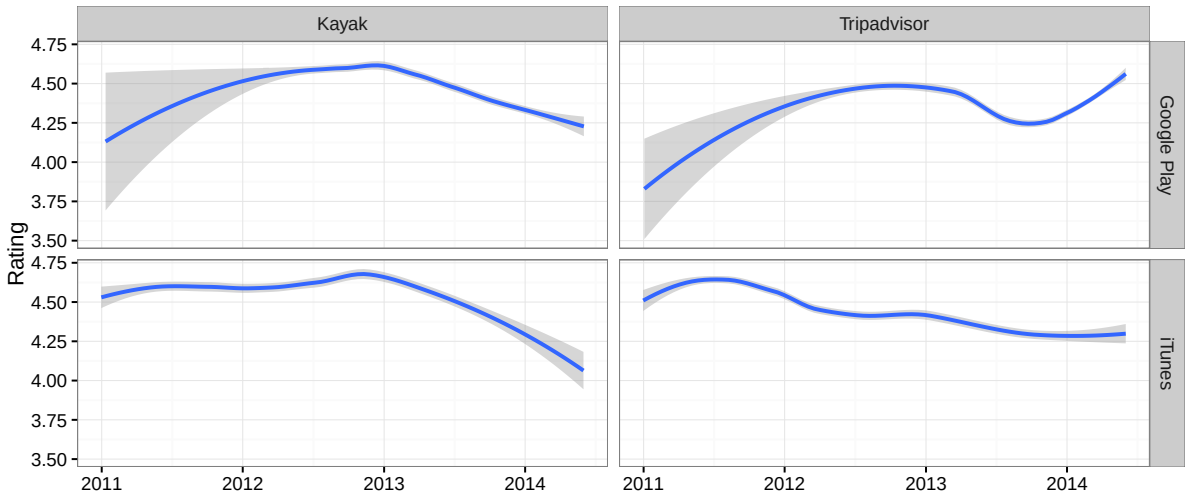

Figure 1: Average app rating over time (with Loess smoothing), by mobile app and platform. [[Update Figure so that there are no fractional values on the y-axis. The AE is pretty anal to comment on this...]]

Kayak and TripAdvisor were the two most reviewed travel apps at the time of collecting the data, which is comprised of 81,291 English reviews across a total of 140 different versions of these apps representing the full history of these apps from their introduction to the iTunes and Google Play marketplaces until November 2014. Even in this specific context, where we limit our attention to a particular industry and pair of apps, we see that there are over a 1,000 reviews per app per year, with even more reviews to be considered if the developer were interested in examining the reviews of competitor apps as well, thus underscoring the need for a statistical and semi-automated approach.

The next section presents in detail the proposed matrix factorization model and estimation framework followed by a review of competing methods in Section 3. Through a detailed simulation study under different generative models (Section 4) as well as with the iTunes and Google Play data (Section 5), we show that the proposed factorization is just as or more accurate when compared to competing methods for out of sample predictions. We also use the results of the model to characterize and contrast both apps over time. The paper concludes with a short discussion on the overall findings, the limitations of our work, and also directions for future research in Section 6.

2 Single Stage Predictions with Matrix Factorization

Prior work in the domain of text analytics and online reviews (Cao et al., 2011; Galvis Carreño and Winbladh, 2013; Tirunillai and Tellis, 2014; Abrahams et al., 2015; Mankad et al., 2016) has followed a two-stage approach, where one first derives text features through topic modeling and subsequently applies linear regression or another statistical model for prediction and inference. In principle there are many ways to perform this two-stage procedure, both in terms of generating text features and properly combining them within a statistical model. We address this issue by integrating both steps together using a matrix factorization framework. The problem we focus on is prediction and explanation of a response variable when given a set of documents. Formally, let $X \in \mathbb{R}_{+}^{n \times p}$ be a document term matrix with n documents on the rows and p terms on the columns. Let $Y \in \mathbb{R}^{n \times 1}$ be a response vector. Though in our application, $Y = \{1, 2, 3, 4, 5\}^n$ will be composed of online review scores on iTunes and Google Play, which are better modeled with an ordered multinomial distribution, we begin by solving in a novel way the case when the response variable is normally distributed and extend in Section 2.2 to the ordinal regression setting.

The objective function for the proposed factorization is

$$\begin{aligned} \min_{\Lambda,\beta} \quad & ||Y - X\Lambda\beta||_2^2 \\ \text{subject to} \quad & (\Lambda)_{ij} \geq 0 \text{ for all } i, j. \end{aligned} \tag{1}$$

The $p \times m$ non-negative matrix Λ are the term-topic loadings, the m-vector β are regression coefficients that reveal the effect of each topic on the response Y.

To enhance interpretability of the model, we require that topic loadings satisfies non-negativity constraints, which has been proposed for matrix factorization with text and other forms of data in previous works, most notably with extensions of the Nonnegative Matrix Factorization and Probalistic Latent Semantic Analysis models (Lee and Seung, 1999, 2001; Ding et al., 2008, 2010). The underlying

intuition for why non-negativity is helpful with text is given in Xu et al. (2003). Documents and terms are grouped together by their underlying topics and are also represented in the document-term matrix as data points in the positive orthant. As a result, non-negativity constraints result in a factorization that is able to better match the geometry of the data by estimating correlated vectors that identifies each group of documents and terms. We build upon this literature and impose non-negativity to better capture the natural geometry of the data. To understand the topic composition for a given document, one can inspect the corresponding row of $X\Lambda$, where larger values indicate greater topic importance to the document.

Since the regression coefficients β can take positive and negative values, the optimization problem most resembles the Semi-Nonnegative Matrix Factorizations in Ding et al. (2010), which was proposed for clustering and visualization problems, and Mankad and Michailidis (2013, 2015), who adapt the factorization for network analysis. The exact form and context of our model is to our knowledge novel, and manages to avoid the well-known issue of overfitting, which plagues other matrix factorization approaches in text analysis. Specifically, with classical techniques like the Latent Semantic Analysis (see the Section 3 for detailed review) (Deerwester et al., 1990) or Probabilistic Latent Semantic Analysis (Hofmann, 1999), one extracts topics by estimating a low-rank matrix factorization of the form $X \approx UDV^T$ subject to, respectively, the orthonormality constraints of Singular Value Decomposition or probability constraints. In both cases, the number of parameters grows linearly with the number of documents in the corpus. With the proposed factorization the number of parameters to estimate does not depend on corpus size, and grows with the size of the vocabulary and number of topics.

We note that the factorization as posed above is not fully identifiable, as the columns of Λ are subject to permutations. The arbitrary ordering of topics is a feature present in all topic modeling techniques other than Latent Semantic Analysis. Moreover, note that ΛD and $D^{-1}\beta$ is another solution with the same objective value, where D is a positive diagonal $m \times m$ matrix. We explored additional constraints on Λ and/or β to fix the scaling, but found that these approaches add complexity without noticeably improving the stability of the algorithm and quality

of the final solution. Thus, we omit further discussion here.

We also note that since the proposed method is not a formal probability model that requires term frequencies as inputs, the document-term matrix X can be preprocessed with term-frequency inverse document frequency (TFIDF) weighting (Salton and Michael, 1983)

$$(X)_{ij} = tf_{ij} \log(\frac{n}{idf_j}),$$

where tf_{ij} denotes the term frequency (word count) of term j in document i, idf_j is the number of documents containing term j, and n is the total number of documents in the corpus. This normalization has its theoretical basis in information theory and has been shown to represent the data in a way that better discriminates groups of documents and terms compared to simple word counts (Robertson, 2004).

Finally, the proposed factorization can be used to generate predictions for any new document by representing the document with the p-vector $\tilde{x}$ so that the prediction is $\hat{y} = \tilde{x}\hat{\Lambda}\hat{\beta}$.

2.1 Estimation

The estimation approach we present alternates between optimizing with respect to Λ and β. The algorithm solves for Λ using a projected gradient descent method that has been effective at balancing cost per iteration and convergence rate for similar problems posed in Nonnegative Matrix Factorization (Lin, 2007).

Starting with β, when holding Λ fixed, it is easy to verify that the remaining optimization problem is the usual regression problem leading to

$$\hat{\beta} = (\Lambda^T X^T X \Lambda)^{-1} \Lambda^T X^T Y.$$

Turning our attention to Λ, a standard gradient descent algorithm would start with an initial guess $\Lambda^{(0)}$ and constants α_i and iterate

1. For $i = 1, 2, \ldots$
2. Set $\Lambda^{(i+1)} = \Lambda^{(i)} - \alpha_i \Delta_\Lambda$,

where the gradient of the objective function with respect to Λ is

$$\Delta_\Lambda = X^T X \Lambda \beta \beta^T - X^T Y \beta^T.$$

Note that $X^T X$ and $X^T Y$ can be precomputed for faster computing time.

Due to the substraction, the non-negativity of Λ cannot be guaranteed. Thus, the basic idea of projected gradient descent is to project elements in Λ to the feasible region using the projection function, which for our problem is defined as $P(\gamma) = max(0, \gamma)$. The basic algorithm is then

1. For $i = 0, 1, 2, \ldots$
2. Set $\Lambda^{(i+1)} = P(\Lambda^{(i)} - \alpha_i \Delta_\Lambda)$.

To guarantee a sufficient decrease at each iteration and convergence to a stationary point, the "Armijo rule" developed in Bertsekas (1999, 1976) provides a sufficient condition for a given α_i at each iteration

$$||Y - X\Lambda^{(i+1)}\beta|| - ||Y - X\Lambda^{(i)}\beta|| \leq \sigma \langle \Delta_{\Lambda^{(i)}}, \Lambda^{(i+1)} - \Lambda^{(i)} \rangle, \tag{2}$$

where $\sigma \in (0, 1)$ and $\langle \cdot, \cdot \rangle$ is the sum of element wise products of two matrices. Thus, for a given α_i, one calculates $\Lambda^{(i+1)}$ and checks whether (2) is satisifed. If the condition is satisfied, then the step size α_i is appropriate to guarantee convergence to a stationary point.

The final algorithm is given in pseudocode in Algorithm 1 within Appendix A.

2.2 Extensions for the iTunes and Google Play Apps Data

In our data and generally with online review scores, $Y = \{1, 2, 3, 4, 5\}^n$, which are not well modeled with a normal distribution. To extend the proposed factorization to better fit our data, we embed the factorization within a popular ordinal regression, the cumulative odds model of McCullagh (1980), followed by a final extension that incorporates time dynamics and multiple corpora (apps).

2.2.1 Ordinal Regression with Embedded Single Stage Matrix Factorization

Keeping with the most popular form of the cumulative odds model would yield

$$logit(Y = k) = \alpha_k + X\Lambda\beta,$$

where $logit(k) = \log(P(Y \leq k)/P(Y > k))$. The key idea here is the so-called proportional odds assumption that the regression coefficients $X\Lambda\beta$ are independent of k, the rating level specified for each review. However, this assumption is not germane to our online reviews data, since the occurrence and discussion of topics can have sentiment to them (shown in Section 5) and thus are related to the overall rating of the review. Thus, we utilize a more general version of the cumulative odds model, where the regression coefficients can vary with the level of the response variable

$$logit(Y = k) = \alpha_k + X\Lambda\beta_k.$$

Before we write the likelihood, we introduce some further notation. Let Y_k be binary response vectors for categories $k = 1, \ldots, K$ created from Y. Specifically,

$$(Y_k)_j = \begin{cases} 1 & \text{if } (Y)_j = k \\ 0 & \text{otherwise} \end{cases}$$

for $j = 1, \ldots, n$ documents. Let $(X)_i$ refer to the ith row of X.

The log-likelihood for the proposed model is

$$\begin{aligned} l(\Lambda, \beta_k | Y, X) &= \sum_{i=1}^{n} \sum_{k=1}^{K-1} (Y_k)_i \log(p(k)) + (1 - \sum_{j=1}^{k} (Y_j)_i) \log(1 - p(k)) \\ &= \sum_{i=1}^{n} \sum_{k=1}^{K-1} (Y_k)_i ((X)_i \Lambda\beta_k - \log(1 + e^{(X)_i \Lambda\beta_k})) - (1 - \sum_{j=1}^{k} (Y_j)_i) \log(1 + e^{(X)_i \Lambda\beta_k}), \end{aligned} \tag{3}$$

where $p(k) = P(Y_i \leq k | (X)_i, \Lambda, \beta_k) = \frac{e^{(X)_i \Lambda\beta_k}}{1 + e^{(X)_i \Lambda\beta_k}}$. Estimating Λ and β_k by maximizing the log-likelihood can be done using a similar alternating algorithm with projected gradient descent as for the basic factorization. When holding Λ fixed,

β_k can be solved with a sequence of logistic regressions. Estimation details are provided in Appendix B.

Finally, when given a new document x, one can predict the rating by selecting the response category with largest probability

$$\begin{aligned} p(Y=1|x) &= p(1) \\ p(Y=k|x) &= p(k)-p(k-1), k=2,\ldots,K-1 \\ p(Y=K|x) &= 1-p(K-1), \end{aligned}$$

where $p(k) = P(Y \leq k|x, \Lambda, \beta_k) = \frac{e^{x\Lambda\beta_k}}{1+e^{x\Lambda\beta_k}}$.

2.2.2 Analyzing Multiple Apps Over Time

This same idea can be applied to further extend the ordinal regression to *multiple* apps over time. Analyzing multiple apps with common topics is particularly important for benchmarking exercises that aim to discover how consumer sentiment around common features changes between apps.

To analyze multiple apps simultaneously around a common set of topics over time, the model becomes

$$logit(Y_{ta}=k) = \alpha_{tak} + X_{ta}\Lambda\beta_{tak},$$

where a indexes the set of apps and t denotes time, which can be defined in multiple ways, including version releases. Note that the number of documents changes with each app and time interval, but that the vocabulary is kept constant across them so that X_{ta} is $n_{ta} \times p$, Y_{tak} are $n_{ta} \times 1$ response vectors, and β_{tak} are $m \times 1$ regression coefficients for each time interval, app, and rating category.

Such a factorization is appropriate over relatively short intervals that capture a handful of app releases. Over larger time intervals one would expect Λ to also change, which raises identifiability concerns. Nonetheless, results from this full model address an important managerial need to understand the trend of consumer sentiment for assessment of the competitive landscape as well the effectiveness of the development team. [[Can put in or reference figures from the real data analysis

here, and say that they will be discussed in detail further in Section 5.]]

The log-likelihood function becomes

$$\begin{aligned} l(\Lambda, \beta_{tak} | Y_{tak}, X_{ta}) &= \sum_{t=1}^{T} \sum_{a=1}^{A} \sum_{i=1}^{n_{ta}} \sum_{k=1}^{K-1} (Y_{tak})_i \log(p(k)) + (1 - \sum_{j=1}^{k} (Y_{taj})_i) \log(1 - p(k)) \\ &= \sum_{t=1}^{T} \sum_{a=1}^{A} \sum_{i=1}^{n_{ta}} \sum_{k=1}^{K-1} (Y_{tak})_i (X_{tai} \Lambda \beta_{tak} - \log(1 + e^{X_{tai} \Lambda \beta_{tak}})) - \\ &\quad (1 - \sum_{j=1}^{k} (Y_{taj})_i) \log(1 + e^{X_{tai} \Lambda \beta_{tak}}). \end{aligned}$$

When maximizing the log-likelihood to estimate Λ and β_{tak}, one can utilize a fusion penalty on the regression coefficients to encourage smoothness over time. This can be advantageous, especially for visualization of the results [[ref]]. However, we find that the results are interpretable without such a penalty, and hence omit it here for parsimony. Without a penalty, the estimation is again extremely similar to the previous ordinal regression model.

3 Relation with Topic Modeling Methods

As shown in Table 1, the historical roots of the proposed factorization go back to Latent Semantic Analysis (LSA), the most classical technique for topic modeling, which is based on the Singular Value Decomposition (SVD) of the document-term matrix $X \approx UDV^T$ (Deerwester et al., 1990). In many information retrieval tasks X is projected onto the word-topic factors XV^T for a low rank representation of the data. We of course are building on this idea with $X\Lambda$. With LSA the interpretability of the resultant factors can be challenging in practice, which led to the development of the Probabilistic Latent Semantic Analysis.

Probabilistic Latent Semantic Analysis (pLSA) developed in Hofmann (1999) is a formal probability model over the joint distribution of words and documents. The idea is that each word in a document is a sample drawn from a mixture of multinomial distributions that correspond to different topics. pLSA can be written in the same algebraic form of SVD but imposes probability constraints, which greatly improved the interpretation of the resultant factors. In fact, Ding et al.

(2008) shows an equivalency between the pLSA model and the Non-Negative Matrix Factorization (NMF) of the document-term matrix when one imposes sum to one constraints in addition to the non-negativity for the NMF.

While pLSA is widely seen as an improvement over LSA, there are two major drawbacks. First, the number of parameters to be estimated grows linearly with the size of the corpus, which can lead to overfitting. Second, there is no systematic way to assign probabilities to new documents after training the model. As discussed above, both of these concerns are addressed in our model.

Method	Decomposition Type	Purpose	Supervised	Incorporates Time	Multiple Corpora
Latent Semantic Analysis (Deerwester et al., 1990)	Orthonormal	Topic Modeling	No	No	No
Probablistic Latent Semantic Analysis (Hofmann, 1999)	Probabilistic	Topic Modeling	No	No	No
Latent Dirichlet Allocation (Blei et al., 2003)	Probabilistic	Topic Modeling	No	No	No
Dynamic Latent Dirichlet Allocation (Blei and Lafferty, 2006)	Probabilistic	Topic Modeling	No	Yes	No
Supervised Latent Dirichlet Allocation (Mcauliffe and Blei, 2008)	Probabilistic	Topic Modeling & Prediction	Yes	No	No
Latent Aspect Rating Analysis (Wang et al., 2010)	Probabilistic	Topic Modeling & Prediction	Yes	No	No
Multinomial Inverse Regression (Taddy, 2013)	Logistic Regression	Sentiment Analysis	Yes	No	Yes
Single Stage Matrix Factorization (Proposed Approach)	Non-negative	Topic Modeling & Prediction	Yes	Yes	Yes

Table 1: Summary and evolution of topic modeling methods.

The latent dirichlet allocation (LDA) of Blei et al. (2003) addresses these two issues with a hierarchical Bayesian generative model for how documents are constructed. LDA has been shown to work very well in practice for data exploration and unsupervised learning, and hence has been used extensively in text mining applications (Blei, 2012). We use the following LDA generating process in the next section to simulate documents in order to study how the proposed and competing methods perform in a controlled setting under various generating processes and signal-to-noise environments.

The idea is that documents are constructed in a multi-stage procedure.

1. Define K topics, which are probability distributions over words and denoted as $\beta_{1:K}$.
2. Randomly draw a distribution over topics for the entire corpus $\theta|\alpha \sim Dir(\alpha)$.
3. For each word in a document:
   (a) Randomly sample a topic according to the distribution of topics created in Step 1, i.e., $z_n \sim Mult(\theta)$.
   (b) Randomly sample a word according to the topic, i.e., $w_n|z_n, \beta_{1:K} \sim Mult(\theta)$.

The topic proportions are distributed according to a Dirichlet distribution. Topic and word assignments are conditionally distributed as multinomial. This generative process defines a joint probability distribution, where the goal is to infer the conditional distribution of the topic structure given the observed documents and word counts

$$p(\beta_{1:K}, \theta_{1:D}, z_{1:D}|w_{1:D}).$$

This task creates a key statistical challenge that has been addressed with tools like Gibbs sampling (Porteous et al., 2008) or variational algorithms (Blei and Jordan, 2006).

Titov and McDonald (2008a) develop the Multi-grain Topic Model, which improves the coherance and interpretability of the topic-keywords by adding a hierarchical topic structure. The dynamic topic model (Blei and Lafferty, 2006) is a related extension that allows the topic loadings to change over time. These works

do not consider document annotations or prediction, as in this work.

The supervised latent dirichlet allocation (sLDA) of Mcauliffe and Blei (2008) also extends the LDA framework to when documents are labeled with a random variable by adding a last stage, where the response variable is generated from each document's topic proportions.

4. For each document, draw a response variable $Y|z_{1:N}, \eta, \sigma^2 \sim N(\eta^T \bar{z}, \sigma^2)$, where the prevalence of topics determine the outcome variable.

Extensions to sLDA have been pursued for recommender systems in the contexts of scientific articles (Wang and Blei, 2011), physical products (Wu and Ester, 2015), among others. For example, Wang and Blei (2011) models users by representing them with topic preferences in addition to assuming the text corpus is generated with LDA. sLDA has also been extended to take into account additional covariates for the regression step (Agarwal and Chen, 2010).

Our approach is similar to the sLDA model, but there are important differences. Because sLDA is a formal probabilistic model, it must take as input the term frequencies and requires specification of hyperparameters. sLDA also is a static model in the sense that the regression coefficients do not evolve in over time or in different conditions. One could in principal extend the sLDA model to this setting, but there are nuanced issues around identifiability that need to be carefully considered when and if both the topic keywords and regression coefficients are changing. In contrast, because our method is not a formal probability model but a constrained factorization, we can represent each document using the term-frequency inverse document frequency (Robertson, 2004), which has been shown to be advantageous for various learning tasks. Our model does not require tuning any parameters, whereas sLDA requires careful specification of hyperparameters. In fact, numerous empirical studies show that the performance of LDA-based methods with online app reviews is sensitive to hyperparameter specification (Lu et al., 2011b; Panichella et al., 2013; Thomas et al., 2013; Bavota, 2016). Our approach is also flexible enough to allow evolving regression coefficients as new app versions are released to the public and for multiple apps to be analyzed simultaneously around the same topics. These are an important and novel extensions, since they allow managers to go beyond a static

summary to understand how the customer experience is evolving with different apps and versions.

Another similar and recent work is the multinomial inverse regression of Taddy (2013), which uses a logistic regression to extract sentiment information from document annotations and phrase counts that are modeled as draws from a multinomial distribution. The nuanced differences in context leads to different modeling decisions. Since sentiment analysis is the main objective in Taddy (2013), where recovering dictionaries is critical, the multinomial inverse regression analysis is done at the phrase or term-level. Our approach performs topic modeling (grouping of the terms) at the same time as regression.

[[Aspect based opinion mining: McAuley et al. (2012); Lu et al. (2011a); Brody and Elhadad (2010); Jo and Oh (2011); Titov and McDonald (2008b); Baccianella et al. (2009) [[LARA (Wang et al., 2010).]]

4 Simulation Study

We test the accuracy of the proposed model relative to competing methods under different settings. The first simulation establishes self-consistency of the proposed factorization, that is, responses are generated from the model implied by the factorization. The second simulation generates responses using the Supervised Latent Dirichlet Allocation model of Mcauliffe and Blei (2008). For a fair comparison, we consider the canonical setting underlying (1) with a normally distributed response and without consideration of time or multiple apps.

The methods we compare are as follows:

1. Latent semantic analysis of the document-term matrix with TFIDF weightings (denoted as LSA). Once the document term matrix has been decomposed with SVD, $X_{train} \approx UDV^T$, the singular vectors in U are used as independent variables in a regression model $Y = X_{test}V\beta + \epsilon$;

2. Probabilistic latent semantic analysis (denoted as pLSA). Similarly, we estimate $Y = X_{test}V\beta + \epsilon$, where V are the probabilistic word-topic loadings estimated from X_{train};

3. Latent Dirichlet Allocation (LDA). Similarly, we estimate $Y = X_{test}V\beta + \epsilon$, where V are the probabilistic word-topic loadings estimated from X_{train}. The Dirichlet parameters are chosen through 5-fold cross validation;

4. Supervised LDA (denoted as sLDA). The Dirichlet parameters for the Document/Topic and Topic/Term distributions are chosen through 5-fold cross validation and σ^2 is set to be the training sample variance;

5. The proposed factorization of the document-term matrix (denoted as SSMF for Single-Stage Matrix Factorization).

All analyses are performed using R (R Core Team, 2014), with the "tm" (Feinerer et al., 2008) and "topicmodels" (Grün and Hornik, 2011) libraries. For sLDA, we use the collapsed Gibbs sampler implemented in the "lda" package (Chang, 2012).

4.1 Self Consistency

Data are generated to study how proposed the model performs under its implied generating process, where $Y|X, \Lambda, \beta, \sigma^2 \sim \text{Normal}(X\Lambda\beta, \sigma^2)$. X is the document term matrix, $(\Lambda)_{ij} \sim \text{Uniform}[0, 1]$, and $(\beta)_j \sim \text{Normal}(0, 1)$. Documents are simulated using the Latent Dirichlet Process (Blei et al., 2003) with both Dirichlet parameters for Document/Topic and Topic/Term distributions set equal to 0.8. The number of terms in each document is given by a Poisson random variable with mean μ, and the number of training and test documents is set to $n = 1000$, and the size of the vocabulary $p = 2000$.

We vary the level of $\sigma = \{1, 3, 5\}$ and $\mu = \{15, 250, 2000\}$ to study how each model performs in different noise and document length levels. Similarly, the estimated number of topics is always equal to the true value and varied from 2 to 20. After training each model, we assess the accuracy of the predictions on the test set using the root mean squared error, which are shown in the top panel of Table 2. The proposed model performs best across all settings. It is notable that the proposed model performs well even when the number of words in each document is small. This is important since a distinguishing property of app mobile reviews is that an overwhelming majority are written on mobile devices, leading to shorter

and less formal writing styles (Burtch and Hong, 2014). In our real app reviews data, the average document length is under 20 words.

4.2 Supervised Latent Dirichlet Allocation

Data are generated under the generating process assumed by sLDA (Mcauliffe and Blei, 2008), where $Y|Z, \beta, \sigma^2 \sim \text{Normal}(\beta^T Z, \sigma^2)$. Z is the Document/Topic probability distribution. All other settings are identical to the previous simulation study. Table 2 shows that the sLDA method performs best in most settings, but that the gain in performance is relatively small over the proposed method especially when the documents are long. The robust performance of SSMF in both simulations with documents of varying length indicates that the proposed factorization should be useful for our app review data as well as with other corpora.

5 iTunes and Google Play App Reviews

Now that we have established using synthetic data that the proposed approach is accurate and robust to different generating processes and signal-to-noise levels, we demonstrate the method's real-life viability and applicability by using the mobile apps marketplace data for Kayak and TripAdvisor. We begin by discussing the preprocessing and model selection steps, followed by a detailed discussion of the findings.

5.1 Pre-processing

To ensure accurate word counts when forming the document term matrix, we follow the standard preprocessing steps (Boyd-Graber et al., 2014) of transforming all text into lowercase, removing words composed of less than three characters and stopwords (e.g., "a", "and", "the"), prefixing a negation flag to the word that follows (e.g., "not enjoyable" becomes "not_enjoyable"), and stemming words, which refers to the process of reducing words to their base (e.g., "enjoyable" and "enjoying" become "enjoy"). We then applied the normalization of term frequency weight in document vector space after removing infrequent terms that have occurred in less

Self-Consistency

Document Length	σ	LSA	pLSA	LDA	sLDA	SSMF
15	1	5.769	5.774	5.770	4.900	2.300
		(3.550)	(3.563)	(3.557)	(2.764)	(0.575)
15	3	6.662	6.663	6.664	5.873	4.632
		(3.350)	(3.356)	(3.364)	(2.519)	(0.652)
15	5	8.212	8.218	8.219	7.417	7.325
		(2.986)	(2.996)	(3.003)	(2.170)	(1.118)
250	1	23.904	23.770	23.751	20.189	11.727
		(15.079)	(15.019)	(14.984)	(11.672)	(4.555)
250	3	23.357	23.249	23.240	19.909	11.978
		(14.118)	(14.035)	(14.006)	(10.825)	(4.225)
250	5	26.296	26.106	26.128	22.188	13.176
		(14.961)	(14.851)	(14.865)	(11.407)	(4.129)
2000	1	64.211	63.781	63.775	55.678	34.015
		(37.507)	(37.298)	(37.308)	(29.851)	(13.022)
2000	3	68.355	67.891	67.868	58.553	33.453
		(41.674)	(41.388)	(41.453)	(33.466)	(13.018)
2000	5	65.539	65.034	65.068	56.208	33.665
		(42.247)	(41.884)	(42.003)	(32.910)	(12.546)

sLDA Generating Process

Document Length	σ	LSA	pLSA	LDA	sLDA	SSMF
15	1	3.404	3.406	3.405	3.396	4.313
		(1.329)	(1.324)	(1.327)	(1.335)	(1.536)
15	3	4.443	4.446	4.440	4.426	5.288
		(0.948)	(0.950)	(0.945)	(0.955)	(1.837)
15	5	6.057	6.059	6.063	6.045	7.064
		(0.799)	(0.800)	(0.798)	(0.808)	(1.743)
250	1	3.155	3.158	3.158	3.181	3.324
		(0.983)	(0.985)	(0.980)	(0.986)	(1.343)
250	3	4.385	4.385	4.387	4.415	4.898
		(0.849)	(0.852)	(0.847)	(0.845)	(1.843)
250	5	5.920	5.922	5.921	5.966	6.977
		(0.620)	(0.622)	(0.621)	(0.616)	(1.118)
2000	1	3.379	3.378	3.379	3.429	3.360
		(1.254)	(1.254)	(1.253)	(1.263)	(1.266)
2000	3	4.557	4.551	4.555	4.623	4.545
		(1.175)	(1.175)	(1.176)	(1.187)	(1.204)
2000	5	6.020	6.017	6.021	6.112	5.993
		(0.769)	(0.770)	(0.765)	(0.785)	(0.775)

Table 2: Root Mean Squared Error averaged over all ranks from the simulation study with standard deviations in parentheses.

than 10 reviews. The resulting total vocabulary size is 2,063 for reviews iTunes and 2,273 for reviews Google Play.

We group reviews into....

5.2 Model Selection

5.3 Results and Discussion

6 Conclusion

We presented a constrained matrix factorization model that performs topic modeling and regression in a single stage. The novelty of this approach is that the model leverages the relationship between terms and star ratings to recover topics that are predictive of the outcome of interest in addition to being useful for understanding the underlying textual themes. Comparing against the most popular supervised and unsupervised topic modeling frameworks, we find that our approach is computationally efficient, and just as or more accurate. Our results are in line with O'Callaghan et al. (2015) who showed that NMF style factorizations may lead to better solutions over LDA-based approaches especially with niche or non-mainstream corpora. We extend the factorization to provide guidance on a single app's performance as well as systematically compare different apps over time for benchmarking of features and consumer sentiment.

An important extension in terms of the application to online app reviews is recovering market structure. In our data the set of competing apps are dictated by the core business of each of the firms. Yet, in general with mobile apps the appropriate set of benchmark apps is unclear, especially from the consumer's perspective. For instance, if an app streams video even without it being a core feature, an average consumer might benchmark this aspect internally against Netflix or the Youtube app, popular apps that specialize in video playback. Thus, identifying which other apps are seen by the consumer as competitors or substitutes using online reviews would address a key challenge in understanding the consumer and in deriving value from online reviews for companies. As such, a growing number of firms have begun

developing dashboards that display summaries of online customer reviews to managers (Han et al., 2016). Our methodology is promising for such summaries that require speed and prediction accuracy.

References

ABIresearch. M-commerce growing to 24smartphone adoption. https://www.abiresearch.com/press/m-commerce-growing-to-24-of-total-e-commerce-marke/, 2012. Accessed: 2016-06-18.

Alan S Abrahams, Weiguo Fan, G Alan Wang, Zhongju John Zhang, and Jian Jiao. An integrated text analytic framework for product defect discovery. *Production and Operations Management*, 24(6):975–990, 2015.

Deepak Agarwal and Bee-Chung Chen. flda: matrix factorization through latent dirichlet allocation. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 91–100. ACM, 2010.

Gary Anthes. Invasion of the mobile apps. *Communications of the ACM*, 54(9): 16–18, 2011.

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. Multi-facet rating of product reviews. In *European Conference on Information Retrieval*, pages 461–472. Springer, 2009.

Gabriele Bavota. Mining unstructured data in software repositories: Current and future trends. In *2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER)*, volume 5, pages 1–12. IEEE, 2016.

Dimitri P Bertsekas. On the goldstein-levitin-polyak gradient projection method. *Automatic Control, IEEE Transactions on*, 21(2):174–184, 1976.

Dimitri P Bertsekas. Nonlinear programming. 1999.

David M Blei. Probabilistic topic models. *Communications of the ACM*, 55(4): 77–84, 2012.

David M. Blei and Michael I. Jordan. Variational inference for dirichlet process mixtures. *Bayesian Analysis*, 1(1):121–143, 03 2006. doi: 10.1214/06-BA104. URL http://dx.doi.org/10.1214/06-BA104.

David M Blei and John D Lafferty. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*, pages 113–120. ACM, 2006.

David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

Jordan Boyd-Graber, David Mimno, David Newman, Edoardo M Airoldi, David Blei, Elena A Erosheva, and Stephen E Fienberg. Care and feeding of topic models: Problems, diagnostics, and improvements. *Handbook of Mixed Membership Models and Their Applications*, 2014.

Samuel Brody and Noemie Elhadad. An unsupervised aspect-sentiment model for online reviews. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 804–812. Association for Computational Linguistics, 2010.

Gordon Burtch and Yili Hong. What happens when word of mouth goes mobile? *Proceedings of the International Conference on Information Systems*, 2014.

Qing Cao, Wenjing Duan, and Qiwei Gan. Exploring determinants of voting for the helpfulness of online user reviews: A text mining approach. *Decision Support Systems*, 50(2):511–521, 2011.

Jonathan Chang. *lda: Collapsed Gibbs sampling methods for topic models.*, 2012. URL http://CRAN.R-project.org/package=lda. R package version 1.3.2.

Ning Chen, Jialiu Lin, Steven CH Hoi, Xiaokui Xiao, and Boshen Zhang. Ar-miner: mining informative reviews for developers from mobile app marketplace. In *Proceedings of the 36th International Conference on Software Engineering*, pages 767–778. ACM, 2014.

Zoey Chen and Nicholas H Lurie. Temporal contiguity and negativity bias in the impact of online word of mouth. *Journal of Marketing Research*, 50(4):463–476, 2013.

Scott C. Deerwester, Susan T Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis. *JAsIs*, 41(6): 391–407, 1990.

C. Ding, Tao Li, and M.I. Jordan. Convex and semi-nonnegative matrix factorizations. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(1): 45–55, 2010. ISSN 0162-8828. doi: 10.1109/TPAMI.2008.277.

Chris Ding, Tao Li, and Wei Peng. On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing. *Comput. Stat. Data Anal.*, 52(8):3913–3927, April 2008. ISSN 0167-9473. URL http://users.cis.fiu.edu/~taoli/pub/NMFpLSIequiv.pdf.

Ingo Feinerer, Kurt Hornik, and David Meyer. Text mining infrastructure in r. *Journal of Statistical Software*, 25(5):1–54, 3 2008. ISSN 1548-7660. URL http://www.jstatsoft.org/v25/i05.

Bin Fu, Jialiu Lin, Lei Li, Christos Faloutsos, Jason Hong, and Norman Sadeh. Why people hate your app: Making sense of user feedback in a mobile app store. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1276–1284. ACM, 2013.

Laura V Galvis Carreño and Kristina Winbladh. Analysis of user comments: an approach for software requirements evolution. In *Proceedings of the 2013 International Conference on Software Engineering*, pages 582–591. IEEE Press, 2013.

David Godes and Dina Mayzlin. Using online conversations to study word-of-mouth communication. *Marketing science*, 23(4):545–560, 2004.

Bettina Grün and Kurt Hornik. topicmodels: An R package for fitting topic models. *Journal of Statistical Software*, 40(13):1–30, 2011. URL http://www.jstatsoft.org/v40/i13/.

Hyun Jeong Han, Shawn Mankad, Srinagesh Gavirneni, and Rohit Verma. What guests really think of your hotel: Text analytics of online customer reviews. *Cornell Hospitality Report*, 2016.

Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57. ACM, 1999.

Claudia Iacob and Rob Harrison. Retrieving and analyzing mobile apps feature requests from online reviews. In *Mining Software Repositories (MSR), 2013 10th IEEE Working Conference on*, pages 41–44. IEEE, 2013.

Selim Ickin, Katarzyna Wac, Markus Fiedler, Lucjan Janowski, Jin-Hyuk Hong, and Anind K Dey. Factors influencing quality of experience of commonly used mobile applications. *Communications Magazine, IEEE*, 50(4):48–56, 2012.

Yohan Jo and Alice H Oh. Aspect and sentiment unification model for online review analysis. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 815–824. ACM, 2011.

SH Kan, Victor R. Basili, and Larry N Shapiro. Software quality: an overview from the perspective of total quality management. *IBM Systems Journal*, 33(1):4–19, 1994.

Mayuram S Krishnan, Charlie H Kriebel, Sunder Kekre, and Tridas Mukhopadhyay. An empirical analysis of productivity and quality in software products. *Management science*, 46(6):745–759, 2000.

D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. *Advances in neural information processing systems*, pages 556–562, 2001. URL hebb.mit.edu/people/seung/papers/nmfconverge.pdf.

Daniel D. Lee and H. Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 10 1999.

Xinxin Li and Lorin M Hitt. Self-selection and information role of online product reviews. *Information Systems Research*, 19(4):456–474, 2008.

Soo Ling Lim, Peter J Bentley, Natalie Kanakam, Fuyuki Ishikawa, and Shinichi Honiden. Investigating country differences in mobile app user behavior and challenges for software engineering. *Software Engineering, IEEE Transactions on*, 41 (1):40–64, 2015.

Chuan-bi Lin. Projected gradient methods for nonnegative matrix factorization. *Neural computation*, 19(10):2756–2779, 2007.

Bin Lu, Myle Ott, Claire Cardie, and Benjamin K Tsou. Multi-aspect sentiment analysis with topic models. In *Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on*, pages 81–88. IEEE, 2011a.

Yue Lu, Qiaozhu Mei, and ChengXiang Zhai. Investigating task performance of probabilistic topic models: an empirical study of plsa and lda. *Information Retrieval*, 14(2):178–203, 2011b. ISSN 1573-7659. doi: 10.1007/s10791-010-9141-9. URL http://dx.doi.org/10.1007/s10791-010-9141-9.

Walid Maalej and Hadeer Nabil. Bug report, feature request, or simply praise? on automatically classifying app reviews. In *Requirements Engineering Conference (RE), 2015 IEEE 23rd International*, pages 116–125. IEEE, 2015.

S. Mankad and G. Michailidis. Discovery of path-important nodes using structured semi-nonnegative matrix factorization. In *Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2013 IEEE 5th International Workshop on*, pages 288–291, Dec 2013. doi: 10.1109/CAMSAP.2013.6714064.

Shawn Mankad and George Michailidis. Analysis of multiview legislative network with structured matrix factorization: Does twitter influence translate to the real world? *Annals of Applied Statistics*, 2015.

Shawn Mankad, Hyun Jeong Han, Joel Goh, and Srinagesh Gavirneni. Understanding online hotel reviews through automated text analysis. *Service Science*, 8(2):124–138, 2016. doi: 10.1287/serv.2016.0126. URL http://dx.doi.org/10.1287/serv.2016.0126.

Julian McAuley, Jure Leskovec, and Dan Jurafsky. Learning attitudes and attributes from multi-aspect reviews. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, pages 1020–1025. IEEE, 2012.

Jon D Mcauliffe and David M Blei. Supervised topic models. In *Advances in neural information processing systems*, pages 121–128, 2008.

Peter McCullagh. Regression models for ordinal data. *Journal of the royal statistical society. Series B (Methodological)*, pages 109–142, 1980.

MobileBusinessInsights. Mobile commerce trends: Retail in 2017, 2018 and beyond. http://mobilebusinessinsights.com/2016/12/mobile-commerce-trends-retail-in-2017-2018-and-beyond/, 2016. Accessed: 2017-04-05.

Derek O'Callaghan, Derek Greene, Joe Carthy, and Pádraig Cunningham. An analysis of the coherence of descriptors in topic modeling. *Expert Systems with Applications*, 42(13):5645–5657, 2015.

Annibale Panichella, Bogdan Dit, Rocco Oliveto, Massimiliano Di Penta, Denys Poshyvanyk, and Andrea De Lucia. How to effectively use topic models for software engineering tasks? an approach based on genetic algorithms. In *Proceedings of the 2013 International Conference on Software Engineering*, ICSE '13, pages 522–531, Piscataway, NJ, USA, 2013. IEEE Press. ISBN 978-1-4673-3076-3. URL http://dl.acm.org/citation.cfm?id=2486788.2486857.

Sebastiano Panichella, Andrea Di Sorbo, Emitza Guzman, Corrado A Visaggio, Gerardo Canfora, and Harald C Gall. How can i improve my app? classifying user reviews for software maintenance and evolution. In *Software Maintenance and Evolution (ICSME), 2015 IEEE International Conference on*, pages 281–290. IEEE, 2015.

A. Parasuraman, Valarie A Zeithaml, and Leonard L Berry. Servqual. *Journal of Retailing*, 64(1):12–40, 1988.

Ian Porteous, David Newman, Alexander Ihler, Arthur Asuncion, Padhraic Smyth, and Max Welling. Fast collapsed gibbs sampling for latent dirichlet allocation. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 569–577, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-193-4. doi: 10.1145/1401890.1401960. URL http://doi.acm.org/10.1145/1401890.1401960.

Roger S Pressman. *Software engineering: a practitioner's approach*. Palgrave Macmillan, 2005.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2014. URL http://www.R-project.org/.

Stephen Robertson. Understanding inverse document frequency: on theoretical arguments for idf. *Journal of documentation*, 60(5):503–520, 2004.

Gerard Salton and J Michael. Mcgill. *Introduction to modern information retrieval*, pages 24–51, 1983.

Nicolas Serrano, Josune Hernantes, and Gorka Gallardo. Mobile web apps. *Software, IEEE*, 30(5):22–27, 2013.

Matt Taddy. Multinomial inverse regression for text analysis. *Journal of the American Statistical Association*, 108(503):755–770, 2013.

S. W. Thomas, M. Nagappan, D. Blostein, and A. E. Hassan. The impact of classifier configuration and classifier combination on bug localization. *IEEE Transactions on Software Engineering*, 39(10):1427–1443, Oct 2013. ISSN 0098-5589. doi: 10.1109/TSE.2013.27.

Seshadri Tirunillai and Gerard J Tellis. Mining marketing meaning from online chatter: Strategic brand analysis of big data using latent dirichlet allocation. *Journal of Marketing Research*, 51(4):463–479, 2014.

Ivan Titov and Ryan McDonald. Modeling online reviews with multi-grain topic models. In *Proceedings of the 17th international conference on World Wide Web*, pages 111–120. ACM, 2008a.

Ivan Titov and Ryan T McDonald. A joint model of text and aspect ratings for sentiment summarization. In *ACL*, volume 8, pages 308–316. Citeseer, 2008b.

Chong Wang and David M Blei. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 448–456. ACM, 2011.

Hongning Wang, Yue Lu, and Chengxiang Zhai. Latent aspect rating analysis on review text data: a rating regression approach. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 783–792. ACM, 2010.

Anthony I Wasserman. Software engineering issues for mobile application development. In *Proceedings of the FSE/SDP workshop on Future of software engineering research*, pages 397–400. ACM, 2010.

Yao Wu and Martin Ester. Flame: A probabilistic model combining aspect based opinion mining and collaborative filtering. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, pages 199–208. ACM, 2015.

Wei Xu, Xin Liu, and Yihong Gong. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, SIGIR '03, pages 267–273, New York, NY, USA, 2003. ACM. ISBN 1-58113-646-3. doi: 10.1145/860435.860485. URL http://doi.acm.org/10.1145/860435.860485.

A Algorithm for the Single Stage Matrix Factorization with Normal Responses

The final algorithm is given in Algorithm 1. Searching for an appropriate α_i when updating Λ is the most time-consuming task. To improve runtime, we utilize the heuristic of using α_{i-1} as an initial guess for α_i, and set $\sigma = 0.01$ and $\gamma = 0.9$.

Algorithm 1 The Alternating Least Squares Algorithm with projected gradient descent for normally distributed Y, where the superscript (i) denotes the iteration number.

Set $i = 0$
Initialize $(\beta)_j^{(i)} \sim N(0, 1)$ for all j
Initialize $\alpha_i = 1, \gamma = 0.9$
while $\delta \geq \epsilon$ **and** $i \leq$ max iterations **do**
  $\alpha_{i+1} = \alpha_i$
  if α_{i+1} satisfies (2) **then**
    repeat
      $\alpha_{i+1} = \frac{\alpha_{i+1}}{\gamma}$
    until α_{i+1} does not satisfies (2)
  else
    repeat
      $\alpha_{i+1} = \alpha_{i+1}\gamma$
    until α_{i+1} satisfies (2)
  end if
  Set $\Lambda^{(i+1)} = P(\Lambda^{(i)} - \alpha_{i+1}(X^T X \Lambda \beta \beta^T - X^T Y \beta^T))$
  Set $\tilde{X} = X\Lambda^{(i+1)}$.
  Set for $\beta^{(i+1)} = (\tilde{X}^T \tilde{X})^{-1} \tilde{X}^T Y$.
  Set $\delta = \frac{||Y - X\Lambda^{(i+1)}\beta^{(i+1)}||_2^2 - ||Y - X\Lambda^{(i)}\beta^{(i)}||_2^2}{||Y - X\Lambda^{(i)}\beta^{(i)}||_2^2}$
  Set $i = i + 1$
end while

[[Put figure of the objective function over iterations for all 3 models. The fact that it monotonically decreases is awesome!]]

B Estimation of the Ordinal Regression with Embedded Single Stage Matrix Factorization

As with the main factorization, the estimation approach we present alternates between optimizing the log-likelihood (3) with respect to Λ and β_k. In fact, the overall algorithm has the same form as Algorithm 1 with updates to how to solve for β_k,

the gradient of Λ, and the step size selection condition.

When holding Λ fixed, β_k can be solved with standard logistic regressions. When solving for the regression coefficients for the kth category, β_k, the response is coded is 1 if the review rating is less than or equal to k.

When solving for Λ, holding β fixed, we again utilize the projected gradient descent algorithm with appropriate updates for the gradient of Λ and the Armijo rule shown below.

The gradient of the log-likelihood with respect to Λ is

$$\Delta_\Lambda = \frac{\partial l}{\partial \Lambda} = \sum_{i=1}^{n} \sum_{m=1}^{K-1} (Y_m)_i \frac{1}{1 + e^{(X)_i \Lambda \beta_m}} (X)_i^T \beta_m^T + (1 - \sum_{j=1}^{m} (Y_j)_i) \frac{-e^{(X)_i \Lambda \beta_m}}{1 + e^{(X)_i \Lambda \beta_m}} (X)_i^T \beta_m^T.$$

To guarantee a sufficient decrease at each iteration and convergence to a stationary point, the Armijo rule is used to select appropriate α_i at each iteration

$$l(\Lambda^{(i+1)}, \beta | Y, X) - l(\Lambda^{(i)}, \beta | Y, X) \leq \sigma \langle \Delta_{\Lambda^{(i)}}, \Lambda^{(i+1)} - \Lambda^{(i)} \rangle,$$

where $\sigma \in (0, 1)$ and $\langle \cdot, \cdot \rangle$ is the sum of element wise products of two matrices.