Submitted to the Annals of Applied Statistics
arXiv: arXiv:1607.07515

SINGLE STAGE PREDICTION WITH EMBEDDED TOPIC MODELING OF ONLINE REVIEWS FOR MOBILE APP MANAGEMENT

By Shawn Mankad*, Shengli Hu*, and Anandasivam Gopal†

*Cornell University** *and University of Maryland*†

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 is combined with ordinal regression to provide guidance from online reviews 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 100,000 mobile reviews over several years for three of the most popular online travel agent apps from the iTunes 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, Hernantes and Gallardo, 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, Basili and Shapiro, 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, mature software firms spend considerable time and effort in surveying customers and

AMS 2000 subject classifications: Primary 62P25; secondary 62H99
Keywords and phrases: mobile apps, online reviews, text analysis, topic modeling, matrix factorization

developing theoretical models of software quality and customer requirements before-hand (Parasuraman, Zeithaml and Berry, 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 readily 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 idiosyncrasies, thereby making it risky to base quality improvement initiatives on single reviews or reviewers (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, topic modeling, 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. Similarly, Fu *et al.* (2013) perform regularized regression with word frequencies as covariates to identify terms with strong sentiment that guide subsequent topic modeling of app reviews. The authors aggregate their findings over time to gain insight into a single app as well as all apps in the market.

This work extends this literature to help understand the evolution of consumer sentiment over time while benchmarking apps against their competitors by systematically incorporating time effects 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 three firms at the heart of the travel ecosystem in the United States, namely Expedia, Kayak, and TripAdvisor. All three of these firms provide apps that 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 Expedia's managerial and development teams heading into 2013 (if not sooner) would be to understand *why* ratings have trended downwards on the iTunes platform and how consumer discussion compares to competing firms, so that appropriate remedial action can be taken to improve their positioning in the mobile marketplace.

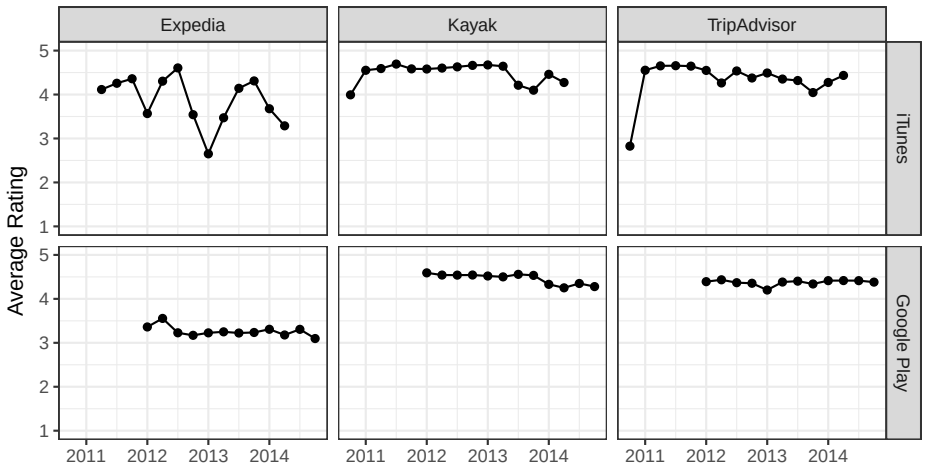

Fig 1: Average app rating over each year-quarter, by mobile app and platform.

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 embedded within an ordinal regression model, 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 associated with an 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 effort appropriately while ensuring a match between the user experience and the appropriate development effort by the development team.

Expedia, Kayak, and TripAdvisor were three of the most reviewed travel apps at the time of collecting the data, which is comprised of 104,816 English reviews across a total of 162 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 trio 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 models 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 model performs favorably when compared to competing methods for out of sample predictions and topic interpretability. We also use the results of the model to characterize and contrast the apps over time. The paper concludes with a short discussion on the overall findings, the limitations of our work, and 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, Duan and Gan, 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 for apps 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 satisfy 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 Probabilistic Latent Semantic Analysis models (Lee and Seung, 1999, 2001; Ding, Li and Peng, 2008; Ding, Li and Jordan, 2010). The underlying intuition for why non-negativity is helpful with text is given in Xu, Liu and Gong (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 identify 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, Li and Jordan (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 Latent Semantic Analysis (see 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 $D^{-1}\beta$, where D is a positive diagonal $m \times m$ matrix, is another solution with the same objective value. We explored additional constraints on Λ and/or β to fix the scaling, but found that these approaches add complexity to the estimation without noticeably improving the quality of the final solution. Thus, we omit further discussion of these approaches here.

We also note that since the proposed method does not estimate a formal probability model for the topic structure, the document-term matrix X can be preprocessed with term-frequency inverse document frequency (TFIDF) weighting (Salton and Michael, 1983)

$$(X)_{ij} = \mathrm{TF}_{ij} \log(\frac{n}{\mathrm{IDF}_j}),$$

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

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

Driven by our upcoming extension to the real data and results therein, we do not to regularize β, though it can be advantageous and easily done in other data contexts.

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

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

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

$$\Delta_\Lambda = X^T X \Lambda \beta \beta^T - X^T Y \beta^T. \tag{2}$$

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

Due to the subtraction, 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 $\Lambda^{(i+1)} = P(\Lambda^{(i)} - \gamma_i \Delta_\Lambda)$.

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

$$||Y - X\Lambda^{(i+1)}\beta|| - ||Y - X\Lambda^{(i)}\beta|| \leq \sigma \langle \Delta_{\Lambda^{(i)}}, \Lambda^{(i+1)} - \Lambda^{(i)} \rangle, \tag{3}$$

where $\sigma \in (0,1)$ and $\langle \cdot, \cdot \rangle$ is the sum of element wise products of two matrices. Thus, for a given γ_i, one calculates $\Lambda^{(i+1)}$ and checks whether (3) is satisfied. If the condition is satisfied, then the step size γ_i is appropriate to guarantee convergence to a stationary point.

The final algorithm is given in pseudocode in Algorithm 1. See the supplementary material and Appendix A for further discussion.

2.2. *Extensions for Online Reviews Data: A Continuation Ratio Model with Embedded Topic Modeling.* 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 better fit our data, we embed the factorization within a type of ordinal regression, the continuation ratio model (Fienberg, 2007, Ch.6), that incorporates time dynamics and multiple corpora (apps).

We use the continuation ratio model instead of the more popular proportional odds model (McCullagh, 1980) for primarily computational reasons, since the regression coefficients can be solved with standard logistic regression with the continuation ratio model. In practice, several researchers have observed that both forms of ordinal regression yield very similar results (Armstrong and Sloan, 1989; Archer and Williams, 2012; Harrell, 2015). The basic idea is start with the following logit function $\text{logit}(Y = k) = \alpha_k + X\beta$, which we adapt to

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

where $\text{logit}(Y = k) = \log\left(\frac{P(Y=k|Y\geq k,X)}{P(Y>k|Y\geq k,X)}\right)$. The corresponding likelihood is then the product of conditionally independent binomial terms for each level of Y. The log likelihood is given by

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

where $p(k) = P(Y_i = k|Y_i \geq k, (X)_i, \alpha_k, \beta) = \frac{e^{\alpha_k+(X)_i\Lambda\beta}}{1+e^{\alpha_k+(X)_i\Lambda\beta}}$, $(X)_i$ refers to the ith row of X, and Y_k are binary response vectors for categories $k = 1,\ldots,K$ created from Y

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

for $j = 1,\ldots,n$ documents.

An important realization from the likelihood function is that it can be partitioned so that estimating the regression coefficients, holding Λ fixed, can be done through standard binary logistic regression techniques. To our knowledge Cox (1988) and Armstrong and Sloan (1989) were the first to show this for the standard continuation ratio model. The basic idea to apply logistic regression is to stack the recoded the response variables $((Y_k)_j)$, including only observations that satisfy the condition $Y \geq k$ for $k = 1,\ldots,K$, and duplicate corresponding rows to form the design matrix with dummy variables added to model the intercepts α_k. In our context, the same trick can be applied when holding Λ fixed.

Recall our goal is to benchmark multiple apps over time, which calls for a dynamic model

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

where a indexes the set of apps and t denotes time. 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_{ta} are $m \times 1$ regression coefficients for each time interval, app category. Such a model is appropriate as long as the focal app or set of apps maintain the same core functionality, since then we could reasonably expect the discussion topics captured in Λ to remain invariant. By visualizing β_{ta} over time, as shown in Section 5, we can begin to understand the trend of consumer sentiment around topics in Λ for different apps as well the effectiveness of development teams at responding to customer feedback.

Another key assumption is that the regression coefficients β_{ta} are independent of k, the rating level specified for each review. Arguably, this assumption is not germane to our online reviews data, since the occurrence and discussion of topics can have sentiment to them, and thus are related to the overall rating of the review. We also consider a saturated version of the model, where the regression coefficients vary with the level of the response variable $\text{logit}(Y_{ta} = k) = \alpha_{tak} + X\Lambda\beta_{tak}$. Likelihood ratio tests as well as out of sample prediction accuracy rates show that the constrained model is preferred, that is, assuming that $\beta_{tak} = \beta_{ta}$ for all k leads to better statistical and predictive models (see the supplementary material and Appendix B for more information).

Estimation of the dynamic model follows a very similar alternating projected gradient descent algorithm as for the base factorization. When solving for Λ, holding α_{tak} and β_{ta} fixed, we again utilize the projected gradient descent algorithm with appropriate updates for the gradient of Λ and the Armijo rule (Bertsekas, 1976, 1999) to guarantee convergence to a stationary point. Some further details are given in Appendix C. When holding Λ fixed, one can estimate α_{tak} and β_{ta} for each app-time by repeatedly utilizing the logistic regression solution from the static case for each app-time combination. To encourage smoothness in the regression coefficients, we utilize a rolling window so that α_{tak} and β_{ta} are estimated using data from time points t and $t-1$. Another approach yielding similar results would be to add a formal smoothness penalty to the log likelihood. A rigorous implementation of such an approach is outside the scope of this paper, but an interesting area of future work.

Finally, when given a new document x_{ta}, one can predict the rating by selecting the response category with largest probability

$$P(Y_{ta} = 1) = p(1) \tag{4}$$

$$P(Y_{ta} = k) = p(k)\prod_{j=1}^{k-1}(1 - p(j)), k = 2, \ldots, K-1 \tag{5}$$

$$P(Y_{ta} = K) = 1 - \sum_{k=1}^{K-1} P(Y_{ta} = k), \tag{6}$$

where $p(k) = P(Y_{ta} = k|Y \geq k, x_{ta}, \Lambda, \alpha_{tak}, \beta_{ta}) = \frac{e^{\alpha_{tak} + x_{ta}\Lambda\beta_{ta}}}{1+e^{\alpha_{tak} + x_{ta}\Lambda\beta_{ta}}}$.

3. Relation with Topic Modeling Methods. As shown in Table 1, the historical roots of the proposed model 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, since V can take elements of any sign, the interpretation 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, Li and Peng (2008) show 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 previously, 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, Ng and Jordan, 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, Lu and Zhai, 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, Ng and Jordan (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). As mentioned previously, within the software quality and mobile app reviews literature, several papers (e.g., Fu *et al.* (2013); Bavota (2016)) use LDA as part of a multi-stage analysis that feeds into regression models and/or visualizations.

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 $\gamma_{1:K}$.
2. Randomly draw a distribution over topics for the entire corpus $\theta|\alpha \sim Dirichlet(\alpha)$.
3. For each word in a document:
   (a) Randomly sample a topic according to the distribution of topics created in Step 1, i.e., $z_n \sim Multinomial(\theta)$.
   (b) Randomly sample a word according to the topic, i.e., $w_n|z_n \sim \gamma$.

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

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

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).

There have been several related extensions to LDA. For example, Titov and McDonald (2008b) develop the Multi-grain Topic Model for modeling online reviews, which improves the coherence and interpretability of the topic-keywords by enforcing a hierarchical topic structure. The dynamic topic model (Blei and Lafferty, 2006) is another related extension that allows the topic loadings to change over time. These models do not consider document annotations or prediction, as in this work.

The supervised latent Dirichlet allocation (sLDA) of Mcauliffe and Blei (2008) does consider document labels by adding a final stage to the LDA generative process, where a response variable is drawn on each document from the 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.

sLDA has been utilized for recommender systems in the contexts of scientific articles (Wang and Blei, 2011) and physical products (Wu and Ester, 2015), and extended to allow for additional covariates for the regression step (Agarwal and Chen, 2010). We note that because these extensions are motivated by recommender systems, the focus is usually on adding latent variables that capture each user's affinity to different aspects of a product as he or she reviews items (McAuley and Leskovec, 2013). Thus, conceptually the emphasis is on identifying preferences to products (or their attributes) at the user-level. Our work is motivated by a different problem that results in conceptual and modeling differences. Specifically, we are primarily interested in benchmarking from the product developer or designer's point of view, which requires understanding preferences at an aggregate (not user) level over time. Thus, one innovation we incorporate is to characterize the time evolution of how discussion on a common set of topic impacts the average customer's experience for multiple

apps. This is an important extension, since this ultimately allows managers to go beyond a static summary of their app's performance to understand how the customer experience is evolving with different apps and versions. Additionally, because method does not estimate a formal probability distribution for the topic structure, 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 also does not require tuning any parameters, whereas sLDA requires careful specification of hyperparameters. Numerous empirical studies show that the performance of LDA-based methods with online app reviews is sensitive to hyperparameter specification (Lu, Mei and Zhai, 2011; Panichella *et al.*, 2013; Thomas *et al.*, 2013; Bavota, 2016).

Another closely related literature stream is aspect modeling, where the main goal is to decompose a review into multidimensional aspects (topics) with ratings on each aspect (Titov and McDonald, 2008a). Conceptually and at a high level, our work can be viewed as being representative of this stream, since in our model the Λ and β parameters encode, respectively, the "aspects" and their sentiment. The main difference between our work and the aspect modeling literature lies in the observable data structure and precise modeling goals. Most aspect modeling research assumes that ratings on each aspect are observable and have the goal of labeling each sentence within a review with an aspect and sentiment. Common modeling approaches are to extend LDA (Brody and Elhadad, 2010; Titov and McDonald, 2008a; Lu *et al.*, 2011; Jo and Oh, 2011) or pursue other similar latent variable models (Snyder and Barzilay, 2007; Brody and Elhadad, 2010; McAuley, Leskovec and Jurafsky, 2012). For example, in our setting, the referenced aspect models would be appropriate if a reviewer provided separate numerical ratings for several dimensions, like functionality, user interface, reliability of the app, and so on. However, this is rarely the case with app reviews, unlike reviews for restaurants on Yelp where such underlying aspects may be available.

To our knowledge there is one work in aspect modeling that assumes an identical observable data structure. The Latent Aspect Rating Analysis model (LARA; Wang, Lu and Zhai (2010)) aims to infer latent aspects and their sentiment scores from a review's text and its overall review rating. The paper follows a two-stage procedure, first using a seeded and iterative algorithm to identify aspects within each review, followed by a latent rating regression model. While LARA can be extended or modified to predict overall ratings, as in this work, the direct use-cases are distinct, namely annotation of sentences and inference of latent aspect ratings.

Finally we discuss 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.

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 V 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 five-fold cross validation;
4. Supervised LDA (denoted as sLDA). The Dirichlet parameters for the Document/Topic and Topic/Term distributions are chosen through five-fold cross validation and σ^2 is set to be the training sample variance;
5. ℓ_1 penalized linear regression (Lasso; Tibshirani (1996); Friedman, Hastie and Tibshirani (2010)) of the response variable on the document term matrix. Ten-fold cross-validation on the training data is used to select the tuning parameter;
6. 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, Hornik and Meyer, 2008) and "topicmodels" (Grün and Hornik, 2011) libraries. For sLDA, we use the collapsed Gibbs sampler implemented in the "lda" package (Chang, 2012). Code for the proposed Single-Stage Matrix Factorization is provided in the supplementary material.

4.1. *Self Consistency.* Data are generated to study how the proposed model performs under its implied generating process, where $Y|X, \Lambda, \beta, \sigma^2 \sim \text{Normal}(X\Lambda\beta, 1)$. 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, Ng and Jordan, 2003) with both Dirichlet parameters for Document/Topic and Topic/Term distributions set equal to 0.8. The size of the vocabulary is set to $p = 2000$ to roughly match our real dataset and others in the online review space (Büschken and Allenby, 2016; Han *et al.*, 2016).

We vary the number of documents $n = \{100, 1000, 10000\}$ and the number of terms in each document $\mu = \{15, 250, 2000\}$ to study how each model performs in different environments. 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.

When the sample size is 1000 or lower, Lasso and the proposed model perform best. Lasso's performance is perhaps expected given that the generative model can be reparameterized as a linear regression $X\Lambda\beta = X\gamma$, where $\gamma_{p\times 1} = \Lambda\beta$ is a vector of coefficients. It is notable that the proposed model performs well when the number of words in each document is small. This is important especially in the mobile apps context since the overwhelming majority of app reviews 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. When the number of documents is large, we see that all methods perform equally well, meaning that the advantages of supervision diminish in larger datasets.

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 sLDA and Lasso perform best with a $n = 100$ and $\mu = 15$, with the proposed method coming in third. In other settings every method tends to perform similarly. 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.

Self-Consistency

μ	n	LSA	pLSA	LDA	sLDA	Lasso	SSMF
15	100	1.090	1.087	1.088	1.044	1.038	1.040
		(0.004)	(0.004)	(0.004)	(0.003)	(0.003)	(0.006)
15	1000	1.039	1.038	1.038	1.037	1.032	1.033
		(0.001)	(0.001)	(0.001)	(0.001)	(0.001)	(0.004)
15	10000	1.032	1.032	1.032	1.036	1.025	1.032
		(0.001)	(0.001)	(0.001)	(0.001)	(0.001)	(0.001)
250	100	1.056	1.058	1.057	1.323	1.018	1.009
		(0.003)	(0.003)	(0.003)	(0.012)	(0.003)	(0.003)
250	1000	1.007	1.007	1.007	1.013	1.003	1.003
		(0.001)	(0.001)	(0.001)	(0.001)	(0.001)	(0.001)
250	10000	1.002	1.002	1.002	1.004	1.002	1.002
		(0.001)	(0.001)	(0.001)	(0.001)	(0.001)	(0.001)
2000	100	1.050	1.049	1.049	1.710	1.011	0.999
		(0.003)	(0.003)	(0.003)	(0.028)	(0.003)	(0.003)
2000	1000	1.004	1.004	1.004	1.027	1.001	1.001
		(0.001)	(0.001)	(0.001)	(0.002)	(0.001)	(0.001)
2000	10000	0.998	0.998	0.998	1.002	0.999	0.998
		(0.002)	(0.002)	(0.002)	(0.002)	(0.002)	(0.002)

sLDA Generating Process

μ	n	LSA	pLSA	LDA	sLDA	Lasso	SSMF
15	100	1.110	1.109	1.112	1.077	1.073	1.106
		(0.005)	(0.005)	(0.005)	(0.005)	(0.004)	(0.013)
15	1000	1.057	1.057	1.057	1.056	1.052	1.053
		(0.002)	(0.002)	(0.002)	(0.002)	(0.002)	(0.008)
15	10000	1.051	1.051	1.051	1.053	1.051	1.052
		(0.002)	(0.002)	(0.002)	(0.002)	(0.002)	(0.002)
250	100	1.102	1.102	1.102	1.293	1.077	1.077
		(0.005)	(0.005)	(0.005)	(0.012)	(0.005)	(0.006)
250	1000	1.068	1.068	1.068	1.070	1.069	1.068
		(0.004)	(0.004)	(0.004)	(0.004)	(0.004)	(0.004)
250	10000	1.054	1.053	1.054	1.059	1.053	1.050
		(0.002)	(0.002)	(0.002)	(0.002)	(0.002)	(0.002)
2000	100	1.100	1.100	1.099	1.730	1.206	1.060
		(0.005)	(0.005)	(0.005)	(0.035)	(0.025)	(0.005)
2000	1000	1.072	1.072	1.072	1.087	1.095	1.070
		(0.004)	(0.004)	(0.004)	(0.004)	(0.011)	(0.004)
2000	10000	1.001	1.001	1.001	1.002	0.999	1.000
		(0.002)	(0.002)	(0.002)	(0.002)	(0.002)	(0.002)

TABLE 2

Root Mean Squared Error averaged over all ranks from the simulation study with standard errors in parentheses.

Platform-App	Number Reviews	Average Review Length (characters)	% Reviews with "!"	Yule's K
iTunes-Expedia	2772	98.715	28.968	62.325
iTunes-Kayak	13120	68.948	31.623	59.289
iTunes-TripAdvisor	19519	107.949	32.235	71.308
Google Play-Expedia	6999	95.246	15.416	69.915
Google Play-Kayak	21059	58.023	15.267	49.637
Google Play-TripAdvisor	41347	65.660	14.069	53.895

TABLE 3

Summary statistics for the online reviews data. Yule's K is a measure of vocabulary richness, where higher numbers indicate a more diverse vocabulary (Holmes, 1985).

5. iTunes and Google Play App Reviews. We now demonstrate the method's real-life viability and applicability by using the mobile apps marketplace data from the apps provided by Expedia, Kayak, and TripAdvisor that we described earlier. We begin by discussing the preprocessing and model selection steps, followed by a detailed discussion of the findings.

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 and removing punctuation, stopwords (e.g., "a", "and", "the"), and any terms composed of less than three characters. In addition to counting the frequency of single words, we also count bigrams, which are all two word phrases that appear in the corpus. For example, the sentence "this is a wonderful app" is tokenized into single words "this", "is", "a", "wonderful", "app" as well as two-word phrases "this is", "is a", "a wonderful", and "wonderful app". After counting all unigrams and bigrams, we remove terms that have occurred in less than 20 reviews and apply TFIDF weighting. The resulting total vocabulary size is 2583 for reviews from iTunes and 1389 for reviews from Google Play.

Table 3 shows an overview of the review data, where we see that despite being a younger platform, Google Play has more reviews for every app. The customer writing style also seems to vary by platform. iTunes reviews tend to be longer, potentially more emotional due to greater number of exclamation points, and have a higher lexical diversity. We analyze each platform separately due to these differences in addition to the fact that the hardware (mobile phones and tablets) that run the mobile apps vary across platforms, as do the underlying development enviroments that used to develop code for the apps. We also define time in terms of year-quarters in our analysis to avoid sparsity issues early in an app's lifecycle and also to roughly match the approximate rate at which upgrades and new functionalities are released for the apps in our sample. The last observed quarter for each platform is withheld as the test set.

Cross-validation applied to the training sample selects five topics for iTunes and four topics for the Google Play platform according to misclassification error rate (MER). Table 4 shows the top ten keywords from our final models using. Each of the topics were manually labeled with headings after inspecting the keywords and reviews that loaded most heavily onto each topic. For instance, Table 5 shows reviews that correspond to the largest values in columns (topics) of $X\Lambda$ for the Google Play data. Due to space constraints, the top reviews for all topics and the iTunes data are omitted.

iTunes

Topic 1 Keywords Usability (Online Reviews)	Topic 2 Keywords Functionality (Reservations)
useful, helpful, good, cool, awsome, nice, great app, availability, great, wish app	indispensable, reviewers, advisor always, since last, app also, establishment, helpfull, properties, helping, reviews pictures
Topic 3 Keywords Overall Quality	**Topic 4 Keywords Versioning**
great, awesome, love, easy, app, best, use, amazing, great app, perfect, easy use	fill, forced, changing, worthless, latest version, returned, happened, old version, back old, bring back
Topic 5 Keywords Functionality (Software Bugs)	
emails, crashes, almost every, dont want, one star, category, glitch, apply, internet connection, customer service	

GooglePlay

Topic 1 Keywords Functionality (Reservations)	Topic 2 Keywords Usability (UI & Design)
brilliant, comment, paid, wasnt, seriously, scroll, coupon, hotel flight, apparently, main	helpful, half, average, expensive, enter, agent, availability, advertised, liked, order
Topic 3 Keywords Usability (Composing Reviews)	**Topic 4 Keywords Installation & Versioning**
write reviews, find way, asking, poor, line, app im, searched, app book, either, downloading	bloatware, stupid, uninstalled, uninstall, useless, crap, remove, month, return, expensive, message

TABLE 4

The top ten topic keywords from estimating five topics on iTunes and four topics on Google Play.

Reviews that load most heavily onto "Usability (Composing Reviews)"
"i downloaded app on my new phone to write reviews i was able to find what i wanted to write reviews and entering the review was easy i havent searched for lists yet"
"just cant submit have to paste cant type into input line for submit name"
"on several occasions reviews that i have submitted have either failed to submit successfully and give an error message or have mysteriously disappeared after apparently being submitted successfully another variant problem is that draught that are saved can also disappear it is extremely frustrating to spend perhaps minutes writing a review on a mobile device only to find that the time has been wasted in terms of displaying tripadvisor information on the move it is reasonable"
"but in entering a comment i went up to add something and it wouldnt allow me to scroll back down to complete my comment so i had to either completely redo the review or just enter and hope for the best since the livelihood of establishments depend upon these comments this should not happen please fix it this happened on a galaxy tablet"
Reviews that load most heavily onto "Installation & Versioning"
"new mobile tablet app is frustrating unable to book hotel reservation with more than one traveler unable to book multiple rooms unable to make changes in reservation unable to cancel reservation through mobile app unable to use online support telephone support slow and useless credit for first time mobile use unavailable after telephone support call hang up dial the hotel direct and dump this app"
"i cant uninstall this app on my s i dont have a need for it please let me remove it im so sick of bloatware apple has the right idea when it comes to controlling what goes on there phonessmh sprint and samsung you make it hard to be a loyal customer when there are apps on my phone that i dont need or use that cant be uninstalled"
"samsung should stop adding crap bloatware and leave us course what we want installed on our phone why we cant remove such app bring the phone with the app installed but leave us remove it i never use such app and i dont need it"
"cant uninstall cant remove"

TABLE 5

Top reviews associated with different topics for the Google Play platform.

Assessing the quality of topic keywords can be challenging, since interpretability is a difficult characteristic to quantify. Mimno *et al.* (2011) provide one solution in a measure called topic coherence, where the general idea is to gauge the interpretability of each topic based on co-occurrences of its keywords. The average keyword coherence is defined as

$$\text{Coherence} = \frac{2}{Kp(p-1)} \sum_{k=1}^{K} \sum_{u=2}^{p} \sum_{v=1}^{u-1} \log \left(\frac{D(w_u^k, w_v^k) + 0.01}{D(w_v^k)} \right),$$

where $(w_1^k, \ldots, w_p^k)$ is the list of p top words in topic k, K is the number of topics, $D(w)$ is the number of reviews containing the word w, and $D(w, w')$ is the number of reviews containing both w and w'. The constant 0.01 is added to avoid taking the log of zero when two keywords do not co-occur over all documents. Coherence is bounded above by zero; model results with larger coherence scores have been shown to be more interpretable by human judges (Mimno *et al.*, 2011). To measure redundancy of the recovered topics, we report Uniqueness, which is defined as the average proportion of keywords in each topic that do not appear as keywords for other topics (similar to "inter-topic similarity" in Arora *et al.* (2013)). Larger values indicate more useful results. The top and middle panels of Table 6 shows that the proposed method is generating interpretable and useful results. In contrast to competing methods that tend to score well on either Coherence or Uniqueness, SSMF is competitive on both dimensions.

A third way to validate our results is to compare out of sample forecasts. We generate predictions on the test set by using the estimated document-topic matrix $\hat{\Lambda}$ and regression coefficients from the most recent quarter $\hat{\beta}_{T-1,a}$. We again benchmark the performance against LSA, pLSA, LDA, sLDA, and Lasso. The two-stage procedures utilize the continuation ratio model in the second stage and Lasso refers to the ℓ_1 penalized continuation ratio model of Archer and Williams (2012). We also include a standard continuation ratio model with all unigrams and bigrams as covariates and no penalty. The bottom panel of Table 6 shows that Lasso and the proposed method produce the most accurate predictions. These results are consistent with the simulation study that showed these two methods performing well among the tested methodologies when the sample size is in the thousands, which approximately matches the number of reviews received each quarter collectively for the three apps.

Keyword Coherence

Platform	LSA	pLSA	LDA	sLDA	Lasso	Ordinal Regression	SSMF
iTunes	-0.566	-0.750	-0.564	-0.688	NA	NA	-0.743
	(0.355%)	(32.979%)	max	(27.035%)			(26.017%)
Google	-0.862	-1.113	-0.826	-1.057	NA	NA	-0.926
Play	(4.358%)	(34.746%)	max	(27.966%)			(12.107%)

Uniqueness

Platform	LSA	pLSA	LDA	sLDA	Lasso	Ordinal Regression	SSMF
iTunes	0.096	0.726	0.132	0.344	NA	NA	0.592
	(86.777%)	max	(81.818%)	(52.617%)			(18.457%)
Google	0.188	0.843	0.170	0.475	NA	NA	0.580
Play	(77.699%)	max	(79.834%)	(43.654%)			(31.198%)

Misclassification Error Rates

Platform	LSA	pLSA	LDA	sLDA	Lasso	Ordinal Regression	SSMF
iTunes	0.319	0.313	0.320	0.712	0.294	0.319	0.299
	(8.503%)	(6.463%)	(8.844%)	(142.180%)	min	(8.503%)	(1.701%)
Google	0.373	0.376	0.376	0.639	0.334	0.377	0.327
Play	(14.067%)	(14.984%)	(14.984%)	(95.413%)	(2.140%)	(15.291%)	min

TABLE 6

Average topic coherence and uniqueness based on the top 100 topic keywords, and out of sample misclassification error rate when predicting online review ratings in the first quarter of 2014 for iTunes and third quarter of 2014 for Google Play. The reported percentage is the relative percentage of difference to the best result. Note that sLDA was run assuming a normally distributed response as this is the only working option in the public code. All other methods were combined with a continuation ratio model.

Having chosen and validated the proposed models, we turn to synthesizing our findings from the mobile apps data and the the estimation of Λ and β, which are summarized in Figures 2 and 3, respectively. Figure 2 shows the amount of discussion in each quarter on each topic, assessed by taking column sums of $X_{ta}\Lambda$. Figure 3 displays the regression coefficients transformed into probabilities, which is necessary to avoid interpretation difficulties that arise with viewing the coefficients directly. Specifically, $P(Y_{ta} = k)$ is calculated by considering a hypothetical document that loads onto a single topic, where $X_{ta}\Lambda = e_m$ and e_m is a vector with 1 in the m-th position and zero elsewhere. The required marginal probabilities can be readily computed using (4)-(6).

These two figures, combined with the ratings evolutions in Figure 1, show several interesting patterns that help characterize the evolution of each app over time while also identifying areas of improvement for the respective app development teams. We see a small dip in the overall ratings for the Kayak app on iTunes in the third and fourth quarter of 2013. This decrease coincided with discussion around two issues: software bugs (crashes, API errors, etc.) and versioning, which are then associated with higher chances of the app being rated lower on the 5-point scale by users. Even though the volume of discussion was fairly stable, the topics became increasingly toxic as users were rating the app more harshly along these dimensions, thereby dragging down the overall rating. Similarly, we can see the odds of receiving 1-star reviews strongly increasing with the occurrence of these topics within reviews, coinciding with a negative episode in the overall ratings for the Expedia app on iTunes between the third quarter of 2012 and the third quarter of 2013. In fact, we can see from Figure 3 that Expedia has persistent problems with versioning and software bugs on both platforms that are on-going at the end of the data. On Google Play, Expedia is generally rated lower than its competitors, and we see that, in addition to versioning, the company had difficulty especially in 2012 with general user interface issues around the launch of the app, followed by difficulties around composing and posting online reviews by its users. In contrast, TripAdvisor has consistently been rated highly on both platforms since the apps were introduced to the public. Interestingly, on both platforms we see installation and versioning as significant sources of discontent from users, though the amount of discussion on these topics has been low. Nonetheless, it is relevant to note that TripAdvisor forces automatic updates of its apps on both platforms and is even embedded in the operating system as a default program on certain versions of Android mobile phones, which is at the heart of the negative feedback from users. This raises an interesting tradeoff for TripAdvisor's mobile strategy - between the options of increasing its user base by being embedded within the Android system versus the cost of alienating some users who may be annoyed at having to uninstall the app manually.

6. Conclusion. Consider a mobile app developer who has introduced an app on the Google Play app store and has received, over a period of time, several thousand reviews from users. Ideally, the developer would like to extract some information from these reviews that will help inform where the main problems are with the developed app, as well as where the app stands with respect to competitor apps on dimensions that relate to user experience or service quality. Furthermore, over time, the developer would like to understand time trends relating to dimensions of feedback from online reviews, and how these are associated with the received app rating. In this paper, we present an ordinal regression framework with embedded topic modeling to recover topics from online reviews that are predictive of the star rating in addition to being useful for understanding the underlying textual themes. Moreover, this model performs particularly well in the specific context of mobile apps, where reviews tend to be short, change over time with app versions, but have common elements in terms of what users tend to discuss in these reviews.

We demonstrated how the model can be applied for benchmarking by analyzing mobile app reviews for Expedia, Kayak, and TripAdvisor. Specifically, by investigating the trend in overall

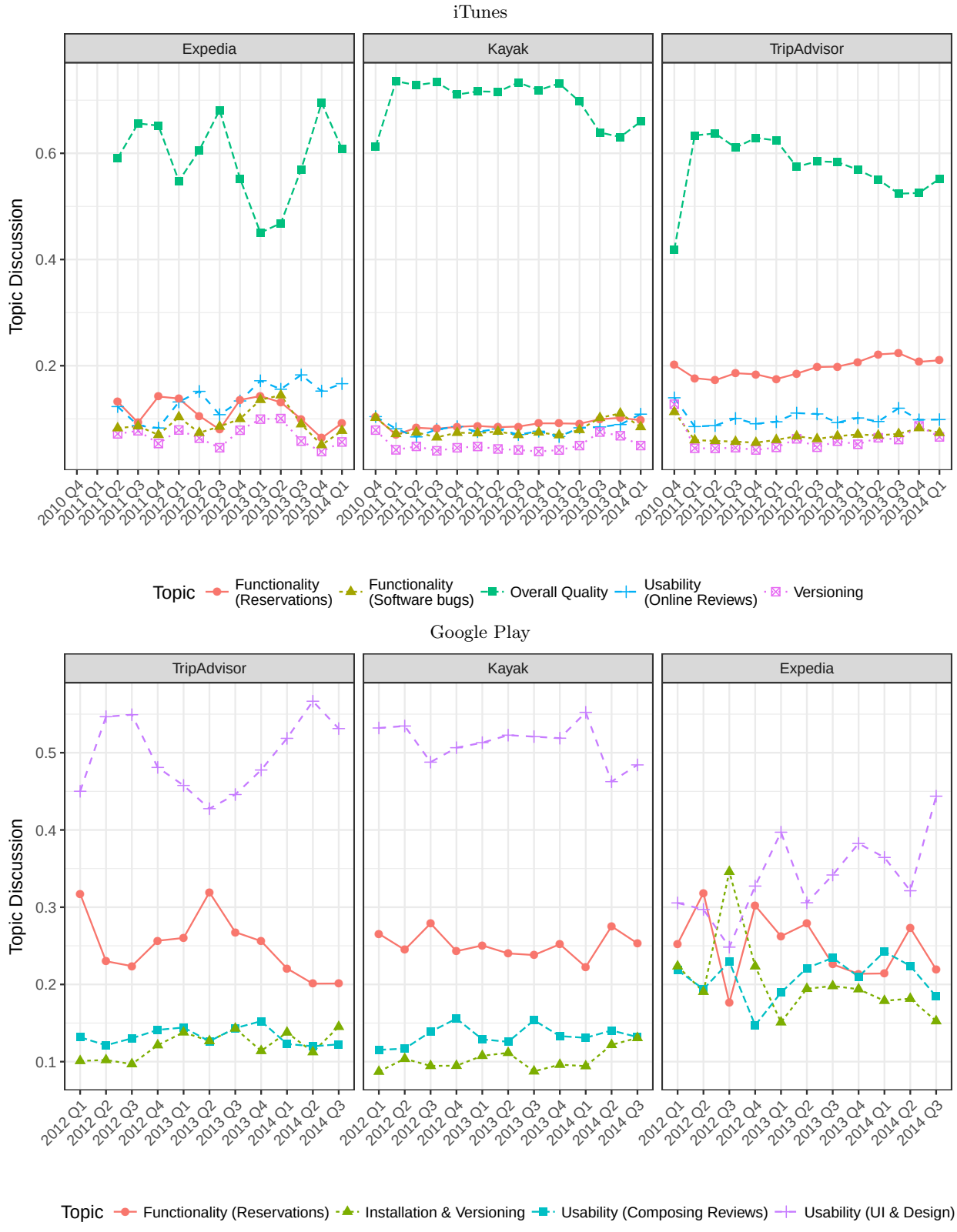

Fig 2: Prevalence of topics in reviews over time.

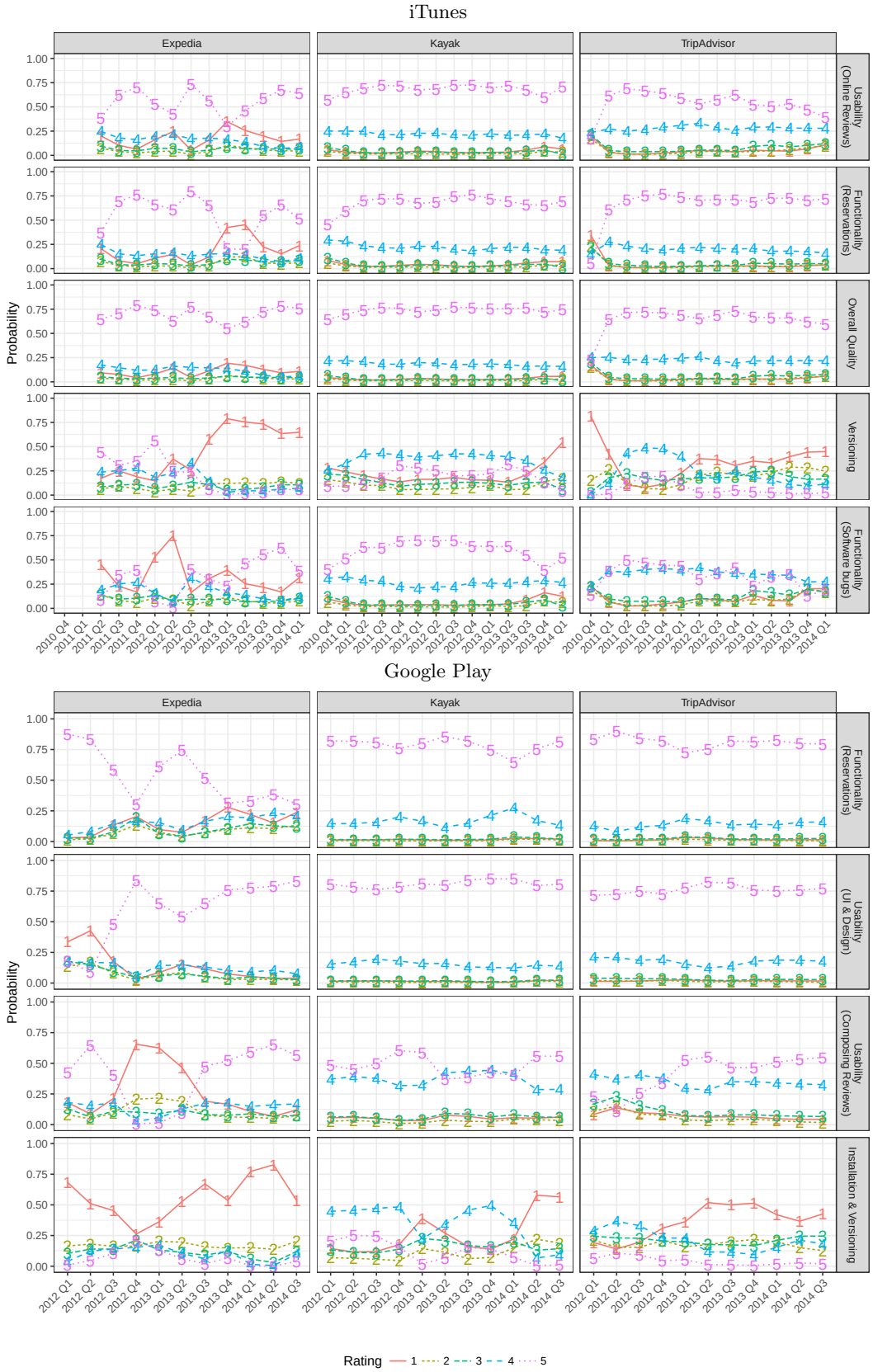

Fig 3: Probability of ratings by topic appearance.

ratings in combination with the estimated ratings probability by topic, we identified potential reasons behind poor user satisfaction that resulted in negative movements in the overall star rating for an app. For instance, we observe that the increased odds of receiving 1-star reviews during the final quarter of our dataset is associated with negative feedback related to the topic described by versioning issues. On deeper examination, we conclude that all three companies on the two platforms should be particularly cognizant of the pros and cons of the versioning strategy they espouse. While forcing users into app updates may help improve the user experience for some users by fixing software bugs or introducing new and important features to the updated app, there is the potential cost of alienating a different and potentially overlapping set of users when such automatic updates are too frequent or add low quality features. Such a potential tradeoff can be deduced by app developers through the use of the SSMF approach we describe. In a related manner, and as discussed above for TripAdvisor, a similar potential downside from a development perspective exists with respect to the strategy of preinstalling the app on mobile phones. While this strategy helps some users, it can cause dissatisfaction to others who are faced with having to delete the app manually. In yet another instance during the final quarter of data, Expedia's iTunes-based users report the presence of critical software bugs while Android-based users complain about the reservation functionality. Based on our methodology, it would be possible for Expedia to corroborate these initial insights through traditional software testing and redirect their app development team's efforts more effectively towards tackling these sources of discontent among its users.

It is interesting to note that our proposed model and Lasso generally performed the best, and on par with each other, among the tested methodologies in terms of forecasting accuracy on both the real reviews data as well as on simulated data. These results are consistent with O'Callaghan *et al.* (2015) who showed that NMF style factorizations may lead to better solutions compared to LDA-based approaches, especially with niche or non-mainstream corpora, such as reviews for mobile apps on mobile devices, which tend to be short and informal. Another factor determining the efficacy of the proposed model, relative to other models, is sample size (number of documents). In the simulation, Lasso and the proposed models were preferred when the sample size was in the thousands or smaller. At larger sample sizes in each time point, our simulation indicated that two-stage procedures with standard topic modeling in the first stage perform equally well.

While we consider three clearly competitive apps within the same industry here, an important and particularly insightful extension of our methodology could be to recover market structure for the entire app market using online app reviews. Market structure is an important factor in firm-level decision making pertaining to product development, pricing, and marketing strategies. Yet, in general with mobile apps, the appropriate set of benchmark or competitive apps is unclear, especially from the consumers's perspective. For instance, if an app streams video even without it being a core feature, the average consumer might benchmark this functionality 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 could be derived from the set of online reviews associated with each of these apps, thereby enhancing the value that companies gain from a better understanding of online reviews. Tackling this problem would likely require analyzing data from a much broader set of mobile apps, potentially the entire marketplace, which raises several methodological issues from preprocessing the data (Fu *et al.*, 2013) to summarizing network structure and trends over time. 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 benchmarking, understanding market dynamics, and prediction accuracy.

Acknowledgements. This material is based upon work supported by the National Science Foundation under Grant No. 1633158 (Mankad).

The authors would also like to thank the anonymous referees, the Associate Editor, and Editor for constructive comments and suggestions that resulted in a much improved paper.

SUPPLEMENTARY MATERIAL

Raw Data and R Code
(doi: COMPLETED BY THE TYPESETTER; .zip). The zip file contains the raw online reviews data for the three apps on both platforms in addition to implementations in R of the proposed matrix factorization.

References.

ABIresearch, (2012). M-Commerce Growing to 24Smartphone Adoption. https://www.abiresearch.com/press/m-commerce-growing-to-24-of-total-e-commerce-marke/ Accessed: 2016-06-18.

Abrahams, A. S., Fan, W., Alan Wang, G., Zhang, Z. J. and Jiao, J. (2015). An Integrated Text Analytic Framework for Product Defect Discovery. *Production and Operations Management* **24** 975–990.

Agarwal, D. and Chen, B.-C. (2010). fLDA: matrix factorization through latent dirichlet allocation. In *Proceedings of the third ACM international conference on Web search and data mining* 91–100. ACM.

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

Archer, K. and Williams, A. (2012). L 1 penalized continuation ratio models for ordinal response prediction using high-dimensional datasets. *Statistics in medicine* **31** 1464–1474.

Armstrong, B. G. and Sloan, M. (1989). Ordinal regression models for epidemiologic data. *American Journal of Epidemiology* **129** 191–204.

Arora, S., Ge, R., Halpern, Y., Mimno, D., Moitra, A., Sontag, D., Wu, Y. and Zhu, M. (2013). A practical algorithm for topic modeling with provable guarantees. In *International Conference on Machine Learning* 280–288.

Bavota, G. (2016). Mining Unstructured Data in Software Repositories: Current and Future Trends. In *2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER)* **5** 1–12. IEEE.

Bertsekas, D. P. (1976). On the Goldstein-Levitin-Polyak gradient projection method. *Automatic Control, IEEE Transactions on* **21** 174–184.

Bertsekas, D. P. (1999). Nonlinear programming.

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

Blei, D. M. and Jordan, M. I. (2006). Variational inference for Dirichlet process mixtures. *Bayesian Analysis* **1** 121–143. 10.1214/06-BA104

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

Blei, D. M., Ng, A. Y. and Jordan, M. I. (2003). Latent dirichlet allocation. *the Journal of machine Learning research* **3** 993–1022.

Boyd-Graber, J., Mimno, D., Newman, D., Airoldi, E. M., Blei, D., Erosheva, E. A. and Fienberg, S. E. (2014). Care and Feeding of Topic Models: Problems, Diagnostics, and Improvements. *Handbook of Mixed Membership Models and Their Applications*.

Brody, S. and Elhadad, N. (2010). 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* 804–812. Association for Computational Linguistics.

Burtch, G. and Hong, Y. (2014). What Happens When Word of Mouth Goes Mobile? *Proceedings of the International Conference on Information Systems*.

Büschken, J. and Allenby, G. M. (2016). Sentence-Based Text Analysis for Customer Reviews. *Marketing Science* **35** 953–975.

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

Chang, J. (2012). lda: Collapsed Gibbs sampling methods for topic models. R package version 1.3.2.

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

Chen, N., Lin, J., Hoi, S. C., Xiao, X. and Zhang, B. (2014). AR-Miner: mining informative reviews for developers from mobile app marketplace. In *Proceedings of the 36th International Conference on Software Engineering* 767–778. ACM.

Cox, C. (1988). Multinomial regression models based on continuation ratios. *Statistics in Medicine* **7** 435–441.

Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W. and Harshman, R. A. (1990). Indexing by latent semantic analysis. *JAsIs* **41** 391–407.

Ding, C., Li, T. and Peng, W. (2008). On the equivalence between Non-negative Matrix Factorization and Probabilistic Latent Semantic Indexing. *Comput. Stat. Data Anal.* **52** 3913–3927.

Ding, C., Li, T. and Jordan, M. I. (2010). Convex and Semi-Nonnegative Matrix Factorizations. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **32** 45-55. 10.1109/TPAMI.2008.277

Feinerer, I., Hornik, K. and Meyer, D. (2008). Text Mining Infrastructure in R. *Journal of Statistical Software* **25** 1–54.

Fienberg, S. E. (2007). *The analysis of cross-classified categorical data*. Springer Science & Business Media.

Friedman, J., Hastie, T. and Tibshirani, R. (2010). Regularization Paths for Generalized Linear Models via Coordinate Descent. *Journal of Statistical Software* **33** 1–22.

Fu, B., Lin, J., Li, L., Faloutsos, C., Hong, J. and Sadeh, N. (2013). 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* 1276–1284. ACM.

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

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

Grün, B. and Hornik, K. (2011). topicmodels: An R Package for Fitting Topic Models. *Journal of Statistical Software* **40** 1–30.

Han, H. J., Mankad, S., Gavirneni, S. and Verma, R. (2016). What guests really think of your hotel: Text analytics of online customer reviews. *Cornell Hospitality Report*.

Harrell, F. (2015). *Regression modeling strategies: with applications to linear models, logistic and ordinal regression, and survival analysis*. Springer.

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

Holmes, D. I. (1985). The analysis of literary style–a review. *Journal of the Royal Statistical Society. Series A (General)* 328–341.

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

Ickin, S., Wac, K., Fiedler, M., Janowski, L., Hong, J.-H. and Dey, A. K. (2012). Factors influencing quality of experience of commonly used mobile applications. *Communications Magazine, IEEE* **50** 48–56.

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

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

Krishnan, M. S., Kriebel, C. H., Kekre, S. and Mukhopadhyay, T. (2000). An empirical analysis of productivity and quality in software products. *Management science* **46** 745–759.

Lee, D. D. and Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature* **401** 788-791.

Lee, D. D. and Seung, H. S. (2001). Algorithms for non-negative matrix factorization. *Advances in neural information processing systems* 556–562.

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

Lim, S. L., Bentley, P. J., Kanakam, N., Ishikawa, F. and Honiden, S. (2015). Investigating country differences in mobile app user behavior and challenges for software engineering. *Software Engineering, IEEE Transactions on* **41** 40–64.

Lin, C.-b. (2007). Projected gradient methods for nonnegative matrix factorization. *Neural computation* **19** 2756–2779.

Lu, Y., Mei, Q. and Zhai, C. (2011). Investigating task performance of probabilistic topic models: an empirical study of PLSA and LDA. *Information Retrieval* **14** 178–203. 10.1007/s10791-010-9141-9

Lu, B., Ott, M., Cardie, C. and Tsou, B. K. (2011). Multi-aspect sentiment analysis with topic models. In *Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on* 81–88. IEEE.

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

Mankad, S. and Michailidis, G. (2013). 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* 288-291. 10.1109/CAMSAP.2013.6714064

Mankad, S. and Michailidis, G. (2015). Analysis of Multiview Legislative Network with Structured Matrix Factorization: Does Twitter Influence Translate to the Real World? *Annals of Applied Statistics*.

Mankad, S., Han, H. J., Goh, J. and Gavirneni, S. (2016). Understanding Online Hotel Reviews through Automated Text Analysis. *Service Science* **8** 124-138. 10.1287/serv.2016.0126

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

McAuley, J. and Leskovec, J. (2013). Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems* 165–172. ACM.

Mcauliffe, J. D. and Blei, D. M. (2008). Supervised topic models. In *Advances in neural information processing systems* 121–128.

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

Mimno, D., Wallach, H. M., Talley, E., Leenders, M. and McCallum, A. (2011). Optimizing semantic coherence in topic models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* 262–272. Association for Computational Linguistics.

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

O'Callaghan, D., Greene, D., Carthy, J. and Cunningham, P. (2015). An analysis of the coherence of descriptors in topic modeling. *Expert Systems with Applications* **42** 5645–5657.

Panichella, A., Dit, B., Oliveto, R., Di Penta, M., Poshyvanyk, D. and De Lucia, A. (2013). 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* 522–531. IEEE Press, Piscataway, NJ, USA.

Panichella, S., Di Sorbo, A., Guzman, E., Visaggio, C. A., Canfora, G. and Gall, H. C. (2015). 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* 281–290. IEEE.

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

Porteous, I., Newman, D., Ihler, A., Asuncion, A., Smyth, P. and Welling, M. (2008). 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* 569–577. ACM, New York, NY, USA. 10.1145/1401890.1401960

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

R Core Team, (2014). R: A Language and Environment for Statistical Computing R Foundation for Statistical Computing, Vienna, Austria.

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

Salton, G. and Michael, J. (1983). McGill. *Introduction to modern information retrieval* 24–51.

Serrano, N., Hernantes, J. and Gallardo, G. (2013). Mobile web apps. *Software, IEEE* **30** 22–27.

Snyder, B. and Barzilay, R. (2007). Multiple Aspect Ranking Using the Good Grief Algorithm. In *HLT-NAACL* 300–307.

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

Thomas, S. W., Nagappan, M., Blostein, D. and Hassan, A. E. (2013). The Impact of Classifier Configuration and Classifier Combination on Bug Localization. *IEEE Transactions on Software Engineering* **39** 1427-1443. 10.1109/TSE.2013.27

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)* 267–288.

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

Titov, I. and McDonald, R. T. (2008a). A Joint Model of Text and Aspect Ratings for Sentiment Summarization. In *ACL* **8** 308–316. Citeseer.

Titov, I. and McDonald, R. (2008b). Modeling online reviews with multi-grain topic models. In *Proceedings of the 17th international conference on World Wide Web* 111–120. ACM.

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

Wang, H., Lu, Y. and Zhai, C. (2010). 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* 783–792. ACM.

Wasserman, A. I. (2010). Software engineering issues for mobile application development. In *Proceedings of the*

FSE/SDP workshop on Future of software engineering research 397–400. ACM.

Wu, Y. and Ester, M. (2015). 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* 199–208. ACM.

Xu, W., Liu, X. and Gong, Y. (2003). 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* 267–273. ACM, New York, NY, USA. 10.1145/860435.860485

APPENDIX A: ALGORITHM FOR THE SINGLE STAGE MATRIX FACTORIZATION WITH NORMAL RESPONSES

The final algorithm for the SSMF is given in Algorithm 1.

Updating Λ and specifically searching for an appropriate γ_i when updating Λ is the most time-consuming task. The major computation when searching for a good step size is $\langle \Delta_{\Lambda^{(i)}}, \Lambda^{(i+1)} - \Lambda^{(i)} \rangle$, where $\langle \cdot, \cdot \rangle$ is the sum of element wise products of two matrices.

Breaking down this specific calculation, we focus on the gradient which is defined in (2). X^TX, $\beta\beta^T$, and $X^TY\beta^T$ can all be precomputed before entering into the step size search. In fact, X^TX and X^TY can be computed before beginning Algorithm 1. Due to these precomputations, the cost of searching for the step size is

$$\underbrace{\mathcal{O}(p^2n) + \mathcal{O}(pn) + \mathcal{O}(pm) + \mathcal{O}(m^2)}_{\text{Precomputed } X^TX,\ X^TY,\ (X^TY)\beta^T, \text{ and } \beta\beta^T} + \#\text{sub-iterations} \times (\underbrace{\mathcal{O}(p^2m)}_{(X^TX)\Lambda} + \underbrace{\mathcal{O}(pm^2)}_{\Lambda(\beta\beta^T)}).$$

Adding in the cost of the element-wise sum and estimating β with standard procedures ($\mathcal{O}((n+m)m^2)$), the overall cost of the algorithm is

$$\mathcal{O}(p^2n) + \mathcal{O}(pn) +$$
$$\#\text{iterations} \times \left(\mathcal{O}((n+m)m^2) + \mathcal{O}(pm) + \mathcal{O}(m^2) + \#\text{sub-iterations} \times \left(\mathcal{O}(p^2m) + \mathcal{O}(pm^2)\right)\right).$$

As long as the number of sub-iterations is small, the algorithm is efficient for the given data, especially since the vocabulary size is not extremely large. To this end, we utilize the heuristic of using α_{i-1} as an initial guess for γ_i, and set $\sigma = 0.01$ and $\gamma = 0.9$. Figure 4 shows the algorithm results in estimates that monotonically improve at each iteration and converge fairly quickly. In our experiments, the relative difference between objective values converged to within 10^{-4} typically within 15 iterations.

APPENDIX B: COMPARING THE CONSTRAINED AND SATURATED CONTINUATION RATIO MODELS

In this section we evaluate whether the constrained or saturated model is preferred. The results presented here use the real data from the app marketplaces and the final proposed model that includes regression coefficients varying over time and app. In this framework, the constrained model specifies that $\beta_{tak} = \beta_{ta}$ for all k.

We compare the nested models using likelihood ratio tests. Define the likelihood ratio statistic

$$G = 2\left(l(\text{Saturated Model}) - l(\text{Constrained Model})\right),$$

following a Chi-Squared distribution with $df_2 - df_1$ degrees of freedom, where

$$\begin{aligned} df_1 &= \#\text{Topics} * \#\text{Apps} * \#\text{Time points} \\ df_2 &= \#Topics * \#Apps * \#\text{Time points} * (\#\text{ Rating categories} - 1). \end{aligned}$$

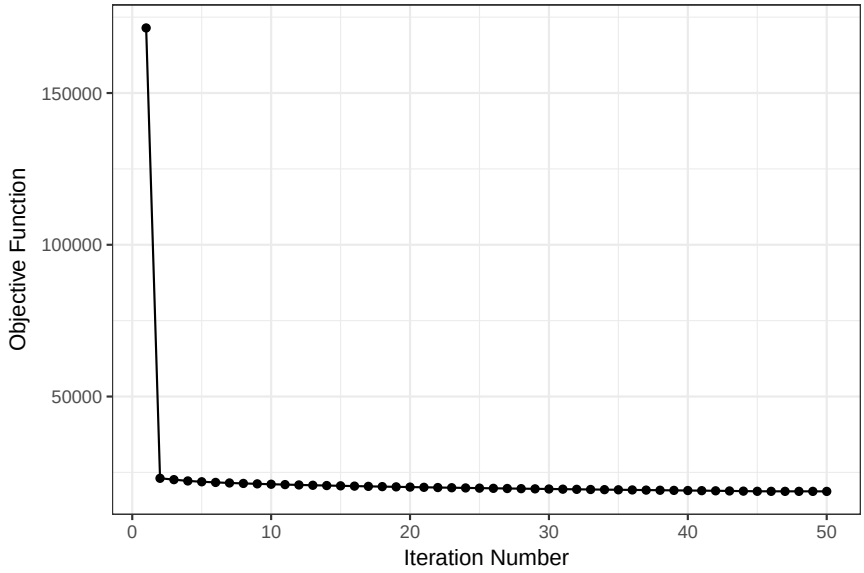

Fig 4: One instance of the objective function at each iteration of the SSMF estimation. The alternating projected gradient descent algorithm monotonically improves the estimates with respect to the objective function.

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 $\gamma_i = 1, \gamma = 0.9$
while $\delta \geq \epsilon$ **and** $i \leq$ max iterations **do**
  $\gamma_{i+1} = \gamma_i$
  if γ_{i+1} satisfies (3) **then**
    repeat
      $\gamma_{i+1} = \frac{\gamma_{i+1}}{\gamma}$
    until γ_{i+1} does not satisfies (3)
  else
    repeat
      $\gamma_{i+1} = \gamma_{i+1}\gamma$
    until γ_{i+1} satisfies (3)
  end if
  Set $\Lambda^{(i+1)} = P\left(\Lambda^{(i)} - \gamma_{i+1}(X^T X \Lambda \beta\beta^T - X^T Y \beta^T)\right)$
  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

Platform	Saturated SSMF	SSMF
iTunes	0.300	0.299
Google Play	0.337	0.327

TABLE 7

Out of Sample Misclassification Error Rates of the proposed model with regression coefficients that vary over time and app. The Saturated SSMF allows the regression coefficients to additionally vary for each category versus fixed over ratings categories.

On the iTunes data the likelihood ratio statistic $G = 12.575$ has a p-value close to 1.000 and on Google Play $G = 423.811$ has a p-value of 0.161. Failing to reject the null hypothesis on both platforms indicates that the constrained model fits as well as the saturated version. Thus, we prefer the constrained version of the model.

This decision is confirmed by the out of sample misclassification error rates on our online reviews data in Table 7 . The constrained model performs favorably, especially on the Google Play data, indicating that the more complex, saturated model likely overfits the data.

APPENDIX C: ESTIMATION OF THE DYNAMIC FACTORIZATION EMBEDDED CONTINUATION RATIO MODEL

The log-likelihood function for the proposed model is

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

When solving for Λ, holding all other parameters 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

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

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

$$l(\Lambda^{(i+1)}, \beta_{ta}|Y_{ta}, X_{ta}) - l(\Lambda^{(i)}, \beta_{ta}|Y_{ta}, X_{ta}) \leq \sigma\langle\Delta_{\Lambda^{(i)}}, \Lambda^{(i+1)} - \Lambda^{(i)}\rangle,$$

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

SHAWN MANKAD AND SHENGLI HU
OPERATIONS, TECHNOLOGY AND INFORMATION MANAGEMENT
CORNELL UNIVERSITY
ITHACA, NY 14850
E-MAIL: smankad@cornell.edu
E-MAIL: sh2264@cornell.edu

ANANDASIVAM GOPAL
DEPARTMENT OF DECISIONS, OPERATIONS & INFORMATION TECHNOLOGY
UNIVERSITY OF MARYLAND
COLLEGE PARK, MD 20910
E-MAIL: agopal@rhsmith.umd.edu