Crumbs of the Cookie: User Profiling in Customer-Base Analysis and Behavioral Targeting

Michael Trusov and Liye Ma

Robert H. Smith School of Business, University of Maryland

Zainab Jamal, HP Autonomy*

WEB APPENDIX

CONTENT:

* Michael Trusov (mtrusov@rhsmith.umd.edu, phone: (301) 405-5878, fax: (301) 405-0146) is Associate Professor, Robert H. Smith School of Business, University of Maryland, College Park, MD 20742; Liye Ma (liyema@rhsmith.umd.edu, phone: (301)405-8982) is Assistant Professor, Robert H. Smith School of Business, University of Maryland, College Park, MD 20742; Zainab Jamal is Research Scientist, HP Autonomy, Sunnyvale, CA 94089. The authors contributed equally and are listed in reverse alphabetical order.

Technical Appendix 1: Model Estimation

Our model is estimated using a hierarchical Bayesian approach with data augmentation. We use Markov Chain Monte Carlo (MCMC) for the estimation. We start with a general overview of the estimation approach followed by a detailed discussion of MCMC steps.

Estimation – Basic Steps

Step 1. Draw the consumer-specific role composition and visitation intensity parameters for each individual consumer $i, i = 1, \dots, I$:

$$\{\theta_i\}_{i=1,\dots,I} | \{\bar{\theta}\}, \Sigma, \{\vec{\rho}_r\}, \vec{\rho}_\lambda, \{\delta_{itr}\}, \{r_{itv}\}$$

In the formula above, r_{itv} is the role taken by consumer i in the v-th visit at time t. This is not observed, but is generated through data augmentation in the estimation, in step 7 below. This step has a computational complexity of $O(I * R * T)$, where $O(x)$ means the number of operations is bounded by x multiplied by a constant. In another word, the number of operations is of the same order of magnitude of $I * R * T$ or lower.

Step 2. Draw population level role composition parameters:

$$\{\bar{\theta}\}, \Sigma | \{\theta_i\}$$

This is a standard procedure for multivariate normal distribution and has a computational complexity of $O(I * R + R^2)$.

Step 3. Draw role-composition parameters of observed characteristics $\vec{\rho}_r$ for each role $r, r = 1, \dots, R-1$ and $\vec{\rho}_\lambda$ for the visit intensity:

$$\{\vec{\rho}_r\}_{r=1,\dots,R-1}, \vec{\rho}_\lambda | \{\theta_i\}, \{\delta_{itr}\}, \{r_{itv}\}$$

This step has a computational complexity of $O(I * T * R^2)$.

Step 4. Draw user- and time-specific role composition parameters δ_{itr} and $\delta_{it\lambda}$ for each individual consumer $i, i = 1, \dots, I$:

$$\{\delta_{itr}, \delta_{it\lambda}\}_{i=1,\dots,I, t=1,\dots,T, r=1,\dots R-1} | \{\theta_i\}, \{\vec{\rho}_r\}, \vec{\rho}_\lambda, \{\phi_r\}, \phi_\lambda, \{\sigma^2_{\epsilon r}\}, \sigma^2_{\epsilon\lambda}, \{r_{itv}\}, \{\xi_{tr}\}, \{\xi_{t\lambda}\}$$

This step has a computational complexity of $O(I * T * R)$.

Step 5. Draw the time-specific role composition parameter ξ_{tr}:

$$\{\xi_{tr}, \xi_{t\lambda}\}_{t=1,\dots,T, r=1,\dots R-1} | \{\delta_{itr}\}, \{\delta_{it\lambda}\}, \{\phi_r\}, \phi_\lambda, \{\sigma^2_{\epsilon r}\}, \sigma^2_{\epsilon\lambda}$$

This step has a computational complexity of $O(I * T * R + T * R^2)$.

Step 6. Draw the auto-regressive and the variance parameter of user- and time-specific role composition parameter ϕ_r for each role $r, r = 1, \dots, R-1$ and ϕ_λ for the visit intensity:

$$\{\phi_r, \sigma^2_{\epsilon r}\}_{r=1,\dots,R-1}, \phi_\lambda, \sigma^2_{\epsilon\lambda} | \{\delta_{itr}\}, \{\xi_{tr}\}$$

This step has a computational complexity of $O(I * T * R)$.

Step 7. Draw role-category probability parameter Φ_r, for each role $r, r = 1, \dots, R$:

$$\{\Phi_r\}_{r=1,\dots,R} | \{r_{itv}\}_{i=1,\dots,I; t=1,\dots,T, v=1,\dots,N_{it}}$$

This step has a computational complexity of $O(R * \sum_{i=1}^{I} \sum_{t=1}^{T} N_{it})$.

Step 8. Data-augmentation: draw the role of individual visit r_{itv} for each visit $v, v = 1, \dots, N_{it}$ of each consumer $i, i = 1, \dots, I$ at each time $t, t = 1, \dots, T$:

$$\{r_{itv}\}_{i=1,\dots,I;t=1,\dots,T,v=1,\dots,N_{it}} | \{\theta_i\}, \{\vec{\rho}_r\}, \vec{\rho}_\lambda, \{\Phi_r\}, \{\delta_{itr}\}$$

This step has a computational complexity of $O(R * \sum_{i=1}^{I} \sum_{t=1}^{T} N_{it})$.

MCMC implementation details

The parameters to be estimated are the population level mean and covariance for role composition and Internet usage intensity, $\bar{\theta}$ and Σ, the individual consumer level role composition and usage intensity parameter $\{\theta_i\}_{i=1,\dots,I}$, the role-composition parameters of observed characteristics $\{\vec{\rho}_r\}_{r=1,\dots,R-1}$ and that for visit intensity $\vec{\rho}_\lambda$, the user- and time-specific role composition parameters $\{\delta_{itr}, \delta_{it\lambda}\}_{i=1,\dots,I,t=1,\dots,T,r=1,\dots R-1}$, the time-specific role composition parameters $\{\xi_{tr}, \xi_{t\lambda}\}_{t=1,\dots,T,r=1,\dots R-1}$, the autoregressive and variance parameters $\{\phi_r, \sigma^2_{\epsilon r}\}_{r=1,\dots,R-1}, \phi_\lambda, \sigma^2_{\epsilon\lambda}$, and the role-category composition parameter $\{\Phi_r\}_{r=1,\dots,R}$. Where the context is clear, we re-write λ as R (where the parameter of R-th role is normalized to zero). We estimate the parameters through Markov Chain Monte Carlo, or MCMC, using a combination of Gibbs sampling with Metropolis-Hastings.

We augment the data to facilitate estimation. First, we rewrite a consumer's visitation profile as:

$$V_{it}^{E} = (c_{it1}, .., c_{itv}, .. c_{itN_{it}}) \quad \text{(A-1)}$$

In equation (A-1), c_{itv} is the category of the v-th visit of the consumer at time t, same as in equation (10) of the main body of the paper.

Next, we denote the vector of roles corresponding to the individual visits:

$$R_{it} = (r_{it1}, .., r_{itv}, .. r_{itN_{it}}) \quad \text{(A-2)}$$

In equation (A-2), r_{itv} is the role of the v-th visit of the consumer, same as in equation (9) of the main body of the paper.

Our MCMC procedure recursively generate draws as follows:

(1) For each consumer i, draw $\theta_i | \bar{\theta}, \Sigma, \{\vec{\rho}_r\}_{r=1,...,R}, \{R_{it}\}, \{N_{it}\}, \{\delta_{itr}\}$

The posterior of θ_i is:

$$\theta_i | \bar{\theta}, \Sigma, \{\vec{\rho}_r\}_{r=1,...,R}, \{\delta_{itr}\}, \{R_{it}\}, N_i \propto MVN(\theta_i; \bar{\theta}, \Sigma) \prod_{t=1}^{T} \left(d_{poisson}(N_{it}; \lambda_{it}) \prod_{v=1}^{N_{it}} p_{itr_{itv}} \right)$$

In the equation, $P_{it} = (p_{it1}, .. p_{itR})$ is calculated from θ_i and the other parameters as specified in equations (5) and (6a) in the main body, λ_{it} is calculated as specified in equation (6b) in the main body, and $MVN(.)$ is the density function of the multivariate normal distribution.

This posterior does not have a closed form, so we use Metropolis-Hastings algorithm to draw θ_i. The proposal is generated from random walk, and the step size is tuned to yield an acceptance rate of about 20%.

(2) Draw $\bar{\theta}, \Sigma | \{\theta_i\}_{i=1,...,I}$

We use diffuse conjugate normal prior $\Sigma_0 = 10000 I_R$ and $\theta_0 = (0)_R$ (where I_R is $R \times R$ identity matrix). Denote the mean of individual level parameter as: $\overline{\theta} = I^{-1} \sum_{i=1}^{I} \theta_i$.

Then the posterior of population mean follows a multivariate normal distribution:

$$\bar{\theta}|\Sigma,\{\theta_i\}_{i=1,\dots,I}\sim MVN((\Sigma_0^{-1}+I\Sigma^{-1})^{-1}\left(\Sigma_0^{-1}\theta_0+I\Sigma^{-1}\overline{\theta}\right),(\Sigma_0^{-1}+I\Sigma^{-1})^{-1})$$

Where MVN denotes multivariate normal distribution.

We use diffuse conjugate inverse Wishart distribution $\nu_0=0$ and $\Psi_0=10000I_R$. The posterior of the population variance follows an inverse Wishart distribution:

$$\Sigma|\bar{\theta},\{\theta_i\}_{i=1,\dots,I}\sim IW(\nu_0+I,\Psi_0^{-1}+\sum_{i=1}^{I}(\theta_i-\bar{\theta})(\theta_i-\bar{\theta})')$$

(3) Draw $\{\vec{\rho}_r\}_{r=1,\dots,R}|\{\theta_i\},\{R_{it}\},\{N_{it}\},\{\delta_{itr}\}$

The posterior of $\{\vec{\rho}_r\}_{r=1,\dots,R}$ is:

$$\{\vec{\rho}_r\}_{r=1,\dots,R}|\{\theta_i\},\{R_{it}\},\{N_{it}\},\{\delta_{itr}\}$$
$$\propto\prod_{r=1}^{R}\pi(\vec{\rho}_r)\prod_{i=1}^{I}\left(\prod_{t=1}^{T}\left(d_{poisson}(N_{it};\lambda_{it})\prod_{v=1}^{N_{it}}p_{itr_{itv}}\right)\right)$$

In the equation, P_{it} and λ_{it} are the same as in step (1) above. We use a diffuse normal prior. This posterior does not have a closed form, so we use the Metropolis-Hastings algorithm to draw $\{\vec{\rho}_r\}_{r=1,\dots,R}$. The proposal is generated from random walk, and the step size is tuned to yield an acceptance rate of about 20%.

(4) For each consumer i, Draw $\{\delta_{itr}\}_{t=2,\dots,T,r=1,\dots,R}|\theta_i,\{R_{it}\},\{N_{it}\},\{\vec{\rho}_r\},\{\xi_{tr}\},\{\phi_r\},\{\sigma_{\epsilon r}^2\}$

The posterior of $\{\delta_{itr}\}_{t=2,\dots,T,r=1,\dots,R}$ is:

$$\{\delta_{itr}\}_{t=2,\dots,T,r=1,\dots,R}|\{\theta_i\},\{R_{it}\},\{N_{it}\},\{\vec{\rho}_r\},\{\xi_{tr}\},\{\phi_r\},\{\sigma_{\epsilon r}^2\}$$
$$\propto\prod_{t=2}^{T}\left(\begin{array}{c}\left(\prod_{r=1}^{R}N(\delta_{itr}-\phi_r\delta_{it-1r};\xi_{tr},\sigma_{\epsilon r}^2)N(\delta_{it+1r}-\phi_r\delta_{itr};\xi_{t+1r},\sigma_{\epsilon r}^2)\right)\\ d_{poisson}(N_{it};\lambda_{it})\prod_{v=1}^{N_{it}}p_{itr_{itv}}\end{array}\right)$$

In the equation, P_{it} and λ_{it} are the same as in step (1) above. This posterior does not have a closed form, so we use the Metropolis-Hastings algorithm. The proposal is generated from random walk, and the step size is tuned to yield an acceptance rate of about 20%.

(5) Draw $\{\xi_{tr}\}_{t=2,\dots,T,r=1,\dots,R}|\{\delta_{itr}\},\{\phi_r\},\{\sigma^2_{\epsilon r}\}$

The step is the standard normal mean posterior draw just as in Step 2, with the individual values being $\delta_{itr} - \phi_r \delta_{it-1r}$.

(6) Draw $\{\phi_r, \sigma^2_{\epsilon r}\}_{r=1,\dots,R}|\{\xi_{tr}\},\{\delta_{itr}\},\{\sigma^2_{\epsilon r}\}$

The posterior of $\{\phi_r\}_{r=1,\dots,R}$ is:

$$\{\phi_r\}_{r=1,\dots,R}|\{\xi_{tr}\},\{\delta_{itr}\},\{\sigma^2_{\epsilon r}\}$$
$$\propto \prod_{r=1}^{R}\left(\pi(\phi_r)\prod_{i=1}^{I}\left(\prod_{t=2}^{T} N(\delta_{itr} - \phi_r \delta_{it-1r}; \xi_{tr}, \sigma^2_{\epsilon r})\right)\right)$$

We use a diffuse normal prior for ϕ_r, logit-transformed. The posterior does not have a closed form, so we use the Metropolis-Hastings algorithm. The proposal is generated from random walk, and the step size is tuned to yield an acceptance rate of about 20%.

For $\{\sigma^2_{\epsilon r}\}$, the step is the standard normal variance posterior draw just as in Step 2, with the individual values being $\delta_{itr} - \phi_r \delta_{it-1r}$.

(7) Draw $\{\Phi_r\}_{r=1,\dots,R}|\{R_{it}\}$

As stated in equation (13) the main text, we choose conjugate Dirichlet prior $Dir(\vec{\alpha})$ for Φ_r, where $\vec{\alpha} = (\alpha_1,..\alpha_C) = (1)_C$. The posterior for each t is:

$$\Phi_r|\{R_{it}\}_{i=1,\dots,I,t=1,\dots,T} \sim Dir(\alpha_1 + m_1, \dots, \alpha_C + m_C)$$

In the equation, $m_c = \sum_{i=1}^{I} \sum_{t=1}^{T} \sum_{v=1}^{N_{it}} I\{r_{itv} = r\} I\{c_{itv} = c\}$, where $I\{.\}$ is the indicator function. In another word, m_c is the number of occurrences of category c when the visitation is of the role r.

(8) Draw $\{r_{itv}\}_{i=1,\ldots,I;t=1,\ldots,T,v=1,\ldots,N_{it}} | \{\theta_i\}, \{\vec{\rho}_r\}, \{\delta_{itr}\}, \{\Phi_r\}$

We assume a uniform prior for the role of each individual visit r_{itv}. The posterior follows a categorical distribution:

$$r_{itv}\{\theta_i\}, \{\vec{\rho}_r\}, \{\delta_{itr}\}, \{\Phi_r\} \sim Categorical\left(\frac{p_{it1}\phi_{1c_{itv}}}{\sum p_{itr}\phi_{rc_{itv}}}, \ldots, \frac{p_{itR}\phi_{Rc_{itv}}}{\sum p_{itr}\phi_{rc_{itv}}}\right)$$

In the equation, $P_{it} = (p_{it1}, .. p_{tiT})$ is the same as in step (1) above.

Technical Appendix 2. Algorithm Scalability

In addition to being able to generate easily interpretable user profiles while admitting rich heterogeneity, our model is also well suited for large dataset, for two reasons. First, by mapping consumers to roles and roles to website categories, our model naturally achieves dimensionality reduction. As shown in the our estimation procedure, the amount of computation for most steps depends on the total number of roles R, instead of the total number of website categories C. While websites can be classified into many categories, the number of roles that represent typical consumer activities would be considerably smaller. Our model thus has considerably lower computational complexity than alternative methods that process multi-dimensional count data directly (e.g., Danaher and Smith 2011). Granted, this reduction in complexity comes at the cost of precision, similar to dimensionality reduction methods, but it also produces easily interpretable results. Furthermore, the number of roles can be chosen by the researcher. If the amount of computing resource is limited, for example, the researcher can choose a small number of roles to reveal the most prominent behavioral profiles. When more computing power is available, the researcher can increase the number of roles to get a more accurate picture. Our model thus also provides additional flexibility that existing methods do not have.

The second reason our model is suitable for large dataset is that the estimation algorithm can be easily adapted for parallel computing. Parallel computing is a powerful way to handle large datasets – when multiple computers process the data in parallel, total computation time can be shortened considerably. With the advent of cloud computing, resources needed for parallel computing, i.e. multiple CPUs, are now commonplace. The core of our proposed method is a topic model. Efficient estimation of topic models, both in a single processor scenario and in a distributed environment, has been a focus of research in Computer Science and Machine Learning, where large datasets with millions of documents have often been used for testing, and estimation of dataset as

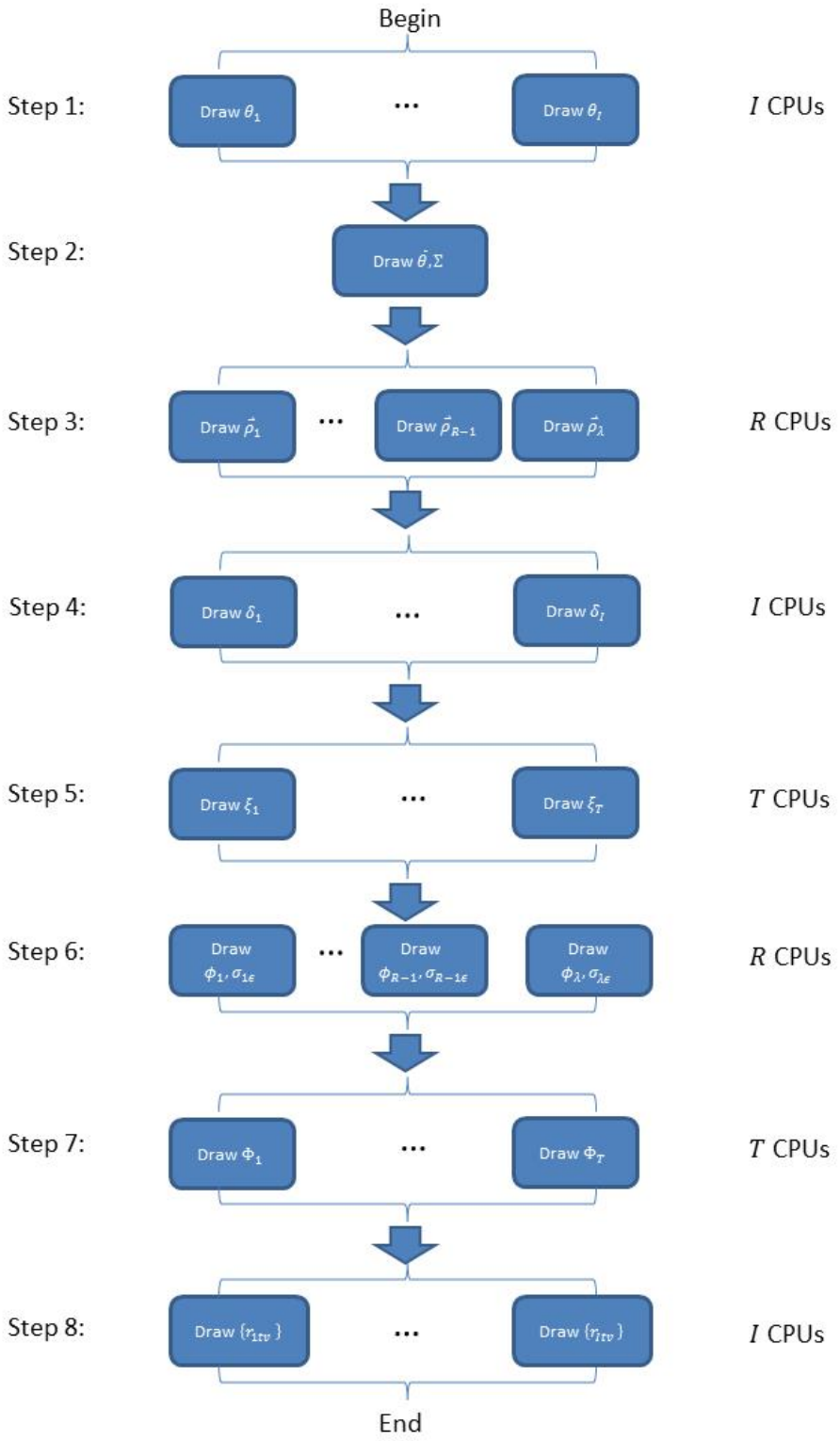
large as 20 million documents and 6 billion words have been shown to take only 12 hours (e.g. Griffiths and Steyvers 2004, Newman et al. 2009, Yao et al. 2009, Smola and Narayanamurthy 2010). The original LDA model can be estimated using MCMC in a straightforward manner, by iterating through draws of word-topic allocation, topic-word mapping, and topic probabilities. Griffiths and Steyvers (2004) develop a collapsed sampler which leads to faster mixing and improves speed of estimation. Yao et al. (2009) evaluate the performance of several alternative estimation methods, including MCMC, variational inference, and a classification based inference method. They also develop a SparseLDA algorithm, which further improves estimation speed using a "binning" approach. Newman et al. (2009) and Smola and Narayanamurthy (2010) both develop algorithms for estimating topic models in a parallel environment.

The original MCMC method for estimating LDA models, iterating through draws of word-topic allocation, topic-word mapping, and topic probabilities, can be easily done in a distributed manner. This is because computation at individual document level, the word-topic allocation, is independent from other documents, and can be done on separate computers concurrently (Newman et al. 2009, Smola and Narayanamurthy 2010). The other two steps are comparatively light, and can also be performed to a large extent in parallel (word counts can be conducted on each computer independently first, and quickly aggregated across computers in the end). Also, much attention has been paid to extending the collapsed sampler or other algorithms specifically designed for LDA to a distributed environment, often through approximation (Newman et al. 2009). Although our model is more complex than LDA, it shares the same characteristics that most of the computation work is at individual consumer level (corresponding to individual document in the classic topic model), e.g. the visitation to role mapping, the consumer's role composition, and the fluctuation of role composition over time. Conditional on the hyper-parameters, these parameters can be drawn independently from those of other consumers. Hence they can be estimated concurrently on multiple computers, similar

to LDA. In fact, our model falls into the general framework of Latent Variable Models as discussed in Ahmed et al. (2012), which demonstrates how this entire category of models can be efficiently estimated in parallel using, for example, Hadoop. Existing literature, therefore, suggests that our model can be efficiently estimated in parallel. To further substantiate this, we develop a detailed estimation plan of our model for a parallel computing environment, which is shown in Figure TA2.1. As the figure shows, except for the second step which is computationally light, all other steps of estimation can be done in parallel.

In step 1, drawing the role-composition parameter of each consumer is a computational task independent from that of other consumers. The parameter for each consumer can thus be drawn independently from each other. Should I CPUs be available and the parameter draw for one consumer takes k milliseconds, the entire step can be finished in k milliseconds instead of $k * I$ milliseconds if done sequentially, a linear reduction in computation time. Similarly, steps 3 to step 8 can each be done in parallel on I, R, or T CPUs. Only step 2 is not obviously parallelizable. However, step 2 is a standard step for multivariate normal distribution which takes little time even when performed sequentially. Our estimation algorithm thus can be parallelized in a way that it achieves close to linear reduction in computation time for up to T or R CPUs, and can achieve considerable further reduction in time for more CPUs, up to I CPUs (the number of consumers is expected to be much larger than the number of roles).[1] This property makes our model a good fit for handling large sets of data.

[1] The algorithm can actually be further parallelized to achieve linear reduction in time beyond T CPUs – the unit task for each role in steps 3 and 4 can be further partitioned by consumers. More details are available from authors by request.



Notation: "I CPU" reads as parallelizable up to *I* CPUs

Figure TA2.1: Estimation on Parallel Computers.

In addition to the above discussion, we also verify the scalability of our estimation algorithm empirically. Since we do not have enterprise level parallel computing resources, we demonstrate this on a smaller scale as a proof-of-concept. We implemented the parallel estimation algorithm in a multi-threaded fashion on a server with 10 physical cores, and evaluated the runtime by varying the number of threads used for estimation. Meanwhile, we also evaluated the runtime by varying sample size, to ensure the runtime does not go up faster than the sample size.

Number of Threads	Execution Time*
1	26.75
2	12.90
4	7.10
6	5.13
8	4.23
10	3.46

*Execution time in seconds, per MCMC draw iteration

Performance measured on a Dell Poweredge T620 server with 10 physical cores

Table TA2.1: Estimation Scalability by Number of Threads

Number of Households	Execution Time*
4000	3.05
8000	5.27
16000	10.24
24000	14.56
32000	19.25
40000	23.89
45300	26.75

*Execution time in seconds, per MCMC draw iteration

Performance measured on a Dell Poweredge T620 server with 10 physical cores, using a single thread

Table TA2.2: Estimation Scalability by Sample Size

The result of this scalability testing is reported in Tables TA2.1 and TA2.2. Table TA2.1 reports the amount of time used for taking one iteration of MCMC draw according to the numbers of threads (each thread runs on a separate CPU core) used for estimation. As the number of threads increases, the runtime decreases consistently and significantly. When 10 threads are used for estimation, the runtime is only about 1/8 of that when only one thread is used. In another word, the runtime is scaling up almost linearly. Meanwhile, Table TA2.2 shows that as the sample size increases, the runtime increases but in a slightly sub-linear fashion. When 40,000 households are used, the runtime is less than 10 times needed for 4,000 household. Both demonstrate the good scalability of our model. We further note that these performance numbers are obtained from a personal server, and the performance can be even better on computing resources available to large-scale enterprises.

References

A. Ahmed, M. Aly, J. Gonzalez, S. Narayanamurthy, and A. Smola, "Scalable inference in latent variable models," in WSDM, 2012, pp. 123–132.

P.J. Danaher and M. Smith, "Modeling multivariate distributions using copulas: Applications in marketing," Marketing Science, 30.1, 4-21, 2011.

T. Griffiths and M. Steyvers, "Finding scientific topics," In Proceedings of the National Academy of Sciences, volume 101, pages 5228–5235, 2004.

D. Newman , A. Asuncion, P. Smyth, M. Welling, " Distributed Algorithms for Topic Models," The Journal of Machine Learning Research, 10, p.1801-1828, 2009.

A. J. Smola and S. Narayanamurthy, "An architecture for parallel topic models," International Conference on Very Large Databases, vol. 3, 2010.

L. Yao, D. Mimno, and A. McCallum, "Efficient methods for topic model inference on streaming document collections," in Knowledge Discovery and Data Mining, 2009.

Technical Appendix 3. Comparison with Cluster Analysis

Consumer profile of multi-dimensional data is often done using cluster analysis. Cluster analysis is a standard classification method for uncovering latent types of mixtures of data. It is a popular approach for consumer profiling, where consumers are assigned to different clusters which represent different behavioral types, so that predictions can be made based on the general characteristics of these types. We use cluster analysis as a benchmark to evaluate our model performance. Specifically, we employed K-Means clustering method with Manhattan distance. We performed the clustering both with and without Principal Component Analysis (PCA). For the former, clustering is performed based on the first 10 principal components of the visit profiles, which account for 97% of the total variation. For the latter, clustering is performed on the visit profiles directly. Since the website visit profile is at consumer-month level, we also performed the clustering analysis at both person-month level and at person level. At person-month level[2], the same consumer's visit profiles in different months can belong to different clusters. At person level, the same consumer's visit profiles in different month are constrained to belong to the same cluster. The former, with more degrees of freedom, is expected to have better in-sample fit. However, since the cluster is not at consumer level, out-of-sample fit cannot be evaluated. The latter, in comparison, should have worse in-sample fit, but out-of-sample prediction is possible (the mean of the cluster a consumer is assigned to). Similar to evaluating our proposed model, for person level clustering we use the January to November data for in-sample analysis and use the December data for out-of-sample prediction.

[2] We also estimated the model using weekly and biweekly data, and the results remain largely similar.

	Clustering With PCA			Clustering Without PCA		
	Method 1	Method 2		Method 1	Method 2	
Number of Clusters	MAPE In-Sample	MAPE In-Sample	MAPE Out-Of-Sample	MAPE In-Sample	MAPE In-Sample	MAPE Out-Of-Sample
5	91.61%	115.61%	136.96%	100.13%	119.30%	141.61%
10	80.82%	102.95%	124.97%	90.22%	100.23%	129.18%
20	72.39%	100.63%	121.85%	81.33%	99.04%	128.19%
30	68.59%	97.30%	122.34%	76.04%	97.61%	128.57%
40	65.97%	95.50%	118.91%	72.78%	98.63%	128.26%
50	64.77%	95.54%	121.89%	70.15%	97.87%	128.42%
60	62.49%	95.05%	118.38%	66.30%	97.69%	129.26%
70	61.32%	96.14%	119.73%	65.49%	97.36%	127.82%
80	59.64%	96.41%	123.02%	63.20%	97.99%	127.58%
90	59.34%	96.01%	120.22%	61.76%	97.56%	126.75%
100	58.44%	95.64%	121.11%	60.32%	97.21%	126.38%

Method 1: Cluster at person-month level
Method 2: Cluster at person level

Table TA3.1: Model Fit – K-means Cluster Analysis

The in-sample and out-of-sample model fit of these cluster analyses are reported in Table TA3.1. Clustering with and without PCA have similar sample fits, with the former slightly better than the latter. For all clustering methods, increasing the number of clusters leads to better fit. However, for person-month level clustering the improvement tapers off after about 60 clusters, and for person level clustering, after about 30 clusters. Clustering at person-month level clearly has better in-sample fit than at person level, although it is not able to give out-of-sample predictions. More importantly, our proposed model has significantly better performance than all these clustering methods (see Table 4 in the paper). For example, both the in-sample and out-of-sample errors of the 12-role estimates are about 50% lower than the in-sample and out-of-sample errors of the clustering methods, even when the latter use 100 clusters. This clearly demonstrates the advantage of the proposed model.

Technical Appendix 4: Predicting Profile Composition with Google Trends Data

In the main text we discuss the results of the two consumer specific demographics variables, age and income. Our model can more generally accommodate a wide range of person-, time-, or person-time- specific covariates. Such capacity is important because firms are expected to possess rich data which they can include in the model to further improve user profiling. For example, a retailer would have consumers' evolving purchase histories which can help predict the roles played by those consumers. To demonstrate that our model can also accommodate time specific explanatory variables (equation 6a), we introduce additional time varying data collected from Google Trends.

Category	Component 1	Component 2	Component 3
Arts & Entertainment	-0.050	-0.050	-0.067
Autos & Vehicles	0.104	-0.196	0.106
Beauty & Fitness	-0.001	-0.209	0.067
Books & Literature	0.166	0.093	-0.267
Business & Industrial	0.116	-0.015	-0.001
Computers Electronics	-0.010	-0.015	-0.057
Finance	0.155	-0.092	-0.076
Food & Drink	-0.442	-0.045	-0.003
Games	-0.079	-0.127	-0.171
Health	0.153	-0.036	0.012
Hobbies & Leisure	-0.172	-0.061	-0.150
Home & Garden	-0.036	-0.167	0.037
Internet & Telecom	-0.079	0.060	0.024
Jobs & Education	0.283	0.129	-0.087
Law & Government	0.126	0.092	0.060
News	0.086	0.681	0.504
Online Communities	-0.448	0.310	-0.050
People & Society	0.082	0.021	-0.029
Pets Animals	-0.031	-0.114	-0.016
Real Estate	0.153	-0.289	0.178
Reference	0.167	0.097	-0.117
Science	0.371	0.204	-0.297
Shopping	-0.389	0.108	-0.238
Sports	-0.070	-0.160	0.584
Travel	0.031	-0.270	0.214

Table TA4.1: Loadings of Top 3 Principal Components on Google Trend Categories

Specifically, we collected Trends' data for all the 25 top-level categories listed at the Google Trend website. We then performed a principal component analysis (PCA) on the data, and extracted the first 3 principal components as the covariates in X_{it}. These three components and their loadings on the 25 categories are reported in Table TA4.1. These three components explain 73.13% of variance in the original Google Trend data.

The coefficient estimates of these three covariates in Equation (6a) are reported in Table TA4.2 below. As the results show, the three components have explanatory power on the role compositions. For example, when the first component is more prominent, which has a somewhat business focus, users are less likely to play the entertainer and family person roles. When the second component is more prominent, which has a somewhat information focus, users are more likely to play the information seeker and online shopper roles.

	Component1	Component2	Component3
Role1	-1.036	0.162	-0.474
	[-1.139, -0.899]	[0.148, 0.180]	[-0.527, -0.395]
Role2	0.559	0.207	0.609
	[0.543, 0.579]	[0.181, 0.229]	[0.583, 0.630]
Role3	-0.078	0.167	0.028
	[-0.112, -0.029]	[0.156, 0.179]	[0.002, 0.062]
Role4	2.077	-0.091	1.095
	[1.951, 2.176]	[-0.135, -0.051]	1.028, 1.135]
Role5	-1.499	-0.504	-0.989
	[-1.587, -1.370]	[-0.589, -0.391]	[-1.078, -0.891]
Role6	0.271	0.104	-0.405
	[0.247, 0.292]	[0.094, 0.116]	[-0.465, -0.347]
Visit Intensity	0.091	0.315	0.19
	[0.075, 0.101]	[0.285, 0.339]	[0.167, 0.211]

Numbers in the parentheses are 95% credible intervals

Table TA4.2: Parameter Estimates – Effect of Google Trends on Role Composition

Granted, this analysis is used only to illustrate our model's ability to incorporate time-specific variables, as components produced through PCA may not be clearly interpretable. Still, the analysis shows that our proposed model can yield more insightful interpretation when richer data is available. We leave the detailed investigation of this aspect for future study.