Predicting Consumption Patterns with Repeated and Novel Events

Dimitrios Kotzias, Moshe Lichman, and Padhraic Smyth, *Member, IEEE*

Abstract—There are numerous contexts where individuals typically consume a few items from a large selection of possible items. Examples include purchasing products, listening to music, visiting locations in physical or virtual environments, and so on. There has been significant prior work in such contexts on developing predictive modeling techniques for recommending new items to individuals, often using techniques such as matrix factorization. There are many situations, however, where making predictions for both previously-consumed and new items for an individual is important, rather than just recommending new items. We investigate this problem and find that widely-used matrix factorization methods are limited in their ability to capture important details in historical behavior, resulting in relatively low predictive accuracy for these types of problems. As an alternative we propose an interpretable and scalable mixture model framework that balances individual preferences in terms of exploration and exploitation. We evaluate our model in terms of accuracy in user consumption predictions using several real-world datasets, including location data, social media data, and music listening data. Experimental results show that the mixture model approach is systematically more accurate and more efficient for these problems compared to a variety of state-of-the-art matrix factorization methods.

Index Terms—Personalization, Repeat Consumption, Mixture Models, User Modeling.

1 Introduction

Applications in which an individual can interact with a large set of items are commonplace, including interacting with social media, navigating e-commerce websites, streaming of video data, and more. Automated generation of item recommendations has become an essential component of the user experience in many such applications [1], enabling an individual to explore a large set of options without being overwhelmed.

In this general context, individual consumption patterns can broadly be divided into two types of behavior. In the first type, consumption of items is driven by *novelty*: users will typically select *new* items with which they have *not* interacted in the past. Examples of such behavior, where consumption is dominated by non-repeat behavior, include movie viewing, purchasing books, and reading online newspaper articles, etc. [2], [3]. Datasets exhibiting this pattern have been the primary focus for much of the predictive user modeling work in machine learning and data mining research in recent years (e.g., [4], [5], [6], [7]). However, in the second broad behavior type, individual consumption patterns are characterized by a *mixture of both repeated and novel items*, often with more emphasis on the repeated aspect. Examples include listening to music artists or songs, visiting physical locations, using apps on mobile phones, or purchasing groceries. In many such situations users tend to have very high repeat consumption rates. For example, analyzing data from the Reddit and LastFM websites (data sets discussed later in the paper) reveals that between 60 to 70% of a user's activity, on average, involves repeat consumption, compared to close to zero for typical datasets used in published evaluations of recommender systems [5].

The combination of novel and repeat consumption patterns can be viewed as an example of the well-known exploration-exploitation phenomenon that has been studied from multiple different perspectives, including psychology (e.g., curiosity arousal [8]), economics (e.g., variety seeking behavior [9] and brand choice [10], [11]), human-computer interaction [12], and reinforcement leaning [1]. Recent work in recommender systems has identified the value of learning users' exploration preferences, suggesting novelty, serendipity and diversity as desirable attributes for sets of recommended items. Focusing only on accuracy can lead to homogeneous recommendations and eventually to user disengagement [4], [13], [14]. More broadly, understanding both the exploitation and exploration aspects of consumption behavior is important from the perspective of personalization and human-computer interaction (e.g., [12], [15]), providing a basis for accurate and responsive experiences at an individual level. As one example, knowing which apps are likely to run on mobile phone or the possible locations a user will visit can result in better battery life of mobile or ubiquitous devices [16].

The balance between exploration and exploitation can be quite heterogeneous across different individuals. This heterogeneity can be driven by individual characteristics such as willingness to take risks and aversion of boredom [9], [10]. Figure 1 shows the rate of *new* consumptions over time for five users. For two of the users (shaded in green) the consumption patterns are relatively stable with little new exploration. Other users, such as those in blue, select new items continuously month after month. From a modeling perspective this means that a model needs to strike a different balance across users between historical behavior and new exploration. From a predictive modeling

- *All authors are associated with the Department of Computer Science, at the University of California Irvine.*
  E-mail: [dkotzias, mlichman, psmyth] @ics.uci.edu
- *Moshe Lichman is also associated with Google.*

Manuscript received December 31, 2017; revised March 26 , 2018.

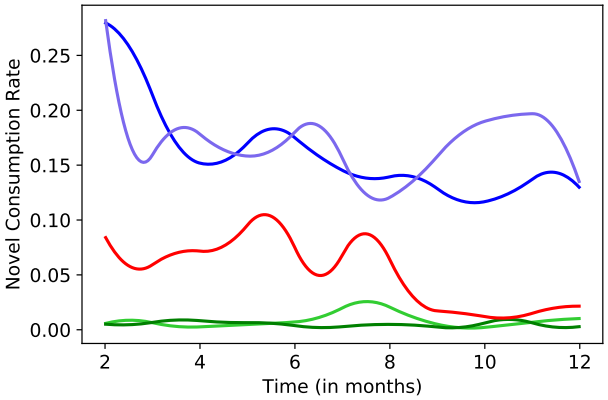


Fig. 1. Smoothed estimates over time of the fraction of items consumed per month that are new (relative to a user's past behavior) for 5 different users. Novel consumption rates (or exploration propensities) vary significantly by user (high in blue, low in green) and can change over time (red user). Data was taken from the Reddit datasets. The set of previously-consumed items for each user for each month is defined cumulatively from the historical data for that user.

perspective a key point is that there is not necessarily any natural notion of "steady-state" or optimal behavior in this context. A user is not necessarily going to converge to a "true" or "optimal" set of items that they will then continue to select forever. Real human consumption patterns evolve over time, and consist of a combination of both repeated and novel events.

This paper focuses on modeling users in data sets where individual choices display this mix of *both repeat and novel item consumption*. This topic has been relatively unexplored in the data mining and machine learning literature (with a few exceptions, which we discuss in more detail later). The most widely used approaches for modeling user-item consumption data in recent years are based on matrix factorization (MF) (e.g., [3], [17]). These methods represent each user and item by a latent vector of size k where (typically) k is much smaller than the number of users or items. The motivation is to find k hidden features and represent users and items as a linear combination of these features, e.g., by movie genre. This has been found to work well for discovery of new items, by leveraging dependencies among columns to generalize about previously unseen entries in the matrix.

However, as we demonstrate later in the paper, for data with significant repeated content, MF techniques are unable to accurately capture detailed aspects of repeat behavior at the individual level. This is driven in some sense by the inherent (lossy) nature of the low-dimensional representation. A more in-depth analysis is presented in Section 3.

Instead, for the problem of consumption prediction and user modeling we propose to model such data through a two-component mixture of multinomials, with one component for repeat (exploitation) behavior and one for novel (exploration) behavior. While multinomial mixture models have been widely and successfully used across a broad number of applications in the past, the specific contribution of this work is in their application to consumption data and studying their efficacy. The mixture approach results in a user model that allows for personalization at two levels: (1) at the item-selection level (by considering a user-specific distribution of item consumption), and (2) at the level of balancing an individual's mix of repeated and novel behavior (through individualized mixture weights). Our primary goal in this paper is to study and illustrate the effectiveness of a relatively simple and scalable model that can learn a user's propensity to explore.

Towards that goal, we investigate real world data in which user choices are focused on a relatively small subset of the overall choice set. We analyze several datasets from different domains that are characterized by different levels of repeat consumption. Our datasets include location data from location-based social media, selection of music artists for music streaming applications, and modeling of participation in online communities.

We find that the mixture approach outperforms all baselines in predictive accuracy. Across seven datasets we show that the method is flexible enough to be used in a range of settings and different levels of repeat consumption. We also show how this approach is scalable, robust to noise, and can readily be extended to incorporate external information.

2 Problem Definition and Data Set Characteristics

2.1 Problem Definition

Assume we have a set of U users and M items, and $n_{uj} \in \{0, 1, 2, \ldots\}$ corresponds to the number of times user u has consumed item j in the past, e.g., the number of times that user u purchased product j or that u listened to artist j. We can represent this data as a $U \times M$ matrix of counts $\mathbf{N}$ which is typically very sparse.

We assume a scenario where the data are collected and aggregated over some time period T (e.g., one year of data), and our goal is to generalize from the historical data and make predictions for some time period $(T, T + t]$, where t is some window of, e.g., corresponding to the next week or month. The goal is to estimate for each user u

$$\theta_u = [\theta_{u1}, \ldots, \theta_{uM}], \quad \sum_j \theta_{uj} = 1$$

where θ_{uj} is the probability that user u consumes item j conditioned on the event that u consumes some item, i.e., a predictive multinomial distribution over M possible items.

2.2 General Characteristics of Repeat and New Consumption Data

To provide some general context for the problem we take a look in Table 1 at characteristics of count matrices $\mathbf{N}$ for several different real-world user-item datasets that exhibit both repeat and novel consumption behavior. The first three data columns show the size, the total number of non-zero entries, and the density (second column divided by the first) for each data set. The overall choice set is very large (ranging from 7k to 20k items). Users typically only consume a very small fraction of these items, which is the driving force behind the the matrix sparsity. The 4th column, n, shows the mean count of the non-zero elements in each matrix, indicating the average number of times that items are consumed in each data set conditioned on the fact that they are consumed at least once. Another way to interpret this number is to see that $n - 1$ is the average number

TABLE 1
General characteristics of dataset matrices: matrix size (total users and items), number of non-zero entries, matrix density, and average number of repeat consumptions per item n.

	Size: $U \times M$	# Entries	Density	n
redditS	20k x 21k	416k	0.10%	13.2
redditT	113k x 21k	7M	0.29%	41.2
lastfm	992 x 15k	547k	3.86%	27.5
goNYloc	1k x 7k	43k	0.61%	1.5
goSFloc	2k x 7k	71k	0.51%	1.6
twOCloc	13k x 11k	94k	0.07%	3.9
twNYloc	30k x 11k	242k	0.07%	2.3

TABLE 2
Mean consumption and repeat consumption characteristics across datasets (for the training data). The datasets vary significantly in terms of the rates of user consumption and repeat percentages.

	Unique items per user (average)	User-item pairs that are repeats
redditS	20.8	61.1%
redditT	61.9	71.4%
lastfm	547.0	69.5%
goNYloc	43.0	16.3%
goSFloc	35.5	18.3%
twOCloc	7.2	40.4%
twNYloc	8.1	40.7%

of *repeat consumptions* per item. There is high variability in these mean repeat consumption rates across the data sets: in digital environments (reddit and lastfm) there are relatively high repeat rates while in the physical location environments (tw and go) the repeat rates are much lower. Below we provide additional details and context for each dataset.

2.3 Data Sets with Repeat and Novel Consumption

2.3.1 Reddit Posts

Reddit is a social network/link aggregator, which has more than 15 million unique monthly users.[1] The website allows users to post links and text, in topic-focused sub-categories (subreddits) resulting in discussions via nested replies (comments). Currently there are 70M comments (30Gb) per month on the website making the amount of text per month be larger than all of Wikipedia.

In this work we focused on posts from 2015 and 2016 and worked with subreddits with at least 1000 subscribers, leading to M 21; 386 subreddits (items).[2] Users with less than 10 posts were filtered out. Each count n_{uj} corresponds to the number of times user u posted a comment in subreddit j. From this set of users we extracted two datasets: reddit sample (redditS) and reddit top (redditT). redditS refers to a random sample of $U = 20;024$ users while redditT refers to $U = 113;557$ users who posted more than 1000 times in 2015. We use Jan to Dec 2015 as training data, Jan to Feb 2016 for validation (parameter tuning), and Mar to April 2016 for testing.[3]

2.3.2 Music Data

We use a publicly available dataset consisting of the listening history of $U = 992$ users from lastfm.com.[4] The items in this case are artists, and we use those with more than 100 songs, leading to $M = 15668$ items. We use counts from the years 2006, 2007, and 2008 as training data, the first three months of 2009 for validation, and the next three months as test data. Each count n_{uj} corresponds to the number of times user u listened to artist j.

2.3.3 Location Data: Twitter

We use two Twitter datasets collected from geo-located tweets (latitude, longitude) from two different areas, Orange County and New York, which we will refer to as twOCloc, twNYloc. The data was collected between May 2015 and February 2016. We discretized the lat/lon geolocations to a set M locations (items) in the same manner described in [18], using reverse geocoding based on publicly-available geoparcel records. We filter out users that did not have tweet events on at least 5 different days and filtered out locations with less than 3 events. We used the first eight months as training, the next month for validation and the next month for test. The counts n_{uj} correspond to how many times a user u tweeted at a specific location j.

2.3.4 Location Data: Gowalla

The Gowalla data set[5] was introduced in [19] and covers San Francisco and New York—these data sets will be referred to respectively as goSFloc and goNYloc. We used the same filters (on user and item counts) as for the Twitter data and used September 2009 through June 2010 for training, the next two months for validation, and the following two months for testing. This dataset comes from *check-ins* at popular locations so the locations are already discretized. The counts n_{uj} correspond to how many times a user u checked-in at a specific location j.

2.4 Characteristics of Repeat Consumption (Exploitation)

We explore the characteristics of repeat consumption for each dataset (training data) in Table 2. The first data column contains the mean number of unique items per user, providing a sense of the density at the user level. All of the data sets are highly sparse at the user level as well, with lastfm dataset being the most dense, since it spans a longer timeframe (3 years for lastfm versus a few months for the other datasets). The second column computes what percentage of the unique items per user consist of repeats, i.e., have counts $n_{uj} > 1$. These numbers are very high for the reddit and lastFM data sets (in the range of 60 to 70%), in the medium range for the Twitter data sets (around 40%), and lower for the Gowalla data sets (16 to 18%).

1. https://www.reddit.com/r/AskReddit/about/traffic/
2. Data from http://redditmetrics.com/top, retrieved in 2016.
3. Publicly available at https://archive.ics.uci.edu/ml/datasets/Repeat+Consumption+Matrices
4. http://www.dtic.upf.edu/ ocelma/MusicRecommendationDataset/lastfm-1K.html
5. Publicly available at https://snap.stanford.edu/data/loc-gowalla.html

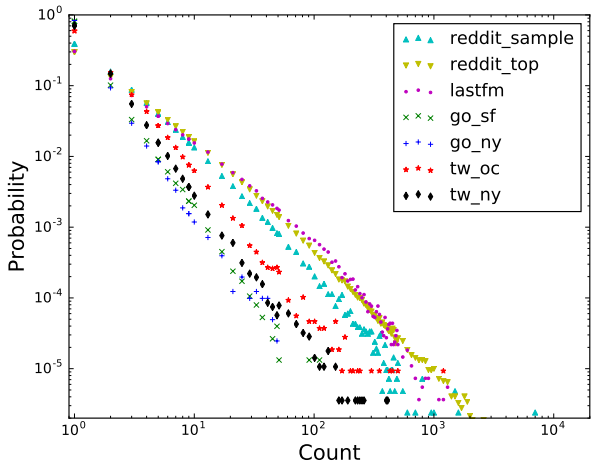


Fig. 2. Empirical probability that any cell in the user-item count matrix is a specific count. Across all datasets the counts tend to follow a heavy-tailed distribution. (Best viewed in color).

These percentages are a key differentiator between repeat consumption data and data sets such as movie reviews or book purchases (which tend to have repeat consumption rates close to 0, e.g., [5]).

All of the datasets also have a non-trivial percentage of high counts, which is a key aspect of repeat consumption behavior. Figure 2 illustrates this phenomenon by plotting the relative frequency of particular counts in each data set on a log-log scale. Each of the datasets has a heavy-tailed distribution with considerable probability mass assigned to counts greater than one. The plots also show that the counts for the digital environments (Reddit and LastFM) are systematically and significantly higher than the counts for the location data sets (in the physical world, Twitter and Gowalla). In one sense this could be viewed as surprising, as one might expect more exploration behavior (i.e., more low counts spread across more items) in digital environments that require less resources (time and energy) to explore.

However, as mentioned before, online datasets have much higher *total* counts, as events are easier to generate. For example users tend to listen to many more songs than visit locations. Another mitigating factor is that there are various constraints on resource consumption in the digital world as well as in the physical world. For Reddit, for example, online communities have their own behavioral norms, and while there is no direct monetary cost to posting, writing a successful comment often requires time in order to adapt to its norms [20].

2.5 Characteristics of New Consumption (Exploration)

To explore characteristics of new item consumption, in Table 3 we analyze the percentage of item pairs that are new relative to the total number of item pairs. A user-item combination is considered **new** if it has a count of 0 in the training $\mathbf{N}$ matrix and has a count greater than 0 in the test data. The first column in Table 3 shows the percentage of new user-item pairs, relative to all non-zero pairs in the test data, for each data set. The percentage ranges from 20% to 73% across the data sets, with higher rates of exploration (new items) for the physical location data sets compared to the digital data sets. The rate of new item consumption is relatively high in general across the data sets. Thus, an important aspect of the user modeling problem is how well a predictive model can generalize to new items beyond items that an individual consumed in the past.

TABLE 3
Exploration rates: percentages of new items and events per user, in the test data, across data sets. There is significant variation across datasets in terms of how much exploration occurs per user.

	User-item pairs that are new in test data	User-item events that are new in test data
redditS	29.8%	11.7%
redditT	20.0%	5.7%
lastfm	21.4%	15.3%
goNYloc	73.2%	62.7%
goSFloc	67.3%	55.0%
twOCloc	45.6%	22.4%
twNYloc	63.5%	42.1%

The second column in Table 3 shows the percentage of **new events**, where events are defined as each individual user-item interaction, i.e., a count of 10 would correspond to 10 events. This results in a re-weighting towards actual user-item interactions rather than unique user-item pairs and amplifies the difference between the digital data sets (which have relatively high repeat counts) and the location data sets (which spread more of their counts towards new interactions). In general, across the datasets, the weighted percentages for events (column 2) are significantly lower than for user-item pairs (column 1), indicating that *on average users embrace change gradually*. They might explore a new item, but they will on average choose it *less often* than a familiar option.

Differences do not only exist across datasets, but also *across users* within the same dataset. Two important differentiating factors are (1) users' individual exploration rates, and (2) their overall consumption rates. The exploration rate is defined for each user as (a) the number of events corresponding to new items for a user divided by (b) the total number of events for the user, where both numbers are computed on the test set. Figure 3 shows the exploration rate as a function of the number of observed events per user, for the redditS dataset (similar plots were observed for the other data sets). Users are placed into equal-sized buckets based on the number of training data points, and each box shows the median exploration rate (orange line) and inter-quartile range.

The plot illustrates that the median tendency for exploration per user tends to decrease systematically as a function of the total number of consumptions per user (x-axis). This is not surprising given that we would expect users who have consumed fewer items to be more inclined to explore new items—and conversely users who have consumed many items tend have fewer options for exploration and tend to explore less. However, as was illustrated in Figure 1, this pattern is not universal across users. The relatively high variance in exploration rates (y-axis) across the boxplots

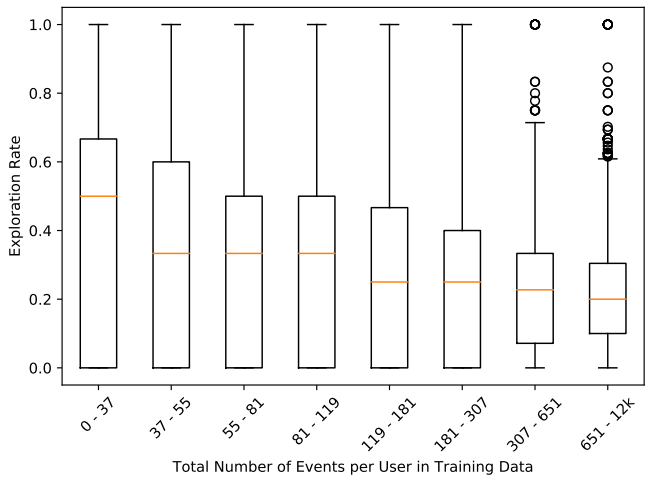


Fig. 3. For each user of the redditS dataset, the y-axis is the fraction of events (posts) in new subreddits in the test data per user, as a function of the number of total events a user has in the training data (x-axis). Exploration rates are clearly user dependent and many users, including those with large amounts of data, do not necessarily converge to a steady-state distribution for their choices (i.e., their exploration rate remains high).

indicates that at all levels of consumption there is considerable heterogeneity in user behavior, with exploration rates ranging from close to 0 to close to 1 across different users. In other words, collecting more observations about a user does not necessarily lead to convergence to a steady-state preference distribution for the user (with exploration rates close to zero). This observation is in agreement with research that suggests that the propensity to explore is an intrinsic user characteristic [9], [10], indicating the need for personalization in environments where exploration is a common feature of user behavior.

3 Matrix Factorization and OverSmoothing

Matrix factorization (MF) is perhaps the most widely-used approach over the past decade for modeling of sparse user-item consumption data sets [3], [7]. Below we summarize the modeling assumptions behind MF and then highlight its primary shortcoming in the context of data that contains both repeat and new items, namely, MF's tendency to significantly oversmooth beyond the historical observations.

3.1 Matrix Factorization Review

Given a $U \times M$ matrix $\mathbf{N}$ with entries $n_{uj} \in \{0;1;2;:::g$ (i.e. counts), the goal of MF is to produce an approximation $\mathbf{N} \approx \mathbf{N}^0 = \mathbf{W}\mathbf{H}$ by estimating matrices $\mathbf{W}$ and $\mathbf{H}$, which are of dimensionality $U \times k$ and $k \times M$ respectively. Typically k is significantly smaller than M (and U) (i.e., a reduced rank approximation) and each row vector in $\mathbf{N}$ is modeled as a linear combination of k M-dimensional basis vectors.

MF can be viewed from a probabilistic perspective by assuming the counts are drawn from some distribution whose mean is determined by the $\mathbf{W}$ and $\mathbf{H}$ matrices. Many MF approaches assume observations with Gaussian noise: $n_{uj} \sim N(\mu = \mathbf{w}_u\mathbf{h}_j; \sigma_0^2)$ where σ_0^2 is some fixed variance.

TABLE 4
Comparison across all data sets of the actual fraction of new items per user compared to the probability mass assigned to new items by NMF. NMF consistently over-estimates how much probability should be assigned to new items, i.e., the likelihood of new items being consumed.

	Fraction of new items	NMF 20	NMF 50	NMF 100
redditS	23.2%	66.0%	60.7%	54.4%
redditT	7.5%	53.1%	47.3%	40.2%
lastfm	30.3%	70.6%	68.4%	61.8%
goSFloc	65.9%	93.5%	90.5%	87.9%
goNYloc	69.5%	94.9%	91.7%	87.9%
twOCloc	35.8%	67.6%	61.8%	62.9%
twNYloc	55.1%	88.2%	82.6%	73.3%
Amount of Oversmoothing		33.6%	29.2%	24.2%

However, for count data a more appropriate model is to use a distribution over non-negative integers, such as a Poisson distribution. This leads naturally to Poisson matrix factorization (PF) [17]: $n_{uj} \sim Poisson(\mu = \mathbf{w}_u\mathbf{h}_j)$. A non-probabilistic alternative, which retains the non-negative constraint, is *non-negative matrix factorization* (NMF), where all elements of $\mathbf{W}$ and $\mathbf{H}$ are constrained to be non-negative, in turn ensuring that the reconstructed matrix $\mathbf{N}^0$ contains only non-negative entries. We investigate both PF and NMF in the experimental results section later in the paper.

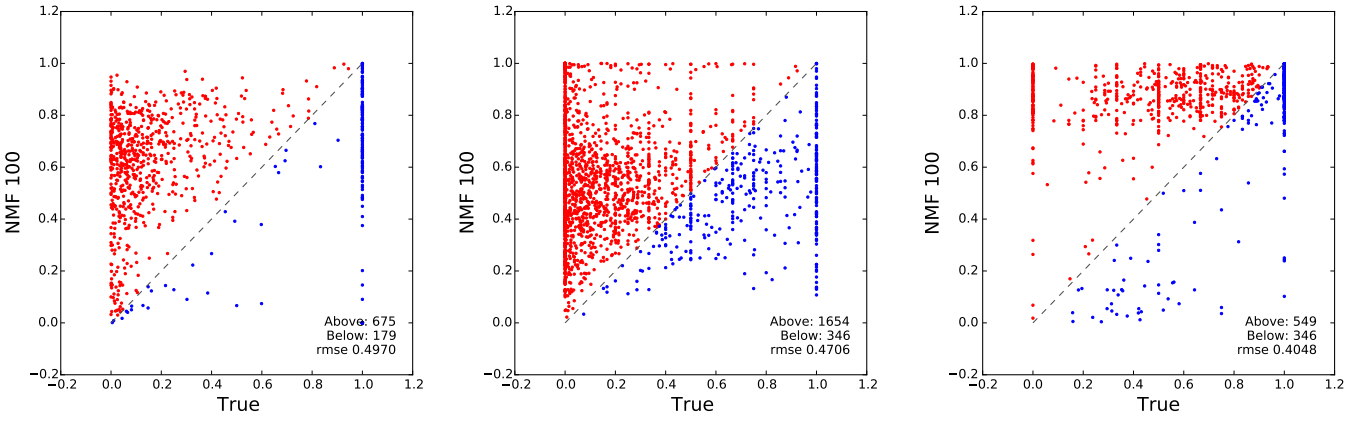
In this paper we use MF techniques such as PF and NMF to generate approximations of observed count matrices over some historical period T. We can view the reconstruction $\mathbf{N}^0$ as a matrix of *expected counts*: the low-rank representation forces the reconstruction to generalize beyond the training observation window T. For prediction purposes we can normalize these counts per user to produce a multinomial probability distribution for each row (user), e.g., for NMF, P_u^{NMF} with entries $P_{uj}^{NMF} = n_{uj}^0 / \sum_l n_{ul}^0$ that sum to 1, where n_{uj}^0 is the reconstructed (expected) count for user u and item j from NMF. These normalized probability estimates are the basis for our discussion of MF in this section and in the experimental results later in the paper.

3.2 Systematic Oversmoothing in Matrix Factorization

The strength of MF techniques lies in their ability to leverage patterns of dependence among columns and to find low-dimensional basis functions that can provide useful approximations of low-order moments of the original matrix, e.g., row/column means and row/column covariances.

However, this also implies that the representation cannot accurately model all rows of the matrix (i.e., the counts patterns for specific users) in detail. The low-dimensional bottleneck inevitably results in a *lossy* compression. For data sets with significant repeat consumption this can be a significant drawback since the compression can cause the model to forget details of an individual's history.

We ran NMF on each of the data sets and evaluated how much probability mass it places on new items relative to repeated items. We measured $\frac{1}{N} \sum_u \sum_{j:n_{uj}=0} P_{uj}^{NMF}$, i.e., the average probability mass per user placed on items that



(a) lastfm (b) redditS (sample of 2000 users) (c) goSFloc

Fig. 4. Scatter plots of the assigned probability mass of NMF with 100 components, versus the ground truth, for the lastfm, redditS (sample of 2000 users) and goSFloc datasets. NMF overestimates the probability of new items (oversmooths) for considerably more users (red dots) than it underestimates (blue dots).

do not occur in the training data. In Table 4 we compare these numbers with the actual fraction of new items observed in the test data, for NMF with $k = 20; 50; 100$. We see that NMF oversmooths by a large margin in all cases. It assigns significantly more probability mass to unseen items in the training data compared to the actual fraction of *new* items that occur in the test data, suggesting that the model is miscalibrated (when used for prediction) in terms of how many new items each user will select. Because of this oversmoothing, NMF will have difficuilty modeling users exhibiting high repeat rates (such as those shaded in green in Figure 1). As the dimensionality k increases, the calibration gradually improves (as expected), suggesting that for very large k this oversmoothing would be less of an issue—but using very large values for k runs counter to MF's goal of generalizing through a low-rank representation.

Figure 4 contains scatter plots showing, *per user*, how much probability was assigned by NMF compared to the true fraction of *new* items, for three of our datasets – one of each type (similar results were obtained for all datasets). Points far away from the diagonal correspond to significant miscalibration. The plots also have the number of users above and below the diagonal, as well as the root mean squared error (RMSE) between true and observed, for each dataset. The model overestimates the probability of new items for many more users than it underestimates, by significantly oversmoothing, on all datasets. In fact very few users were predicted to have very few or no new items, which should not be the case: there are a significant number of individuals with no (or almost no) new items in their test data (see the $x = 0$ point in each plot in Figure 4).

Home-Work Simulation. A simple example of the importance of reconstruction can be derived from our location datasets. Many users have a home and work location, from where they create the majority of their datapoints. A model that captures this should have a spike in probability for each users' (rows') home and work location, even though this location is largely different for every user. The smoothing tendency of MF methods however tends to discourage such *bursty* behavior.

To illustrate this point, we generate an artificial dataset with similar properties. We create a $U \quad L$ matrix simulating an extreme case of users with home and work locations. There are $L = 1000$ locations $h\%$ of which are considered home and the rest are work. We experiment with different values of U and h. Each user selects a home and work location at random and adds one datapoint in each. While some home and work locations do coincide, this effect is small, as there are only 5 times more users than locations at most, and the choice is done uniformly. We then factorize the matrix with k components into W and H and normalize each row of the resulting factorized matrix $fm = W \quad H$ to sum to one. We then measure the total probability assigned to each users' home and work locations (combined) in the matrix fm, which should be close to 1 for an ideal model.

Figure 5 illustrates how much probability mass was assigned by NMF to the home and work locations combined, for various $U; h$ for a number of components. While 200 components constitute 20% of the rows, which is an unusually high number of components, the model still cannot model this behavior with more than 65% accuracy.

4 Mixture Model Approaches

As stated in Section 2 our goal is to estimate for each user u, a multinomial probability distribution over all items, representing the probability with which user u will select each item. As a simple alternative to the latent space decomposition methods, we consider the use of multinomial-based mixture models. A two-component mixture representation provides a simple way for a model to combine both (a) details about an individual's history of consumption with (b) broader global population preferences. In the multinomial mixture approach one of the components can memorize the details of an individual's past consumption patterns, while the other component can use broader population patterns to generalize beyond the user's observed data. The mixture weights of the two components reflect an individual's relative propensity for repeat versus novel item consumption.

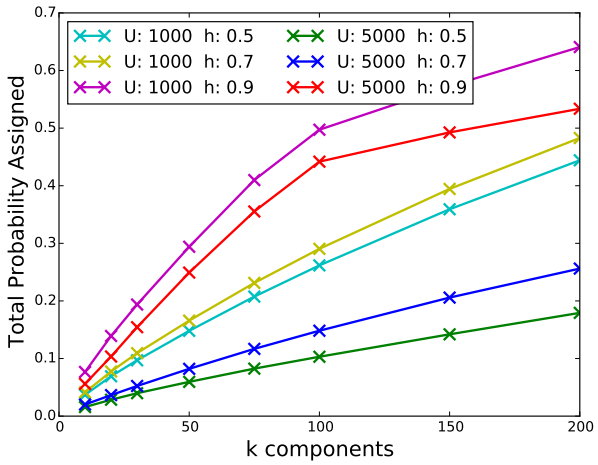


Fig. 5. Probability mass that NMF puts on home and work locations for simulated data with various values of k. The probability mass should ideally be close to 1, but NMF's tendency to oversmooth inhibits such behavior.

4.1 Maximum Likelihood and Bayesian Estimation

To motivate the mixture model approach we first review maximum likelihood and Bayesian estimation of multinomial parameters. A common approach for estimating $\hat{}_u$ is via maximum likelihood estimates (MLEs). For multinomial models the MLEs are the relative frequencies from the observed data:

$$\hat{}^{MLE}_{uj} = \frac{n_{uj}}{n_u} \; ; \qquad n_u = \sum_{j'} n_{uj'}$$

where n_u is the total number of items consumed by u. This estimate $\hat{}^{MLE}_{uj}$ captures perfectly the user's historical data but it is not as useful in a predictive sense since it assigns probability mass zero to any previously unseen items.

In order to generalize beyond a users previously consumed items to items that are new for that user, one approach is to smooth the probability mass by adopting a Bayesian perspective where we view $_u$ as a random variable with a prior, and infer a posterior distribution for $_u$ given the data and the prior. A natural choice is a conjugate prior for the multinomial model, i.e., the Dirichlet prior with parameters $= [\,_1; \ldots; \,_M]$, which results in a posterior Dirichlet distribution on $_u$. The mean of this posterior distribution, for each parameter u and items j, $\hat{}^{MPE}_{uj}$, is:

$$\hat{}^{MPE}_{uj} = \frac{n_{uj} + \,_j}{n_u + \sum_j \,_j}$$

where MPE refers to the mean posterior estimate.

For sparse data this posterior estimate is more useful than the MLE as can generalize by assigning non-zero probability mass to events that have not occurred in the past (observed) data, i.e, to items that a user has not yet consumed. For example, if $n_{uj} = 0$ for some user u and item j, we still get a non-zero estimate of the probability of that combination occurring in the future due to the presence of $_j$ in the equation above. Note that $= \sum_j \,_j$ plays the role of the "strength" of the prior above since it acts as a pseudocount in the denominator relative to the actual total count n_u for user u.

In the context of our problem, with multiple users u, we use an informative prior for the $_j$ values, e.g., one that reflects population preferences. We use a simple "global" or population prior where we assume that each $_j$ is proportional to $n_j = \sum_u n_{uj}$ (the popularity of item j across all users), corresponding to an empirical Bayes approach. (In practice we make the $_j$ proportional to $n_j + 1$ to avoid issues with items that have zero count in a training data set). We can thus write the MPE as

$$\hat{}^{MPE}_{uj} = \frac{n_{uj} + \,_j}{n_u + } \tag{1}$$

where $_j = \frac{n_j + 1}{n + M}$, $n = \sum_j n_j$, and the scalar > 0 controls the degree of smoothing. Since we are interested in the predictive capabilities of our models, is treated as a global hyperparameter that can be optimized. We determine its value via grid search on validation data.

4.2 Learning Predictive Mixture Weights

It is informative to rewrite Equation 1 above in the form of a mixture model:

$$\hat{}^{MPE}_{uj} = \,_u \frac{n_{uj}}{n_u} + (1 \quad \,_u)\frac{\,_j}{} \; ; \qquad \,_u = \frac{n_u}{n_u + } \tag{2}$$

This form illustrates that the first component in this mixture is the frequency-based MLE (from the user's history) and the second component models the global population pattern of item usage. This mixture-based view of the MPE provides another way to see how Bayesian smoothing works in this context, namely as a weighted convex combination of a user's historical data and a smoother population pattern.

It is important to note that in the MPE approach, the mixture weight $_u = \frac{n_u}{n_u + }$ in Equation 2 is entirely a function of the total amount of data n_u for user u and the strength of the prior . For global, fixed , as we accumulate more data for a user u, this model places more emphasis on the user's history (i.e., repeat behavior) and less on new items (i.e., novel behavior).

This exposes a significant limitation of this smoothing approach, namely that the balance between **exploit** and **explore**, for a user, is *a function only of the number of observed points in the training data* rather than reflecting any intrinsic explore-exploit characteristics of the individual. The mismatch arises because so far we have implicitly assumed that we are trying to learn a static distribution $_u$ for each user. However, our analysis from Section 2 shows that this assumption does not match the reality of real-world data sets in the sense that users are constantly exploring new items over time (e.g., see Figures 1 and 3). Thus, to a large extent there is no natural notion of a steady-state distribution for $_u$ and the exploration rate of a user does not solely depend on the number of observed data points.

To address this we propose using the general mixture form of the MPE in Equation 2 but learn the mixture weights on an individual basis to reflect the explore/exploit characteristics of individual users u. For convenience of notation we refer to the two multinomial components as

Individual (^{I}) to reflect repeated behavior, and *Population* (^{P}) to reflect novel consumption:

$$ {}^{I}_{uj} = \frac{n_{uj}}{n_u} \; ; \qquad {}^{P}_{j} = \frac{n_j + 1}{n + M} $$

The probability of user u consuming item j is computed as:

$$ P_{uj} = \; _u \; {}^{I}_{uj} + (1 \quad _u) \; {}^{P}_{j} \tag{3} $$

where $_u$ is a mixture parameter that we learn. We estimate the component parameters, ${}^{I}_{uj}$ (per individual) and ${}^{P}_{j}$ (global), using the training data.

The model's ability to weight each component allows it to represent the trade-off between **exploitation** (^{I} component) and **exploration** (^{P} component) on a per user basis. This model also has a simple and intuitive generative interpretation. If a user wishes to choose an item for consumption, she will flip a biased coin. With probability $_u$ she will engage in exploit behavior and sample an item from her "repeat" component ${}^{I}_{u}$ (recall that all items that were not consumed before have probability 0 in this component). With probability $1 \quad _u$ she makes a selection that she has not made before in accordance with the global item popularity.

4.3 Parameter Estimation

The learning process of our method is split into two steps, learning the parameters for the components and then learning the weights. We enable personalization in our model by learning different mixing weights $_u$ for each user on a disjoint validation set. However, since not all users have an adequate amount of validation data, learning these weights independently is prone to overfitting. To overcome this problem we share information across users in the form of another empirical Bayesian prior: we first calculate a global population mixing weight and then use the value of that as a prior. We model the $_u$'s with a binomial distribution using a Beta prior $B(\,^{I};\,^{P})$ and calculate the MPE. The strength of the Beta prior can be interpreted as pseudocounts—we set it to be proportional to the average number of points in the validation set, specifically

$$ B(\,^{I};\,^{P}) = B(\quad n;(1 \quad) \quad n) $$

where n is the average number of points per user in the validation set.

The mixing weight $_u$ represents the propensity of user u to repeat their choices. While on the one hand we want the model to tailor this number to each individual, on the other hand the prior provides an effective shield against overfitting. The prior allows active users (with a lot more data than average) to tend towards their own personal exploitation rate, while users that have considerably less validation datapoints than the population remain close to the global weight. It also provides a solution for users that have no data in the validation set; their mixing weight is set to the population value, $_u =$.

We use the Expectation-Maximization (EM) algorithm to learn the global and individual mixing weights. Since the component multinomials are fixed during learning of the weights, the algorithm tends to converge quickly in just a few iterations. A more detailed description of the E and M steps can be found in the Appendix.

Setting the novelty component in the mixture to be proportional to global population preferences is a relatively simple approach for modeling the exploration of new items by users. One could also incorporate into this framework more informed component models, for both the repeated and novel consumption components, using ideas such as those presented in [4], [21], [22]. For example, a natural modification of our model is to make the second component more tuned to each individual u, e.g., by smoothing towards data from a set of individuals that are in some sense similar to u. We explore the idea of more components in Section 5.5, where we introduce additional components for the model to smooth towards, based on row-smoothing and column-smoothing in a manner similar to collaborative filtering.

5 Experiments and Results

5.1 Evaluation Metrics

We use two evaluation metrics in our experiments. The first metric is the log loss i.e., the negative of the average log-probability of each event (e.g., user-item consumption) in the test data:

$$ \log P = \quad \frac{1}{N_{te}} \sum_{u} \sum_{j} n_{uj} \log P(j|u) \tag{4} $$

where n_{uj} here corresponds to counts in the test data set. This metric is widely used in the evaluation of machine learning algorithms that produce probabilistic predictions (e.g., [23], [24], [25], [26]). A model that assigns higher probability, or lower negative log-probability, to the observed test data is preferred over a model that assigns lower probability (or a higher negative log-probability).

In terms of a baseline, using the same population probabilities (i.e., the marginal probabilities of each item across all users) to make predictions for each user u will get a score roughly equivalent to the entropy of the data. If users consumed only one item, and a model assigned probability 1 to that item, the log-loss would be 0. However, since users typically consume multiple items, the lower bound for the log-loss is obtained by using the MLE of the test data for prediction for each user. Making perfect predictions (i.e., assigning probability proportional to the number of times a user selected each item in the test set) achieves this lower bound. Thus, the log-loss can be thought of as measuring the predictive ability of a model in terms of compression (of future data) with a natural scale between the entropy of the global model as an upper bound and the log-loss of the MLE of the test data as a lower bound.

The second metric Recall@k is defined as

$$ \text{Recall@}k = \frac{1}{N_{te}} \sum_{u} \sum_{j} \frac{n_{uj} \, \mathbb{I}(rank(u;j) \quad k)}{\sum_{j'} n_{uj'}} \tag{5} $$

which evaluates the ability of the model to assign high rank to observed items in the test data set for each user. For a given model, for each user u, the predicted probabilities over the M items are sorted in descending order, and Recall@k measures what fraction of items (that u actually

consumed in the test data set) were ranked in the top k by the probability model for u. Higher scores are better for recall and the global model (using the same ranking for everyone, based on global popularity) can be viewed as a lower-bound baseline. Note that Recall@k is less fine-grained than log-probability: it only focuses on the top-K ranked items and only analyzes the order of these items, not if their probabilities are calibrated correctly.

5.2 Models Evaluated

We compare the mixture-based approach to a set of general-purpose recommender system methods that model user-item interactions through latent factors [3]. These approaches are frequently used in modeling of large sparse arrays of user-item interaction data. For all methods we used a disjoint validation set to optimize hyper-parameters for each method and report the best performance results on a separate held-out test dataset.

Global: This is a baseline global multinomial model with probabilities proportional to the global popularity n_j of each item.

NMF: Non-negative matrix factorization based on the work of [27], [28]. In our evaluation we used an off-the-shelf implementation available via scikit-learn.[6]

HBPF: Hierarchical Bayes Poisson factorization [17]. This method predicts user-item consumption counts using the Poisson distribution. Hierarchical priors are set on both the user and item latent factors in order to take item general popularity and user general activity into accounts, making this method suitable for skewed and sparse data.

LDA: Latent Dirichlet Allocation [29]. A Bayesian probabilistic model that learns multinomial latent factors via a generative process. LDA can be viewed as a general-purpose matrix factorization method for count data [30].

MPE: The MPE-based smoothed multinomial as defined in Equation 2. The population preference is used as a prior, and its strength is optimized using the validation set.

Mixture: The proposed predictive mixture model as defined in Equation 3 with personalized mixture weights.

For the MF methods we evaluated different numbers of latent factors k from 50 to 500 and invariably found that the largest number performed best, so all results below are for $k = 500$.

The latent factor methods based on matrix factorization (NMF, HBPF) do not directly produce multinomial probabilities of the user preferences, $\hat{}_u$. The Poisson based method HBPF produces estimates of $_{uj}$, the mean rate (counts per unit time) at which user u consumes item j in the training data. The NMF method produces an estimate n^0_{uj} of the expected count for user u consuming item j given the model and given the training data. We convert these rates and expected counts to multinomial probabilities by normalizing each user's rates/counts over items as follows:

$$_{uj} = \; _{uj} = \sum_{j'} \; _{uj'} \quad \text{or} \quad _{uj} = n^0_{uj} = \sum_{j'} n^0_{uj'} \tag{6}$$

In order to avoid zero probabilities when computing the log-loss in the predictive distribution, we add a small constant to each count before normalizing each row. The value of this constant is optimized on the validation data for each MF model, to make each one as competitive as possible in terms of prediction.

6. http://scikit-learn.org

TABLE 5
Average negative log probability (log-loss) on the test data for different algorithms across different data sets. Lower scores are better. Best-performing methods indicated in bold font. LB is the lower bound for optimal predictions per data set.

	Global	NMF	HBPF	LDA	MPE	Mixture	LB
redditS	6.89	4.09	6.39	4.32	3.39	**3.37**	1.70
redditT	6.70	3.82	6.36	3.99	3.07	**3.04**	1.97
lastfm	8.53	6.63	8.02	6.76	6.43	**6.35**	4.09
goSFloc	8.19	7.56	9.05	7.76	7.66	**7.26**	2.88
goNYloc	8.53	8.08	9.15	8.21	8.71	**7.90**	3.34
twOCloc	7.02	4.39	9.57	4.37	3.50	**3.42**	0.71
twNYloc	7.29	5.84	9.46	6.08	**4.73**	4.74	0.72

5.3 Prediction Accuracy

Table 5 contains the results for all methods and all datasets for log loss, as well as the theoretical lower bound (LB), corresponding to optimal probability distributions for each test data set, to provide a sense of scale of each problem. We can see that MF methods (NMF, HBPF) are significantly outperformed by the competing methods and on some data sets are worse in some cases (e.g., for HBPF) than the global model. This is a direct consequence of the oversmoothing discussed earlier in Section 3.2. LDA is in some cases more accurate than the MF methods but can be significantly worse than the MPE and predictive mixture models on some datasets. The predictive mixture is the best-performing model on 6 of the 7 of the datasets, but on several datasets the MPE model is very close and is slightly better on one dataset. These differences between MPE and the predictive model are not significant on most of the data sets, indicating that the main ingredient for higher accuracy on this prediction task (relative to MF methods and LDA) is the incorporation of a memory-specific component that can represent the details of the specific items that a user has consumed in the past. However, as discussed earlier, given a choice of MPE versus the predictive mixture, we would favor the latter since it has the ability to learn mixing weights that reflect true explore/exploit characteristics at the individual user level rather than weights that are just functions of how much data has been observed for that user.

As discussed earlier, we observed that matrix factorization techniques improve in accuracy as the number of latent factors is increased. However such increases in the number of latent factors results in the complexity and time being increased a significant factor, which is potentially problematic for scalability to large real-world problems.

TABLE 6
Average Recall@100 on the test data for different algorithms across different data sets. Higher scores are better. Best-performing methods indicated in bold font.

	Global	NMF	HBPF	LDA	MPE	Mixture
redditS	0.38	0.58	0.69	0.73	**0.82**	**0.82**
redditT	0.45	0.72	0.84	0.81	**0.92**	**0.92**
lastfm	0.18	0.33	0.34	0.40	0.49	**0.49**
goSFloc	0.27	0.30	0.23	0.37	0.42	**0.44**
goNYloc	0.22	0.22	0.17	0.30	0.37	**0.38**
twOCloc	0.43	0.49	0.57	0.67	0.74	**0.76**
twNYloc	0.32	0.35	0.42	0.50	0.61	**0.62**

TABLE 7
Average values of exploration probability (over all users) for the mixture model (1-), for MPE, and for NMF with a large number of latent factors.

	True	Mixture	MPE	NMF 100	NMF 500
redditS	23.2%	15.0%	36.5%	54.4%	40.72%
redditT	7.5%	7.5%	2.9%	40.2%	24.13%
lastfm	30.3%	24.0%	12.5%	62.9%	39.18%
goSFloc	65.9%	65.9%	72.8%	84.0%	67.58%
goNYloc	69.5%	66.2%	78.6%	82.8%	59.43%
twOCloc	35.8%	32.9%	71.8%	63.2%	56.36%
twNYloc	55.1%	59.2%	77.0%	71.9%	60.18%
MAE	-	3.54%	15.7%	24.59%	11.49%

Table 6 shows the average Recall@100 for all methods across all datasets. The general trends are the same as in the table of log-loss scores. The mixture and MPE methods are very close in performance and are significantly better than LDA across all data sets. In turn, LDA is consistently better than the MF methods across all data sets. The recall results show that the performance improvements of the mixture models (predictive and MPE) are not due to the quality of probability estimation alone, but also are evident when performing standard ranking tasks.

To provide a visual representation of how the different models are making predictions at the individual user level, Figure 6 shows the top 20 categories for a randomly-selected user from the redditS dataset. The barplots indicate the probability mass assigned by different modeling methods, including, from the top, the maximum likelihood component (relative frequencies in the training data), the global component, the predictive mixture model, the NMF 100, NMF 200 and NMF 500 models, and the relative frequencies observed in the test data. We see that the test data frequencies (bottom) are quite different to the training data frequencies (top). The plots also show that the NMF models tend to systematically assign too little probability mass to the repeated items from the training data, and tend to predict distributions that follow the global frequency of items rather than reflecting the specific item-consumption patterns of the user.

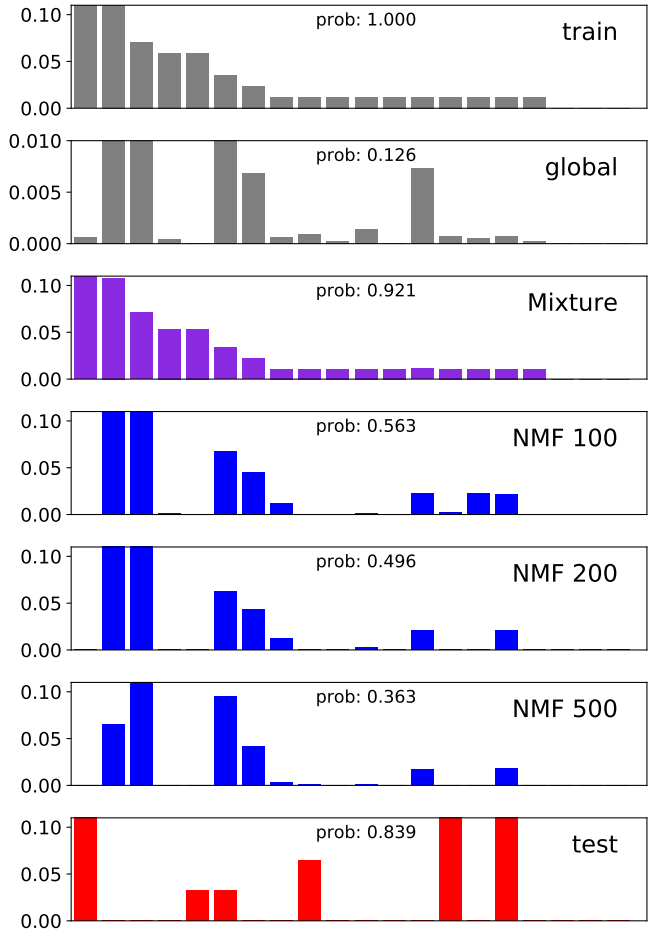
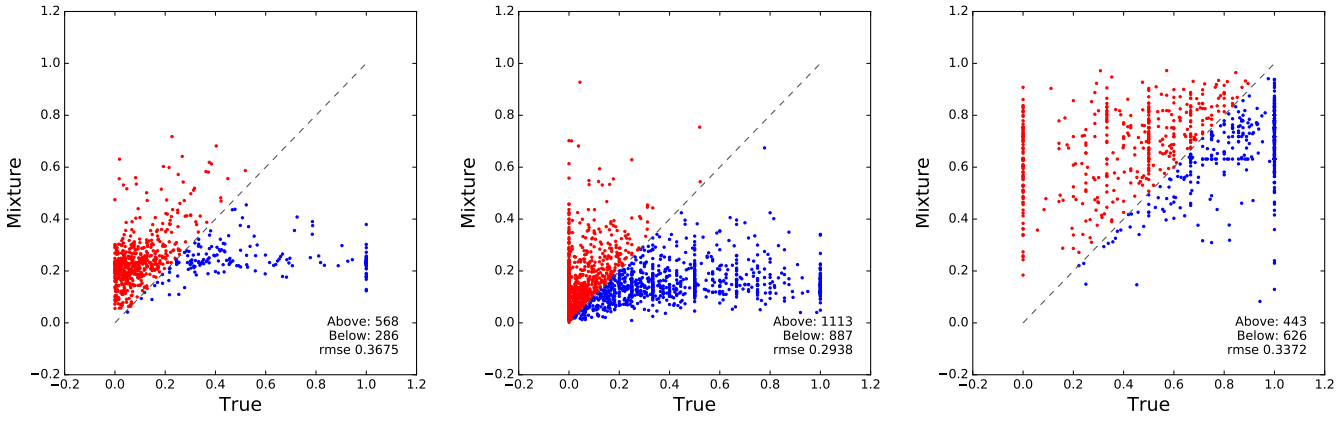


Fig. 6. A user from redditS (upper plot). The x-axis corresponds to the 20 most frequent items in the training data for each user, ordered by frequency from left to right. The different bar plots correspond to relative frequencies in the training data on top and in the test data on the bottom, with the probabilities assigned (to the selected items in the test data) by different methods (see text for details). The number displayed in each bar plot is the total (sum) probability mass (or relative frequency) assigned by each method to the displayed top 20 training items. For illustration purposes the y-axis has been cut-off to only show probabilities up to 0.1 (0.01 for the global case).

5.4 Evaluating Learned Mixing Weights

For all datasets, the first column of Table 7 shows the true percentage of new user-item entries in the test data (same shown in Table 4). The second column ("Mixture") contains the average value (over all users) of the exploration rate 1 , i.e., the mixture weight for the population component. This component represents the amount of probability mass that the mixture model assigned in aggregate to new items. The results show that the mixture model is much better calibrated than MPE and NMF in that it is assigning probability mass to new items that is relatively close to the true fraction of new items. MPE assigns too little probability in exploration in datasets where each user has a lot of data points (such as redditSand lastfm) and too much in the opposite case. NMF, even with a very large number of components, is significantly oversmoothing, as discussed earlier in Section 3.2.

Analyzing these values at the *user level*, Figure 7 shows scatter plots between the true fraction of new items per user



(a) lastfm (b) redditS (sample of 2000 users) (c) goSFloc

Fig. 7. Scatter plots of the assigned probability mass for the mixture model, versus ground truth, for the lastfm, redditS (sample of 2000 users) and goSFloc datasets. Comparing with Figure 4, the mixture model is more accurate overall in estimating exploration propensity and more balanced in terms of its errors.

and the mixing weight 1 $_u$ per user. This is a similar plot to Figure 4 where NMF predictions were compared to the true fraction of new items for the same datasets. Comparing both it is clear that the mixture model provides better calibrated results in the sense that its predictions are more balanced between over- and under-estimation and the root mean squared error is significantly lower for the mixture model on all of the datasets. Similar results were obtained for the other datasets but are omitted for brevity.

Referring back to the users whose Reddit data was shown in Figure 1, the model assigned an exploration weight of 14% and 17% for the blue shaded users and 1% and 2% for the green ones; values that are indicative of their behavior. In contrast MPE assigned an exploration weight of 2-3% for the blue users and 1% for the green, significantly underestimating the exploration tendency of the blue group.

The learned mixing weights represent a quantification of *user behavior* in terms of the preference of an individual to select items based on their personal history or to select new items based on global popularity. Learning this preference for each individual user can in principle enable more accurate personalized services. In addition, the weights are interpretable, allowing for the identification of specific data points in the validation set associated with the inference of individual weights.

5.5 Using Additional Components

Another useful aspect of the mixture model approach is that additional components can be included to potentially improve predictive power. Rather than relying only on a population component to generalize a user's model towards global population patterns, we can also generalize towards more specific subsets of users and items, e.g., that are similar to a particular user u and the items that u has consumed in the past. For example, following the social recommender paradigm [3], [31], if a friendship graph is available, we can use a component that predicts items for u based on items selected by u's social circle. As another example, for location data, the distance between locations [18] can be informative and a good proxy of similarity between locations.

When these types of external sources of information are unavailable, the model can be extended by adding implicit similarity information from the user-item consumption matrix. To illustrate this we calculated an N N similarity matrix $\mathbf{S}_{row}$ and M M column similarity matrix $\mathbf{S}_{col}$, both using cosine similarity. These matrices can be thought of as containing neighborhood information for rows and columns and act as a proxy to identify similar users and categories, a technique that is common in recommender systems based on collaborative filtering [3]. To create additional components for each user we multiply $\mathbf{S}_{row}$ by the training count matrix $\mathbf{N}$ and normalize each row, resulting in a matrix that essentially indicates which items a user u might choose, based proportionally on users similar to u. Likewise we multiply $\mathbf{N}$ by $\mathbf{S}_{col}$ to get a matrix of counts that are smoothed towards category similarity.

We then add the two new components and use EM to learn the weights as before. We found that there was a small (order of .01 to .02) but systematic improvement in log-loss and recall across all of the data sets. While these improvements are small, they illustrate the potential of the method, and when higher quality extrinsic information is available, larger gains can be expected (e.g., see [18] for an application in location modeling).

To assess the potential sensitivity of our model to overfitting when more components are added, we also added a mixture component consisting of random noise terms and again learned mixture weights via EM on validation data. EM consistently assigned weights in the range of 10^{-4} to 10^{-8} to these components across data sets and the prediction performance of the model was unchanged, providing evidence that the procedure is relatively robust to noise and can effectively ignore pure noise components. In general of course, as more components are added and with limited validation data there is the potential for any model to overfit in terms of estimating the weights, and for generalization performance to be affected negatively.

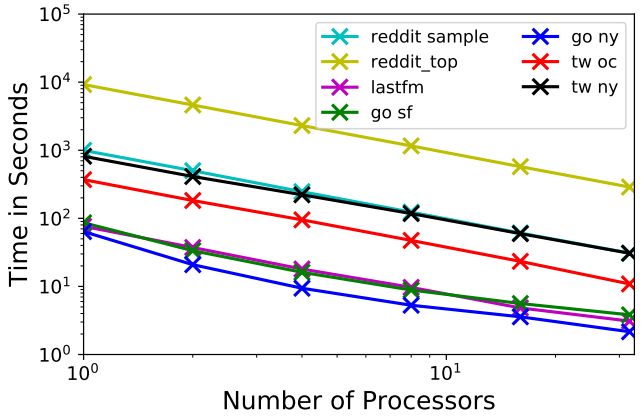


Fig. 8. Time in seconds for training the predictive mixture model as a function of the number of processors, across all data sets, on a log-log scale. Our proposed approach improves roughly linearly as a function of the number of processors added.

5.6 Scalability

The complexity of our method for calculating the components is $O(s \quad N \quad M)$ where $0 \quad s \quad 1$ is the density of the matrix; and $O(e \quad jV_u j \quad N)$ for learning the mixing weights, where e is the number of iterations for EM, and $jV_u j$ is the number of validation data points for a user. In practice e tends to be less than 10 in most cases (since the component parameters are fixed). Combined with the fact that the count matrices are often very sparse (so s is very small) this means that learning the predictive mixture model can be done very efficiently.

In addition the training of the mixture model is easy to parallelize: EM can be run in parallel for each user, and the additions required for calculating the components can be carried out in parallel as well. Figure 8 shows the amount of time in seconds taken for training a predictive mixture model for each of the datasets when using $[1; 2; 4; 8; 16; 32]$ processes. Using parallelization on 32 processors, we were able to run the model in times ranging from 2 to 1000 seconds depending on the data set, where each doubling of processor numbers resulted in close to halving the runtime. In contrast, NMF optimization has a complexity of $O(k \quad N \quad M)$ [27], [28]. Using the standard scikit-learn implementation used in this paper it took more than 3 hours on a 16-core machine to train an NMF model on the `redditT` dataset for $k = 500$.[7] It should be noted that our method could be extended to handle data in an online streaming fashion in a straightfoward manner given that the updates for each user can happen independently on different machines.

6 Related Work

We separate related work into three broad categories: (1) models that focus on discovering new items, (2) those that focus on reconsumption of old items, and (3) models that predict a mix of new and repeat consumption. Our work belongs in the last category as it focuses on balancing the two primary driving forces of human behavior in this context—discovery and reconsumption [10], [21].

The first category includes latent factor models based on matrix factorization techniques, which are widely used in research on recommender systems. In such applications, the model is primarily tasked with learning user preferences from historical observations and predicting future user-item interactions with a focus on the discovery of new items [3], [7], [14]. Due to the nature of this task, the observations are often treated as *implicit feedback* and it is typical for the evaluation to focus on a binary outcome; which new items a user will consume [3]. Zhao and Lee [13], for example, propose a method to model the curiosity of each user as a distribution over the novelty of items they have consumed. They develop a Probabilistic Curiosity Model as well as a method to measure the novelty of each item. Based on these two components, they make item recommendations that satisfy the optimal amount of curiosity for each user. In contrast we learn an implicit user exploration preference through a validation set, allowing the learning of exploration preferences directly from the consumption data, without external item information.

The second category of related work focuses on repeat consumption data, the importance of which has also been analyzed in literature [21], [22], [32], [33]. For example, in [32] the authors explore how many search engines queries are repeated and advocate for the value of recognizing them for a better user experience. In [21] the authors analyze a setup that is similar to ours, in which users make selections from a limited set of items and tend to heavily repeat their choices. They also examine statistics for multiple datasets with this property and explore the effects of popularity, satiation, and recency of previously selected items for prediction. They conclude that these are important features for modeling repeat consumption, but do not focus on modeling or predicting new item selections.

A third category of related work consists of literature that uses the *sequence* of events, focusing on prediction of the next event. Benson et. al., [22] examine time intervals between consumptions to predict whether a consumption will be novel or not, and builds models for both. Kapoor et. al., [4] argue that users are characterized by a novelty seeking propensity, which is both user and time-dependent and propose a metric to measure it. They build a prediction model for this propensity based on diversity in recent consumption. Within the literature for sequential data prediction, a different strand of work attempts to predict the time of consumption as well as the item. For instance authors in [34] model events as a point process, while in [35] the process is augmented with a recurrent neural network.

Sequence prediction has a different focus than the approach we propose here. We do not attempt to model time or assume that data is available in the form of sequences of events, but rather an aggregation of events is available in the form of a user-item matrix.Sequential data can sometimes be unavailable and sequential models can incur a large cost in space and time complexity (depending on the approach). Furthermore personalization can be difficult, since many approaches require a relatively large amount of data for each user. This often limits the work above to experiments involving a few *power* users [4], [22], or not be personalized

7. Our parallelizable implementation can be found here: https://github.com/UCIDataLab/repeat-consumption

at all [34]. In contrast our method is scalable, personalized, and can be used for users with little or no data.

Finally, the mixture approach pursued here is similar to a model that was originally proposed in [18]. In that work only global population-based weights were used in the mixture model and the analysis was limited to the context of spatial data. In this paper, as well as investigating individual weights and broader classes of datasets, we provide a significantly more in-depth analysis of explore/exploit behavior in the context of user-item consumption.

7 Discussion

In this paper we investigated the broad characteristics of user-item consumption for problems where there is both repeat and novel item consumption. We found that commonly used traditional matrix factorization methods have limitations in terms of being able to accurately reconstruct individual details, leading to a lack of predictive accuracy at the individual user level. In contrast, we propose a method based on mixture models, which is a simple, robust, and scalable approach for user modeling in this context. We use the EM algorithm to learn personalized mixing weights, without assuming a static distribution over user preferences, allowing the model to account for the propensity of some users to continuously explore. Experimental results on seven real-world data sets provide validation of the benefits of mixture approaches over matrix factorization in this general context.

One potential pitfall of our method is that it depends on a good validation set to learn the explore/exploit preferences of a user. This is prone to overfitting if the validation set is not representative, although the use of an empirical Bayes prior can mitigate this effect.

Future directions include leveraging temporal information in a scalable manner for additional improvements in predictive accuracy. Given that *recent* consumption is often indicative of the future behavior of a user [21], an obvious extension of the approach proposed in this paper would be to decay counts over time (e.g., exponentially), in effect upweighting the more recent user-item events. Another potentially useful direction would be to address the cold start problem, for both new users and new items, for example by developing an online version of the mixture approach.

Appendix
EM Equations

For learning of individual weights the EM algorithm can be applied to each user u's data independently. In the case where there are only two components, the mixing weights for a user u can be denoted as $_u$ and $(1 \quad _u)$. Let D_V represent the validation data for the user and n_{uj} be the number of times user u selected item j in the validation set, the likelihood of the validation data for the mixture model is:

$$p(D_V j \; ; \;) = \prod_{j=1}^{M} p(jj \; ; \;)^{n_{uj}} \tag{7}$$
$$= \prod_{j=1}^{M} \quad _u p^I (jj \quad ^I) + (1 \quad \quad _u) p^P (jj \quad ^P)^{\; n_{uj}}$$

For the **E-step** of the EM algorithm, we calculate the probability (or responsibility) that a single event (one selection of item j) was generated by the individual component p^I as

$$z_j = \frac{_u p^I \quad jj \quad ^I}{_u p^I \quad jj \quad ^I \quad + (1 \quad \quad _u) p^P \quad jj \quad ^P} \tag{8}$$

The **M-step** updates the mixing weights after summing the responsibilities for all points and normalizing the total component responsibility to sum to one. If ^{I} is the parameter of the Dirichlet prior for the component, the M-step becomes:

$$\begin{matrix} (t+1) \\ u \end{matrix} = \frac{\sum_{j=1}^{M} n_{uj} z_j + \quad ^I \quad 1}{\sum_{j=1}^{M} n_{uj} + \quad ^I + \quad ^P \quad 2} \tag{9}$$

Learning *global* mixing weights does not change the E and M steps, but the data for all users is used in the updates rather than per user.

Acknowledgments

The authors thank Eric Nalisnick for helpful discussions. This work was supported in part by a grant from the US National Science Foundation (NSF) under award numbers IIS 1320527 and CNS-1730158, by a Google Faculty Award, and by Adobe Research.

References

[1] Q. Wu, H. Wang, L. Hong, and Y. Shi, "Returning is believing: Optimizing long-term user engagement in recommender systems," in *Proceedings of the 26th ACM International on Conference on Information and Knowledge Management*, ser. CIKM '17. New York, NY, USA: ACM, 2017.

[2] J. McAuley and J. Leskovec, "Hidden factors and hidden topics: Understanding rating dimensions with review text," in *Proceedings of the 7th ACM Conference on Recommender Systems*, ser. RecSys '13. New York, NY, USA: ACM, 2013, pp. 165–172.

[3] X. Yang, Y. Guo, Y. Liu, and H. Steck, "A survey of collaborative filtering based social recommender systems," *Comput. Commun.*, vol. 41, pp. 1–10, Mar. 2014.

[4] K. Kapoor, V. Kumar, L. Terveen, J. A. Konstan, and P. Schrater, ""i like to explore sometimes": Adapting to dynamic user novelty preferences," in *Proceedings of the 9th ACM Conference on Recommender Systems*, ser. RecSys '15. New York, NY, USA: ACM, 2015, pp. 19–26.

[5] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," *ACM Trans. Interact. Intell. Syst.*, vol. 5, no. 4, pp. 1–19, Dec. 2015.

[6] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," in *Proceedings of the 20th International Conference on Neural Information Processing Systems*, ser. NIPS'07. USA: Curran Associates Inc., 2007, pp. 1257–1264.

[7] X. Ning and G. Karypis, "Slim: Sparse linear methods for top-n recommender systems," in *Proceedings of the 11th IEEE International Conference on Data Mining*, ser. ICDM '11. IEEE, 2011, pp. 497–506.

[8] D. E. Berlyne, *Conflict, Arousal, and Curiosity.* New York, NY, US: McGraw-Hill Book Company, 1960.

[9] L. McAlister and E. Pessemier, "Variety seeking behavior: An interdisciplinary review," *Journal of Consumer research*, vol. 9, no. 3, pp. 311–322, 1982.

[10] K. Bawa, "Modeling inertia and variety seeking tendencies in brand choice behavior," *Marketing Science*, vol. 9, no. 3, pp. 263–278, 1990.

[11] J. Jacoby and D. B. Kyner, "Brand loyalty vs. repeat purchasing behavior," *Journal of Marketing research*, vol. 10, no. 1, pp. 1–9, 1973.