When Actions Speak Louder than Clicks: A Combined Model of Purchase Probability and Long-term Customer Satisfaction

Gal Lavee
eBay Research, Israel
glavee@ebay.com

Noam Koenigstein
Microsoft R&D, Israel
Tel Aviv University, Israel
noamko@microsoft.com

Oren Barkan
Microsoft R&D, Israel
orenb@microsoft.com

ABSTRACT

Maximizing sales and revenue is an important goal of online commercial retailers. Recommender systems are designed to maximize users' click or purchase probability, but often disregard users' eventual satisfaction with purchased items. As result, such systems promote items with high appeal at the selling stage (e.g. an eye-catching presentation) over items that would yield more satisfaction to users in the long run. This work presents a novel unified model that considers both goals and can be tuned to balance between them according to the needs of the business scenario.

We propose a multi-task probabilistic matrix factorization model with a dual task objective: predicting binary purchase/no purchase variables combined with predicting continuous satisfaction scores. Model parameters are optimized using Variational Bayes which allows learning a posterior distribution over model parameters. This model allows making predictions that balance the two goals of maximizing the probability for an immediate purchase and maximizing user satisfaction and engagement down the line. These goals lie at the heart of most commercial recommendation scenarios and enabling their balance has the potential to improve value for millions of users worldwide. Finally, we present experimental evaluation on different types of consumer retail datasets that demonstrate the benefits of the model over popular baselines on a number of well-known ranking metrics.

CCS CONCEPTS

• **Information systems → Probabilistic retrieval models**.

KEYWORDS

Recommendations, Continuous Implicit Data, Variational Methods

ACM Reference Format:
Gal Lavee, Noam Koenigstein, and Oren Barkan. 2019. When Actions Speak Louder than Clicks: A Combined Model of Purchase Probability and Long-term Customer Satisfaction. In *Thirteenth ACM Conference on Recommender Systems (RecSys '19), September 16–20, 2019, Copenhagen, Denmark*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3298689.3347044

RecSys '19, September 16–20, 2019, Copenhagen, Denmark

ACM ISBN 978-1-4503-6243-6/19/09…$15.00
https://doi.org/10.1145/3298689.3347044

1 INTRODUCTION

Much of the early literature in the recommendation system community has been devoted to learning models from *explicit* satisfaction scores such as the ratings available in the Netflix prize competition. Since those early days, many have observed that such explicit ratings are often not available in practice and most of the information for user preference is obtained by collecting *implicit* binary signals such as click/no click, purchase/no purchase or pairwise preferences [15, 21, 26]. However, by considering only binary signals, the system does not take into account the ultimate satisfaction of users with the consumed item.

While explicit ratings are often unavailable, satisfaction scores can often be implicitly extracted. With the increased abundance of telemetry data, a rich signal of customer preference has become available. One example of such a scenario is online video streaming, where the amount of time a customer spends watching a movie is recorded. In a music streaming scenario, we can track the number of times a user listens to the same song or artist (or alternatively the number of times that song/artist is skipped when it appears in a randomly generated playlist). In software application (app) recommendations, such as those in the Windows Store, we can track the amount of time a user spends engaging with the app or the number of times the app has been launched. Another notable type of implicit satisfaction score is dwell time in online recommendation experiences (studied in [3, 35, 36]). Thus, a *continuous* implicit signal is generated that gives a finer granularity of user preference than binary interactions.

In the examples above, the continuous preference signals encode a customer's satisfaction and engagement with the purchased items. Utilizing such signals enables systems that directly model and predict the customer's engagement at the consumption stage beyond the initial purchase. It has the double benefit of allowing online retailers to increase revenue while presenting items that users are more likely to enjoy.

This paper proposes a model that combines continuous and binary implicit signals. To our knowledge, models and algorithms for considering implicit continuous signals have not been studied in the literature of recommendation systems. Beyond the obvious business advantage of recommending satisfactory items, our empirical evaluation demonstrates that in some cases the additional information on user engagement and satisfaction can even yield improved performance across standard ranking metrics.

This paper makes the following contributions: (1) we propose a model that combines the implicit continuous *score* of user preference with a binary consumption signal using a multi-task likelihood

function. This model enables balancing immediate purchase probabilities with implicit satisfaction scores. (2) we derive a computationally tractable algorithm for learning the model parameters from data, applying the algorithmic framework of Variational Bayes, (3) we evaluate the proposed approach empirically over several datasets, both proprietary and public.

The remainder of this paper is organized as follows: Section 2 discusses related work, Section 3 describes the model and algorithm. Section 4 presents the experimental setup and results, and finally Section 5 provides our conclusions.

2 RELATED WORK

Recommender systems have great potential to help users navigate the long-tail of retail catalogs to locate items relevant to themselves. However, it has been noticed that improved performance on the standard collaborative filtering task often does not translate into increased user satisfaction [5, 14, 19]. Hence, in this work we present a model that attempts to learn both a binary probability for an item to be clicked or purchased as well as learning the eventual satisfaction that the user will get from that item (if purchased).

The work is motivated by the paradigm of multi-task learning [4, 28] applied to Collaborative filtering based on Matrix Factorization. Initial Matrix Factorization (MF) models were based on a regression objective parameterized by low-dimensional latent vectors corresponding to each user and item. These parameters are usually optimized by way of minimizing a squared loss function using either stochastic gradient descent (SGD), or alternating least squares (ALS) [17]. The MF objective can also be cast as a Bayesian probabilistic model with different likelihood functions [12, 16, 22, 24, 29, 34].

The lack of *explicit* data in collaborative filtering has led to concentrated study of models for *implicit* data [10, 13, 21, 26]. Training data is often binary such as click/no click or purchase/no purchase [7–9, 25, 31–33]. The implicit data setup is sometimes referred to as a 'One-Class' problem [16, 24, 27, 37] owing to the fact that users are unlikely to interact with items they have negative preference for. Accounting for the (non-observed) negatives makes the implicit data problems computationally harder. Previous models have either based on the "whole data approaches" [10], which give a small weight to every unobserved item, or based on negative sampling approaches [23, 24, 26]. The model in this paper follows a negative sampling approach very similar to the one in [24], while comparisons with [26] and [10] are discussed in Section 4.

3 OUR APPROACH

Our objective in this work is to construct a model that considers both the probability for a user to click or purchase an item as well as a continuous valued satisfaction score (measured following the purchase). For example, in the case of Xbox games, we learn the probability to purchase a game together with the time the user would play it if she were to buy the game. In this case, the time the user plays a game is a proxy for her satisfaction. Obviously, the choice of satisfaction score is not limited to engagement time and can be chosen in-order to satisfy different business needs. In the interest of generality, henceforth we simply consider two types of observations: a binary observation and a continuous satisfaction score. As such we devise a multi-task objective composed of a binary term and a continuous term where the relative importance of each term is determined by a tunable hyperparameter. Formalized this way, the model generalizes both binary matrix factorization as well as continuous (regression) matrix factorization. In what follows we give a formal description of the model.

Definitions and Notation

We distinguish matrices and vectors from scalars by denoting them with bold letters. We capitalize when denoting matrices and use minuscule letters for vectors, e.g., $\mathbf{X}$ is a matrix , $\mathbf{x}$ is a vector and both x and X denote scalars. We denote an expectation over a random variable using angle brackets e.g., $\langle \mathbf{a} \rangle$ is the expectation of the random vector $\mathbf{a}$. We reserve special indexing letters for distinguishing users from items: for users we use the letter m and for items the letter n. The total number of users in the model is M and the total number of items is N.

The multi-task model considers two datasets: a dataset for binary observations denoted $\mathcal{D}^b$ and a dataset for continuous scores denoted $\mathcal{D}^s$. Each dataset consists of user/item tuples and an observation. A binary observation of user m to item n in $\mathcal{D}^b$ is denoted by r_{mn}. A continuous observation of the same user to the same item in $\mathcal{D}^s$ is denoted by s_{mn}. Namely, the datasets consist of the following tuples: $(m, n, r_{mn}) \in \mathcal{D}^b$ and $(m, n, s_{mn}) \in \mathcal{D}^s$. Finally, we denote by $\mathcal{D}^s_m$ and $\mathcal{D}^b_m$ all the observations in datasets $\mathcal{D}^s$ and $\mathcal{D}^b$ that relate to user m, respectively. Similarly, we denote by $\mathcal{D}^s_n$ and $\mathcal{D}^b_n$ all the observations in datasets $\mathcal{D}^s$ and $\mathcal{D}^b$ that relate to item n, respectively.

3.1 Intuition and Overview

Our model is a Bayesian multi-task model based on matrix factorization (Section 3.2). Model parameters are estimated using Variational Bayes inference [2]. Namely, we find the best approximation to the posterior distribution given the observed training data. This is in contrast to Bayesian approaches that only seek a point-estimate of the parameters which maximize the posterior (or non-Bayesian models that minimize a loss function with respect to the parameters). In complex models, such as the one presented in this work, the posterior distribution is intractable. In Variational Bayes we seek a factorized approximation distribution that minimizes the Kullback-Leibler divergence from the true posterior (Section 3.3). Optimization proceeds by an iterative algorithm which optimizes each factor's parameters in turn. We present full update equations in Section 3.4. With the estimated posterior distribution at hand, we can make predictions considering all possible parameter configurations (Section 3.6).

3.2 Model Formalization

For each user m and item n, we denote by $\mathbf{u}_m, \mathbf{v}_n \in \mathbb{R}^d$ the d dimensional latent user and item vector parameters, respectively. Similarly, the scalar bias of each user and item is denoted by b_m and b_n, respectively. The model includes two additional scalars to allow the location and scale of the distribution of score values to move away from that of the binary observations: ψ for location, and κ for scale. We collectively denote all the model's parameters by:

$$\theta = \{\{\mathbf{u}_m\}_{m=1}^M, \{\mathbf{v}_n\}_{n=1}^N, \{b_m\}_{m=1}^M, \{b_n\}_{n=1}^N, \kappa, \psi\}. \quad (1)$$

3.2.1 Likelihood and Priors. We model the joint probability of a single paired binary observation r_{mn} and score s_{mn} as a factorized product with mixing hyper-parameter γ as follows:

$$\begin{aligned} &p(s_{mn}, r_{mn} | \mathbf{u}_m, \mathbf{v}_n, b_m, b_n, \kappa, \psi) = \\ &\text{Bern}\Big(r_{mn}; \sigma\left(\gamma(\mathbf{u}_m^\top \mathbf{v}_n + b_m + b_n)\right)\Big) \\ &\cdot \mathcal{N}\Big(s_{mn}; \Big(\kappa(\mathbf{u}_m^\top \mathbf{v}_n + b_m + b_n) + \psi\Big), ((1-\gamma)\beta)^{-1}\Big) = \\ &\sigma\left(\gamma(\mathbf{u}_m^\top \mathbf{v}_n + b_m + b_n)\right)^{r_{mn}} \cdot \left[1 - \sigma\left(\gamma(\mathbf{u}_m^\top \mathbf{v}_n + b_m + b_n)\right)\right]^{1-r_{mn}} \\ &\cdot \sqrt{\tfrac{(1-\gamma)\beta}{2\pi}} \exp\left\{-\tfrac{(1-\gamma)\beta}{2}\left[s_{mn} - \Big(\kappa(\mathbf{u}_m^\top \mathbf{v}_n + b_m + b_n) + \psi\Big)\right]^2\right\} \end{aligned} \tag{2}$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function. This form arises naturally by modeling

$$p\left(r_{mn}, s_{mn} \mid \theta\right) \propto f(r_{mn} \mid \theta)^{\gamma} \cdot g\left(s_{mn} \mid \theta\right)^{(1-\gamma)},$$

where we choose $f(r_{mn} \mid \theta) = \text{Bern}\Big(r_{mn}; \sigma\left(\mathbf{u}_m^\top \mathbf{v}_n + b_m + b_n\right)\Big)$ and $g(s_{mn} \mid \theta) = \mathcal{N}\left(s_{mn}; \kappa\left(\mathbf{u}_m^\top \mathbf{v}_n + b_m + b_n\right) + \psi, \beta^{-1}\right)$ to be Bernoulli and Gaussian distribution density functions, respectively. The hyper-parameter γ controls the importance of binary observation with respect to the continuous observation: when $\gamma = 1$ we are left with just the binary term and when $\gamma = 0$ we have only the regression term. Hence the model naturally generalizes both binary and regression matrix factorization.

The use of shared parameters for both the binary and the continuous terms adds structure to the model which assumes a positive correlation between the two components of the likelihood. However, the assumption that these shared components are distributed identically is too strong. One distribution may need to have a different dynamic range and be centered around a different mean than the other. Therefore, parameters κ and ψ (collectively referred to as *stretch parameters*) allow the location and scale of the distribution of score values to vary from that of the binary component of the likelihood. Without these additional parameters, one would need to find a non-trivial normalization on the score values in the data so that their distribution would match that of the latent parameters that best describe the binary component or else suffer degradation in accuracy.

We define simple Gaussian priors on all parameters as follows:

$$\begin{aligned} &p(\mathbf{u}_m|\alpha_u) = \mathcal{N}(\mathbf{u}_m; 0, \alpha_u^{-1}\mathbf{I}_d), && p(\mathbf{v}_n|\alpha_v) = \mathcal{N}(\mathbf{v}_n; 0, \alpha_v^{-1}\mathbf{I}_d), \\ &p(b_m|\alpha_b) = \mathcal{N}\Big(b_m; 0, \alpha_{b_m}^{-1}\Big), && p(b_n|\alpha_b) = \mathcal{N}\Big(b_n; 0, \alpha_{b_n}^{-1}\Big), \\ &p(\kappa|\alpha_\kappa) = \mathcal{N}(\kappa; 1, \alpha_\kappa^{-1}), && p(\psi|\alpha_\psi) = \mathcal{N}\Big(\psi; 0, \alpha_\psi^{-1}\Big), \end{aligned} \tag{3}$$

where $\mathbf{I}_d$ is a d dimensional identity matrix and the α's are precision hyper-parameters. All model hyper-parameters are collectively denoted by $\mathcal{H} = \{\gamma, \alpha_u, \alpha_v, \alpha_{b_u}, \alpha_{b_i}, \alpha_\kappa, \alpha_\psi, \beta\}$. Note that all the priors are normally distributed with zero mean except κ which has mean 1.

3.2.2 The Posterior Distribution. Our goal is to find the posterior distribution:

$$\begin{aligned} p(\theta|\mathcal{D}, \mathcal{H}) \quad \propto \quad & p(\theta, \mathcal{D}|\mathcal{H}) = p(\mathcal{D}|\theta, \mathcal{H}) p(\theta|\mathcal{H}) = \\ & \prod_{(m,n,r_{mn}) \in \mathcal{D}^b} \text{Bern}\Big(r_{mn}; \sigma\left(\gamma(\mathbf{u}_m^\top \mathbf{v}_n + b_m + b_n)\right)\Big) \\ & \prod_{(m,n,s_{mn}) \in \mathcal{D}^s} \mathcal{N}\Big(s_{mn}; \Big(\kappa(\mathbf{u}_m^\top \mathbf{v}_n + b_m + b_n) + \psi\Big), ((1-\gamma)\beta)^{-1}\Big) \\ & \cdot \prod_{m=1}^{M} \mathcal{N}(\mathbf{u}_m; 0, \alpha_u^{-1}\mathbf{I}_d) \cdot \mathcal{N}\Big(b_m; 0, \alpha_{b_u}^{-1}\Big) \\ & \cdot \prod_{n=1}^{N} \mathcal{N}(\mathbf{v}_n; 0, \alpha_v^{-1}\mathbf{I}_d) \cdot \mathcal{N}\Big(b_n; 0, \alpha_{b_v}^{-1}\Big) \\ & \cdot \mathcal{N}(\kappa; 1, \alpha_\kappa^{-1}) \cdot \mathcal{N}\Big(\psi; 0, \alpha_\psi^{-1}\Big). \end{aligned} \tag{4}$$

Estimating the parameters of this posterior is computationally intractable. Hence, we turn to Variational Bayes in order to approximate it.

3.3 Variational Bayes

We seek an approximate distribution $q(\theta)$ that will minimize the Kullback-Leibler divergence from the true posterior:

$$\mathbb{D}_{KL}(q(\theta) \| p(\theta|\mathcal{D}, \mathcal{H})) \stackrel{\text{def}}{=} \int q(\theta) \log \tfrac{q(\theta)}{p(\theta|\mathcal{D}, \mathcal{H})} d\theta. \tag{5}$$

This divergence can be rewritten in terms of the model's marginal log-likelihood and the variational free energy $\mathcal{F}[q(\theta)]$ (see [2]):

$$\mathbb{D}_{KL}(q(\theta) \| p(\theta|\mathcal{D}, \mathcal{H})) + \mathcal{F}[q(\theta)] = \log p(\mathcal{D}|\mathcal{H}), \tag{6}$$

where the variational free energy is $\mathcal{F}[q(\theta)] \stackrel{\text{def}}{=} \int q(\theta) \log \frac{p(\theta, \mathcal{D}|\mathcal{H})}{q(\theta)} d\theta$. From the expression above we can see that minimizing the Kullback-Leibler divergence is equivalent to maximizing the variational free energy $\mathcal{F}[q(\theta)]$. Henceforth, our objective is to maximize $\mathcal{F}[q(\theta)]$ with respect to the approximate posterior distribution $q(\theta)$.

3.3.1 Logistic Bound. Before we proceed, we notice that the joint distribution in (4) includes Gaussian priors which are not conjugate with respect to the sigmoid functions used in the likelihood. This lack of conjugacy precludes closed form parametric solutions to the factors of the approximate posterior distribution. Therefore, we apply the Jaakola-Jordan logistic bound [11] to replace these sigmoid functions with a "squared exponential" term forming a tight lower bounds on $\mathcal{F}[q(\theta)]$. The Jaakola-Jordan introduces additional variational parameter $\xi_{m,n} > 0$ to bound each binary observation $(m, n, r_{mn}) \in \mathcal{D}^b$ as follows:

$$\begin{aligned} &\sigma(\gamma a)^r \cdot [1 - \sigma(\gamma a)]^{1-r} \quad \geq \\ &\sigma(\xi) \exp\{\gamma r a - \frac{1}{2}(\gamma a + \xi) - \lambda(\xi)(\gamma^2 a^2 - \xi^2)\}, \end{aligned} \tag{7}$$

where we have dropped subscripts m and n for clarity, $\lambda(\xi) \stackrel{\text{def}}{=} \frac{1}{2\xi}[\sigma(\xi) - \frac{1}{2}]$ and $a \stackrel{\text{def}}{=} \mathbf{u}_m^\top \mathbf{v}_n + b_m + b_n$ is used for simplicity. By maximizing the $\xi_{m,n}$ values, this bound becomes a tight lower bound (see [11]) that can be used to replace the sigmoid terms in the likelihood with "Gaussian like" exponentiated quadratic terms.

Using the inequality in (7) we substitute all the sigmoids in $p(\theta, \mathcal{D}|\mathcal{H})$ to get $p_\xi(\theta, \mathcal{D}|\mathcal{H})$. We now have a lower bound on the the variational free energy $\mathcal{F}[q(\theta)] \geq \mathcal{F}_\xi[q(\theta)] \stackrel{\text{def}}{=} \int q(\theta) \log \frac{p_\xi(\theta, \mathcal{D}|\mathcal{H})}{q(\theta)} d\theta$, and

$\mathcal{F}_{\tilde{\xi}}[q(\theta)]$ becomes our new maximization objective. Next, we explain the optimization procedure.

3.3.2 Optimization Procedure. We approximate the posterior distribution with the following factorized approximation function $q(\theta)$:

$$q(\theta)=\prod_m^M q(\mathbf{u}_m)\cdot\prod_n^N q(\mathbf{v}_n)\cdot\prod_m^M q(b_m)\cdot\prod_n^N q(b_n)\cdot q(\kappa)\cdot q(\psi). \quad (8)$$

For the sake of uncluttered notation, we overload the letter q in Equation (8) above. $q(\cdot)$ denotes both the entire approximate posterior and each factor of the posterior. The different meanings of q can be distinguished by the argument to the function.

Optimization of $\mathcal{F}_{\tilde{\xi}}$ with respect to $q(\theta)$ is achieved through coordinate ascent in the function space of the variational distributions. Namely, we compute the functional derivative $\partial\mathcal{F}_{\tilde{\xi}}/\partial q$ with respect to each distribution q in (8). Setting these derivatives to zero, together with a Lagrange multiplier constraint to make each q integrate to one, we get the update steps for each q distribution. The Jakkola-Jordan bound in (7) ensures that the factors in Equation (8) takes the form of a Gaussian distribution parameterized by a mean and a covariance. We iteratively update the q distributions, where each update increases our objective $\mathcal{F}_{\tilde{\xi}}$ with respect to one parameter in θ. Since $\mathcal{F}_{\tilde{\xi}}$ is bounded from above by $\log p(\mathcal{D}|\mathcal{H})$, the optimization is guaranteed to converge. Next, we present a pseudocode for the optimization procedure.

Algorithm 1: Learn the parameters of the approximate posterior (Equation 8) from data

repeat
  Add Negative samples to training data (see section 3.5)
  for $m = 1 \ldots M$ **do**
    Update $q(\mathbf{u}_m)$ parameters $\mu_{\mathbf{u}_m}$ and $\Sigma_{\mathbf{u}_m}$ using Equation (9)*
  end for
  for $n = 1 \ldots N$ **do**
    Update $q(\mathbf{v}_n)$ parameters $\mu_{\mathbf{v}_n}$ and $\Sigma_{\mathbf{v}_n}$ using Equation (10)*
  end for
  for $m = 1 \ldots M$ **do**
    Update $q(b_m)$ parameters μ_{b_m} and $\sigma^2_{b_m}$ using Equation (11)*
  end for
  for $n = 1 \ldots N$ **do**
    Update $q(b_n)$ parameters μ_{b_n} and $\sigma^2_{b_n}$ using Equation (12)*
  end for
  Update $q(\kappa)$ parameters μ_κ and σ^2_κ using Equation (13)
  Update $q(\psi)$ parameters μ_ψ and σ^2_ψ using Equation (14)
until convergence of $\mathcal{F}[q(\theta)]$
(*) Whenever $\xi_{m,n}$ appears in the update equation calculate it "on the fly" using Equation (15)

3.3.3 Pseudocode. Algorithm 1 gives the pseudocode for learning the approximate posterior parameters [1]. The update equations, described in Section 3.4, allow for independence between all user parameters given the item parameters (and vice versa). Therefore, all of the *for* loops in the above pseudocode can be carried out in parallel, providing a substantial speed-up in training runtime. Indeed, our implementation applies such parallelization.

3.4 Update Equations

As stated above, each factor in our approximate posterior can be shown to be Gaussian. Thus, we describe the update steps in terms of their sufficient statistic - namely the mean and the covariance or variance of each factor in the approximate posterior.

[1] parameters are initialized randomly using a multivariate Gaussian with zero mean and small variance (0.01).

User vectors update $q(\mathbf{u}_m)$:

The posterior for each user vector is approximated with $q(\mathbf{u}_m)=\mathcal{N}\left(\mathbf{u}_m;\boldsymbol{\mu}_{\mathbf{u}_m},\Sigma_{\mathbf{u}_m}\right)$, where the mean $\boldsymbol{\mu}_{\mathbf{u}_m}$ and covariance $\Sigma_{\mathbf{u}_m}$ are given by:

$$\Sigma_{\mathbf{u}_m}=\left(\alpha_u\cdot\mathbf{I}_d+2\gamma^2\sum_{n\in\mathcal{D}_m^b}\lambda(\xi_{m,n})\langle\mathbf{v}_n\mathbf{v}_n^\top\rangle+(1-\gamma)\beta\langle\kappa^2\rangle\sum_{n\in\mathcal{D}_m^s}\langle\mathbf{v}_n\mathbf{v}_n^\top\rangle\right)^{-1}, \quad (9)$$

and

$$\boldsymbol{\mu}_{\mathbf{u}_m}=\Sigma_{\mathbf{u}_m}\left(\gamma\sum_{n\in\mathcal{D}_m^b}\left[r_{mn}-\tfrac{1}{2}-2\gamma\lambda(\xi_{m,n})\langle b_n+b_m\rangle\right]\langle\mathbf{v}_n\rangle+(1-\gamma)\beta\sum_{n\in\mathcal{D}_m^s}\left[\langle\kappa\rangle(s_{mn}-\langle\psi\rangle)-\langle\kappa^2(b_m+b_n)\rangle\right]\langle\mathbf{v}_n\rangle\right).$$

Item vectors update $q(\mathbf{v}_n)$:

The posterior for each item vector is approximated with $q(\mathbf{v}_n)=\mathcal{N}\left(\mathbf{v}_n;\boldsymbol{\mu}_{\mathbf{v}_n},\Sigma_{\mathbf{v}_n}\right)$, where the mean $\boldsymbol{\mu}_{\mathbf{v}_n}$ and covariance $\Sigma_{\mathbf{v}_n}$ are given by:

$$\Sigma_{\mathbf{v}_n}=\left(\alpha_v\cdot\mathbf{I}_d+2\gamma^2\sum_{m\in\mathcal{D}_n^b}\lambda(\xi_{m,n})\langle\mathbf{u}_m\mathbf{u}_m^\top\rangle+(1-\gamma)\beta\langle\kappa^2\rangle\sum_{m\in\mathcal{D}_n^s}\langle\mathbf{u}_m\mathbf{u}_m^\top\rangle\right)^{-1}, \quad (10)$$

and

$$\boldsymbol{\mu}_{\mathbf{v}_n}=\Sigma_{\mathbf{v}_n}\left(\gamma\sum_{m\in\mathcal{D}_n^b}\left[r_{mn}-\tfrac{1}{2}-2\gamma\lambda(\xi_{m,n})\langle b_n+b_m\rangle\right]\langle\mathbf{u}_m\rangle+(1-\gamma)\beta\sum_{m\in\mathcal{D}_n^s}\left[\langle\kappa\rangle(s_{mn}-\langle\psi\rangle)-\langle\kappa^2(b_m+b_n)\rangle\right]\langle\mathbf{u}_m\rangle\right).$$

User biases update $q(b_m)$:

The posterior for each user bias is approximated with $q(b_m)=\mathcal{N}\left(b_m;\mu_{b_m},\sigma^2_{b_m}\right)$, where the mean μ_{b_m} and variance $\sigma^2_{b_m}$ are given by:

$$\sigma^2_{b_m}=\left(2\gamma^2\sum_{n\in\mathcal{D}_m^b}\lambda(\xi_{m,n})+(1-\gamma)\beta\langle\kappa^2\rangle|\mathcal{D}_m^s|+\alpha_{b_m}\right)^{-1}, \quad (11)$$

and

$$\mu_{b_m}=\sigma^2_{b_m}\cdot\left(\gamma\sum_{n\in\mathcal{D}_m^b}\left(r_{mn}-\tfrac{1}{2}-2\gamma\lambda(\xi_{m,n})\langle\mathbf{v}_n^\top\mathbf{u}_m+b_n\rangle\right)+(1-\gamma)\beta\sum_{n\in\mathcal{D}_m^s}\left(\langle\kappa\rangle(s_{mn}-\langle\psi\rangle)-\langle\kappa^2(\mathbf{v}_n^\top\mathbf{u}_m+b_n)\rangle\right)\right)$$

where $|\mathcal{D}_m^s|$ is the number of observations in $\mathcal{D}_m^s$.

Item biases update $q(b_n)$:

The posterior for each user bias is approximated with $q(b_n)=\mathcal{N}\left(b_n;\mu_{b_n},\sigma^2_{b_n}\right)$, where the mean μ_{b_n} and variance $\sigma^2_{b_n}$ are given by:

$$\sigma^2_{b_n}=\left(2\gamma^2\sum_{m\in\mathcal{D}_n^b}\lambda(\xi_{m,n})+(1-\gamma)\beta\langle\kappa^2\rangle|\mathcal{D}_n^s|+\alpha_{b_n}\right)^{-1}, \quad (12)$$

and

$$\mu_{b_n}=\sigma^2_{b_n}\cdot\Bigg(\gamma\sum_{m\in\mathcal{D}^b_n}\big(r_{mn}-\tfrac{1}{2}-2\gamma\lambda\big(\xi_{m,n}\big)\langle\mathbf{v}_n^\top\mathbf{u}_m+b_m\rangle\big)+$$
$$(1-\gamma)\beta\sum_{m\in\mathcal{D}^s_n}\big(\langle\kappa\rangle(s_{mn}-\langle\psi\rangle)-\langle\kappa^2(\mathbf{v}_n^\top\mathbf{u}_m+b_m)\rangle\big)\Bigg)$$

where $|\mathcal{D}^s_n|$ is the number of observations in $\mathcal{D}^s_n$.

Updating the stretch scale parameter $q(\kappa)$:

The posterior for κ is approximated with $q(\kappa)=\mathcal{N}(\kappa;\mu_\kappa,\sigma^2_\kappa)$, where the mean μ_κ and variance σ^2_κ are given by:

$$\sigma^2_\kappa=\Bigg((1-\gamma)\beta\sum_{\mathcal{D}^s}\langle\big(\mathbf{u}_m^\top\mathbf{v}_n+b_m+b_n\big)^2\rangle+\alpha_\kappa\Bigg)^{-1}, \tag{13}$$

and

$$\mu_\kappa=\sigma^2_\kappa\cdot\Bigg(\alpha_\kappa+(1-\gamma)\beta\sum_{\mathcal{D}^s}(s_{mn}-\langle\psi\rangle)\langle\mathbf{u}_m^\top\mathbf{v}_n+b_m+b_n\rangle\Bigg).$$

Updating the stretch location parameter $q(\psi)$:

The posterior for ψ is approximated with $q(\psi)=\mathcal{N}\Big(\psi;\mu_\psi,\sigma^2_\psi\Big)$, where the mean μ_ψ and variance σ^2_ψ are given by:

$$\sigma^2_\psi=\Bigg((1-\gamma)\beta|\mathcal{D}^s|+\alpha_\psi\Bigg)^{-1}, \tag{14}$$

and

$$\mu_\psi=\sigma^2_\psi\cdot\Bigg((1-\gamma)\beta\sum_{\mathcal{D}^s}\big(s_{mn}-\langle\kappa\rangle\langle\mathbf{u}_m^\top\mathbf{v}_n+b_m+b_n\rangle\big)\Bigg),$$

where $|\mathcal{D}^s|$ is the number of observations in $\mathcal{D}^s$.

Update for variational parameters

As explained in Section 3.3.1, we are optimizing a lower bound on the logistic free energy $\mathcal{F}_{\xi}[q(\theta)]$ which includes variational parameters $\xi_{m,n}$. These parameters are computed as needed (on the fly) according to:

$$\xi_{m,n}=\gamma\sqrt{\mathbb{VAR}[\mathbf{u}_m^\top\mathbf{v}_n]+\sigma^2_{b_m}+\sigma^2_{b_n}+\big(\mu_{\mathbf{u}_m}^\top\mu_{\mathbf{v}_n}+\mu_{b_m}+\mu_{b_n}\big)^2}. \tag{15}$$

3.5 Negative Sampling

Implicit data recommenders, such as in this work, often form one-class problems [24]. Our approach adopts 'negative sampling' to deal with this issue - a strategy popular in the implicit data paradigm of recommendation systems [16, 23, 24, 26] as well as other areas of machine learning (most notably for learning word embeddings[20]). The approach entails augmenting the positive data in the training dataset with additional data representing examples that have negative labels (i.e. items disliked by the user in a recommendation scenario).

Recall that, the training data $\mathcal{D}^b$ and $\mathcal{D}^s$ includes both binary and continuous observations. Thus, the augmenting 'negative' data points should also contain a binary and continuous component. Clearly the binary component should be set to 0, however it is not clear what is the correct value of the continuous component. We opted for the following negative sampling scheme: for each user we sample a number of items equal to the number of items available in the training data (i.e. 'user positive examples'). The choice of item is done uniformly across all catalog items excluding those 'positive items'. That is, we select item n as a negative for user m with probability:

$$P(\text{User } m \text{ dislikes Item } n)=\begin{cases}\frac{1}{\left(N-\left|\mathcal{D}^b_m\right|\right)} & n\notin\mathcal{D}^b_m\\ 0 & n\in\mathcal{D}^b_m\end{cases} \tag{16}$$

For the sampled item n, we add $r_{mn}=0$ to $\mathcal{D}^b$ and $s_{mn}=-1$ to $\mathcal{D}^s$. The motivation for this choice is to create a stark separation between the scores observed in the data (which are all positive) and the scores sampled as part of the data augmentation.

Although adding negative sampling has significant impact on the performance of the approach (see results in Section 4), the method is fairly robust to the specifics of how the negative sampling is carried out. We experimented with a number of such schemes and found no major differences in the results, and hence we opted for a straightforward approach.

3.6 Prediction Using the Learned Model

As explained earlier, unlike other algorithms (e.g., Expectation Maximization) that use point-estimates of the parameters, Variational Bayes estimates the full posterior distribution. Namely, we approximated the posterior $p(\theta|\mathcal{D},\mathcal{H})$ with $q(\theta)$. At prediction time, we compute expectations over all possible values of the parameters according to $q(\theta)$ which gives a prediction that is more robust to over-fitting, under-fitting and general modeling choices such as hyper-parameters.

Our model is a multi-task model capable of two types of predictions: the predicted binary probability (e.g., the probability for buying an item) and the expected satisfaction score (e.g., how much time the user will spend playing the game after purchasing it). The exact utilization of these two predictions varies according to the company's business needs and is out of scope for this paper, however combining these two types of predictions allow powering interesting scenarios such as choosing items that the user is both likely to purchase and enjoy after purchasing. These two predictions values can also be used as features by a second layer of learning in order to solve the list recommendation problem [30].

Binary Signal Prediction:

We predict the probability for a positive signal by computing the following expectation:

$$\begin{aligned}p(r_{mn}=1|\mathcal{D},\mathcal{H})&\approx\Big\langle\sigma\big(\gamma\big(\mathbf{u}_m^\top\mathbf{v}_n+b_m+b_n\big)\big)\Big\rangle\\&\approx\int\sigma(a)\mathcal{N}(a;\mu_a,\sigma^2_a)\mathrm{d}a,\end{aligned} \tag{17}$$

where $a\overset{\text{def}}{=}\gamma\big(\mathbf{u}_m^\top\mathbf{v}_n+b_m+b_n\big)$ is approximated with a Gaussian using its first and second moments according to $q(\theta)$:

$$\begin{aligned}\mu_a&=\Big\langle\gamma\big(\mathbf{u}_m^\top\mathbf{v}_n+b_m+b_n\big)\Big\rangle\\\sigma^2_a&=\Big\langle\big(\gamma\big(\mathbf{u}_m^\top\mathbf{v}_n+b_m+b_n\big)-\mu_a\big)^2\Big\rangle.\end{aligned} \tag{18}$$

The integral in Equation 17 is non-analytical and, thus, approximated with

$$\int\sigma(a)\mathcal{N}(a;\mu_a,\sigma^2_a)\mathrm{d}a\approx\sigma\Big(\mu_a/\sqrt{1+\pi\sigma^2_a/8}\Big), \tag{19}$$

which follows MacKay [18].

Dataset	# Users	# Items	# Data Points
Xbox Games	100,000	596	1,409,332
Windows Games	10,000	1,000	57,778
LastFM	9,982	1,000	234,089
MovieLens	6040	3,681	1,001,80

Table 1: Dataset statistics

Continuous Signal Prediction:

The continuous score is predicted by approximating the expected score μ_s as follows:

$$\mu_s \stackrel{\text{def}}{=} \int s \cdot p(s_{mn}=s \mid \mathcal{D}, \mathcal{H}) \mathrm{d}s \approx \left\langle \kappa(\mathbf{u}_m^\top \mathbf{v}_n + b_m + b_n) + \psi \right\rangle. \quad (20)$$

4 EVALUATION

We evaluate several variants of our model against popular baselines for related problems from the literature. We used several datasets representing different types of on-line retail products across both proprietary and publicly available data.

4.1 Datasets

Evaluations are based on four datasets: Xbox games, Windows games (WinGames), LastFM [1], and MovieLens (1M) [6]. The Xbox games and WinGames datasets are proprietary datasets from commercial stores. These datasets were constructed by sampling a fixed size of users uniformly. As such, the patterns represent real usage patterns. The binary signal is the purchase signal and we use engagement time as a proxy for satisfaction. Namely, the implicit satisfaction score is based on some proprietary combination of the absolute time spent playing the game and the number of times the game was launched. These domains are a bit different from one another: In the Xbox domain most games are purchased for a non-negligble monetary amount, but in WinGames most games are downloaded at little (or no) monetary cost. We thus expect the implicit binary signal to be much weaker in determining user preferences in WinGames.

LastFM [1] is a publicly available dataset of musical playlists. It gives a natural continuous implicit preference signal- the number of times a particular artist was heard by the user. MovieLens (1M) [6] is a publicly available dataset in which *explicit* ratings are used as a measure of user satisfaction. It was chosen because numerous relevant methods were evaluated using this type of data. All continuous signals were normalized using min-max normalization before training the models. The statistics of each of the datasets are presented in Table 1.

4.2 Evaluation Metrics

We evaluate using several well known metrics, defined below:

RMSE	$\sqrt{\frac{1}{\lvert\mathcal{T}^s\rvert}\sum_{(m,n)\in\mathcal{T}^s}(s_{mn}-\hat{s}_{mn})^2}$
Precision@k	$\frac{1}{\lvert M_\mathcal{T}\rvert}\sum_{m\in M_\mathcal{T}}\sum_{n\in\mathcal{T}_m^s}\frac{\mathbb{I}[\mathsf{Rank}(m,n)<=k]}{k}$
Recall@k	$\frac{1}{\lvert M_\mathcal{T}\rvert}\sum_{m\in M_\mathcal{T}}\sum_{n\in\mathcal{T}_m^s}\frac{\mathbb{I}[\mathsf{Rank}(m,n)<=k]}{\lvert\mathcal{T}_m^s\rvert}$
MPR	$1-\left(\frac{1}{\lvert M_\mathcal{T}\rvert}\sum_{m\in M_\mathcal{T}}\frac{\mathsf{ARank}(m)}{N}\right)$

where $\mathcal{T}^s$ is our test dataset of scores, s_{mn} is the (ground truth) score observed for user m on item n, and $\hat{s}_{mn}$ is the estimated rating predicted by our algorithm for user m on item n. $M_\mathcal{T}$ denotes the set of test users in our test set and $\mathcal{T}_m^s$ denotes the set of test items relevant to user m. $\mathsf{Rank}(m,n)$ denotes the rank of item n by using the model to rank all catalog items for user m. $\mathsf{ARank}(m) = \frac{1}{\lvert\mathcal{T}_m^s\rvert}\sum_{n\in\mathcal{T}_m^s}\mathsf{Rank}(m,n)$ is the average rank for test items associated with user m. $\mathbb{I}[\cdot]$ denotes the indicator function. Note that our method for computing MPR means that higher measurements a correspond to better performance.

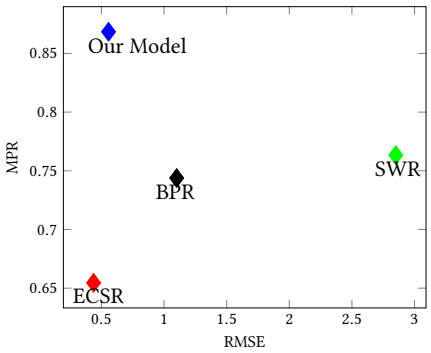
4.3 Baseline Models

Evaluations are provided against several well-known baselines from the literature. For each algorithm, we conducted an exhaustive grid search across the hyper-parameter space, the values reported below represent the best configuration of hyper-parameters for each method.

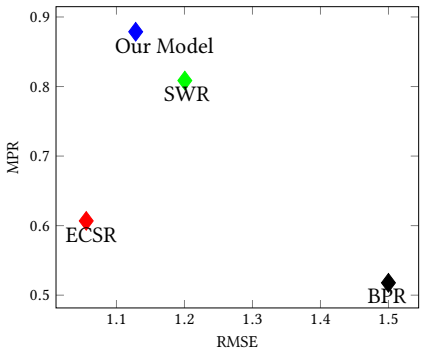
(1) **Our Model**- This is the model proposed in Section 3, which uses a dual task objective and learns the posterior over user/item vector and bias parameters and additional stretch parameters, κ and ψ (which control location and scale of the continuous valued scores) from data.
(2) **Score Weighted Regression (SWR)**- In this baseline we implement the approach of Hu et al. [10], which uses the implicit continuous signal to give additional weight to squared error terms in the optimization objective corresponding to positive data points. Using decomposition properties of the model an efficient algorithm is able to consider all negative data points rather than sampling as in our approach.
(3) **Explicit Continuous Signal Regression (ECSR)**- In this baseline we treat the implicit signals as if they were explicit. That is, we ignore the fact that no continuous signals are available for items the user has not consumed. This baseline is equivalent to our model with $\gamma = 0$ and without sampling of negatives. It is also equivalent to the probabilistic matrix factorization model of Salakhutdinov and Mnih [29].
(4) **Popularity Model (Pop)** - In this naive baseline we simply recommend the most popular items in the training data for all users (excluding those items already installed by the user). This baseline lacks the personalization expected from real-world recommendation systems. However, many retail scenarios exhibit strong skew towards popular item and in such a scenario this baseline can outperform other, more complex, models in ranking metrics.
(5) **Bayesian Personalized Ranking (BPR)** - This baseline implements the approach of Rendle et al. [26], which deals with the one-class nature of recommendation data by sampling negative items for each user. In contrast to our method, BPR optimizes a pairwise ranking objective which seeks parameters that yield a large difference in scores between positive and negative items.

4.4 Results and Discussion

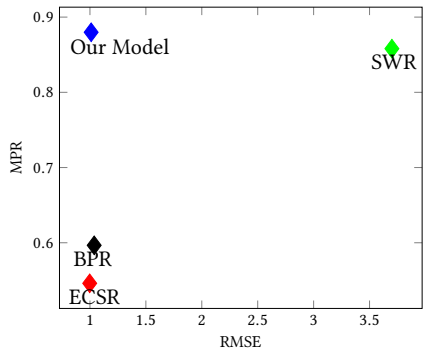
We evaluate our model and baselines on both ranking and prediction metrics. Ranking metrics quantify how well the model was able to predict the purchase of held out items. Prediction metrics quantify



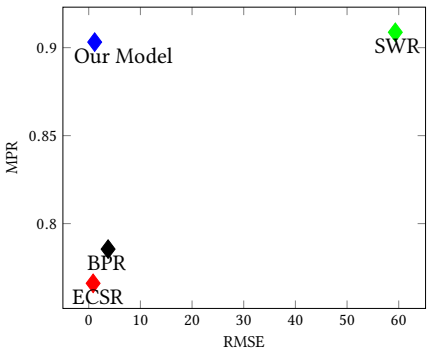
(a) Windows Games Dataset



(b) XBox Games Dataset



(c) Last FM Dataset



(d) MovieLens Dataset

Figure 1: The figure illustrates the tradeoff between ranking performance (measured in MPR) and prediction performance (measured in RMSE). A scenario that wants to maximize customers' post-purchase satisfaction in addition to predicting which items will be purchased requires a model that achieves high MPR and low RMSE.

how well it was able to predict the satisfaction score of held-out items.

Table 2 shows the results of our empirical evaluation. We can see that our method performs well w.r.t to the baselines across the various ranking metrics and datasets. We conclude that using the multi-task objective and corresponding algorithm allows for improved recommendations in terms of showing more items that are expected to be purchased by the user. When it comes to predicting the satisfaction score, the ECSR baseline outperforms all others on RMSE. This result is not surprising as ECSR directly optimizes this metric. The table shows our model is a close second to ECSR on this metric across all datasets.

Our motivating scenarios seek to strike a balance between ranking and prediction performance. Figure 1 visualizes this trade-off by plotting each of our baselines in the space of RMSE vs MPR. A method that achieves a good tradeoff would appear in the upper-left quadrant of this plot. Examining this figure, we can observe that even for datasets where our methods were not able to achieve the top performance in one of the metrics (e.g. MPR in the MovieLens dataset), the achieved tradeoff dominates the other methods.

In addition to summary metrics, which give an average performance picture, it is instructive to examine the impact that making use of the continuous customer satisfaction signal has on user interaction with the recommendation system . Tables 3 and 4 show some anecdotal examples of this kind of user impact. Here, we present the top ranking items that would be shown to a user who purchased a single item. This scenario is sometimes used to compute "Item to Item" similarities. In the first table we see how *Forza Motorsport 6* (a car racing game) is linked to general sports titles if continuous satisfaction scores are not considered but linked more closely to other racing games once we refine the model with this additional signal. Similarly in the Windows Games domain, popular title *Cut the Rope* is grouped with other popular Windows games when satisfaction scores are ignored but considering these scores allows us to discover a finer granularity of similar games, for instance the sequel game *Cut the Rope 2*. These examples suggest that by carefully considering user satisfaction or engagement, the multi-task model learns a more accurate latent space which leads to improved ranking results in the binary task. From this interesting observation and from earlier results we conclude that even when we only care about the binary problem, user satisfaction can serve as "side information".

Finally, we show that making use of the user satisfaction signal leads to better results in recommending unpopular ("cold") items.

	Ranking Metrics (Higher is better)			Error Metric (Lower is better)
Model Variant	P@10	R@10	MPR	RMSE
LastFM Dataset				
BPR	0.011	0.023	0.597	1.038
ECSR	0.007	0.015	0.546	**0.998***
Pop	0.033	0.063	0.681	-
SWR	0.064	**0.148**	0.858	3.700
Our Model	**0.070***	0.145	**0.880***	1.011
MovieLens Dataset				
BPR	0.039	0.016	0.786	3.750
ECSR	0.052	0.025	0.766	**0.852***
Pop	0.101	0.050	0.842	-
SWR	**0.136***	**0.080***	**0.909***	59.343
Our Model	0.121	0.068	0.903	1.154
WinGames Dataset				
BPR	0.007	0.035	0.744	1.101
ECSR	0.003	0.014	0.655	**0.438***
Pop	0.041	**0.227***	0.821	-
SWR	0.031	0.161	0.763	2.850
Our Model	**0.045***	0.218	**0.868***	0.557
Xbox Dataset				
BPR	0.007	0.034	0.518	1.500
ECSR	0.013	0.052	0.607	**1.056***
Pop	0.043	0.214	0.825	-
SWR	0.040	0.196	0.809	1.201
Our Model	**0.048***	**0.249***	**0.879***	1.128

Table 2: A comparison of our proposed model across 4 datasets and several metrics. $*$ denotes statistical significance at the 0.01 level. Our model consistently leads in MPR and comes in a close second to ECSR in RMSE. See Figure 1 for a visualization of the trade-off between ranking and prediction.

Seed	Results w/ Score Signal	Results w/o Score Signal
Forza Motorsport 6	Forza Horizon 2 Project CARS Forza Motorsport 5 Need for Speed The Crew	EA SPORTS FIFA 16 Forza Horizon 2 Just Dance 2016 RIDE Project CARS
Gears of War: Judgment	Gears of War 3 Gears of War: Ultima... Gears of War 2 Halo: Reach Gears of War	Gears of War 3 Gears of War 2 Gears of War: Ultima... Halo: Reach Gears of War
Plants vs. Zombies 2	Plants vs. Zombies Unravel Batman: Arkham Kn... ROBLOX Need for Speed	LEGO Dimensions LEGO Jurassic Worl... Disney Infinity 3.0 ... The LEGO Movie Video... Just Dance 2016

Table 3: Anecdotal item similarities for Xbox Games

Figure 2 plots the MPR when considering only the least popular items in the test set. The plot is cumulative in the sense that bin 2 considers all the items in bin 1 and so on. Examining the plot we conclude that applying our model to consider user satisfaction signals yields improved recommendation performance (as measured by MPR) across all item subsets considered including the regime of very "cold" items.

Seed	Results w/ Score Signal	Results w/o Score Signal
Shopping Cart Hero 3	Jetpack Joyride Endless Skater Block World Nyan Cat The Game Drift Mania Champion...	Guns 4 Hire Jetpack Joyride Offroad Racing Drift Mania Champion... Zombie HQ
Cut The Rope	Angry Gran Run Cut the Rope 2 Jetpack Joyride Agent P Strikes Back... Wo ist mein Wasser? ...	Judge Dredd vs. Zomb... Jetpack Joyride Agent P Strikes Back... Parking Mania Wo ist mein Wasser? ...
FarmVille 2	Logo Quiz Game Cloud Raiders The Tribez 2020: My Country Dragon Mania Legends...	Alles steht Kopf Eri... Crossy Road My Talking Tom Dragon Mania Legends... Cut the Rope 2

Table 4: Anecdotal item similarities for Windows Games

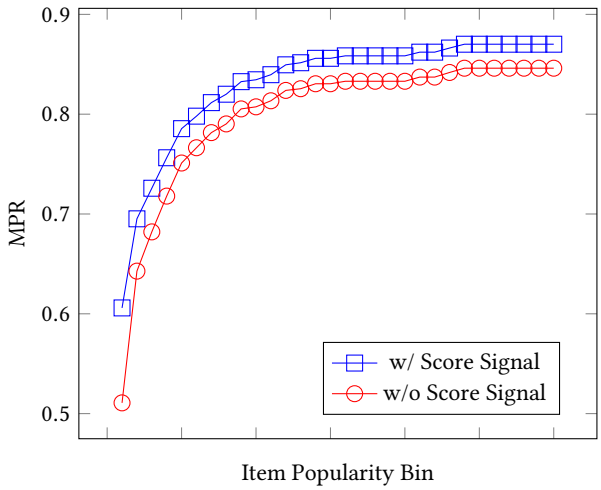


Figure 2: MPR of the model on the Xbox Games dataset as increasingly popular items in the test set are considered. Unpopular or "cold" items are harder to predict. However, considering the user satisfaction score improves recommendation performance on cold items.

5 CONCLUSION

In this work, we propose a multi-task Bayesian model which combines binary purchase or click signals with continuous satisfaction scores. We propose an efficient parallelizable algorithm to estimate the model parameters using the Variational Bayes framework. Our evaluations demonstrate that the proposed model strikes a desired balance between maximizing immediate purchase probability with long term user satisfaction. Interestingly, in some cases the user satisfaction or engagement scores serve as "side information" which help achieve more accurate purchase probabilities. We believe that applying methods similar to those proposed herein, has the potential to improve customer experience for millions of users relying on such recommendation services.

REFERENCES

[1] T. Bertin-Mahieux, D. P. W. Ellis, B. Whitman, and P. Lamere. 2011. The Million Song Dataset. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*.

[2] C. M. Bishop. 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc.

[3] V. Bogina and T. Kuflik. 2017. Incorporating Dwell Time in Session-Based Recommendations with Recurrent Neural Networks. In *RecTemp@RecSys*.

[4] R. Caruana. 1997. Multitask Learning. *Machine Learning* (1997).

[5] M. Gelderman. 1998. The relation between user satisfaction, usage of information systems and performance. *Information & Management* (1998).

[6] F. M. Harper and J. A. Konstan. 2015. The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.* (2015).

[7] X. He, H. Zhang, M. Kan, and T. Chua. 2016. Fast Matrix Factorization for Online Recommendation with Implicit Feedback. In *Proceedings of SIGIR Conference*.

[8] B. Hidasi and D. Tikk. 2012. Enhancing Matrix Factorization Through Initialization for Implicit Feedback Databases. In *Proceedings of CaRR Workshop*.

[9] B. Hidasi and D. Tikk. 2016. Speeding up ALS learning via approximate methods for context-aware recommendations. *Knowledge and Information Systems* (2016).

[10] Y. F. Hu, Y. Koren, and C. Volinsky. 2008. Collaborative Filtering for Implicit Feedback Datasets. In *Proceedings of ICDM Conference*.

[11] T. S. Jaakkola and M. I. Jordan. 1996. A variational approach to Bayesian logistic regression models and their extensions.

[12] Y. Kim and S. Choi. 2014. Scalable Variational Bayesian Matrix Factorization with Side Information. In *Proceedings of AISTATS Conference*.

[13] D. Kluver, T. Nguyen, M. Ekstrand, S. Sen, and J. Riedl. 2012. How Many Bits Per Rating?. In *Proceedings of Recsys Conference*.

[14] B. P. Knijnenburg, M. C. Willemsen, Z. Gantner, H. Soncu, and C. Newell. 2012. Explaining the User Experience of Recommender Systems. *User Modeling and User-Adapted Interaction* (2012).

[15] N. Koenigstein. 2017. Rethinking Collaborative Filtering: A Practical Perspective Based on Real World Insights. In *Proceedings of RecSys Conference*.

[16] N. Koenigstein and U. Paquet. 2013. Xbox Movies Recommendations: VB Matrix Factorization w/ Embedded Feature Selection. In *Proceedings of RecSys Conference*.

[17] Y. Koren, R. M. Bell, and C. Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *IEEE Computer* (2009).

[18] D. J. C. MacKay. 1992. The Evidence Framework applied to Classification Networks. *Neural Computation* (1992).

[19] S. M. McNee, J. Riedl, and J. A. Konstan. 2006. Being Accurate is Not Enough: How Accuracy Metrics Have Hurt Recommender Systems. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems (CHI EA '06)*.

[20] T. Mikolov, I. Sutskever, K. Chen, and others. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS Conference*.

[21] D. Oard and J. Kim. 1998. Implicit feedback for recommender systems. In *Proceedings of the AAAI recommender systems workshop*.

[22] J. Hofman P. Gopalan and D. Blei. 2015. Scalable recommendation with hierarchical poisson factorization. In *Proceedings of UAI Conference*.

[23] R. Pan, Y. Zhou, B. Cao, N. N Liu, R. Lukose, M. Scholz, and Q. Yang. 2008. One-class collaborative filtering. In *Proceedings of ICDM Conference*.

[24] U. Paquet and N. Koenigstein. 2013. One-class collaborative filtering with random graphs. In *Proceedings of WWW Conference*.

[25] I. Pilászy, D. Zibriczky, and D. Tikk. 2010. Fast ALS-based Matrix Factorization for Explicit and Implicit Feedback Datasets. In *Proceedings of RecSys Conference*.

[26] S. Rendle, C. Freudenthaler, Z. Gantner, et al. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *Proceedings of UAI Conference*.

[27] R. Ronen, G. Lavee, and E. Yom-Tov. 2016. Recommendations meet web browsing: enhancing collaborative filtering w/internet logs. In *Proceedings of ICDE*.

[28] S. Ruder. 2017. An Overview of Multi-Task Learning in Deep Neural Networks. *arXiv preprint arXiv:1706.05098* (2017).

[29] R. Salakhutdinov and A. Mnih. 2008. Probabilistic Matrix Factorization. In *Proceedings of NIPS Conference*.

[30] O. Sar-Shalom, N. Koenigstein, U. Paquet, et al. 2016. Beyond Collaborative Filtering: The List Recommendation Problem. In *Proceedings of WWW Conference*.

[31] G. Takács, I. Pilászy, and D. Tikk. 2011. Applications of the Conjugate Gradient Method for Implicit Feedback Collaborative Filtering. In *Proceedings of Recsys*.

[32] M. Volkovs and G. W. Yu. 2015. Effective latent models for binary feedback in recommender systems. In *Proceedings of SIGIR Conference*.

[33] M. Wan and J. McAuley. 2018. Item Recommendation on Monotonic Behavior Chains. In *Proceedings of RecSys Conference*.

[34] Y. Xin and H. Steck. 2011. Multi-value Probabilistic Matrix Factorization for IP-TV Recommendations. In *Proceedings of RecSys Conference*.

[35] X. Yi, L. Hong, E. Zhong, N. N. Liu, and S. Rajan. 2014. Beyond Clicks: Dwell Time for Personalization. In *Proceedings of RecSys Conference*.

[36] P. Yin, Pi. Luo, W.C. Lee, and M. Wang. 2013. Silence is Also Evidence: Interpreting Dwell Time for Recommendation from Psychological Perspective. In *Proceedings of ICDM Conference*.

[37] T. Zhao et al. 2015. Improving latent factor models via personalized feature projection for one class recommendation. In *Proceedings of CIKM Conference*.