A Gradient-based Adaptive Learning Framework for Efficient Personal Recommendation

Yue Ning
Virginia Tech
yning@vt.edu

Yue Shi*
Yahoo Research
yueshi@acm.org

Liangjie Hong
Etsy Inc.
lhong@etsy.com

Huzefa Rangwala
George Mason University
rangwala@cs.gmu.edu

Naren Ramakrishnan
Virginia Tech
naren@cs.vt.edu

ABSTRACT

Recommending personalized content to users is a long-standing challenge to many online services including Facebook, Yahoo, Linkedin and Twitter. Traditional recommendation models such as latent factor models and feature-based models are usually trained for all users and optimize an "average" experience for them, yielding sub-optimal solutions. Although multi-task learning provides an opportunity to learn personalized models per user, learning algorithms are usually tailored to specific models (e.g., generalized linear model, matrix factorization and etc.), creating obstacles for a unified engineering interface, which is important for large Internet companies. In this paper, we present an empirical framework to learn user-specific personal models for content recommendation by utilizing gradient information from a global model. Our proposed method can potentially benefit any model that can be optimized through gradients, offering a lightweight yet generic alternative to conventional multi-task learning algorithms for user personalization. We demonstrate the effectiveness of the proposed framework by incorporating it in three popular machine learning algorithms including logistic regression, gradient boosting decision tree and matrix factorization. Our extensive empirical evaluation shows that the proposed framework can significantly improve the efficiency of personalized recommendation in real-world datasets.

CCS CONCEPTS

•Information systems → Personalization; Recommender systems;

1 INTRODUCTION

Personalized content recommendation plays a key role in online services such as Yahoo, Facebook, LinkedIn and Twitter. User engagements are primarily driven by how content/items are tailored to their personal preferences and interests. In the simplest setting, when only user-item interactions are available, traditional *collaborative filtering* (CF) methods are usually utilized to produce recommendations. State-of-the-art matrix factorization-based CF models (MF) assume that each user has his/her own latent factors and the recommendation score for a particular user-item pair is a dot-product between user- and item-factors, leading to the effect of personalization. In more complex settings, when richer information or features are available, a common practice is to resort to the paradigm of *learning to rank* (LtR) where a feature vector of each user-item interaction is constructed and a regression/classification model is learned from a large amount of historical interaction data to predict recommendation scores [17]. A wide range of predictive models can be used for LtR, including logistic regression [12, 31] (LogReg), gradient boosting decision tree [11, 13] (GBDT) and LambdaMART [8]. In order to achieve personalization in the LtR setting, heavy feature engineering that characterizes user preferences and interests is usually required and few challenges still remain:

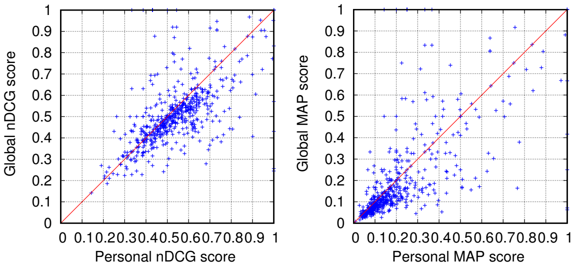


Figure 1: An example of global and personal models. Left figure showcases the nDCG score of users from global (y-axis) and personal (x-axis) models. (Right: MAP score).

- **Alleviate "average" experience for users.** Although both MF and LtR frameworks provide mechanisms to obtain personalization, they cannot easily leverage individual's data and offer a unique experience for each user. For LtR, a particular model is commonly optimized based on a global objective function. It is therefore an "average" model for everyone and remains suboptimal for individual users [7]. In addition, a global model may be biased to features that are frequent across all interaction data, while an individual user may never have interacted with those features in the past. For MF, while user factors capture user preferences to some degree, item factors are global as one item only has the same set of latent features for all users. Thus, similar to LtR, item factors for a particular item are heavily influenced by how many users have interacted with it. If a user happens to interact with a popular item,

[*] Now at Facebook

RecSys'17, August 27–31, 2017, Como, Italy.
© 2017 ACM. 978-1-4503-4652-8/17/08…$15.00.
DOI: http://dx.doi.org/10.1145/3109859.3109909

the user factors might be "polluted" by the item factors despite of his/her own peculiar preferences. In other words, traditional MF is also a global model.

To demonstrate this effect, we compare the performance of a global model and personal models in terms of normalized Discounted Cumulative Gain (nDCG) and Mean-Average-Precision (MAP), on a dataset from a popular online service, shown in Figure 1. All models are logistic regression, optimizing Click-Through-Rate (CTR). The global model is trained on all users' data where personal models are trained on individuals' own data. Each point on the figure represents a user where X-axis and Y-axis represent nDCG/MAP scores of personal models and a global model respectively. The figure shows that personal models exhibit better performance than global models for most users (72.3% and 73.6% points are in the lower triangle area for nDCG and MAP score). For a small portion of users, the global model tends to have better performance. This observation motivates us to explore the possibility of building personal models by borrowing information from a global model.

• **Generic empirical frameworks for different models.** Multi-task learning (MTL) algorithms have been exploited to tackle the issue mentioned above. Under the context of recommendation, to name a few recent efforts, Generalized Linear Mixed Models [34] and Distributed Personalization [25] are proposed to train a global model and personal models simultaneously. Although these approaches provide opportunities for personalization, they are tied to a particular form of model, in this case, linear model, and the proposed learning (e.g., neighborhood-based collaborative filtering [26], variational inference [27] and etc). Similarly, under the context of information retrieval, MTL have been explored for specific models (e.g., linear model [32] and tree-based boosting method [10]) while no generic framework exists for a wide range of models.

• **Distributed model learning and less access of global data.** Even if most MTL formalisms involve different flavors of personal-level modeling, learning algorithms to solve these problems are usually **global**, meaning that learning algorithms need to access all data during the training, unless sophisticated distributed learning systems are utilized [25]. This is particularly a hurdle for the mobile era where user interaction data is gathered naturally on their mobile devices. For a global model, it requires such data being transmitted to the server-side and to re-train the model there, accruing non-trivial communication costs. On the other hand, personal models can be trained and immediately used on mobile devices, without transmitting data or model updates to devices, resulting in better user experiences with less communication burden.

In this paper, we propose a generic empirical framework to address the aforementioned challenges of learning personal models. The proposed framework provides a lightweight yet effective alternative to MTL for personalization. The framework is generic to a wide variety of machine learning models where we demonstrate three instantiations, LogReg, GBDT, and MF, in this paper. Any learning algorithms relying on gradient information can potentially utilize this framework. Thus, the proposed approach offers a systematic solution to build personal models, not tackling the problem just for a particular model. The central idea of the method is to allow each personal model to leverage information from a global model, to a certain extent, based on the richness of each user's own data. We also show parallel learning procedures that enable us to train personal models at scale. Our contributions in this paper can be summarized as follows:

- We present a novel generic empirical framework for building personal models by leveraging a global model through gradient adaptation, offering a more general personalization approach than MTL.
- We demonstrate three different instantiations of the framework: LogReg, GBDT, and MF, with detailed optimization procedures for building personal models.
- We provide an efficient learning paradigm by exploiting a parallel computing scheme for building large number of personal models.
- Through extensive experiments with a variety of datasets, we show the effectiveness of our framework for improving content recommendation performance.

2 RELATED WORK

We provide a brief overview of state-of-the-art approaches [3, 30] for content recommendation systems as well as MTL for personalization.

Generalized Linear Models: One simple way to build a large-scale content recommendation system is to use generalized linear models to predict responses (e.g., clicks, ratings and etc.) for each user-item feature vector. Recent work [34] demonstrated how to achieve it in an industrial scale. Similar models are used in online advertising as well [2]. Note that, as these models induce global optimization problems, special learning algorithms like ADMM, with sophisticated communication schemes are proposed to solve them.

Tree Boosting: GBDT [13, 14] has been proven effective in many machine learning applications. Together with LambdaMART, tree boosting methods show state-of-the-art performance on many LtR and recommendation tasks [6, 8, 35]. Recently, methods [11] are developed to learn global tree models on large-scale datasets.

Matrix Factorization: MF-based models are widely used for recommendation [19] and assume that there exists a latent vector associated with each user and each item. Also, user and item bias terms are exploited in the optimization [15]. Regression-based LFM [1] and Factorization Machines [28] further extended MF to incorporate arbitrary user and item features. In most cases, gradient descent techniques or alternating least squares (ALS) can be applied to solve the optimization problems. Recent work [7] proposed a customized matrix factorization objective for improved recommendations.

Personalization/Multi-Task Learning: The importance of a personalized application extends to many fields such as social networks [20, 22] and ranking system [29] while social network has also been incorporated in personalization [9]. It is necessary for suggesting relevant/interesting results for users and making distributed computation efficient in mobile side, such as Distributed Matrix Factorization [16]. MTL algorithms have been extensively studied to tackle the problem of personalization in both information retrieval (IR) and recommender systems (RecSys). Here, we highlight a few representative papers. In IR, MTL-style linear ranking model [32] (including feature-hashing methods [33]) and tree-based boosting ranking models [10, 18]) have been proposed to adapt global ranking

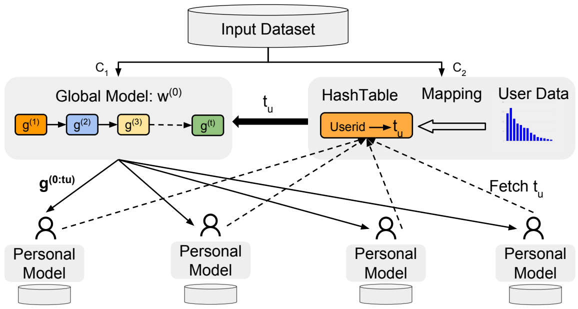


Figure 2: System Framework. Component C_1 trains a global model. Component C_2 generates a hashtable based on users' data distribution. Users request t_u from C_2 and C_1 returns a subsequence of gradients $g^{(0:t_u)}$ to users.

models to user specific ones. In RecSys, early work [26] formulated a MTL based solution for neighborhood-based collaborative filtering methods. Recommending items to users based on *expert* opinions has also been explored [4]. However, most MTL algorithms need to solve global optimization problems. Our propose framework differs from MTL in that personal models can be optimized independently from a global objective function. Also, the generic framework accepts different objective functions.

Curriculum Learning [5] is proposed to solve progressively harder problems, supplying the training examples in a meaningful order may actually lead to improved performance and better convergence. In comparison to Curriculum Learning, our proposed framework focus on efficient learning for personal recommendation models without reordering. It is general and flexible to tailor to different model formalisms.

3 THE PROPOSED FRAMEWORK

A global model is usually learned through an optimization process from all users' data, lacking any personalization. On the contrary, personal models are learned from individuals' data, under the risk of not having enough data to learn stable models. We want to balance these two tradeoffs and design a learning paradigm that adaptively transfers global knowledge to personal models. Figure 2 provides an overview of our framework, that is based on two general assumptions. First, *we assume that both the global model and personal models share the same structure of objective functions*. This assumption is already satisfied by two extreme cases mentioned above. In addition, this assumption also implies that both the global model and personal models can share the same optimization process, which is agnostic to the framework described below. This is a significant advantage over other similar methods such as MTL where optimization algorithms are revised towards more complicated objective functions. Secondly, *we assume the model can be optimized through gradient methods.* This assumption covers a wide range of models including LogReg, GBDT and deep learning models. Note that, our framework does not assume any explicit relationships between global model parameters and personal ones, differing from basic framework of MTL algorithms. This is the key aspect for our proposed approach that allows it to be applicable to multiple learning paradigms.

Algorithm 3.1 Coordination Algorithm

```
input: $C$ (#Groups), $(|D_0|, |D_1|, ..., |D_U|)$, $g^{(0)}, g^{(1)}, ..., g^{(T)}$
output: $f(u, |D_u|) \rightarrow t_u$
procedure SCHEDULER
    $t_1, ..., t_u, ..., t_{|\mathcal{U}|} = 0, u \in \mathcal{U}$
    $d_0, d_1, ..., d_U = \log|D_0|, \log|D_1|, ..., \log|D_U|$
    Sort $(d_0, d_1, ..., d_U)$ in non-ascending order.
    $d_{\max} = \max(d_0, d_1, ..., d_U)$
    $d_{\min} = \min(d_0, d_1, ..., d_U)$
    $s = \frac{d_{\max} - d_{\min}}{C}$
    for $u \in \mathcal{U}$ do
        for $i \in [1, C]$ do
            if $d_u \in [d_{\min} + i * s, d_{\min} + (i+1) * s]$ then
                $p_u = \frac{i}{C}$; $t_u = \lfloor T * p_u \rfloor$; break
    return $\{t_u\}, u \in \mathcal{U}$
```

During the training process for the global model, gradients at each iteration are saved and transferred to a coordinator for later personal adaptation. The coordinator maintains a mapping between users and gradients. When training a personal model for a given user, we initialize it based on a version of global model and a portion of global gradients that correspond to the training data specific to the user. Then, the personal model extends the training procedure solely based on its own data. The overall learning paradigm is described as follows:

$$\widetilde{\boldsymbol{\theta}}_u = \boldsymbol{\theta}^{(0)} - \eta_1 \sum_{t=1}^{t_u - 1} g^{(t)}(\boldsymbol{\theta}) - \eta_2 \sum_{t=t_u}^{T} g^{(t)}(\boldsymbol{\theta}_u)$$

where $\boldsymbol{\theta}$ represents the model parameter, u is the index for one user. In this paradigm, personal models adapt global knowledge by leveraging global gradients, while tailoring themselves to their individual characteristics by concentrating on their own data.

Each user u owns his/her data $\mathcal{D}_u$. The adaptive personalization can be formulated as a learning process, which takes global model gradients and user's data for training a personal model:

$$(\mathcal{G}, \mathcal{D}_u) \xrightarrow{\text{adaptation}} \mathcal{M}_u$$

where $\mathcal{G} = (g^0, g^1, ..., g^T)$ is a series of gradients for updating the global model, and $\mathcal{M}_u$ is the personal model for user u. In the adaptation process, based on user's own dataset $\mathcal{D}_u$, personal models can be learned locally and in parallel. More importantly, users who have insufficient data can still train their models based on global parameters. The generic strategy of our proposed framework is summarized as follows:

(1) Given a dataset of $\mathcal{U}$ users, we learn the global model based on a gradient-based learning algorithm until the iteration T, and save global gradients from each iteration into $\mathcal{G} = (g^0, g^1, ..., g^T)$.

(2) For each user u in $\mathcal{U}$, adaptively distribute a sub-sequence $(g^{0:t_u})$ from global gradients $\mathcal{G}$ based on his/her data $\mathcal{D}_u$.

(3) For each user we locally train a personal model based on the user's data and the adapted global gradients. This step can be implemented in parallel for large-scale data.

Adaptation Mechanism: Assuming, a global model trained with T iterations; for each iteration, model parameters are updated along the opposite direction of their gradient. The gradients are calculated using all the instances in the training set (D^{Tr}). If an

Algorithm 3.2 Adaptive Personal LogReg

```
input: $\mathcal{D} = (\mathbf{x}_{ui}, y_{ui}), \mathbf{x}_{ui} \in \mathcal{R}^K, y_{ui} \in \{1, 0\}, u \in \mathcal{U}$
output: $\widetilde{\mathbf{W}} = \{\mathbf{w}_u\}, u \in \mathcal{U}$
procedure AP-LogReg($\mathcal{D}, \mathbf{W}$)
    $(g^{(0)}, g^{(1)}, ..., g^{(T)}) \leftarrow$ Global Training          ▷ Eq.1
    HashTable $A[u] = t_u \leftarrow$ Scheduler ($\mathcal{D}$)          ▷ Alg. 3.1
    $\widetilde{\mathbf{W}} = \{\}$
    Distribute global gradients $\mathbf{g}^{(0:T)}$
    for user $u \in \mathcal{U}$ do          ▷ Parallel Computing
        Fetch adapted gradients split $t_u$ from $A$
        $\widetilde{\mathbf{w}}_u \leftarrow$ LocalTraining($\mathcal{D}_u, g^{(0:t_u)}$)          ▷ Eq.3
        $\widetilde{\mathbf{W}} = \widetilde{\mathbf{W}} \cup \widetilde{\mathbf{w}}_u$
    return $\widetilde{\mathbf{W}}$
```

Algorithm 3.3 Adaptive Personal GBDT

```
input: $\mathcal{D} = (\mathbf{x}_{ui}, y_{ui}), \mathbf{x}_{ui} \in \mathcal{R}^K, y_{ui} \in \{1, 0\}, u \in \mathcal{U}$
Output: $\{\mathcal{H}\}$
procedure AP-GBDT($\mathcal{D}$)
    Initialize the global tree $h_0$
    $(h^{(0)}, h^{(1)}, ..., h^{(T)}) \leftarrow$ Global Training          ▷ Eq.6
    HashTable $A[u] = t_u \leftarrow$ Scheduler($\mathcal{D}$)          ▷ Alg. 3.1
    $\mathcal{H} = \{\}$
    Distribute global gradient trees $h^{(0:T)}$ to users
    for user $u \in \mathcal{U}$ do          ▷ Parallel Computing
        Fetch adapted gradients split $t_u$ from $A$
        $\widetilde{F}_u \leftarrow$ LocalTraining ($\mathcal{D}_u, h^{(0:t_u)}$)          ▷ Eq.7
        $\mathcal{H} = \mathcal{H} \cup \widetilde{F}_u$
    return $\mathcal{H}$
```

individual has watched a large amount of specific types of movies, it is more reasonable for the system to recommend personalized movie selections than to recommend popular movies that have been watched by other users. However, given a new user without sufficient historical data, it is safe for the system to recommend well known popular items.

It is difficult to determine a good adaptation locally since the local models for users do not have global knowledge of other users. Under this assumption, a centralized coordinator is designed for distributing global gradients to individual users. Each user is assigned an iteration split t_u given the data distribution across all users. The linear mapping of users data size $|D_u|$ to t_u helps coordinate users. For instance, given C groups of users, the linear mapping adaptation sorts user by their data size in descending order. Then according to the group index, it assigns different subsequences of global gradients to different groups such as $t_u = \lfloor T * \frac{\tau_u}{C} \rfloor$ where τ_u is the group index for user u. For the group with the largest dataset, our method assigns early gradients such as $g^{0:1}$ to this group. The detailed description of the adaptation is presented in Algorithm 3.1.

3.1 Methods

We instantiate the proposed framework to three machine learning models, LogReg, GBDT, and MF, which are widely used for content recommendation. We shall point out that each of these models have individual characteristics, which differentiate it from the others. However, the way we extend these models to their personal versions remains consistent. The demonstration of these three formulations allows us to justify the effectiveness of our proposed framework for learning personal models.

•***Adaptive Logistic Regression***. Generalized linear models, such as LogReg, have been exploited in forecasting users' click events (clicking on advertisements or news articles) [23, 24]. In the LogReg method, the probability of an instance $\mathbf{x}_{ui}$ being positive is estimated by a logistic function, which is $\hat{y}_{ui} = \sigma(\mathbf{w}_u^T \mathbf{x}_{ui})$. And it optimizes the log-likelihood with instances as:

$$\min_{\mathbf{w}} L(\mathbf{w}) = \sum_{d=1}^{N} f(\mathbf{w}) + \lambda r(\mathbf{w}) \tag{1}$$

where $f(\mathbf{w})$ is the negative log likelihood function, $r(\mathbf{w})$ is a regularization function and λ is the coefficient. In our framework, besides a global parameter $\mathbf{w}$, each user has his/her own local model $\mathbf{w}_u$. First, we train a global model for all users and save gradients at each iterations as $\mathcal{G} = (g^{(0)}, g^{(1)}, ..., g^{(T)})$. Then we initialize each user's parameter by an adaptive global model as, using SGD:

$$\widetilde{\mathbf{w}}_u^{(0)} = \mathbf{w}^{(0)} - \eta_1 \sum_{t=1}^{t_u - 1} g^{(t)}(\mathbf{w}) \tag{2}$$

where each user has a different adaptive t_u. In this case, users with more examples start with an early global parameter (less global gradient descents) and users with less examples start with a relative late global parameter (more global gradient descents). And we update the personal model parameters:

$$\widetilde{\mathbf{w}}_u^{(T)} = \widetilde{\mathbf{w}}_u^{(0)} - \eta_2 \sum_{t=1}^{T - t_u} g^{(t)}(\mathbf{w}_u) \tag{3}$$

Using L_2 regularization in Eq. 1, Eq. 3 is expanded as:

$$\widetilde{\mathbf{w}}_u^{(T)} = \mathbf{w}^{(0)} - \eta_1 \sum_{t=1}^{t_u - 1} \sum_{d}^{N} \frac{\partial f}{\partial \mathbf{w}} - \eta_2 \sum_{t=t_u}^{T} \sum_{j}^{N_u} \frac{\partial f}{\partial \mathbf{w}_u}$$
$$- \eta_1 \lambda \sum_{t=1}^{t_u - 1} \mathbf{w} - \eta_2 \lambda \sum_{t=t_u}^{T} \mathbf{w}_u \tag{4}$$

With the above updating rule, it is easy to prove that the adaptive personal method based on gradients is a special case of a general multi-task learning (MTL-LogReg) formula for users as follows:

$$L'(\mathbf{w}_u) = \sum_{j}^{N_u} f(y_{uj}, \hat{y}_{uj}) + \frac{\lambda_1}{2} ||\mathbf{w}_u - \mathbf{w}||^2 + \frac{\lambda_2}{2} ||\mathbf{w}_u||^2 \tag{5}$$

Our application is not limited to only multi-task learning problem. It also fits other algorithms such as Gradient Boosting Decision Tree.

•***Adaptive Gradient Boosting Decision Tree***. Tree boosting has been shown to give state-of-the-art results on many standard classification benchmarks [21]. Unlike LogReg and MF methods, training of Gradient Boosting Decision Tree (GBDT) fits a regression tree h for the residual which is the gradient descent of the loss function. We initialize the estimate as $F(x) = \sum_{u=1}^{U} \sum_{j=1}^{N_u} \frac{y_j}{N}$. During each iteration t in stochastic gradient descent, we calculate the gradient descent $-g(x_j) = y_j - F(x_j)$ and a regression tree h^t is fitted to $-g(x_j)$. In this case, we get $F^t = F^{t-1} + \rho h^{(t-1)}$. The global objective function for GBDT is defined as an additive format:

$$L^{(t)} = \sum_{d}^{N} l(y_d, F_d^{(t-1)} + \rho h^{(t)}) + \Omega(h^{(t)})$$
$$= \sum_{d}^{N} l(y_d, F_d^{(0)} + \rho h^{(0:t)}) + \Omega(h^{(t)}) \quad (6)$$

where $l()$ refers to the loss function such as least square loss, or logistic loss. We apply square loss in our experiment. Global gradient boosting trees are assumed to have more complicated structure than local trees because the global feature space is sparse. In order to avoid overfitting for each user, we add more constraints on the local tree fitting in the regularization part than the global tree regularization.

For adaption, we initialize each user's regression tree from the global regression tree as $\widetilde{F}_u^{(0)}$ and for each personal model, we train the local GBDT as

$$\widetilde{F}_u^{(0)} = F^{(0)} + \rho h^{(0:t_u)}, \ \widetilde{F}_u^{(T)} = \widetilde{F}_u^{(0)} + \rho h_u^{(t_u:T)} \quad (7)$$

Basically, the global GBDT adapted to a user $h^{(0:t_u)}$ generates an initial score for each instance in the user's data. Based on the initial score, the personal GBDT extends training regression trees to fit its own data. Through the adaptation, adaptive GBDT balances the global tree and local trees and also avoids tree combination with different structures.

•***Adaptive Matrix Factorization***. Off-the-shelf algorithms for recommendation systems, such as Matrix Factorization (MF), have been widely applied in various applications such as movies, music, and products. MF assumes each user and item is represented by a global latent vector $\mathbf{q}_u$ and $\mathbf{p}_i$ respectively. The global objective function is defined as:

$$\min_{q_*, p_*, b_*} \sum_{u,i} (r_{ui} - \mu - b_u - b_i - \mathbf{q}_u^T \mathbf{p}_i)$$
$$+ \lambda(||\mathbf{q}_u||^2 + ||\mathbf{p}_i||^2 + b_u^2 + b_i^2) \quad (8)$$

where r_{ui} is the rating score of user u on item i, μ is the global average rating score for current item, b_u and b_i are bias variables for user u and item i.

This optimization problem can be solved by gradient descent technique. For a given training instance r_{ui}, we modify the parameters by moving in the opposite direction of the gradient, yielding:

$$e_{ui} = \mathbf{q}_u^T \mathbf{p}_i + \mu + b_u + b_i - r_{ui} \quad (9)$$

$$g(\mathbf{q}_u) = e_{ui}\mathbf{p}_i + \lambda \mathbf{q}_u, \ g(\mathbf{p}_i) = e_{ui}\mathbf{q}_u + \lambda \mathbf{p}_i \quad (10)$$

$$g(b_u) = e_{ui} + \lambda b_u, \ g(b_i) = e_{ui} + \lambda b_i \quad (11)$$

Observing the SGD solution, the user latent vectors help improve personalization in recommendation. The learning process for updating $\mathbf{q}_u$ is shaped by the global item vector $\mathbf{p}_i$ (Eq.??). Furthermore, the item latent vector is shaped by other users (Eq.10).

For our proposed adaptive learning paradigm, we update the personal user latent models $(\widetilde{\mathbf{q}}_u, \widetilde{b}_u)$ by their own dataset and adaptively use global gradients $g(\mathbf{q}_u), g(\mathbf{p}_i)$ to each user.

$$\widetilde{\mathbf{q}}_u^{(0)} = \mathbf{q}_u^{(0)} - \eta_1 \sum_{t=0}^{t_u} g^{(t)}(\mathbf{q}_u), \widetilde{\mathbf{q}}_u^{(T)} = \widetilde{\mathbf{q}}_u^{(0)} - \eta_2 \sum_{t=0}^{T-t_u} g^{(t)}(\widetilde{\mathbf{q}}_u) \quad (12)$$

$$\widetilde{b}_u^{(0)} = b_u^{(0)} - \eta_1 \sum_{k=0}^{t_u} g^{(t)}(b_u), \widetilde{b}_u^{(T)} = \widetilde{b}_u^{(0)} - \eta_2 \sum_{t=0}^{T-t_u} g^{(t)}(\widetilde{b}_u) \quad (13)$$

Algorithm 3.4 Adaptive Personal MF

```
input: $\mathcal{D} = (u, i, r), u \in \mathcal{U}, v \in \mathcal{V}, r \in \mathcal{R}^+$
output: $\{\widetilde{\mathbf{Q}}, \widetilde{\mathbf{b}}\}$
procedure AP-MF($\mathcal{D}$)
    $(\mathbf{P}^{(0)}, \mathbf{P}^{(1)}, ..., \mathbf{P}^{(T)}) \leftarrow$ Global Training                ▷ Eq.8
    $(\mathbf{Q}^{(0)}, \mathbf{Q}^{(1)}, ..., \mathbf{Q}^{(T)}) \leftarrow$ Global Training                ▷ Eq.8
    HashTable $A[u] = t_u \leftarrow$ Scheduler($\mathcal{D}$)                   ▷ Alg. 3.1
    $\widetilde{\mathbf{Q}} = \{\}, \widetilde{\mathbf{b}} = \{\}$
    Distribute global gradients $g(\mathbf{q}_u)^{(0:T)}, g(\mathbf{P})^{(0:T)}$
    for user $u \in \mathcal{U}$ do                              ▷ Parallel Computing
        Fetch adapted gradients split $t_u$ from $A$
        $\widetilde{\mathbf{q}}_u, \widetilde{b}_u \leftarrow$ LocalTraining $(\mathcal{D}_u, g(\mathbf{q})^{(0:t_u)}, g(\mathbf{P}))$
                                                                 ▷ Eq.12, 13
        $\widetilde{\mathbf{Q}} = \widetilde{\mathbf{Q}} \cup \widetilde{\mathbf{q}}_u, \widetilde{\mathbf{b}} = \widetilde{\mathbf{b}} \cup \widetilde{b}_u$
    return $(\widetilde{\mathbf{Q}}, \widetilde{\mathbf{b}})$
```

In the global training, we observe that each user latent vector $\mathbf{q}_u$ and item latent vector $\mathbf{p}_i$ are influenced by global instances. In other words, they are influenced by other users' ratings. Both user and item latent vectors are learned and updated through multiple iterations.

In the adaptive personal training, we trace back to an early user latent vector $\widetilde{\mathbf{q}}_u^{(0)}$ which means we adapt global gradients $g^{(0:t_u)}(\mathbf{q}_u)$ and we fix the global item latent vectors $\mathbf{p}_i^{(t)}$. Then we update the user latent vector $\widetilde{\mathbf{q}}_u$ by its own dataset which means the user latent vector is less influenced by the global items and more influenced by its personal rating instances.

In the MF experiment, user and item latent vectors are not easy to be learned if only local data is been exploited. In this case, for comparison, we setup a situation for local-MF that a global MF model is first trained for t iterations and then we send its learned parameters to all users. Local models at user level are initialized by the same global information but they keep updating their user latent vectors locally for multiple runs till convergence.

3.2 Properties

We point out a few properties of the proposed framework in order to understand its usefulness.

•**Generality:** The framework is generic to a variety of machine learning models that can be optimized by gradient-based approaches. Although in the following of the paper our discussion is limited within three use cases, it is straightforward to apply this framework to other models with only a few configuration changes.

•**Extensibility:** The framework is extensible to be used for more sophisticated use cases. First, it is straightforward to extend our framework to incorporate different gradient computation, such as the gradient from a single example or from a mini-batch. Second, it is convenient to extend this framework to the online learning setting, for which we can maintain a window (with fixed size) of gradients from the global model, rather than keeping all the gradients as in the offline setting. Note that in this paper, we focus our presentation and experimentation on the offline setting, while leaving the implementation of this framework for online learning to future work.

Table 1: Dataset Statistics

News Portal	
# users	54845
# features	351
# click events	2,378,918
# view events	26,916,620
avg # click events per user	43
avg # events per user	534

Movie Ratings		
	Netflix	Movielens
# users	478920	1721
# items	17766	3331
sparsity	0.00942	0.039

•**Scalability:** In this framework, the training process of a personal model for one user is independent of all the other users. As such, we can deploy the training process for all the users in an embarassingly parallel manner, making our framework highly scalable. This is useful in the case of mobile device users where the training of personal models can happen on the client side. The training procedure of a global model can be in batch-mode and offline while the training procedure of personal models is parallelizable and online.

The proposed framework is more flexible than conventional MTL algorithms. First, we do not explicit define relationships between global model parameters and personal ones. Second, our approach is straightforward and easy to implement for a wide range of use cases whereas MTL is usually tied to a specific model formalism.

4 EXPERIMENTAL EVALUATION

In this section, we conduct a series of experiments to evaluate the effectiveness of the proposed framework for improving personal models for content recommendation. We first describe the experimental settings including the datasets, evaluation metrics and protocols. Then, we demonstrate the performance of our framework for learning adaptive personal models for LogReg, GBDT, and MF.

4.1 Datasets

The datasets used in our experiments are based on two different domains. One dataset was collected from a main-stream news portal site, which serves news feeds to millions of users. We sampled the event logs from one month in 2016 with user id and article id anonymized. Key statistics of this dataset are presented in Table 1. In this dataset, each example represents an event that a user clicked a news article, or a user viewed but not clicked an article. Note that we take the user click action as a positive label, and the view-but-not-click action as a negative label. Also note that each event is associated with a feature vector. Since this dataset is suitable for ranking content items, we use this dataset to evaluate the LogReg and GBDT-based personal models, which are the major ranking models in real-world applications.

To evaluate the personal models for MF, we use the datasets from the movie domain, i.e., MovieLens 1M dataset and Netflix dataset, which are conventional benchmark datasets in the research community of recommender systems. The statistics of the two datasets are summarized in Table 1.

4.2 Comparative Approaches

We summarize the objective functions for the comparison methods in Table 2. Global models are trained on all users' data. Local models are learned locally on per user's data. MTL represents the approach that users models are averaged by a global parameter. Note that for local-MF, the user vector and item vector are learned globally. We initialize each user model by a trained global model $(\widetilde{\mathbf{q}}_u, \widetilde{\mathbf{p}}_i, \widetilde{b}_u, \widetilde{b}_i)$ and then we train a localized MF model per user by only using each user's data. The result is evaluated per user.

4.3 Protocol and Metrics

In our experiments, each dataset is split into a training set and a test set. Specifically, for each user we randomly select 80% of his data points to be used in the training set, and leave the rest 20% into the test set. We also use a small fraction of the training set as a validation set to tune the parameters involved in different models. Since LogReg and GBDT are specifically used as ranking models on the news dataset, we adopt the standard ranking metrics from the information retrieval community, i.e., Mean Average Precision (MAP), Mean Reciprocal Rank (MRR), Area Under Curve (AUC), and normalized Discounted Cumulative Gain (nDCG), to evaluate their ranking performance. For evaluating MF-related models on the MovieLens and Netflix datasets, we adopt conventional metrics (as used in the Netflix Prize competition and other public contests), Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE).

In addition to the overall performance, we particularly focus on the performance for different user groups in order to understand the effectiveness of personal models. On the news dataset, we divide users into seven groups based on their data size, indexed from group 1 (smallest) to group 7 (largest). For example, users in group 1 have data points in range $(0, 10)$ and users in group 7 have the range of $(3162, 20000)$. For the MovieLens and Netflix datasets, we sample and then cluster users into five groups and the group with the largest data points in the range of $(200, \infty)$.

4.4 Results

Through the experimental evaluation, we aim to address the following specific research questions: 1) Can we learn personal models from the proposed framework to improve recommendation performance over the global-, local- model and the personalized MTL-models (Sec.4.2)? 2) Can we learn personal models that are adaptive to the characteristics of individual users? 3) Is the proposed framework generally effective for content recommendation across different models and use cases?

4.4.1 *Evaluation for Logistic Regression.*

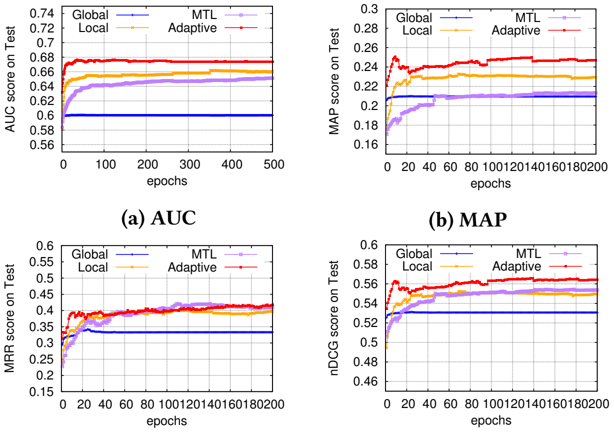
Performance Across User Groups. Figure 3 shows the AUC, MRR, MAP and nDCG (averaging across users) scores on test dataset for different LogReg models with varying training epochs. The proposed adaptive LogReg models achieve higher scores on AUC, MAP, MRR, and nDCG with relatively fewer epochs. In terms averaged AUC, MAP, MRR, and nDCG scores across groups, global LogReg models perform the worst compared to other methods.

Performance with Changing # of Training Samples. We compare the performance in terms of AUC score for MTL-LogReg (Eq. 5) and Adaptive-LogReg with # of training samples from 20% to 100% in Figure 4a. Adaptive-LogReg performs better than Global-LogReg, Local-LogReg, and MTL-LogReg in AUC.

4.4.2 *Evaluation for Gradient Boosting Decision Tree.*

Table 2: Objective functions for different methods.

Model	LogReg	GBDT	MF
Global	$\sum_{d=1}^{N} f(\mathbf{w}) + \lambda\|\|\mathbf{w}\|\|_2^2$	$\sum_{d}^{N} l(y_d, F_d^{(0)} + \rho h^{(0:t)}) + \Omega(h^{(t)})$	$\sum_{u,i}(r_{ui} - \mu - b_u - b_i - \mathbf{q}_u^T \mathbf{p}_i) + \lambda(\|\|\mathbf{q}_u\|\|^2 + \|\|\mathbf{p}_i\|\|^2 + b_u^2 + b_i^2)$
Local	$\sum_{j=1}^{N_u} f(\mathbf{w}_u) + \lambda\|\|\mathbf{w}_u\|\|_2^2$	$\sum_{j}^{N_u} l(y_j, F_j^{(0)} + \rho h^{(0:t)}) + \Omega(h^{(t)})$	$\sum_{i \in N_u}(r_{ui} - \mu - \widetilde{b}_u - \widetilde{b}_i - \widetilde{\mathbf{q}}_u^T \widetilde{\mathbf{p}}_i) + \lambda(\|\|\widetilde{\mathbf{q}}_u\|\|^2 + \|\|\widetilde{\mathbf{p}}_i\|\|^2 + \widetilde{b}_u^2 + \widetilde{b}_i^2)$
MTL	$\sum_{j}^{N_u} f(\mathbf{w}_u) + \frac{\lambda_1}{2}\|\|\mathbf{w}_u - \mathbf{w}\|\|^2 + \frac{\lambda_2}{2}\|\|\mathbf{w}_u\|\|^2$	-	global+$\lambda_2[(\mathbf{q}_u - \mathbf{q})^2 + (\mathbf{p}_i - \mathbf{p})^2 + (b_u - A_u)^2 + (b_i - A_i)^2]$



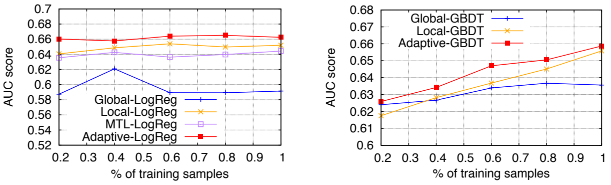
(a) AUC

(b) MAP

(c) MRR

(d) nDCG

Figure 3: AUC, MAP, MRR and nDCG scores of LogReg models with X-axis as the epochs in the SGD algorithm.



(a) LogReg

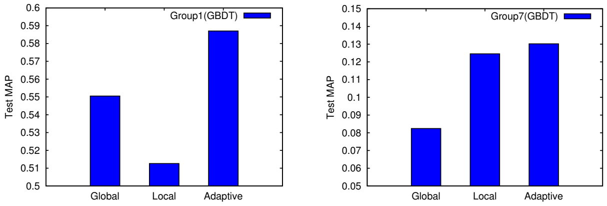
(b) GBDT

Figure 4: Comparison of Global, Local, and Adaptive models with varying # of training examples.

Performance Across User Groups. Given the space limitation, we only show the MAP score for the groups of users with least data and most data for GBDT models in Figure 5. We observe that adaptive-GBDT outperform both global and local GBDT models in terms of MAP for all groups of users.

Performance with Changing # of Training Samples. We compare the performance in terms of AUC score for Global-GBDT, Local-GBDT, and Adaptive-GBDT with # of training samples from 20% to 100% in Figure 4b. On average of AUC, Adaptive-GBDT performs better than other methods. We also observe that with the increase of training samples, GBDT based methods tend to perform better while LogReg methods achieve relatively stable scores.

Overall Comparison of LogReg and GBDT. In Table 3, we show the average MAP, MRR, and AUC scores across users with respect to different iterations for logistic regression adaptation and GBDT. Global-LogReg is not shown here because it performs the worst as in Fig. 4a. With relatively less iterations, adaptive method outperforms global, local, and MTL in terms of all metrics. Specifically, for AUC score, adaptive-LogReg achieves 4.4% and 2.9% rise compared to the best MTL and local-LogReg models. Adaptive-GBDT outperforms the best globa-GBDT by 1.6% with p-value (=0.003) from a paired t-test experiment.



(a) Group 1

(b) Group 7

Figure 5: MAP Comparison of Group 1 (least) and Group 7 (most) for GBDT methods.

4.4.3 Evaluation for Matrix Factorization.

Overall Comparison. In Table 4, we show the average RMSE and MAE scores across groups of users with respect to different values of K (the dimension of latent vectors). In this experiment, global-MF denotes the baseline MF algorithm on the rating data of all users. However, it is not natural to train a local model for each user since the latent vectors **q**, **p** are updated using a global dataset. Thus we use a different setting as "Local" to denote that we train a global MF model and save the global user latent vectors for each iteration ($\mathbf{U}^{(0)}$ to $\mathbf{U}^{(T)}$) until it converges. Then we use the last model parameter $\mathbf{U}^{(T)}$ to initialize each user model for a localized training process. "Adaptive" indicates the proposed method in the algorithm in Section 3. MTL-MF model is a generalized MTL method on MF. We introduce global parameters (**p**, **q** and global bias) and regularize the user parameters to be close to global parameter as in the LogReg models. With respect to different choices of K (5, 10, 20), adaptive-MF achieves the best performance (lowest RMSE and MAE values) compared to global-MF, local-MF, and MTL-MF. It outperforms global-MF, local-MF, and MTL-MF by 3.7%, 0.8% and 1.4% (p-values are 0.008, 0.04, 0.002 from paired t-tests) in terms of RMSE(K=5). MTL-MF did not show good performance when ratings on items have a large variance. The model forced the learned item vectors to be similar to the global one.

Performance Across User Groups. Figure 6 shows the quartile analysis of the group level RMSE and MAE for different MF models and for two datasets. We group users in descending order based on their dataset sizes. Comparing to global-MF, both local and adaptive-MF achieve better predictive performance in terms of these two metrics.

Table 3: Performance comparison based on MAP, MRR, AUC and nDCG for LogReg and GBDT. Each value is calculated from the average of 10 runs with standard deviation.

	MTL-LogReg					**Global**-GBDT			
T	MAP	MRR	AUC	nDCG	#Trees	MAP	MRR	AUC	nDCG
20	0.1937(8e-5)	0.3493(4e-4)	0.6226(8e-5)	0.5341(8e-5)	20	0.2094(1e-3)	0.3617(2e-3)	0.6290(1e-3)	0.5329(6e-4)
50	0.2096(4e-5)	0.3834(2e-4)	0.6376(8e-5)	0.5493(4e-5)	50	0.2137(1e-3)	0.3726(1e-3)	0.6341(1e-3)	0.5372(6e-4)
100	0.2104(5e-6)	0.4059(6e-5)	0.6417(9e-7)	0.5512(5e-6)	100	0.2150(8e-3)	0.3769(1e-3)	0.6356(8e-4)	0.5392(6e-4)
200	0.2132(1e-6)	0.4089(7e-5)	0.6459(2e-6)	0.5538(3e-6)	200	0.2161(5e-4)	0.3848(1e-3)	0.6412(6e-4)	0.5415(5e-4)
	Local-LogReg					**Local**-GBDT			
T	MAP	MRR	AUC	nDCG	#Trees	MAP	MRR	AUC	nDCG
20	0.2221(1e-3)	0.3681(6e-3)	0.6469(8e-4)	0.5443(1e-3)	20	0.2262(2e-3)	0.4510(5e-3)	0.6344(3e-3)	0.5604(2e-3)
50	0.2289(2e-3)	0.3890(4e-3)	0.6521(1e-3)	0.5485(1e-3)	50	0.2319(2e-3)	0.4446(4e-3)	0.6505(2e-3)	0.5651(2e-3)
100	0.2308(1e-3)	0.3938(4e-3)	0.6543(9e-4)	0.5504(1e-3)	100	0.2328(1e-3)	0.4465(5e-3)	0.6558(2e-3)	0.5651(2e-3)
200	0.2294(1e-3)	0.3959(2e-3)	0.6555(8e-4)	0.5495(1e-3)	200	0.2322(2e-3)	0.4431(2e-3)	0.6566(1e-3)	0.5649(1e-3)
	Adaptive-LogReg					**Adaptive**-GBDT			
T	MAP	MRR	AUC	nDCG	#Trees	MAP	MRR	AUC	nDCG
20+50	0.2343(3e-4)	0.3919(5e-4)	0.6633(8e-5)	0.5542(2e-4)	20+50	**0.2343**(2e-3)	0.4474(4e-3)	0.6555(2e-3)	0.5661(2e-3)
50+50	0.2353(1e-3)	0.4013(8e-3)	0.6623(7e-4)	0.5559(1e-3)	50+50	0.2325(2e-3)	0.4472(1e-4)	0.6561(8e-4)	**0.5666**(6e-4)
10+200	**0.2454**(5e-5)	**0.4160**(4e-4)	**0.6744**(5e-5)	**0.5643**(4e-5)	10+100	0.2329(2e-3)	0.4423(3e-3)	**0.6587**(1e-3)	0.5650(3e-3)

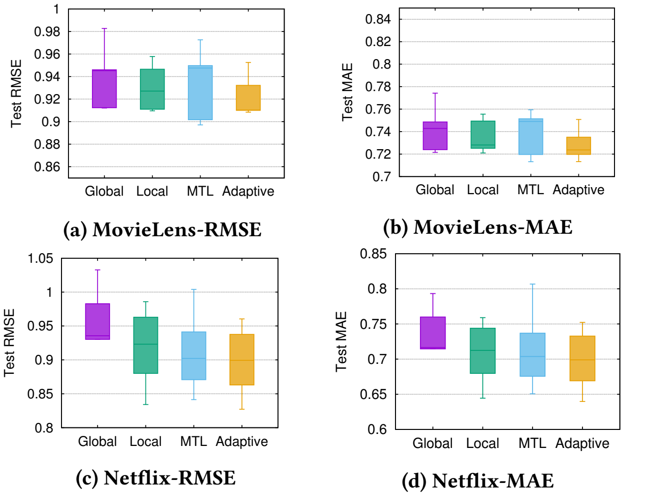
Table 4: Performance comparison based on RMSE, MAE for MF. NF refers to Netflix and ML refers to MovieLens.

NF	**Global**-MF		**Local**-MF		**MTL**-MF		**Adaptive**-MF	
K	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
5	0.9343	0.7336	0.907	0.7012	0.9119	0.7148	**0.8991**	**0.6988**
10	0.9215	0.7168	0.8997	0.6933	0.9328	0.7245	**0.8912**	**0.6910**
20	0.9218	0.7166	0.8999	0.6920	0.9208	0.7144	**0.8907**	**0.6892**
ML	**Global**-MF		**Local**-MF		**MTL**-MF		**Adaptive**-MF	
K	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
5	0.9309	0.7338	0.9254	0.7283	0.9337	0.7365	**0.9188**	**0.724**
10	0.9337	0.7357	0.9253	0.7269	0.9352	0.7401	**0.9178**	**0.7223**
20	0.9348	0.7386	0.9263	0.7285	0.9475	0.751	**0.9184**	**0.7236**

4.4.4 Summary. Based on all the experimental results across three different applications, we do observe that the proposed framework allows us to effectively and efficiently build personal models that lead to improved recommendation performance over either the global model or the local model. This observation confirms a positive answer to our first research question. Based on the results across different user groups in each scenario, we also observe that the proposed framework can adaptively learn personal models by exploiting the global gradients according to individual's characteristic. This observation allows us to answer our second research question affirmatively. Finally, as our experiments demonstrate the usefulness of our framework across a wide scope, in terms of both model classes and application domains, it provides a solid evidence for us to give a positive answer to our third and last research question.

5 CONCLUSION

In this paper we sought to improve users' experience in personal recommendation where users have varying amount of historical data. We presented a general purpose framework for learning personal models based on adapting the popular gradient descent optimization techniques. We instantiate our proposed framework with three different algorithms and provide training procedures for each of the cases. Through extensive empirical evaluation, we demonstrate the strengths of our proposed framework in terms of predictive performance on real world datasets. In the future, we plan to implement an online learning framework for learning personal models through gradient adaptation. Also, we are interested in studying the effects of adaptive learning rates for users in personal models.



(a) MovieLens-RMSE

(b) MovieLens-MAE

(c) Netflix-RMSE

(d) Netflix-MAE

Figure 6: RMSE and MAE scores with respect to Global, Local and Adaptive MF models across 5 groups of users.

ACKNOWLEDGMENTS

We thank Yahoo Research for its support on this work with an internship and DAC in Virginia Tech on its continuing support. We also thank Ting Chen, Qian Zhao and Qingyun Wu for their helpful discussions. The work was partially supported by NSF Grant No. 1447489.

REFERENCES

[1] Deepak Agarwal and Bee-Chung Chen. Regression-based Latent Factor Models. In *Proceedings of KDD 2009*. 19–28. DOI:http://dx.doi.org/10.1145/1557019.1557029

[2] Deepak Agarwal, Bo Long, Jonathan Traupman, Doris Xin, and Liang Zhang. LASER: A Scalable Response Prediction Platform for Online Advertising. In *Proceedings of WSDM 2014*. 173–182. DOI:http://dx.doi.org/10.1145/2556195.2556252

[3] Xavier Amatriain and Deepak Agarwal. 2016. Tutorial: Lessons Learned from Building Real-life Recommender Systems. In *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys '16)*. ACM, New York, NY, USA, 433–433. DOI:http://dx.doi.org/10.1145/2959100.2959194

[4] Xavier Amatriain, Neal Lathia, Josep M. Pujol, Haewoon Kwak, and Nuria Oliver. 2009. The Wisdom of the Few: A Collaborative Filtering Approach Based on Expert Opinions from the Web. In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '09)*. ACM, New York, NY, USA, 532–539. DOI:http://dx.doi.org/10.1145/1571941.1572033

[5] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum Learning. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML '09)*. ACM, New York, NY, USA, 41–48. DOI:http://dx.doi.org/10.1145/1553374.1553380

[6] J. Bennett and S. Lanning. 2007. The Netflix Prize. In *Proceedings of the KDD Cup Workshop 2007*. 3–6.

[7] Alex Beutel, Ed H. Chi, Zhiyuan Cheng, Hubert Pham, and John Anderson. 2017. Beyond Globally Optimal: Focused Learning for Improved Recommendations. In *Proceedings of the 26th International Conference on World Wide Web (WWW '17)*.

[8] Christopher J. C. Burges. 2010. *From RankNet to LambdaRank to LambdaMART: An Overview*. Technical Report. Microsoft Research. http://research.microsoft.com/en-us/um/people/cburges/tech_reports/MSR-TR-2010-82.pdf

[9] Allison J.B. Chaney, David M. Blei, and Tina Eliassi-Rad. 2015. A Probabilistic Model for Using Social Networks in Personalized Item Recommendation. In *Proceedings of the 9th ACM Conference on Recommender Systems (RecSys '15)*. ACM, New York, NY, USA, 43–50. DOI:http://dx.doi.org/10.1145/2792838.2800193

[10] Olivier Chapelle, Pannagadatta Shivaswamy, Srinivas Vadrevu, Kilian Weinberger, Ya Zhang, and Belle Tseng. 2011. Boosted multi-task learning. *Machine Learning* 85, 1 (2011), 149–173. DOI:http://dx.doi.org/10.1007/s10994-010-5231-6

[11] Tianqi Chen and Carlos Guestrin. XGBoost: A Scalable Tree Boosting System. In *Proceedings of KDD 2016*. 785–794.

[12] David R. Cox. 1958. The regression analysis of binary sequences (with discussion). *J Roy Stat Soc B* 20 (1958), 215–242.

[13] Jerome H. Friedman. 2000. Greedy Function Approximation: A Gradient Boosting Machine. *Annals of Statistics* 29 (2000), 1189–1232.

[14] Jerome H. Friedman. 2002. Stochastic gradient boosting. *Computational Statistics & Data Analysis* 38, 4 (2002), 367 – 378. DOI:http://dx.doi.org/10.1016/S0167-9473(01)00065-2 Nonlinear Methods and Data Mining.

[15] Simon Funk. 2006. Netflix Update: Try this at Home. (2006).

[16] Bikash Joshi, Franck Iutzeler, and Massih-Reza Amini. 2016. Asynchronous Distributed Matrix Factorization with Similar User and Item Based Regularization. In *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys '16)*. ACM, New York, NY, USA, 75–78. DOI:http://dx.doi.org/10.1145/2959100.2959161

[17] Alexandros Karatzoglou, Linas Baltrunas, and Yue Shi. 2013. Learning to Rank for Recommender Systems. In *Proceedings of the 7th ACM Conference on Recommender Systems (RecSys '13)*. ACM, New York, NY, USA, 493–494. DOI:http://dx.doi.org/10.1145/2507157.2508063

[18] Gunhee Kim and Eric P. Xing. Time-sensitive Web Image Ranking and Retrieval via Dynamic Multi-task Regression. In *Proceedings of WSDM 2013*. 163–172. DOI:http://dx.doi.org/10.1145/2433396.2433417

[19] Yehuda Koren. Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model. In *Proceedings of KDD 2008*. 426–434. DOI:http://dx.doi.org/10.1145/1401890.1401944

[20] Roy Levin, Hassan Abassi, and Uzi Cohen. 2016. Guided Walk: A Scalable Recommendation Algorithm for Complex Heterogeneous Social Networks. In *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys '16)*. ACM, New York, NY, USA, 293–300. DOI:http://dx.doi.org/10.1145/2959100.2959143

[21] Ping Li. Robust LogitBoost and Adaptive Base Class (ABC) LogitBoost. In *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence (UAI2010)*. 302–311. https://dslpitt.org/uai/displayArticleDetails.jsp?mmnu=1&smnu=2&article_id=2158&proceeding_id=26

[22] Peter Lofgren, Siddhartha Banerjee, and Ashish Goel. 2016. Personalized PageRank Estimation and Search: A Bidirectional Approach. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining (WSDM '16)*. ACM, New York, NY, USA, 163–172. DOI:http://dx.doi.org/10.1145/2835776.2835823

[23] H. Brendan McMahan. 2011. Follow-the-Regularized-Leader and Mirror Descent: Equivalence Theorems and L1 Regularization. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)*.

[24] H. Brendan McMahan, Gary Holt, D. Sculley, Michael Young, Dietmar Ebner, Julian Grady, Lan Nie, Todd Phillips, Eugene Davydov, Daniel Golovin, Sharat Chikkerur, Dan Liu, Martin Wattenberg, Arnar Mar Hrafnkelsson, Tom Boulos, and Jeremy Kubica. Ad Click Prediction: a View from the Trenches. In *Proceedings of KDD 2013*. 1222–1230.

[25] Xu Miao, Chun-Te Chu, Lijun Tang, Yitong Zhou, Joel Young, and Anmol Bhasin. Distributed Personalization. In *Proceedings of KDD 2015*. 1989–1998. DOI:http://dx.doi.org/10.1145/2783258.2788626

[26] Xia Ning and George Karypis. 2010. Multi-task learning for recommender systems. *Journal of Machine Learning Research* 13 (2010), 269–284.

[27] Alexandre Passos, Piyush Rai, Jacques Wainer, and Hal Daumé III. Flexible Modeling of Latent Task Structures in Multitask Learning. In *Proceedings of ICML 2012*. 1103–1110.

[28] Steffen Rendle. 2012. Factorization Machines with libFM. *ACM Trans. Intell. Syst. Technol.* 3, 3, Article 57 (May 2012), 22 pages.

[29] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence (UAI '09)*. AUAI Press, Arlington, Virginia, United States, 452–461. http://dl.acm.org/citation.cfm?id=1795114.1795167

[30] Dhaval Shah, Pramod Koneru, Parth Shah, and Rohit Parimi. 2016. News Recommendations at Scale at Bloomberg Media: Challenges and Approaches. In *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys '16)*. ACM, New York, NY, USA, 369–369. DOI:http://dx.doi.org/10.1145/2959100.2959118

[31] S. H. Walker and D. B. Duncan. 1967. Estimation of the probability of an event as a function of several independent variables. *Biometrika* 54, 1 (June 1967), 167–179. http://view.ncbi.nlm.nih.gov/pubmed/6049533

[32] Hongning Wang, Xiaodong He, Ming-Wei Chang, Yang Song, Ryen W. White, and Wei Chu. Personalized Ranking Model Adaptation for Web Search. In *Proceedings of SIGIR 2013*. 323–332. DOI:http://dx.doi.org/10.1145/2484028.2484068

[33] Kilian Weinberger, Anirban Dasgupta, John Langford, Alex Smola, and Josh Attenberg. Feature Hashing for Large Scale Multitask Learning. In *Proceedings of ICML 2009*. 1113–1120. DOI:http://dx.doi.org/10.1145/1553374.1553516

[34] XianXing Zhang, Yitong Zhou, Yiming Ma, Bee-Chung Chen, Liang Zhang, and Deepak Agarwal. GLMix: Generalized Linear Mixed Models For Large-Scale Response Prediction. In *Proceedings of KDD 2016*. 363–372. DOI:http://dx.doi.org/10.1145/2939672.2939684

[35] Zhaohui Zheng, Hongyuan Zha, Tong Zhang, Olivier Chapelle, Keke Chen, and Gordon Sun. A General Boosting Method and its Application to Learning Ranking Functions for Web Search. In *Proceedings of NIPS 2008*. 1697–1704.