Predicting Response in Mobile Advertising with Hierarchical Importance-Aware Factorization Machine

Richard J. Oentaryo
Living Analytics Research Centre
Singapore Management University
80 Stamford Road, Singapore
roentaryo@smu.edu.sg

Ee-Peng Lim
Living Analytics Research Centre
Singapore Management University
80 Stamford Road, Singapore
eplim@smu.edu.sg

Jia-Wei Low
Living Analytics Research Centre
Singapore Management University
80 Stamford Road, Singapore
jwlow@smu.edu.sg

David Lo
Living Analytics Research Centre
Singapore Management University
80 Stamford Road, Singapore
davidlo@smu.edu.sg

Michael Finegold
Department of Statistics
Carnegie Mellon University
229K Baker Hall, Pittsburgh, PA
mfinegol@andrew.cmu.edu

ABSTRACT

Mobile advertising has recently seen dramatic growth, fueled by the global proliferation of mobile phones and devices. The task of predicting ad response is thus crucial for maximizing business revenue. However, ad response data change dynamically over time, and are subject to cold-start situations in which limited history hinders reliable prediction. There is also a need for a robust regression estimation for high prediction accuracy, and good ranking to distinguish the impacts of different ads. To this end, we develop a *Hierarchical Importance-aware Factorization Machine* (HIFM), which provides an effective generic latent factor framework that incorporates importance weights and hierarchical learning. Comprehensive empirical studies on a real-world mobile advertising dataset show that HIFM outperforms the contemporary temporal latent factor models. The results also demonstrate the efficacy of the HIFM's importance-aware and hierarchical learning in improving the overall prediction and prediction in cold-start scenarios, respectively.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Algorithms, Experimentation

Keywords

Factorization machine, hierarchy, importance weight, mobile advertising, response prediction

WSDM'14, February 24–28, 2014, New York, New York, USA.
Copyright 2014 ACM 978-1-4503-2351-2/14/02 ...$15.00.
http://dx.doi.org/10.1145/2556195.2556240.

1. INTRODUCTION

The proliferation of mobile phones and devices has made mobile advertising an effective means for businesses to target the desired market on the fly. Global mobile advertising revenue is projected to hit $11.4 billion by 2013, creating new opportunities in many countries [7]. A mobile advertising ecosystem is typically governed by an *ad network*, which mediates all transactions between advertisers and content publishers. The *advertiser* plans a budget, provides ads, and agrees on commission for customer actions (e.g., clicking an ad, or bidding in an auction). The *publisher* contracts with the ad network to display ads on their (web)pages, and earns commission based on the traffic driven to the advertisers.

The most popular monetary paradigm in mobile advertising is the *cost-per-click* (CPC) scheme. An advertiser places a bid for an ad to be placed on some publisher's webpage, and pays the bid amount to the publisher only if the ad is displayed (i.e., an *exposure*) and chosen by a user (i.e., a *click*). Among the competing ads, the publisher chooses to display the ad with the highest *expected revenue*, which is the ad's bid price times the probability it is clicked. This probability is known as *clickthrough rate* (CTR), and accurately estimating it is important for maximizing revenue. The task of estimating CTR (or similar metrics such as conversion rate) falls under the term *response prediction* [11, 15].

Response prediction tasks present several challenges. First, ad response data are dynamic and change over time. There is a need for a method that can account for the *temporal aspects* of the data. Second, most ads have limited or no past history in a page–also known as the *cold-start* issue. That is, the response estimation is highly unreliable if we have very low ad exposures. Robust estimation in cold-start cases is important for the advertisers, as it provides a means to measure/explore the potential of investing in new pages and ads. To increase robustness, we can plausibly leverage on prior knowledge in the form of *hierarchical structure* of pages and ads. Such hierarchy contains useful information about correlations between responses at various levels, which can smoothen the estimation for limited historical data.

From the modeling standpoint, response prediction tasks involve two requirements: good *regression* and good *ranking* performances [16]. In real-time auctions for mobile advertising, ads are often ranked based on $bid \times CTR$. It is thus important that CTR gives good ranking that distinguishes the impacts of different ads. Also, ads are usually priced via a next-price auction, i.e., the price of a click on an ad at rank i is based on the ad's expected $bid \times CTR$ at the next lower rank [3]. In this, the CTR (regression) estimates must be as accurate as possible, especially for ads that are displayed many times. That is, high *importance weights* should be assigned to ads with high exposures, as the cost of inaccurately predicting the response of such ads is high.

To address all the above challenges and requirements, we present in this paper a novel latent factor model for response prediction, termed *Hierarchical Importance-aware Factorization Machine* (HIFM). Our HIFM is built upon the state-of-the-art factorization machines (FM) [12, 13], with new extensions to utilize importance weights and hierarchical knowledge for handling temporal/dynamic ad response data. We summarize our key contributions as follows:

- We develop a new latent factor framework that utilizes *importance-aware* and *hierarchical learning* for handling *temporal* ad response data. These two ideas help improve response prediction by putting priorities based on ads' exposures and by making more informative prediction in cold-start cases, respectively. Deviating from the existing approaches, which are often very specialized (e.g., only works for dyads) and not able to cater for temporal data, HIFM offers a generic approach to simultaneously address importance weight, hierarchical knowledge, and temporal dynamics.

- We adapt and generalize two learning algorithms, the *stochastic gradient descent* (SGD) and *coordinate descent* (CD), to facilitate importance-aware and hierarchical learning in HIFM. We adopt roulette wheel sampling strategy and weighted least square update to cater for the importance weight in SGD and CD respectively, while hierarchical learning is achieved in both SGD and CD by means of hierarchical fitting and regularization. For each algorithm, we develop several variants with (the same) linear runtime complexity. These extensions in turn facilitate comprehensive studies for evaluating different methods.

- Extensive experiments on real-world mobile advertising data show that HIFM outperforms contemporary temporal latent factor models. We also find statistically significant improvements due to the HIFM's importance-aware and hierarchical learning, in terms of the overall (weighted) prediction and prediction for cold-start cases, respectively. This shows the applicability of HIFM in complex response prediction tasks.

2. RELATED WORK

Traditionally, the data mining approaches to ad response prediction broadly fall into two camps: *feature-based* and *maximum likelihood-based* [11]. In feature-based methods, prediction models are constructed based on explicit features of a page and/or an ad. These features–also termed as *side information*–may include textual content of an ad, its placement on the (web)page, etc. Feature-based methods typically utilize prediction models from logistic regression family [8, 15]. However, constructing these models requires extensive manual intervention or domain knowledge. On the other hand, maximum likelihood-based methods try to smoothen the response estimation using statistical models of ad clicks and exposures, e.g., the Gamma-Poisson model [1, 2]. These methods, however, are based on simple linear models and lack the ability to capture rich latent structure in data.

Table 1: Global count statistics

Entity	Minimum exposure (e_{min})			
	1	10	100	1000
#records	24,172,134	10,535,658	3,587,160	931,032
#(web)pages	244,341	138,351	55,260	16,374
#publishers	3,945	3,539	2,654	1,643
#countries	243	239	226	199
#channels	8	8	8	8
#ads	23,500	18,365	15,600	10,877
#advertisers	1,989	1,406	1,245	1,124
#banner types	5	4	3	3

In this light, Menon *et al.* [11] proposed a hybrid approach by combining matrix factorization (MF) with explicit page and ad features as well as hierarchical information about pages and ads. This method learns a set of latent features from data through MF, and at the same time allows side information–typical of traditional feature-based methods–to be incorporated for improving the prediction. However, the MF representation used in this method is only restricted to dyadic relations (e.g., page vs. ad), and does not cater for higher-order relations such as those that include temporal dynamics (e.g., page vs. ad vs. day). Similar hierarchical MF methods were proposed in [23, 18], but like [11] all of them assume a restrictive dyadic representation.

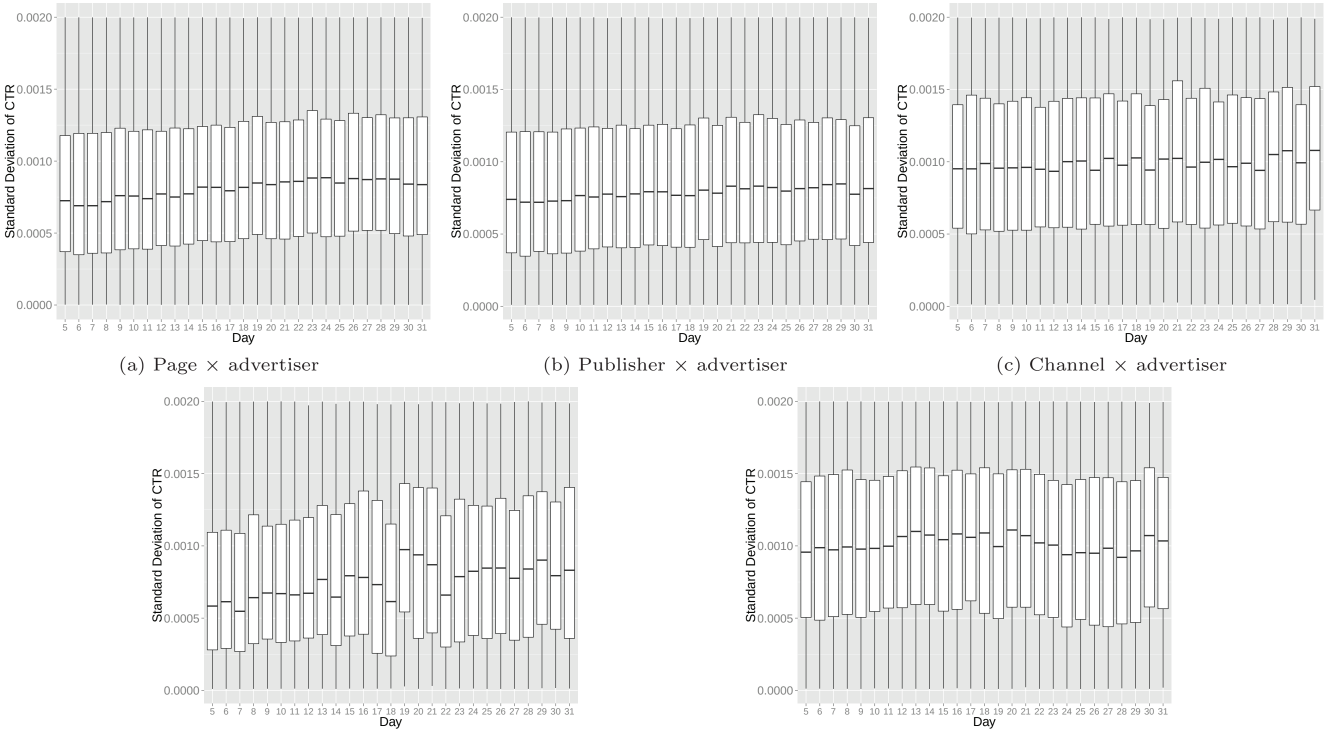
On the other hand, several time-aware factorization approaches have been proposed, which take into account the temporal dynamics in data. Comon *et al.* presented a tensor factorization method trained using an alternative least square algorithm [5]. Koren developed TimeSVD++ to address temporal dynamics via specific parameterization with factors drifting from a central time [9]. More recently, Shen *et al.* [19] proposed a tensor factorization model to address the issue of personalization in click modeling. Other models have been developed that combine MF and Kalman filter to simultaneously model the spatial and temporal patterns in data [10, 21]. Regardless, all these methods use restrictive representation and cannot be easily extended to augment side information such as hierarchical prior knowledge.

Recently, Rendle [12, 13] proposed a generic factorization machine (FM) framework, which can efficiently handle arbitrary relationships (i.e., dyad, triad, etc.). The approach can thus be used to address temporal data and can easily incorporate side information. As noted in [13], the FM approach can mimic many state-of-the art factorization models (e.g., pairwise tensor factorization, TimeSVD++, attribute-aware models, and neighborhood models), and exhibits good performances on many competitive tasks [13]. Nonetheless, the current FM does not yet cater for hierarchical structure and/or instance importance in its model. In light of response prediction task, we extend and generalize the FM framework in this work by combining importance-aware learning and hierarchical fitting and regularization.

3. PROBLEM FORMULATION

3.1 Dataset

In our study, we use the data provided by our industry partner, a global mobile advertising network based in Sin-



(a) Page × advertiser

(b) Publisher × advertiser

(c) Channel × advertiser

(d) Publisher × ad

(e) Channel × ad

Figure 1: Temporal distribution of click-through rates grouped at various hierarchy levels

gapore. The supplied data consist of a *campaign* table and an *ad click* table. The latter contains records of clicks and exposures for various (web)pages and ads at different days. The data were taken from a 27-day period, 05-31 October 2012, and have 24,172,134 records. In this period, there are over 17 billion ad exposures and over 42 million ad clicks.

The ad click table has several fields: `day` (yyyy-mm-dd format), `page` (page ID), `publisher` (publisher ID), `ad` (ad campaign ID), `channel` (publisher's channel type), and `numclick` and `numexpose` (number of clicks and exposures for a page-ad pair at a given day, respectively). Each record in the table is uniquely identified by a $(\texttt{page}, \texttt{ad}, \texttt{day})$ triplet. There are five (i.e., `channel`) types, each denoting the publisher category (e.g., adult sites, entertainment and lifestyle, glamour and dating, etc.). From `numclick` and `numexpose`, we can then estimate $CTR = \frac{\texttt{numclick}}{\texttt{numexpose}}$, though precaution needs to be exercised when `numexpose` is low. Also note that all records in the ad click table have $\texttt{numexpose} \geq 1$.

The campaign table, on the other hand, complements the ad click table and comprises several fields: `ad` (as with the ad click table), `advertiser` (advertising company of the ad), and `bannertype` (type of ad banners). That is, the table maintains a list of ads with their corresponding advertisers and banner types; the latter involve five categories: $\{image, remote, splash, tablet, text\}$. It is also worth noting that all fields (except for *day*) in the ad click and campaign tables are anonymized for security protection.

Table 1 summarizes the count statistics derived from the ad click and campaign tables. It shows the counts of various entities as the minimum exposure threshold e_{min} is varied from 1 to 1000. Here the number of records refers to that of $(\texttt{page}, \texttt{ad}, \texttt{day})$ triplets in the ad click table. As e_{min} gets increased, we can see rather drastic reductions in the counts (except for `channel` and `bannertype`), suggesting a heavy-tail distribution where most pages have small number of ad exposures and thus low-confidence CTR estimates.

3.2 Response Prediction

We now formulate our response prediction task: *Given a pair of page p and ad a at a specific day d, we predict the click probability CTR.* That is, we wish to compute the probability $P_{p,a,d} = Pr(Click|Exposure, p, a, d)$, where p, a and d denote a particular page, ad, and day respectively.

A simple solution to this task is to use the maximum likelihood estimate (MLE) [11], i.e., $P_{p,a,d}^{MLE} = \frac{c_{p,a,d}}{e_{p,a,d}}, \forall e_{p,a,d} > 0$, where $c_{p,a,d}$ and $e_{p,a,d}$ are the numbers of ad clicks and exposures respectively. However, when $e_{p,a,d}$ is very small, the MLE estimate would be highly noisy. For instance, if an ad has been exposed 3 times on a page and received no click, a CTR estimate of 0 would make no sense. As this extreme case is common in practice, we need an alternative estimate $\hat{P}$ that smoothens the MLE so as to make it more reliable.

As advocated in [11], a good predictive model for estimating $\hat{P}$ should fulfill several criteria. First, it should provide smoothened estimates in the face of very few exposures $e_{p,a,d}$. Second, it should output meaningful probabilities, within the range $[0, 1]$. Finally, the MLE should be consistent, meaning that as $e_{p,a,d} \to \infty$, $\hat{P}_{p,a,d} \to P_{p,a,d}^{MLE}$. In other words, when the number of exposures is large enough, any estimator $\hat{P}$ should converge to the MLE.

3.3 Page and Ad Hierarchies

Further investigations on the ad click and campaign tables reveal that page and ad hierarchies exist in the data. That is, a page is associated with a publisher, and a publisher belongs to a unique channel. Similarly, an ad belongs to an advertiser. Hierarchical structure encodes useful prior knowledge for improving CTR estimation when we have little data at granular level. For instance, if a (page, ad) pair has only very small numexpose, but the siblings of the ad has large numexpose, we can increase the confidence of its estimate by leveraging the siblings' estimates. This allows a "back-off" mechanism for borrowing data from coarse level to smoothen the estimation at the granular level [11].

For our hierarchy analysis, we compute the CTR distribution for different pairs (Cartesian products) of entities in the page and ad hierarchies: page × advertiser, publisher × ad, channel × ad, publisher × advertiser, and channel × advertiser. Figure 1 shows the box plots of the CTR variations (i.e., first quantile, median, and third quantile) for each entity pair over 27 days. To obtain Figure 1(a), for instance, we first group each ad by the advertiser, and then look at the CTR variation under the page-advertiser pair for a given day. Specifically, we calculate the standard deviation of the CTR for each page-advertiser pair, which measures the homogeneity of the CTR under the group. We repeat the procedure for the remaining (four) combinations, the results of which are shown in Figures 1(b)-(d).

From these plots, we can make several observations. First, the CTR values vary across different days, which justifies the need to model temporal dynamics. Second, comparing Figure 1(a)-(c) and 1(d)-(e), we can see that the CTR variability increases as we go up in the hierarchy, implying that the CTR values are less (more) homogeneous at the higher (lower) level. This suggests the feasibility of the "back-off" mechanism, i.e., when a child node lacks data, we can leverage on the estimates from its parent(s), and then those from its grandparent(s), and so on.

4. FACTORIZATION MACHINE

The HIFM model we proposed to predict CTR is built upon a generic latent factor model called *factorization machine* (FM) [12, 13]. We first describe the basic FM in this section, and then new extensions that we develop in-house for HIFM in Section 5, based on different importance-aware learning methods and use of page and ad hierarchies.

4.1 Model Overview

The FM takes as its inputs a real-valued feature matrix (also called *design matrix*) $X \in \mathbb{R}^{N \times J}$. For an input vector $\mathbf{x}_i$ ($i \in \{1, \ldots, N\}$, $\mathbf{x}_i \in \mathbb{R}^J$), the output $\hat{y}_i$ of an (order-2) FM is defined in (1):

$$\hat{y}(\mathbf{x}_i) = w_0 + \sum_{j=1}^{J} w_j x_{i,j} + \underbrace{\sum_{j=1}^{J} \sum_{j'=j+1}^{J} x_{i,j} x_{i,j'} \sum_{k=1}^{K} v_{k,j} v_{k,j'}}_{=\hat{p}(\mathbf{x}_i)} \tag{1}$$

where K is the dimensionality of the interaction factors, and the model parameters $\Theta = \{w_0, w_1, \ldots, w_J, v_{1,1}, \ldots, v_{K,J}\}$ consist of $w_o \in \mathbb{R}$, $\mathbf{w} \in \mathbb{R}^J$, and $\mathbf{V} \in \mathbb{R}^{K \times J}$.

According to [12], we can efficiently compute the term $\hat{p}(\mathbf{x}_i)$ in (1) based on its equivalent formulation (2):

$$\hat{p}(\mathbf{x}_i) = \frac{1}{2} \sum_{k=1}^{K} \left[\left(\sum_{j=1}^{J} v_{k,j} x_{i,j} \right)^2 - \sum_{j=1}^{J} v_{k,j}^2 x_{i,j}^2 \right] \tag{2}$$

This makes it possible to quickly process all N instances of a design matrix X in linear time $O(KN_z(X))$, where $N_z(X)$ is the number of non-zero entries in X.

4.2 Multilinearity Property

The FM has an appealing *multilinearity* trait: for each model parameter $\theta \in \Theta$, the FM output $\hat{y}(\mathbf{x}_i)$ is a linear combination of two functions $r_\theta(\mathbf{x})$ and $g_\theta(\mathbf{x})$, as per (3):

$$\hat{y}(\mathbf{x}_i) = r_\theta(\mathbf{x}_i) + \theta g_\theta(\mathbf{x}_i), \quad \forall \theta \in \Theta \tag{3}$$

where $g_\theta(\mathbf{x}_i)$ corresponds to the gradient term (4):

$$g_\theta(\mathbf{x}_i) = \frac{\partial \hat{y}(\mathbf{x}_i)}{\partial \theta} = \begin{cases} 1 & \text{if } \theta = w_0 \\ x_{i,j} & \text{if } \theta = w_j \\ x_{i,j}\,(q_{i,k} - v_{k,j} x_{i,j}) & \text{if } \theta = v_{k,j} \end{cases} \tag{4}$$

and $q_{i,k} = \sum_{j=1}^{J} v_{k,j} x_{i,j}$ is the inner-product term that can be precomputed (cached) for efficient learning (cf. Section 5.3). Note that in our learning procedures, the residual term $r_\theta(\mathbf{x}_i)$ need not be computed; we only use the term $g_\theta(\mathbf{x}_i)$.

5. THE PROPOSED HIFM FRAMEWORK

We extend the basic FM model to incorporate importance weights and hierarchical structure for response prediction tasks, which results in the proposed HIFM model. We first discuss how importance weights and hierarchy can be incorporated into our model in Sections 5.1 and 5.2, respectively. Section 5.3 then presents our several variants of optimization algorithm for learning the HIFM's model parameters.

5.1 Importance-Aware Loss Functions

We consider the problem of predicting the click probability $\hat{y}_i(\mathbf{x}|\Theta) \in [0, 1]$ (i.e., CTR), given a design matrix X and model parameters Θ. The goodness-of-fit between the predicted and actual probabilities can be expressed using the two objective functions in (5) and (6):

$$L_{square}(\Theta) = \sum_{i=1}^{N} e_i \left(\frac{c_i}{e_i} - \hat{y}(\mathbf{x}_i|\Theta)\right)^2 + \Omega(\Theta) \tag{5}$$

$$\begin{aligned} L_{logistic}(\Theta) = &- \sum_{i=1}^{N} (c_i \log \sigma(\hat{y}(\mathbf{x}_i|\Theta)) + \\ &(e_i - c_i) \log(1 - \sigma((\hat{y}(\mathbf{x}_i|\Theta))) + \Omega(\Theta)) \end{aligned} \tag{6}$$

where $c_i \geq 0$ and $e_i \geq 1$ denote the numbers of observed ad clicks and exposures (impressions) for the i^{th} instance $\mathbf{x}_i \in X$, respectively, $\hat{y}_i(\mathbf{x}|\Theta) \in [0, 1]$ is the click probability to be predicted by FM, $\Omega(\Theta)$ is the regularization term, and $\sigma(x) = \frac{1}{1+e^{-x}}$ is the logistic function. These formulae are also known as the *weighted square* and *weighted logistic* loss functions, using e_i as the importance weights.

The regularization term $\Omega(\Theta)$ is useful to reduce the risk of data overfitting, due to a large number of model parameters Θ (especially when K is large). Typically, L2 regularization is used [20] as given in (7):

$$\Omega(\Theta) = \frac{1}{2} \sum_{\theta \in \Theta} \lambda_\theta \theta^2 \tag{7}$$

which is equivalent to imposing a Gaussian prior on θ, i.e., $\theta \sim \mathcal{N}(0, \lambda_\theta^{-1})$. This regularization essentially tries to penalize model parameters that have large (squared) norm, thus driving the parameter values to be small.

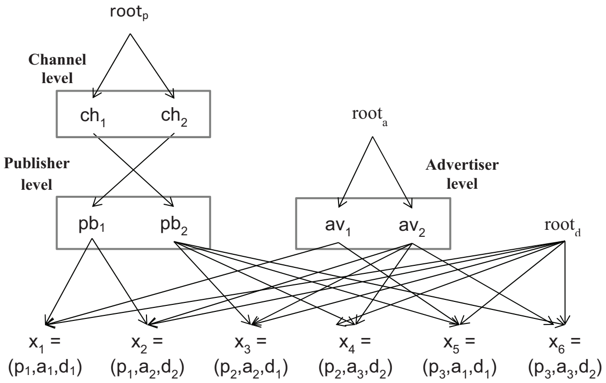

root$_p$
Channel level
ch$_1$
ch$_2$
root$_a$
Publisher level
Advertiser level
pb$_1$
pb$_2$
av$_1$
av$_2$
root$_d$
x_1 = (p_1,a_1,d_1)
x_2 = (p_1,a_2,d_2)
x_3 = (p_2,a_2,d_1)
x_4 = (p_2,a_3,d_2)
x_5 = (p_3,a_1,d_1)
x_6 = (p_3,a_3,d_2)

Figure 2: Example of page and ad hierarchies

5.2 Hierarchical Learning

Inspired by [11, 18], we utilize hierarchical information in our FM model via two ways: *hierarchical fitting* and *regularization*. For illustration, we consider the page and ad hierarchies in Figure 2. There are six instances at the bottom level, each corresponding to a page-ad-day (p, a, d) triplet. For the page hierarchy, each page belongs to a publisher node (pb) and then a publisher belongs to a unique channel (ch). Meanwhile, for the ad hierarchy, each ad belongs to an advertiser node (av). We implement the hierarchies as a *directed graph* (DG). It should be noted that the hierarchy structure need not be a tree; the DG representation allows us to work with non-tree hierarchy as well.

To incorporate hierarchical constraints into FM, we let every node in the hierarchy to possess its *own* latent factors (i.e., $\{w_j\}$ and $\{v_{k,j}\}$). Using this setup, we impose *hierarchical regularization* on the latent factors, so that each factor has a prior that makes it more similar to its parent in expectation. This is expressed in (8):

$$\theta_0 \sim \mathcal{N}(0, \lambda_{\theta_0}^{-1}), \forall \theta_0 \in \mathcal{H}^{root} \tag{8}$$

$$\theta_j \sim \mathcal{N}(\bar{\theta}_{Par}, \lambda_{\theta_j}^{-1}), \forall \theta_j \in \mathcal{H} - \mathcal{H}^{root} \tag{9}$$

where $\bar{\theta}_{Par}$ is the average of the parents' latent factors as defined in (10):

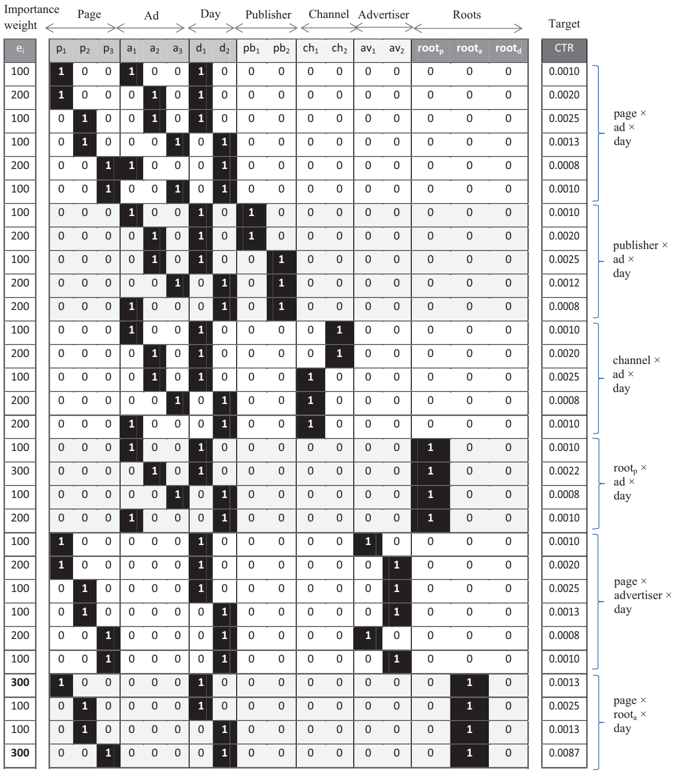
$$\bar{\theta}_{Par} = \frac{1}{|Par(\theta)|} \sum_{\theta_p \in Par(\theta)} \theta_p \tag{10}$$

where $Par(\theta)$ is the set of all parents of θ.

The resulting hierarchical regularization is given in (11):

$$\Omega(\Theta) = \begin{cases} \frac{1}{2}\sum_\theta \lambda_\theta \left(\theta - \bar{\theta}_{Par}\right)^2 & \text{if } \theta \in \mathcal{H} - \mathcal{H}_{root} \\ \frac{1}{2}\sum_\theta \lambda_\theta \theta^2 & \text{if } \theta \in \mathcal{H}_{root} \end{cases} \tag{11}$$

where $\mathcal{H}$ is the set of all nodes in the hierarchy, and $\mathcal{H}_{root}$ is the set of root nodes sitting at the top of the hierarchy (i.e., with no incoming edges). In other words, *every* node is expected to be similar to its parents (if any). Note that our regularization approach is more general than that in [11, 18], which is only restricted to dyads (pages and ads in this case) and does not model temporal dynamics. Also, in our approach, the interactions between pages, ads (and days) are governed by shared latent factors $V_{k,j}$, instead of separate latent factors for pages and ads as in [11, 18]. Together with hierarchical regularization, this enables our method to handle sparse data more effectively.



Importance weight	Page			Ad			Day		Publisher		Channel		Advertiser		Roots			Target	
e_i	p_1	p_2	p_3	a_1	a_2	a_3	d_1	d_2	pb_1	pb_2	ch_1	ch_2	av_1	av_2	root$_p$	root$_a$	root$_d$	CTR	
100	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0.0010	page × ad × day
200	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0.0020	
100	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0.0025	
100	0	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0.0013	
200	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0.0008	
100	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0.0010	
100	0	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0.0010	publisher × ad × day
200	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0.0020	
100	0	0	0	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0.0025	
200	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0.0012	
200	0	0	0	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0.0008	
100	0	0	0	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0.0010	channel × ad × day
200	0	0	0	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0.0020	
100	0	0	0	0	1	0	1	0	0	0	1	0	0	0	0	0	0	0.0025	
200	0	0	0	0	0	1	0	1	0	0	1	0	0	0	0	0	0	0.0008	
200	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0.0010	
100	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0.0010	root$_p$ × ad × day
300	0	0	0	0	1	0	1	0	0	0	0	0	0	0	1	0	0	0.0022	
100	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	0.0008	
200	0	0	0	1	0	0	0	1	0	0	0	0	0	0	1	0	0	0.0010	
100	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0.0010	page × advertiser × day
200	1	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0.0020	
100	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0.0025	
100	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0.0013	
200	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0.0008	
100	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0.0010	
300	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0.0013	page × root$_a$ × day
100	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0.0025	
100	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0.0013	
300	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0.0087	

-
- The remaining combinations ...
-

Figure 3: Example of feature representation for FM

While hierarchical regularization alone can help constrain the parameters of the nodes, the final derived priors may not conform directly to the underlying click and exposure data. More informative priors can be obtained by fitting the data directly, that is, by *agglomerating* the click/expose data over various pages and ads. As an example, for a publisher node pb, we agglomerate the clicks/exposures of all page-ad-day triplets (p, a, d) where p belongs to publisher. In this way, a reasonable prior for the children's latent factors can be learned. Formally, the aggregated clicks and exposures of a *non-leaf* node u in the hierarchy are defined in (12):

$$c_u = \sum_{v:u \in Par(v)} c_v, e_u = \sum_{v:u \in Par(v)} e_v \tag{12}$$

The base case is when v is the leaf node, such that c_u (or e_u) is the standard click (or expose) value c_i (or v_i).

Figure 3 illustrates the FM-compatible feature representation for the example in Figure 2. We use binary categorical features to represent various entities in the hierarchy. Each row represents a data instance, and is associated with a target response (CTR) and importance weight (exposures) e_i. The base instances (i.e., `page` × `ad` × `day`) correspond to the first six rows. In the next level `publisher` × `ad` × `day`, we aggregate the exposures e_i (and similarly the clicks c_i), yielding the next five rows. We repeat this for the remaining combinations to obtain the full design matrix X for FM.

One may notice from Figure 3 that, due to the agglomeration process, the exposures and clicks of the base cases `page`×`ad`×`day` will be outweighed by the ones in the parent level (e.g., $\texttt{publisher} \times \texttt{ad} \times \texttt{day}$). As our model is weighted by the exposures, the contribution of the base cases would be "ignored" in our HIFM training. To remedy this, we rescale the importance weights of all *non-base* cases using (13):

$$e_i \leftarrow \frac{e_i}{2} \frac{\sum_{p \in \texttt{page} \times \texttt{ad} \times \texttt{day}} e_p}{\sum_{q \notin \texttt{page} \times \texttt{ad} \times \texttt{day}} e_q}, \forall i \notin \texttt{page} \times \texttt{ad} \times \texttt{day} \tag{13}$$

That is, we rescale based on the ratio of the sum of exposures of the base cases over that of the non-base cases.

5.3 Optimization Algorithms

To support hierarchical and importance-aware learning in HIFM, we develop two classes of efficient iterative methods: *stochastic gradient descent* (SGD) and *coordinate descent* (CD), as described in Sections 5.3.1 and 5.3.2 respectively.

5.3.1 Stochastic Gradient Descent

The standard, unweighted SGD update for a specific model parameter θ is given in (14):

$$\theta \leftarrow \theta - \eta \left[\frac{\partial l(y_i, \hat{y}(\mathbf{x}_i|\Theta))}{\partial \theta} + \frac{\partial \Omega(\Theta)}{\partial \theta} \right] \tag{14}$$

where y_i is the target value of the i^{th} instance, and $l(a,b) = (b-a)^2$ for square loss or $l(a,b) = -a \log(\sigma(b)) - (1-a)\log(1-\sigma(b))$ for logistic loss. The (sub)gradients of the square and logistic losses translate to (15)-(16) respectively:

$$\frac{\partial l_{square}(y_i, \hat{y}(\mathbf{x}_i|\Theta))}{\partial \theta} = (\hat{y}(\mathbf{x}_i|\Theta) - y_i)\, g_\theta(\mathbf{x}_i) \tag{15}$$

$$\frac{\partial l_{logistic}(y_i, \hat{y}(\mathbf{x}_i|\Theta))}{\partial \theta} = (\sigma(\hat{y}(\mathbf{x}_i|\Theta)) - y_i)\, g_\theta(\mathbf{x}_i) \tag{16}$$

A simple way to incorporate importance weights is to oversample the instances as many as the number of exposures e_i. However, for large e_i, the oversampled data would be much larger than the original data, leading to inefficient computation. Another tempting approach is to multiply the gradient term $g_\theta(\mathbf{x}_i)$ by e_i. However, such approach violates the principle that an instance $\mathbf{x}_i$ with importance weight e_i should be treated as if it appears e_i times in the data.

Algorithm 1 Importance-aware SGD with regularization

```
Input: Design matrix $X$, targets $\mathbf{y} = \{y_i\}$, regularization
    parameters $\{\lambda_\theta\}$, initial width $\sigma$, learning rate $\eta$
Output: Model parameters $\Theta$
$w_0 \leftarrow 0$; $\mathbf{w} \leftarrow (0, \ldots, 0)$; $V \sim N(0, \sigma^2)$
repeat
    for $i \in \{1, \ldots, N\}$ do
        $i \leftarrow \textit{RouletteWheelSelection}()$
        $w_0 \leftarrow w_0 - \eta \left( \frac{\partial l(\hat{y}(\mathbf{x}_i|\Theta), y_i)}{\partial w_0} + \frac{\partial \Omega(\Theta)}{\partial w_0} \right)$
        for $j \in \{1, \ldots, J\} \wedge x_j \neq 0$ do
            $w_j \leftarrow w_j - \eta \left( \frac{\partial l(\hat{y}(\mathbf{x}_i|\Theta), y_i)}{\partial w_j} + \frac{\partial \Omega(\Theta)}{\partial w_j} \right)$
            for $k \in \{1, \ldots, K\}$ do
                $v_{k,j} \leftarrow v_{k,j} - \eta \left( \frac{\partial l(\hat{y}(\mathbf{x}_i|\Theta), y_i)}{\partial v_{k,j}} + \frac{\partial \Omega(\Theta)}{\partial v_{k,j}} \right)$
            end for
        end for
    end for
until stopping criterion is met
```

In light of these drawbacks, we propose to use a simple roulette wheel selection mechanism [4]. That is, the probability of selecting instance $\mathbf{x}_i$ with importance weight e_i is given by $p(\mathbf{x}_i) = \frac{e_i}{\sum_{i'=1}^{N} e'_i}$, and subsequently a random sampling (with replacement) is carried out according to $p(\mathbf{x}_i)$ until N instances are obtained[1]. In this way, we can avoid explicitly oversampling the data, while achieving asymptotically the same effect of presenting an instance e_i times, given sufficient number of iterations. Note that, for each instance, the runtime complexity of our importance-aware SGD remains the same as the original SGD, i.e., $O(KN_z(X))$.

Our importance-aware SGD method (with generic regularization term) is summarized in Algorithm 1. With respect to the different regularizations (7) and (11), the term $\frac{\partial \Omega(\Theta)}{\partial \theta}$ evaluates to (17) and (18) respectively:

$$\frac{\partial \Omega(\Theta)}{\partial \theta} = \lambda_\theta \theta \tag{17}$$

$$\frac{\partial \Omega(\Theta)}{\partial \theta} = \lambda_\theta \left(\theta - \bar{\theta}_{Par} \right) \tag{18}$$

where we have a special case $\bar{\theta}_{Par} = 0$ if $\theta \in \mathcal{H}_{root}$.

Finally, in cold-start situations, we perform a special procedure when dealing with a new column j in unseen (test) data. Specifically, whenever we have hierarchy information, we augment our prediction using the latent factors of the ancestors. This is realized via the "back-off" formula (19):

$$w_j \leftarrow \sum_{j' \in Path(j) - \{j\}} w_{j'}, \; v_{k,j} \leftarrow \sum_{j' \in Path(j) - \{j\}} v_{k,j'} \tag{19}$$

where $Path(j)$ refers to the path from the root node to node j in the hierarchy. For a non-tree hierarchy, we utilize the average latent factor of the parent nodes level by level.

5.3.2 Coordinate Descent

The basic idea of the CD algorithm is to start with an initial (random) guess of Θ, and then iterate over and update each model parameter $\theta \in \Theta$, assuming all the other parameters are fixed. We propose an extended importance-aware CD with generic regularization. For the weighted square loss in (5), the importance-aware CD algorithm performs an update of the form (20):

$$\theta = \arg\min_\theta \left(\sum_{i=1}^{N} e_i (\hat{y}(\mathbf{x}_i|\Theta) - y_i)^2 + \Omega(\Theta) \right) \tag{20}$$

Note that, for logistic loss, there is no closed-form solution for the update, so we focus only on square loss in this paper.

Consequently, the update steps for the regularization in (17) and (18) are given by (21) and (22) respectively:

$$\theta \leftarrow \frac{\theta \sum_{i=1}^{N} e_i g_\theta^2(\mathbf{x}_i) + \sum_{i=1}^{N} e_i g_\theta(\mathbf{x}_i) r_i}{\sum_{i=1}^{N} e_i g_\theta^2(\mathbf{x}_i) + \lambda_\theta} \tag{21}$$

$$\theta \leftarrow \frac{\theta \sum_{i=1}^{N} e_i g_\theta^2(\mathbf{x}_i) + \sum_{i=1}^{N} e_i g_\theta(\mathbf{x}_i) r_i + \lambda_\theta \bar{\theta}_{Par}}{\sum_{i=1}^{N} e_i g_\theta^2(\mathbf{x}_i) + \lambda_\theta} \tag{22}$$

where $r_i = y_i - \hat{y}(\mathbf{x}_i|\Theta)$ is the residual term and, again, $\bar{\theta}_{Par} = 0$ if $\theta \in \mathcal{H}_{root}$. The update for standard (L2) regularization is thus a special case of (22), where $Par(\theta) = \emptyset$. In the case of a cold-start column such that all $g_\theta(\mathbf{x}_i) = 0$, hierarchical regularization gives us $\theta \leftarrow \bar{\theta}_{Par}$, which should lead to a more informative prediction.

Algorithm 2 outlines our importance-aware CD method with generic regularization. Here, linear-time learning can

[1] The standard, unweighted SGD refers to a special case of roulette wheel selection with uniform probability $p(\mathbf{x}_i) = \frac{1}{N}$

Algorithm 2 Importance-aware CD with regularization

```
Input: Design matrix $X$, targets $\mathbf{y} = \{y_i\}$, regularization parameters $\{\lambda_\theta\}$, initial width $\sigma$
Output: Model parameters $\Theta$
$w_0 \leftarrow 0$; $\mathbf{w} \leftarrow (0, \ldots, 0)$; $V \sim N(0, \sigma^2)$
repeat
    $\hat{\mathbf{y}} \leftarrow$ predict all instances $X$
    $\mathbf{r} \leftarrow \mathbf{y} - \hat{\mathbf{y}}$
    Update $w_0$ using (21) or (22)
    Update cache $\mathbf{r}$ using (25)
    for $j \in \{1, \ldots, J\}$ do
        Update $w_j$ using (21) or (22)
        Update cache $\mathbf{r}$ using (25)
    end for
    for $k \in \{1, \ldots, K\}$ do
        Initialize cache $\mathbf{q}_{\cdot,k}$ using (24)
        for $j \in \{1, \ldots, J\}$ do
            Update $v_{k,j}$ using (21) or (22)
            Update cache $\mathbf{r}$ using (25)
            Update cache $\mathbf{q}$ using (26)
        end for
    end for
until stopping criterion is met
```

be achieved using *r-cache* and *q-cache*, which store the results of (pre)computing the residuals $\{r_i\}$ and inner-product terms $\{q_{i,k}\}$. They are given in (23) and (24) respectively:

$$r_i = y_i - \hat{y}(\mathbf{x}_i|\Theta) \tag{23}$$

$$q_{i,k} = \sum_{j=1}^{J} v_{k,j} x_{i,j} \tag{24}$$

After each update of model parameter θ, we update the e-cache accordingly. Similarly, for every update of $V_{k,j}$, we update the q-cache. The cache update steps are summarized in (25) and (26) respectively:

$$r_i \leftarrow r_i - \left(\theta^{new} - \theta^{old}\right) g_\theta(\mathbf{x}_i) \tag{25}$$

$$q_{i,k} \leftarrow q_{i,k} + \left(v_{k,j}^{new} - v_{k,j}^{old}\right) x_{i,j} \tag{26}$$

Referring to (22), the bulk of the CD computations lies on two terms: $\sum_{i=1}^{N} e_i g_\theta^2(\mathbf{x}_i)$ and $\sum_{i=1}^{N} e_i g_\theta(\mathbf{x}_i) r_i$. By caching residuals $\{r_i\}$ and inner-products $\{q_{i,k}\}$, each full iteration over all $\theta \in \Theta$ can be done quickly in $O(KN_z(X))$.

We also note that Algorithm 2 actually corresponds to the *cyclic coordinate descent*, which traverses all columns j in a fixed order (i.e., from $j = 1$ to $j = J$). In a similar vein to [17], we also develop a stochastic variant called *stochastic coordinate descent*, which iterates over the columns in a random order instead. In our experiments, we found that this stochastic version exhibits faster convergence, albeit producing similar overall prediction quality (cf. Section 6.3).

6. EXPERIMENTS

6.1 Setup

We conducted 10 runs of experiments to evaluate our algorithms, based on the ad response dataset supplied by our partner for the period of 05-31 October 2012 (cf. Section 3). For each run, we take 9 days and split the data by time into a training set (7 days) and a test set (2 days). For our evaluations, we use the average and standard deviation of the prediction results on the 10 test sets. Table 2 summarizes the configuration of our training and test sets.

Table 2: Experiment setup

Evaluation	Training period	Test period
Trial 1	05-11 October	12-13 October
Trial 2	07-13 October	14-15 October
Trial 3	09-15 October	16-17 October
Trial 4	11-17 October	18-19 October
Trial 5	13-19 October	20-21 October
Trial 6	15-21 October	22-23 October
Trial 7	17-23 October	24-25 October
Trial 8	19-25 October	26-27 October
Trial 9	21-27 October	28-29 October
Trial 10	23-29 October	30-31 October

For all algorithms, we set the number of interaction factors as $K = 5$, as we find it gives good overall results. We also fix the regularization parameters as $\lambda_\theta = 0$ for $\theta = w_0$, and $\lambda_\theta = 0.001$ for $\theta = w_j$ or $\theta = V_{k,j}$. As our stopping criterion, we fix a total number of iterations for each algorithm. For the SGD algorithm, we set the total iterations to 100, while we use 10 for the CD algorithm (since CD generally requires fewer iterations than SGD to reach convergence).

6.2 Evaluation Metrics

As mentioned in Section 1, a desirable model for response prediction should give good regression and ranking results. Accordingly, we evaluate the prediction performances of our algorithms using several regression and ranking metrics. For regression, we use weighted root mean square error (wRMSE) and weighted negative log likelihood (wNLL), as defined in (27) and (28) respectively, for a test set with N instances:

$$wRMSE = \sqrt{\frac{\sum_{i=1}^{N} e_i \left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{N} e_i}} \tag{27}$$

$$wNLL = -\frac{\sum_{i=1}^{N} e_i (y_i \log \sigma(\hat{y}_i) + (1 - y_i) \log (1 - \sigma(\hat{y}_i))}{\sum_{i=1}^{N} e_i} \tag{28}$$

Here y_i and $\hat{y}_i$ denote the target and predicted outputs respectively, and (again) e_i is the number of exposures. The $wRMSE$ and $wNLL$ can be thought as complementary metrics quantifying the importance-weighted "distance" between predicted and actual probabilities. They also correspond to the loss functions (5) and (6) respectively.

For ranking, on the other hand, we use the weighted area under the receiver-operating characteristic curve (wAUC) [6]. Essentially, this metric reflects the probability that a predictive model will rank a randomly chosen positive instance higher than a randomly chosen negative one. In this, we treat the problem as a binary classification task, where a click (no click) indicates a positive (negative) instance. For example, if the number of exposures $e_i = 10$ and the number of clicks $c_i = 3$ (i.e., $CTR = y_i = 0.3$), we treat this as 3 positive instances and 7 negative instances. By "unrolling" all CTR entries this way, we can compute AUC weighted by ad exposures so as to measure the ranking quality.

6.3 Overall Results

The overall prediction results, evaluated using the three performance metrics, are summarized in Tables 3, 4 and 5,

Table 3: Consolidated results averaged over 10 trials for $e_{min} = 10$

Method	Type	Importance	Hierarchy	wRMSE	wNLL	wAUC
TensorALS	-	-	-	0.0245347 ± 0.0006238	0.1141281 ± 0.0031088	0.5 ± 0.0
TimeSVD++	-	-	-	0.0230917 ± 0.0005482	0.0369275 ± 0.0009549	0.7046705 ± 0.0101725
FM-SGD	Square	No	No	0.0105449 ± 0.0040673	0.0190836 ± 0.0033935	0.7600722 ± 0.0348881
		No	Yes	0.0092290 ± 0.0034284(*)	0.0190393 ± 0.0040543	0.7588117 ± 0.0426068
		Yes	No	0.0043385 ± 0.0003909(*)	0.0139876 ± 0.0021322(*)	0.8263158 ± 0.0327539(*)
		Yes	Yes	0.0044152 ± 0.0006106(*)	0.0143682 ± 0.0024797(*)	0.8240209 ± 0.0343667(*)
FM-SGD	Logistic	No	No	0.0067266 ± 0.0003933	0.0154207 ± 0.0013011	0.7730542 ± 0.0314904
		No	Yes	0.0068068 ± 0.0008265	0.0149345 ± 0.0014383(*)	0.7804171 ± 0.0339882(*)
		Yes	No	0.0052925 ± 0.0001996(*)	0.0139813 ± 0.0014130(*)	0.8136070 ± 0.0334504(*)
		Yes	Yes	0.0048635 ± 0.0001841(*)	0.0136824 ± 0.0014207(*)	0.8152107 ± 0.0340372(*)
FM-CD	Cyclic	No	No	0.0061768 ± 0.0001934	0.0145682 ± 0.0015581	0.7617378 ± 0.0441938
		No	Yes	0.0063543 ± 0.0001996	0.0151101 ± 0.0013793	0.7629488 ± 0.0455320
		Yes	No	0.0044806 ± 0.0002127(*)	0.0135587 ± 0.0017265(*)	0.8262493 ± 0.0283374(*)
		Yes	Yes	0.0047210 ± 0.0001874(*)	0.0141638 ± 0.0021363	0.8212097 ± 0.0321086(*)
FM-CD	Stochastic	No	No	0.0062104 ± 0.0001935	0.0146287 ± 0.0015349	0.7617537 ± 0.0441836
		No	Yes	0.0063543 ± 0.0001996	0.0151101 ± 0.0013793	0.7629527 ± 0.0455303
		Yes	Yes	0.0044815 ± 0.0002135(*)	0.0135992 ± 0.0017595(*)	0.8255169 ± 0.0290100(*)
		Yes	Yes	0.0047006 ± 0.0001915(*)	0.0138494 ± 0.0021000	0.8233171 ± 0.0326712(*)

(*) indicates statistically significant improvements over the baseline based on Wilcoxon signed-rank test at significance level of 0.01

Table 4: Consolidated results averaged over 10 trials for $e_{min} = 100$

Method	Type	Importance	Hierarchy	wRMSE	wNLL	wAUC
TensorALS	-	-	-	0.0148571 ± 0.0007870	0.0898490 ± 0.0033757	0.5 ± 0.0
TimeSVD++	-	-	-	0.0133219 ± 0.0009000	0.0300316 ± 0.0011584	0.7104049 ± 0.0104965
FM-SGD	Square	No	No	0.0066994 ± 0.0039766	0.0159641 ± 0.0030365	0.7906827 ± 0.0399742
		No	Yes	0.0063514 ± 0.0036174	0.0155720 ± 0.0030424	0.7947463 ± 0.0436061
		Yes	No	0.0035637 ± 0.0002817(*)	0.0136790 ± 0.0022342(*)	0.8263953 ± 0.0345580(*)
		Yes	Yes	0.0036013 ± 0.0003965(*)	0.0143439 ± 0.0028064(*)	0.8216659 ± 0.0377304(*)
FM-SGD	Logistic	No	No	0.0052633 ± 0.0006386	0.0144573 ± 0.0013365	0.7831293 ± 0.0378750
		No	Yes	0.0048462 ± 0.0009245	0.0139555 ± 0.0014297(*)	0.7940811 ± 0.0385131(*)
		Yes	No	0.0048071 ± 0.0002476(*)	0.0137598 ± 0.0014451(*)	0.8085024 ± 0.0353346(*)
		Yes	Yes	0.0043588 ± 0.0002261(*)	0.0134459 ± 0.0014542(*)	0.8119756 ± 0.0355739(*)
FM-CD	Cyclic	No	No	0.0052848 ± 0.0002765	0.0139085 ± 0.0017059	0.7725149 ± 0.0481081
		No	Yes	0.0054034 ± 0.0002760	0.0141956 ± 0.0015021	0.7735657 ± 0.0484978
		Yes	No	0.0035659 ± 0.0002768(*)	0.0131528 ± 0.0016376(*)	0.8246543 ± 0.0288612(*)
		Yes	Yes	0.0037860 ± 0.0002213(*)	0.0134757 ± 0.0020551	0.8253329 ± 0.0321349(*)
FM-CD	Stochastic	No	No	0.0053125 ± 0.0002773	0.0139288 ± 0.0016660	0.7725211 ± 0.0481107
		No	Yes	0.0054034 ± 0.0002760	0.0141957 ± 0.0015021	0.7735673 ± 0.0484932
		Yes	Yes	0.0035798 ± 0.0002810(*)	0.0131411 ± 0.0016117(*)	0.8248637 ± 0.0276094(*)
		Yes	Yes	0.0037806 ± 0.0002244(*)	0.0133138 ± 0.0019167	0.8262658 ± 0.0313433(*)

(*) indicates statistically significant improvements over the baseline based on Wilcoxon signed-rank test at significance level of 0.01

Table 5: Consolidated results averaged over 10 trials for $e_{min} = 1000$

Method	Type	Importance	Hierarchy	wRMSE	wNLL	wAUC
TensorALS	-	-	-	0.0105002 ± 0.0011214	0.0698346 ± 0.0037189	0.5 ± 0.0
TimeSVD++	-	-	-	0.0093541 ± 0.0012304	0.0244543 ± 0.0009942	0.7066218 ± 0.0113939
FM-SGD	Square	No	No	0.0035179 ± 0.0006461	0.0140867 ± 0.0031550	0.8032424 ± 0.0479115
		No	Yes	0.0033768 ± 0.0005637	0.0141022 ± 0.0031252	0.8029501 ± 0.0465531
		Yes	No	0.0027669 ± 0.0003007(*)	0.0127455 ± 0.0023743(*)	0.8253467 ± 0.0383887(*)
		Yes	Yes	0.0027779 ± 0.0002821(*)	0.0136180 ± 0.0027904	0.8191953 ± 0.0401579(*)
FM-SGD	Logistic	No	No	0.0041532 ± 0.0003815	0.0133235 ± 0.0014580	0.7802751 ± 0.0425936
		No	Yes	0.0035746 ± 0.0004055(*)	0.0128660 ± 0.0014666(*)	0.7920499 ± 0.0432757(*)
		Yes	No	0.0043950 ± 0.0003841	0.0130108 ± 0.0015346(*)	0.7984587 ± 0.0402334(*)
		Yes	Yes	0.0038005 ± 0.0003306(*)	0.0126841 ± 0.0015381(*)	0.8057917 ± 0.0396685(*)
FM-CD	Cyclic	No	No	0.0043290 ± 0.0004144	0.0131398 ± 0.0022141	0.7795976 ± 0.0543661
		No	Yes	0.0044521 ± 0.0003917	0.0129831 ± 0.0016401	0.7811009 ± 0.0542086(*)
		Yes	No	0.0025551 ± 0.0003352(*)	0.0124212 ± 0.0017798(*)	0.8232776 ± 0.0334076(*)
		Yes	Yes	0.0027313 ± 0.0002812(*)	0.0123107 ± 0.0018302(*)	0.8268533 ± 0.0333732(*)
FM-CD	Stochastic	No	No	0.0043391 ± 0.0004133	0.0129884 ± 0.0020158	0.7797752 ± 0.0543298
		No	Yes	0.0044521 ± 0.0003917	0.0129832 ± 0.0016401	0.7811113 ± 0.0542079(*)
		Yes	Yes	0.0025507 ± 0.0003444(*)	0.0124773 ± 0.0018751(*)	0.8223575 ± 0.0337542(*)
		Yes	Yes	0.0027390 ± 0.0002842(*)	0.0123266 ± 0.0018478(*)	0.8262701 ± 0.0335686(*)

(*) indicates statistically significant improvements over the baseline based on Wilcoxon signed-rank test at significance level of 0.01

for different minimum exposures $e_{min} = 10, 100$ and 1000, respectively. For our SGD algorithm, we employ two variants that optimize for square and logistic loss functions respectively. Similarly, we utilize the cyclic and stochastic variants of our CD algorithm. For comparison, we also experiment with two popular time-aware factorization algorithms: TensorALS [5] and TimeSVD++ [9], which should serve as good reference methods. For fairness, similar configurations were used in these methods, e.g., 5 latent factors ($K = 5$) and 100 training iterations (as with our SGD).

To investigate the effects of importance-weighted and hierarchical learning, we conduct experiments by turning on and off the importance weight and hierarchy, resulting in four configurations for each algorithm variant. Subsequently, to see how significant the result improvements are due to importance weight and/or hierarchy, we use the non-parametric Wilcoxon signed-rank statistical test[2] [22]. All the tests were performed at a significance level of 1%, with the baseline being the "no importance weight and no hierarchy" case.

Several observations can accordingly be made based on the overall prediction results in Tables 3, 4 and 5:

- All our FM-based algorithms outperform the TensorALS and TimeSVD++ methods, in terms of both regression and ranking results. (Separate Wilcoxon tests using TensorALS and TimeSVD++ as baselines also show that all our algorithms are significantly better.) Note also that TensorALS gives "flat" predictions on the test set (where all the time indices never appear in the training set), leading to a constant $wAUC = 0.5$.

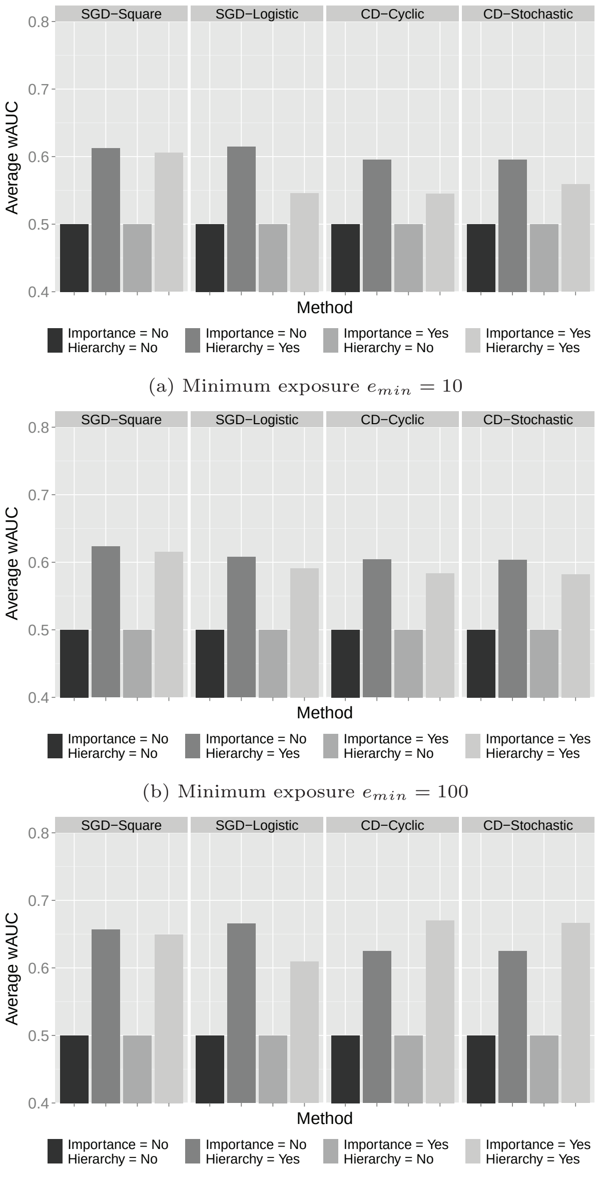
- As the minimum exposure threshold e_{min} increases, we see improvements in all performance metrics. This is expected, since a higher e_{min} implies a higher confidence in the CTR estimation and thus cleaner data for training and testing (recall that $CTR = \frac{\texttt{numclick}}{\texttt{numexpose}}$).

- We observe large improvements when we incorporate importance weights (i.e., importance-aware learning), while incorporating hierarchical information does not necessarily improve the overall prediction results. This is reasonable nonetheless, since we can expect that hierarchical learning helps in (near) cold-start cases, which would likely have small importance weights in our (weighted) evaluation on the test data. Further investigation on the performance of our methods in cold-start scenarios is presented in Section 6.4.

- In general, the CD algorithm produces similar performance to the SGD procedure, though the former needs fewer iterations to converge and has fewer user parameters (i.e., no learning rate η). Comparing the square and logistic SGD variants, we observe that the former generally exhibits better results in terms of $wRMSE$ and $wAUC$, while the latter is superior in terms of $wNLL$, which is expected. Meanwhile, the cyclic and stochastic CD variants yield very similar performances, though the stochastic approximation in the latter is expected to give faster training convergence.

6.4 Cold-Start Results

We carried out further studies on how hierarchical learning in our HIFM approach can help address the cold-start situations. In this study, we concentrate on the extreme cold-start scenarios whereby we deal with new pages *and* ads. We define new pages and ads as those that only appear in the test data, and never in the training set. We conducted the experiments as before, and averaged the results for these cold-start cases over 10 trials. We can expect that, when no hierarchical regularization or fitting takes place, our algorithms will produce a "flat' prediction (i.e., $wAUC = 0.5$). In this case, we may obtain reasonably good regression performance, but the ranking produced will be meaningless. Due to this reason (and space constraint), we shall focus on evaluating the ranking performance (i.e., $wAUC$) here.



(a) Minimum exposure $e_{min} = 10$

(b) Minimum exposure $e_{min} = 100$

(c) Minimum exposure $e_{min} = 1000$

Figure 4: Results for cold-start cases

Our experimental results are presented in Figure 4, with the minimum exposure e_{min} varied from 10 to 1000. It can be seen that the ranking results always improve as we incorporate hierarchy information in our algorithms. In addition, we observe that the result improvements get amplified as e_{min} is increased, which can (again) be attributed to the higher confidence in the CTR estimates (i.e., cleaner data). It is also worth noting that, for these cold-start cases, the results of importance-aware learning and those of unweighted learning are not directly comparable. In fact, the cold-start cases tend to have low exposures in the training data, for which the importance-aware learning would give lower priority. Nevertheless, the results illustrate that the importance-aware learning and hierarchy information are complementary in handling complex response prediction tasks.

[2] The Wilcoxon test can be used as an alternative to the t-test for matched pairs, or the t-test for dependent samples when they cannot be assumed to be normally distributed.

7. CONCLUSION

In this paper, we put forward a latent factor model, termed the *Hierarchical Importance-Aware Factorization Machine* (HIFM), for predicting dynamic ad response. Using the factorization machine as the base generic framework, we develop new importance-aware and hierarchical learning mechanisms to improve the model's predictive abilities in complex response prediction tasks where cost-varying instances and cold-start cases are ubiquitous. Several variants of efficient learning methods have been developed to explore different ways of incorporating importance weight and hierarchy information in HIFM. The efficacy of our HIFM model has been exemplified through extensive empirical studies on real-world mobile advertising data from a global ad network.

Moving forward, we consider several avenues for future research. First, there is a need to further scale up the HIFM for large datasets, either by means of parallelization or a more efficient data representation (e.g., [14]). Second, we can further enhance the HIFM's predictive power by incorporating hierarchical Bayesian learning on top of a deep, multi-layer representation. Last but not least, we wish to explore the applicability of the HIFM in other task domains, such as temporal item adoption or rating prediction.

Acknowledgment

This research is supported by Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office, Media Development Authority (MDA).

8. REFERENCES

[1] D. Agarwal, R. Agrawal, R. Khanna, and N. Kota. Estimating rates of rare events with multiple hierarchies through scalable log-linear models. In *Proceedings of the ACM SIGKDD Knowledge Discovery and Data Mining*, pages 213–222, 2010.

[2] D. Agarwal, B.-C. Chen, and P. Elango. Spatio-temporal models for estimating click-through rate. In *Proceedings of the International World Wide Web Conference*, pages 21–30, New York, NY, 2009.

[3] G. Aggarwal, A. Goel, and R. Motwani. Truthful auctions for pricing search keywords. In *Proceedings of the ACM Conference on Electronic Commerce*, pages 1–7, 2006.

[4] T. Bäck. *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford University Press, Oxford, UK, 1996.

[5] P. Comon, X. Luciani, and A. L. F. de Almeida. Tensor decompositions, alternating least squares and other tales. *Journal of Chemometrics*, 23:393–405, 2009.

[6] T. Fawcett. ROC graphs: Notes and practical considerations for researchers. Technical report, 2004.

[7] Gartner. Gartner says worldwide mobile advertising revenue to reach $11.4 billion in 2013. Available: http://www.gartner.com/newsroom/id/23062152013.

[8] T. Graepel, J. Q. Candela, T. Borchert, and R. Herbrich. Web-scale Bayesian click-through rate prediction for sponsored search advertising in Microsoft's Bing search engine. In *Proceedings of the International Conference on Machine Learning*, 2010.

[9] Y. Koren. Collaborative filtering with temporal dynamics. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 447–456, New York, NY, 2009.

[10] Z. Lu, D. Agarwal, and I. S. Dhillon. A spatio-temporal approach to collaborative filtering. In *Proceedings of the ACM conference on Recommender systems*, pages 13–20, 2009.

[11] A. K. Menon, K.-P. Chitrapura, S. Garg, D. Agarwal, and N. Kota. Response prediction using collaborative filtering with hierarchies and side-information. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 141–149, 2011.

[12] S. Rendle. Factorization machines. In *Proceedings of the IEEE International Conference on Data Mining*, Sydney, Australia, 2010.

[13] S. Rendle. Factorization machines with libFM. *ACM Transactions on Intelligent Systems and Technology*, 3:57:1–57:22, 2012.

[14] S. Rendle. Scaling factorization machines to relational data. In *Proceedings of the International Conference on Very Large Data Bases*, pages 337–348, 2013.

[15] M. Richardson. Predicting clicks: Estimating the click-through rate for new ads. In *Proceedings of the International World Wide Web Conference*, pages 521–530, 2007.

[16] D. Sculley. Combined regression and ranking. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 979–988, New York, NY, USA, 2010.

[17] S. Shalev-Shwartz and A. Tewari. Stochastic methods for l1 regularized loss minimization. In *Proceedings of the International Conference on Machine Learning*, pages 929–936, 2009.

[18] H. Shan, J. Kattge, P. B. Reich, A. Banerjee, F. Schrodt, and M. Reichstein. Gap filling in the plant kingdom–trait prediction using hierarchical probabilistic matrix factorization. In *Proceedings of the International Conference on Machine Learning*, Edinburgh, 2012.

[19] S. Shen, B. Hu, W. Chen, and Q. Yang. Personalized click model through collaborative filtering. In *Proceedings of the ACM International Conference on Web Search and Data Mining*, pages 323–332, 2012.

[20] N. Srebro, J. D. M. Rennie, and T. S. Jaakola. Maximum-margin matrix factorization. In *Proceedings of the Advances in Neural Information Processing Systems*, volume 17, pages 1329–1336, 2005.

[21] J. Z. Sun, K. R. Varschney, and K. Subbian. Dynamic matrix factorization: A state space approach. In *Proceedings of the IEEE International Conference on Speech and Signal Processing*, pages 1897–1900, Kyoto, Japan, 2012.

[22] F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945.

[23] E. Zhong, W. Fan, and Q. Yang. Contextual collaborative filtering via hierarchical matrix factorization. In *Proceedings of the SIAM International Conference on Data Mining*, pages 744–755, 2012.