Large Scale CVR Prediction through Dynamic Transfer Learning of Global and Local Features

Hongxia Yang HONGXIA@YAHOO-INC.COM

Quan Lu QLU@YAHOO-INC.COM

Angus Xianen Qiu QIUXE@YAHOO-INC.COM

Chun Han CHUNHAN@YAHOO-INC.COM

Yahoo! Inc, 701 1st Ave, Sunnyvale, California 94089

Abstract

This paper presents a combination of strategies for conversion rate (CVR) prediction deployed at the *Yahoo!* demand side platform (DSP) *Brightroll*, targeting at modeling extremely high dimensional, sparse data with limited human intervention. We propose a novel probabilistic generative model by tightly integrating components of natural language processing, dynamic transfer learning and scalable prediction, named Dynamic ***Transfer*** Learning with ***R***einforced ***W***ord ***M***odeling (a.k.a. ***Trans-RWM***) to predict user conversion rates. Our model is based on assumptions that: on a higher level, information can be transferable between related campaigns; on a lower level, users who searched similar contents or browsed similar pages would have a higher probability of sharing similar latent purchase interests. Novelties of this framework include (i) A novel natural language modeling specifically tailored for semantic inputs of CVR prediction; (ii) A Bayesian transfer learning model to dynamically *transfer* the knowledge from *source* to the future *target*; (iii) An automatic new updating rule with adaptive regularization using Stochastic Gradient Monte Carlo to support the efficient updating of *Trans-RWM* in high-dimensional and sparse data. We demonstrate that on *Brightroll* our framework can effectively discriminate extremely rare events in terms of their conversion propensity.

Keywords: CVR Prediction, Natural Language Processing, Dynamic Transfer Learning, Computational Advertisement.

1. Introduction

Display advertising has been the subject of rigorous research with extremely fast development during the past decade. This area has generated billions of revenue, originated hundreds of scientific papers and patents, saw a broad variety of implementations, yet the accuracy of prediction technologies leaves to desire more. Our work is motivated by the challenges from a world leading advertising platform *Brightroll*, which is the flagship of *Yahoo!*'s programmatic ad buying application suite that provides access to *Yahoo!* and third party inventories and capitalizes on relevant billions of users' data. Such data access allows us to find and target users across all available inventories while it is a big challenge to set up a flexible complete model framework that can consistently integrate information from different dimensions.

Advertisers in display advertising may design ad campaigns with different product goals in mind. Usually, advertisers can start several campaigns and each campaign is associated with a couple of ads. Some advertisers focus on building brand awareness for promoting products targeting at specific users, which is similar to television and magazine advertisement. Advertisers with this objective usually adopt cost-per-milli (CPM) model, which are priced in bundles of 1,000 impressions (or ads delivery). In such scenario, advertisers are charged by the number of impressions that are shown irrespective of user actions and the model performance is usually characterized by the demographic distribution of the targeted audiences. If advertisers care more about immediate sales, they usually prefer pricing types like cost-per-click (CPC) or cost-per-action (CPA). Actions may include credit card application, online course registration or products purchase. Criterion to characterize the performances of CPC and CPA models are click-through-rate (CTR) and conversion-rate (CVR) respectively. Compared to clicks, actions may need the targeted audiences to spend more efforts, thus CVR is usually much 10 to 100 times smaller compared to CTR and more challenging to model. Advertisers can also be somewhere in between and care both future and immediate sales thus adopt a mixture of the above pricing types.

In this paper, we focus on developing strategies for CVR prediction deployed on *Brightroll*. There are several challenges in successfully deploying a large scale CVR prediction model in practice (Mahdian and Tomak, 2007; Rosales et al., 2012; Chapelle et al., 2015). First, usually only a very small portion of the users that click or have been shown ads eventually convert. This constrains the modeling techniques to parsimoniously work with the data. Second, user profiles are high dimensional and sparse, ranging from user demographics to search queries and page browsing. Dealing with such different activities in the presence of limited conversion information is non-trivial. To add to this, the data is highly volatile due to cookie churn, changes in campaigns, variability in user interests and other temporal effects that do not allow accumulating long-standing data. These challenges require the modeling approach to have a quick start and dynamically adapt over time as new data accumulate.

1.1. Contributions

In view of these challenges, we propose a novel approach for conversion prediction that relies on two distinct sources of information: (a) The *global* features associated with the advertising campaign, such as campaign specific conversion and retargeting pixels. Advertisers instrument their ads with a pixel that gets triggered and stores the ad view information by the user (e.g. in the browser cookie or some user data store) when a user gets exposed to the ad on a publisher site. However, we can only get partial information of the pixel firing, e.g., we only get access to the complete information of converters that are attributed to us (e.g., conversions are lead by the ads that we showed before). For the left non-attributed majority, we are only informed the action time that are not recognized at user level. We use *global* features to characterize the external competitiveness and also the relationship between campaigns that we serve. (b) We define information directly related to users and user events as *local* features. We use *local* features to model the users' purchase behaviors and the key is that certain search queries or browsing content from certain user segments are relatively higher related to specific brand conversions.

We combine these two complimentary sources of information in a principled way and propose *Trans-RWM*, a novel probabilistic generative model that tightly integrates components of natural language processing, dynamic transfer learning and scalable prediction to support learning from the extremely sparse and high-dimensional conversion data. In summary, we make the following contributions in the paper:

1. We extend the word2vec (a.k.a, *W2V*, Mikolov et al. (2013a,b)) through the regularized Bayesian co-clustering to learn more reinforced word representations.
2. We propose a novel model for Bayesian transfer learning where the knowledge is dynamically transferred from *source* to the future *target* campaigns.

3. We connect part 1 and 2 through a novel probabilistic generative framework, named Dynamic ***Transfer*** Learning with ***R***einforced ***W***ord ***M***odeling (a.k.a. ***Trans-RWM***) to predict user conversion rates.
4. To automatically and efficiently learn *Trans-RWM* in the large scale sparse data on *Brightroll*, we design a new updating algorithm using Stochastic Gradient Monte Carlo.

2. Related Work

In this section, we briefly review *W2V* and transfer learning.

2.1. Word to Vector

We use *global* features to characterize the external competitiveness and relationship between the campaigns and use *local* features to model the users' purchase behaviors. Some of these features are contextual, e.g., campaign descriptions, user online search and browsing history, etc. Better word representations will help characterize the underlying action affinity that might lead to better predictive capabilities.

The recently popular *W2V* model Mikolov et al. (2013a,b) is an interesting method for learning distributed vector representations that can potentially capture a large number of precise syntactic and semantic word embeddings. The *W2V* engine is targeted at maximizing the conditional probability of the words under their context in the corpus with the skip-gram model (Mikolov et al., 2013a; Google):

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq l\leq c, l\neq 0}\log p(w_{t+l}|w_t), \tag{1}$$

where c is the size of the training context, which can be a function of the center word w_t. Larger c includes more training samples leading to higher accuracy but lower efficiency of model training. The original skip-gram formulations defines $p(w_{t+l}|w_t)$ through the softmax function:

$$p(w_O|w_I) = \frac{\exp({v'_{w_O}}^T v_{w_I})}{\sum_{w=1}^{W}\exp({v'_w}^T v_{w_I})}, \tag{2}$$

where v_w and v'_w are the "input" and "output" vector representations of w, and W is the number of words in the vocabulary. In this way, the vectors of the words are learnt and the dot product between vectors of a word and its possible neighbors are maximized. However, the exact solution of this formulation is impractical. Since the time complexity is proportional to the size of W, which is often on the scale of $10^5 - 10^7$.

Mikolov et al. (2013b) presented several extensions that improve both the quality of the learnt vectors and the training speed. They propose hierarchical softmax and negative sampling as computational efficient approximations of the full softmax. However, in practice, we notice that distances among word vectors are very often not comparable if the delta vectors are not in the same direction. In this paper, we extend the *W2V* model through the regularized Bayesian co-clustering to learn more reinforced word representations. The regularization enforces the priors of word vectors using a rectangular lattice across the vector space. Thus word vectors are more likely to be close to one of the lattice vertices after the training instead of staying at any random location in the vector space. This helps remove the ambiguity for measuring with small vector distances and thus more comparable. More detailed derivations are given in section 3.2.1.

2.2. Transfer Learning

Pan and Yang (2010) gives a comprehensive survey for transfer learning. At a high level, the idea of transfer learning includes (i) the learning (in part) is done for a task that differs from the real target task in either the sampling distribution of the examples, the features describing the samples, the exact quantity being modeled (the "label"), or the functional dependences between the features and the label, and (ii) that the knowledge obtained from this alternative learning task is transferred to the real task, somehow used to improve the learning in the target task. Only recently, Perlich et al. (2014) and Dalessandro et al. (2014) introduced transfer learning to model post-view conversions through combining data from general conversions to improve targeting performances across a large number of campaigns. To help better understand how transfer learning works in practice, we give a visualization in Figure 1 which shows the model updates in the dynamic evolving environment. Transfer learning is particularly useful when we do not have sufficient amount of labeled training data in some tasks, which may be very costly, laborious, or even infeasible to obtain. This deals exactly with the CVR prediction challenges.

Transfer learning is defined formally as following: a classification task $\{\mathcal{X}, p(X), \mathcal{Y}, p(Y|X)\}$ composes a feature space $\mathcal{X}$, a probability distribution $p(X)$ with $X \in \mathcal{X}$ of feature space, an outcome space $\mathcal{Y}$, an objective function $p(Y|X)$ where $Y \in \mathcal{Y}$. Throughout this paper, we will focus on the binary classification, with class labels $Y \in \{1, -1\}$ denoting conversion or not. In our particular situation, we refer to the current campaign data as the *target* and the auxiliary data from other streams as the *source*. We generally assume that the source and the target tasks share the same feature space $\mathcal{X}$ but $p^s(X) \neq p^t(X)$ for all X and we use superscript s and t to differentiate source and target. Outcome spaces $\mathcal{Y}$ are usually different but related. In our problem, $\mathcal{Y}^s$ represents conversion labels observing related traffic from other campaigns and the campaign itself (if exists) in the past. $\mathcal{Y}^t$ is derived from the campaign conversions for the most recent time period. $\mathcal{Y}^t$ become part of $\mathcal{Y}^s$ as time proceeds. We use the standard log-likelihood formulation and denote the sample logistic loss function as

$$l(\beta) = \sum_i l_i(\beta) = \sum_i -y_i \log p_i(\beta) - (1 - y_i) \log(1 - p_i(\beta)), \tag{3}$$

where $p_i(\beta) = 1/(1 + \exp(-\beta' X_i))$. Given the source data $\mathcal{D}^s = \{X^s, Y^s\}$, we optimize as

$$\beta^s = \arg\min_{\beta^s} \sum_{i=1}^{N^s} l_i^s(\beta^s) + \lambda^s r(\beta^s) \tag{4}$$

where $r(\cdot)$ is a suitable regularization function and λ^s is its regularizer. Denote the target data set as $\mathcal{D}^t = \{X^t, Y^t\}$. For the information transfer, β^s is included in the loss function and the target objective is optimized as following:

$$\beta^t = \arg\min_{\beta^t} \sum_{i=1}^{N^t} l_i^t(\beta^t) + \lambda^t r(\beta^t - \beta^s). \tag{5}$$

In order to meet the actual needs, we extend the above formulation to allow for incremental updating as we do not have only one *source* and one *target* but a series of them. More detailed derivations are given in Section 3.2.2.

3. The Proposed Framework

In this section, we describe the proposed *Trans-RWM* model for CVR prediction, which is a probabilistic generative framework that jointly models global campaign competitiveness and their relationship, purchase preferences discovery from users' search and browsing history, and dynamically transfer these knowledge across campaigns. The notations to be used is summarized in Table 1. In Section 3.1, we will first formalize our sampling strategy which is essential in training a generic CVR prediction model. In Section 3.2, we detailed the model

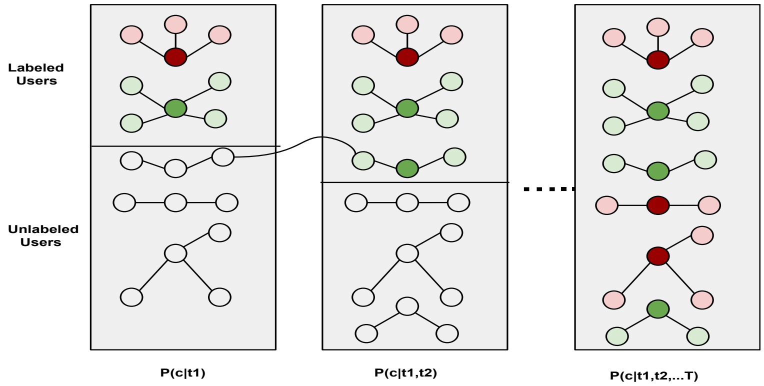

Figure 1: Transfer Learning Visualization: nodes represent users, edges stand for relationships between users and advertiser campaigns (edges exist if users converted for the advertiser campaigns before); dark green and red nodes represent labeled converted and non-converted users; light green and red represent previous interacted advertiser campaigns; gray nodes and edges stand for unlabeled users; $t_1, t_2, \ldots, T$ are sequential time stamps. As time evolves, information is dynamically transferred from the previous time stamps and model is updated accordingly.

Notation and Description	
a_i	ith advertiser
p_j	jth publisher
u_k	kth user segment
$z_{w,a_i p_j u_k}$	The topic assignment of word w in users' profile belonging to $\{a_i, p_j, u_k\}$
$\pi_{a_i p_j u_k}$	Topic distribution of words belonging to $\{a_i, p_j, u_k\}$
π_{a_i}, π_{p_j}, π_{u_k}	Topic distribution decompositions of advertiser, publisher and user dimensions
v_w	Vector representation for word w
$\mathbf{x}_i^s$ and $\mathbf{x}_i^t$	Source and target user features respectively, including contextual information of user online activity and user demographic information
y_i^s and y_i^t	Source and target user response respectively
β^s and β^t	Source and target feature weights vectors respectively
q^s and q^t	Source and target feature variances vectors respectively

Table 1: Notations used in *Trans-RWM* model

formulation and finalize this section with an efficient model updating algorithm in Section 3.3.

3.1. Sampling Strategy

Our goal is to train generic CVR prediction models to automatically serve hundreds of campaigns running on *Brightroll*. Existing CVR prediction models are usually trained based on attributed conversions, as well as the impressions from the same campaign shown by the DSP. For example, in post-view scenario, impression events are treated as positive training samples, if there are actions followed by and negative otherwise. However, in reality, this sampling strategy is insufficient. Firstly, impressions from a specific campaign is ad selection algorithm dependable and not representative of the non-converted users' generic behaviors. Secondly, considering only the winning impressions increase the selection bias in the generated training dataset. This is because impressions are purchased through bidding at public auctions. For one DSP, each bid typically contains only one ad, which has the highest evaluation among all the possible ads could be served by this DSP. Thus, the winning impressions and clicks are survivors from both external and internal competitions. Thirdly, for one DSP, the attributed conversions are only a small proportion of the total conversions. To conclude, the fundamental shortage of the commonly used sampling strategy is the partial audience representation. Here, we propose the following sampling strategy to mimic the real time bidding environment of both external and internal competitiveness to select samples that try to represent the whole population.

- **CASE 1** *Brightroll* attributed conversions (positive samples): For these samples, both *global* and *local* features are recorded when attributed impression or click happened. This CASE represents *Brightroll* attributed conversions.
- **CASE 2** Global conversions but not attributed to *Brightroll* (positive samples): For these samples, only *global* features can be recorded, and *local* features such as exact impression time and contextual information when user converted are not available. For each such sample, we randomly generate a time in the past click window (usually 3 days) or view window (usually 7 days) and use *local* features at that time for modeling. This CASE represents global conversions.
- **CASE 3** *Brightroll* generated campaign specific impressions or clicks but no conversions (negative samples): Both *global* and *local* features are recorded when impression or click happened. Down sampling rate is usually 1%. This CASE represents negative samples that we reached before.
- **CASE 4** No conversions and *Brightroll* did not generate campaign specific impressions or clicks (negative samples): For each sample, we randomly generate a time in the past view/click window and use *global* and *local* features at that time for modeling. Down sampling rate is usually 0.01%. This CASE represents negative samples that we haven't had a chance to reach.

Since conversions are rare, we choose a relatively large downsampling rates for CASE 3 and 4 (negative samples) according to our experience.

3.2. Model Formulation

Our proposed CVR prediction framework mainly consists of two parts: 1. A novel natural language-based algorithm that extends from *W2V* and is specifically tailored for both *global* and *local* features; and 2. A novel dynamic Bayesian transfer learning prediction framework that includes *global* and *local* semantic features processed from part 1. We then propose a probabilistic generative framework *Trans-RWM* to tightly integrate part 1 and 2.

3.2.1. Part 1: An Extension of W2V

W2V is a popular tool for word proximity processing. As defined in *W2V* (Mikolov et al., 2013a), the objective of the skip-gram model is to maximize the average log probability:

$$L(\mathbf{v}) = \frac{1}{T}\sum_{t=1}\sum_{-c\leq l\leq c, j\neq 0} \log p(w_{t+l}|w_t) = \frac{1}{T}\log\prod_{t=1}\prod_{-c\leq l\leq c, l\neq 0} p(w_{t+j}|w_t), \tag{6}$$

where c is the size of the training context, which can be a function of the center word w_t. In this paper, we consider the hierarchical softmax as introduced in Mikolov et al. (2013b). More specifically, each word w can be reached by an appropriate path from the root of the tree. Let $n(w, l)$ be the l-th node on the path from the root to w and $l(w)$ be the length of this path. So $n(w, 1)$=root and $n(w, l(w)) = w$. In addition, for any inner node n, let $\mathrm{ch}(n)$ be an arbitrary fixed child of n. Different from the original formulation, we use the frequency that a fixed node n will be the child for w and the objective function is reformulated as (for simplicity, we discard the constant 1/T):

$$\begin{aligned} L(\mathbf{v}) &\propto \log \prod_{w} \prod_{l=1}^{l(w)-1} \sigma\Big(\#\{n(w,l+1) \in \mathrm{ch}(n(w,l))\} \times v'_{n(w,l)} v_w\Big) \\ &= \log \prod_{w} \prod_{l=1}^{l(w)-1} \sigma\big(v'_{n(w,l)} v_w\big)^{\#\{n(w,l+1) \in \mathrm{ch}(n(w,l))\}} \end{aligned}$$

where $\sigma(v_w) = 1/(1+\exp(-v_w))$ is the sigmoid function; v_w and v'_n are the "input" vector representation for each word w and "output" vector representation for every inner node n of the binary tree. We denote d_{wl} the frequency that $n(w,l+1) \in \mathrm{ch}(n(w,l))$ and abbreviate $v_{n(w,l)}$ as v_{wl}. The above formulation can be simplified as following:

$$L(\mathbf{v}) \propto \log \prod_{w} \prod_{l=1}^{l(w)-1} \sigma\big(v'_{wl} v_w\big)^{d_{wl}}.$$

We can prove the following facts that are convenient for later model learning:

$$\begin{aligned} \frac{\partial \sigma(v_w)}{\partial v_w} &= \sigma(v_w)(1-\sigma(v_w)), \\ \frac{\partial \log \sigma(v_w)}{\partial v_w} &= 1-\sigma(v_w), \\ \frac{\partial L(v_w)}{\partial v} &= \sum_{l=1}^{l(w)-1} d_{wl}\big(1-\sigma(v'_{wl} v_w)\big) v_{wl}. \end{aligned} \tag{7}$$

For the contextual information that are collected in our scenario, there is a unique tuple mapping {User, Publisher, Advertiser} (e.g., the specific user browsed a publisher web page and purchased a product form the ads shown on the page) and the current set up of *W2V* cannot easily include this important structure in the modeling. We also notice that the resulting word vectors from *W2V* are very often not comparable if the delta vectors are not in the same direction. With these limitations, we are considering the following regularized multivariate mixture Lasso prior for v_w that can help both embed the tuple structure and also account for the sparsity of the individual word representation:

$$v_w \sim \sum_{i,j,k} \pi_{a_i p_j u_k} f(v_{a_i p_j u_k}), \tag{8}$$

where $f(\cdot)$ is the multivariate lasso distribution (West, 1992; Park and Casella, 2008): $f(v_w) \sim \text{Normal-InverseGamma}\big(0, \sigma^2_{1:q}, \frac{\alpha}{2}, \frac{\alpha}{2}\big)$ and v_w is a q dimensional vector. This is a regularization prior that accounts for the sparsity of the individual word representation. To include the tuple structure of {User, Publisher, Advertiser} we decompose $\pi_{a_i p_j u_k}$ as following:

$$\pi_{a_i p_j u_k} \sim \mathrm{Beta}\big(c\pi_{a_i}\pi_{p_j}\pi_{u_k}, c(1-\pi_{a_i}\pi_{p_j}\pi_{u_k})\big), \tag{9}$$

where π_{a_i}, π_{p_j} and π_{u_k} are the marginal probabilities of the a_ith Advertiser, p_jth Publisher u_kth User segment respectively.

To conclude, in our formulation, the regularization enforces the priors of word vectors using a rectangular lattice ({User, Publisher, Advertiser}) across the vector space. Thus word vectors are more likely to be close to one of the lattice vertices after the training instead of staying at any random location in the vector space. This helps remove the ambiguity for measuring with small vector distances and makes the distances among the resulting word vectors more comparable (the limitation of the original *W2V*). The tuple structure that is embedded in the modeling can help us achieve better performance through borrowing information from related words according to their contexts.

3.2.2. Part 2: Bayesian Transfer Learning Model

As the positive labeled data are extremely sparse, we would like to design a framework that can dynamically transfer the knowledge from *source* (or related campaigns) to the *target* (or the current running campaign). Transfer learning is particularly useful when we do not have sufficient amount of labeled training data in some tasks, which may be very costly, laborious, or even infeasible to obtain. This deals exactly with the CVR prediction challenges. However, the current transfer learning is not easy for incremental updating and could not fit very well in our situation: as we do not have only one *source* and one *target* but a series of them. A major advantage of Bayesian logistic regression is that it can be naturally adapted to the online update setting. So to extend to the dynamic transfer learning, we accommodate the regularized Bayesian logistic regression in the transfer learning as following. We first estimate $\{\beta^s, q^s\}$ from the source dataset $\mathcal{D}^s = \{X^s, Y^s\}$ through:

$$p(\beta^s, q^s|\mathcal{D}^s) \propto \Big\{\prod_{i=1}^{n^s} p(y_i^s|x_i^s, \beta^s, q^s)\Big\} p(\beta^s, q^s), \tag{10}$$

with

$$p(y_i^s = 1|x_i^s, \beta^s) = \Phi(\beta^{s\prime} x_i^s), \tag{11}$$

and $\Phi(\cdot)$ is the logistic link function $\Phi(z) = \frac{e^z}{1+e^z}$. $p(\beta^s, q^s|\mathcal{D}^s)$ on the left hand side is the required posterior distributions of β^s and q^s given the data set $\mathcal{D}^s$, $\big\{\prod_{i=1}^{n^s} p(y_i^s|\beta^s, q^s)\big\}$ on the right hand side is the likelihood of $\mathcal{D}^s$ and $p(\beta^s, q^s)$ is the prior distribution. Equation (10) is equivalent to Equation (4), though the former will learn distributions of the coefficients and the latter will only supply point estimations.

We then estimate the posterior of β^t for the target dataset $\mathcal{D}^t = \{X^t, Y^t\}$ using the priors that we learnt from the *source*:

$$p(\beta^t, q^t|\mathcal{D}^t, \beta^s, q^s) \propto \Big\{\prod_{i=1}^{n^t} p(y_i^t|\beta^t, q^t)\Big\} p(\beta^t, q^t|\beta^s, q^s) \tag{12}$$

Similarly, Equation (12) is equivalent to Equation (5) but supply distribution estimations. Notice that we work in a dynamic environment, and as time proceeds, target data can become source data and the way that we integrate transfer learning with penalized Bayesian logistic regression gives us more flexibility.

Different penalties have been considered: Bayesian Lasso (Tibshirani, 1994; Park and Casella, 2008) (a.k.a, L1) and Bayesian Tikhonov regularization or Ridge regression (Marquardt and Snee, 1975) (a.k.a, L2). However, we could not get satisfactory results with L1 penalty in reality and we believe that for dynamic updates the smoothness of L2 will be more beneficial for our problem. The L2 penalty corresponds to the Bayesian logistic regression with the normal distribution as the prior. The posterior distribution of β^t and q_t are thus proportional to

$$\prod_{i=1}^{n^t} \frac{1}{1+\exp\left\{-y_i^t \beta^{t\prime} x_i^t\right\}} \prod_{j=1}^{d} \frac{1}{\sqrt{2\pi/q_j^t}} \exp\left\{-\frac{q_j^t(\beta_j^t - \beta_j^s)^2}{2}\right\} \tag{13}$$

where the prior $\beta_j^t \sim \mathcal{N}(\beta_j^s, 1/q_j^s)$ with β_j^s and $1/q_j^s$ being prior means and variances learnt from the source data $\mathcal{D}^s$.

3.2.3. Dual Part Bridge: Model Generative Process

We connect Part 1 and Part 2 through the following probabilistic generative process. The graphical representation of *Trans-RWM* is also presented in Figure 2, and the detail is summarized as following:

1. For each word w that is assigned to $\{a_i, p_j, u_k\}$th tuple:
   (a) Draw the topic assignment of each dimension $z_{w,a_ip_ju_k}$ through
   $$p(z_{w,a_ip_ju_k}) \sim \text{Multinomial}\big(\pi_{a_ip_ju_k} f(v_{a_ip_ju_k})\big). \tag{14}$$
   (b) Draw the topic probability through
   $$\pi_{a_ip_ju_k} \sim \text{Beta}\big(c\pi_{a_i}\pi_{p_j}\pi_{u_k}, c(1-\pi_{a_i}\pi_{p_j}\pi_{u_k})\big). \tag{15}$$
   (c) Draw the baseline distribution of the q dimensional vector representation from
   $$f(v_{a_ip_ju_k}) \sim \text{Normal-InverseGamma}\big(0, \sigma^2_{1:q}, \frac{\alpha}{2}, \frac{\alpha}{2}\big). \tag{16}$$
2. For each user i at target time t, draw a response variable
   $$p(y_i^t = 1 | \mathbf{x}_i^t, \beta^t) = \Phi(\beta^{t'} \mathbf{x}_i^t), \tag{17}$$
   where $\mathbf{x}_i^t = (\mathbf{v}_i^{t'}, \mathbf{xu}_i^{t'})'$ and $\mathbf{v}_i^t$ is user search and browsing related word vectors from Step 1, $\mathbf{xu}_i^t$ is user demographical information. $\Phi(\cdot)$ is the logit link function.
3. Update the target coefficients distribution from
   $$p(\beta^t, q^t | \mathcal{D}^t) \propto \Big\{\prod_{i=1}^{n^t} p(y_i^t | \beta^t, q^t)\Big\} p\big(\beta^t, q^t | \beta^s, q^s\big). \tag{18}$$

Given model parameters $\Theta = \{\{\sigma^2\}_{1:q}, \alpha, \{v_{a_ip_ju_k}\}, \beta^t, q^t, \beta^s, q^s\}$, the joint probability of the observed and hidden variable is

$$\begin{aligned} P\big(\{z_{w,a_ip_ju_k}\}^{I,J,K}_{a_ip_ju_k}, \{x_i^t, y_i^t\} | \Theta\big) \quad = \quad & \prod_{w \in \{a_i, p_j, u_k\}} p(\pi_{a_ip_ju_k} | \pi_{a_i}, \pi_{p_j}, \pi_{u_k}) p(z_{w,a_iu_jp_k} | \pi_{a_ip_ju_k}, v_{a_ip_ju_k}) p(v_{a_ip_ju_k} | \sigma^2_{1:q}, \alpha) \\ & \times \prod_i p(y_i^t | \mathbf{x}_i^t, \beta^t) p(\beta^t, q^t | \beta^s, q^s). \end{aligned} \tag{19}$$

3.3. Model Learning: Stochastic Gradient Monte Carlo

In *Brightroll*, we have to deal with billions of bid requests each day, so algorithm efficiency is critical. Although the Markov Chain Monte Carlo (MCMC) algorithm is straightforward for *Trans-RWM*, efficiency problems will arise in our situation to scale to such high dimensions. We adapt the Stochastic Gradient Monte Carlo (SGMC, Welling and Teh (2011)) for *Trans-RWM* and derive some fundamental steps for the updating as follows. The complete steps are listed in Algorithm 1. First, with Facts (7) we can show that the gradient for the baseline distribution of $\{a_i, p_j, u_k\}$th word vector cluster is:

$$g(v^*_{ijk}) \quad = \quad \sum_{w \in \{a_i, p_j, u_k\}} \sum_{j'=1}^{l(w)-1} d_{wj'} \big(\sigma(v'_{wj'} v_w) - 1\big) v_{wj'} + \sum_{t=1}^{p} \sigma_t^2 v^*_{ijk,t} \tag{20}$$

Notice that $v_w = v^*_{ijk}$ if $z_{w,ijk} = 1$. And the Hessian matrix is:

$$H(v^*_{ijk}) = \sum_{w \in \{a_i, p_j, u_k\}} \sum_{j'=1}^{l(w)-1} d^2_{wj'} v'_{wj'} \mathbf{R} v_{wj'} + \sigma^{\mathbf{2}} I, \tag{21}$$

Algorithm 1 *Trans-RWM* Updating Algorithm

```
Input: $x_i^t$: High dimensional local and global features;
    $y_i^t$: Labels for $i = 1, 2, \cdots, n^t$
for $t = 1, \ldots, T$ do
    for Iterations Until Convergence do
Input: Words collected for each record which are naturally connected to {User, Publisher, Advertiser}.
        for Iterations Until Convergence do
            Update $\pi_{ijk}$ through:
                $\pi_{ijk} \sim \text{Beta}\big(c\pi_{a_i}\pi_{p_j}\pi_{u_k} + \sum z_{w,ijk},$
                $c(1 - \pi_{a_i}\pi_{p_j}\pi_{u_k}) + N - \sum z_{w,ijk}\big)$
Output: $z_{w,ijk} = 1$ if $v_w$ is assigned to $\{a_i, p_j, u_k\}$th cluster and $N$ the total number of observations.
            Update $z_{w,ijk}$ as following:
                $p(z_{w,ijk}|\cdots) \propto \pi_{ijk} f(V_{ijk}^*)$
            $f(v_{ijk}^*)$ can be formulated as following through scale mixture normal distribution:
                $v_{ijk,t}^* \sim \text{N}(0, \sigma_t^2), \text{ for } t = 1, \ldots, p,$
                $\sigma_{1:p}^2 \sim \text{IG}(\alpha/2, w\alpha/2).$
Input: The gradient in Eq(20) and Hessian matrix in Eq(21)
            The baseline distribution of $\{a_i, p_j, u_k\}$th cluster is:
                $\prod_{w \in \{a_i, p_j, u_k\}} \prod_{c=1}^{l(w)-1} \sigma\big(v'_{wc} v_{ijk}^*\big)^{d_{wj}} f(v_{ijk}^*)$
Output: $v_w = v_{ijk}^*$ if $z_{w,ijk} = 1$
        end for
        Update the posterior distribution of $v^t$ with the results from Laplace approximation. Update $x_i^t$ with $v^t$.
Output: $\beta^t$ is updated through Eq (25) and $q_j^t$ is updated through Eq (26).
    end for
end for
```

where $\boldsymbol{\sigma^2} = \{\sigma_t^2\}_{t=1}^p$ and $\mathbf{R} = \text{diag}(\sigma(v_w)(1 - \sigma(v_w)))$. We use Laplace approximation to update v_{ijk}^* as:

$$v_{ijk}^* \sim \text{MN}\big(g(v_{ijk}^*), H(v_{ijk}^*)\big). \tag{22}$$

The following steps are repeated until convergence:

$$\{v_{ijk}^*\}_i = \{v_{ijk}^*\}_{i-1} - H(v_{ijk}^*)^{-1} g(v_{ijk}^*). \tag{23}$$

Second, a major advantage for the Bayesian logistic regression is that it can be naturally adapted to the online update setting with Laplace approximation. With the normal prior distribution $\beta_j^t \sim \mathcal{N}(\beta_j^s, 1/q_j^s)$, the posterior log-likelihood can be rewritten as

$$\sum_{i=1}^{n^t} \log \left\{1 + \exp\left(-y_i^t \beta^{t\prime} x_i^t\right)\right\} + \sum_{j=1}^{d} \frac{q_j^t(\beta_j^t - \beta_j^s)^2}{2} \tag{24}$$

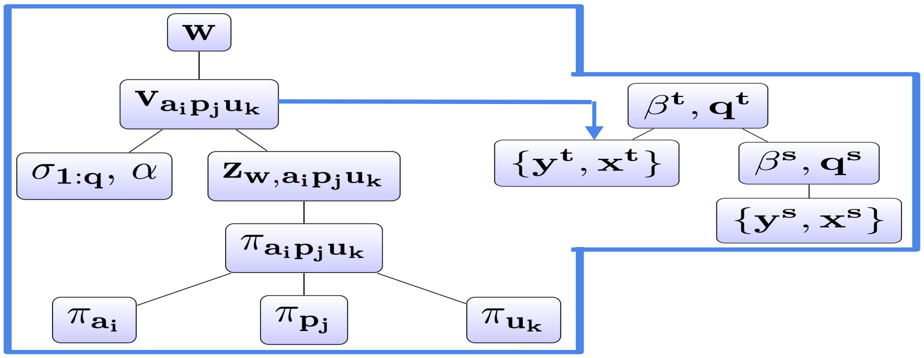
$\mathbf{w}$
$\mathbf{V_{a_i p_j u_k}}$
$\sigma_{1:q},\ \alpha$
$\mathbf{z_{w,a_i p_j u_k}}$
$\pi_{\mathbf{a_i p_j u_k}}$
$\pi_{\mathbf{a_i}}$
$\pi_{\mathbf{p_j}}$
$\pi_{\mathbf{u_k}}$
$\beta^{\mathbf{t}}, \mathbf{q^t}$
$\{\mathbf{y^t}, \mathbf{x^t}\}$
$\beta^{\mathbf{s}}, \mathbf{q^s}$
$\{\mathbf{y^s}, \mathbf{x^s}\}$

Figure 2: Generative Model Visualization: w, $\{y^t, x^t\}$, and $\{x^s, y^s\}$ are observations, the left are parameters that need to learn.

up to some constant. Laplace approximation yields a normal posterior distribution which could be used as a prior distribution for the next *source* data. In this way we could sequentially update the model through the training of each time stamp. In particular, the posterior distribution of β^t is still normal distribution with mean

$$\beta^t = \arg\min_{\beta^t} \sum_{i=1}^{n^t} \log\left(1 + \exp\left(-y_i^t \beta^{t\prime} x_i^t\right)\right) + \sum_{j=1}^{d} \frac{q_j^s(\beta_j^t - \beta_j^s)^2}{2} \tag{25}$$

and variance

$$q_j^t = q_j^s + \sum_{i=1}^{n^t} x_{ij}^t{}^2 p_i^t(1-p_i^t), \quad p_i^t = (1+\exp(-\beta^{t\prime} x_i{}^t))^{-1}. \tag{26}$$

We generalize the SGMC for *Trans-RWM* in **Algorithm 1**. To deploy on *Brightroll*, we implement through Spark and its optimization package L-BFGS [1] by developing efficient and encapsulated interface to integrate Spark and *Trans-RWM* to tackle large datasets. Convergence of β_j^t is done through checking if $\|\nabla p(\beta_j^t|D_t^t)\| = 0$. And for practical reasons, we will always employ a tolerance parameter when checking this condition.

4. Experimental Results

In this section we demonstrate empirically the improvements induced by *Trans-RWM*. The analyses have been conducted experimentally on the whole platform with significant performance lifts. Due to the reason of confidentiality, we could not report all campaign results from *Brightroll*. But in order to have a better understanding of the approach performance and the benefits it brings in, we report results from 50 different randomly chosen campaigns, belonging to a wide variety of advertisers from different industries. Different campaigns have very different CVRs and CPA goals and thus possess various inherent predictabilities. A quick summary goes as follows: 1) *Global* features can help characterize the external competitiveness, the relationship between campaigns, and thus are very helpful for the effective information transfer; 2) Incorporate user online activity similarities (*local* features) into their ad response prediction modeling indeed brought in improvements; 3) Using the *Trans-RWM* which transfers distributions of the coefficients achieves better results compared to with no transfer learning (Chapelle et al., 2015) or transferring point estimators alone (Dalessandro et al., 2014). In all of these experiments, we perform and report experimental results conducted on data from each individual campaign. The source and target data sets used are specific to each particular campaign.

[1] https://spark.apache.org/docs/1.2.0/mllib-optimization.html

Chapelle et al. (2015) (a.k.a, *Max Ent*) and Dalessandro et al. (2014) (a.k.a, *Transfer Learning*) are the methods that are most similar to ours in design. We will not compare with typical strategies such as combining samples from the target campaigns and other related campaigns with different weights for model training, since it is a simplified case of transfer learning and significant improvements over these baselines were already noticed in Dalessandro et al. (2014). Compared to *Max Ent* and *Transfer Learning*, there are several fundamental differences and extensions in our work. First, they do not use *global* features as we defined to neither characterize the external competitiveness nor campaign relationship. Second, neither *Max Ent* nor *Transfer Learning* considers modeling users' online activities and proposes a probabilistic generative model by tightly integrating users' purchase preferences from online activities to conversion prediction. Third, Dalessandro et al. (2014)'s work transfers point estimations of coefficients while in our framework, we automatically include their distributions. Chapelle et al. (2015)'s work does not include information from source but only from target data. Fourth, neither work has proposed a comprehensive sampling strategy that can mimic the real time bidding environment of both external and internal competitiveness. In order to show the benefits induced by the above four extensions and improvements, we will focus on the importance of users' interests transfer in Section 4.1 and the transfer sensitivity of source campaigns in Section 4.2.

4.1. Importance of Users' Interests Transfer

In this subsection, we focus on studying the effects by including the semantic information that we learnt from the user search and browsing history.

Our first set of experiments was conducted on data from a single time frame spanning two time periods. We carefully chose 10 related campaigns and for each campaign created three data sets: (1) a source data set $\mathcal{D}_0^s$ in period τ_0, (2) a target data set $\mathcal{D}_0^t$ in period τ_0 and (3) a target data set $\mathcal{D}_1^t$ in period τ_1. In this scenario periods τ_0 and τ_1 are disjoint but consecutive. The sources here are extracted from data other than the target campaign (e.g., the other 9 campaigns). With this set up, we can consider that the source data are reliable and informative for target data. We perform the experiments on *Max Ent*, *Transfer Learning* and *Trans-RWM* with and without semantic information. For the "semantic" variant, we train the whole model of Equation (19) with $\mathcal{D}^s = \{\mathcal{X}^s, \mathcal{Y}^s\}$ and $\mathcal{D}^t = \{\mathcal{X}^t, \mathcal{Y}^t\}$. For the "non-semantic" variant, we will only consider $\mathcal{D}_u^s = \{\mathcal{X}_u^s, \mathcal{Y}^s\}$ and $\mathcal{D}_u^t = \{\mathcal{X}_u^t, \mathcal{Y}^t\}$. To recap, $\mathbf{x}_i^t = (\mathbf{v}_i^{t\prime}, \mathbf{xu}_i^{t\prime})'$, where $\mathbf{v}_i^t$ is the contextual information of user search and browsing, $\mathbf{xu}_i^t$ is user demographical information. $\mathbf{xu}_i^t \in \mathcal{X}_u^t$, $\mathbf{x}_i^t \in \mathcal{X}^t$ and $\mathbf{xu}_i^s \in \mathcal{X}_u^s$, $\mathbf{x}_i^s \in \mathcal{X}^s$.

We first collect the users' online history from *Brightroll*, tokenize and stemming the query phrases into tokens. Then, the online history tokens for each user in the previous 30 days are combined together to form the profile document of the user, which are recorded in our campaign logs that compose the training corpus for *Trans-RWM*. We use AUC to compare the model performances (McMahan et al., 2013; He et al., 2014). AUC is defined as the algorithms' areas under the receiver operating characteristic (ROC) curve, which is usually used to quantify the quality of the predicted ranking that results from the algorithm according to the predicted probability.

For this set of experiments we use a 3x3 factorial design to examine the modeling. The first variant is the "**Semantic**" information. For the "Semantic" variant, we use the complete information $\mathcal{D}^s$ and $\mathcal{D}^t$ and use the partial information $\mathcal{D}_u^s$ and $\mathcal{D}_u^t$ for the "Non Semantic" variant. The second experimental factor is whether or not we use transfer learning. For the "**Transfer Learning**" variant, we first learn β^s and q^s as in Equation (4) using the source data $\mathcal{D}^s$. We then use $\mathcal{D}^t$ and β^s, q^s as in Equation (5) to optimize β^t and

q^t. This is similar to *Transfer Learning*(Dalessandro et al., 2014). For the "No Transfer" (control) variant, we only train Equation (5) using $\mathcal{D}^t$ and do not include β^s and q^s from $\mathcal{D}^s$. This set up is the same as *Max Ent*(Chapelle et al., 2015). The third design factor represents the coefficients distribution transfer. For the "**Distribution**" variant, we train *Trans-RWM* and for the "No Distribution" variant, we train *Transfer Learning* which only transfers point estimators.

First, we look at the improvements induced by the **Semantic** information included in the model *Trans-RWM*. Improvements are calibrated using the increments of AUC. On average, around 15.34% of users have search or browsing history. In general, the larger the proportion of users that have search or browsing history, the more increments of AUC will be induced by *Trans-RWM*. In Figure 3, the benefits from the **Semantic** information is also obvious by comparing the results of *Trans-RWM* with and with no **Semantic** information where *Trans-RWM* always achieves better AUC compared to *Trans-RWM* with no **Semantic** information. In Figure 3, we also report the 4 model performances of the 10 campaigns. Overall, *Trans-RWM* outperforms all the other competitors. *Transfer Learning* ranks next to *Trans-RWM* and we believe that transferring the **Distributions** of the coefficients will give us more benefits compared to transferring the point estimators alone where the latter will not always be robust. Notice that we train *Transfer Learning* with complete information (including semantic data). *Trans-RWM* with no semantic information usually ranks the 3rd overall which means lacking the users' online activity information diminishes even more model performances (or the "Transfer" variant is more important compared to "Distribution" variant in practice). *Max Ent* ranks the worst and it is clear that both "Transfer Learning" and "Semantic" variants are necessary in CVR prediction where the conversions are extremely sparse.

4.2. Transfer Sensitivity of Source Campaigns

In this subsection, we focus on studying the effects of the transfer sensitivity of the source campaigns. In Section 4.1, 10 campaigns are selected with care so the source data are reliable and informative. In this section, we randomly choose 50 campaigns and would like to see how different models perform when the source information is mixed together.

We extensively compare *Trans-RWM* with *Max Ent* and *Transfer Learning* from different aspects that should be paid most attention for *online* performances. These measures include AUC, CVR and business performance index (BPI). In online A/B testing, total spending and eCPA (expected cost for each action) are the two most important criterion. In order to quantify the performance that can reflect these two criterion in a consistent way, we proposed the following BPI:

$$\mathrm{BPI} = \frac{\mathrm{rev.test} + (\mathrm{cost.ctrl} - \mathrm{cost.test})}{\mathrm{rev.ctrl}}. \tag{27}$$

where *rev.test* and *rev.ctrl* are calculated through number of conversions multiplied by CPA goal and *cost.test* and *cost.ctrl* are mainly inventory costs. BPI characterizes the profit margin improvement.

We run the *Max Ent*, *Transfer Learning* and *Trans-RWM* dynamically for 20 days using the complete data (including semantic information) and report AUC, CVR and BPI in Figure 4. Data before day t are considered as source data $\mathcal{D}^s$ and data on day t are considered as target data $\mathcal{D}^t$. As we can see, at very beginning, the performances of *Max Ent*, *Transfer Learning* and *Trans-RWM* are mixed together. However, as time goes on, *Trans-RWM* achieves more and more improvements compared to *Max Ent* and *Transfer Learning*. Overall, *Trans-RWM* ranks the best, *Transfer Learning* ranks the second and *Max Ent* ranks the worst. Especially there is almost no change for the performances of *Max*

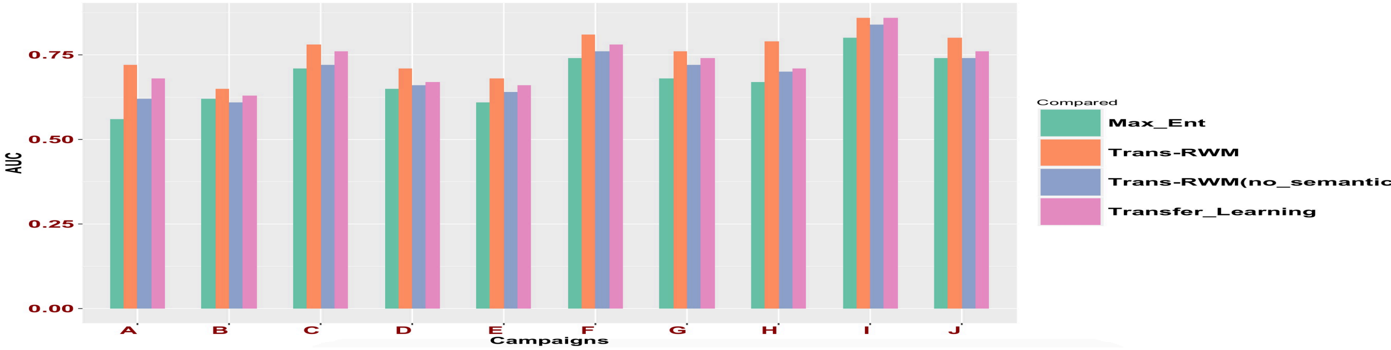

Figure 3: Users Interests Transfer Comparison Reports: overall, *Trans-RWM* with complete information ranks the first, *Transfer Learning* with complete information ranks the next, *Trans-RWM* with no semantic information ranks the 3rd and *Max Ent* ranks the worst. "Transfer" variant is more important compared to "Distribution" variant in practice.

Ent over the time, while both *Transfer Learning* and *Trans-RWM* can identify more useful information as time proceeds. From these results, we can generalize several conclusions. First when comparing to *Transfer Learning*, the positive effects of transfer learning alone wears off overtime compared to *Trans-RWM*, although *Transfer Learning* is indeed effective for improving campaign performances especially in early stages of the campaigns. Two possible reasons here: point estimations are not robust and sufficient compared to carried over distributions; dynamic transfer is critical in CVR prediction. Second, the benefits of CVR and BPI are more pronounced after deploying *Trans-RWM* compared to *Transfer Learning*. Third, as time proceeds, *Trans-RWM* can discriminate reliable source information and incur more benefits even though the source information are mixed together at very beginning.

5. Conclusions

In this paper we propose a novel two-stage modeling framework *Trans-RWM* for CVR prediction. This work has several key contributions. First, the newly proposed natural language modeling results in a good improvement in the quality of the learnt word and phrase representations. By extending the *W2V* model through learning sparser word representations while borrowing information across similar clusters, word vectors are more likely to be close to one of the lattice vertices after the training instead of staying at any random location in the vector space. This helps to remove the ambiguity for measuring with small vector distances and makes the distances among the resulting word vectors more comparable. Second, we propose a dynamic Bayesian transfer learning model accompanied with an automatic new updating rule using Stochastic Gradient Monte Carlo to dynamically transfer the knowledge from source to the future target. This is motivated by the goal of transferring knowledge dynamically instead of training expensive models. *Trans-RWM* learning provides an attractive framework for representing, learning, and reasoning about shared information. Our focus here is on producing a scalable, accurate, and robust system. We have achieved that through tightly integrating components of natural language processing, dynamic transfer learning and scalable prediction to support learning from extremely sparse, high-dimensional data with adaptive regularization in a very efficient way. To our knowledge, this is among the pioneering works that consider applying these impor-

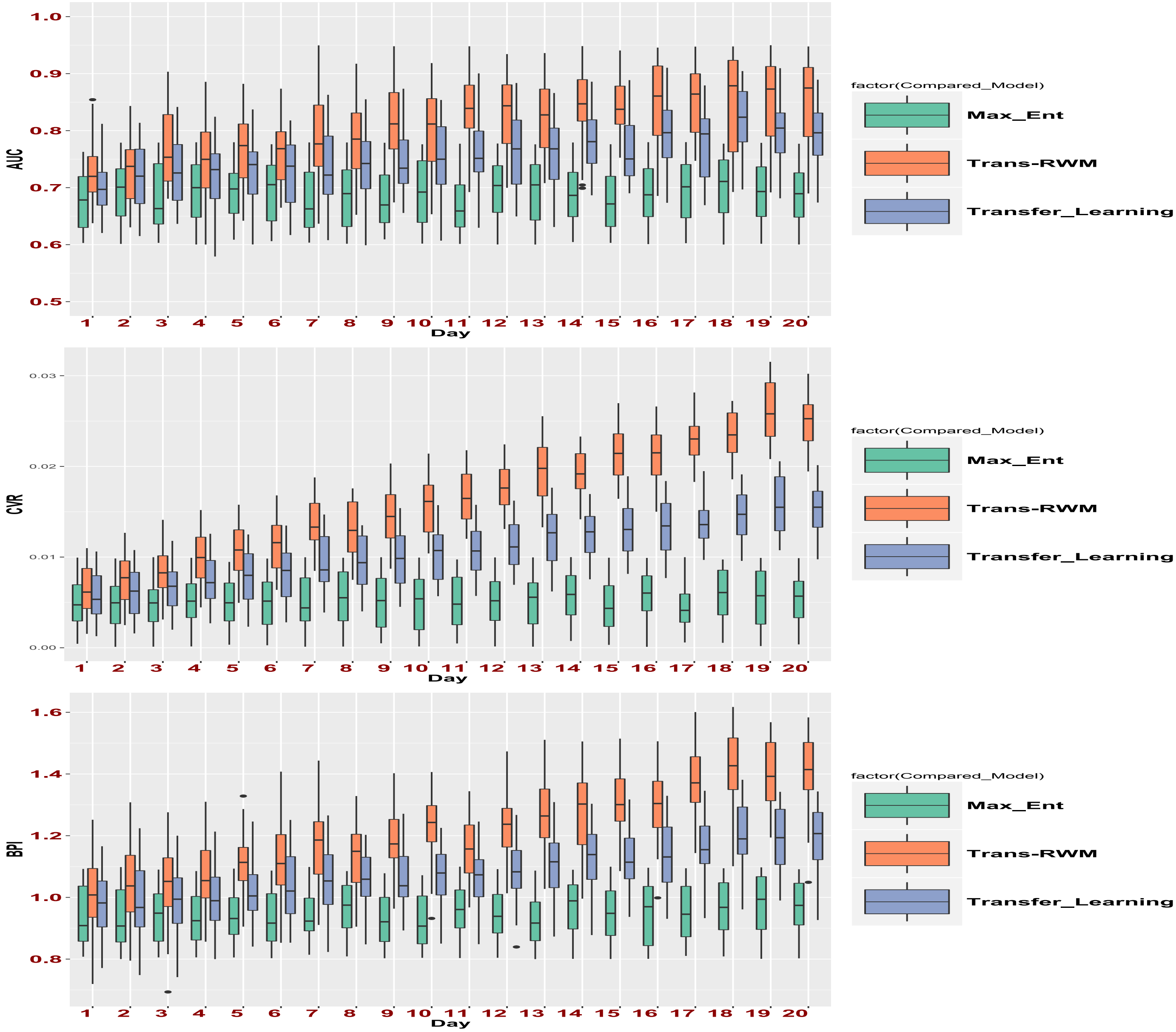

Figure 4: Dynamic Transfer Sensitivity of Source Campaigns: the positive effects of *Transfer Learning* wears off overtime compared to *Trans-RWM*; the benefits of CVR and BPI are more pronounced after deploying *Trans-RWM* compared to *Transfer Learning*; *Trans-RWM* can discriminate reliable source information and incur more benefits even though the source information are mixed together at very beginning.

tant components in the context of CVR prediction, which arguably the learning setting where Bayesian methods may have the most impact.

References

O. Chapelle, E. Manavoglu, and R. Rosales. Simple and scalable response prediction for display advertising. *ACM Trans. Intell. Syst. Technol.*, 5(4):61:1–61:34, 2015.

B. Dalessandro, D. Chen, T. Raeder, C. Perlich, M. Han Williams, and F. Provost. Scalable hands-free transfer learning for online advertising. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 1573–1582, 2014.

Google. Google's word2vec open source.

X. He, J. Pan, O. Jin, T. Xu, B. Liu, T. Xu, Y. Shi, A. Atallah, R. Herbrich, S. Bowers, and Joaquin Q. Candela. Practical lessons from predicting clicks on ads at facebook. In *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising*, ADKDD'14, pages 5:1–5:9, New York, NY, USA, 2014. ACM.

M. Mahdian and K. Tomak. Pay-per-action model for online advertising. *Internet and Network Economics, Lecture Notes in Computer Science*, 4858:549–557, 2007.

D. Marquardt and R. Snee. Ridge regression in practice. *The American Statistician*, 29: 3–20, 1975.

H. McMahan, G. Holt, D. Sculley, M. Young, D. Ebner, J. Grady, L. Nie, T. Phillips, E. Davydov, D. Golovin, S. Chikkerur, D. Liu, M. Wattenberg, A. Hrafnkelsson, T. Boulos, and J. Kubica. Ad click prediction: A view from the trenches. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 1222–1230, New York, NY, USA, 2013. ACM.

T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013a.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *Proceedings of Workshop at ICLR*, 2013b.

J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineer*, 22(10):1345–1359, 2010.

T. Park and G. Casella. The bayesian lasso. *Journal of the American Statistical Association*, 103:681–686, 2008.

C. Perlich, B. Dalessandro, T. Raeder, O. Stitelman, and F. Provost. Machine learning for targeted display advertising: transfer learning in action. *Machine Learning*, 95:103–127, 2014.

Rómer Rosales, Haibin Cheng, and Eren Manavoglu. Post-click conversion modeling and analysis for non-guaranteed delivery display advertising. In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, WSDM '12, pages 293–302, New York, NY, USA, 2012. ACM.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1994.

M. Welling and Y.W. Teh. Bayesian learning via stochastic gradient langevin dynamics. *Proceedings of the 28th International Conference on Machine Learning, Bellevue, WA, USA*, 2011.

M. West. Modelling with mixtures (with discussion). In *Bayesian Statistics 4*, pages 503–524. Oxford University Press, 1992. URL http://www.stat.duke.edu/~mw/MWextrapubs/West1992b.pdf.