Estimating Rates of Rare Events with Multiple Hierarchies through Scalable Log-linear Models

Deepak Agarwal
Yahoo! Research
Santa Clara, CA, USA
dagarwal@yahoo-inc.com

Rahul Agrawal, Rajiv Khanna,
Nagaraj Kota
Yahoo! Labs
Bengaluru,Karnataka,India
{arahul,krajiv,nagarajk}@yahoo-inc.com

ABSTRACT

We consider the problem of estimating rates of rare events for high dimensional, multivariate categorical data where several dimensions are hierarchical. Such problems are routine in several data mining applications including computational advertising, our main focus in this paper. We propose LMMH , a novel log-linear modeling method that scales to massive data applications with billions of training records and several million potential predictors in a map-reduce framework. Our method exploits correlations in aggregates observed at multiple resolutions when working with multiple hierarchies; stable estimates at coarser resolution provide informative prior information to improve estimates at finer resolutions. Other than prediction accuracy and scalability, our method has an inbuilt variable screening procedure based on a "spike and slab prior" that provides parsimony by removing non-informative predictors without hurting predictive accuracy. We perform large scale experiments on data from real computational advertising applications and illustrate our approach on datasets with several billion records and hundreds of millions of predictors. Extensive comparisons with other benchmark methods show significant improvements in prediction accuracy.

Categories and Subject Descriptors

H.1.1 [**Information Systems**]: Models and Principles

General Terms

Algorithms, Theory, Experimentation

Keywords

Computational Advertising, Display Advertising, Spike and Slab Prior, Gamma-Poisson, Spars Contingency Tables, Count Data

1. INTRODUCTION

Jointly estimating occurence rates of rare events for large number of attribute combinations (*cells*) is an important data mining problem that arises in several applications like computational advertising[5], disease mapping[9], ecology[10], adverse drug reaction[11], and many others. The main difficulty in such simultaneous rate estimation is the paucity of data and absence of events at fine resolutions, it is common to observe a large fraction of cells with zero or a few event occurences. Hence rate estimates obtained independently for each cell are often unreliable and noisy. In general, a "small sample size correction" obtained by *properly* pooling information across different data aggregates provide better estimates. In fact, aggregating data reduce variance due to larger sample size but introduces bias; disaggregation on the other hand reduce bias but incurs more variance. When data is hierarchical, *borrowing strength* from aggregates across multiple dimensions and multiple resolutions often lead to estimates with a good bias-variance tradeoff. How to perform such borrowing in an accurate and scalable fashion when working with high dimensional data is the problem we address in this paper. Specifically, we describe rate estimation methods that exploit cell correlations in data that is hierarchical along more than one dimensions. Such data are commonplace in many scenarios including computational advertising, our main motivating application in this paper.

Computational advertising is a new scientific sub-discipline that is at the foundation of building large scale automated systems to select advertisements (ads) in online advertising applications. An important goal in online advertising is to find the best match between a given *user* in a given *context* and a suitable *ad*. Different variations of the problem arise depending on the context considered. For instance, in sponsored search the context is a query issued by the user; in contextual and display advertising the context is a publisher page visited by the user and so on. The definition of what constitutes a "best match" is a complex one and involves maximizing value for users, publishers and advertisers. However, one key input that is often required to facilitate good matches are estimates of rare events like click rates and conversion rates. In fact, a significant fraction of online advertising is performance based whereby an advertiser pays if an user performing a web search or visiting a publisher page responds positively to the ad. The positive response is typically measured in terms of click-through rate (CTR) on ads. Revenue models based on conversion rates (CVR) where payment is made when users perform some positive action (e.g. buying a product) on the advertiser landing page are also becoming popular. In this paper we provide a novel log-linear model to estimate such rates in high dimensional and large scale computational advertising problems.

The rate estimation problems described above entail several challenges. First, success (click and conversion) rates are typically low, especially in display and contextual advertising. Second, although massive amounts of data is obtained from large scale advertising

KDD'10, July 25–28, 2010, Washington, DC, USA.

systems (several billion ads served), the dimensionality of the attribute space is large and typically consists of cells that can potentially run into several millions. Furthermore, data on new cells are frequently added to the system. In fact, to see this curse of dimensionality issue, note that an ad display is an interaction among elements in the tryad (user, publisher ,ad) and typically generates a response (no-click, click, conversion). However, the number of publishers, ads and users in the system is typically large. The data distribution among cells is also unbalanced since the best ad matches typically tend to be more concentrated. Hence, it is usual to see a small number of cells account for a large fraction of data (head) with the remaining sparsely distributed among a large number of cells (long tail). Thus, there is often extreme data sparseness at the finest resolution cells for which estimates are required to perform good ad matches.

The main assumption we make is that although each dimension (publisher, ad, user) is a high-dimensional categorical variable with large number of values, they are hierarchical and hence facilitate data pooling at multiple resolutions. For instance, publishers maybe arranged in a hierarchy based on URL prefix rollups, users maybe characterized by several attributes some of which are hierarchical (e.g. Geo) and there is a natural hierarchy for ads that aggregates into campaigns which in turn are aggregated into advertisers.

We provide estimation methods that can exploit correlations when considering cross-product of such multi-dimensional hierarchical categorical variables. In fact, our method borrows from estimates at coarser resolutions when there is sparseness at finer resolutions. For instance, if users from Manhattan have seen a Honda Accord ad on New York Times 10,000 times in the past and clicked 500 times, we do not use pooling and provide an estimate that is close to $500/10000 = 1/20$. Consider another scenario where San Jose Mercury News site saw only 100 visits from users in Sunnyvale California and obtained 5 clicks, an estimate of $1/20$ is perhaps not as reliable in this case. Here we may want to "borrow strength" from aggregate click rate estimate of all California visits to Mercury News. But how much should we borrow and from where should we borrow under different sample size scenarios? In the example above, is it best to borrow from user visits in California? Or is it better to borrow from user visits to all news sites in California? Or shall we use some other aggregates? The answer often depends on correlations among cells at finer resolutions that have common ancestor cells at coarser resolutions. This is a non-trivial problem with high-dimensional multivariate categorical data. We provide a solution to this problem through a novel statistical method which we shall call LMMH (Log-linear Model for Multiple Hierarchies) in the rest of the paper.

Other than estimation accuracy of rates, it is also desirable to have a parsimonious model (i.e. a model with small number of parameters). Models with large number of parameters have high memory requirements that makes cost-effective online ad selection difficult. Disk access for large models is an option but it has an adverse impact on throughput, often not acceptable in large scale systems. Our method achieves both competing objectives of accuracy and size. The idea employed is simple and based on *thresholding* - cells that borrow too much strength from ancestors are pruned and completely fallback on ancestors. This eliminates the need to store parameters for pruned cells and significantly reduces the model size. However, the thresholding operation is not applied as a post-processing step to model output but is in fact a built-in feature of the model itself. This provides a principled modeling framework to obtain models that are both accurate and parsimonious.

Finally, our model fitting procedure have to scale to massive amounts of data that are routine in computational advertising applications. Massive in this paper would refer to applications with terabytes of training data and millions of potential predictors. More explicity, the entire training data cannot fit in memory using commodity hardware, computing paradigms like map-reduce[7] provide an attractive way to scale computations in such scenarios. We provide a scalable and simple model fitting algorithm for LMMH that scales gracefully to massive data mining applications in a map-reduce framework. In fact, we demonstrate scalability by fitting models to datasets obtained from a real-world display advertising system that consists of several billions records with hundreds of millions of cells.

Our **contributions** are as follows. We propose LMMH, a novel statistical method to estimate rates of rare events with high dimensional, multivariate and hierarchical categorical data. LMMH improves prediction by exploiting correlations in aggregates and extends previous work for a single hierarchy. Besides accuracy, we also address the issue of parsimony to reduce memory requirements for online scoring by taking recourse to a novel variable screening procedure. Our screening procedure is part of the model and based on a "spike and slab" prior that ensures parsimony without hurting accuracy. We provide a scalable model fitting procedure through a sequential "one-at-a-time-update" iterated conditiona modes (ICM) model fitting algorithm in a map-reduce framework. We illustrate our method on large datasets obtained from a real-world computational advertising system.

The rest of the paper is organized as follows — We begin with a description of data underlying our method in section 2 with model description and fitting in section 3 Experiments are described in section 6 and we end with a brief discussion in section 7.

2. DATA

In this section, we describe the data characteristics underlying our LMMH approach. We begin with a description of event level data that is generated when there is a three way interaction between a user and ad on a publisher. This is followed by assumptions we make about our hierarchical attributes.

2.1 Event Data

Each record in our data represents information for a tryadic interaction which occurs when an ad is shown to a user in a context, we shall call this an *event*. For exposition, we consider display advertising where context is a publisher. The user interacts with the ad and may respond in several ways - do nothing, click on it and convert subsequently on the advertiser landing page, or click on it but not convert. Our goal is to predict response for future interactions accurately to facilitate selection of best matching ads. As described in section 1, this is challenging due to high-dimensionality and data sparseness. However, we assume curse of dimensionality could be mitigated to some extent since some of the categorical variables are hierarchical.

2.2 Hierarchies

For ease of exposition, we consider two hierarchical attributes. Using display advertising as our running example, we assume the attributes are publisher and advertiser hierarchies respectively. Each element in the publisher and advertiser hierarchies are directed paths of length m and n respectively and are denoted by $\boldsymbol{i} = i_1 \rightarrow i_2 \rightarrow \cdots \rightarrow i_m$ (publisher hierarchy) and $\boldsymbol{j} = j_1 \rightarrow j_2 \rightarrow \cdots \rightarrow j_m$ (advertiser hierarchy)respectively. Nodes with increasing suffix in a path represent finer resolution of data aggregation. For example, one of the dataset used in this paper estimate conversion rates with a publisher hierarchy where paths are of length 2 (i.e. $m = 2$),

i_1 represents the publisher type and i_2 provides the publisher id associated with publisher type i_1. For the advertiser hierarchy in this dataset, we have paths of length 4 (i.e. $n = 4$) where j_1 is the advertiser, j_2 is a converion-id instrumented by advertiser j_1 to track conversions on traffic routed to the landing page, j_3 is the campaign-id and j_4 is the ad-id. While for publisher hierarchy i_2 is strictly nested within i_1 (each publisher is of exactly of one type), this is not true for advertiser hierarchy. For instance, a single ad can participate in multiple campaigns and a single campaign can be part of multiple conversion-ids for a given advertiser. However, we assume that an ad associated with an event belongs to a unique path in the advertiser hierarchy; our methodology works for any hierarchical attributes with such directed acyclic (DAG)structure which is more general than strict trees. Also, for notational ease we assume paths of fixed lengths m and n for each event but our method easily generalizes to scenarios where events are associated with paths of varying lengths.

3. MODEL

In this section, we provide technical details of our LMMH approach. We describe our model for two hierarchies and then provide generalizations to multiple hierarchies.

The raw event data is aggregated for cross-product of paths $z = (\boldsymbol{i}, \boldsymbol{j})$ to obtain sufficient statistics (S_z, E_z) referred to as Successes and Tries respectively. For instance, in our running example the key z aggregates data from all displays of ad j_4 that belongs to advertiser j_1 on a publisher i_2 when it participates in campaign j_3 and conversion-id j_2. The definition of Success and Tries depends on the response prediction problem. In estimating conversion rate per click for example, S_z represents number of conversions obtained on key z out of E_z clicks. Denoting by λ_z the true rate parameter for key z, our goal is to estimate λ's for different keys z. The simple ratio estimator $\hat{\lambda_z} = S_z/E_z$ that obtains estimates independently for different keys z is reliable only for large sample sizes, this is not true for computational advertising applications where an overwhelming fraction of keys z have small sample size. Hence estimating the λ's by borrowing strength at different resolutions is an attractive way to reduce variance.

3.1 Baseline Probabilities through Covariates

The node ids in the hierarchical paths of our attributes often have additional meta-data associated with them. For instance, publishers and advertisments can be classified into content categories like sports, finance; it is possible to extract additional information from page content and so on. Models that simultaneously utilize both meta-data and event statistics for nodes have better accuracy for sparse data. For instance, we may have little data for a particular sports advertiser when displayed on a small sports publisher but large amounts of response data for sports ads when displayed on sports publisher. *Fusing* such information with node statistics may lead to improved performance. Use of such meta-data is also useful in cold-start scenarios where one has to predict response for a new ad on a new/old publisher. Such methods that combine meta-data with statistics at node-ids have been shown to work better in the context of other applications[2, 16]. Other than meta-data on node-ids, additional information like daypart, behavioral targeting attributes for users may also be available. We denote by $\boldsymbol{x}_k$ the covariates obtained through meta-data and additional non-hierarchical attributes for the k^{th} event; $\boldsymbol{i}_k$ and $\boldsymbol{j}_k$ denote the associated hierarchy paths. Denoting by p_k the true rate for event k with covariate $\boldsymbol{x}_k$ and paths $z_k = (\boldsymbol{i}_k, \boldsymbol{j}_k)$, we assume the following decomposition

$$p_k = b_k \lambda_{z_k} \tag{1}$$

where b_k is a probability estimate(baseline probability) that is computed from a baseline model which is a function of covariates $\boldsymbol{x}_k$ and λ_{z_k} is a cell-specific *correction* factor which is only a function of the hierarchical paths and does not depend on covariates. Covariate-based baseline models have been reported in the literature — while several of them are based on logistic regression techniques[15], there are exceptions that use other methods[8]. Since one has access to large amounts of data, it is possible to obtain reliable estimates of b_k with techniques like logistic regression. Given baseline probabilities at event level, sufficient statistics for key z now consist of (S_z, E_z), where E_z is now the expected number of successes under the baseline model instead of total number of tries. More concretely, if $\mathcal{F}_z$ denote the set of events corresponding to key z, E_z is given by

$$E_z = \sum_{k:k \in \mathcal{F}_z} b_k$$

3.2 Estimating cell corrections

We now describe our method for estimating corrections λ_z associated with keys z by exploiting correlations in multi-dimensional hierarchical aggregates. The main idea is to assume λ_z for each z can be written as a log-linear model with $m.n$ terms that are associated with all cross-products of nodes in paths $\boldsymbol{i}$ and $\boldsymbol{j}$. More specifically, we assume

$$\lambda_z = \prod_{s=1}^{m} \prod_{t=1}^{n} \phi_{i_s, j_t} \tag{2}$$

where ϕ_{i_s,j_t} is the state parameter associated with node pair (i_s, j_t). Denoting by $\boldsymbol{\phi}$ the state parameter vector associated with all node-pairs, we note that the dimension of $\boldsymbol{\phi}$ in our applications is extremely large; in one of our example datasets there are approximately 100M unknown state parameters.

As discussed in section 1, it is also important to obtain a *parsimonious* solution without sacrificing much accuracy. Here, parsimony implies a large fraction of estimated state parameters have a value that is exactly 1.0 and hence need not be stored. This is important since non-parsimonious solutions bloat up the model size and may require prohibitive amounts of memory. Finally, our model fitting method should scale to high-dimensional data that consists of billions of records and hundreds of millions of node pairs. We now describe our modeling techniques for estimating $\boldsymbol{\phi}$ that achieves all three objectives — accuracy, parsimony and scalability.

3.3 Estimating ϕ

Our model assumes $S_z | E_z, \boldsymbol{\phi} \sim \text{Poisson}(E_z \lambda_z)$, where S_zs are conditionally independent given $\boldsymbol{\phi}$ and λ_z is a log-linear function of $log(\phi)$s as described in equation 2. Since our goal is to estimate rates of rare events, Poisson assumption is reasonable and have been widely used in applications involving rare event estimation(see [11],[9] for examples). Thus, the log-likelihood of $\boldsymbol{\phi}$ under the Poisson model is given by

$$\begin{aligned} l(\boldsymbol{\phi}) &= \sum_z (-E_z \lambda_z + log(\lambda_z) S_z) + \text{constant} \\ \lambda_z &= \prod_{s=1}^{m} \prod_{t=1}^{n} \phi_{i_s, j_t} \end{aligned} \tag{3}$$

Problems with MLE — One can compute maximum likelihood estimates (MLE) of $\boldsymbol{\phi}$ by maximizing $l(\boldsymbol{\phi})$. In fact, state parame-

ters of node pairs at coarser resolutions are *shared* by larger number of keys z and are hence estimated with higher precision. The problem occurs with pairs at finer resolutions where MLE overfit the data. This happens since our data is high dimensional and rates are small, it is natural to observe large fraction of keys z with low Tries and zero Success S_z. However, the true rates in such cases are not zero but small, MLE nevertheless provide zero probability estimates. To see this more clearly, we consider a toy example with a two level hierarchy where level 1 has one node (root) and level 2 has two child nodes. Denote by (S_1, E_1, ϕ_1), $(S_{11}, E_{11}, \phi_{11})$ and $(S_{12}, E_{12}, \phi_{12})$ statistics and states for the root and two child nodes respectively, where $S_1 = S_{11} + S_{12}$ and $E_1 = E_{11} + E_{12}$. We shall use this toy example extensively to explain other concepts later on in this section. The log-likelihood of this three node hierarchy is given by

$$\begin{aligned} &- \phi_1(E_{11}\phi_{11} + E_{12}\phi_{12}) + \\ &S_1 log(\phi_1) + S_{11} log(\phi_{11}) + S_{12} log(\phi_{12}) \end{aligned}$$

If $S_{11} = 0$, it is easy to see that the maximizer of log-likelihood should occur at a point where $\phi_{11} = 0$; and hence $\lambda_{11} = 0$. This is so since the only term involving ϕ_{11} when $S_{11} = 0$ is $-\phi_1 E_{11}\phi_{11}$ which is maximized at $\phi_{11} = 0$. Such an estimator is problematic in our application due to data sparseness and rareness of success; it is more intuitive in such cases to exploit correlations that are induced due to hierarchical aggregations of data. For instance, in this case assume $S_{11} = 0, E_{11} = 5$ but $S_{12} = 10, E_{12} = 1000$. Since the two leaf nodes are siblings (e.g. two ads from the same advertiser), the rates are expected to be correlated and since node 12 has more data, one can intuitively improve the estimate of λ_{11} by borrowing strength from the estimate of λ_{12}. Next, we describe how to enhance our model to exploit such correlations and improve the MLE.

Incorporating hierarchical correlations — The problem of incorporating correlations for hierarchical data have been studied in the literature before where hierarchical correlations are incorporated through a tree-structured autoregressive model[18]. In our toy example, the autoregressive model will assume λ_{11} and λ_{12} equals the parent rate λ_1 in expectation but there is statistical variation that is given by some error distribution. The spread of this error distribution depends on the correlation among siblings; small spread imply higher correlation and in the extreme when there is no spread, sibling rates completely fallback on the parent rate. By writing $\lambda_{11} = \phi_{11}\phi_1$ and $\lambda_{12} = \phi_{12}\phi_1$ and assuming ϕ_{11}, ϕ_{12} equals 1 in expectation, we also obtain a model where the correlation is now induced by sharing parameter ϕ_1 with both children. The latter is sometimes referred to as non-centered parametrization while the former is called centered parametrization[14]. In fact, existing work assume an additive autoregressive structure on the logarithmic scale, i.e.,

$$\begin{aligned} \log(\lambda_{11}) &= \log(\lambda_1) + \epsilon_{11} \sim \mathcal{D}(0, \sigma) \\ \log(\lambda_{12}) &= \log(\lambda_1) + \epsilon_{12} \sim \mathcal{D}(0, \sigma) \\ \log(\lambda_1) &\sim \mathcal{D}(0, \sigma) \end{aligned}$$

where $\mathcal{D}(\mu, \sigma)$ is an error distribution with mean μ and scale σ. For most applications this is assumed to be Gaussian[18] but recent work also use double exponential priors[10]. The corresponding non-centered parametrization for this model would assume

$$\begin{aligned} \log(\lambda_{11}) &= \log(\phi_1) + \log(\phi_{11}) \\ \log(\lambda_{12}) &= \log(\phi_1) + \log(\phi_{12}) \\ \log(\phi)\text{'s} &\sim \mathcal{D}(0, \sigma) \end{aligned}$$

In fact, it is easy to show that for an additive tree-structured autoregressive model with Gaussian errors, the centered and non-centered parametrization are equivalent due to the reproducibility property of the Gaussian distribution; a similar result may not hold for other distributions like double exponential. However, both representations exploit hierarchical correlations; the centered does so by constraining parameters at finer resolutions to be close to ancestors while non-centered achieves the same effect by sharing ancestor parameters among descendants. In our toy example, the parameter ϕ_1 is shared with the two descendants. The choice of a representation (centered or non-centered) is generally dictated by computational considerations and the structure of the hierarchical data.

In this paper we adopt the non-centered parametrization primarily because it is easier to generalize to complex multi-resolution structures induced by multiple hierarchies. We also model the state parameters in our non-centered parametrization on the original scale instead of working on the logarithmic scale, i.e., ϕs are identically and independently (i.i.d) distributed as $\sim \mathcal{D}(1, a)$; the error distribution $\mathcal{D}$ in this case is centered at 1 with scale a. For computational scalability and to ensure parsimonious parameter estimates, we assume $\mathcal{D}(1, a)$ to be $\pi(\phi; a, P)$, a 2-component mixture of a Dirac and a Gamma distribution given by

$$\pi(\phi; a, P) = P1(\phi = 1) + (1 - P)\text{Gamma}(\phi; 1, 1/a)$$

i.e., with probability P the parameter ϕ is exactly 1 (i.e., state not important) and with probability $(1 - P)$ it is drawn from a Gamma distribution with mean 1 and variance $1/a$. Such priors are known as "spike and slab" priors in the literature[12] but their use for modeling hierarchical data have not been considered before. They encourage automatic variable selection in regression problems and their use for non-hierarchical count data was explored by [11] in a fraud detection application. We experiment with two versions in the paper - a) 1-component Gamma which assumes $P = 0$; this leads to dense solutions and b) 2-component Gamma which assumes $P = .5$; i.e. a-priori before seeing the data there is a 50% chance of a variable not being important (performance was not sensitive to the choice of P). For a fixed value of a, combining the prior on $\boldsymbol{\phi}$ with the likelihood gives the log-posterior of $\boldsymbol{\phi}$ as

$$l(\boldsymbol{\phi}) + \sum_{ij} log(\pi(\phi_{ij}; a, P)) \tag{4}$$

The state estimates $\tilde{\boldsymbol{\phi}}$ for a fixed a are now obtained by maximizing the log-posterior in equation 4.

Estimating the mode of $\boldsymbol{\phi}$ — One can optimize Equation 4 to obtain a mode of $\boldsymbol{\phi}$ by using standard sub-gradient descent methods but to ensure scalability, we instead work with a simple "one-at-a-time" sequential update procedure that is also known as Iterated conditional mode (ICM) algorithm in the literature[13]. The fitting algorithm is simple - we cycle through the state parameters for node pairs and update them one at a time, by computing the one dimensional mode of the conditional posterior of node state assuming others are fixed at their latest values. More specifically, indexing node pair suffixes ij from $1, \cdots, M$ without any loss of generality and denoting by $-k$ all nodes except the k^{th} one, we iteratively find the one dimensional modes of the conditional posterior $[\phi_k | \phi_{-k}, \text{Data}]$ until convergence, i.e., at the t^{th} iteration of our algorithm we update the state of k^{th} node to ϕ_k^t, the mode of the conditional posterior

$$[\phi_k | \phi_1^t, \cdots, \phi_{k-1}^t, \phi_{k+1}^{t-1}, \cdots, \phi_M^{t-1}, \text{Data}]$$

For our toy example for instance, at iteration t we compute modes

of $[\phi_1|\phi_{11}^{t-1}, \phi_{12}^{t-1}, \text{Data}]$, $[\phi_{11}|\phi_1^t, \phi_{12}^{t-1}, \text{Data}]$ and $[\phi_{12}|\phi_1^t, \phi_{11}^t, \text{Data}]$ respectively.

3.4 Conditional mode for a state

In this section, we show that the conditional mode of a state given others is given by a closed form expression. To simplify notations, we again appeal to our toy example which is sufficient to understand the derivation. Consider the conditional distribution of $[\phi_1|\phi_{11}, \phi_{12}, \text{Data}]$ which is proportional to

$$[S_{11}, S_{12}|E_{11}, E_{12}, \phi_{11}, \phi_{12}, \phi_1]\pi(\phi_1)$$

Since S_{11}, S_{12} are conditionally independent given the node states, it can be easily shown that the conditional distribution is proportional to Poisson$(S_1, E_1^*\phi_1)\pi(\phi_1)$ where $E_1^* = \phi_{11}E_{11}+\phi_{12}E_{12}$. In fact, E_1^* can be interpreted as expected Success after adjusting for the corrections of all nodes that appear along paths which includes the node being updated. Note that this is a simple model that involves combining a Poisson likelihood of a single observation with a Gamma mixture and admits a closed form posterior that we will derive in a moment. We note that this fact generalizes to all states in our model and the analytical conditional posterior for every node state can be obtained in our problem by combining a Poisson likelihood based on observed Success at that node with adjusted expected Success and the 2-component Gamma prior. Thus, it is enough to derive the mode of ϕ for the following model for sequential update algorithm—

$$\begin{aligned}[S|E^*, \phi] &\sim \text{Poisson}(E^*\phi) \\ [\phi] &\sim \pi(\phi; a, P)\end{aligned} \tag{5}$$

We first provide the expression for the posterior mode of the model in Equation 5 and then derive the formula in the next section. The tools used in the derivation would also help in understanding some results that are presented later. Before providing the expession for the mode, we begin with some preliminaries to introduce essential notations.

Prelim 1: A random variable ϕ is said to follow a Gamma distribution with mean μ and effective sample size a if and only if the density function is given as $g(\phi; a\mu, a) = \frac{a^{a\mu}}{\Gamma(a\mu)}e^{-a\phi}\phi^{a\mu-1}$. We note that $\text{Var}(\phi) = \sigma^2 = \mu/a$ and we shall use the notation $\phi \sim \text{Gamma}(\mu, \sigma^2 = \mu/a)$. If $a\mu > 1$, the mode of Gamma$(\mu, \sigma^2 = \mu/a)$ is given as $\tilde{\mu} = (a\mu - 1)/a$. In our context, $a\mu$ and a can be interpreted as psuedo number of Successes and Tries respectively.

Prelim 2: Here we point out the construction of a negative binomial distribution as scale mixture of Poisson. Let $S|\phi, E^* \sim Poisson(E^*\phi)$ and $\phi \sim \text{Gamma}(\mu, \mu/a)$. Then, the marginal distribution of S is a negative binomial with probability mass function

$$[S] = \int [S|\phi, E^*][\phi]d\phi$$

Some algebra yields

$$[S] = \frac{a^{a\mu}E^{*s}\Gamma(S + a\mu)}{(E^* + a)^{S+a\mu}\Gamma(a\mu)\Gamma(S + 1)}$$

We shall denote this by NB$(S; a, \mu, E^*)$. Note that $\mathcal{E}(S) = E^*\mu$ ($\mathcal{E}$ is the expectation operator) and $Var(S) = E^*\mu(1 + \frac{E^*}{a})$. For a Poisson, mean equals variance; thus a negative binomial distribution is more heavy tailed than Poisson. This is often referred to as overdispersion in the literature[6].

Prelim 3: The marginal distribution when $S|\phi, E^* \sim Poisson(E^*\phi)$ and $\phi \sim \pi(\phi; a, P) = P1(\phi = 1) + (1 - P)\text{Gamma}(\phi; 1, 1/a)$ is a mixture of Poisson and negative binomial, i.e.,

$$[S] = P\text{Poisson}(S; E^*) + (1 - P)\text{NB}(S; a, 1, E^*)$$

We are now ready to provide the expression for the mode of model in Equation 5. We state the result in the form of a theorem but first we begin with a simple Lemma

Lemma 1 *Assuming $a > 1$ and $P = 0$, the posterior mode $\tilde{\phi}_m$ for model in Equation 5 is given by*

$$\tilde{\phi}_m = (S + a - 1)/(E^* + a) \tag{6}$$

The proof follows from conjugacy of Gamma-Poisson model, it is easy to see from Bayes theorem that the posterior distribution of ϕ when $P = 0$ is Gamma$(\phi; \frac{S+a}{E^*+a}, (E^* + a))$. We now state our main theorem that gives us the mode for arbitrary value of P.

Theorem 1 *Assuming $a > 1$ and $P \in [0, 1]$, the posterior mode $\tilde{\phi}$ for model in Equation 5 is given by*

$$\left\{ \begin{array}{c} \tilde{\phi} = 1 \textit{ if} \\ Q > \log(g(\phi_m; S + a, E^* + a) - g(1; S + a, E^* + a)) \\ \tilde{\phi} = \phi_m \textit{ otherwise} \end{array} \right\}$$

where

$$Q = log\frac{\textit{Poisson}(S, E^*)}{\textit{NB}(S; 1, E^*, a)} + log(\frac{P}{1 - P})$$

Assuming $P = .5$ for simplicity of interpretation, Q is the log-likelihood ratio of data (S, E^*) to test if the data is generated from a Poisson distribution with $\phi = 1$ relative to a heavy tailed negative binomial counterpart. If the data is well supported by the Poisson distribution, the mode equals 1 and the variable ϕ is *pruned*. The proof of the theorem follows by computing the posterior distribution of ϕ which from Bayes theorem can be shown to be

$$[\phi|S, E^*] = q1(\phi = 1) + (1 - q)\text{Gamma}(\phi; \frac{S + a}{E^* + a}, (E^* + a))$$

where $Q = log(\frac{q}{1-q})$. To compute the mode of this distribution, note that the mode has to be either 1 or $\tilde{\phi}_m$, the mode of the 1-component Gamma$(\phi; \frac{S+a}{E^*+a}, E^* + a)$ depending on the value of the density at these points. Thus, the mode is 1 if density at 1 is higher than at $\tilde{\phi}_m$, i.e.,

$$\begin{aligned} &q + (1 - q)g(1; s + a, E^* + a) > (1 - q)g(\tilde{\phi}_m; s + a, E^* + a) \\ &\log(g(\tilde{\phi}_m; s + a, E^* + a) - g(1; s + a, E^* + a)) < Q \end{aligned}$$

and hence the theorem follows.

Correlations induced by LMMH — LMMH induces correlations among nodes sharing same ancestor(s). Although it is not easy to obtain expressions for such correlations analytically, we provide mathematical intuition on how the correlations get induced through our simple toy example using a 1-component Gamma prior (i.e. $P = 0$). First, note that the marginal density of Success in each sibling node (S_{11} and S_{12}) conditional on state of parent node ϕ_1 being known are independently distributed negative-binomials NB$(S_{11}; E_{11}\phi_1; a)$ and NB$(S_{12}; E_{12}\phi_1; a)$ respectively. However, due to shared parameter ϕ_1, the joint marginal density of S_{11} and S_{12} are no longer independent and given by

$$\int NB(S_{11}; E_{11}\phi_1; a)NB(S_{12}; E_{12}\phi_1; a)d\phi_1$$

expected value of product of two negative binomial probabilities with respect to the Gamma distribution on ϕ_1. This simple example clearly illustrates the fact that although our multi-hierarchy model is composed of conditionally independent sub-models, it induces correlations in the counts of our multi-dimensional data. In fact, the smoothing induced due to such correlation structure is an important aspect that provides good predictive performance.

3.5 Generalization to Multiple Hierarchies

For $K(> 2)$ hierarchies, there are several options. A natural appraoch is to model the cells in the entire K dimensional space; we did not find this to work well in practice due to enormous increase in sparseness when working with cross-product of more than 2 large hierarchies. Instead, we advocate the use of all 2-factor model; i.e., a log-linear model that is the product of node-pair terms from $\binom{K}{2}$ hierarchies. There is no conceptual or programming difficulty with this extension since we use an iterative fitting algorithm. However, we assume that K is small in our applications (e.g $3-5$) but each hierarchy is high dimensional. With large K, this issue requires further research.

4. MODEL FITTING

We describe our scalable model fitting procedure in a map-reduce framework based on the ICM algorithm described in section 3. We describe our algorithm for updating states with a cross-product of two hierarchies that are DAGs with K_1 and K_2 levels respectively. We note that the conditional posteriors of states for nodes at the $(m,n)^{th}$ level ($m = 1, \cdots, K_1$, $n = 1, \cdots, K_2$) are independent of each other and can be updated together in parallel. This forms the basis of our scalable map-reduce algorithm. Also, since the correction for a key $z = (\boldsymbol{i}, \boldsymbol{j})$ is the product of states for all node pairs, we *linearize* the 2-d cross product space and update the node pairs in the following order by hierarchy levels: $(1,1),(1,2),\ldots,(1,K_2),(2,1),\ldots,(K_1,K_2)$ which we index as $k = 1,\ldots,M$, where $M = K_1K_2$. In the mapper when processing nodes at k^{th} level, we join the correction from the corresponding parent node at level $(k-1)$ in the linearized hierarchy to get corrected expected success. In the reducer, we aggregate over these success, and corrected expected success, while discounting previous iteration's correction for the node pair in question, to compute the updated correction for the current node pair. The reducer task uses the multiple-outputs feature in hadoop 0.20.1 for speed, scalability and outputs both the data and corrections simultaneously. Also the whole logic is implemented nicely into a single map-reduce task which reduces hadoop task setup/intialization overhead.

Specifically, for each $k = (i_s, j_t)$ node pair in the conjunction of paths $z = (i,j)$, we compute the state variable ϕ_k^t and update the expected success E_z^*. Each map task gets a chunk of joined input records of conjunction of paths ($data$) with success S_z, and expected success E_z^*, and parent state variable ϕ_{k-1}^t. Mapper then outputs the key k and value $data$, S_z and $E_z^*\phi_{k-1}^t$. Reducer tasks join the input with previous iterations state variable ϕ_k^{t-1} and outputs the key k, value $data$ with S_z and $E_z^*\phi_{k-1}^t/\phi_k^{t-1}$. It also computes the updated ϕ_k^t with aggregated S_z, $E_z^*\phi_{k-1}^t/\phi_k^{t-1}$ and outputs key k, value ϕ_k^t.

For $K(> 2)$ hierarchies, we compute the state variables of node pairs from any 2 hierarchies at a time to get corrected expected success. We then take these updated success, and expected success to apply onto the next two hierarchies, So overall we will have $\binom{K}{2}$ invocation of the algorithm. Order of the hierarchies does not matter as we iterate till convergence. We provide psuedo code in Algorithm 1

Algorithm 1 Psuedocode for map-reduce implementation

Initialize the global constant a, the state variables $\phi_0^0 = 1$.
Iterate until convergence,
Iterate t over the conjunction of paths $z = (i,j)$ in the data,
Iterate over all node pairs (i_s, j_t), indexed by $k = 1,\ldots,M$. Note that $(k-1)$ is M from $(t-1)$'th iteration, when $k = 1$ and $t > 1$. For 1'st iteration with k=1, $(k-1)$ would be treated as record id and the corresponding parent node state variable as 1.

$$\begin{aligned} Map &: (k-1, data, S_z, E_z^*) \bowtie (k-1, \phi_{k-1}^t) \\ &\rightarrow (k, \{data, S_z, E_z^*\phi_{k-1}^t\}) \\ Reduce &: (k, \{data, S_z, E_z^*\phi_{k-1}^t\}) \bowtie (k, \phi_k^{t-1}) \\ &\rightarrow \left\{ \begin{array}{c} (k, \{data, S_z, E_z^*\phi_{k-1}^t/\phi_k^{t-1}\}) \\ (k, \phi_k^t) \end{array} \right\} \end{aligned}$$

where, ϕ_k^t is computed for key k using $\sum S_z, \sum E_z^*\phi_{k-1}^t/\phi_k^{t-1}$, using mode formula described in Theorem 1.

5. RELATED WORK

Other than work already mentioned in other sections, there is a rich literature in statistics on multi-level hierarchical model that is directly related to our work[3]. However, these methods have been applied to single hierarchies and for small problems and are generally referred to as nested random-effects model. There is no work in this literature that considers multiple hierarchies in a high dimensional scenario as we do in this paper. Reliable rate estimation by exploiting hierarchical correlations for large scale computational advertising applications was considered in our earlier [1] but only for single hierarchies. The method proposed only applies to tree data, it does not work with general directed acyclic graphs. Moreover, the computational efficiency discussed in that paper only works with Gaussian response which is not a satisfactory assumption when the goal is to estimate the absolute rates. In machine learning,[10] also considered such a problem for a single class problem when predicting species distribution by geographic location but also for single hierarchy. They considered model parsimony by assuming L_1 prior through a centered parametrization. However, their application was not large scale compared to the datasets we illustrate in this paper. We provide a scalable generalization to multiple hierarchies that exploits correlations and provides parsimonious model (through a spike and slab prior) in large scale applications. Recent work that performs fast and large scale regression with embarassingly large number of predictors on very large applications is also directly related to our work[17]. However, such methods fail to exploit the hierarchical correlations that are often present in data arising in computational advertising. We show that LMMH outperforms such methods significantly by exploiting the correlations in section 6.

6. EXPERIMENTS

In this section, we illustrate performance of LMMH through several datasets obtained from a real-world computational advertising application. None of the datasets are publicly available, we were not able to obtain benchmark dataset that was large and where the goal was to estimate rates of rare events with hierarchical and high-dimensional categorical variables. Instead of creating contrived examples from datasets available in existing repositories, we provide a thorough analysis on real-world scenarios by comparison with state-of-the-art and simple baseline methods on our datasets. We note that comparison with some state-of-the-art methods required additional map-reduce implementations on our part to en-

sure scalability. Every attempt would be made to release some of the datasets used at a later date.

6.1 Display Advertising

We provide a brief overview and motivation for the display advertising response prediction problems that are illustrated later with real-world datasets.

Background—Online display advertising is a multi-billion dollar industry where advertisers display ads on publisher pages (e.g. Nordstrom, Nike, Coke). Unlike performance based advertising in sponsored search and contextual matching, advertisers in display advertising may design ad campaigns with different product goals in mind. One important objective is building brand awareness for promoting future sales, possibly targeted at a user segment. This is similar in spirit to advertising on television and popular magazines. Advertisers with this objective in mind generally opt for the Cost-per-Milli (CPM) model, whereby ad opportunities (user visits on publisher pages) are priced in bundles of 1000 and are paid for by the advertiser irrespective of user actions. On the other extreme, advertisers with products that have rare repeat sales (e.g. auto, refrigerator, insurance) may care more about immediate than future sales, the goal however may still be to target a certain user segment. For instance, an automobile manufacturer may target users who are interested in baseball for selling a new sports car model. Other advertisers maybe somewhere in between and care both about future and immediate sales; a major telecom company may want to build brand awareness but still promote immediate sales of its new calling plan.

Guaranteed and Non-guaranteed display advertising—Given the diverse nature of advertiser objectives, it is no surprise display advertising is sold under different revenue models and have given rise to a complex ecosystem of buyers (advertisers), sellers (publishers) and intermediaries (ad-networks). We provide a brief overview to motivate the data mining problem addressed in this paper. At a high level, there are two broad ways of delivering display advertising - guaranteed and non-guaranteed. In guaranteed delivery, advertisers reserve a fixed number of user visits targeted at a user group ahead of time on publisher's pages who guarantee these visits at a certain price. For instance, Johnson & Johnson may wish to target 100 million visits by females on Yahoo! astrology last week of January 2010. The publisher has to guarantee the delivery of visits, advertisers typically pay higher CPM to procure such a guarantee. Non-guaranteed delivery on the other hand does not provide such a guarantee on visit volumes. It follows the "pay as you go" strategy where each ad opportunity is sold through a real-time auction. The auction is facilitated by commercial intermediaries called ad-networks who connect advertisers to publishers (e.g. Ad.com, Value Click etc) and share a certain percentage of revenue that accrue from transactions. Beyond ad-networks, the "exchange" provides a platform for buyers and sellers to transact across network boundaries (e.g. RightMedia exchange). This is sometimes also referred to as "network-of-networks" model. In this section, we analyze a subset of data obtained from the RightMedia exchange. We emphasize that the data analyzed in this paper is in no way representative of data obtained from our entire system, it is a subset obtained from a certain set of geographic locations for the purposes of illustration in this paper.

Response Prediction Problem: normalization across pricing types— Advertisers participate in an auction using multiple *pricing types* like CPM, Cost-per-click(CPC), Cost-per-action (CPA) on the exchange, hence it raises a fundamental question of how to select a winner. The solution we currently employ is to *normalize* across pricing types and create a single denomination - expected CPM (eCPM). Thus, for CPC and CPA, eCPM = Pr(click or conversion)$*$Cost, where Cost is a function of advertiser bid discounted by revenue shared with intermediaries. Thus, predicting rates of rare response (clicks and conversions) is a fundamental problem to successfully conduct such auctions in the exchange using this strategy. In fact, errors in estimating rates may lead to arbitrage and provide unfair advantage to some pricing type. Underestimation on the other hand for a pricing type (e.g. CPA) would make it an unattractive mode of participation and the exchange may lose a certain set of advertisers. Thus, accurate estimation of click and conversion rates is an important problem in non-guaranteed display advertising on Right Media exchange.

6.2 Datasets from Right Media

We created three large datasets to conduct our experiments — (1) Post-View conversions (**PVC**) (2) Post-Click conversions (**PCC**) and (3) Click data (**CLICK**). We provide a brief description of event level data for all three types and then provide details of the models we fitted to these datasets.

PVC — Each event in this case consists of a binary response where success happens if a post-view conversion takes place. Advertisers participating as post-view conversion 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. If the same user then ends up on the advertiser site (through some other path) and converts (e.g. buys a product) according to the pixel definition, a conversion is registered. The conversion rates for PVC are extremely low (not revealed due to reasons of confidentiality). The PVC data we consider had approximately $7B$ events in training and roughly $240M$ in test. Other than age and gender for users, we have information on *sizeid* for each ad. We also have recency and frequency information on when ads were displayed to each user that are categorized into several bins. Our data consists of two hierarchical attributes, namely, publisher hierarchy and an advertiser hierarchy. The former has two levels (publisher type $\rightarrow$ publisher id) while the latter has four levels (advertiser $\rightarrow$ conversion-id $\rightarrow$ campaign $\rightarrow$ ad-id). We note that the advertiser hierarchy is not a strict tree but a DAG, a single ad can participate in multiple campaigns for instance. Conversion-id is the pixel id instrumented by advertiser to track conversions on their landing page.

CLICK—Success in this case occurs if a user clicks on an ad, all other variables in this dataset are same as **PVC** except for the advertiser hierarchy that consists of 2 levels (advertiser $\rightarrow$ adid). The number of events in this data is much larger, training set $90B$ while test set $3B$.

PCC— Here, an event is generated when a user clicks on an ad; success occurs when the user converts after the click-through. This is distinct from **PVC** since here the conversion has to occur subsequent to click-through on the ad and not by following any other path in a stipulated time period. Recency and frequency are of course absent in this data since they are measurements associated with ad displays and not ad clicks. Other variables are all same as **PVC**. The total number of events in this data was approximately $.5B$ in training and $20M$ in test. For the purposes of illustration in this paper, we note that **PCC** and **CLICK** data are not aligned and collected from different subsets, hence no valid inference could be made by combining **PCC** and **CLICK** event information.

6.3 Variations of LMMH

We now describe the LMMH variations that we ran on the three datasets. The **baseline** model for all datasets include covariates based on user age, user gender, publisher type, recency, frequency and sizeid. Appropriate baseline models were fitted after testing several variations using generalized linear mixed effects models (GLMM) in R[4]; the models we selected for each dataset are the ones with minimums AIC. We note that the generalized linear mixed model routines are computationally expensive, hence we performed the operations using a map-reduce approximation. In particular, we randomly partition the datasets, fit separate GLMM to each and combine the results by using a weighted average with weights being inversely proportional to the estimated variance. In practice, one can run other methods like logistic regression with L_2 regularization, we found the GLMM to provide better results for small models we fitted to these datasets. We shall refer to these as **GLMMB** when reporting results.

For the hierarchical correction models, we computed cell estimates for cross-products of following hierarchies — a) For **PCC** we used publisher $\times$ advertiser hierarchies. b) For **CLICK** we used (publisher-type $\rightarrow$ (recency, frequency) $\rightarrow$ publisher-id) $\times$ (advertiser $\rightarrow$ adid). We included (recency,frequency) bins in the hierarchy since click rates are known to vary by degree of previous exposure. c) For **PVC**, we used the same model as CLICK except for a four level advertiser hierarchy. For all models, we ran 1) 1-component Gamma (i.e. we choose $P = 0$) which provide estimates that are not exactly 1 and 2) 2-component Gamma (i.e. we choose $P = .5$) for which some ϕ are estimated as 1 and hence could be removed when storing the model in ad-servers for online scoring. For all models (including other variations we describe later), tuning parameters (e.g. a) are selected through cross-validation. For LMMH, a has an intuitive interpretation as psuedo number of successes, we have found that it is enough to consider values in the range of $2 - 10$ for cross-validation. We shall refer to the two variations with $P = 0, .5$ as **LMMH-1C** and **LMMH-2C** respectively when reporting results. Other than these, we also ran two models that 3)Only considers publisher-id $\times$ adid corrections and 4)Only considers publisher-id $\times$ advertiser. We do this to show the benefits of using the entire hierarchies for the purposes of estimation. These will be referred to as **FINE** and **COARSE** respectively. Next, we describe the methods that were used for comparison with LMMH.

Variations of logistic regression— We tested three different variations of logistic regression that differ in the number of features included in the model. We shall refer to them as **Log I**,**Log II** and **Log III** respectively. All logistic regressions were fitted in a map-reduce framework using conjugate gradient method (CG) along with L_2 regularization on the coefficients to ensure stable model fitting. The regularization parameter was selected through cross-validation and the maximum number of CG iterations was set to 50. The three variations only differ in the features, the fitting procedure was the same.

- **LogI**— For the three datasets (**PVC**,**PCC** and **CLICK**), this includes the main effects of all variables we have in our dataset. Thus for **CLICK**,

  $$\text{log-odds}(rate) = \text{pub-type} + \text{pub-id} + \text{age} + \text{gender} + \text{adv-id} + \text{ad-id} + \text{recency} + \text{frequency} + \text{sizeid}$$

  For **PVC**, we augmented the equation above with conv-id + campaign-id; for **PCC** the equation was same as **PVC** but did not include recency and frequency. The total number of features are 325307, 28380 and 206291 for PCC, PVC and CLICK respectively

- **LogII**—In this version we augmented the features used in **LogI** by adding paths of lengths > 1 on both the publisher and advertiser hierarchies. This still does not include any cross-product terms between publisher and advertiser hierarchies. The total number of additional features that got added are 708925, 61082, 202890 for PCC, PVC and CLICK respectively.

- **LogIII**—In this variation we added conjunctions of publisher and advertiser hierarchy features. However, adding all such conjunctions explodes the feature space and leads to scalability problems with logistic regression code. We employ a *hashing* trick that have recently been proposed in the literature for massively large scale regression problems[17]. In particular, we take the feature ids of all conjunctions and hash them into a reasonable number of bins. After experimenting with a few bin sizes, we decided to use $400K$ hashes for all datasets, both for reasons of scalability and ensuring accuracy, beyond this the predictive accuracy did not improve much.

6.4 Metrics

Since obtaining the absolute estimates are important in our display advertising application, we report on average test-loglikelihood under Bernoulli model as our prediction accuracy measure. Specifically, for a model the average test log-likelihood *avgLL* is

$$\frac{\sum_k (Succ_k * log(\hat{p}_k) + (Tries_k - Succ_k) * log(1 - \hat{p}_k))}{\sum_k Tries_k}$$

Instead of reporting the absolute log-likelihood numbers that can provide information on absolute value of probabilities, we report percent improvement is log-likelihood of a method relative to our covariate-only based baseline **GLMMB**, i.e.,

$$\frac{avgLL(\textbf{Model}) - avgLL(\textbf{GLMMB})}{|avgLL(\textbf{GLMMB})|}$$

Further, we split test data into 20 equal parts and report the distribution of log-likelihood lifts for each method to provide a measure of statistical variation in test set metrics.

6.5 Results

We first begin by providing statistics on the number of cells in each of our datasets. The total number of state parameters that were estimated by our LMMH were 81595746($\approx 81M$), 6039376($\approx 6M$) and 16517629($\approx 16.5M$) for CLICK, PVC and PCC respectively. Our best 2-component models for these datasets based on spike and slab prior had 4429106 ($\approx 4.4M$), 35694 ($\approx 35K$) and 148748($\approx 150K$) estimates that were different than 1 and hence needs to be stored. The 1-component gamma did not provide solutions that were exactly 1, however a large number of estimates were close to 1 but variable removal would now require taking recourse to post-processing procedures which distorts the canonical model results. Our 2-component model provides extremely parsimonious models that are lightweight and could be easily stored in memory to faciliate cost-effective online ad selection.

We also note that our modeling approach coupled with computation in a map-reduce framework is extremely scalable; we are able to process billions of records with hundreds of millions of cells in a few hours. For CLICK data set, we took 135 minutes with 50 reducers, for PVC 123 minutes with 25 reducers and PCC 109 minutes with 20 reducers for learning the models. In contrast to this,

the **LogI**, **LogII** and **LogIII** took 4, 6 and 7 hours for CLICK each employing 80 reducers, 3, 4.5, 5 hours for PVC with 40 reducers and 4.5, 8 and 9 hours for PCC with 80 reducers. Next, we examine the predictive accuracy of our models as given in Figure 1 for three datasets for different variations of our approach (includes simple baselines) and three different variations of logistic regression described before.

First, we note that all algorithms tested show lift relative to our covariate only baseline GLMMB, which clearly shows that using information at nodes is essential for good performance. Our two baseline variants {COARSE, FINE} that use only partial hierarchical information are significantly worse than others. Surprisingly, COARSE is better than FINE in PCC only; this is probably since PCC is sparse both in terms of Success and Tries unlike PVC and CLICK which are sparse in terms of Successes but have a lot more Tries. All three variants of logistic regression are worse than our LMMH variants; increasing the number of features helps logistic regression except in PVC which is too sparse and starts over-fitting. Both LMMH variants (1-C and 2-C) have similar performance and are significantly better than all other methods. This clearly shows that incorporating hierarchical correlations through our approach helps improve accuracy compared to other log-linear models like logistic regression that does not incorporate such information. The fact that both COARSE and FINE baselines are significantly worse also shows that smoothing alone is not enough to achieve good performance, it is imperative to combine smoothing with hierarchical information at multiple resolutions.

Although not visible on the plots, a two-sample test conducted on the 20 partition statistics revealed 2-C is slightly better than 1-C but the difference although statistically significant (we will always find statistical significance with such massive data) is not practically significant. However, as seen before LMMH 2-C can get comparable accuracy along with model parsimony induced through the spike and slab 2-component Gamma prior.

7. DISCUSSION

We proposed a new log-linear model LMMH for estimating rare rates in high dimensional, multivariate categorical data consisting of several hierarchies. Our method provides accurate predictions by exploiting hierarchical correlations, parsimony by using a spike and slab prior and is scalable to extremely large applications with billions of records and hundreds of millions of predictors in a map-reduce framework.

Several aspects of our problem needs further research. We were motivated by computational advertising applications and assumed a small number of large hierarchies; in applications where this is not the case the issue of appropriately selecting the relevant cross-product of hierarchies to consider is an open issue. Another important issue is incremental learning of our LMMH model in an online fashion. Variance computation for fast explore/exploit through our multi-hierarchy model is also an interesting direction.

8. ACKNOWLEDGEMENTS

We thank Ozgur Cetin for kindly providing us with the map-reduce logistic regression code. We thank Krishna Prasad Chitrapura and Sachin Garg for helpful discussions.

9. REFERENCES

[1] D. Agarwal, A. Z. Broder, D. Chakrabarti, D. Diklic, V. Josifovski, and M. Sayyadian. Estimating rates of rare events at multiple resolutions. In *KDD '07*, pages 16–25, 2007.

[2] D. Agarwal and B.-C. Chen. Regression-based latent factor models. In *KDD '09*, pages 19–28, 2009.

[3] A.Gelman and J.Hill. *Data Analysis using Regression/Multi-level Hierarchical Models*. Cambridge University Press, 2007.

[4] D. Bates and D. Sarkar. *lme4: Linear mixed-effects models using S4 classes*, 2007.

[5] A. Broder. Computational advertising. In *SODA '08: Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 992–992, 2008.

[6] A. C. Cameron and P. K. Trivedi. *Regression Analysis of Count Data*. Cambridge University Press, 1998.

[7] J. Dean and S. Ghemawat. Mapreduce: a flexible data processing tool. *Commun. CACM*, 53(1):72–77, 2010.

[8] K. Dembczynski, W.Kotlowski, and D.Weiss. Predicting ads' click-through rate with decision rules. In *WWW '08*, 2008.

[9] D.G.Clayton and J.Kaldor. Empirical bayes estimates of age-standardized relative risks for use in disease mapping. *Biometrics*, 43:671–681, 1987.

[10] M. Dudik, D. M. Blei, and R. E. Schapire. Hierarchical maximum entropy density estimation. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, pages 249–256, 2007.

[11] W. DuMouchel and D. Pregibon. Empirical bayes screening for multi-item associations. In *KDD '01*, pages 67–76, 2001.

[12] H.Ishwaran and J. Rao. Spike and slab variable selection: Frequentist and bayesian strategies. *Annals of Statistics*, 33(2):730–773, 2005.

[13] J.Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society, Series B*, 48:259–302, 1986.

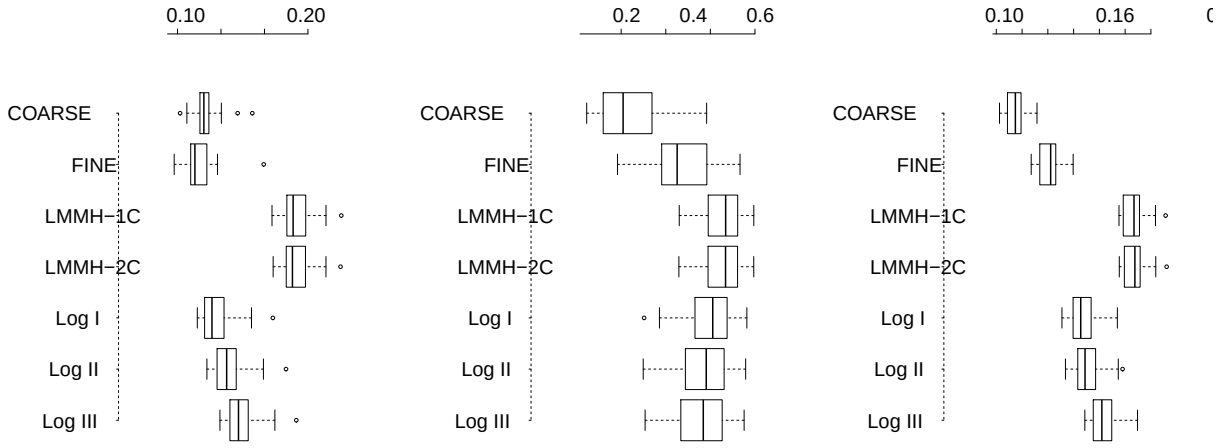
[14] O.Papaspiliopoulos, G. O. Roberts, and M. Skold. A general framework for the parametrization of hierarchical models. *Statistical Science*, 22(1):59–73, 2007.

[15] M. Richardson, E. Dominowska, and R. Ragno. Predicting clicks: estimating the click-through rate for new ads. In *WWW '07*, 2007.

[16] D. H. Stern, R. Herbrich, and T. Graepel. Matchbox: large scale online bayesian recommendations. In *WWW '09*, pages 111–120, 2009.

[17] K. Weinberger, A. Dasgupta, J. Langford, A. Smola, and J. Attenberg. Feature hashing for large scale multitask learning. In *ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1113–1120, 2009.

[18] L. Zhang and D. Agarwal. Fast computation of posterior mode in multi-level hierarchical models. In *NIPS*, pages 1913–1920, 2008.



(a) PCC (b) PVC (c) CLICK

Figure 1: ***Model performance based on lift in log-likelihood relative to*** **GLMMB.**