Estimating Rates of Rare Events with Multiple Hierarchies through Scalable log-linear models

Deepak Agarwal* Y! Research, Santa Clara, USA

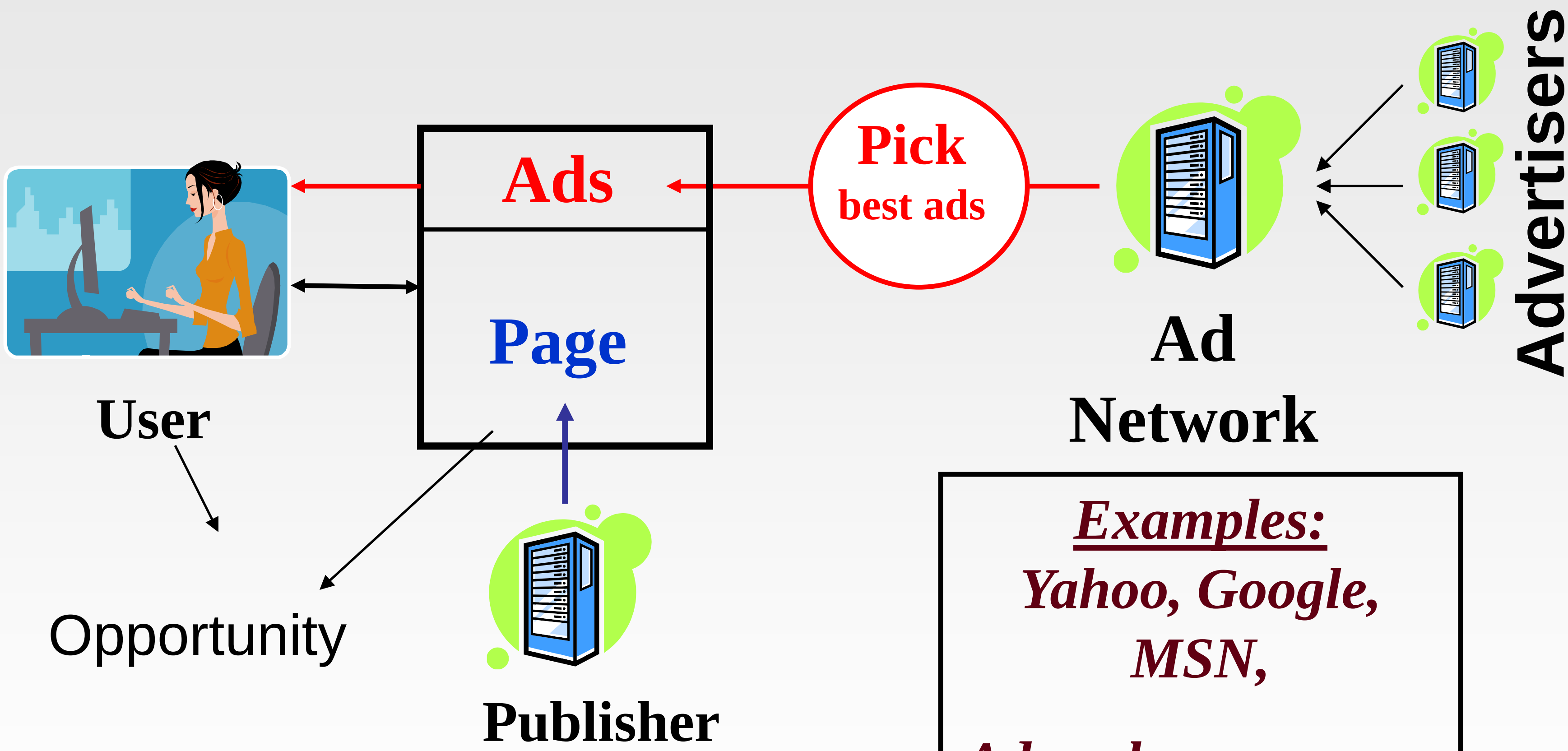
Rahul Agrawal, Nagaraj Kota and Rajiv Khanna
Y! Labs, Bangalore, India

KDD 2010, Washington D.C, 26th July, 2010

Agenda

- Motivating Example --- Computational Advertising
  - Display advertising in ad exchange
- Problem Definition ---- Predicting response rates of rare events by exploiting multiple hierarchies
- Log-linear models for multiple hierarchies (LMMH)
  --- Our multi-resolution model
- Scalable model fitting in a map-reduce framework
- Experiments --- Data from Right Media Ad Exchange
- Summary

Computational Advertising: Matching ads to opportunities

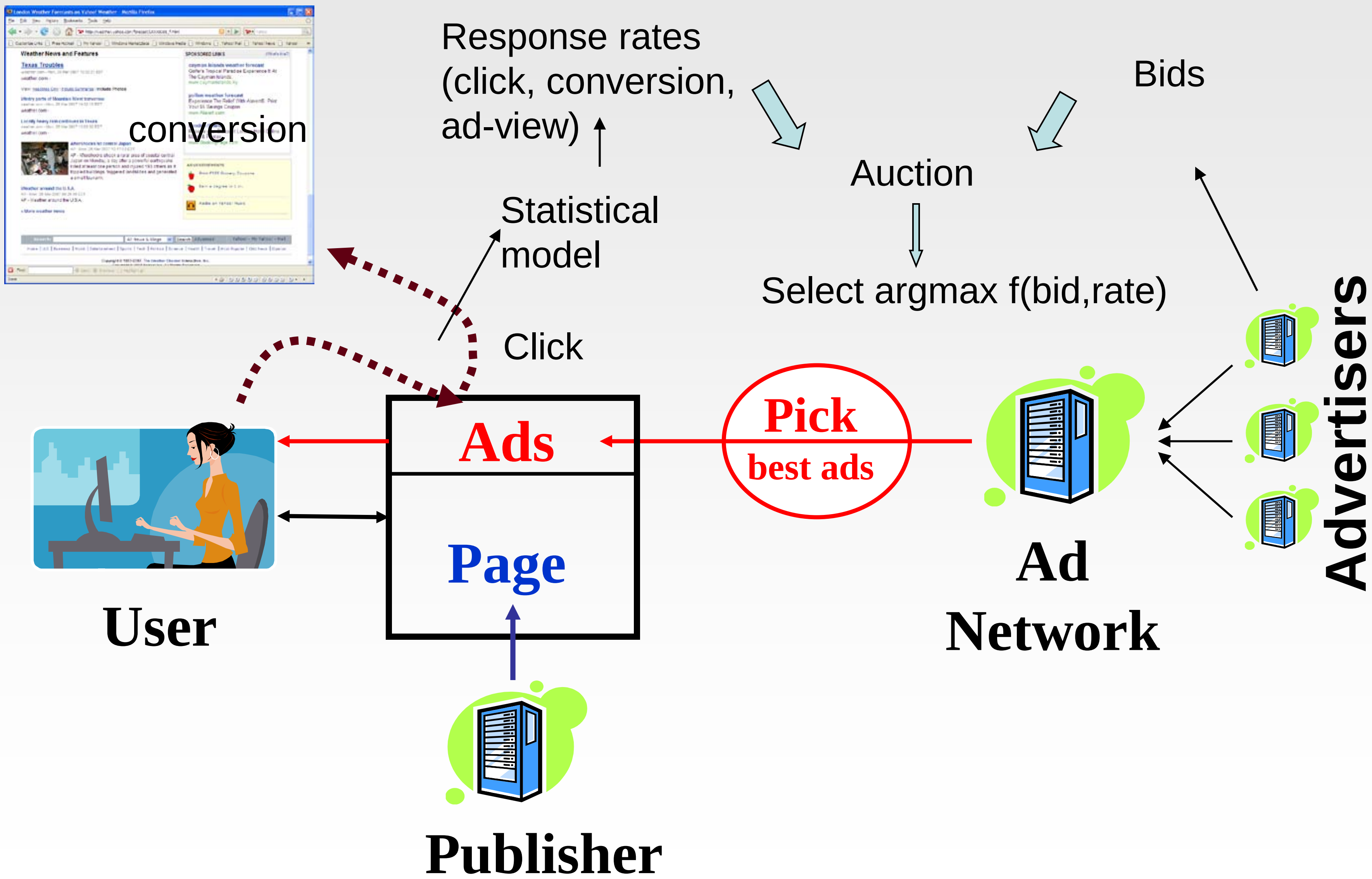


Examples:

Yahoo, Google, MSN,

***Ad exchanges** (network of "networks")* …

How to Select “Best” ads



Estimating response rates --- Challenges

- f(bid, rate) ---- rate is unknown, needs to be estimated
- Goal: maximize revenue, advertiser ROI
- Explore/exploit problem
  – Exploit based on rates that are high and have been learnt precisely, explore what looks "potentially good" by taking risks (quantified by variance estimates)
- Auction conducted based on some f*(bids, est-rates, est-var)
  – E.g. bid x (est-rate + 2 est-sd)
- This paper
  – Focus on a method to estimate rates by exploiting hierarchies
    - Reduces variance, faster convergence to best ads

Our data --- Ad- exchange (RightMedia)

- Advertisers participate through different pricing types
  - CPM (pay by ad-view)
  - CPC (pay per click)
  - CPA (pay per conversion)

- To run auction, normalize across pricing types
  - Compute eCPM (expected CPM)
    - Click-based ---- eCPM = click-rate*CPC
    - Conversion-based ---- eCPM = conv-rate*CPA
  - Require “absolute” response rate estimates

Data (2)

- Two kinds of conversion rates
  - Post-Click --- conv-rate = click-rate*conv/click
  - Post-View --- conv-rate = conv/ad-view

- Three response rate models
  - Click-rate (CLICK), conv/click (PCC),
  - post-view conv/view (PVC)

Notations: Ignoring user for simplicity

- Opportunity: (i, x_p)
  - publisher covariates (x_p), publisher-id i
- Ad (j, x_a)
  - Ad attributes(x_a), **ad-id** j
- Response
  - nSuccesses --- S_{ij}
  - nTries --- N_{ij}
- Goal is to estimate response rates with "cells" in a high dimensional, sparse contingency table

Challenges

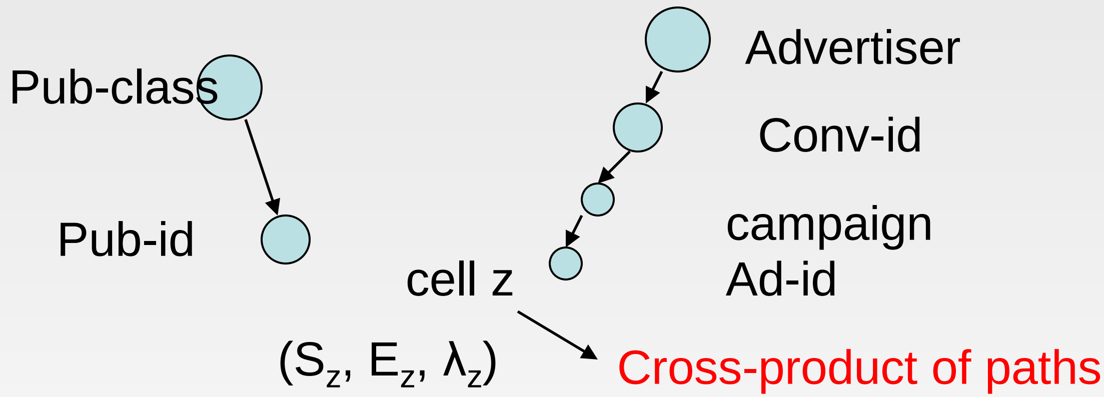
- Data sparsity
  - Response rates extremely rare
  - Number of cells too large, large fractions have 0 nSucc
    - High dimensional categorical variables
    - E.g. In CLICK data, 100M cells
  - Imbalanced sample size
    - nTries in cells have huge variation
  - Smoothing to perform small sample corrections important
- How do we perform such corrections in a scalable way?

Solution: high level idea

- Data aggregated hierarchically along dimensions (OLAP style)
- Exploit correlations induced by aggregates at different resolutions to improve estimates at fine resolutions
- Shrinkage estimation
  - If cell has enough sample size, use MLE o.w. fallback on estimates along lineage path
- Another interpretation
  - Estimates at cell weighted average of cells along lineage paths
  - Weights based on sample size and correlations

Hierarchical structure

- Assuming two hierarchies (Publisher and advertiser)



- Collaborative filtering perspective
  - Incomplete matrix but a DAG in each dimension
  - Estimating rates of rare events
    - Different from ratings, want to fallback on cell-specific estimators when sample size is large

Model

- For the k^{th} record $$p_k = b_k \lambda_{z_k}$$

baseline

Cell corrections in Table

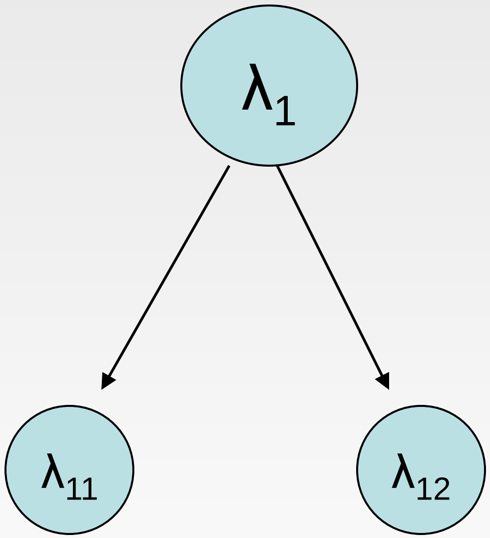
- Baseline model: based on covariates (low variance estimates)
- Tries now replaced by expected success

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

- Modeling assumption – $[S_z \mid E_z, \lambda_z] \sim \text{Poisson}(E_z \lambda_z)$
- Naïve estimator --- $$\hat{\lambda}_z = S_z / E_z$$
  - Doesn't work, too many zeroes with small sample size
  - Smoothing required

Lets look at simple single hierarchy example

- Proximity to parent

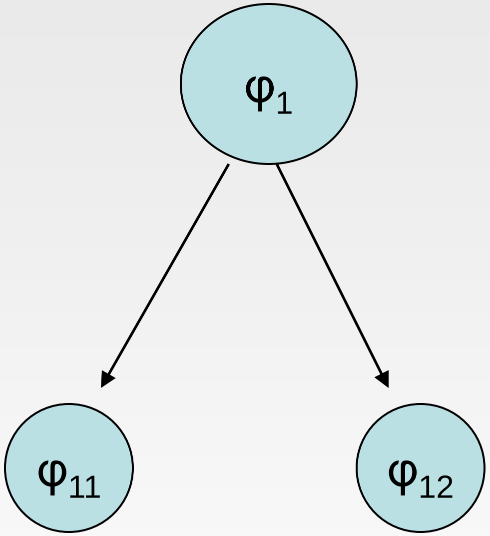


Centered parametrization

$\lambda_{11} \sim \pi(\lambda_1, \sigma)$

$\lambda_{12} \sim \pi(\lambda_1, \sigma)$

Sharing parameters



Non-centered parametrization

$\lambda_{11} = \varphi_1 \varphi_{11}$

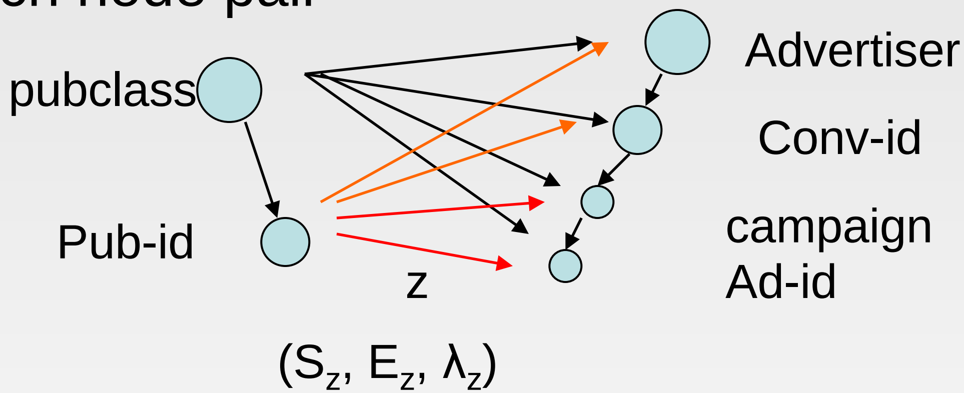
$\lambda_{12} = \varphi_1 \varphi_{12}$

$\varphi_1, \varphi_{11}, \varphi_{12} \sim \pi(1, \sigma)$

Model for 2 hierarchies

- Product of states for each node pair

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



- Spike and Slab prior

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

  - Known to encourage parsimonious solutions
    - Several cell states have no corrections
  - Not used before for multi-hierarchy models, only in regression
  - We choose P = .5 (and choose “a” by cross-validation)
    - a – psuedo number of successes

Optimization problem

- Find a solution that optimizes

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

- Not convex, non-differentiable (sub-gradient methods)
- For scalability, we use "sequential-one-at-a-time" update

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

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

Conditional mode – closed form

- Reduces to computing the mode of the following

$$[S|E^*, \phi] \sim \text{Poisson}(E^* \phi)$$
$$[\phi] \sim \pi(\phi; a, P)$$

- E* = Adjusted eSucc aggregating statistics on all paths that include the node being updated

- In the toy example for instance,

$$\text{Poisson}(S_1, E_1^* \phi_1)\pi(\phi_1) \text{ where } E_1^* = \phi_{11} E_{11} + \phi_{12} E_{12}$$

Conditional model --- closed form

- Threshold estimator : conducts hypothesis test

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

$$\begin{cases} \tilde{\phi} = 1 \; if \, Q - \log(g(\phi_m; S + a, E^* + a) - g(1; S + a, E^* + a)) \\ \qquad = \phi_m \; otherwise \end{cases}$$

where

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

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

Scalable Map-reduce implementation

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.

$$Map : (k-1, data, S_z, E_z^*) \bowtie (k-1, \phi_{k-1}^t) \rightarrow (k, \{data, S_z, E_z^* \phi_{k-1}^t\})$$

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

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.

Multiple (K) hierarchies

- Product of ${}^{K}C_{2}$ pair wise hierarchies
- Primarily done to deal with data sparseness
- Ongoing research
  - Find small subset of 3-way, 4-way combinations that are important through multiple testing procedures
  - Main idea is to adjust for multiple tests by "shrinking" obs/expected from all 2-factor models to detect significant higher order interactions

Datasets : RMX

- CLICK [~90B training events]
- PCC (~.5B training events)
  - Conversion only through click
- PVC – Post-View conversions (~7B events)
  - Cookie gets augmented with pixel and triggers success
- Features
  - Age, gender, sizeid, pubclass, recency, frequency
  - 2 hierarchies (publisher and advertiser)
- Two baselines
  - Pubid x adid [FINE] (no hierarchical information)
  - Pubid x advertiser [COARSE] (collapse cells)

Other methods: Variations of logistic regression

- Runs on map-reduce

- **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**,

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

  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

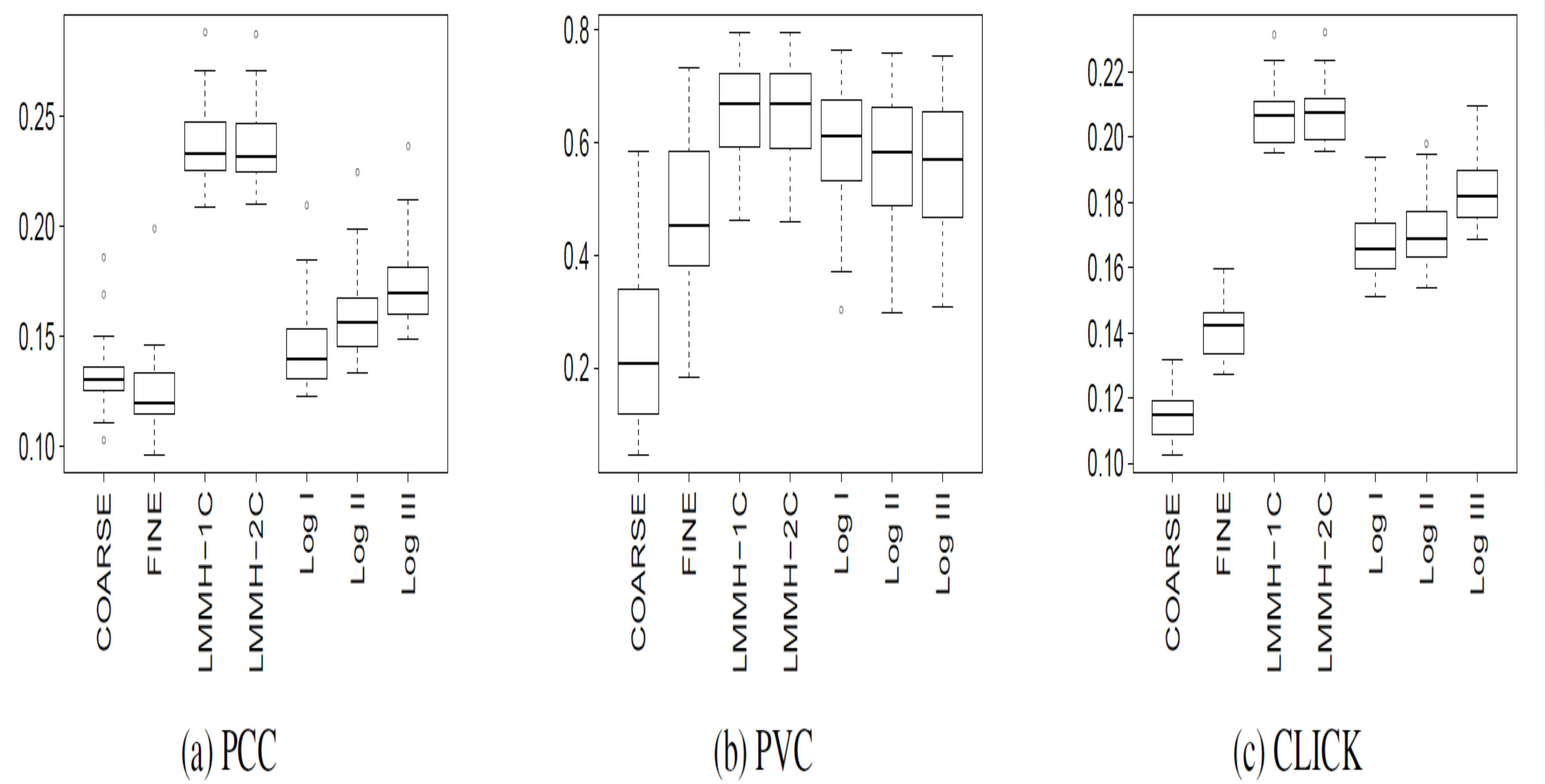
Logistic regression variations

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

LogII + conjuctions of features but with hashing. Included 400K hash bins

Accuracy: Average test log-likelihood



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

LMMH variations

- 2-component spike and slab prior
- 1-component prior (spike removed, only the slab)
  - Non parsimonious solutions
- Parsimony

data	#cells	#retained
PCC	~81M	4.4M
PVC	~6M	35K
CLICK	~16.5M	150K

Some rough computation time

- CLICK : 135 mins, 50 reducers
- PVC : 123 minutes, 25 reducers
- PCC: 109 minutes, 20 reducers

- LogI, II, III (CLICK) : 4, 6,7 hours; 80 reducers
  - PVC: 3,4.5,5 hours with 40 reducers
  - PCC: 4.5, 8, 9 hours with 80 reducers

Summary

- Scalable map-reduce log-linear models to precisely estimate rare response rates by exploiting correlation structures with cross-product of hierarchies (OLAP structure)

- Models both accurate and parsimonious through “spike and slab” prior

- Significantly better than state-of-the-art logistic regression methods widely used in computational advertising