Multiview Hierarchical Bayesian Regression Model : Application to Online Advertising

Tianbing Xu
UC, Irvine
Bruce Zhang and Zhen Guo
Yahoo ! Labs

Outline

- Problem Statement
- Motivation
- Hierarchical Bayesian Mixture Regression Model(HBMR)
- EM Inference and Learning
- Map-Reduce Implementation
- Experimental Results
- Conclusion and Future Work

Problem Statement

- In RMX each campaign has a list of targeting attributes, objective is to receive desirable user actions (clicks/conversions) cost-effectively (ROI positive)
- Click is used in our work to measure campaign performance
- Billions of ads impressions (samples) from different campaigns are served daily
- Each sample is sparse in high dimensional space
- Each sample can have a click happened on it

Click is rare event, CTR < 0.1%

Two tasks

- 1) Predict performance (click)

  Develop a generative model to fit the impression/click data better

  Predict whether an impression can generate a click or not

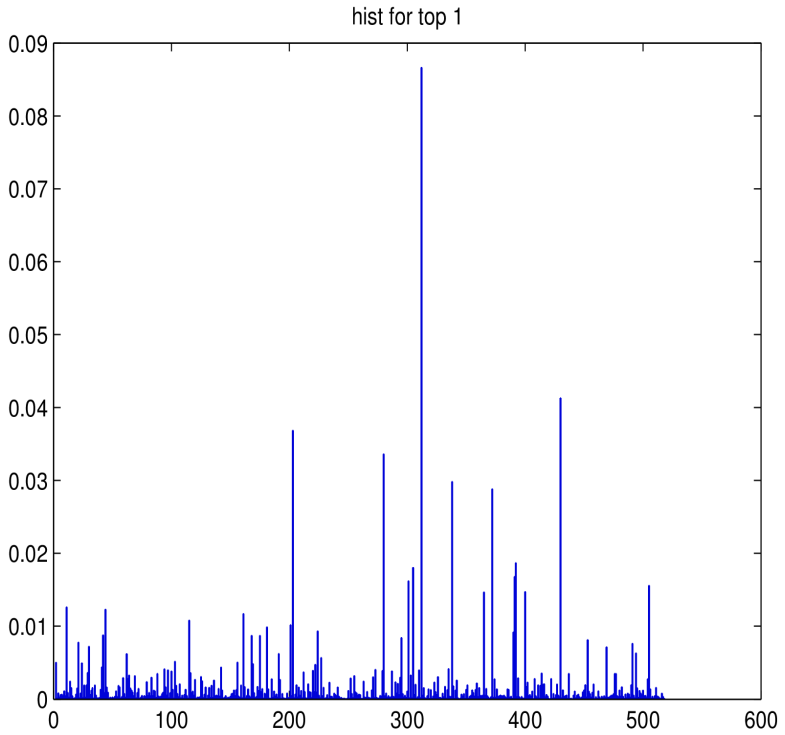
- 2) A generic method for targeting recommendation

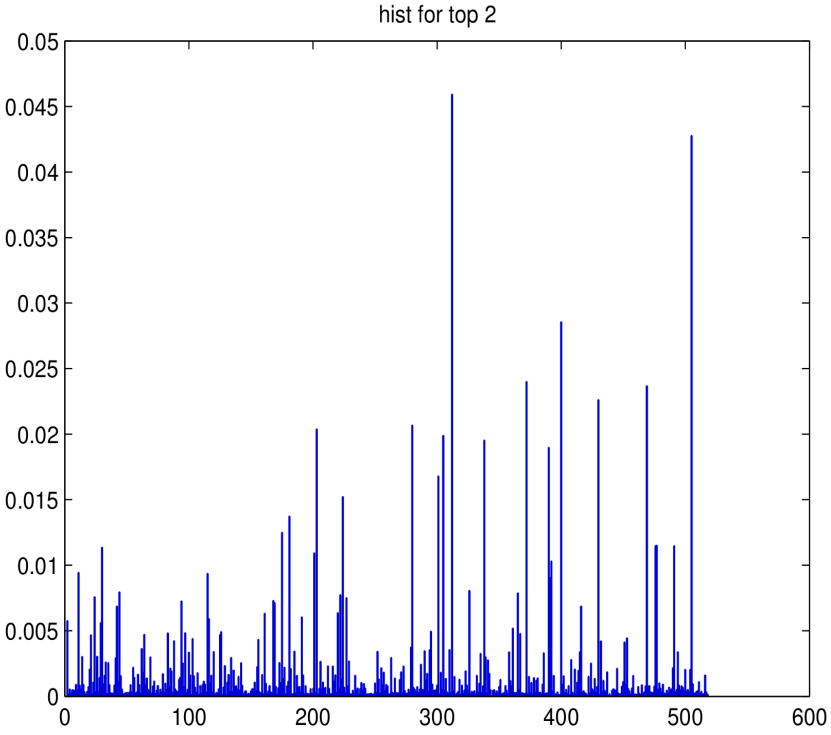
  Quantify targeting feature weights based on the learned model

  Recommend targeting features to be set or unset to maximize expected response (CTR) for each campaign

Motivation

Different clustering pattern for sample features





Motivation

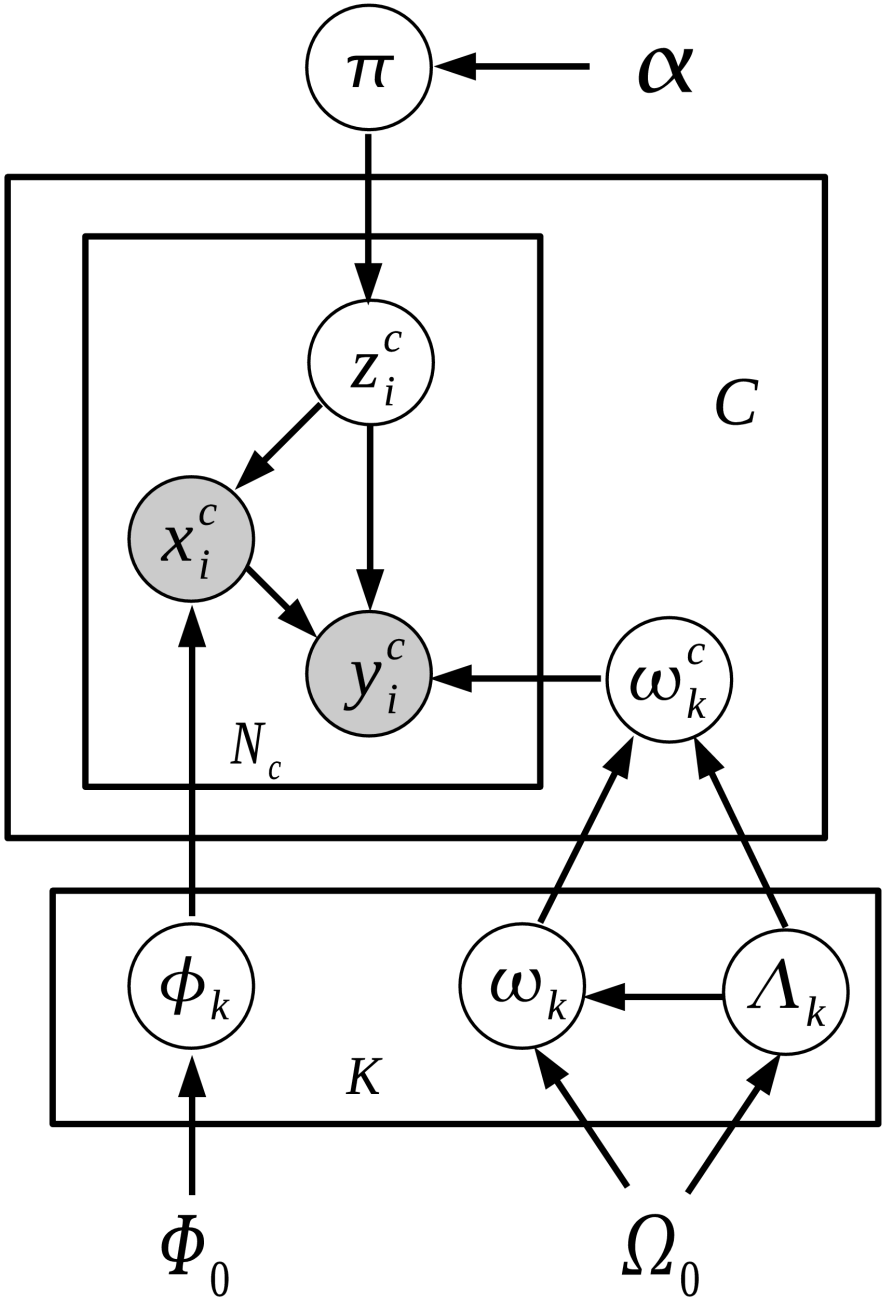
- For a specific Campaign, the targeting users come from different clusters based on co-occurrence of features (geological, publisher ID,

  user segments ...)

  For different Campaigns, they may share the same feature clusters
- For a user to click an impression, her behavior depends on her feature cluster and the ad campaign served

Hierarchical Bayesian Mixture Regression

- All impressions from all campaigns share global feature clusters ϕ_k
- Click or not is a logistic signal from interactions of features (impressions) ; this interaction ω_k^c (coefficient weights) shared across campaign and global user clusters
- For impressions in all campaigns, the interaction ω_k is shared across global feature clusters; for a specific campaign, interaction ω_k^c is a sample from distribution taking ω_k as mean


π
α
z_i^c
C
x_i^c
y_i^c
ω_k^c
N_c
ϕ_k
ω_k
Λ_k
K
Φ_0
Ω_0

Formalization

$$\pi \sim stick(\alpha)$$
$$z_i^c \sim mult(\pi)$$
$$x_i^c \sim f(\phi_{z_i^c})$$
$$ctr_i^c = \sigma(< \omega_i^c, x_i^c >)$$
$$y_i^c \sim bernoulli(ctr_i^c)$$

$$\Phi = \{\phi_k\}$$
$$\phi_k \sim G(\Phi_0)$$

$$\Lambda_k \sim \mathcal{W}(\nu_k, W_k)$$
$$\omega_k \sim \mathcal{N}(\mu_k, (\kappa_k \Lambda_k)^{-1})$$
$$\omega_k^c \sim \mathcal{N}(\omega_k, (\Lambda_k)^{-1})$$

$$\Omega_0 = \{\mu_k, \nu_k, \kappa_k\}$$
$$\Omega_c = \{\omega_k^c\}$$
$$\Omega_k = \{\omega_k, \Lambda_k\}$$
$$\Omega = \{\Omega_c, \Omega_k\}$$

Formalization

- log likelihood of Data, integrate out ϕ_k π

$$p(y_i^c, x_i^c|\Omega_c) = \sum_k p(z_i^c = k)p(x_i^c|z_i^c = k)p(y_i^c|x_i^c, z_i^c, \Omega_c)$$

Inference (Stochastic EM)

- We need to know which impression assigned to which user cluster ? --- Gibbs Sampling

- We need to learn the interactions ω_k^c between different features for specific campaign and feature cluster --- parameter estimation by maximizing MAP of parameters given cluster assignments and data

E-Step

- Sample feature cluster assignment for each impression given other impressions, clicks, and old parameters

$$D = \{\{x_i^c\}, \{y_i^c\}\}$$

$$p(z_i^c = k | Z_{-i}, x_i^c, y_i^c, D, \Omega_c) \propto$$
$$p(z_i^c = k | Z_{-i}) p(x_i^c | z_i^c = k, D) p(y_i^c | x_i^c, z_i^c = k, D, \Omega_c)$$

E-Step

- Chinese restaurant process
- Topic assignment prior

$$p(z_i^c = k | Z_{-i}) = \begin{cases} \frac{n_k}{n-1+\alpha} & k \leq K \\ \frac{\alpha}{n-1+\alpha} & k = K+1 \end{cases}$$

E-Step

- Data likelihood by integrating out ϕ_k by prior-posterior Conjugacy

$$p(x_i^c|z_i^c = k, D_{-i}) = \int p(x_i^c|\phi_k)p(\phi_k|D_{-i}, z_j^c = k)$$
$$= \frac{\int p(x_i^c|\phi_k)\prod_{z_j^c=k} p(x_j^c|\phi_k)p(\phi_k|\phi_0)}{\int \prod_{z_j^c=k} p(x_j^c|\phi_k)p(\phi_k|\phi_0)}$$
$$= \frac{B(\phi_k^* + x_i^c)}{B(\phi_k^*)}$$

$$\phi_k^* = \phi_0 + \sum_{z_j^c=k} x_j$$

M-Step

- posterior of ω_k^c

$$L(\omega_k^c|D,Z) = p(\omega_k^c|\{x_i^c\},\{y_i^c\},\{z_i^c\},\Omega_k)$$
$$\propto p(\omega_k^c|\Omega_k)\prod_{z_i^c=k} p(y_i^c|x_i^c,z_i^c=k,\omega_k^c)$$

- Optimize posterior likelihood to estimate ω_k^c

$$\omega_k^c = argmax_{\omega_k^c} \quad L(\omega_k^c|D,Z)$$

- Equivalent to logistic regression with L2 regularization, Newton Raphson to solve it

M-Step

- Update Posterior of $\Omega_k = \{\omega_k, \Lambda_k\}$
- Prior is Gaussian-Wishart on Ω_0

$$p(\omega_k, \Lambda_k|\Omega_0) = \mathcal{N}(\mu_k, (\kappa_k\Lambda_k)^{-1})\mathcal{W}(\nu_k, W_k)$$

Posterior is again Gaussian-Wishart

$$p(\omega_k, \Lambda_k|\Omega_0, \{\omega_k^c\}) = \mathcal{N}(\mu_k^*, (\kappa_k^*\Lambda_k)^{-1})\mathcal{W}(\nu_k^*, W_k^*)$$

$$\mu_k^* = \frac{\kappa_k^* + C\bar{\omega}_k}{\kappa_k^* + C} \quad \kappa_k^* = \kappa_k + C \quad \nu_k^* = \nu_k + C$$

$$[W_k^*]^{-1} = [W_k]^{-1} + C\bar{S}_k + \frac{C\kappa}{C+\kappa}(\mu_k - \bar{\omega}_k)(\mu_k - \bar{\omega}_k)'$$

$$\bar{\omega}_k = \frac{1}{C}\sum_c \omega_k^c \quad \bar{S}_k = \frac{1}{C}\sum_c \omega_k^c\omega_k^{c\prime}$$

M-Step

- Once, we have updated the posterior parameters, need to sample $\Omega_k = \{\omega_k, \Lambda_k\}$

  from Guassian and Wishart distribution
- $\Omega_k = \{\omega_k, \Lambda_k\}$ L2 regularized term to estimate ω_k^c

Map-Reduce framework

- We have billions of samples, how to fit EM inference and learning into Map Reduce?
- Idea: Each iteration has 2 Jobs,

  job1 for E-Step, job2 for M-Step
- Sufficient statistics to estimate parameters, calculate data likelihood or update posterior are in summation form, good to fit into MR framework
- After job2, still need to get the total sum of posterior parameters to sample $\Omega_k = \{\omega_k, \Lambda_k\}$

Map-Reduce

- For each EM iteration

  1. Job1, E-Step, Gibbs Sampling

  2. Job2, M-Step, Newton Raphson

  3. sum the partial sum to update the posterior distribution of $\Omega_k = \{\omega_k, \Lambda_k\}$

Map-Reduce

- Job1

  do Gibbs Sampling to get topic assignment for each sample based on random subset of samples and old parameters

- Use random subset to approximate the distribution of the full data set
- If each Mapper has sufficient data, this approximation would be reasonable

- Job 2

  estimate regression weights by Newton Raphson

- need to calculate gradient and hessian matrix based on

  the samples in Reducer

- Update parameters of posterior distribution of

  $\Omega_k = \{\omega_k, \Lambda_k\}$; depends on

  all samples in certain topic;

  need to calculate after job2

Job1 (E-Step)

- Mapper

  <lineID, line> -->

  <randkey, line>

  randomly emit lines (samples) to different

  reducer

- Reducer

  <randkey, list <line>>

  -->

  <z_i+campaign, line>

- Do gibbs sampling, emit each sample i to its cluster zi = k and campaign c

Job2 (M-Step)

- Mapper (read from output of job1)

  <k+c, line>

  ----- >

  <k+c, line>

- Identity Mapping

- Reducer

  <k+c, list<line>>

  ----->

  <k+c, parameters>

- Parameters include:

  $\{\omega_k^c, n_k^c, \bar{x}_k^c, S_k^c\}$

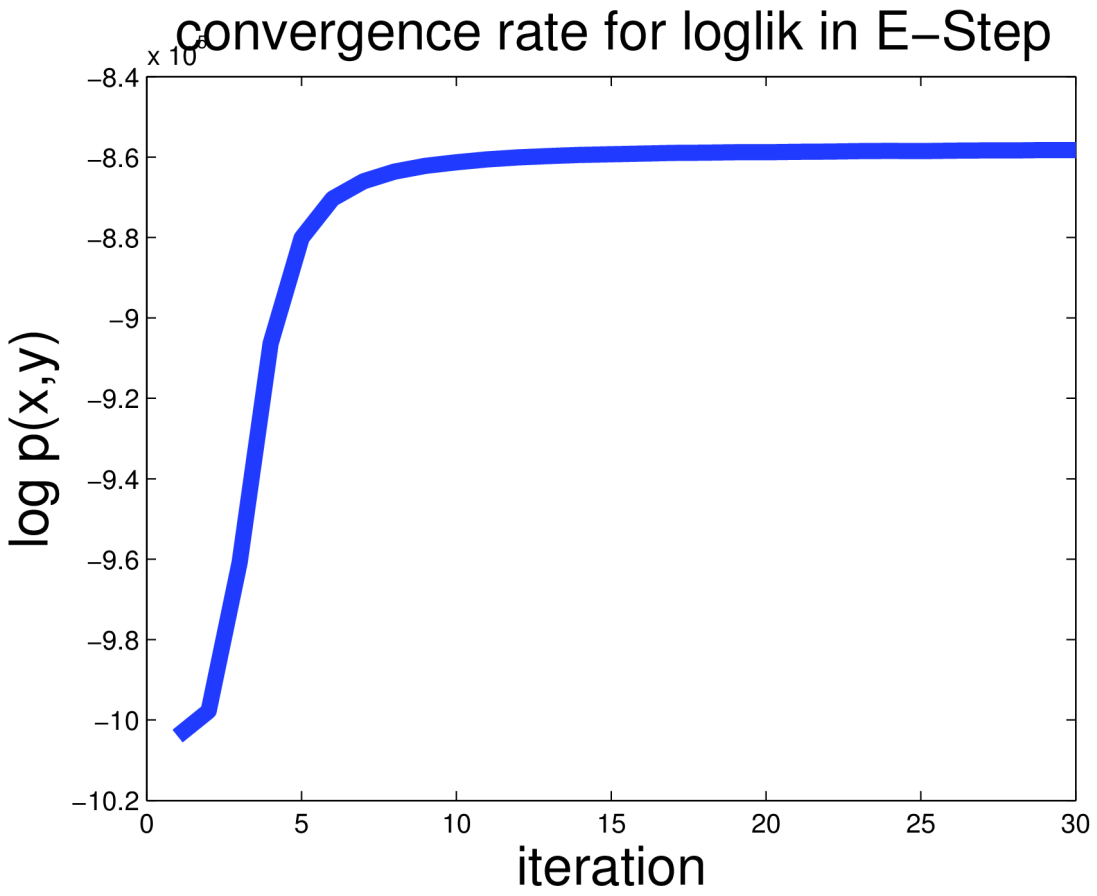
Distributed Cache

- Mainly used to store and pass parameters
- In Job1, we need to load old parameters from distributed Cache to do Sampling
- In Job2, we estimate the current parameters and store them to distributed Cache

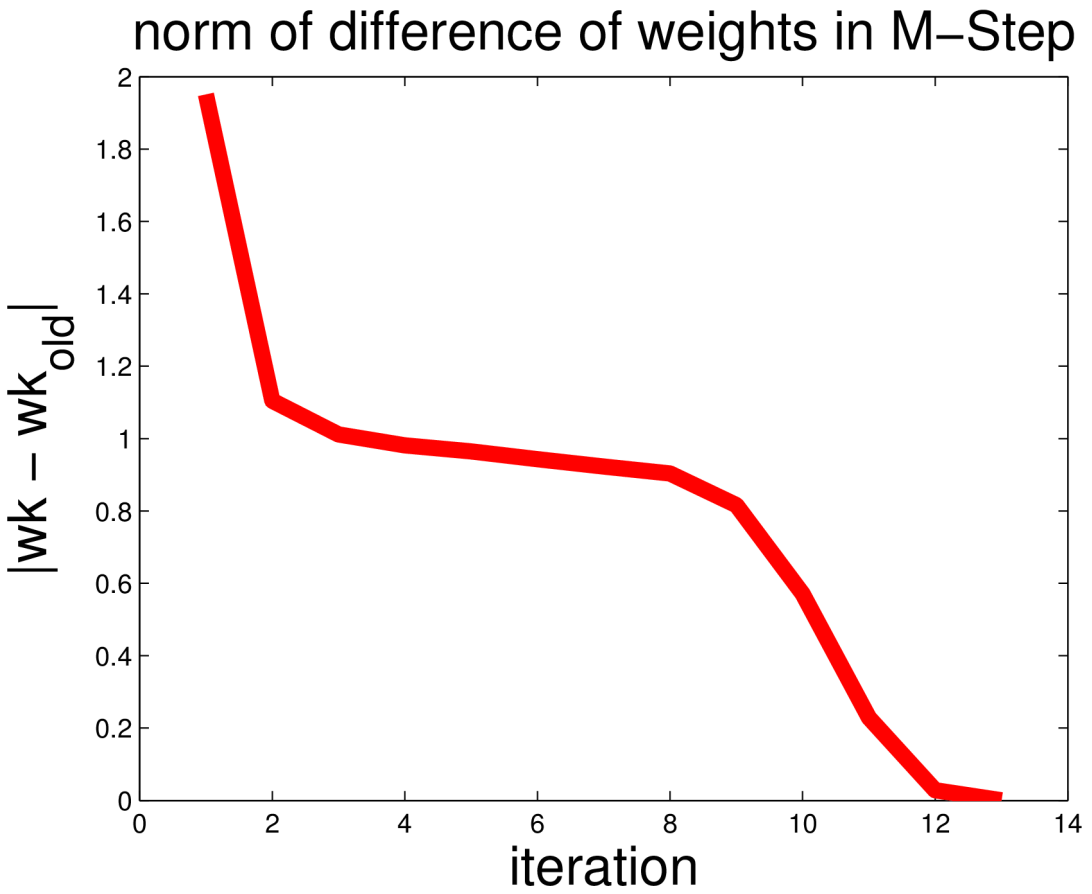
Experiment Result

- Convergence
- Feature pruning
- CTR prediction
- Targeting feature recommendation

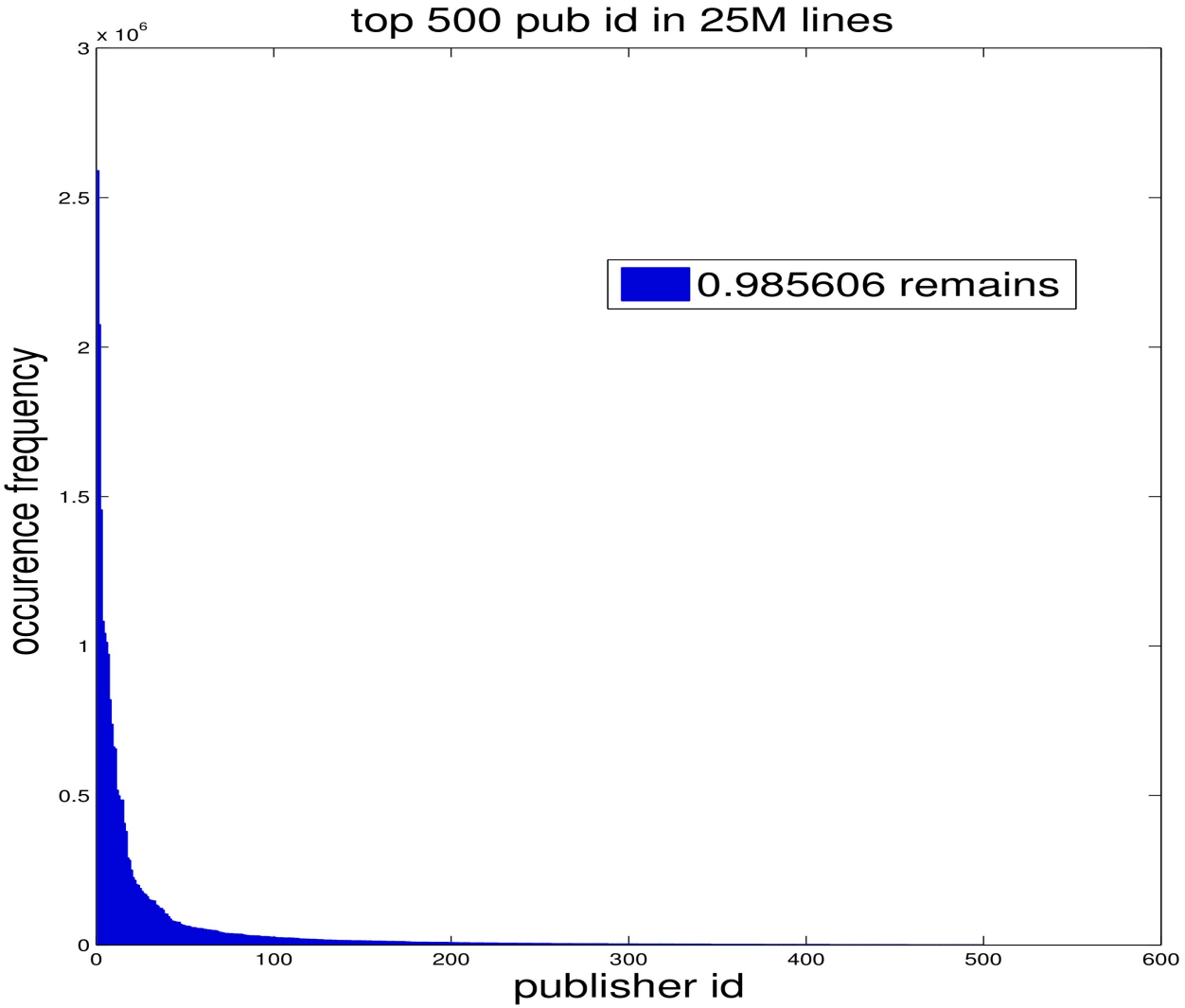
Convergence of E-Step



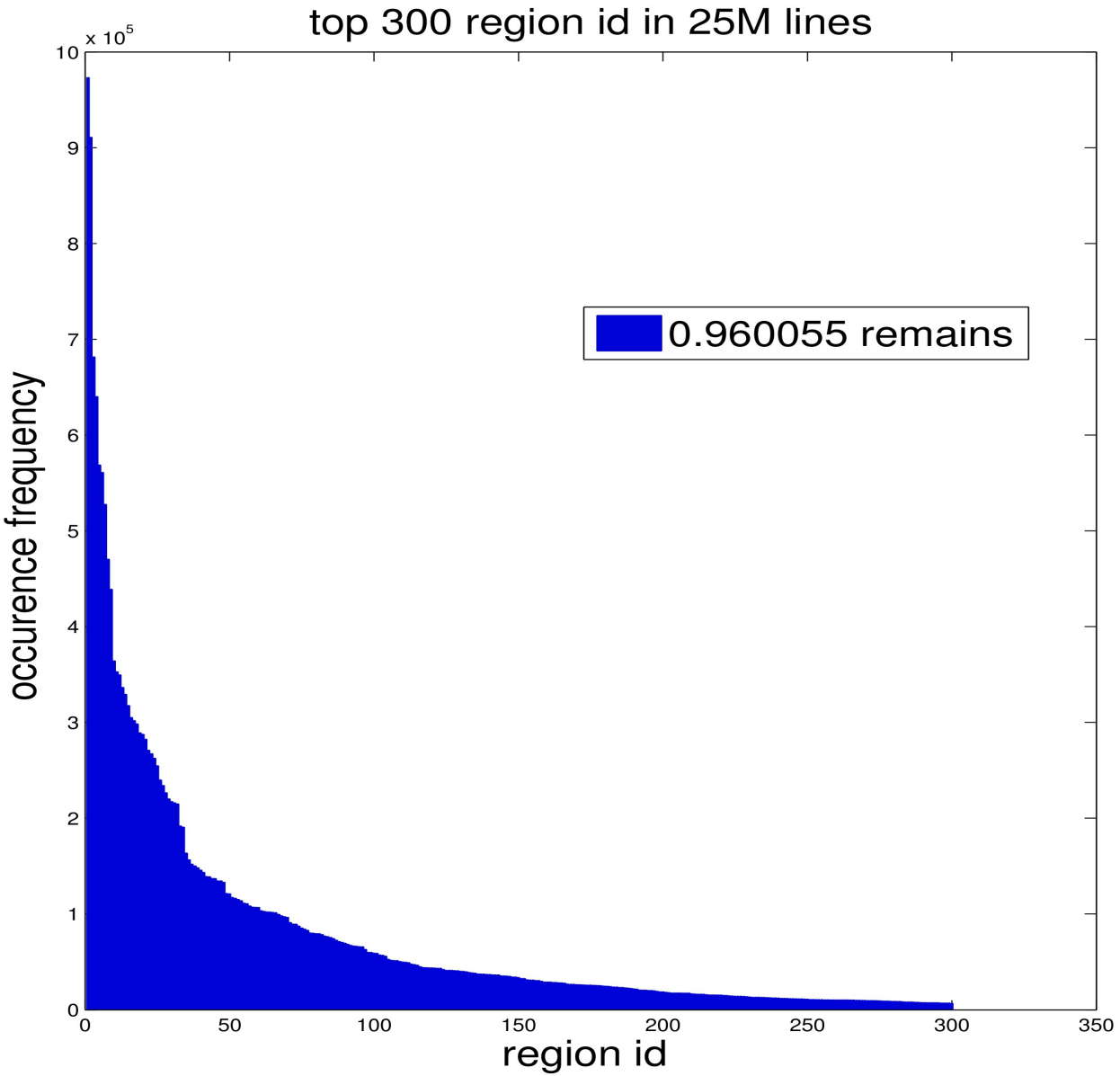
Convergence of M-Step



Feature pruning (publisher ID)



Region ID


top 300 region id in 25M lines
x 10^5
10
9
8
7
6
5
4
3
2
1
0
occurence frequency
0.960055 remains
0
50
100
150
200
250
300
350
region id

Scalability

- K = 10, dim = 518

Data set	E-Step(30 iterations)	M-Step(20 iterations)	J1 M/R	J2 M/R
25M lines	6'11''	4'14'	5/100	100/20
1M lines	5'2''	1'8''	2/5	5/5

- Able to scale to billions of samples or even larger as long as we have enough mappers and reducers

CTR Prediction

- CTR prediction

  We randomly split data into training data and testing data
- Training data is used to estimate the

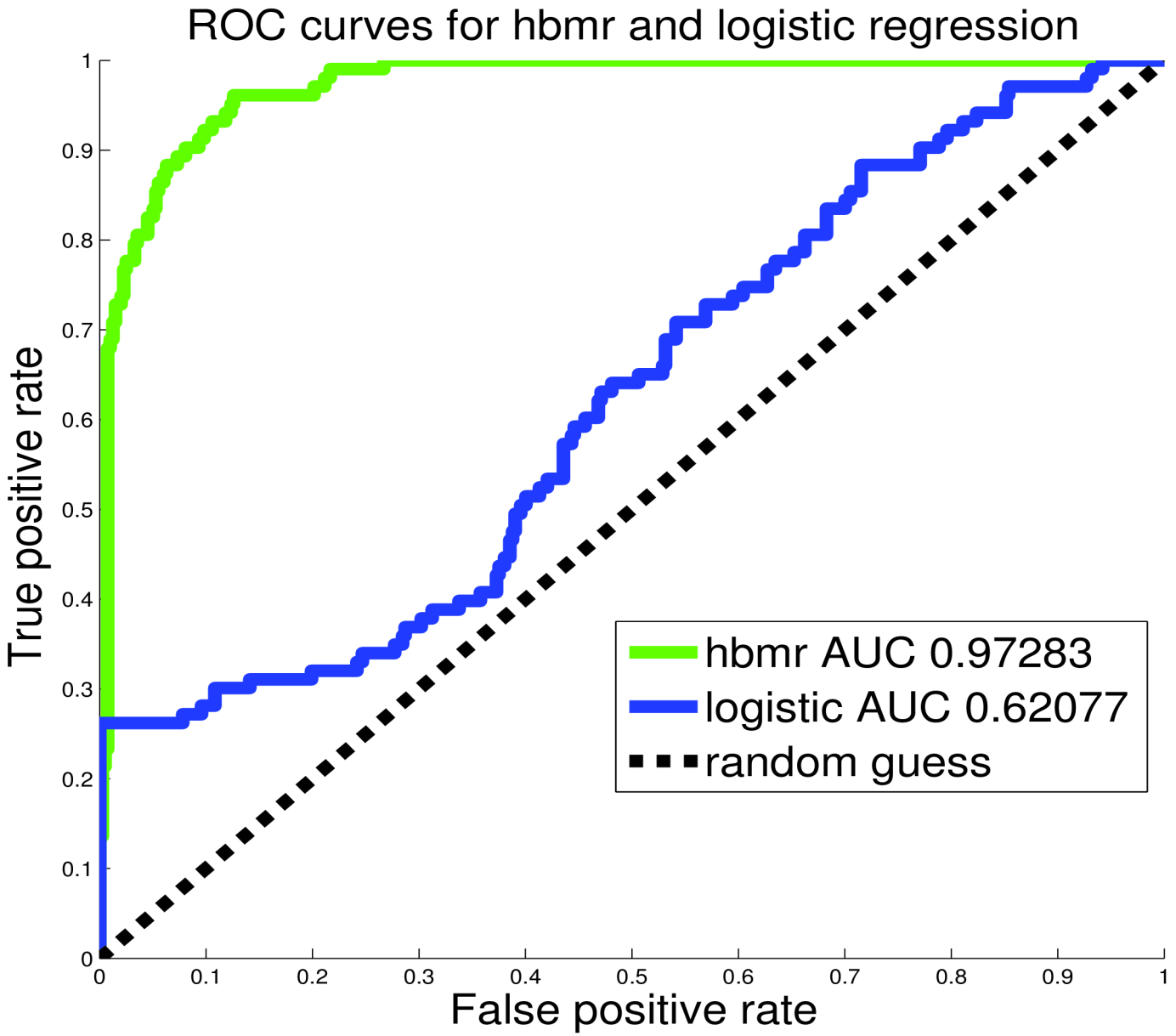
  coefficient weights ω_k^c , data likelihood posterior parameter, feature cluster mixture proportions
- Test data is used to calculate the CTR given the above parameters

CTR prediction

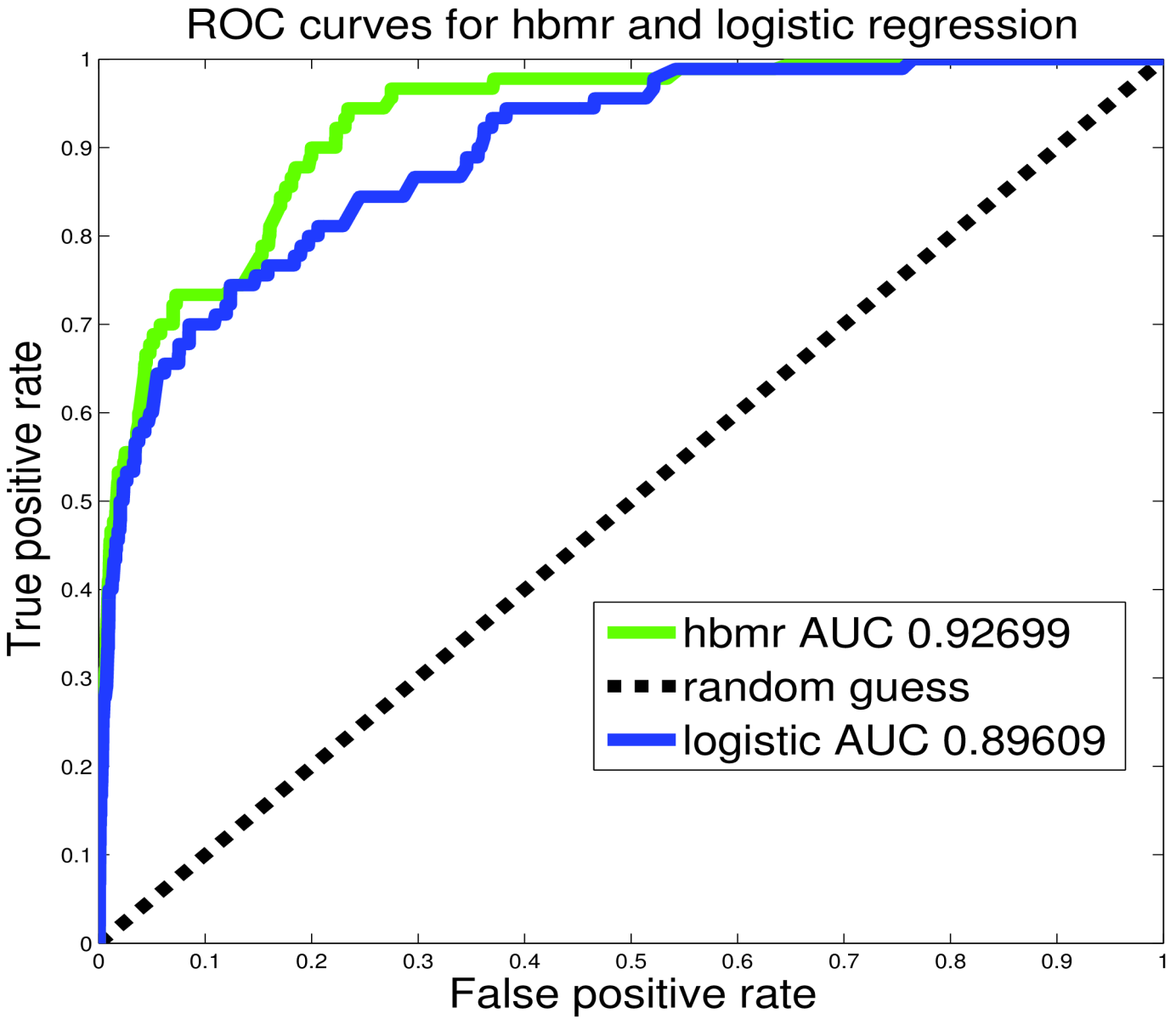
$$p(y_i^c = 1 | x_i^c, D, \Omega) = \frac{p(y_i^c = 1, x_i^c | D, \Omega)}{\sum_{y_i^c \in 0,1} p(y_i^c, x_i^c | D, \Omega)}$$

$$p(y_i^c, x_i^c | D, \Omega) = \sum_k p(z_i^c | D) p(x_i^c | D, z_i^c = k) p(y_i^c | x_i^c, z_i^c = k, \Omega)$$

Synthetic Data



25M impressions real data



Targeting Feature Recommendation

- Given training data, we have estimation of parameters
- Fix these parameters, vary campaign feature x^c , to optimization CTR

$$L(x^c|D,\Omega) = p(y^c = 1|x^c, D, \Omega)$$
$$x^c = argmax_{x^c \in \{0,1\}^d} L(x^c|D,\Omega)$$

- Ranking feature bit with $x_j^c = 1$ according to weighted average coefficient weights

Illustration (left: our model, right: logistic regression)

campaign 2330222

number of zero weight 331 non zero weight 187

500 -- 2.470525 – zshare.net

488 -- 2.295471 -- oyunlar1.com

335 -- 1.660604 -- Filestube

425 -- 1.572763 -- Pizap - Admeld

510 -- 1.498376 -- MediaFire - Rubicon

408 -- 1.308869 -- E Lyrics

396 -- 1.164445 -- rekza.com

447 -- 1.127962 -- Adreactor - Rubicon

372 -- 1.029930 -- INTL News - Rubicon

384 -- 0.982318 -- TVfun.ma

411 -- 0.974220 -- MySpace - Rubicon

392 -- 0.969694 -- tubidy.mobi

477 -- 0.891442 -- Edge Media Group, LLC

96 -- 0.881549 -- NA_23424846

208 -- 0.660765 -- NA_23424911

279 -- 0.593520 -- NA_23424922

314 -- 0.487143 -- radioreloaded.com

number of zero weight 356 non zero weight 162

477 -- 1.945231 -- Edge Media Group, LLC

500 -- 1.937136 -- zshare.net

425 -- 1.401103 -- Pizap - Admeld

335 -- 1.334491 -- Filestube

488 -- 1.159914 -- oyunlar1.com

208 -- 1.057995 -- NA_23424911

430 -- 0.975261 -- pinoy-ako.info

510 -- 0.964224 -- MediaFire - Rubicon

80 -- 0.752674 -- NA_23424778

193 -- 0.718785 -- NA_23424800

284 -- 0.718064 -- NA_24865674

498 -- 0.716463 -- LiveTV.ru

384 -- 0.633917 -- TVfun.ma

392 -- 0.601269 -- tubidy.mobi

215 -- 0.599884 -- NA_23424824

39 -- 0.598310 -- NA_23424897

43 -- 0.595442 -- NA_23424969

campaign 2811039

number of zero weight 418 non zero weight 100

175 -- 1.804266 -- District of Columbia

382 -- 0.573735 -- Games_new

512 -- 0.377319 -- Finance

304 -- 0.350371 -- Network

487 -- 0.318669 -- Verizon home

341 -- 0.281278 -- Market - Huffington Post Direct

390 -- 0.257686 -- mail_sp

412 -- 0.228539 -- uk sports

505 -- 0.161309 -- Groups

171 -- 0.143111 -- Delaware

391 -- 0.142294 -- mail_ukie

130 -- 0.113230 -- Mississippi

300 -- 0.110154 -- Mail

313 -- 0.096021 -- Pager

501 -- 0.095348 -- Astrology

489 -- 0.078123 -- Chat

317 -- 0.068138 -- Sports

number of zero weight 425 non zero weight 93

390 -- 0.910818 -- mail_sp

219 -- 0.818121 -- Ohio

125 -- 0.540816 -- Michigan

331 -- 0.515448 -- Flickr

410 -- 0.347384 – Mail_partners_att

174 -- 0.053287 -- Florida

453 -- -0.011030 -- germany

33 -- -0.014635 -- Wyoming

512 -- -0.017393 -- Finance

225 -- -0.018363 -- North Dakota

24 -- -0.023201 -- South Dakota

374 -- -0.023946 -- Knowledge search

304 -- -0.029093 -- Network

26 -- -0.031584 -- Vermont

501 -- -0.032342 -- Astrology

86 -- -0.032430 -- Hawaii

134 -- -0.033991 -- Montana

Conclusion

- Develop Hierarchical Bayesian Mixture Regression Model
- Develop EM inference and learning algorithm
- Fit EM into Map-Reduce framework to scale to large real data set
- Superior experiment results on CTR prediction and targeting feature recommendation compared to logistic regression baseline

Future Work

- Experiment on larger data set
- Test more features, such as user segments, channel id, user gender, age, …
- How to learn K in Map-Reduce in implementation?

Acknowledgement

- Bo Long, Jerry Ye
- Yike Ren, Yang Zhou, Ying Cui, Javad Azimi

Thanks

- Any Question?

$$\mathcal{W}(\Lambda_k | W_k, \nu_k) \propto \det(\Lambda_k)^{\frac{\nu_k - d - 1}{2}} \exp\{-\frac{1}{2} Trace(W_k^{-1} \Lambda_k)\}$$