Recommender Problems for Content Optimization

Deepak Agarwal

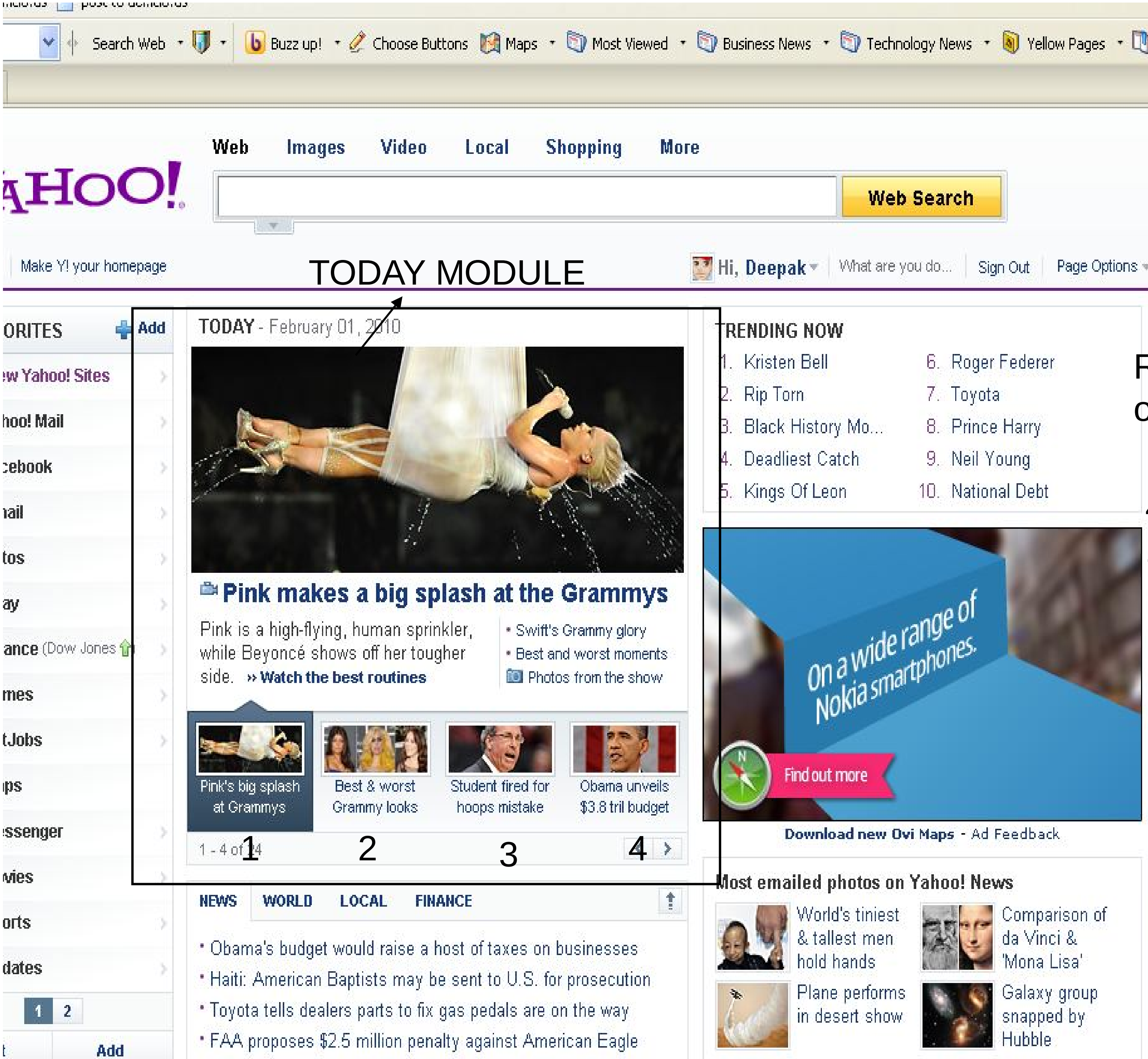
Yahoo! Research

MMDS, June 15th, 2010

Stanford, CA

Main Collaborators in Research

- Bee-Chung Chen (Yahoo!)
- Pradheep Elango (Yahoo!)
- Raghu Ramakrishnan (Yahoo!)
- Several others in Engineering, Product contributed to the ideas presented in this talk



Routes Traffic to other Y! pages

4 slots exposed 1,2,3,4

First slot has Max exposure

Problem definition

- Display “best” articles for each user visit
- Best - Maximize User Satisfaction, Engagement
  - BUT Hard to obtain quick feedback to measure these
- Approximation
  - Maximize utility based on immediate feedback (click rate) subject to constraints (relevance, freshness, diversity)
- Inventory of articles?
  - Created by human editors
  - Small pool (30-50 articles) but refreshes periodically

Where are we today?

- Before this research : Articles created and selected for display by editors
- After this research : Article placement done through statistical models
- How successful ?

"Just look at our homepage, for example. Since we began pairing our content optimization technology with editorial expertise, we've seen click-through rates in the Today module more than double. And we're making additional improvements to this technology that will make the user experience ever more personally relevant."

----- Carol Bartz, CEO Yahoo! Inc (Q4, 2009)

We've always been focused on specific events like the Olympics – not just as short-term traffic drivers, but also as ways to draw users into the Yahoo! experience and more deeply engage with them over time. Yet we know we can't run a business just waiting for major sporting events, awards shows and natural disasters. In earlier quarters, you've heard me mention that we need to attract these types of audiences every day.

That's why we've been using our unique approach of combining human editors to choose great stories – and letting our content optimization engine determine the best content for our users. I want to talk about this content engine for a second, because it's an amazing technology that has been growing more and more advanced over the last several months.

In its first iteration, our content optimization engine recommended the most popular news items to our users. The result was a 100% click-thru rate increase over time. In January, we introduced release 2 of the engine, which included some of our behavioral targeting technology. This capability – coupled with great content – led our Today Module to experience click-thru rates 160% over pre-engine implementation.

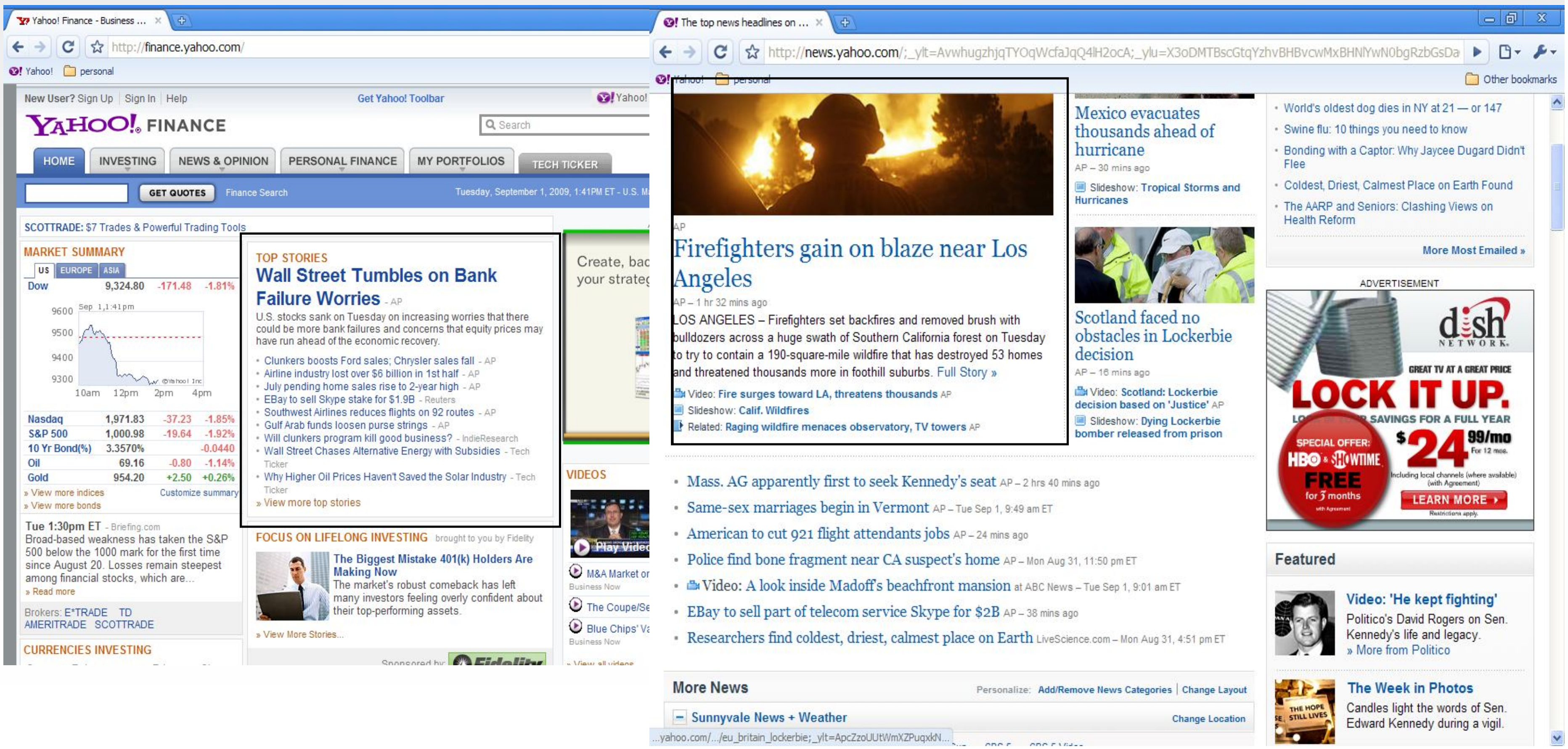
----- Carol Bartz, CEO Yahoo! (Q1, 2010)

Main Goals

- Methods to select most popular articles
  - This was done by editors before
- Provide personalized article selection
  - Based on user covariates
  - Based on per user behavior
- Scalability: Methods to generalize in small traffic scenarios
  - Today module part of most Y! portals around the world
  - Also syndicated to sources like Y! Mail, Y! IM etc

Similar applications

- Goal: Use same methods for selecting most popular, personalization across different applications at Y!
- Good news! Methods generalize, already in use



Rest of the talk

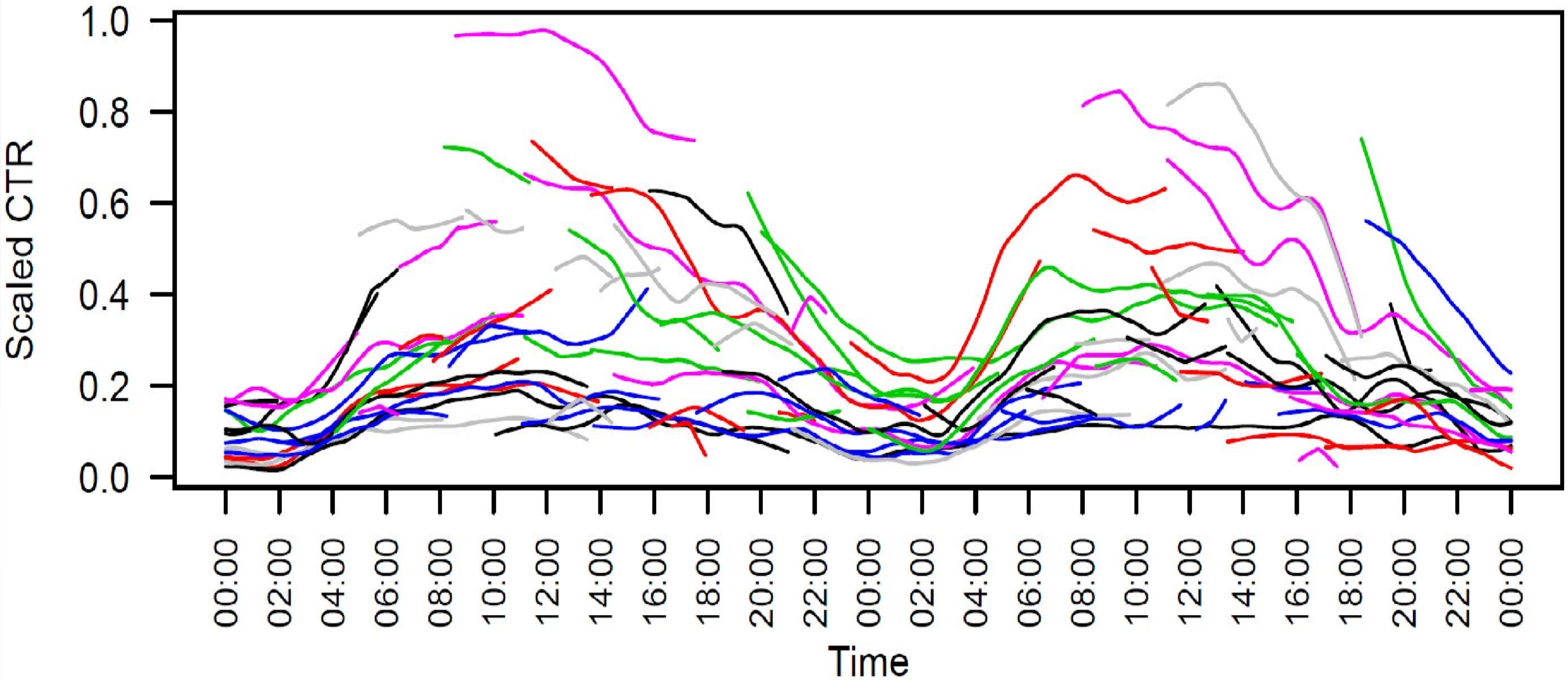
- Selecting most popular with dynamic content pool
  - Time series, multi-armed bandits
- Personalization using user covariates
  - Online logistic regression, reduced rank regression
- Personalization based on covariates and past activity
  - Matrix factorization (bilinear random-effects model)

Assumptions made in this talk

- Single slot optimization (Slot 1 with maximum exposure)
  - Multi-slot optimization with differential exposure future work
- Inventory creation and statistical models decoupled
  - Ideally, there should be a feedback loop
- Effects like user-fatigue, diversity in recommendations, multi-objective optimization not considered
  - These are important

Selecting Most Popular with Dynamic Content Pool

Article click rates over 2 days on Today module



No confounding, traffic obtained from a controlled randomized experiment

Things to note:

a) Short lifetimes b) temporal effects c) often breaking news story

Statistical Issues

- Temporal variations in article click-rates
- Short article lifetimes → quick reaction important
  - Cannot miss out on a breaking news story
  - Cold-start : rapidly learning click-rates of new articles
- Monitoring a set of curves and picking the best
  - Set is not static

- Approach
  - Temporal - Standard time-series model coupled with
  - Bayesian sequential design (multi-armed bandits)
    - To handle cold-start

Time series Model for a single article

- Dynamic Gamma-Poisson with multiplicative state evolution

$$c_t \mid n_t, p_t \sim \text{Poisson}(n_t p_t)$$

$$p_{t+1} = p_t \epsilon_{t+1}$$

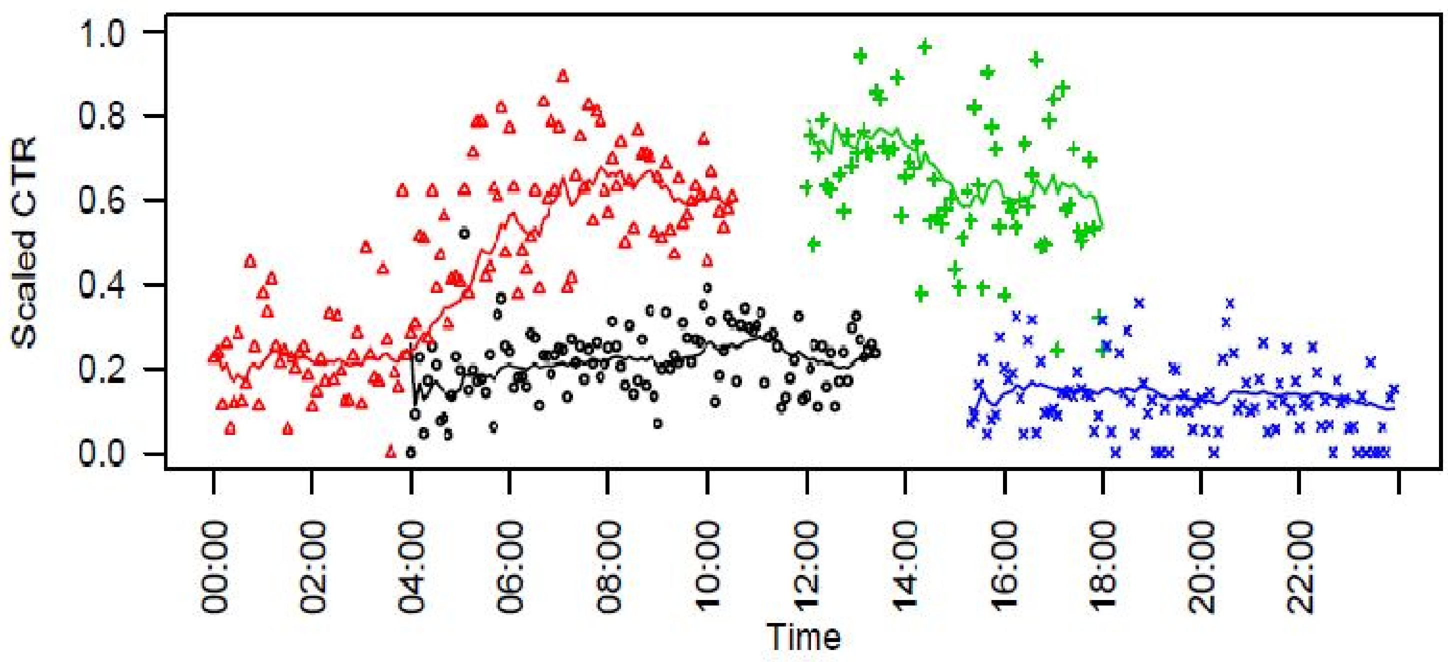
$$\epsilon_{t+1} \sim \mathcal{D}(\text{mean} = 1, \text{var} = \eta)$$

- Click-rate distribution at time t+1
  - Prior mean: $E(p_{t+1} \mid D_t) = \hat{p}_{t|t}$
  - Prior variance: $Var(p_{t+1} \mid D_t) = \hat{\sigma}^2_{t|t} + \eta(\hat{p}^2_{t|t} + \hat{\sigma}^2_{t|t})$

High CTR items more adaptive

Tracking behavior of Gamma-Poisson model

- Low click rate articles – More temporal smoothing



Explore/exploit for cold-start

- New articles (or articles with high variance) with low mean
- How to learn without incurring high cost
- Slow reaction:
  - can be bad if article is good
- Too aggressive:
  - may end up showing bad articles for a lot of visits
- What is the optimal trade-off?
  - Article 1: CTR = 2/100; Article 2: CTR = 25/1000
  - Best explore/exploit strategy
  - Look ahead in the future before making a decision
    - Bandit problem

Cold-start: Bayesian scheme, 2 intervals, 2 articles

- 2 interval look-ahead : # visits N_0, N_1
- Article 1 prior CTR p_0 ~ Gamma(α, γ)
  - Article 2: CTR q_0 and q_1, $\mathrm{Var}(q_0) = \mathrm{Var}(q_1) = 0$

- Design parameter: x (fraction of visits allocated to article 1)

- Let $c \mid p_0 \sim \mathrm{Poisson}(p_0(xN_0))$: clicks on article *1, interval 0.*

- Prior gets updated to posterior: Gamma(α+c,γ+xN_0)

- Allocate visits to better article in interval 2
  - i.e. to item *1* iff post mean item 1 = $E[p_1 \mid c, x] > q_1$

Optimization

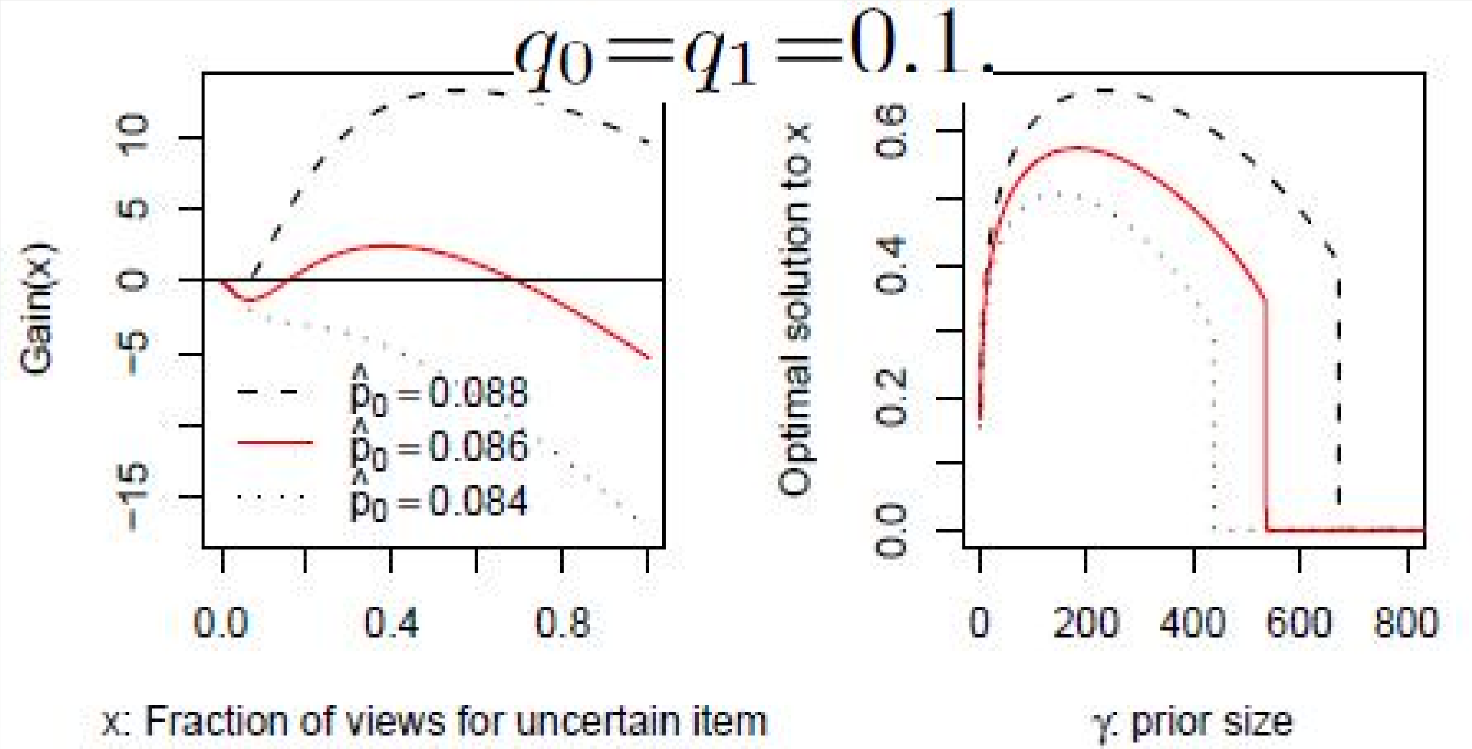
- Expected total number of clicks

$$N_0(x\hat{p}_0 + (1-x)q_0) + N_1 E_{c|x}[\max\{\hat{p}_1(x,c),\ q_1\}]$$

$$= \underbrace{N_0 q_0 + N_1 q_1}_{\text{E[\#clicks] if we always show the certain item}} + \underbrace{N_0 x(\hat{p}_0 - q_0) + N_1 E_{c|x}[\max\{\hat{p}_1(x,c) - q_1,\ 0\}]}_{\substack{Gain(x,\ q_0,\ q_1) \\ \text{Gain from experimentation}}}$$

$$x_{opt} = \text{argmax}_x\ \text{Gain}(x,\ q_0,\ q_1)$$

Example for Gain function



(a) Gain function (b) Optimal solution

Generalization to K articles

- Objective function

$$R(\mathbf{x}, \boldsymbol{\theta}_0, N_0, N_1) = N_0 \sum_i x_{i0} \mu(\boldsymbol{\theta}_{i0}) + N_1 \sum_i E_{\boldsymbol{\theta}_1} [x_{i1}(\boldsymbol{\theta}_1) \mu(\boldsymbol{\theta}_{i1})].$$

Our goal is to find

$$R^*(\boldsymbol{\theta}_0, N_0, N_1) = \max_{0 \leq \mathbf{x} \leq 1} R(\mathbf{x}, \boldsymbol{\theta}_0, N_0, N_1), \text{ subject tc}$$

$$\sum_i x_{i0} = 1 \text{ and } \sum_i x_{i1}(\boldsymbol{\theta}_1) = 1, \text{for all possible } \boldsymbol{\theta}_1.$$

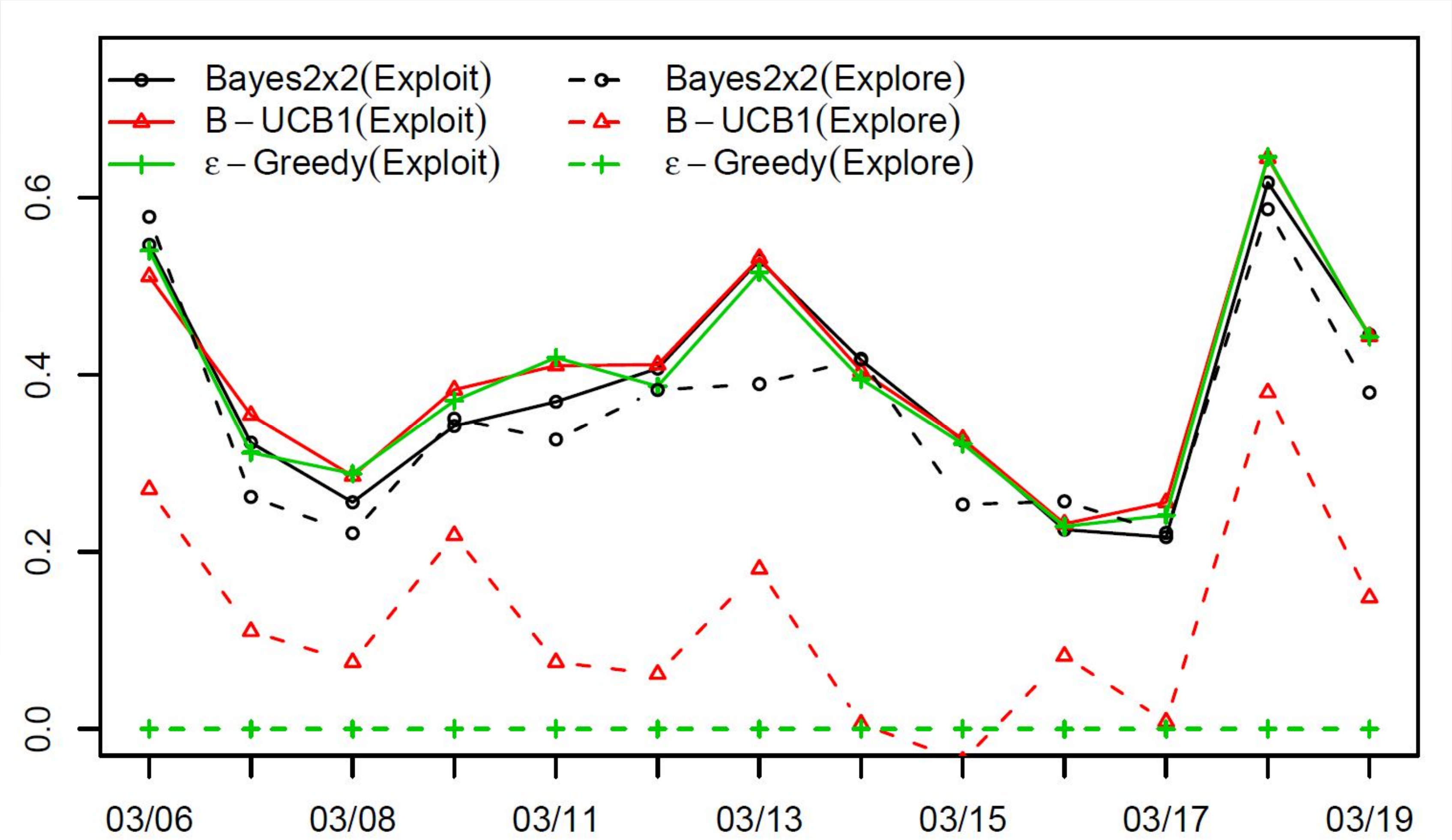
- Langrange relaxation (Whittle)

$$R^+(\boldsymbol{\theta}_0, N_0, N_1) = \max_{0 \leq \mathbf{x} \leq 1} R(\mathbf{x}, \boldsymbol{\theta}_0, N_0, N_1$$

$$\Sigma_i \; x_{i0} = 1 \text{ and } E_{\boldsymbol{\theta}_1} \left[\Sigma_i \; x_{i1}(\boldsymbol{\theta}_1)\right] = 1.$$

Test on Live Traffic

15% explore (samples to find the best article);
85% serve the “estimated” best (false convergence)



Covariate based personalization

DATA

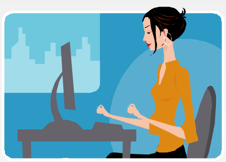
INVENTORY

Time t

Algorithm selects Item j

covariates $\mathbf{x}_j$

(keywords, content categories,..)



User *i* visits (i,j) : response y_{ijt} (click/no-click)

covariates $\mathbf{x}_{it}$

(demographics, Browse history, search history)

click-rate p_{ijt}

Model: (Y_t; P_t); t=1,2,….

Natural model: Logistic regression

- Estimating (user, item) interactions for a large, unbalanced and massively incomplete 2-way binary response matrix
- Natural (simple) statistical model

$$y_{ijt} \sim \text{Bernoulli}(p_{ijt})$$

$$s_{ijt} = \log \frac{p_{ijt}}{1-p_{ijt}}$$

Item coefficients

$$s_{ijt} = \boldsymbol{x}'_{it}\mathbf{A}\boldsymbol{x}_j + \boldsymbol{x}'_{it}\boldsymbol{v}_{jt}$$

High dimensional random-effects
In our examples, dimension ~ 1000

- Per-item online model
  - **must estimate quickly for new items**

Connection to Reduced Rank Regression (Anderson, 1951)

- N x p response matrix (p= #items, N=#users)
- Each row has a covariate vector $\mathbf{x}_i$ (user covariates)

- p regressions, each of dim q: $(x_i' \boldsymbol{v}_1, x_i' \boldsymbol{v}_2, \ldots, x_i' \boldsymbol{v}_p)$
  - $\boldsymbol{V}_{q \times p}$: too many parameters
  - Reduced rank: $\boldsymbol{V}^T = \mathbf{B}_{p \times r} \, \mathbf{\Theta}_{r \times q}$ (r << q; rank reduction)
- Generalization to categorical data
  - Took some time, happened in around '00 (Hastie et al)
- Difference
  - Response matrix highly incomplete
  - Goal to expedite sequential learning for new items

Reduced Rank for our new article problem

- Generalize reduced rank for large incomplete matrix

$$s_{ijt} = \boldsymbol{x}'_{it}\mathbf{A}\boldsymbol{x}_j + \boldsymbol{x}'_{it}B\boldsymbol{\theta}_j$$

Low dimension (5-10),

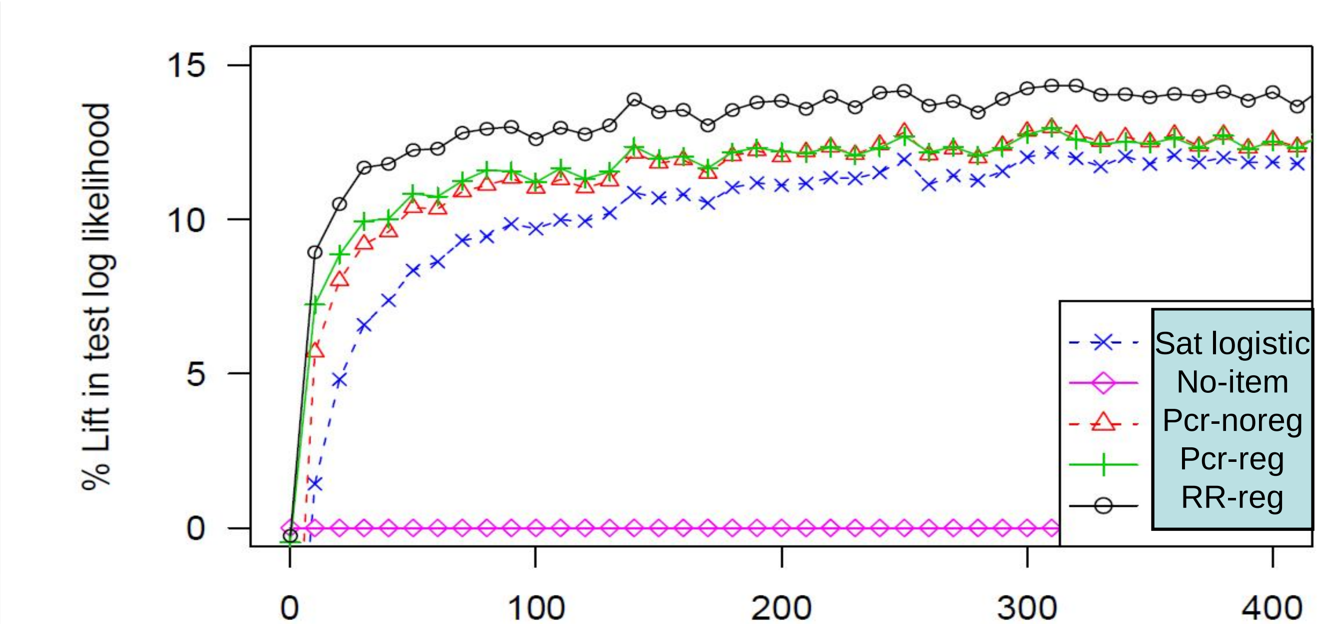
B estimated retrospective data

- Application different than in classical reduced rank literature
  - Cold-start problem in recommender problems

Experiment

- Front Page Today module data ~ 1000 user covariates (age, gender, geo, browse behavior)
- Reduced rank trained on historic data to get *B* of ranks 1,2,..,10
- For out-of-sample predictions, items all new
- Model selection for each item done based on predictive log-likelihood
- We report performance in terms of out-of-sample log-likelihood
- Baseline methods we compare against
  - *Sat-logistic : online logistic per item with ~1000 parameters*
  - *No-item: regression based only on item features*
  - *Pcr-reg; Pcr-noreg: principal components used to estimate B*
  - *RR-reg: reduced rank procedure*

Results for Online Reduced Rank regression



- Summary:
  – Reduced rank regression significantly improves performance compared to other baseline methods

– *Sat-logistic : online logistic per item with ~1000 parameters*

– *No-item: regression based only on item features*

– *Pcr-reg; Pcr-noreg: principal components used to estimate B*

– *RR-reg: reduced rank procedure*

Per user, per item models
via bilinear random-effects model

Factorization – Brief Overview

- Latent user factors: (α_i , $\mathbf{u_i}$=(u_{i1},…,u_{ir}))
- Latent movie factors: (β_j , $\mathbf{v_j}$=(v_{j1},….,v_{jr}))

Interaction

$$\alpha_i + \beta_j + u_i' v_j$$

- (N + M)(r+1) parameters → will overfit for moderate values of *r*

- Key technical issue: → *Regularization*
- Usual approach: → Gaussian ZeroMean prior

Existing Zero-Mean Factorization Model

Observation Equation

$$y_{ij} \sim N(m_{ij}, \sigma^2)$$

$$x'_{ij}\boldsymbol{b} + \alpha_i + \beta_j + u'_i v_j$$

State Equation

$$\alpha_i \sim N(0, a_\alpha)$$
$$\beta_j \sim N(0, a_\beta)$$
$$u_i \sim MVN(\mathbf{0}, A_u)$$
$$v_j \sim MVN(\mathbf{0}, A_v)$$

Predict for new dyad: $(x_{ij}^{new})'\hat{\boldsymbol{b}} + \hat{\alpha}_i + \hat{\beta}_j + \hat{u}'_i\hat{v}_j$

Regression-based Factorization Model (RLFM)

- Main idea: Flexible prior, predict factors through regressions
- Seamlessly handles cold-start and warm-start

- Modified state equation to incorporate covariates

$$
\begin{aligned}
\alpha_i &= g_0' w_i + \epsilon_i^{\alpha}, \quad & \epsilon_i^{\alpha} &\sim N(0, a_{\alpha}) \\
\beta_j &= d_0' z_j + \epsilon_j^{\beta}, \quad & \epsilon_j^{\beta} &\sim N(0, a_{\beta}) \\
u_i &= G w_i + \epsilon_i^{u}, \quad & \epsilon_i^{u} &\sim MVN(\mathbf{0}, A_u) \\
v_j &= D z_j + \epsilon_j^{v}, \quad & \epsilon_j^{v} &\sim MVN(\mathbf{0}, A_v)
\end{aligned}
$$

Advantages of RLFM

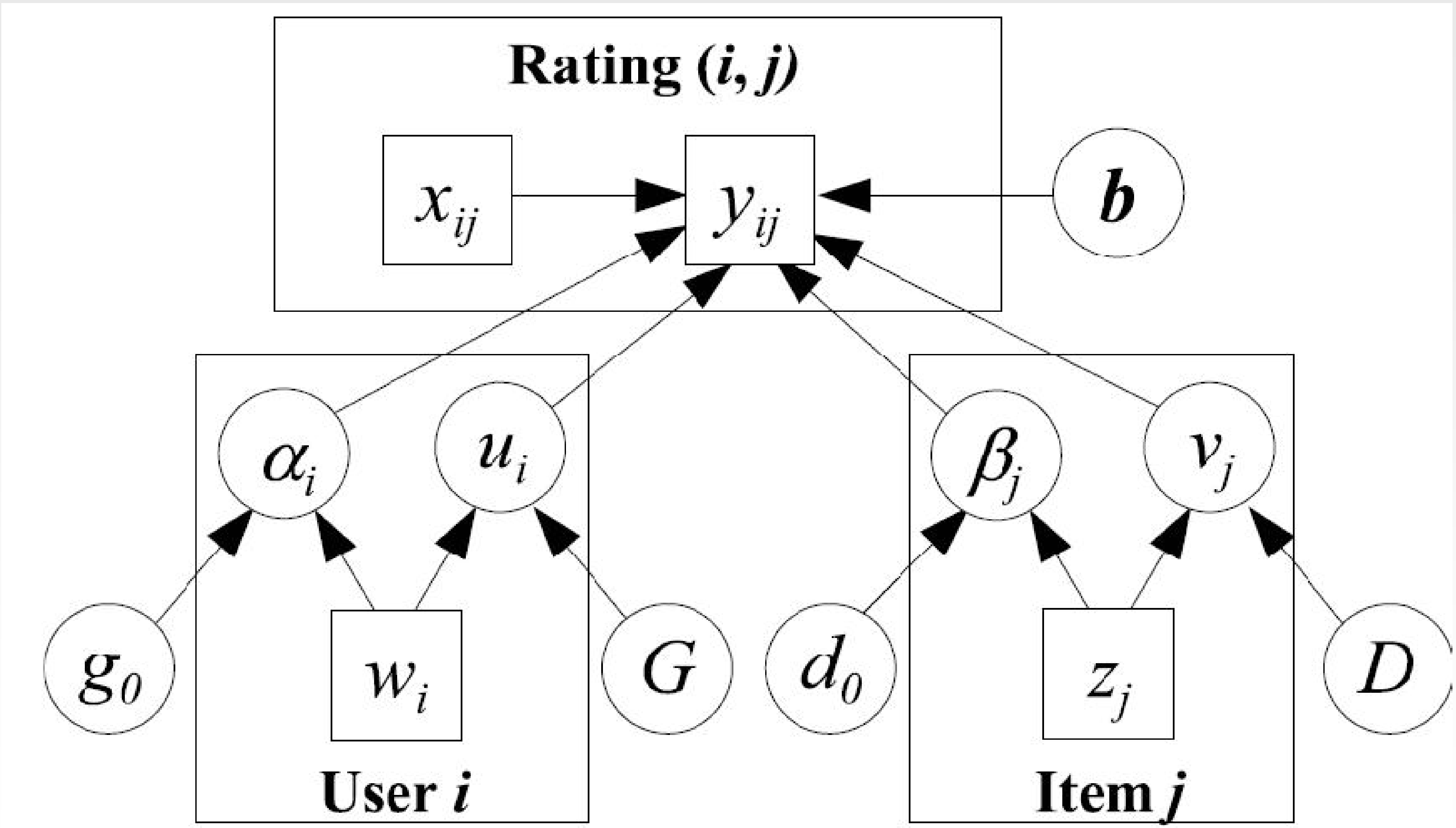
- Better regularization of factors
  - Covariates “shrink” towards a better centroid

- Cold-start: Fallback regression model (**Covariate Only**)

$$y_{ij} \sim N(m_{ij}, \sigma^2)$$

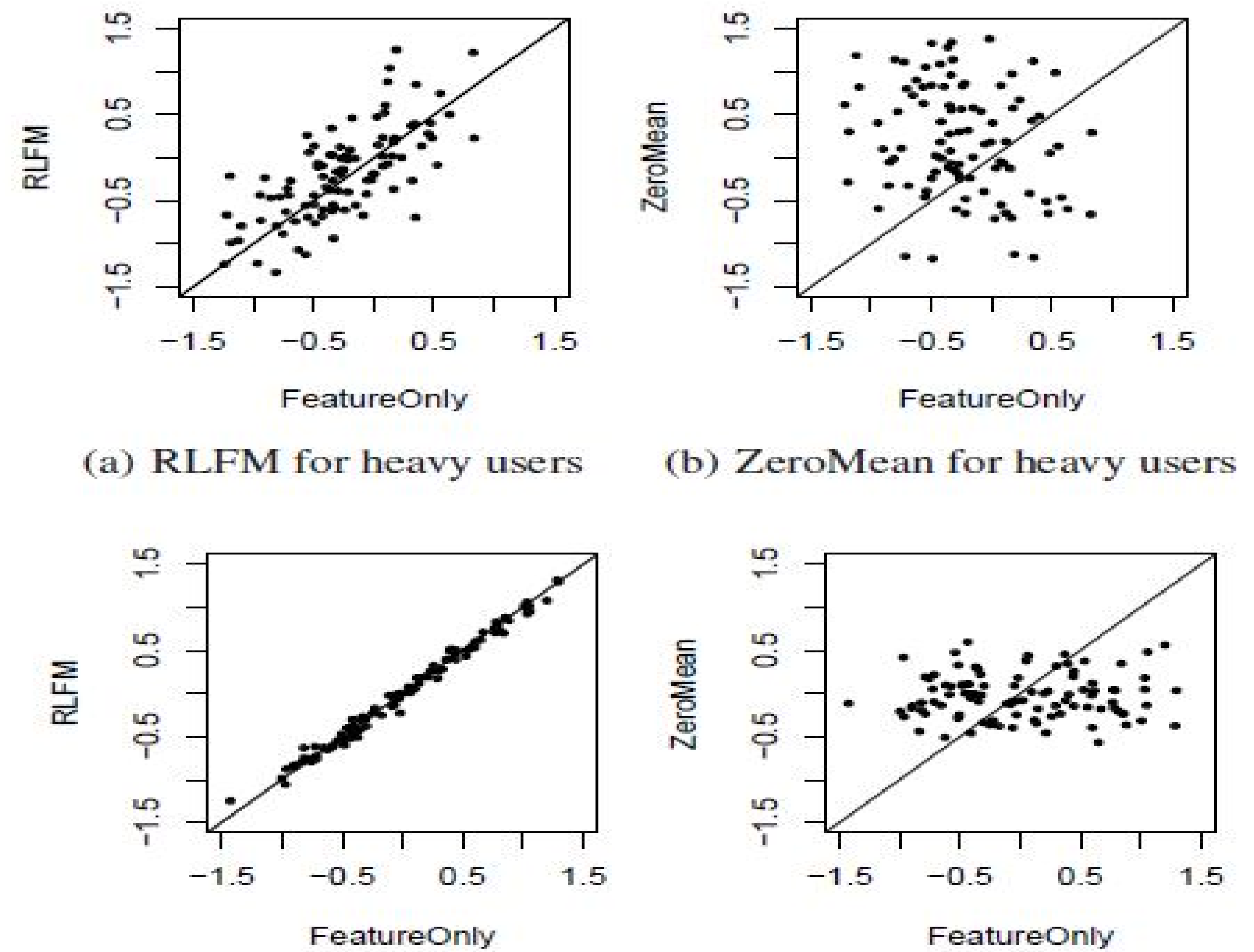
$$m_{ij} = x'_{ij}\boldsymbol{b} + g'_0 w_i + d'_0 z_j + w'_i G' D z_j$$

Graphical representation of the model



Advantages of RLFM illustrated on Yahoo! FP data

Only the first user factor plotted in the comparisons



(a) RLFM for heavy users (b) ZeroMean for heavy users

(c) RLFM for light users (d) ZeroMean for light users

Closer look at induced marginal correlations for gaussian

$$E(y_{ij}) = x'_{ij}\boldsymbol{b} + g'_0 w_i + d'_0 z_j + w'_i G' D z_j$$

$$Var(y_{ij}) = \sigma^2 + a_\alpha + a_\beta + tr(A_u A_v) + z'_j D' A_u D z_j + w'_i G' A_v G w_i$$

$$cov(y_{ij}, y_{ij^*}) = a_\alpha + z'_j D' A_u D z_{j^*}$$

$$cov(y_{ij}, y_{i^*j}) = a_\beta + w'_i G' A_v G w_{i^*}$$

Model Fitting

- Challenging, multi-modal posterior
- Monte-Carlo EM (MCEM)
  - E-step: Sample factors through Gibbs sampling
  - M-step: Estimate regressions through off-the-shelf linear regression routines using sampled factors as response
    - We used t-regression, others like LASSO could be used
- Iterated Conditional Mode (ICM)
  - Replace E-step by CG : conditional modes of factors
  - M-step: Estimate regressions using the modes as response
- Incorporating uncertainty in factor estimates in MCEM helps

Latent dimension r	2	5	10	15
ICM	.9736	.9729	.9799	.9802
MCEM	.9728	.9722	.9714	.9715

Monte Carlo E-step

- Through a vanilla Gibbs sampler (conditionals closed form)

$$\text{Let } o_{ij} = y_{ij} - \alpha_i - \beta_j - x'_{ij}\boldsymbol{b}$$

$$Var[u_i|\text{Rest}] = (A_u^{-1} + \sum_{j \in \mathcal{J}_i} \frac{v_j v'_j}{\sigma^2_{ij}})^{-1}$$

$$E[u_i|\text{Rest}] = Var[u_i|\text{Rest}](A_u^{-1} G w_i + \sum_{j \in \mathcal{J}_i} \frac{o_{ij} v_j}{\sigma^2_{ij}})$$

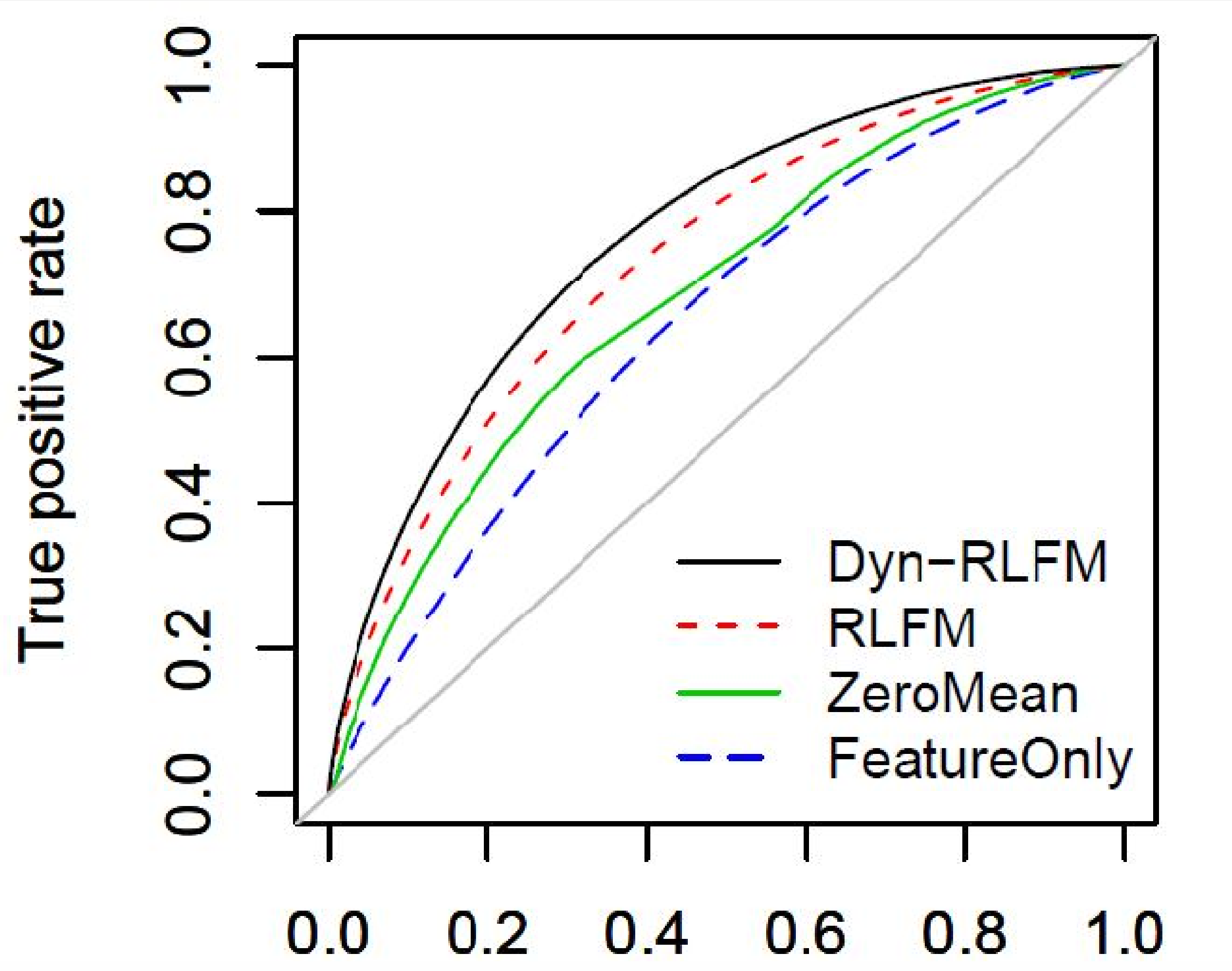
- Other conditionals also Gaussian and closed form
- Conditionals of users (movies) sampled simultaneously
- Small number of samples in early iterations, large numbers in later iterations

Experiment 2: Better handling of Cold-start

- MovieLens-1M; EachMovie
- Training-test split based on timestamp
- Covariates: age, gender, zip1, genre

Model	MovieLens-1M 30%	MovieLens-1M 60%	MovieLens-1M 75%	EachMovie 30%	EachMovie 60%	EachMovie 75%
RLFM	0.9742	0.9528	0.9363	1.281	1.214	1.193
ZeroMean	0.9862	0.9614	0.9422	1.260	1.217	1.197
FeatureOnly	1.0923	1.0914	1.0906	1.277	1.272	1.266
FilterBot	0.9821	0.9648	0.9517	1.300	1.225	1.199
MostPopular	0.9831	0.9744	0.9726	1.300	1.227	1.205
Constant Model	1.118	1.123	1.119	1.306	1.302	1.298

Results on Y! FP data



Online Updates through regression

- Update u's and v's through online regression
- Generalize reduced rank idea

$$s_{ijt} = (G_{p1\times r1}\boldsymbol{x}_{it} + \epsilon_i^u)' B_{r1\times r2}(D_{p2\times r2}\boldsymbol{x}_j + \epsilon_{jt}^v)$$

$$u_{it} = G\boldsymbol{x}_{it} + \epsilon_i^u \qquad v_{jt} = D\boldsymbol{x}_j + \epsilon_{jt}^v$$

$$s_{ijt} = u'_{it} B v_{jt}$$

- Our observations so far : Reduced rank does not improve much if factor regressions are based on good covariates
- Online updates help significantly : (In movie-lens; reduced RMSE from .93 to .86)

Summary

- Simple statistical models coupled with fast sequential learning in near-real time effective for web applications

- Matrix factorization provides state-of-the-art recommendation algorithms with
  - Generalization to include covariates
  - Reduced dimension to facilitate fast sequential learning

Summary: Overall statistical methodology

Initialize

Offline Modeling

Regression, collaborative filtering, latent factor models

Reduce candidate inventory for opportunities through cheap rule

Online Models

Time series

Historical data
Noisy response

Explore/Exploit

Multi-armed bandits

What we did not cover today

- Multi-slot optimization (for a fixed slot design)
  - Correlated response
  - Differential exposure (how to adjust for these statistically?)
    - E.g. good articles shown on high exposure slots, how to adjust for this bias to obtain intrinsic quality score

To Conclude

- Rich set of statistical problems key to web recommender systems; require both mean and uncertainty estimates
- Scale, high dimensionality and noisy data challenges
- Good news:
  - Statisticians can design experiments to collect data
  - If these problems excite you, Y! one of the best places
  - Rich set of applications, large and global traffic.
    - (Y! front page is the most visited content page on the planet)