Recommender Problems for Web Applications

Deepak Agarwal

Yahoo! Labs

Bay Area ASA Chapter

Raleigh, Dec 16th,2009

MOTIVATING Applications

Match making: Select Content for User visits

Ultimate Goal: Maximize User Satisfaction, Engagement

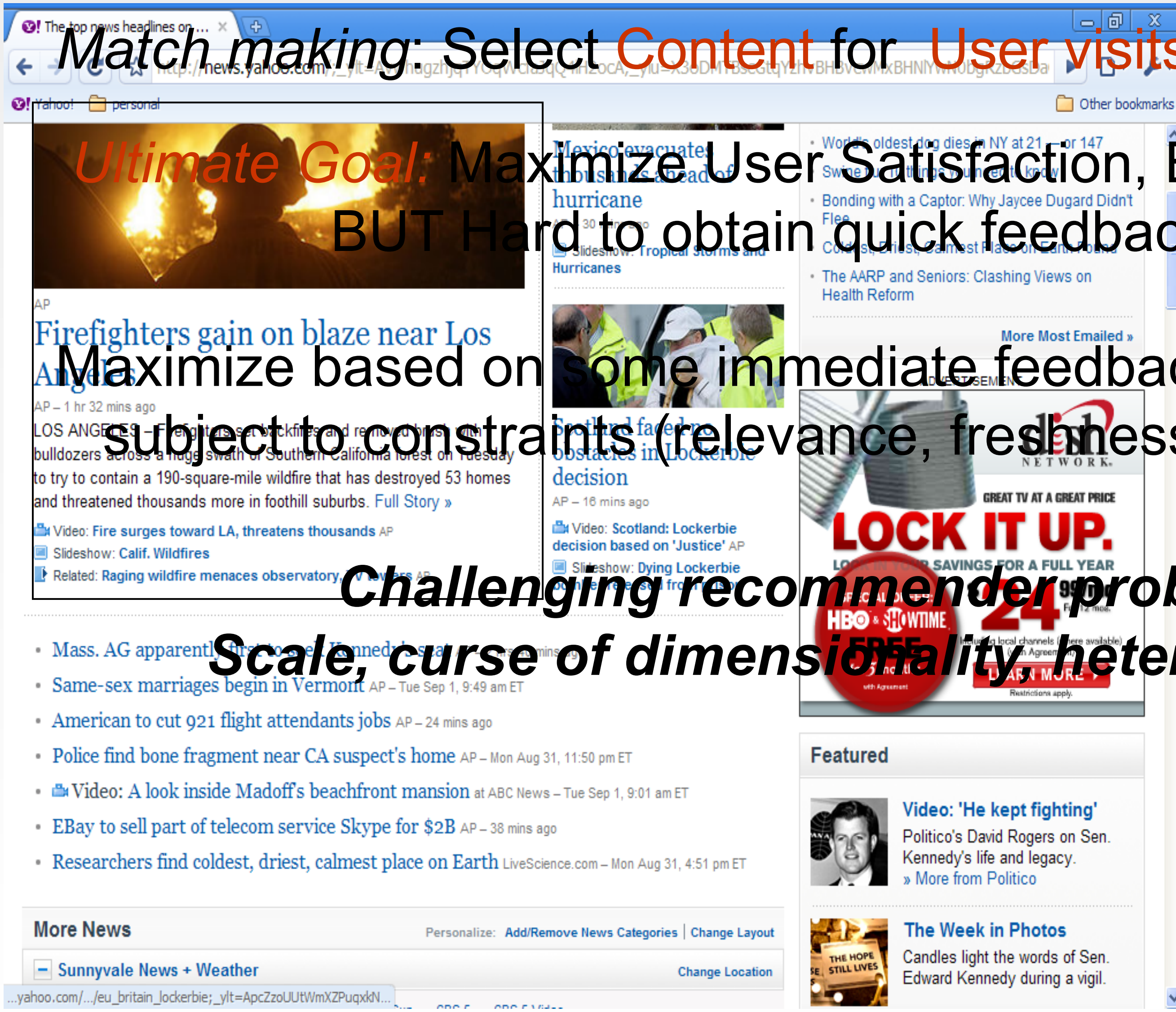
BUT Hard to obtain quick feedback:

Maximize based on some immediate feedback (click rate)

subject to constraints (relevance, freshness, diversity)

Challenging recommender problem

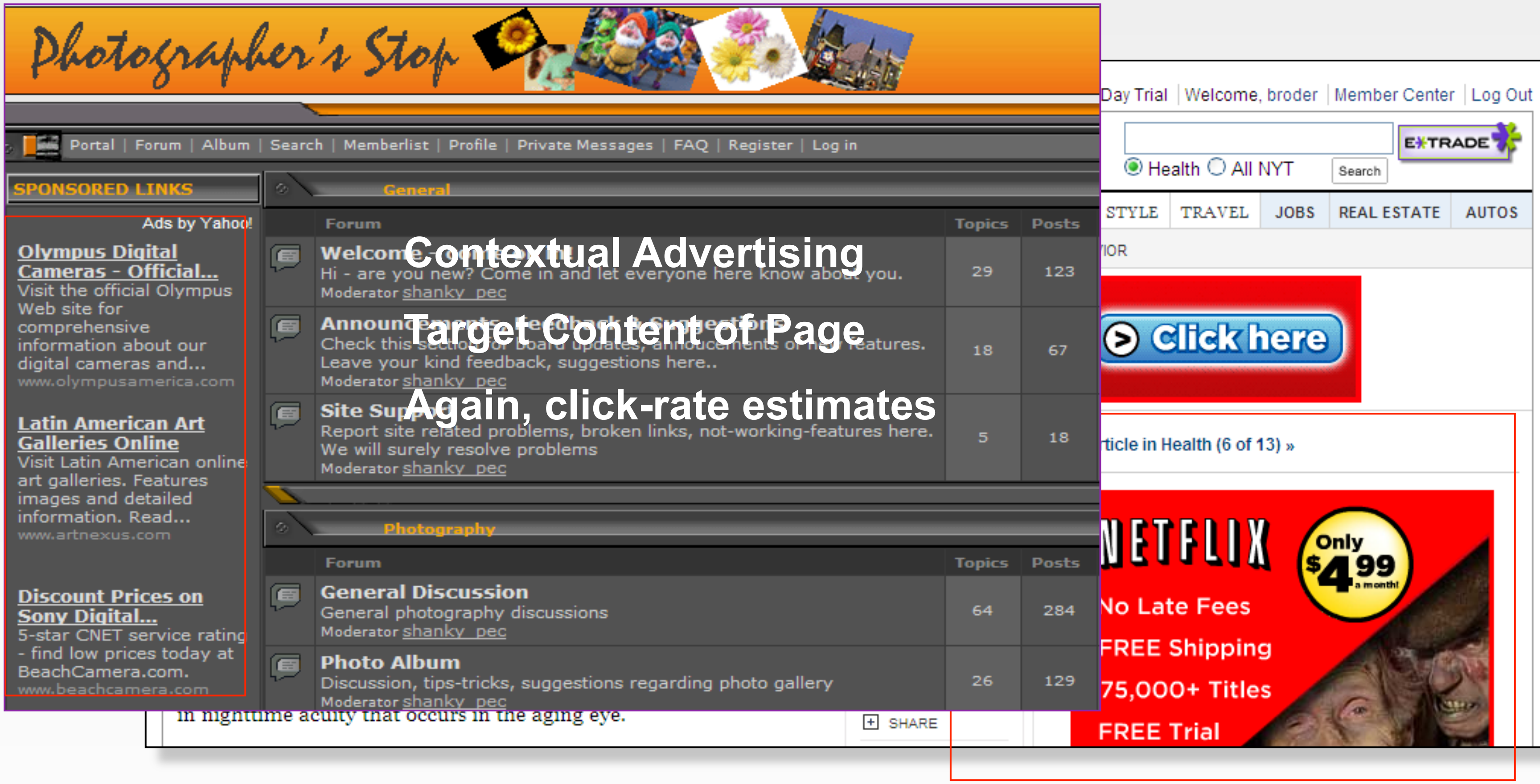
Scale, curse of dimensionality, heterogeneity



Main Collaborators

- Bee-Chung Chen
- Pradheep Elango
- Raghu Ramakrishnan

Advertising: Not the focus of this talk



Contextual Advertising

Target Content of Page

Again, click-rate estimates

Problem discussed in this talk

- We look at content optimization.
- *Recommend* items (articles) to users visiting a webpage.
- Objective: Maximize click-rates
  - (other utilities like engagement are also of interest but not considered here)

- Simplifying assumption
  - Consider recommending items on a **single "slot"**.
  - Assume ***no interaction*** with other slots.

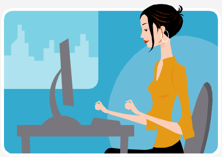
    (this is generally not true)

DATA

INVENTORY

Time t

Algorithm selects Item j covariates $\mathbf{x}_j$



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

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

click-rate p_{ijt}

Model: $(\mathbf{Y}_t; \mathbf{P}_t)$; t=1,2,….

Statistical Problem

- Decision problem: Starting at time 0,
  - Display items from inventory, for each user visit in [0,T) such that we maximize overall expected clicks

$$\boldsymbol{a} = \operatorname{argmax}_{\boldsymbol{j}} \sum_{i:t \in [0,T)} p_{i,j,t}$$

- Items selected by the algorithm for visits in [0,T)

- We have access to all historic data
  - except most recent observations
    - (latency depends on engineering constraints)

Visits and Inventory

- User visits
  - not known in advance, maybe able to forecast

- Item inventory:
  - Decided by domain expert
    - (editors, experts setting crawling policies)
  - Interesting scenario
    - Statistical methods to provide input on the inventory composition (will not be covered in this talk)

Greedy Solution

- Select item with maximum estimated posterior mean for each visit

$$a = \text{argmax}_j \hat{p}_{i,j,t}$$

- Estimated click-rate : all data in $[0, t\text{-}t_0]$
- Does not incorporate uncertainty in estimates
  – May be sub-optimal for our sequential problem

Sequential design, Explore/Exploit

- Select items **now,** to maximize expected overall click rates in (0,T] (adjust for uncertainty in estimates)

$$\text{Construct design } \pi = (a_0, \ldots, a_n, \ldots.) \text{ maximize}$$

$$V(\mathbf{P}_0) = r(\mathbf{P}_0, a_0) + E(\sum_{j=1,..,T} r(\mathbf{P}_i, a_j)) =$$

$$r(\mathbf{P}_0, a_0) + \sum_{i=1}^{T} \int r(\mathbf{P}, a_j) Trans(\mathbf{P} \mid \mathbf{P}_{i-1}, a_{i-1}) d\mu(\mathbf{P})$$

- Hard problem (MDP), but studied in multi-armed bandit literature

Degree of Personalization

$$\boldsymbol{a} = \operatorname{argmax}_{\boldsymbol{j}} \sum_{j:t\in[0,T)} p_{j,t}$$

Most Popular

Breaking news,
Broad appeal
inventory pool

$$\boldsymbol{a} = \operatorname{argmax}_{\boldsymbol{j}} \sum_{i:t\in[0,T)} p_{\boldsymbol{x}_i,j,t}$$

Most Popular per
User segment

Larger, more
diverse inventory

$$\boldsymbol{a} = \operatorname{argmax}_{\boldsymbol{j}} \sum_{i:t\in[0,T)} p_{i,j,t}$$

Per user

Larger, more
diverse inventory
and engaged
users

Rest of the talk

- Most Popular: Yahoo! front Page (www.yahoo.com)
  - Models, Sequential design
- Personalization per user segment
  - Models, Sequential design
- Personalization per user
  - Models, Sequential design

Illustrative Application: Today Module on www.yahoo.com

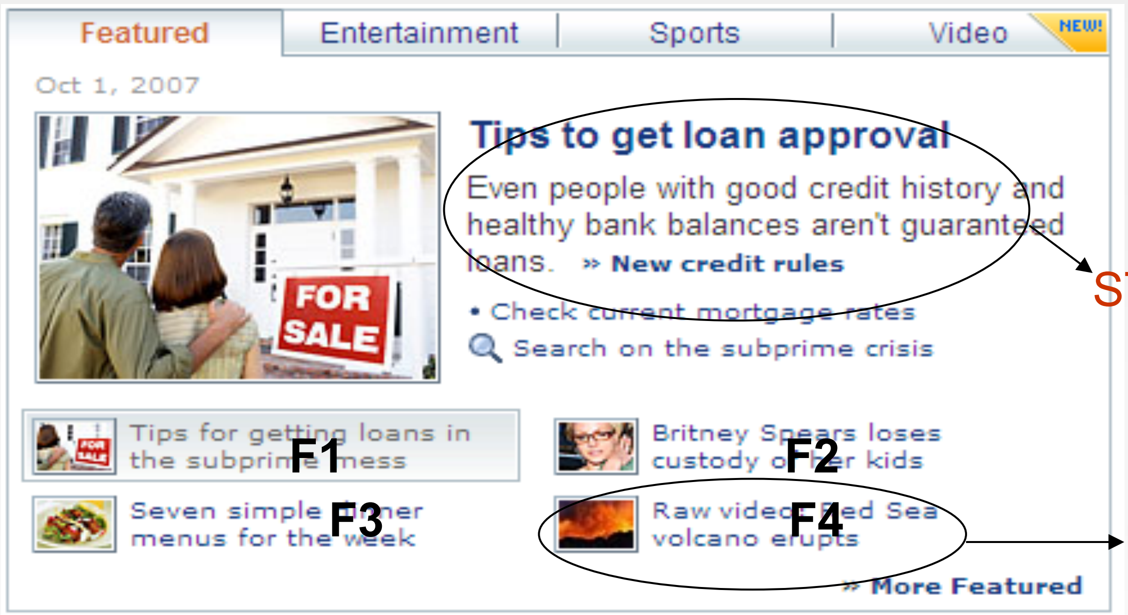
Today Module

Defaults to the Featured Tab

- Today Module is the top-center part
- Four tabs: **Featured**, Entertainment, Sports, and Video
- **Featured**: displays content from all categories
- **Today Module**: Routes traffic to other Y! pages, increases user engagement

Some More Background…

Featured Tab in Detail



Four articles on F1,F2,F3,F4

F1 article as *story by* default

STORY POSITION

FOOTER POSITION

- Footer click → corresponding article as story
- Click rates (CTR): Story clicks per display (maximize this)
- F1 → max exposure, large fraction of story clicks

Content Programming for Today Module

- Editorial → ensures high content quality
- Preserves editorial “Voice” (typical mix of content)

Complete automation: Scalable, but may hurt user experience

- Article pool on Today Module : *dynamic* and small
  - New ones *pushed*, old ones taken out
    - Few tens of unique articles per day
  - Why? Keep up with novel articles and remove fading ones
  - Typically, articles have short lifetimes (6-24 hours)

DATA Characteristics

- Large volumes: Several hundred million visits per day
  - Estimate per article CTR at 5-minute resolution
- Two data sources
  - **Serving bucket**: shows current best until we find better one
  - **Small Random bucket**: Randomly selects 4 for each user visit

Advantages of Randomization

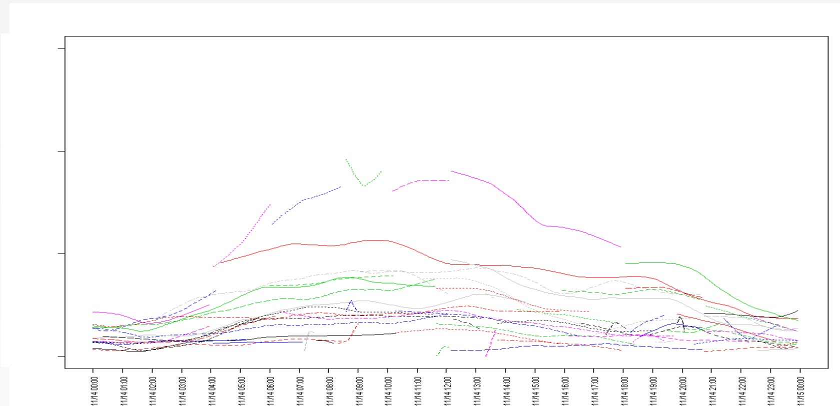
Item **COLD START (estimation for new items)**

Unbiased data to help study user-content interaction

Previous studies (Wu et al.) use biased data
Details (Conference paper Agarwal,Chen,Elango WWW’09)

Challenges

- Dynamic Content, short lifetimes (quick reaction key)
- Temporal variation in user visit composition
  - Implies temporal variation in click-rates
- User fatigue due to repeat exposure, Positional effects
- Cold start (new articles)



- Tracking based on popularity
  - Time series tracking models
  - Cold start through Explore/Exploit (sequential design) strategies
    - Randomization is one way but we can do better

Time series tracking for an item

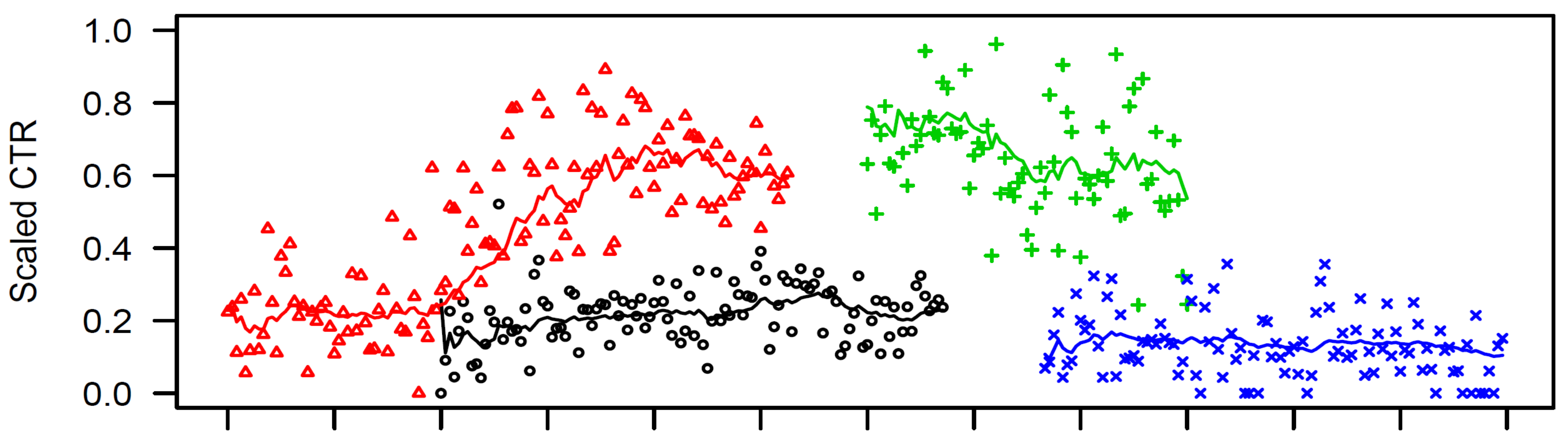
- Dynamic Gamma-Poisson with multiplicative state evolution

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

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

High CTR items more adaptive

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



Explore/Exploit: some basics

High level overview

- *Two Items*: Item 1 CTR= 2/100 ; Item 2 CTR= 25/1000
  - *Greedy:* Show Item 2 to all; not a good idea
  - Item 1 CTR estimate noisy; item could be potentially better
    - Invest in Item 1 for better overall performance on average
  - Show both Item 1 and Item 2
    - Optimal choice of design is the Explore/Exploit problem

- Classical solutions: Multi-armed bandit
  - Gittins' approach
    - maximize discounted cumulative reward)
  - Lai's approach:
    - Upper confidence bound schemes (minimize regret from best)

Background: Bandits

p_1

p_2

p_3

Bandit "arms"

(unknown payoff probabilities)

- "Pulling" arm i yields a reward:
  - reward = 1 with probability p_i (success)
  - reward = 0 otherwise (failure)

Background: Bandits

p_1

p_2

p_3

Bandit "arms"

(unknown payoff probabilities)

- Goal: Pull arms sequentially to maximize the total expected reward; achieve the best trade-off between
  - Exploit: Use estimates of payoff probabilities $\{p_i\}$
  - Explore: Don't be misguided by uncertainty in estimates; play arms that are potentially good.

Background: Bandits

- ***bandit policy*** : sequential scheme to play arms
- *Regret of a policy* = Expected loss relative to best hypothetical policy (plays the best arm at all times)
  - Of course, the best arm is not known
  - Hence, the regret is the price of exploration
  - Low regret implies quick convergence to the best
- Large number of policies to choose from
  - What is the optimal policy?
    - Difficult problem, took several years to find the solution

Overview continued

- Discounted reward case: special case of MDP
  - Items not shown do not change state
- Gittin's landmark result (Gittin's index policy)
  - K-dim optimization can be solved through K one dim optim
  - Each 1-d problem computes a stopping time
    - Still difficult to compute these stopping times

- Upper Confidence Bound (UCB) policies (Lai, Auer)
  - Use an optimistic estimate as arm priority (e.g. mean + 2*sd)
  - Logarithmic bound on regret, several policies available
    - UCB

$$\hat{p}_i + \sqrt{\frac{2\log(n)}{n_i}}$$

Key differences from classical settings (e.g. UCB1, EXP3,..)

- Dynamic content pool
  - Restless bandits (Whittle, 88)
- Non-stationary CTR
  - Adversarial bandits (Auer, 95)
- Batched serving plan
  - System constraints, click-view latency
  - New, no prior work in bandit literature

Bayesian solution in the 2 item scenario

- 2 step look-ahead at each time interval (supply known)

- Two items
  - **Certain** item with CTR q_0 and q_1, Var(q_0) = Var(q_1) = 0
  - **Uncertain** item *1* with CTR p_0 ~ Gamma(α, γ)

Interval 0: What fraction *X* of views to give to item *i*

Let c ~ Poisson($p_0(XN_0)$) : clicks on item *i, interval 0.*

Prior gets updated to posterior: Gamma(α+c,γ+XN_0)

Interval 1: Always give all the views to the better item

Give all the views to item *1* iff E[$p1$ | c, x] > $q1$

Find *X* that maximizes the expected total number of clicks

More details on 2 items, 2 steps

- *Expected total number of clicks*

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

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

E[#clicks] if we always show the certain item

Gain(x, q_0, q_1)
Additional gain from exploration

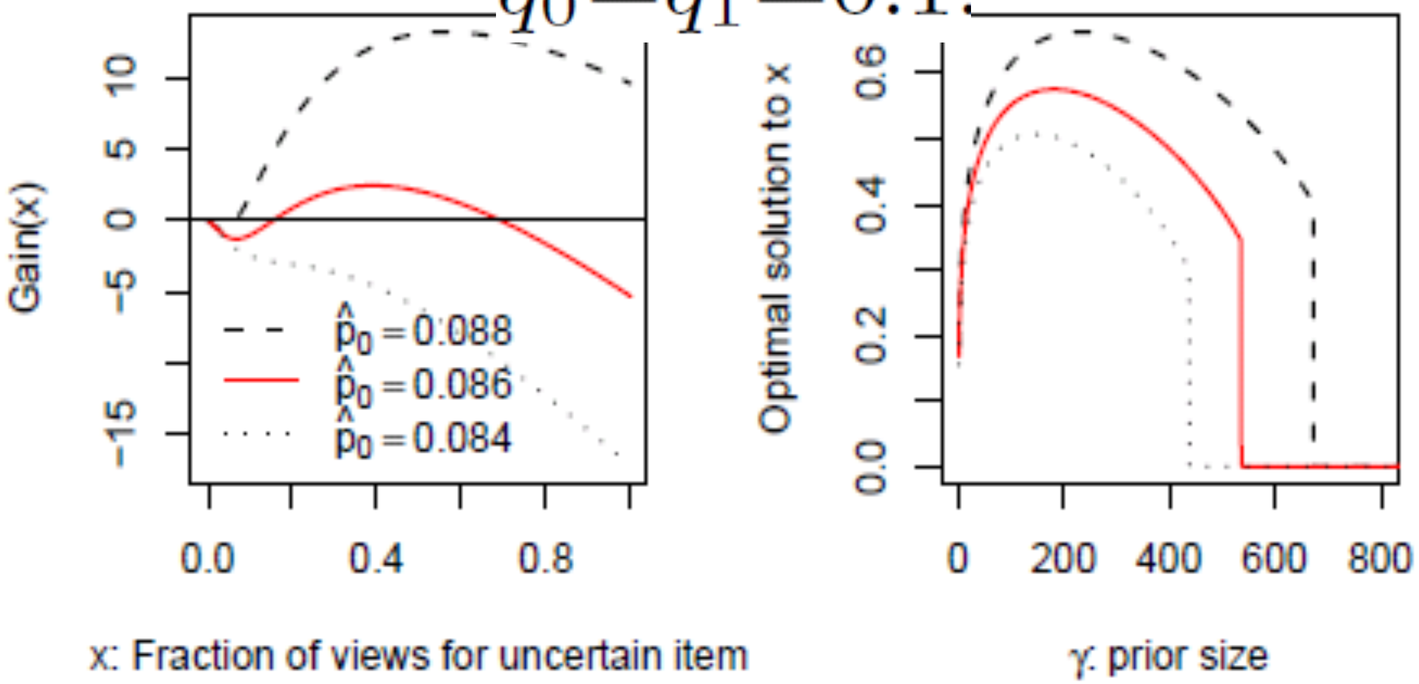
Goal: argmax_x Gain(x, q_0, q_1)

- *Expected max: Normal approximation*

$$Gain(x, \boldsymbol{\theta}_0, q_0, q_1, N_0, N_1) \approx N_0 x(\hat{p}_0 - q_0) + N_1\left[\sigma_1(x)\phi\left(\frac{q_1 - \hat{p}_0}{\sigma_1(x)}\right) + \left(1 - \Phi\left(\frac{q_1 - \hat{p}_0}{\sigma_1(x)}\right)\right)(\hat{p}_0 - q_1)\right]$$

Example for Gain function

$q_0 = q_1 = 0.1.$



(a) Gain function (b) Optimal solution

K items, 2 intervals

- Expected total clicks

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

Our goal is to find

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

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

Lagrange relaxation (Whittle)

$$R^+(\boldsymbol{\theta}_0, N_0, N_1) = \max_{0 \le \mathbf{x} \le 1} R(\mathbf{x}, \boldsymbol{\theta}_0, N_0, N_1)$$
$$\Sigma_i \; x_{i0} = 1 \text{ and } E_{\boldsymbol{\theta}_1}\left[\Sigma_i \; x_{i1}(\boldsymbol{\theta}_1)\right] = 1.$$

$$V(\boldsymbol{\theta}_0, q_0, q_1, N_0, N_1) = \max_{0 \le \mathbf{x} \le 1} \{R(\mathbf{x}, \boldsymbol{\theta}_0, N_0, N_1)$$
$$- q_0 N_0(\Sigma_i \; x_{i0} - 1) - q_1 N_1(E[\Sigma_i \; x_{i0}] - 1)\}$$

Efficient computation

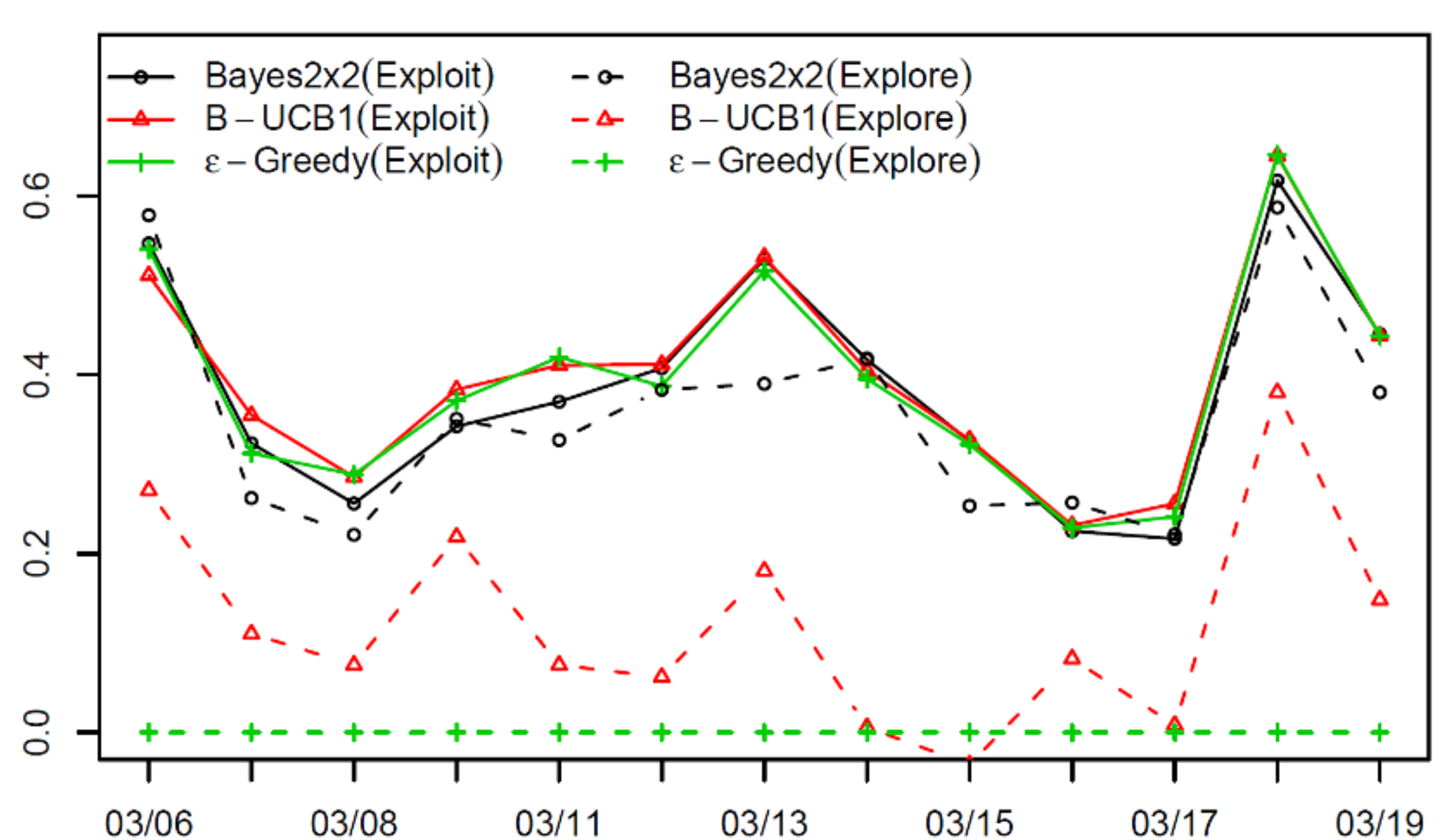
Separability: $V(\boldsymbol{\theta}_0, q_0, q_1, N_0, N_1) =$

$$\sum_i \left(\max_{0 \le x_{i0} \le 1} Gain(x_{i0}, \boldsymbol{\theta}_{i0}, q_0, q_1, N_0, N_1) \right) + q_0 N_0 + q_1 N_1$$

Convexity: $V(\boldsymbol{\theta}_0, q_0, q_1, N_0, N_1)$ $convex\ in\ (q_0, q_1)$.

Test on Live Traffic

Constraint: 15% explore: 85% has to serve based on Greedy



Summary & Discussion

- Main message: Bayesian schemes that “exploit” good modeling assumptions far superior than adaptations of existing explore/ exploit schemes
  - Good news! Significant effort spend on building models BUT uncertainty estimates should also accompany predictions
- Bounds on schemes are generally loose and do not accurately reflect impact in practice. We strongly argue for a Bayesian approach incorporating modeling assumptions
- If no modeling has been done, one could use adaptations of existing schemes as a good starting point
- Publications (Agarwal, Chen and Elango)
  - Tracking (Online models for content optimization NIPS 08)
  - Explore/Exploit schemes for web applications, ICDM 09

Personalization per-user segment

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

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

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

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

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

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

Reduced Rank Regression (Anderson, 1951)

- *N x p* response matrix
- Each row has a covariate vector $\mathbf{x}_i$

- *p* regressions, each of dim *q*: $(x_i' \boldsymbol{v}_1, x_i' \boldsymbol{v}_2, \ldots, x_i' \boldsymbol{v}_p)$
  - $\boldsymbol{V}_{q\,xp}$: too many parameters
  - Reduced rank: $\boldsymbol{V}^T = \mathbf{B}_{p \times r}\ \boldsymbol{\Theta}_{r \times q}$ (r << q; rank reduction)

- Generalization to categorical data
  - Took a long time, happened in around '00 (Hastie et al)

Reduced Rank for our cold-start problem

- Generalize reduced rank for large incomplete matrix

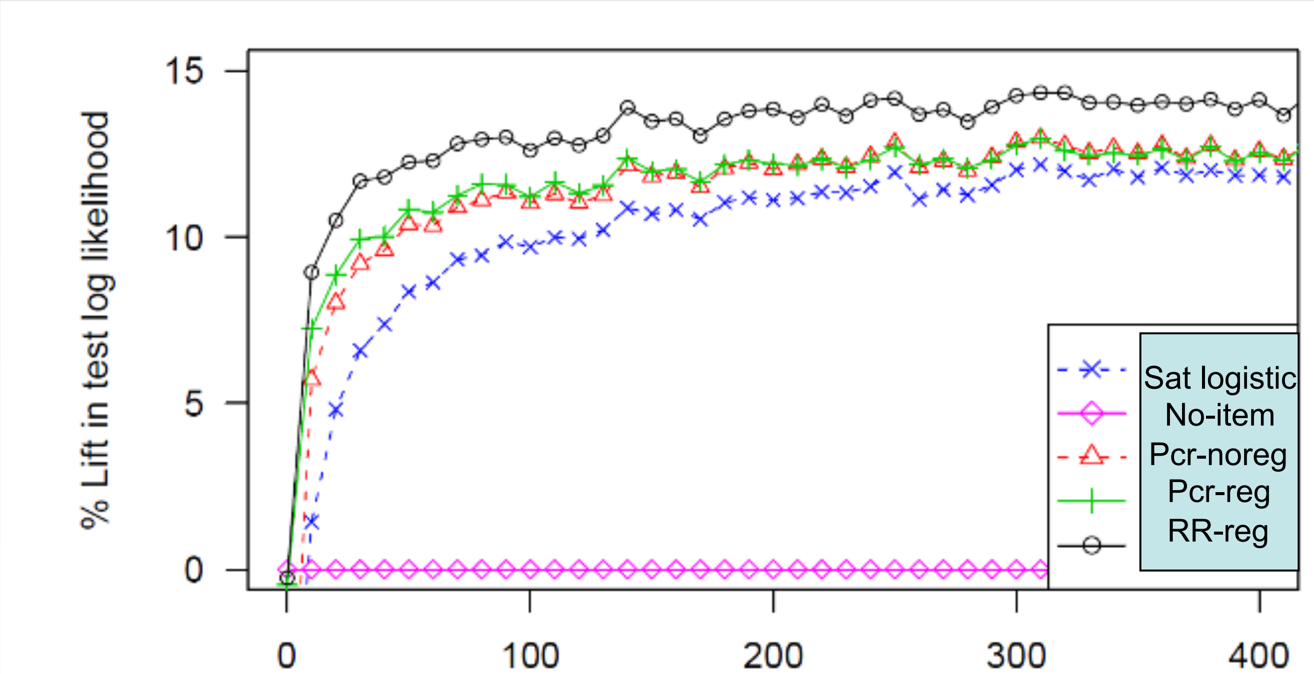
$$s_{ijt} = \boldsymbol{x}'_{it}\boldsymbol{A}\boldsymbol{x}_j + \boldsymbol{x}'_{it}B\boldsymbol{\theta}_j$$

Low dimension (5-10),

B estimated retrospective data

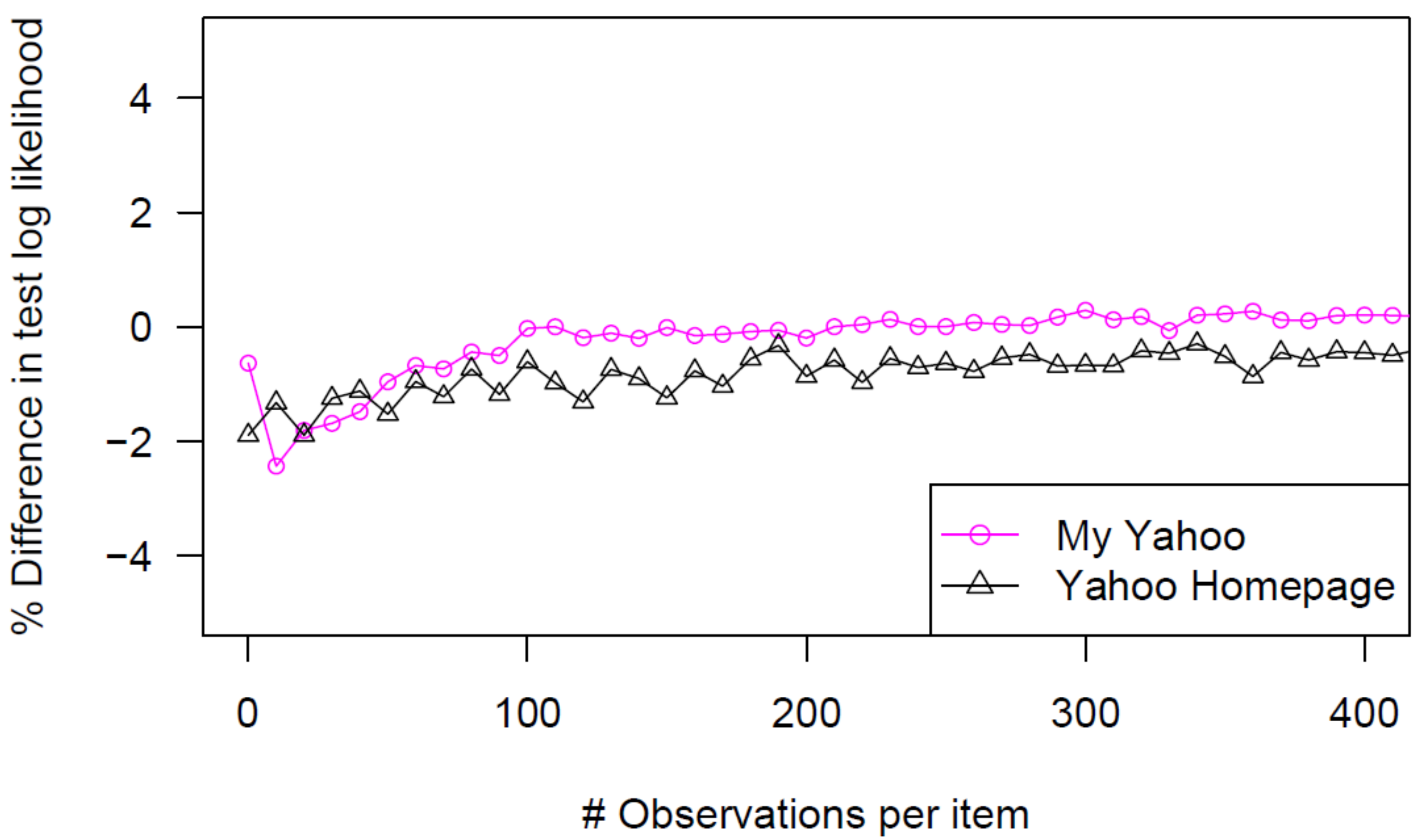
- Application different than in classical reduced rank literature
  - Cold-start problem in recommender problems

Results for Online Reduced Rank regression



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

Reduced Rank initialization with and without item covariates



Per user, per item: Bilinear factor models

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

User i latent factors

$$u_{it} = G\boldsymbol{x}_{it} + \epsilon_i^u$$

Item j latent factor

$$v_{jt} = D\boldsymbol{x}_j + \epsilon_{jt}^v$$

$$s_{ijt} = u_{it}' B v_{jt}$$

$r1 = r2 = r, B = I$ studied extensively (Netflix)

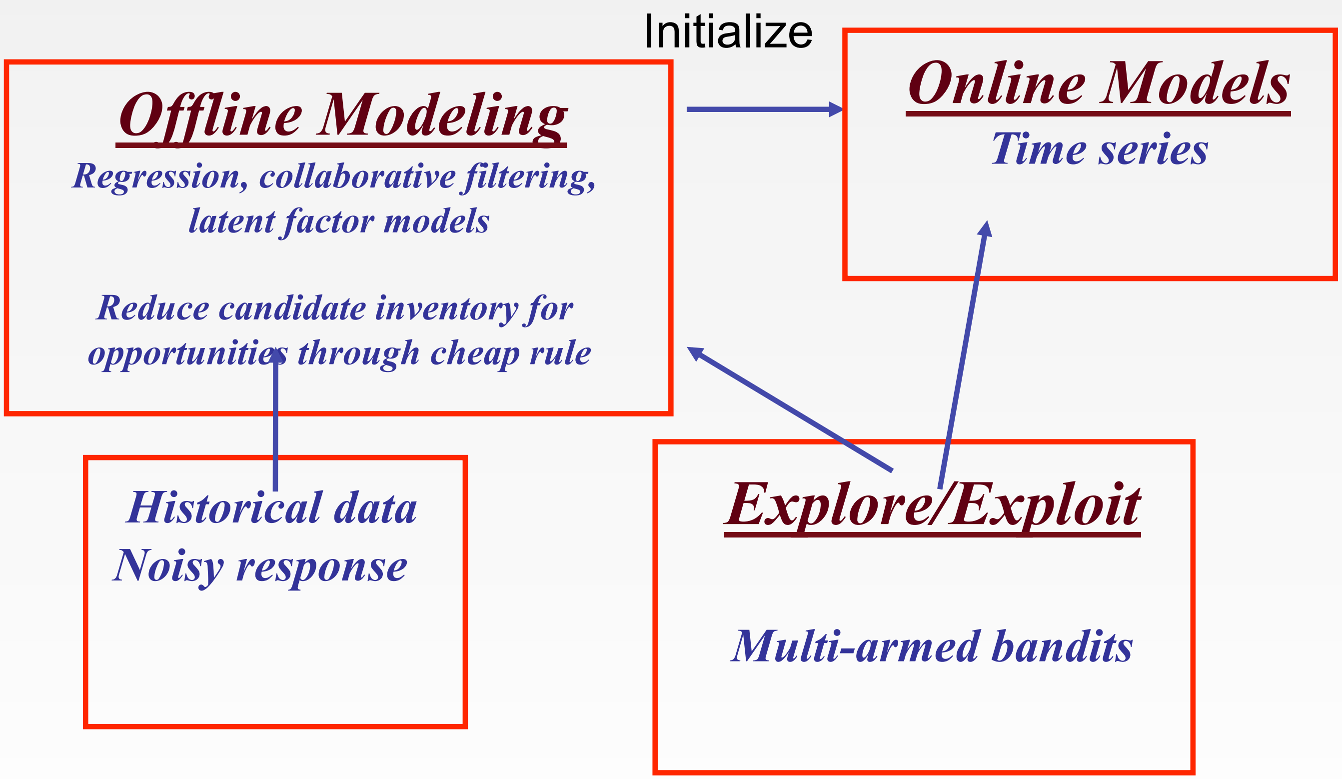
RLFM: Regression based factorization (Agarwal & Chen, KDD 2009)

$$
\begin{aligned}
y_{ij} &\sim N(m_{ij}, \sigma^2), \text{or} \qquad \text{(Gaussian)} \\
y_{ij} &\sim \text{Bernoulli}(m_{ij}) \qquad \text{(Logistic)} \\
l(m_{ij}) &= x'_{ij}\boldsymbol{b} + \alpha_i + \beta_j + u'_i v_j
\end{aligned}
$$

$$
\begin{aligned}
\alpha_i &= g'_0 w_i + \epsilon_i^\alpha, & \epsilon_i^\alpha &\sim N(0, a_\alpha) \\
\beta_j &= d'_0 z_j + \epsilon_j^\beta, & \epsilon_j^\beta &\sim N(0, a_\beta) \\
u_i &= G w_i + \epsilon_i^u, & \epsilon_i^u &\sim MVN(\mathbf{0}, A_u) \\
v_j &= D z_j + \epsilon_j^v, & \epsilon_j^v &\sim MVN(\mathbf{0}, A_v)
\end{aligned}
$$

Model fitting: MCEM

Summary: Overall statistical methodology



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
- Sequential tests
  - How to choose between several sequential schemes?
    - Can we develop model selection criteria for our sequential problem?
    - If not, efficient sequential tests to help us conduct a large number of experiments using small amount of traffic

What we did not cover today --continued

- Statisticians playing a role in deciding on future inventory
  - E.g. Front page application
    - Who visits during 10-11 on Tuesday, what do they like? Do we have enough inventory?
    - Given finite resources and a knowledge of user-to-item affinities, how to manage inventory to maximize overall clicks

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
  - Sequential designs attractive
    - Lose some power but we maximize expected yield

- 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)