# Response prediction using collaborative filtering with hierarchies and side-information

Aditya Krishna Menon[1] Krishna-Prasad Chitrapura[2] Sachin Garg[2] Deepak Agarwal[3] Nagaraj Kota[2]

[1]UC San Diego

[2]Yahoo! Labs Bangalore

[3]Yahoo! Research Santa Clara

KDD '11, August 22, 2011

# Outline

1. Background: response prediction
2. A latent feature approach to response prediction
3. Combining latent and explicit features
4. Exploiting hierarchical information
5. Experimental results

# The response prediction problem

- Basic workflow in computational advertising:

  Content publisher (e.g. Yahoo!) receives bids from advertisers:


- Amount paid on some action e.g. ad is clicked, conversion, ...

# The response prediction problem

- Basic workflow in computational advertising:

  Compute expected revenue using clickthrough rate (CTR):

$\times \Pr[\text{Click}|\text{Display}]$ $\quad \times \Pr[\text{Click}|\text{Display}]$ $\quad \times \Pr[\text{Click}|\text{Display}]$

- Assuming pay-per-click model

# The response prediction problem

- Basic workflow in computational advertising:

Ads are sort by expected revenue, best ad is chosen


$\times \Pr[\text{Click}|\text{Display}]$ $\times \Pr[\text{Click}|\text{Display}]$ $\times \Pr[\text{Click}|\text{Display}]$

- **Response prediction**: Estimate the CTR for each candidate ad

# Approaches to estimating the CTR

- Maximum likelihood estimate (MLE) is straightforward:

$$\widehat{\Pr}[\text{Click}|\text{Display}; (\text{Page}, \text{Ad})] = \frac{\#\text{ of clicks in historical data}}{\#\text{ of displays in historical data}}$$

  - Few displays $\rightarrow$ too noisy, not displayed $\rightarrow$ undefined
  - Can apply statistical smoothing [Agarwal et al., 2009]
- Logistic regression on page and ad features [Richardson et al., 2007]
- LMMH [Agarwal et al., 2010], a log-linear model with hierarchical corrections, is state-of-the-art

# This work

- We take a collaborative filtering approach to response prediction
  - "Recommending" ads to pages based on past history
  - Learns latent features for pages and ads
- Key ingredient is exploiting hierarchical structure
  - Ties together pages and ads in latent space
  - Overcomes extreme sparsity of datasets
- Experimental results demonstrate state-of-the-art performance

# Outline

# Response prediction as matrix completion

- Response prediction has a natural interpretation as matrix completion:

| | pepsi | | SONY |
|---|---|---|---|
| Y | 0.5 | 1.0 | ? |
| CNN | ? | 0.5 | 0.25 |
| f | 0.0 | 1.0 | 1.0 |

  - Cells are historical CTRs of ads on pages; many cells “missing”
  - Wish to fill in missing entries, but also smoothen existing ones

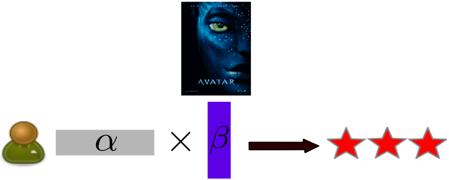
# Connection to movie recommendation

- This is reminiscent of the movie recommendation problem:


  - Cells are ratings of movies by users; many cells “missing”
  - Very active research area following Netflix prize

# Recommending movies with latent features

- A popular approach is to learn latent features from the data:
  - User $i$ represented by $\alpha_i \in \mathbb{R}^k$, movie $j$ by $\beta_j \in \mathbb{R}^k$
  - Ratings modelled as (user, movie) affinity in this latent space




- For a matrix $X$ with observed cells $\mathcal{O}$, we optimize

$$\min_{\alpha,\beta} \sum_{(i,j)\in\mathcal{O}} \ell(X_{ij}, \alpha_i^T \beta_j) + \Omega(\alpha, \beta).$$

  - Loss $\ell =$ square-loss, hinge-loss, ...
  - Regularizer $\Omega = \ell_2$ penalization typically

# Why try latent features for response prediction?

- State-of-the-art method for movie recommendation
  - Reason to think it can be successful for response prediction also
- Data is allowed to "speak for itself"
  - Historical information mined to determine influential factors
- Flexible, analogues to supervised learning
  - Easy to incorporate explicit features, domain knowledge

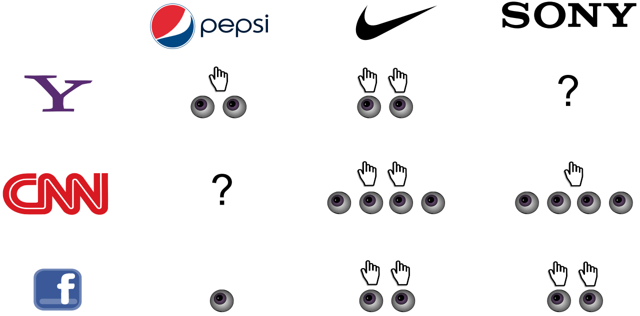
# Response prediction via latent features - I

- Modelling raw CTR matrix with latent features is not sensible
  - Ignores the confidence in the individual cells
- Instead, split each cell into # of displays and # of clicks:

| | pepsi | | SONY |
|---|---|---|---|
| Y | 0.5 | 1.0 | ? |
| CNN | ? | 0.5 | 0.25 |
| f | 0.0 | 1.0 | 1.0 |

# Response prediction via latent features - I

- Modelling raw CTR matrix with latent features is not sensible
  - Ignores the confidence in the individual cells
- Instead, split each cell into # of displays and # of clicks:




  - Click = +ve example, non-click = -ve example
  - Now focus on modelling entries in each cell

# Response prediction via latent features - II

- Important to learn meaningful probabilities
  - Discrimination of click versus not-click is insufficient
- For page $p$ and ad $a$, we may use a sigmoidal model for the individual CTRs:

$$\hat{P}_{pa} = \Pr[\text{Click}|\text{Display}; (p, a)] = \frac{\exp(\alpha_p^T \beta_a)}{1 + \exp(\alpha_p^T \beta_a)}$$

  - $\alpha_p, \beta_a \in \mathbb{R}^k$ are the latent feature vectors for pages and ads
  - Corresponds to a logistic loss function [Agarwal and Chen, 2009, Menon and Elkan, 2010, Yang et al., 2011]

# Confidence weighted objective

- We use the sigmoidal model on each cell entry
  - Treats them as independent training examples
- Now maximize conditional log-likelihood:

$$\min_{\alpha,\beta} - \sum_{(p,a)\in\mathcal{O}} C_{pa} \log \hat{P}_{pa}(\alpha,\beta) + (D_{pa} - C_{pa}) \log(1 - \hat{P}_{pa}(\alpha,\beta)) +$$
$$\frac{\lambda_\alpha}{2}||\alpha||_F^2 + \frac{\lambda_\beta}{2}||\beta||_F^2$$

where $C = \#$ of clicks, $D = \#$ of displays

- Terms in objective are confidence weighted
- Estimates will be meaningful probabilities

# Outline

# Incorporating explicit features

- We'd like latent features to complement, rather than replace, explicit features
  - For response prediction, explicit features quite predictive
  - Makes sense to use this information
- Incorporate features $s_{pa} \in \mathbb{R}^d$ for the (page, ad) pair $(p, a)$ via

$$\begin{aligned}\hat{P}_{pa} &= \sigma(w^T s_{pa} + \alpha_p^T \beta_a) \\ &= \sigma(\left[w; 1\right]^T \left[s_{pa}; \alpha_p^T \beta_a\right])\end{aligned}$$

- Alternating optimization of $(\alpha, \beta)$ and $w$ works well
  - Predictions from factorization $\rightarrow$ additional features into logistic regression

# An issue of confidence

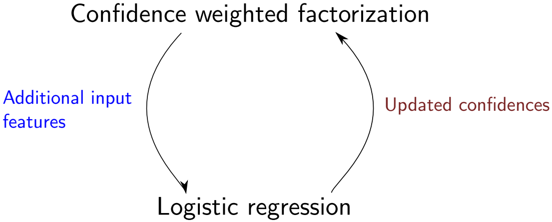
- Rewrite objective as

$$\min_{\alpha,\beta,w} - \sum_{(p,a)\in\mathcal{O}} D_{pa} \Big( M_{pa} \log \hat{P}_{pa}(\alpha,\beta,w) + (1 - M_{pa}) \log(1 - \hat{P}_{pa}(\alpha,\beta,w)) \Big)$$
$$\frac{\lambda_\alpha}{2}||\alpha||_F^2 + \frac{\lambda_\beta}{2}||\beta||_F^2 + \frac{\lambda_w}{2}||w||_2^2$$

  where $M_{pa} := C_{pa}/D_{pa}$ is the MLE for the CTR
- **Issue**: $M_{pa}$ is noisy $\rightarrow$ confidence weighting is inaccurate
  - Ideally want to use true probability $P_{pa}$ itself

# An iterative heuristic

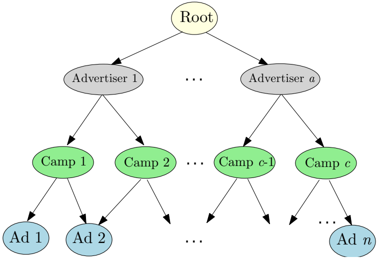
- After learning model, replace $M_{pa}$ with model prediction, and re-learn with new confidence weighting
  - Can iterate until convergence
- Can be used as part of latent/explicit feature interplay:



# Outline

# Hierarchical structure to response prediction data

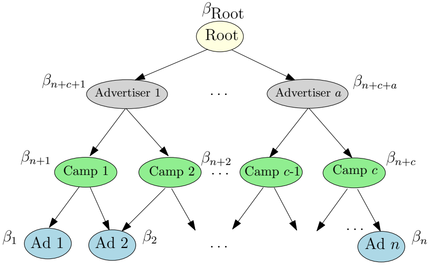
- Webpages and ads may be arranged into hierarchies:




- Hierarchy encodes correlations in CTRs
  - e.g. Two ads by same advertiser $\rightarrow$ similar CTRs
  - Highly structured form of side-information
- Successfully used in previous work [Agarwal et al., 2010]
  - How to exploit this information in our model?

# Using hierarchies: big picture

- **Intuition**: "similar" webpages/ads should have similar latent vectors
- Each node in the hierarchy is given its own latent vector



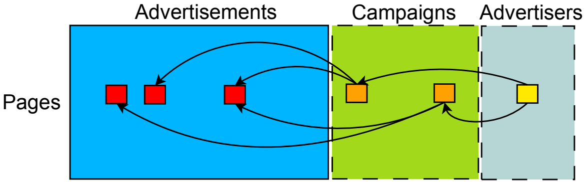

  - We will tie parameters based on links in hierarchy
  - Achieved in three simple steps

# Principle 1: Hierarchical regularization

- Each node's latent vector should equal its parent's, in expectation:
$$\alpha_p \sim \mathcal{N}(\alpha_{\mathsf{Parent}(p)}, \sigma^2 I)$$
- With a MAP estimate of the parameters, this corresponds to the regularizer
$$\Omega(\alpha) = \sum_{p,p'} S_{pp'} ||\alpha_p - \alpha_{p'}||_2^2$$
where $S_{pp'}$ is a parent indicator matrix
    - Latent vectors constrained to be similar to parents
    - Induces correlation amongst siblings in hierarchy

# Principle 2: Agglomerative fitting

- Can create meaningful priors by making parent nodes' vectors predictive of data:
  - Associate with each node clicks/views that are the sums of its childrens' clicks/views
  - Then consider an augmented matrix of all publisher and ad nodes, with appropriate clicks and views



# Principle 2: Agglomerative fitting

- We treat the aggregated data as just another response prediction dataset
  - Learn latent features for parent nodes on this data
  - Estimates will be more reliable than those of children
- Once estimated, these vectors serve as prior in hierarchical regularizer
  - Children's vectors are shrunk towards "agglomerated vector"

# Principle 3: Residual fitting

- Augment prediction to include bias terms for nodes along the path from root to leaf:

$$\hat{P}_{pa} = \sigma(\alpha_p^T \beta_a + \alpha_p^T \beta_{\mathsf{Parent}(a)} + \alpha_{\mathsf{Parent}(p)}^T \beta_{\mathsf{Parent}(a)} + \ldots)$$

  - Treats the hierarchy as a series of categorical features
- Can be viewed as decomposition of the latent vectors:

$$\tilde{\alpha}_p = \sum_{u \in \mathsf{Path}(p)} \alpha_u$$

$$\tilde{\beta}_a = \sum_{v \in \mathsf{Path}(a)} \beta_v$$

# The final model

- Our final model has the following ingredients:
  - Confidence weighting of the objective
  - Logistic loss to estimate meaningful probabilities
  - Incorporation of explicit features
    - Iterative heuristic for improving confidence weighting
  - Tying together of latent features via hierarchy
- Optimization can be done in alternating manner
  - Fix $\alpha$ and optimize for $\beta$, and vice-versa
  - Optimization for each $\beta_j$ can be done in parallel
    - Individual optimization via stochastic gradient descent

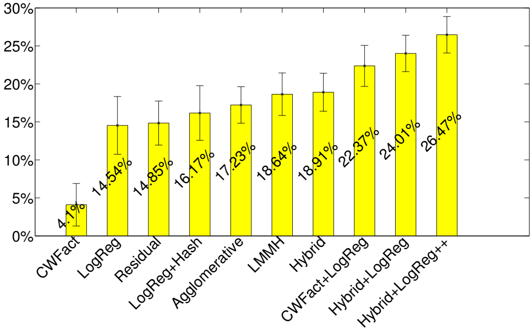
# Outline

# Experimental setup

- We compare the latent feature approach to three methods:
  1. Generalized linear model (GLM) on explicit features
  2. Logistic regression with cross-features [Richardson et al., 2007]
  3. Hierarchical log-linear model (**LMMH**) [Agarwal et al., 2010]
- Comparison is on three Yahoo! ad datasets:
  1. **Click**: (90B, 3B) (train, test) pairs
  2. Post-view conversions (**PVC**): (7B, 250M) (train, test) pairs
  3. Post-click conversions (**PCC**): (500M, 20M) (train, test) pairs
- Report % improvement in Bernoulli log-likelihood over GLM
  - Measure of quality of probabilities

# Results on Click

- Learning predictive latent features challenging due to sparsity
  - Using biases from hierarchy improves performance significantly
- With hierarchical tying, outperforms existing methods
- With explicit features, our model has clear lift over LMMH
  - Value in combining complementary information in the two




CWFact = Confidence weighted factorization

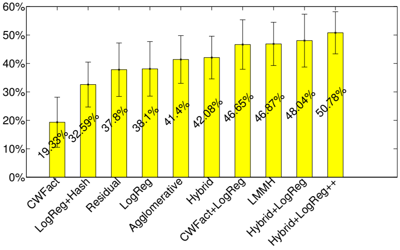
Hybrid = CWFact + All hierarchical components

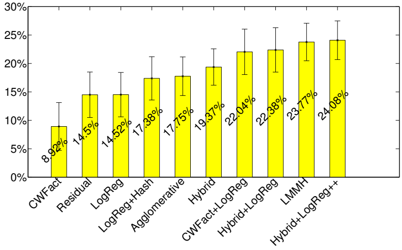
Hybrid+LogReg = With explicit features

Hybrid+LogReg++ = With iterative heuristic

# Results on PVC and PCC

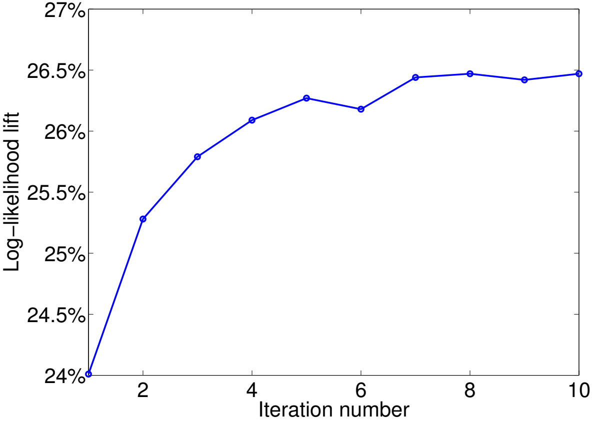
- Our combined model gives the best results on these datasets also
- Explicit features again important for best performance
  - Latent features alone are only competitive with LMMH
- On PCC, iterative heuristic helps outperform LMMH
  - Reliability of confidence weighting is important



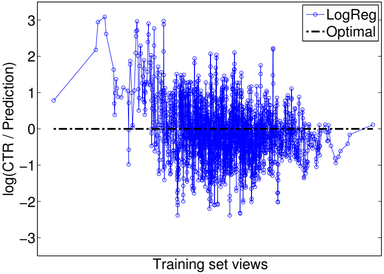
# Value of iterative confidence reweighting

- Trick of iteratively recomputing confidence-weighting by model prediction gives useful performance boost
  - Generally, log-likelihood improves after each such iteration

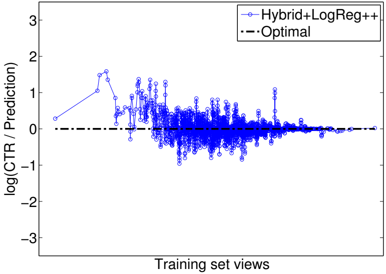


# Latent and explicit features

- Ideally, predictions should be $\sim$ MLE when $\#$ of displays is large
- With latent features, model converges to MLE faster
  - Variance of logistic regression model, which uses explicit features only, is significantly reduced




(a) Explicit features only




(b) Latent + explicit features

# Conclusions

- Response prediction can be approached from a collaborative filtering perspective
- Learning latent features for pages and ads gives state-of-the-art performance
- Some adaptation required for success in this domain
  - Had to use confidence weighting scheme
    - Iteratively refined the confidences
  - Incorporating explicit features gives important boost to lifts
  - Hierarchical information helps overcome data sparsity

# References I


Agarwal, D., Agrawal, R., Khanna, R., and Kota, N. (2010).
Estimating rates of rare events with multiple hierarchies through scalable log-linear models.
In *KDD '10*, pages 213–222, New York, NY, USA. ACM.

Agarwal, D. and Chen, B.-C. (2009).
Regression-based latent factor models.
In *KDD '09*, pages 19–28, New York, NY, USA. ACM.

Agarwal, D., Chen, B.-C., and Elango, P. (2009).
Spatio-temporal models for estimating click-through rate.
In *WWW '09*, pages 21–30, New York, NY, USA. ACM.

Menon, A. K. and Elkan, C. (2010).
A log-linear model with latent features for dyadic prediction.
In *ICDM '10*.

Richardson, M., Dominowska, E., and Ragno, R. (2007).
Predicting clicks: estimating the click-through rate for new ads.
In *WWW '07*, pages 521–530, New York, NY, USA. ACM.

Yang, S., Long, B., Smola, A., Sadagopan, N., Zheng, Z., and Zha, H. (2011).
Like like alike – joint friendship and interest propagation in social networks.
In *WWW '11*.