Part I:

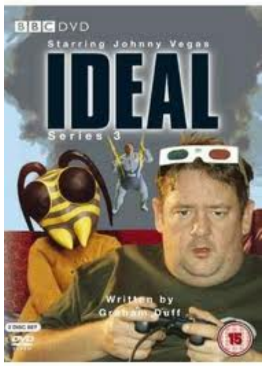
Latent feature models for dyadic prediction

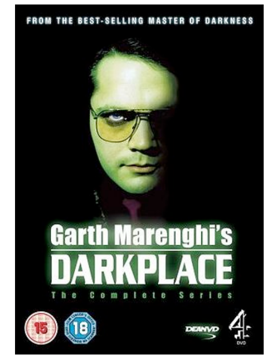
Aditya Krishna Menon

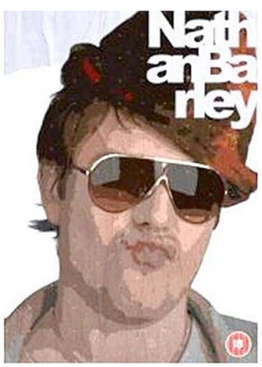
Outline of this talk

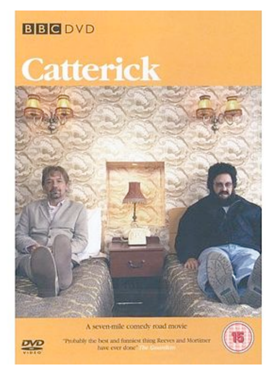
- **What is dyadic prediction?**
  - Flavour of its flexibility
- A generic model for dyadic prediction
  - The latent feature approach
- Applications to specific instantiations
  - Collaborative filtering
  - Response prediction

What to watch next?



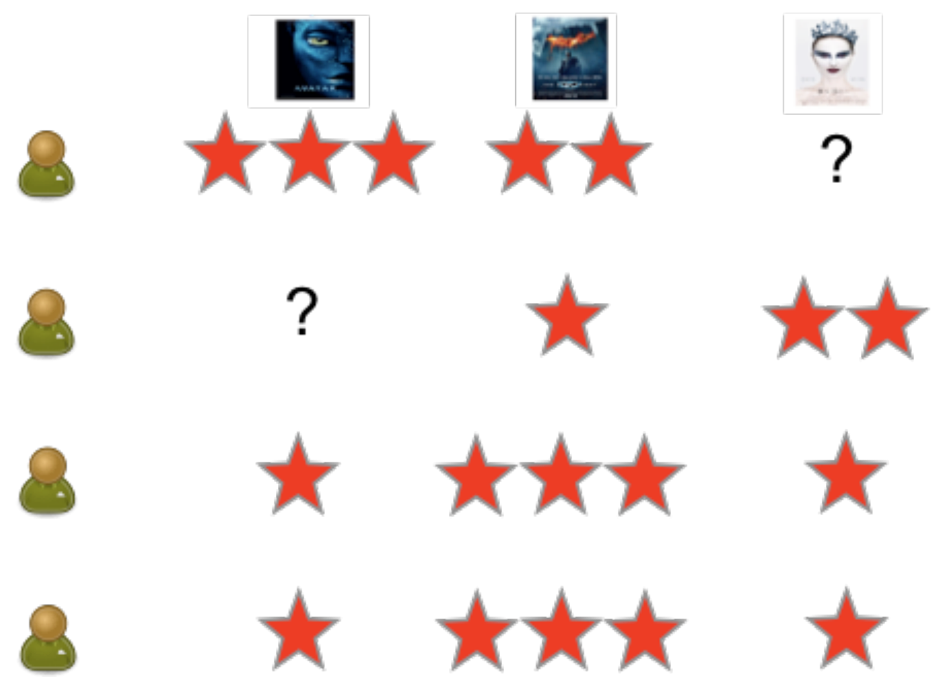




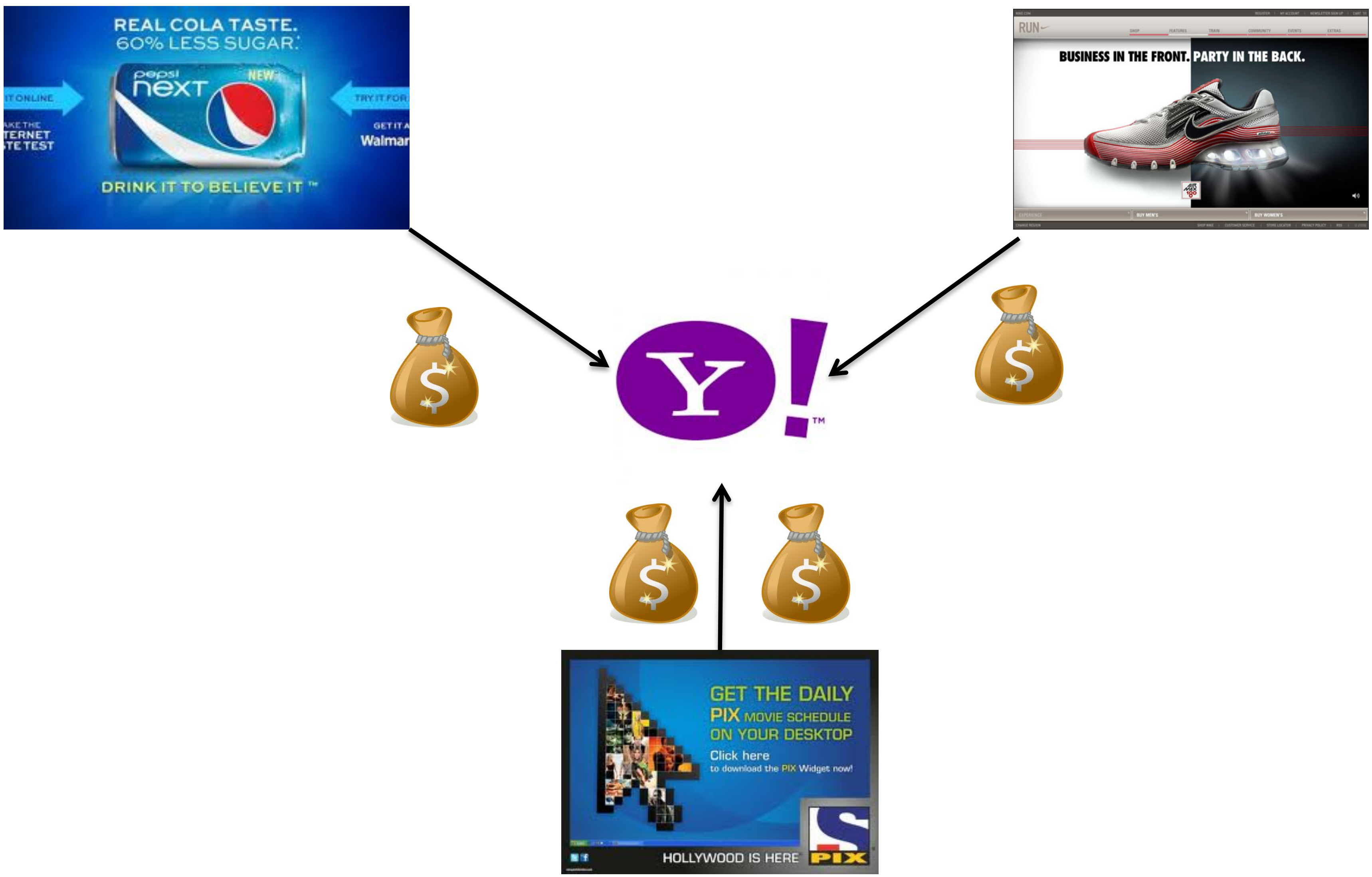


Formalism: Collaborative filtering

- Based on database of users' ratings for movies, predict rating user will give to a movie



Which ad will pay off?



Formalism: Response prediction

- Given historical data, predict clickthrough rate for ad on a webpage

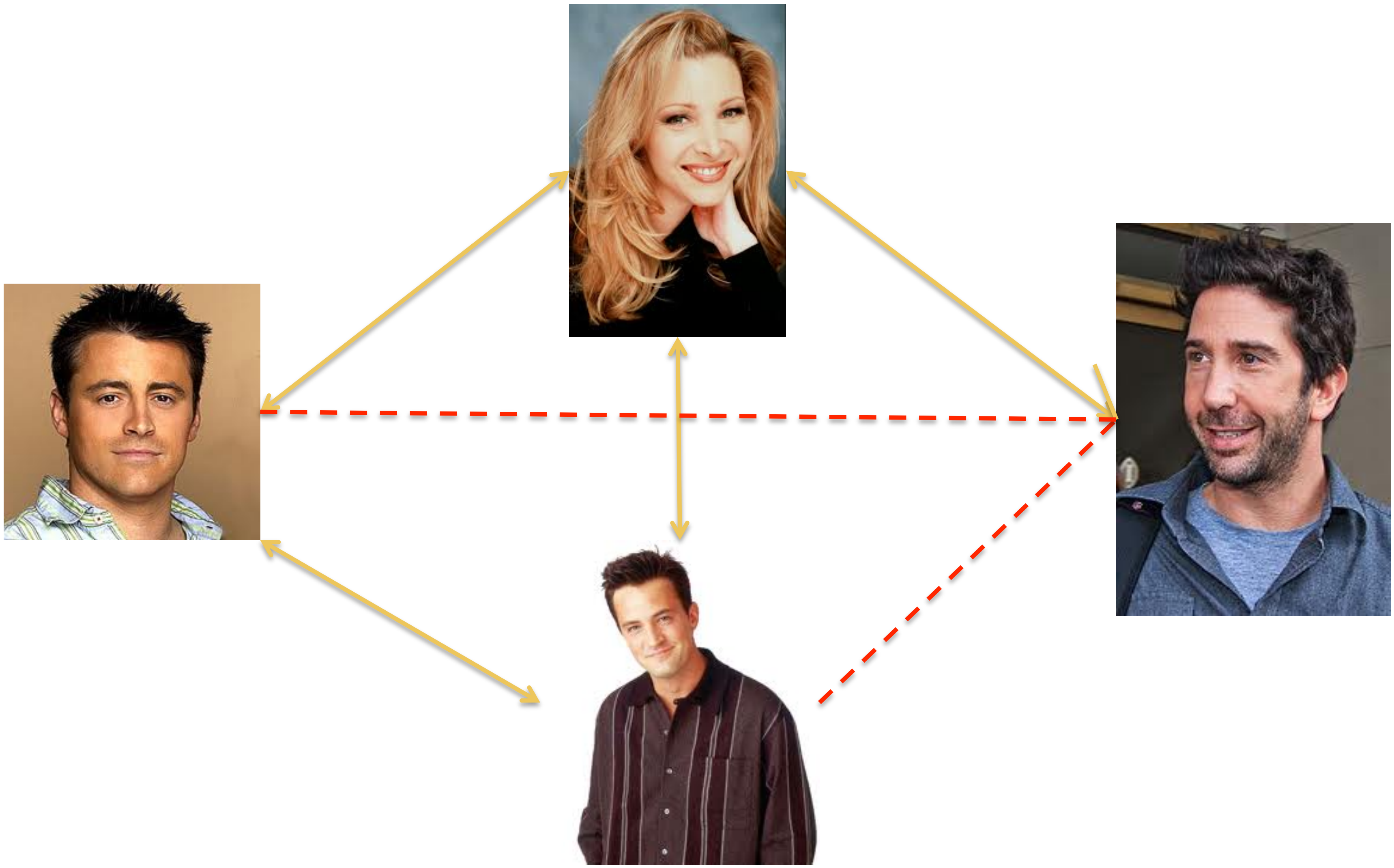
0.1

0.2

0.001

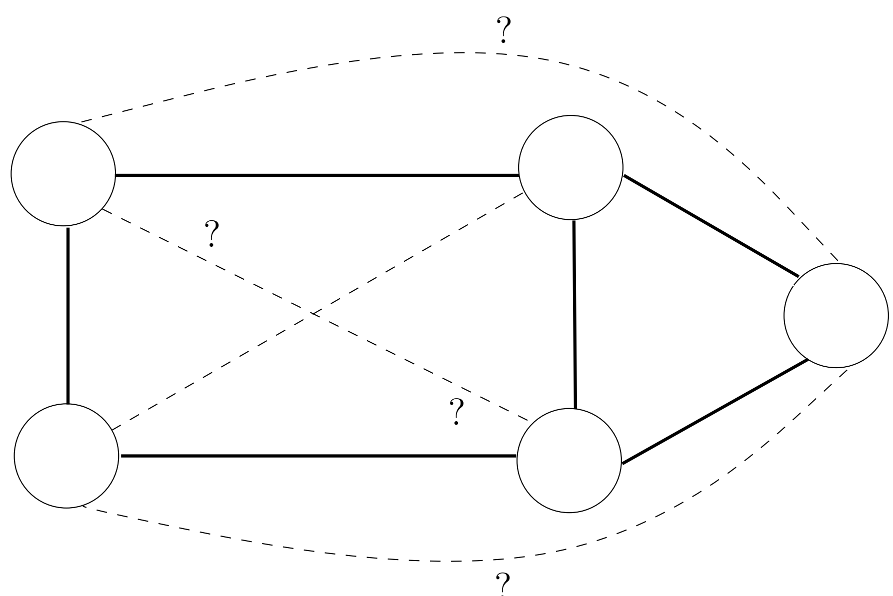
?

Do I know you?



Formalism: Link prediction

- Given known links between nodes in a graph, predict which other node pairs are likely to have an edge



An abstract view

- Rating a user gives to a movie

- Clickthrough rate of ad on webpage

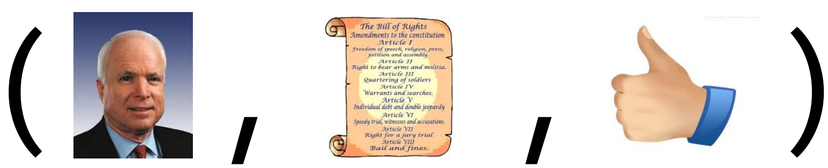
- Friendship status between users

The list goes on…

- Correctness of student responses to test questions

- Suspiciousness of staff accesses to patient records

- Politician's vote on a bill



Dyadic prediction: informally

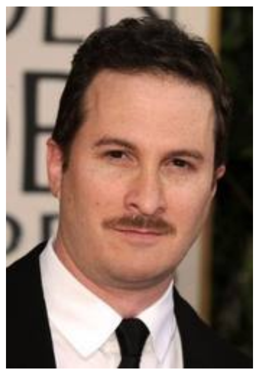
- Predict label for interaction of pair of entities (dyad)
  - (User, movie) dyad, star rating label
  - (User, user) dyad, friendship relation label
  - (Webpage, ad) dyad, clickthrough rate label

Flexibility- I

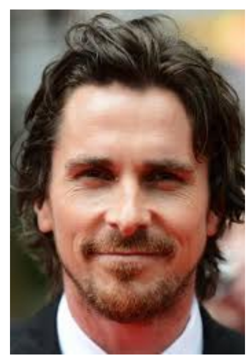
- Dyad members may possess explicit features…

Director **Lead Star** **…**

=



=



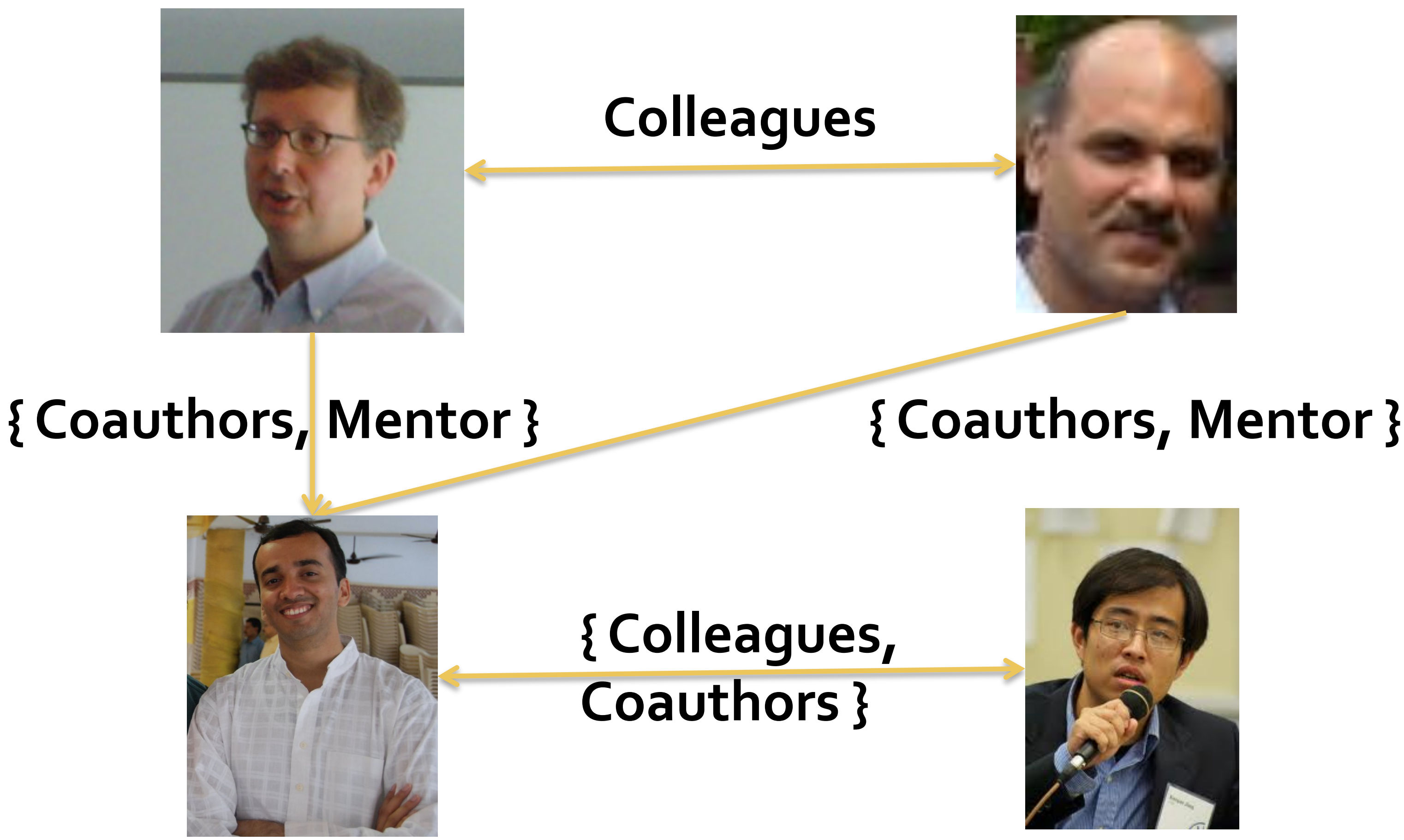
Flexibility- II

- ...or only unique identifiers

User ID	Movie ID	Rating
10001	330	1.5
10001	2451	4.5
4003	84794	3.0

Flexibility- III

- Labels may be nominal and/or multidimensional



Dyadic prediction: formally

- **Input**: Training set $\{((i^{(t)}, j^{(t)}, x^{(t)}), y^{(t)})\}_{t=1}^{T}$
  - Each $(i^{(t)}, j^{(t)}) \in [M] \times [N]$ is the dyad, represented as a pair of unique identifiers
    - **Example**: (User ID, Movie ID) = (10001, 330)
  - Each $x^{(t)} \in \mathbb{R}^D$ is the optional set of side-information
    - **Example**: Movie director, lead star, …
  - Each $y^{(t)} \in \mathcal{Y}$ is the label
    - **Example**: User rating for movie
- **Output**: predictor $f : [M] \times [N] \times \mathbb{R}^D \to \mathcal{Y}$

Existing approaches

- Different problem instances use different models
  - Collaborative filtering → matrix factorization
  - Response prediction → supervised learning
  - Link prediction → graph-theoretic scores
- All good ideas, and work well
  - But is it necessary to use different techniques?

This talk

- We'll study a generic model for dyadic prediction
  - The latent feature approach
- Applications to the preceding problems
  - Is there value in unified interpretation?
    - Comparison of empirical performance
    - Adaptability to problem-specific constraints

Outline of this talk

- What is dyadic prediction?
  - Flavour of its flexibility
- **A generic model for dyadic prediction**
  - The latent feature approach
- Applications to specific instantiations
  - Collaborative filtering
  - Response prediction

The log-linear framework

- We build on the log-linear framework
  - Captures logistic regression, CRFs, et cetera
- For input *x* and label *y*:

$$\Pr[y|x;\theta] \propto \exp(\theta^T f(x,y))$$

  - Elements of $f(x, y)$ are called feature functions
  - Note *y* could be nominal, multidimensional, sequence, …

Dealing with dyadic data

- In the basic dyadic setting, we have $x = (i, j)$
  - Identities for the dyad members e.g. (10001, 330)
- Essentially two choices for feature functions*:*

Dealing with dyadic data

- In the basic dyadic setting, we have $x = (i, j)$
  - Identities for the dyad members e.g. (10001, 330)
- Essentially two choices for feature functions*:*

Independent

$$\Pr[y|i,j;\theta] \propto \exp(u_i^{(y)} + v_j^{(y)})$$

Underfits: for fixed *i*, ranking over dyads independent of *j*!

Dealing with dyadic data

- In the basic dyadic setting, we have $x = (i, j)$
  - Identities for the dyad members e.g. (10001, 330)
- Essentially two choices for feature functions:

Independent

$$\Pr[y|i,j;\theta] \propto \exp(u_i^{(y)} + v_j^{(y)})$$

Underfits: for fixed *i*, ranking over dyads independent of *j*!

Joint

$$\Pr[y|i,j;\theta] \propto \exp(W_{ij}^{(y)})$$

Overfits: memorizes training data, does not generalization

How to generalize?

- To allow generalization, we factorize the weights:

$$W_{ij}^{(y)} = u_i^T \Lambda^{(y)} v_j$$

where $u_i, v_j \in \mathbb{R}^K$ and $\Lambda^{(y)} \in \mathbb{R}^{K \times K}$

- Can employ other factorizations too, e.g.:

$$W_{ij}^{(y)} = u_i^T v_j^{(y)}$$

  – More parameters, may overfit

Alternate perspective

- Can interpret as factorization of the log-odds

$$\log \frac{\Pr[y|i,j;\theta]}{\Pr[y_{\text{base}}|i,j;\theta]} = u_i^T \Lambda^{(y)} v_j$$

when we fix a base class y_{base}

  - Series of matrix factorizations

- c.f. logistic regression:

$$\log \frac{\Pr[y|x;\theta]}{\Pr[y_{\text{base}}|x;\theta]} = (w^{(y)})^T x$$

Exploiting features

- Given explicit features x_{ij} for dyad (*i*, *j*), model:

$$\Pr[y|i,j;\theta] \propto \exp(u_i^T \Lambda^{(y)} v_j + (w^{(y)})^T x_{ij})$$

- Given separate features for *i* and *j*, fuse them via bilinear model:

$$\Pr[y|i,j;\theta] \propto \exp(u_i^T \Lambda^{(y)} v_j + x_i^T (W^{(y)}) x_j)$$

Latent feature log-linear model

- The final model looks like

$$\Pr[y|i,j;\theta] \propto \exp(u_i^T \Lambda^{(y)} v_j + (w^{(y)})^T x_{ij})$$

which we call the latent feature log-linear model (LFL)

- Exploits both identity and feature information

Training

- We minimize the regularized negative log-likelihood

$$\mathcal{L}(\theta) = -\frac{1}{T}\sum_{t=1}^{T} \log \Pr[y^{(t)}|i^{(t)}, j^{(t)}; \theta] + \frac{\lambda}{2}||\theta||_2^2$$

on a training set $\{((i^{(t)}, j^{(t)}), y^{(t)})\}_{t=1}^{T}$

  – Does **not** impute labels for unobserved dyads

  – Amenable to stochastic gradient training

Why use this model?

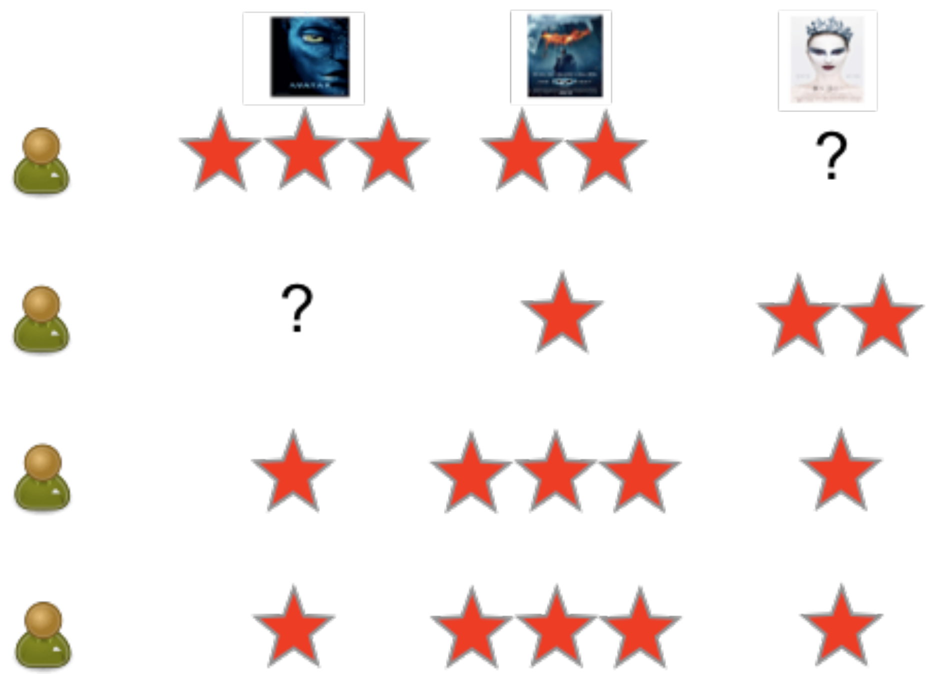
- Many appealing properties in log-linear model
  - Learns predictive latent representation for dyad members
  - Easy to incorporate explicit features
  - Training is scalable
  - Familiar framework, easily extensible
- We'll now closely study specific applications
  - Can it easily adapt to domain-specific challenges?

Outline of this talk

- What is dyadic prediction?
  - Flavour of its flexibility
- A generic model for dyadic prediction
  - The latent feature approach
- Applications to specific instantiations
  - **Collaborative filtering**
  - Response prediction

Recall: collaborative filtering

- Predict missing (user, movie) ratings



Why use LFL?

- The log-linear approach models a rating distribution
  - Measures confidence in prediction

Prediction: mean rating

- Optimal prediction for squared error = expected rating
- Easy to use as prediction:

$$\begin{aligned}\mathrm{Pred}(i, j; \theta) &= \mathbb{E}[y] \\ &= \sum_{y \in \mathcal{Y}} y \cdot \mathrm{Pr}[y | i, j; \theta]\end{aligned}$$

  - Recall that $\mathcal{Y} = \{1, 2, \ldots, 5\}$ e.g.

Adapting to numeric labels

- Would like model to exploit fact that labels are numeric
- Can modify underlying model, e.g. enforce ordering on scaling factors:

$$\Lambda^{(y+1)} - \Lambda^{(y)} \succeq 0$$

- Can directly optimize error of expected rating:

$$\frac{1}{T}\sum_{t=1}^{T}(y^{(t)} - \mathbb{E}[y|i^{(t)}, j^{(t)}; \theta])^2 + \lambda||\theta||_2^2$$

  - Empirically, work better

Assessing uncertainty

- Prediction for user *i* and movie *j* is

$$\mathrm{Pred}(i,j;\theta) = \mathbb{E}[y|i,j;\theta]$$

- Prediction confidence is

$$\mathrm{Pred}(i,j;\theta) = \mathbb{E}[y^2|i,j;\theta] - (\mathbb{E}[y|i,j;\theta])^2$$

  – Dyads may have same mean, but different confidences

  – Can take into account when recommending

    • Surrogate for diversity

Comparison to basic factorization

- Basic matrix factorization ("SVD") predicts

$$\mathrm{Pred}(i, j; \theta) = u_i^T v_j$$

- LFL predicts

$$\mathrm{Pred}(i, j; \theta) = \frac{1}{Z_{ij}} \sum_{y \in \mathcal{Y}} y \cdot \exp(u_i^T \Lambda^{(y)} v_j)$$

  – Weighted combination of several low rank terms

Comparison to RBM

- RBM requires marginalization over hidden units

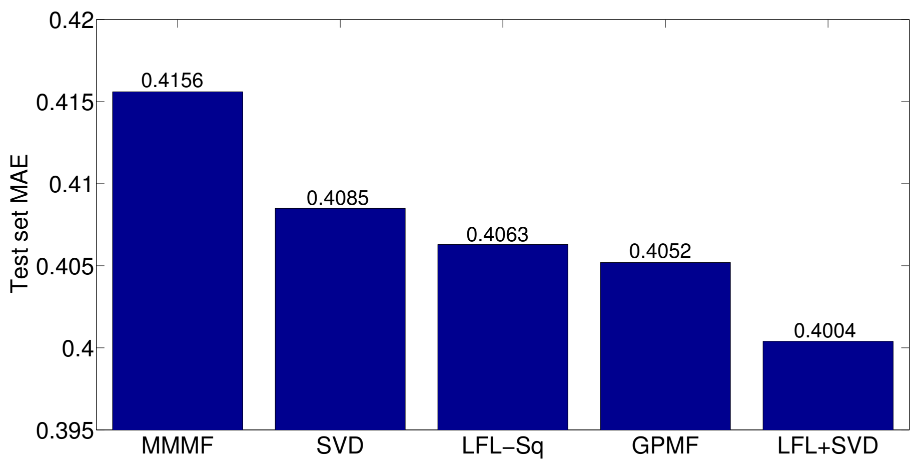
$$\Pr[y|i,j;\theta] \propto \sum_{h \in \{0,1\}^K} \exp\left(\sum_{k=1}^{K} h_k W_{jk}^{(y)}\right)$$

- LFL is "discriminative" alternative
  - Stochastic gradient vs contrastive divergence training

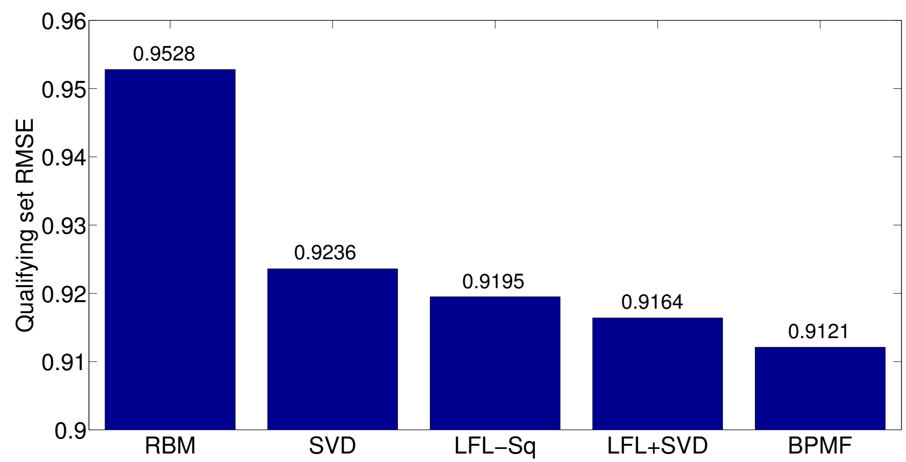
Results on benchmark datasets

- Competitive with baselines, including SVD
  - Blends well with SVD
  - Nonlinear/Bayesian methods work well, as expected

Movielens 1M



Netflix



Analysis of learned probabilities

- Analysis on data from RateYourMusic

User likes:

Most certain:

Most uncertain:

Outline of this talk

- What is dyadic prediction?
  - Flavour of its flexibility
- A generic model for dyadic prediction
  - The latent feature approach
- Applications to specific instantiations
  - Collaborative filtering
  - **Response prediction**

Recall: Response prediction

- Given historical data, predict clickthrough rate for ad on a webpage

Estimating the CTR

- Simplest estimate is counting: for page *p* and ad *a*,

$$\Pr[y = 1 | p, a; \theta] = \frac{\#\ \text{of clicks}}{\#\ \text{of displays}}$$

  – Noisy, possibly undefined

- Possibly smoother estimates with logistic regression

$$\Pr[y = 1 | p, a; \theta] = \sigma(w^T x_{pa} + b)$$

  – Collecting features not always simple
    - "Annoyance" factor of ad

Latent feature model

- Binary LFL applied to an individual click event:

$$\Pr[y = 1|p, a; \theta] = \sigma(u_p^T v_a + w^T x_{pa})$$

  - Logistic regression + latent component

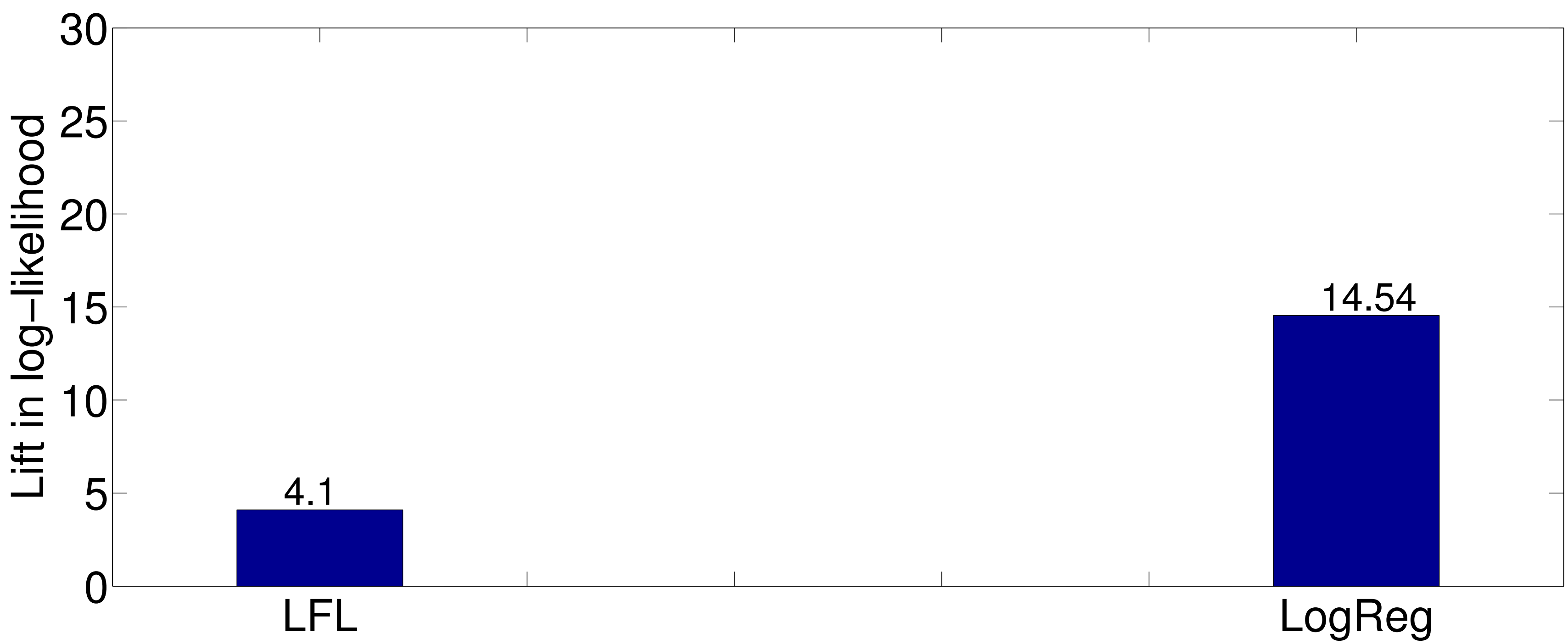
- Overall objective is confidence weighted

$$\sum_{(p,a)} -C_{pa} \log \Pr[y = 1|p, a; \theta] - N_{pa} \log \Pr[y = 0|p, a; \theta]$$

where *C* is the # of clicks, *N* is the # of non-clicks

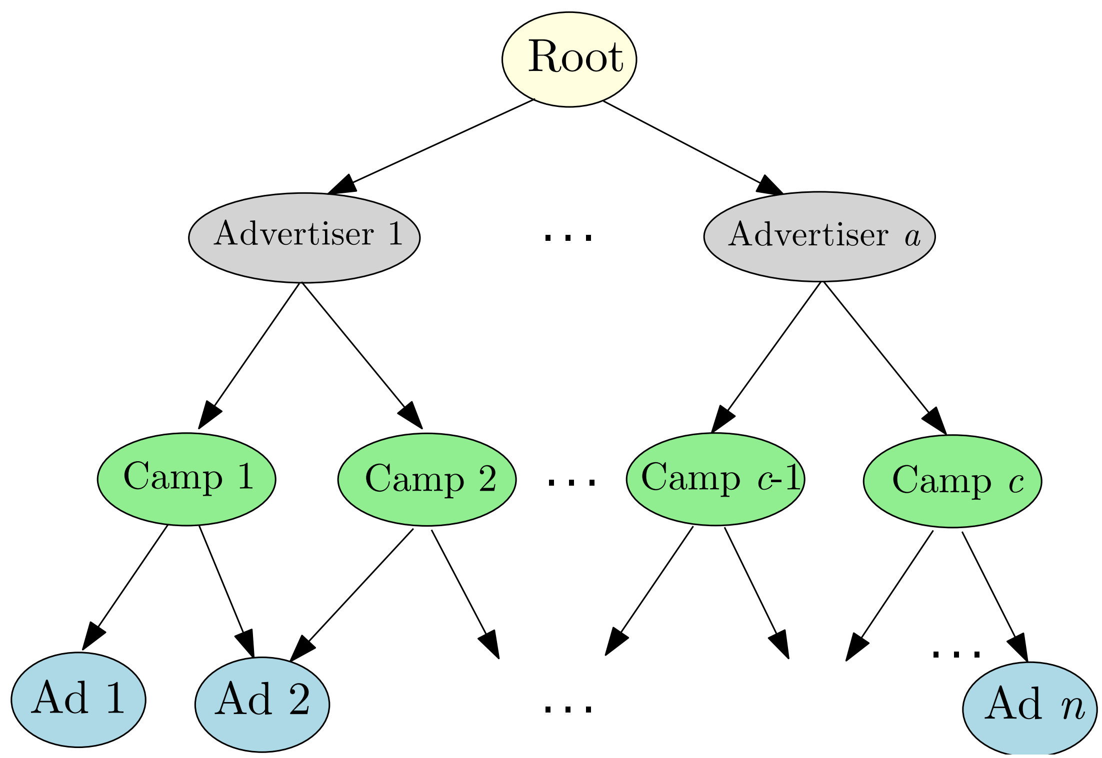
Too sparse for latent features?

- Basic latent feature model performs poorly
  - Sparsity is a major challenge
  - Difficult to reliably estimate page/ad latent features



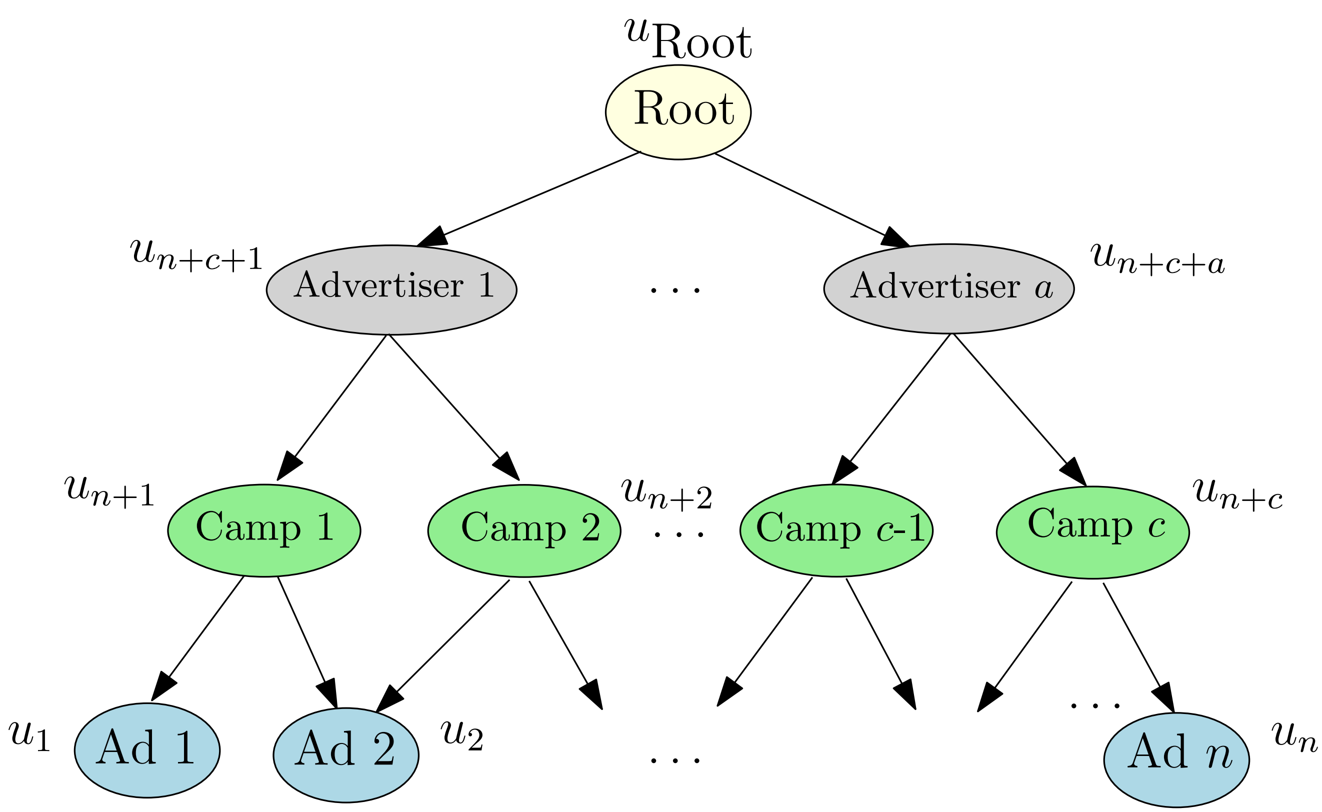
Overcoming sparsity with hierarchies

- Webpages and ads may be arranged in a hierarchy
  - Valuable source of prior information



Exploiting hierarchical information

- Learn latent features for all nodes in hierarchy



Regularization

- Standard ℓ_2 regularization corresponds to prior:

$$u_i \sim \mathcal{N}(0, \sigma^2 I)$$

- Hierarchy-informed prior:

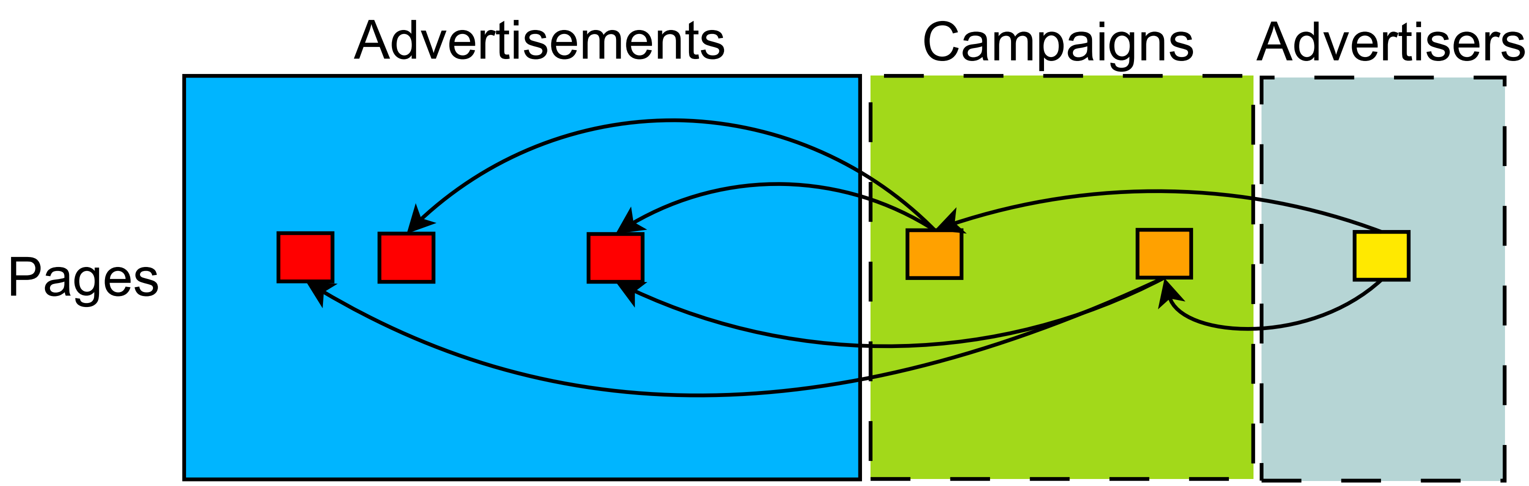
$$u_i \sim \mathcal{N}(u_{\mathrm{Par}(i)}, \sigma^2 I)$$

where Par(*i*) denotes the parent of *i*

  - Encodes relationships between pages and ads
  - But how to estimate parent nodes' vectors?

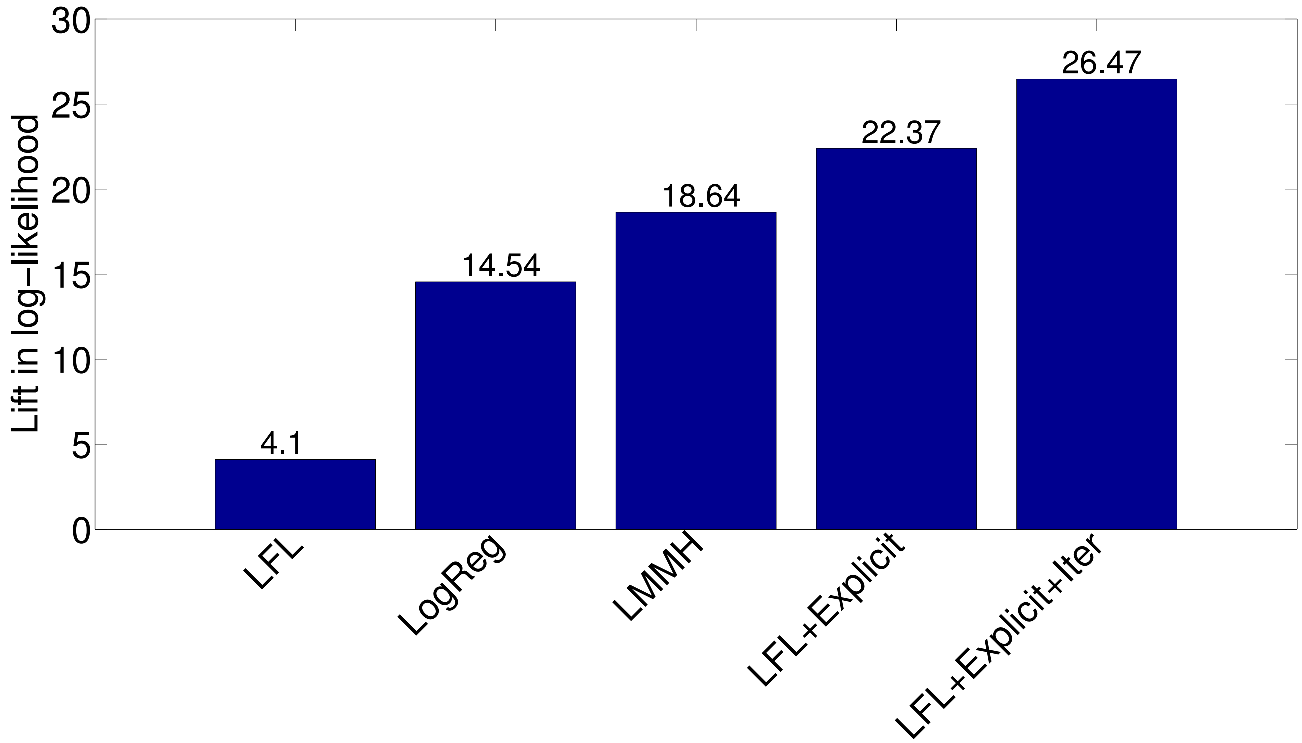
Agglomeration

- Estimation based on agglomeration at each level
  - e.g. for advertiser parameters, use all the labels corresponding to ads by that advertiser



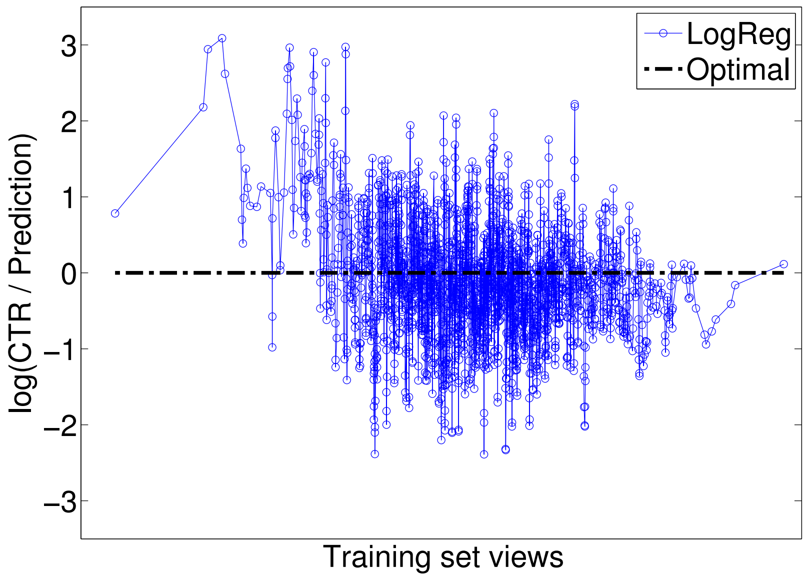
Results on Yahoo! Click data

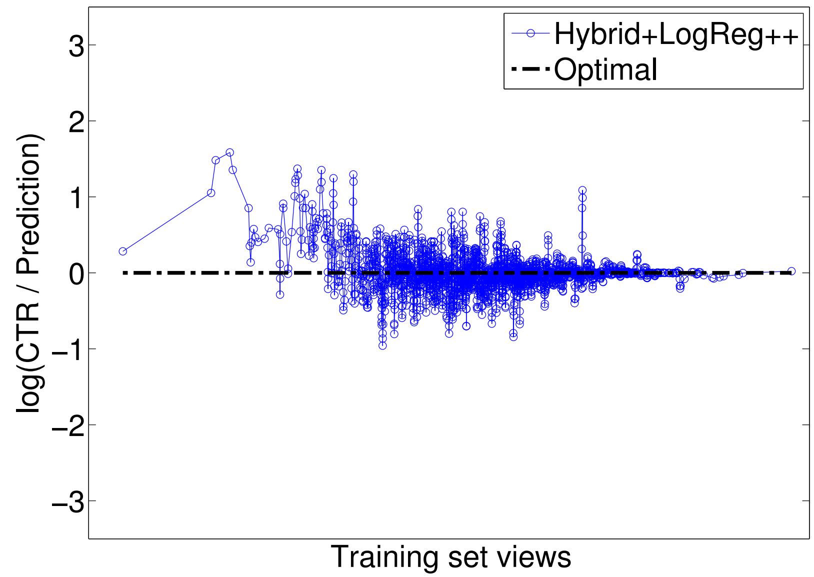
- 90B training examples, 20K features (categorical)
- LFL improves over previous state-of-the-art, LMMH



Value of latent features

- Latent features significantly reduce noise in predicted probabilities





Summary

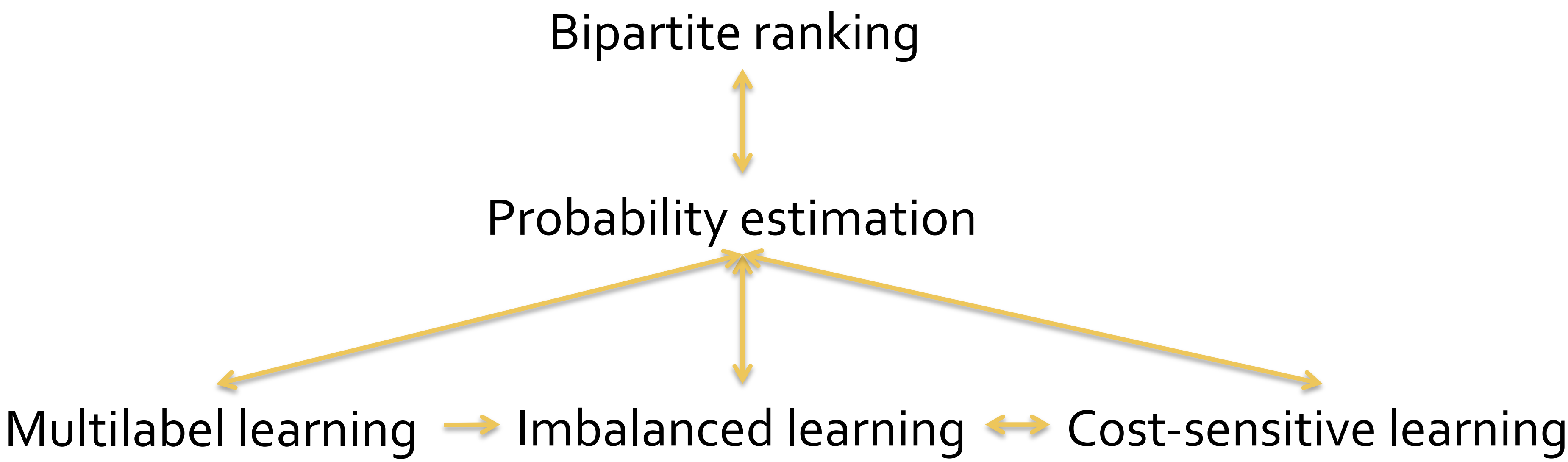
- Many important problems may be cast as instances of dyadic prediction
- Latent feature modelling is an appealing foundation for dyadic prediction tasks
  - Good empirical performance in domains of collaborative filtering, link prediction, response prediction
  - Unified perspective helps borrow good ideas from other fields

Part II

Probability estimation, ranking and friends

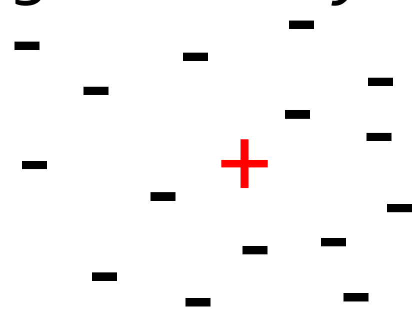
Aditya Krishna Menon

Some nascent interests



Imbalanced learning

- Supervised learning when Pr[*y*] is close to zero



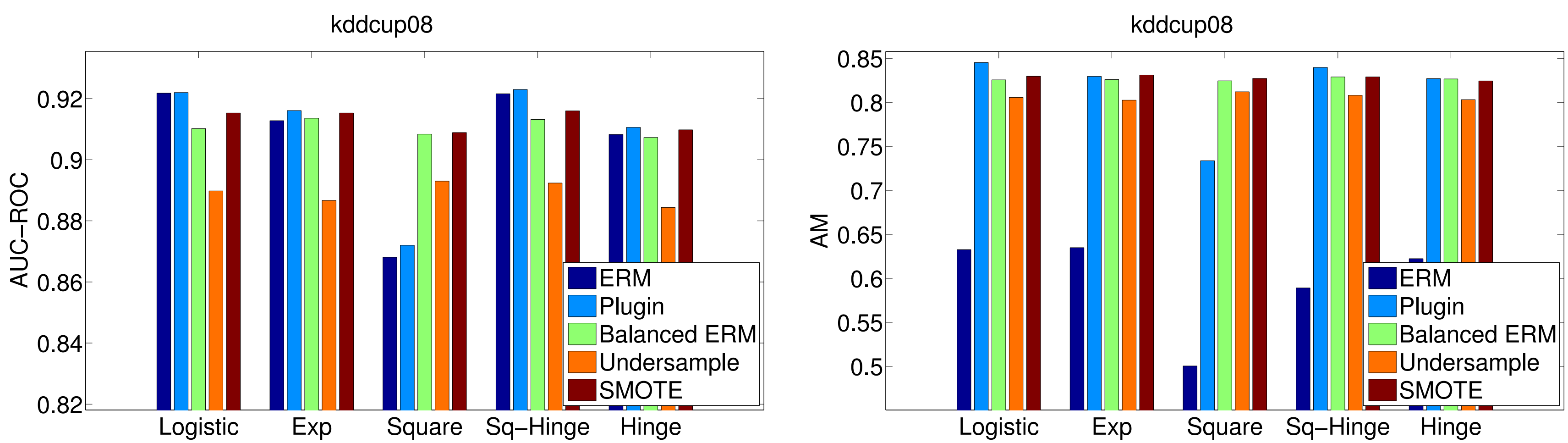
- 0-1 error not a suitable metric
  - Balanced error rate (BER) sensible:

$$\mathrm{BER}[s] = 1 - \frac{\mathrm{TPR}[s] + \mathrm{TNR}[s]}{2}$$

  - Area under ROC (AUC) another de-facto choice

An empirical observation

- As a baseline, linear/logistic regression:
  - is difficult to beat in AUC
  - is easy to beat in BER



- Why is this so?

Probability estimation for ranking

- Recently, [Agarwal '13] showed the regret bound:

$$\text{Reg}^{\text{rank}}[g] \leq \frac{C}{\Pr[y=0] \cdot \Pr[y=1]} \cdot \sqrt{\text{Reg}^{\ell}[g]}$$

where *l* is a proper loss

  - Reg denotes excess risk over Bayes optimal
  - Good probability estimation → good ranking wrt AUC

- Probability estimators are AUC-consistent
  - Empirically, robust to finite samples and misspecification

Probability estimation vs ranking

- When (if ever) should we optimize AUC directly?
  - Compare to a direct regret bound for latter
  - Distributional assumptions?
- What about finite sample effects?
  - Logistic regression is biased under imbalance
  - Contrast to bias for ranking
- Statistical properties of the objectives
  - Does ranking optimization generalize faster?

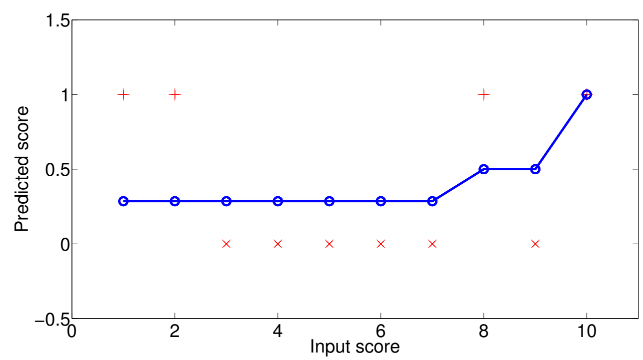
Probability estimation vs ranking

- Translate theory on proper losses to ranking
  - Ranking and *f*-divergences
  - Hand's incoherence argument for AUC
- What about measures other than AUC?
  - Partial AUC, AUPRC, …
  - Margin-based generalizations

Ranking for probability estimation

- Using ranking to estimate probabilities:

1. Find scores that maximize the AUC
   – Discovers a monotone transform of probabilities
2. Apply isotonic regression to recover probabilities
   – Minimize squared error subject to rank preservation



Ranking for probability estimation

- Learns probabilities from single-index model family

$$\Pr[y = 1|x] = f(w^T x)$$

where *f* is a (not a-priori known) monotone function

- – Unlike GLMs, *f* must itself be estimated

Ranking for probability estimation

- The Isotron [Kalai & Sastry '09] is similar, but is:
  - Requires multiple IR calls
  - Not gradient-following of a clear objective
- Are there provable virtues of the AUC+IR approach?
  - Better sample complexity?
  - Effect of misspecification?
- Would LogReg + IR work as well?

Back to imbalance!

- For AUC, probability estimators are consistent
- For BER, we need to specify thresholding scheme:
  - Learn probabilities, threshold at $\Pr[y = 1]$
  - Apply (cost-sensitive) weighting $1/\Pr[y]$, threshold at 0.5
- Both can be shown to be BER-consistent
  - Form of regret bound is similar

BER and beyond

- Choosing between thresholding and weighting?
  - Effect of misspecification
  - Sample complexity
- Weighting and proper losses
  - Better estimation of probabilities under imbalance?
  - Relation to cost-sensitive integral representation?
- Consistency for any f(TPR, TNR)?
  - How about Precision?

Summary

- Many basic problems have connections to probability estimation
- Better understanding of these connections may:
  - Give alternative perspective of existing models
  - Lead to new models
  - Lead to more questions!

Questions?