Collaborative Deep Learning for Recommender Systems

Presenter: Muhammad Umair Khan

Outline

01 Overview for Recommender System

02 Collaborative Deep Learning

03 Experiments

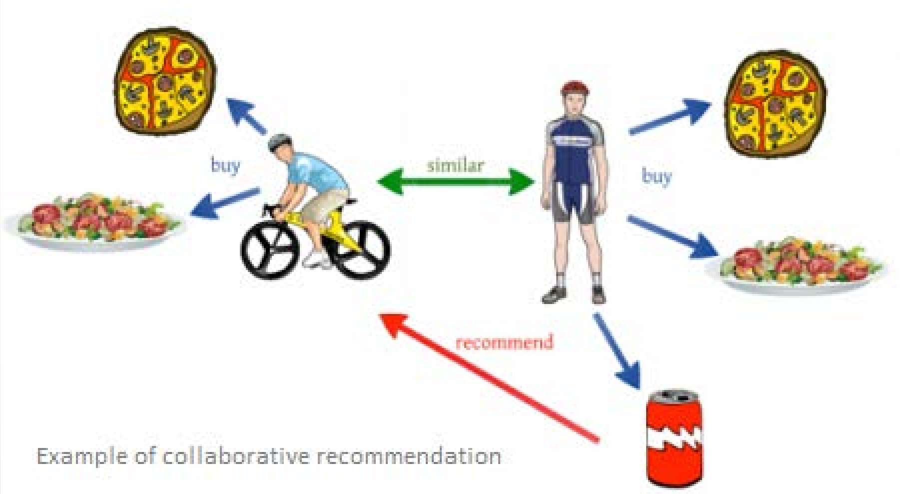
04 Conclusion

Overview for Recommender System

Kinds of Recommendation System

- Content-based methods
- CF-based methods
  - ✓ user-based
  - ✓ item-based
  - ✓ model-based (eg. Matrix Factorization)
- Hybrid methods:
  - ✓ Loosely coupled methods
  - ✓ Tightly coupled methods (CTR、CDL)

Example of content based recommendation



Example of collaborative recommendation

Overview for Recommender System

Collaborative Topic Regression (CTR)

- tightly coupled method
- probabilistic graphical model
  - topic model: Latent Dirichlet Allocation (LDA)
  - model-based CF method: probabilistic matrix factorization (PMF)

Motivations

- The latent representation learned by CTR is often not effective enough especially when the auxiliary information is very sparse.

- Deep learning models can learn effective feature representations from text content automatically.

- DL models are inferior to shallow models such as CF in capturing and learning the similarity and implicit relationship between items (and users).

Outline

01 **Overview for Recommender System**

02 **Collaborative Deep Learning**

03 **Experiments**

04 **Conclusion**

Collaborative Deep Learning

- A novel tightly coupled method for RS.
- Taking implicit feedback as the training and test data.
- Collaborative Filtering
  - for the ratings (feedback) matrix
  - to capture the similarity and implicit relationship between items (and users)
- Deep Learning
  - for the content information
  - to extract an effective deep feature representation

Collaborative Deep Learning

01 Matrix Factorization

02 Stacked Denoising Autoencoders

03 Collaborative Deep Learning

04 Learning and Prediction

06

Matrix factorization

- Users and items are represented in a shared but unknown latent space (lantent factor model)
  - user $i - u_i \in R^k$
  - item $j - v_j \in R^k$
- Each dimension of the latent space is assumed to represent some kind of *unknown factors*
- The rating of item j by user i is achieved by the dot product,

$$r_{ij} = u_i^T v_j,$$

where $r_{ij} = 1$ indicates *like* and 0 *dislike*. In the matrix form,

$$R = U^T V.$$

Matrix factorization

Learning and Prediction

- Learning the latent vectors for users and items

$$\min_{U,V} \sum_{i,j} (r_{ij} - u_i^T v_j)^2 + \lambda_u \|u_i\|^2 + \lambda_v \|v_j\|^2,$$

  where λ_u and λ_v are regularization parameters.
- Prediction for user i on item j (not rated by user i before),

$$r_{ij} \approx u_i^T v_j.$$

Probabilistic matrix factorization (PMF)

- This matrix factorization for collaborative filtering can be generalized as a probabilistic model. In probabilistic matrix factorization (PMF), we assume the following generative process,

1. For each user i, draw user latent vector $u_i \sim \mathcal{N}(0, \lambda_u^{-1} I_K)$.
2. For each item j, draw item latent vector $v_j \sim \mathcal{N}(0, \lambda_v^{-1} I_K)$.
3. For each user-item pair (i, j), draw the response

$$r_{ij} \sim \mathcal{N}(u_i^T v_j, c_{ij}^{-1}),$$

where c_{ij} is the precision parameter for r_{ij}.

Disadvantages for Matrix Factorization

Two main disadvantages to matrix factorization for recommendation

- learnt latent space is not easy to interpret
- only uses information from the users-cannot to geralize to completely unrated items

Stacked Denoising Autoencoders (SDAE)

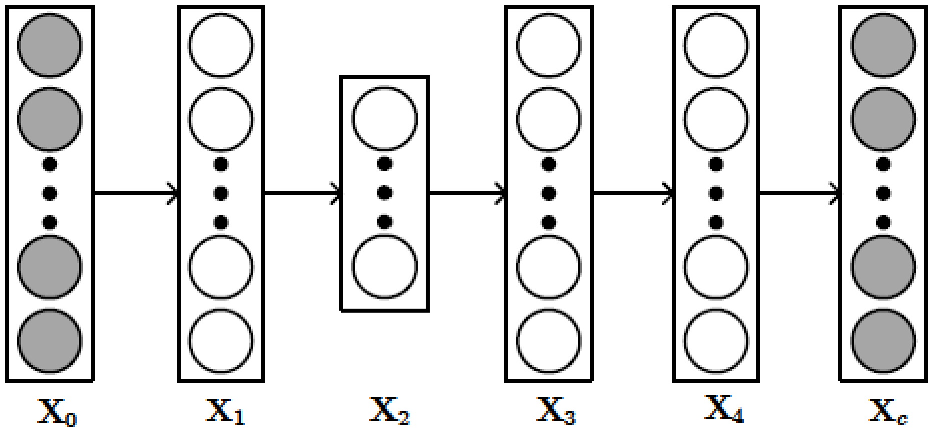


Figure 2: A 2-layer SDAE with $L = 4$.

SDAE solves the following optimization problem:

$$\min_{\{\mathbf{W}_l\},\{\mathbf{b}_l\}} \|\mathbf{X}_c - \mathbf{X}_L\|_F^2 + \lambda \sum_l \|\mathbf{W}_l\|_F^2,$$

Collaborative Deep Learning (CDL)

1. For each layer l of the SDAE network,

   (a) For each column n of the weight matrix $\mathbf{W}_l$, draw

   $$\mathbf{W}_{l,*n} \sim \mathcal{N}(\mathbf{0}, \lambda_w^{-1}\mathbf{I}_{K_l}).$$

   (b) Draw the bias vector $\mathbf{b}_l \sim \mathcal{N}(\mathbf{0}, \lambda_w^{-1}\mathbf{I}_{K_l})$.

   (c) For each row j of $\mathbf{X}_l$, draw

   $$\mathbf{X}_{l,j*} \sim \mathcal{N}(\sigma(\mathbf{X}_{l-1,j*}\mathbf{W}_l + \mathbf{b}_l), \lambda_s^{-1}\mathbf{I}_{K_l}).$$

2. For each item j,

   (a) Draw a clean input $\mathbf{X}_{c,j*} \sim \mathcal{N}(\mathbf{X}_{L,j*}, \lambda_n^{-1}\mathbf{I}_J)$.

   (b) Draw a latent item offset vector $\boldsymbol{\epsilon}_j \sim \mathcal{N}(\mathbf{0}, \lambda_v^{-1}\mathbf{I}_K)$ and then set the latent item vector to be:

   $$\mathbf{v}_j = \boldsymbol{\epsilon}_j + \mathbf{X}^T_{\frac{L}{2},j*}.$$

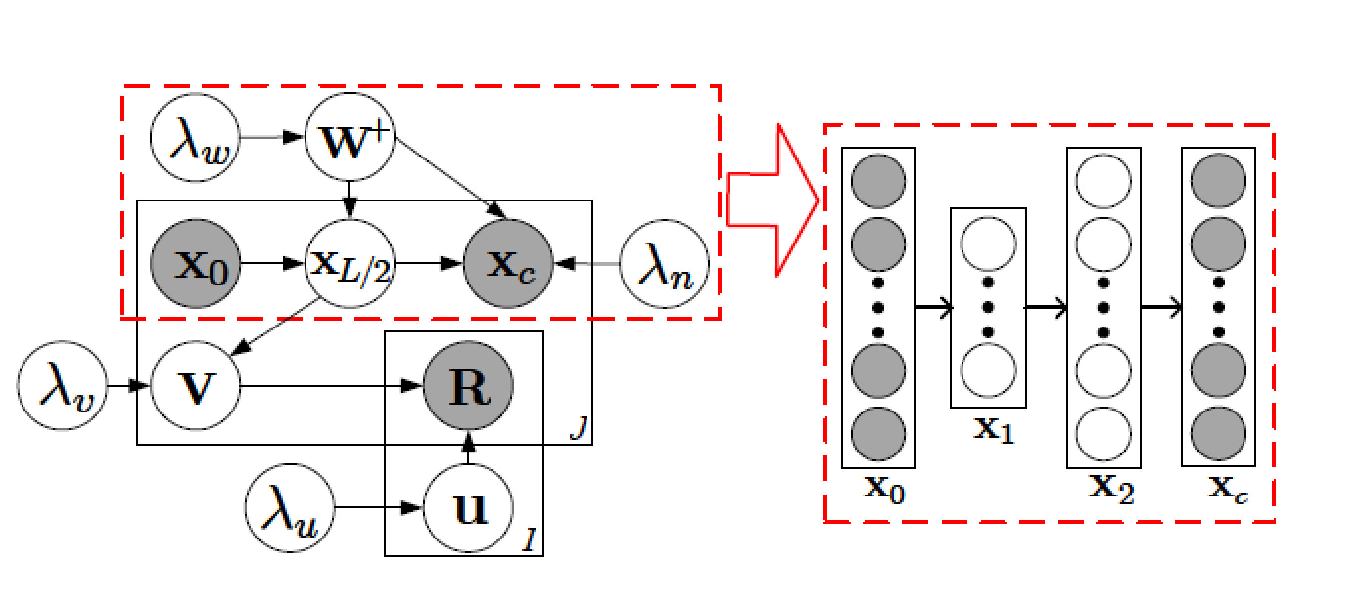
3. Draw a latent user vector for each user i:

   $$\mathbf{u}_i \sim \mathcal{N}(\mathbf{0}, \lambda_u^{-1}\mathbf{I}_K).$$

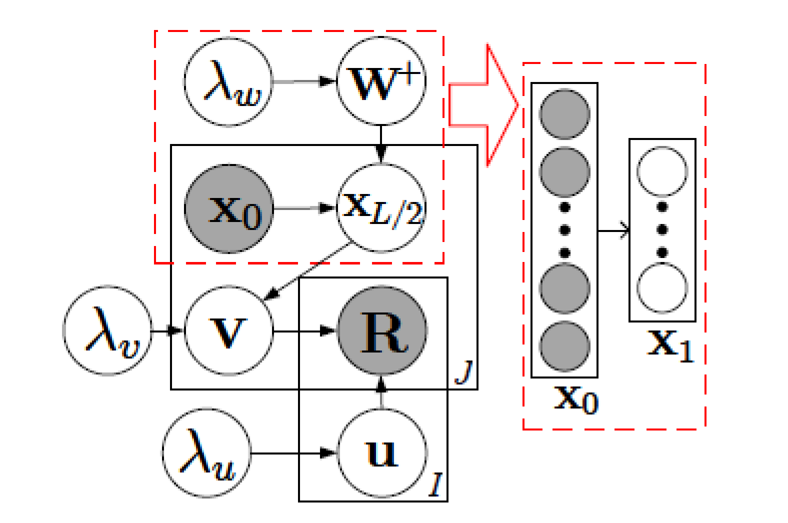
4. Draw a rating $\mathbf{R}_{ij}$ for each user-item pair (i, j):

   $$\mathbf{R}_{ij} \sim \mathcal{N}(\mathbf{u}_i^T\mathbf{v}_j, \mathbf{C}_{ij}^{-1}).$$

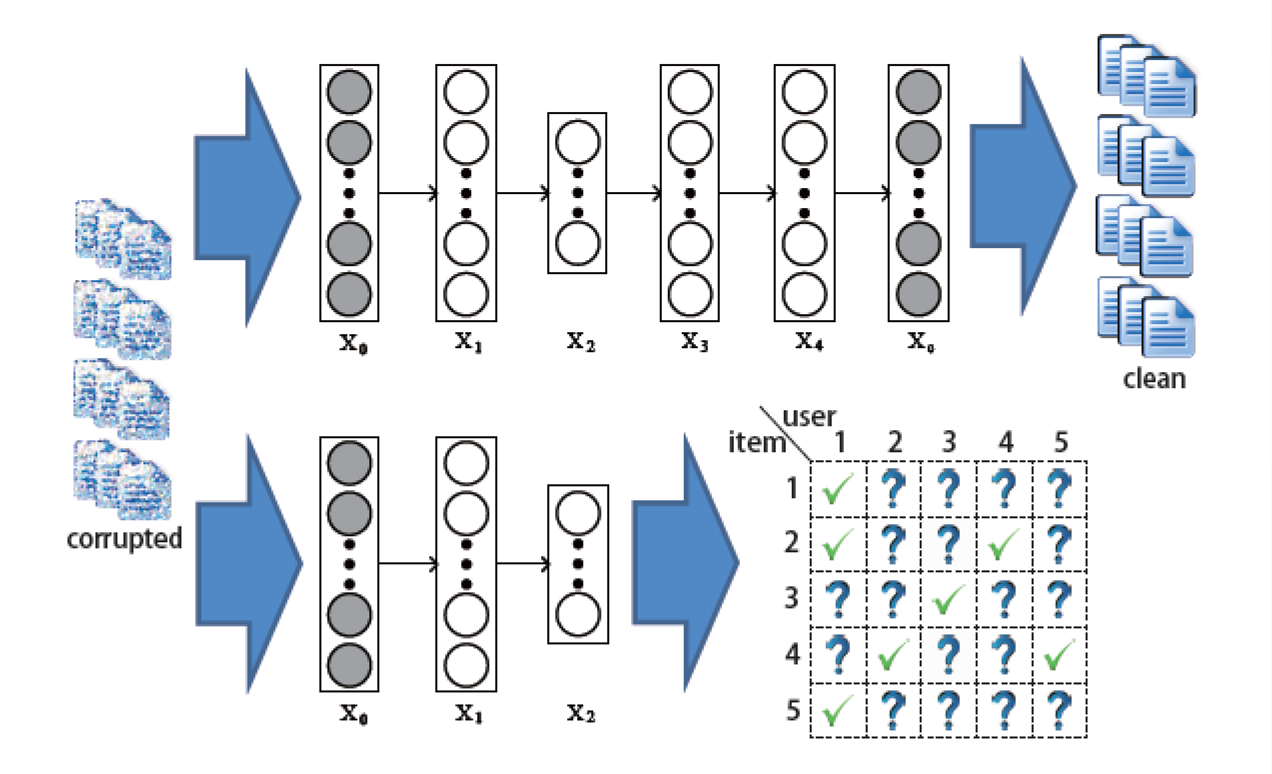
The graphical model of CDL



The graphical model of the degenerated CDL



NN representation for degenerated CDL



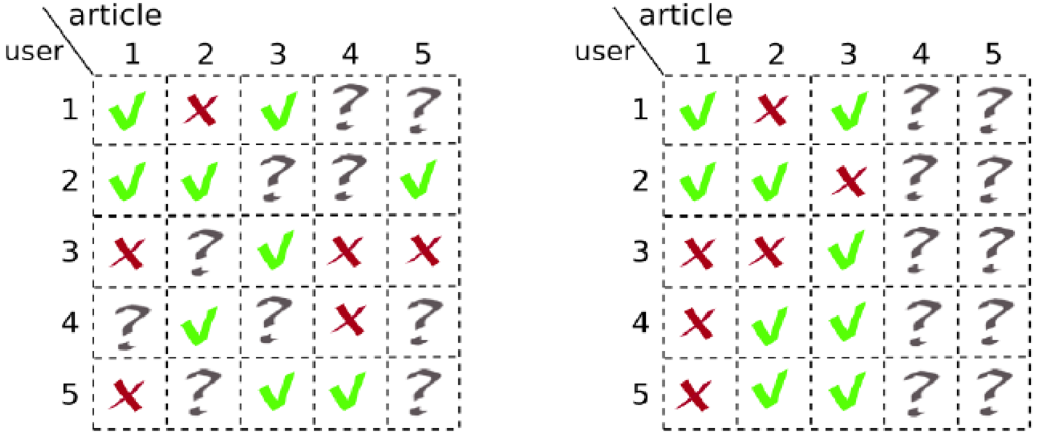
Learning and Prediction

- **Learning**: use a standard EM algorithm to learn the maximum a posteriori (MAP) estimates.
- **Prediction**: consider two scenarios,
  - In-matrix prediction: items have been rated before

$$\mathbf{R}^*_{ij} \approx (\mathbf{u}^*_j)^T (f_e(\mathbf{X}_{0,j*}, \mathbf{W}^{+*})^T + \epsilon^*_j) = (\mathbf{u}^*_i)^T \mathbf{v}^*_j.$$

  - Out-of-matrix prediction: items have never been rated

offset ϵ^*_j will be $\mathbf{0}$.



(a) in-matrix prediction (b) out-of-matrix prediction

Outline

01 Overview for Recommender System

02 Collaborative Deep Learning

03 Experiments

04 Conclusion

Experiments

Dataset

	users	items	ratings	sparsity
citeulike-a	5,551	16,980	207,363	0.22%
citeulike-t	7,947	25,975	144,496	0.07%
Netflix	407,261	9,228	15,348,808	0.41%

- removing users with less than 3 positive ratings (or articles)
- removing stop words and choosing the top S discriminative words according to the tf-idf values to form the vocabulary (S is 8000, 20000, and 20000 for the three datasets)

Experiments

Evaluation:

- five-fold cross-validation with recall under both sparse and dense settings,

$$\text{recall@}M = \frac{\text{number of items that the user likes among the top } M}{\text{total number of items that the user likes}}.$$

Baselines:

- CMF: Collective Matrix Factorization is a model incorporating different sources of information by simultaneously factorizing multiple matrices.
- SVDFeature: SVDFeature is a model for feature-based collaborative filtering.
- DeepMusic: DeepMusic is a model for music recommendation
- CTR: Collaborative Topic Regression is a model performing topic modeling and collaborative filtering simultaneously.
- CDL: Collaborative Deep Learning is our proposed model. It allows different levels of model complexity by varying the number of layers.

Results

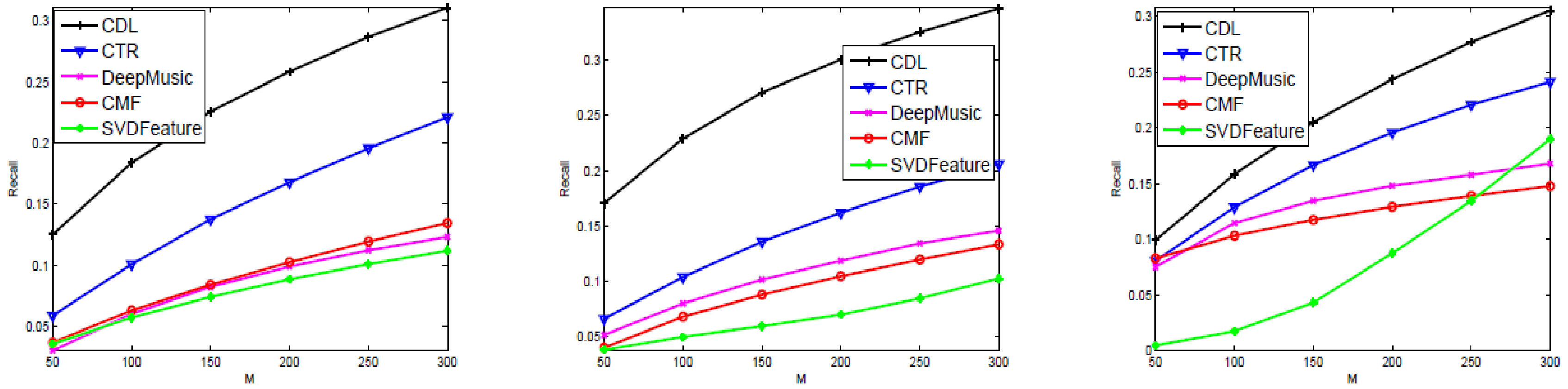


Figure 4: Performance comparison of CDL, CTR, DeepMusic, CMF, and SVDFeature based on recall@M for datasets *citeulike-a*, *citeulike-t*, and *Netflix* in the sparse setting. A 2-layer CDL is used.

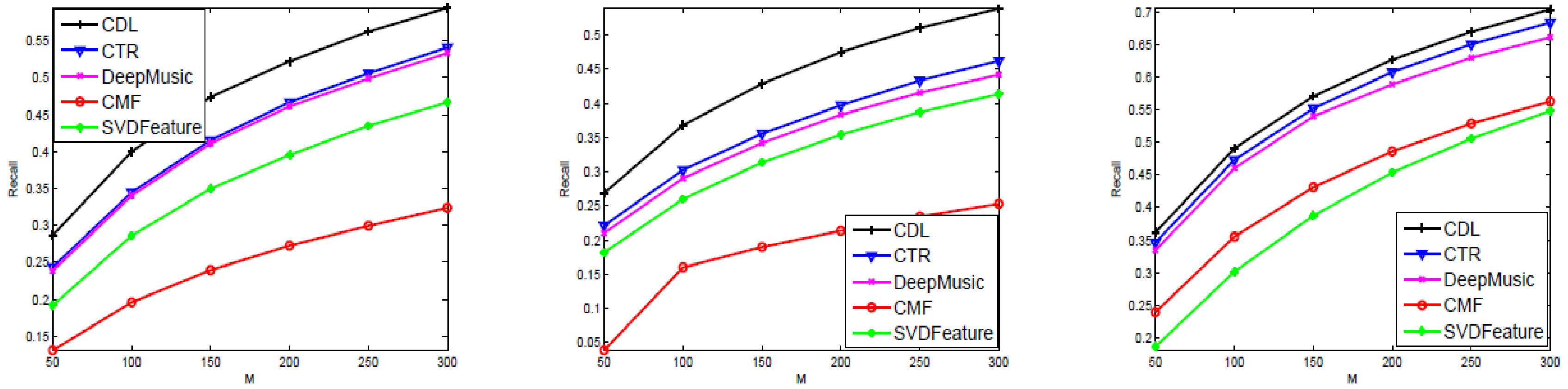


Figure 5: Performance comparison of CDL, CTR, DeepMusic, CMF, and SVDFeature based on recall@M for datasets *citeulike-a*, *citeulike-t*, and *Netflix* in the dense setting. A 2-layer CDL is used.

Results

Table 2: Recall@300 in the sparse setting (%)

#layers	1	2	3
citeulike-a	27.89	**31.06**	30.70
citeulike-t	32.58	34.67	**35.48**
Netflix	29.20	30.50	**31.01**

Table 3: Recall@300 in the dense setting (%)

#layers	1	2	3
citeulike-a	58.35	**59.43**	59.31
citeulike-t	52.68	53.81	**54.48**
Netflix	69.26	70.40	**70.42**

Interpretation: Example user I

	user I (CDL)	in user's lib?
top 3 topics	1. search, image, query, images, queries, tagging, index, tags, searching, tag 2. social, online, internet, communities, sharing, networking, facebook, friends, ties, participation 3. collaborative, optimization, filtering, recommendation, contextual, planning, items, preferences	
top 10 articles	**1. The structure of collaborative tagging Systems**	yes
	2. Usage patterns of collaborative tagging systems	yes
	3. Folksonomy as a complex network	no
	4. HT06, tagging paper, taxonomy, Flickr, academic article, to read	yes
	5. Why do tagging systems work	yes
	6. Information retrieval in folksonomies: search and ranking	no
	7. tagging, communities, vocabulary, evolution	yes
	8. The complex dynamics of collaborative tagging	yes
	9. Improved annotation of the blogosphere via autotagging and hierarchical clustering	no
	10. Collaborative tagging as a tripartite network	yes

	user I (CTR)	in user's lib?
top 3 topics	1. social, online, internet, communities, sharing, networking, facebook, friends, ties, participation 2. search, image, query, images, queries, tagging, index, tags, searching, tag 3. feedback, event, transformation, wikipedia, indicators, vitamin, log, indirect, taxonomy	
top 10 articles	**1. HT06, tagging paper, taxonomy, Flickr, academic article, to read**	yes
	2. Structure and evolution of online social networks	no
	3. Group formation in large social networks: membership, growth, and evolution	no
	4. Measurement and analysis of online social networks	no
	5. A face(book) in the crowd: social searching vs. social browsing	no
	6. The strength of weak ties	no
	7. Flickr tag recommendation based on collective knowledge	no
	8. The computer-mediated communication network	no
	9. Social capital, self-esteem, and use of online social network sites: A longitudinal analysis	no
	10. Increasing participation in online communities: A framework for human-computer interaction	no

Outline

01 Overview for Recommender System

02 Collaborative Deep Learning

03 Experiments

04 Conclusion

Conclusion

- the first hierarchical Bayesian model to bridge the gap between state-of-the-art deep learning models and RS
- combine the merits of traditional collaborative filtering and deep learning model
- provides an interpretable latent structure for users and items
- can form recommendation about both existing and newly items
- conduct extensive experiments on three real-world datasets from different domains

Thank you!

Any Questions!!