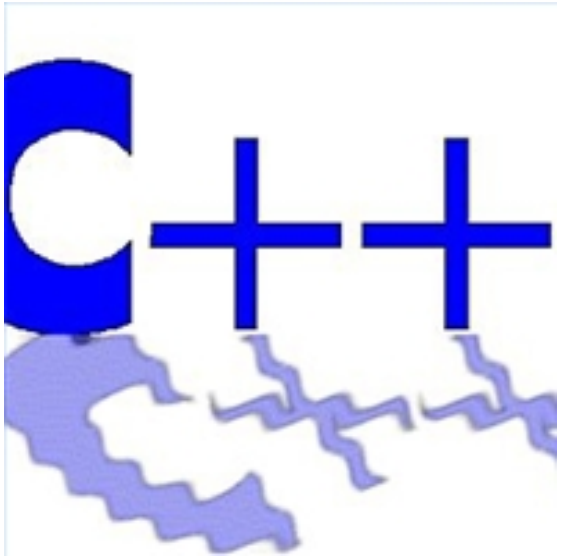
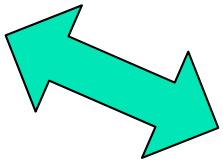
Transfer Learning
for Link Prediction

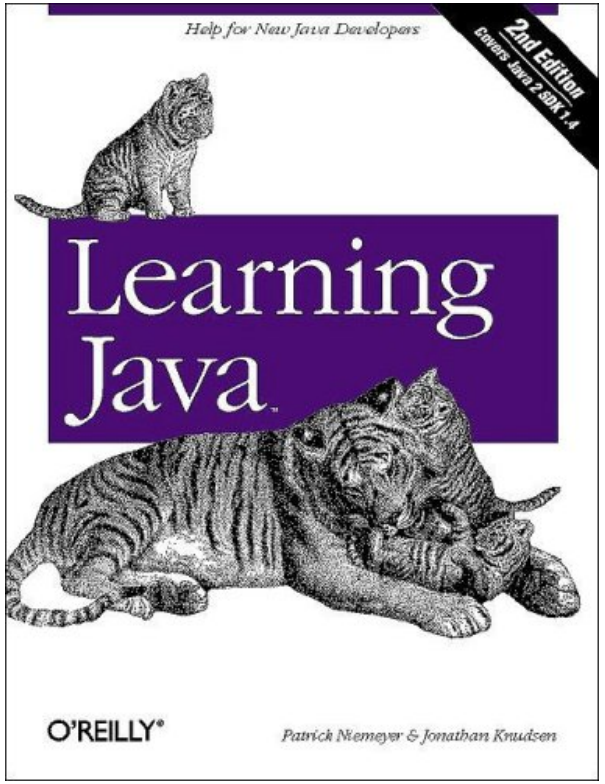
Qiang Yang，楊強，香港科大
Hong Kong University of Science and Technology
Hong Kong, China

http://www.cse.ust.hk/~qyang

We often find it easier…







Transfer Learning? 迁移学习...

- People often transfer knowledge to novel situations
  - Chess → Checkers
  - C++ → Java
  - Physics → Computer Science

Transfer Learning:

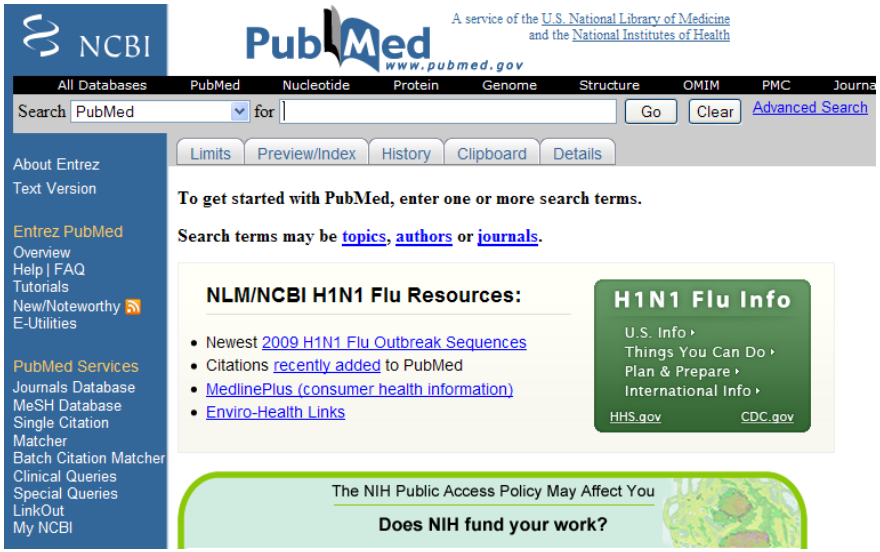
The ability of a system to recognize and apply knowledge and skills learned in previous tasks to novel tasks (or new domains)

But, direct application will not work

Machine Learning:

- Training and future (test) data
  - follow the same distribution, and
  - are in same feature space

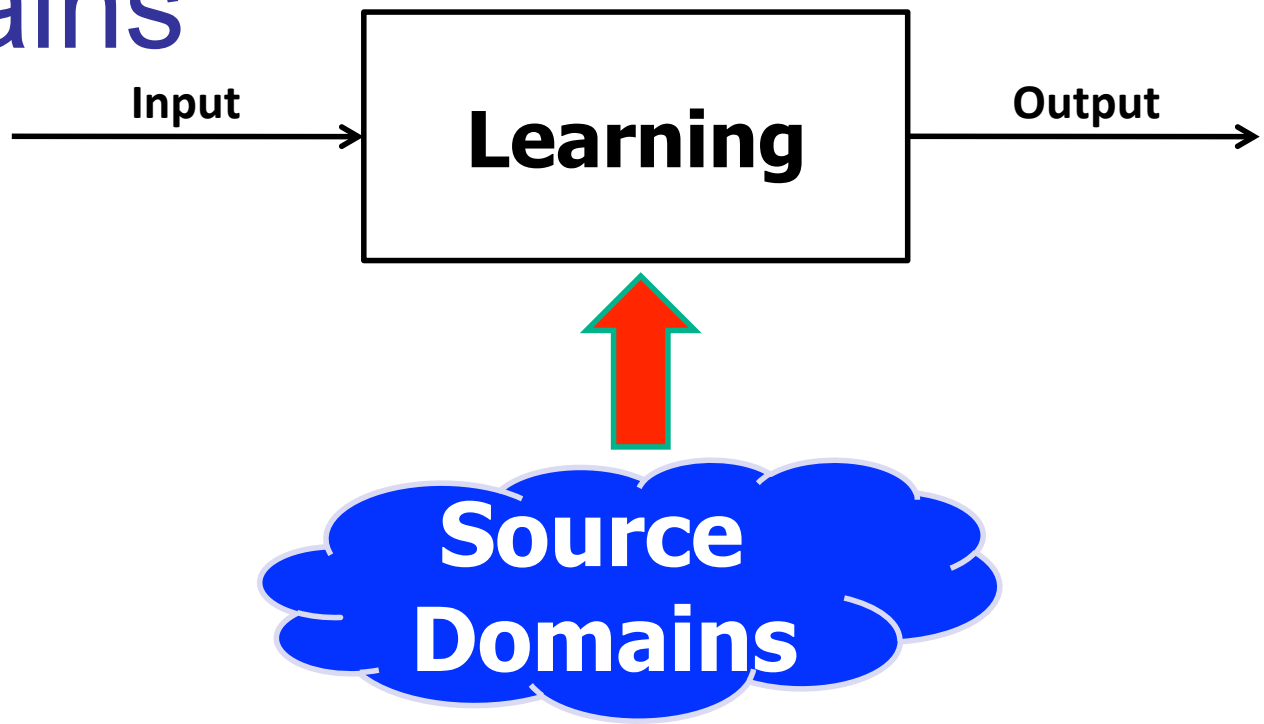
When distributions are different



Classification Accuracy (+, -)

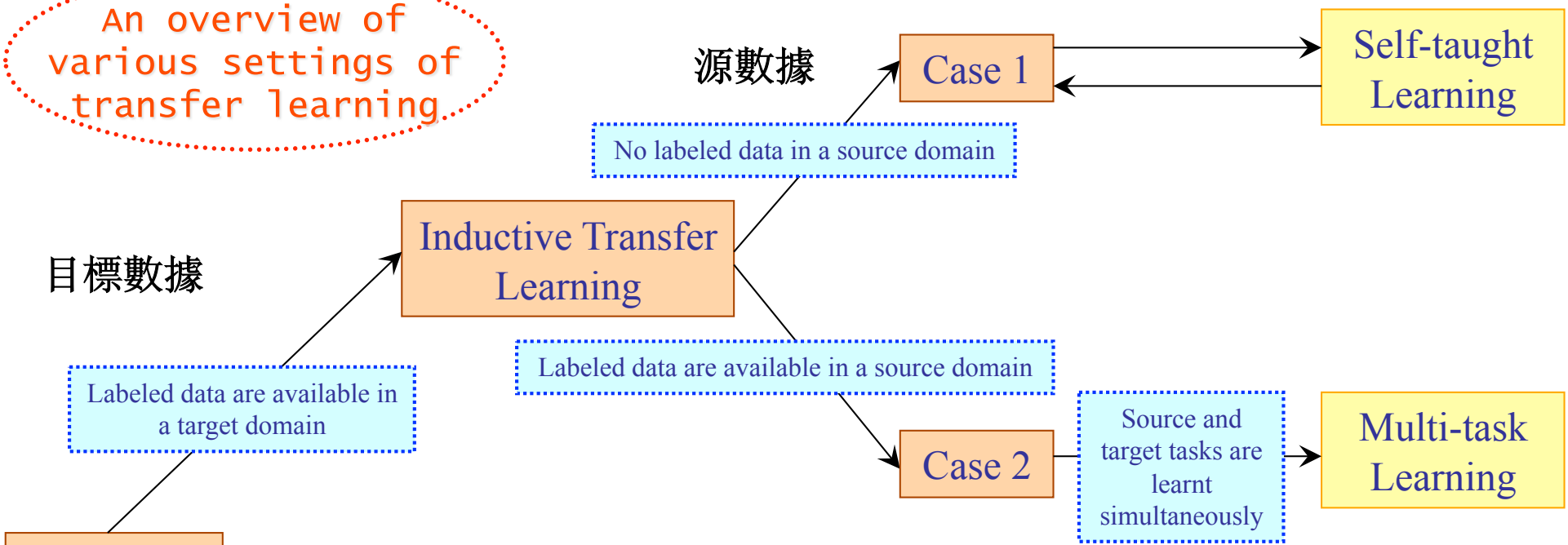
- Training: 90%
- Testing: 60%

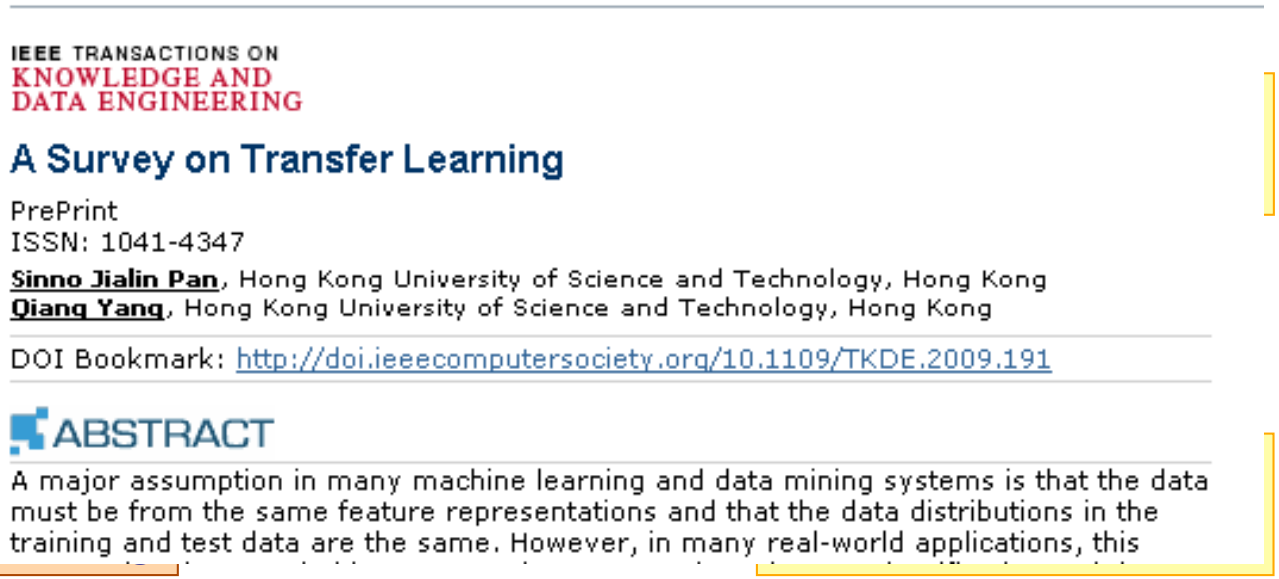
Transfer Learning: Source Domains



	Source Domain	Target Domain
Training Data	Labeled/Unlabeled	Labeled/Unlabeled
Test Data		Unlabeled

An overview of various settings of transfer learning





IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING

A Survey on Transfer Learning

PrePrint
ISSN: 1041-4347

Sinno Jialin Pan, Hong Kong University of Science and Technology, Hong Kong
Qiang Yang, Hong Kong University of Science and Technology, Hong Kong

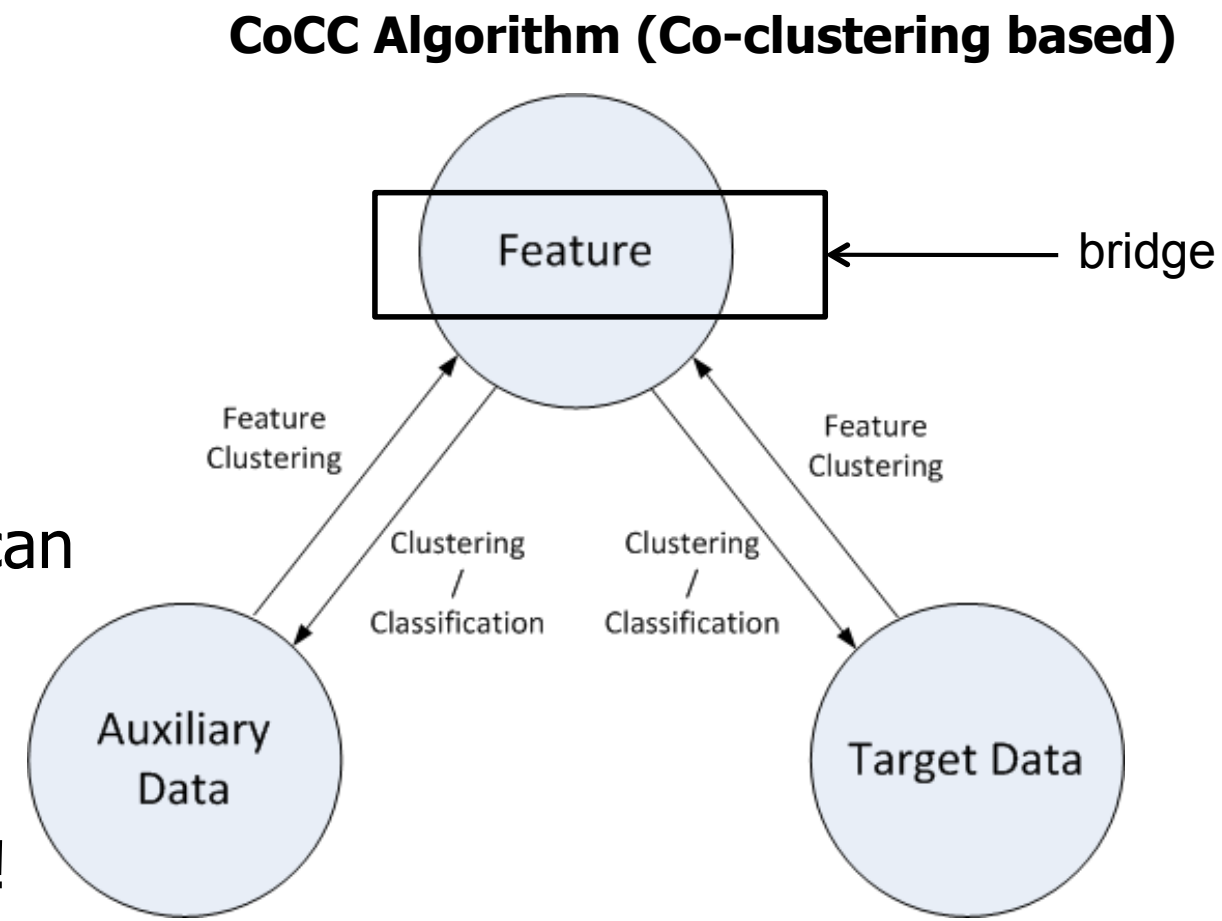
DOI Bookmark: http://doi.ieeecomputersociety.org/10.1109/TKDE.2009.191

ABSTRACT

A major assumption in many machine learning and data mining systems is that the data must be from the same feature representations and that the data distributions in the training and test data are the same. However, in many real-world applications, this

Feature-based Transfer Learning (Dai, Yang et al. ACM KDD 2007)

- Source:
  - Many labeled instances
- Target:
  - All unlabeled instances
- Distributions
  - Feature spaces can be different, but have overlap
  - Same classes
  - P(X,Y): different!



Co-Clustering based Classification (on 20 News Group dataset)

- In test error rate

Using Transfer Learning

Data Set	NBC	SVM	TSVM	SGT	CoCC
real vs simulated	0.259	0.266	0.130	0.130	**0.120**
auto vs aviation	0.150	0.228	0.102	0.087	**0.068**
rec vs talk	0.235	0.233	0.040	0.091	**0.035**
rec vs sci	0.165	0.212	0.062	0.062	**0.055**
comp vs talk	0.024	0.103	0.097	0.028	**0.020**
comp vs sci	0.207	0.317	0.183	0.279	**0.130**
comp vs rec	0.072	0.165	0.098	0.047	**0.042**
sci vs talk	0.226	0.226	0.108	0.083	**0.054**
orgs vs places	0.377	0.454	0.436	0.385	**0.320**
people vs places	0.216	0.266	0.231	0.192	**0.174**
orgs vs people	0.289	0.297	0.297	0.306	**0.236**

Talk Outline

- Transfer Learning: A quick introduction
- Link prediction and collaborative filtering problems
- Transfer Learning for Sparsity Problem
  - Codebook Method
  - CST Method
  - Collective Link Prediction Method
- Conclusions and Future Works

A Real World Study [Leskovec-Horvitz WWW '08]

- Who talks to whom on MSN messenger
  - Network: 240M nodes, 1.3 billion edges
- Conclusions:
  - Average path length is 6.6
  - 90% of nodes is reachable <8 steps

Local Network Structures

- Link Prediction
  - A form of Statistical Relational Learning (Taskar and Getoor)
  - Object classification: predict category of an object based on its attributes and links
    - Is this person a student?
  - Link classification: predict type of a link
    - Are they co-authors?
  - Link existence: predict whether a link exists or not

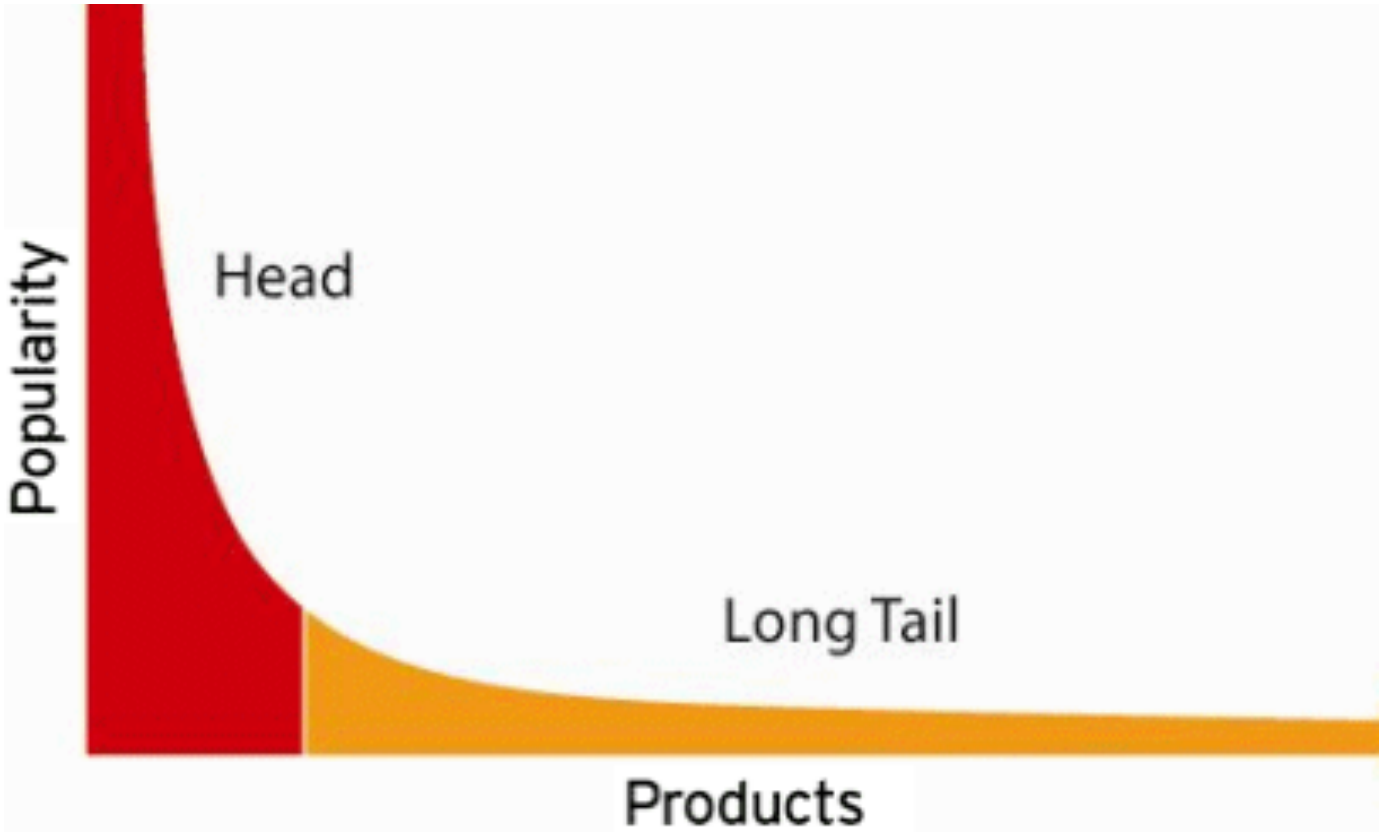
(credit: Jure Leskovec, ICML '09)

Link Prediction

- Task: predict missing links in a network
- Main challenge: Network Data Sparsity

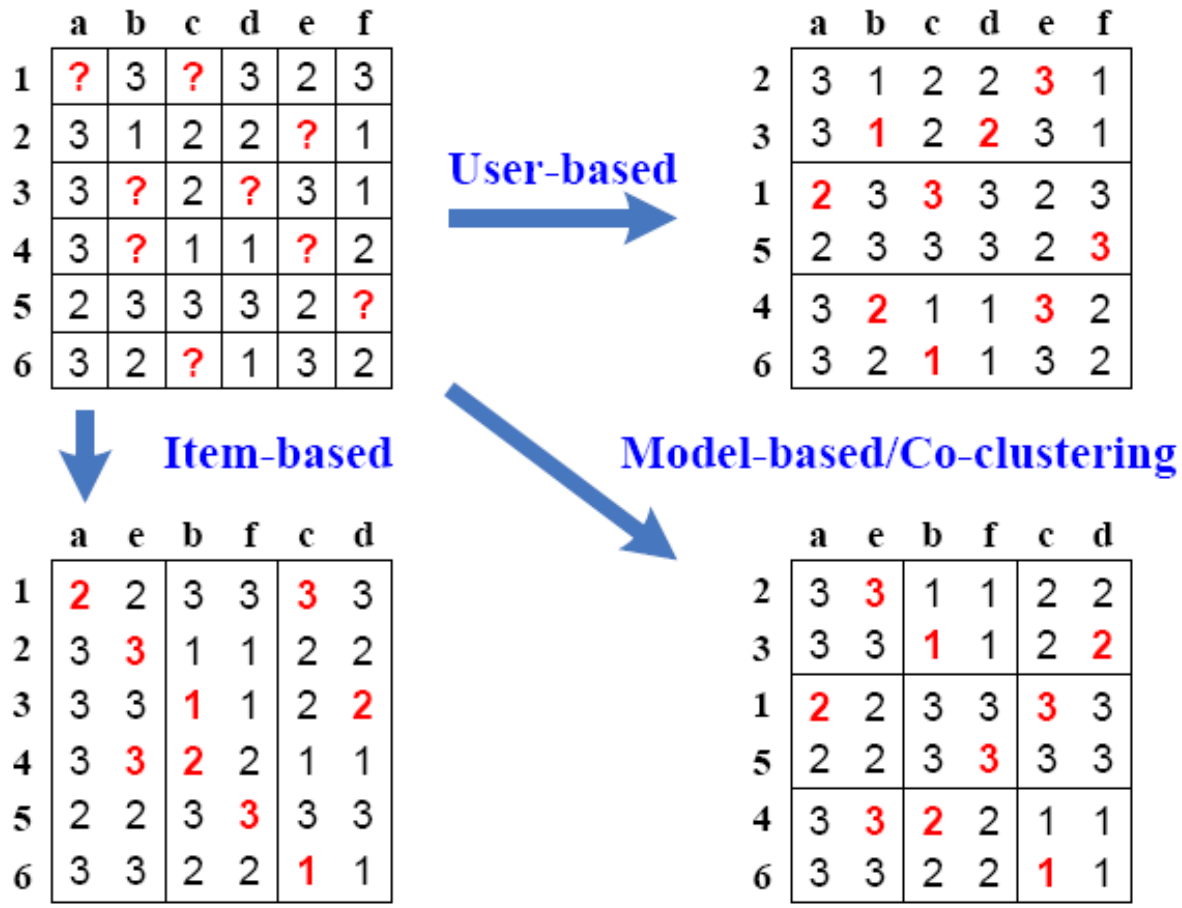
Long Tail in Era of Recommendation

- Help users discover novel/rare items
  - The long-tail ➔ recommendation systems



Essentials of Collaborative Filtering

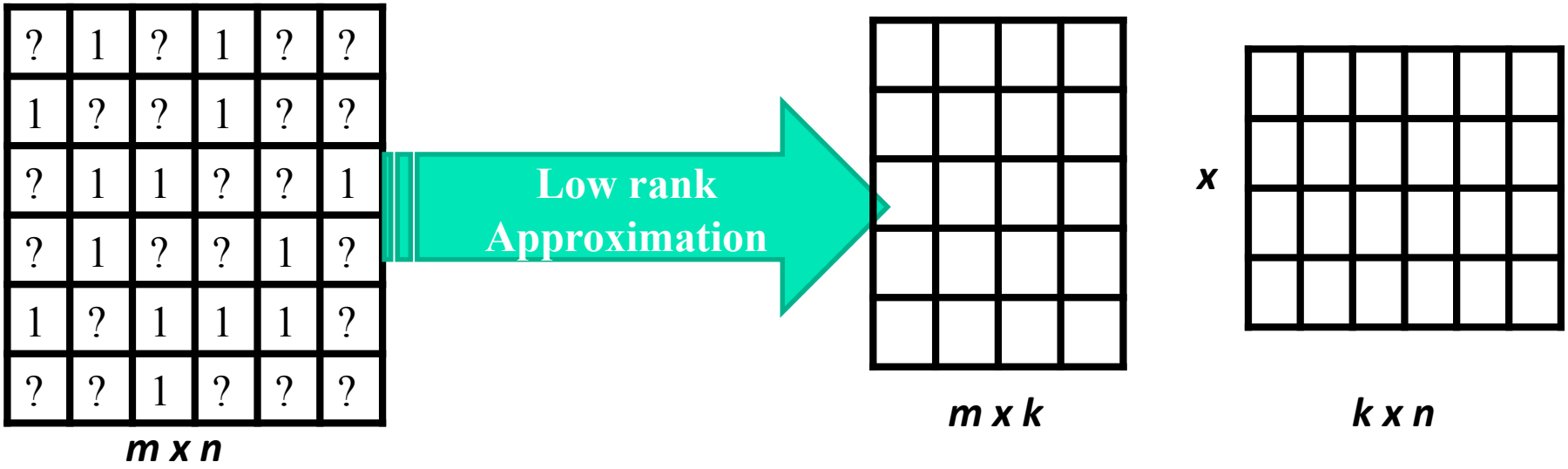
- Discover latent user/item groups by (co)-clustering
- Share ratings within clusters to fill in missing values



Matrix Factorization model for Link Prediction

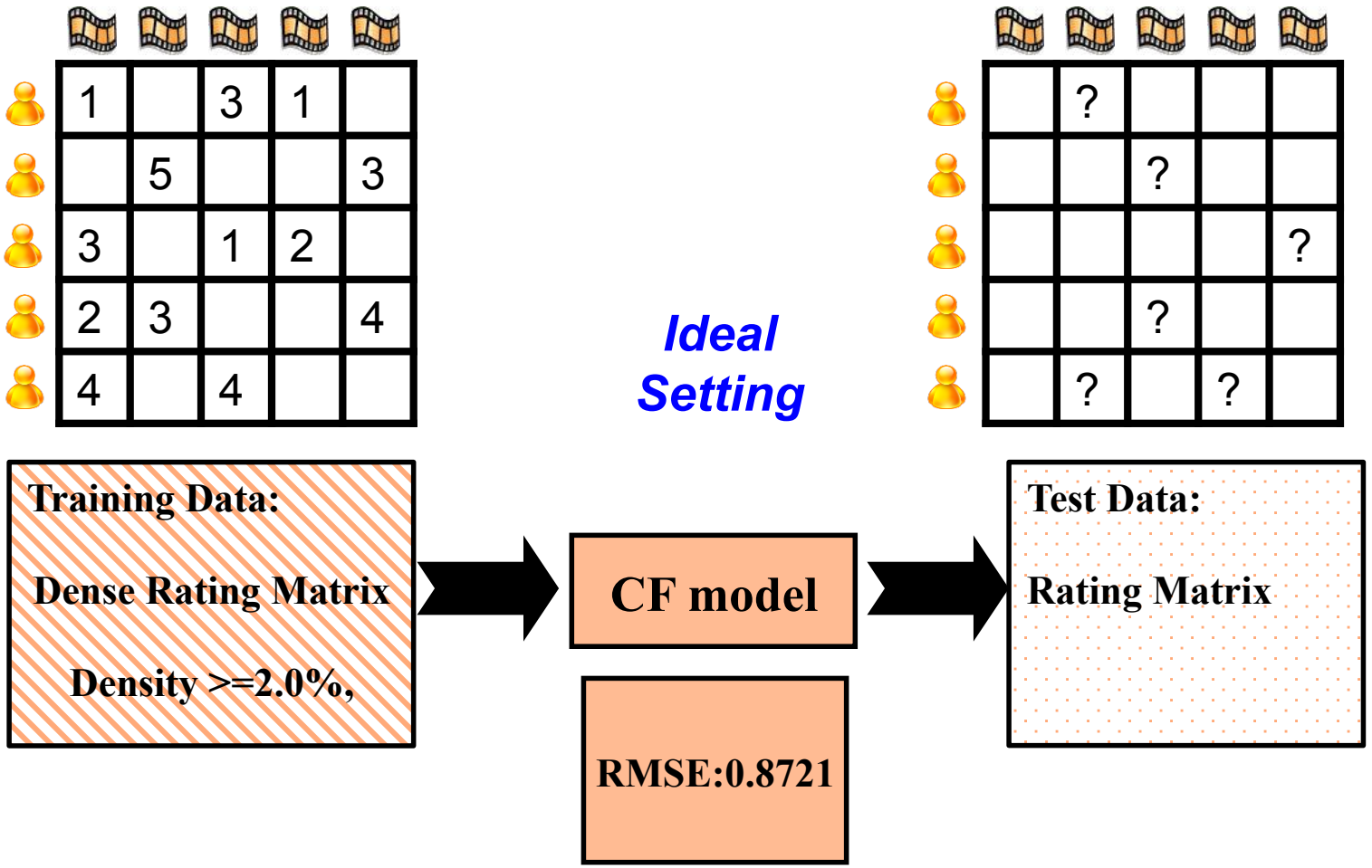
- We are seeking a low rank approximation for our target matrix

$$X \approx UV^T$$

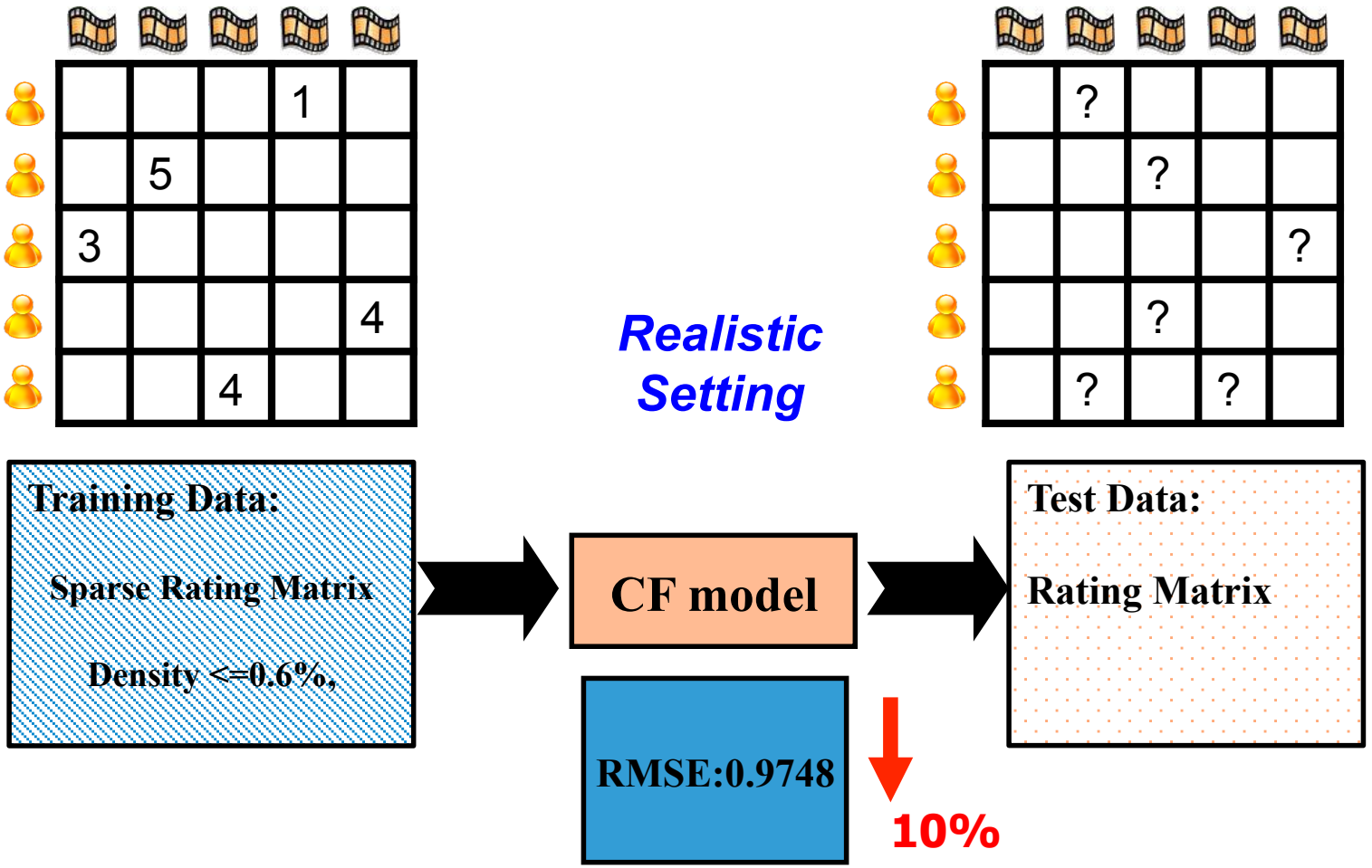


- Such that the unknown value can be predicted by $\hat{X} = UV^T$

Examples: Collaborative Filtering



Data Sparsity in Collaborative Filtering

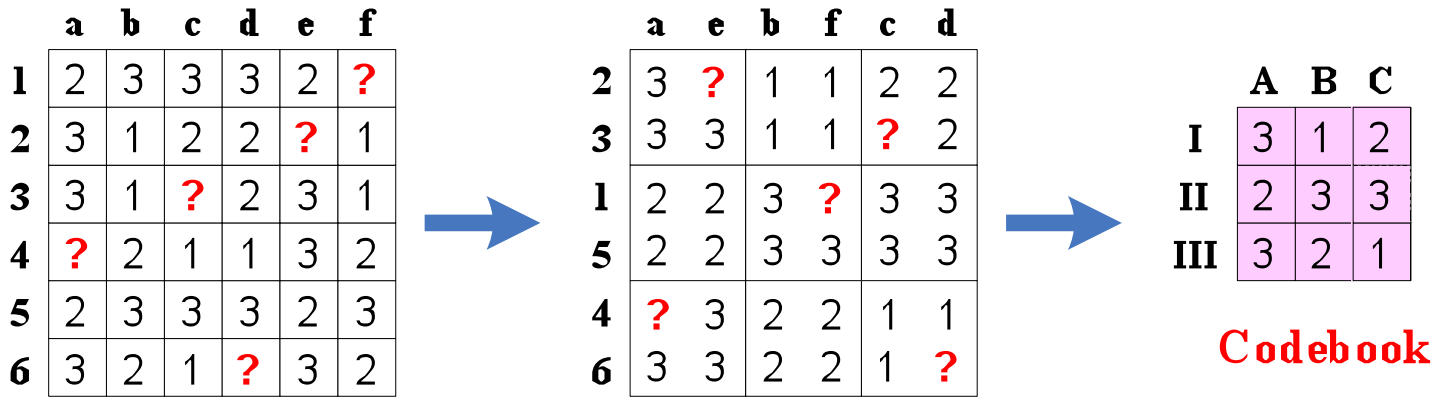


Codebook Transfer

- Bin Li, Qiang Yang, Xiangyang Xue.
- Can Movies and Books Collaborate? Cross-Domain Collaborative Filtering for Sparsity Reduction.
- In *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence* (**IJCAI '09**), Pasadena, CA, USA, July 11-17, 2009.

Codebook Construction

- **Definition 2.1** (Codebook). A $k \times l$ matrix which compresses the cluster-level rating patterns of k user clusters and l item clusters.



	a	b	c	d	e	f
1	2	3	3	3	2	?
2	3	1	2	2	?	1
3	3	1	?	2	3	1
4	?	2	1	1	3	2
5	2	3	3	3	2	3
6	3	2	1	?	3	2

	a	e	b	f	c	d
2	3	?	1	1	2	2
3	3	3	1	1	?	2
1	2	2	3	?	3	3
5	2	2	3	3	3	3
4	?	3	2	2	1	1
6	3	3	2	2	1	?

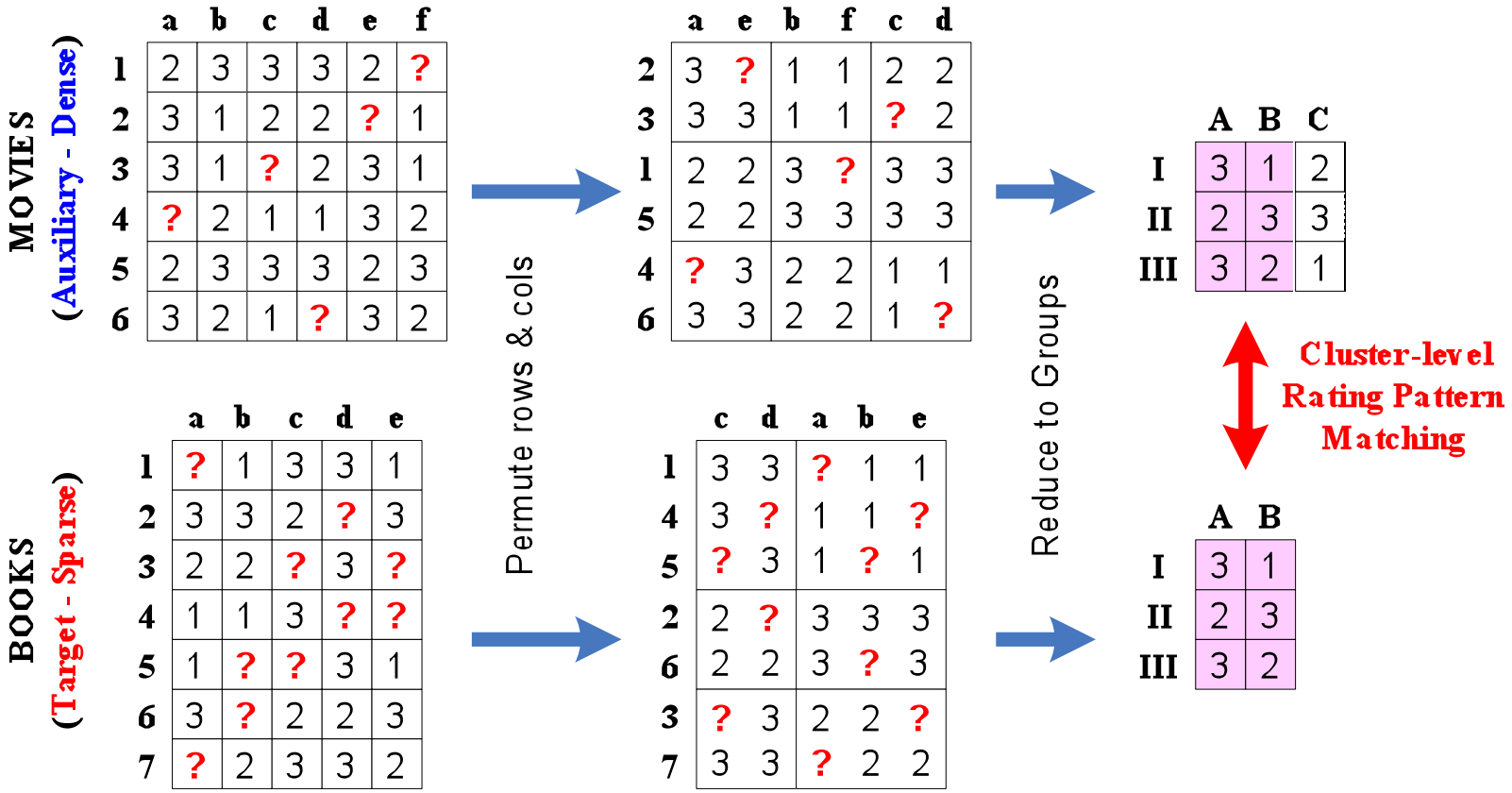
	A	B	C
I	3	1	2
II	2	3	3
III	3	2	1

Codebook

- Codebook: User prototypes rate on item prototypes
- Encoding: Find prototypes for users and items and get indices
- Decoding: Recover rating matrix based on codebook and indices

Knowledge Sharing via Cluster-Level Rating Matrix

- Source (Dense): Encode cluster-level rating patterns
- Target (Sparse): Map users/items to the encoded prototypes

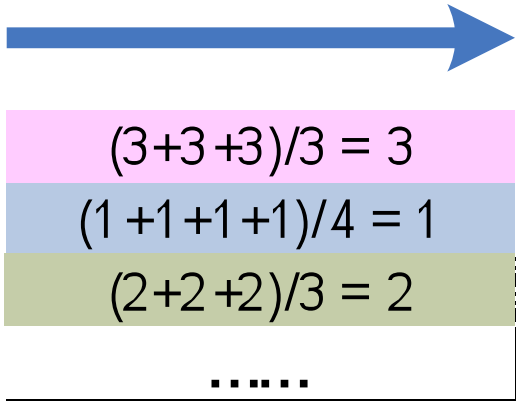


Step 1: Codebook Construction

- Co-cluster rows (users) and columns (items) in $\mathbf{X}_{aux}$
- Get user/item cluster indicators $\mathbf{U}_{aux} \in \{0, 1\}^{n \times k}$, $\mathbf{V}_{aux} \in \{0, 1\}^{m \times l}$

$$\mathbf{B} = [\mathbf{U}_{aux}^{\top} \mathbf{X}_{aux} \mathbf{V}_{aux}] \oslash [\mathbf{U}_{aux}^{\top} \mathbf{1}\mathbf{1}^{\top} \mathbf{V}_{aux}]$$

	a	e	b	f	c	d
2	3	?	1	1	2	2
3	3	3	1	1	?	2
1	2	2	3	?	3	3
5	2	2	3	3	3	3
4	?	3	2	2	1	1
6	3	3	2	2	1	?



	A	B	C
I	3	1	2
II	2	3	3
III	3	2	1

Codebook B

Step 2: Codebook Transfer

- Objective

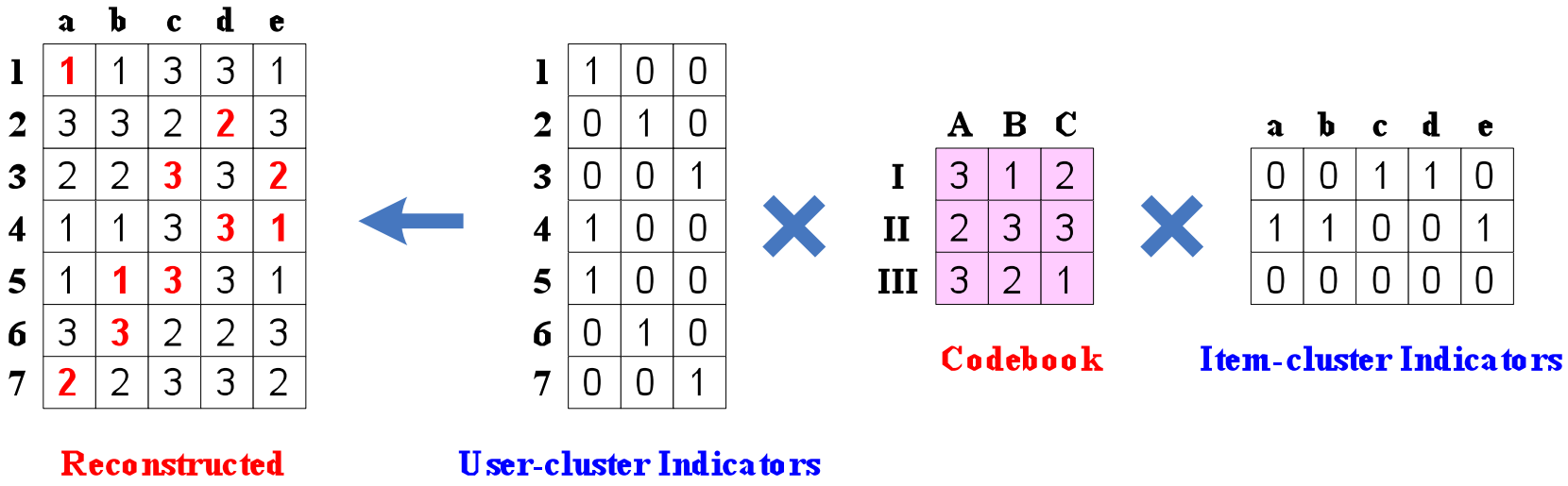
  Expand target matrix, while minimizing the difference between $\mathbf{X}_{tgt}$ and the reconstructed one

$$\min_{\substack{\mathbf{U}_{tgt}\in\{0,1\}^{p\times k}\\ \mathbf{V}_{tgt}\in\{0,1\}^{q\times l}}} \left\| [\mathbf{X}_{tgt} - \mathbf{U}_{tgt}\mathbf{B}\mathbf{V}_{tgt}^{\top}] \circ \mathbf{W} \right\|_F^2$$

$$\text{s.t.} \quad \mathbf{U}_{tgt}\mathbf{1} = \mathbf{1}, \mathbf{V}_{tgt}\mathbf{1} = \mathbf{1}$$

- User/item cluster indicators $\mathbf{U}_{tgt}$ and $\mathbf{V}_{tgt}$ for $\mathbf{X}_{tgt}$
- Binary weighting matrix $\mathbf{W}$ for observed ratings in $\mathbf{X}_{tgt}$
- Alternate greedy searches for $\mathbf{U}_{tgt}$ and $\mathbf{V}_{tgt}$ to a local minimum

Codebook Transfer



- Each user/item in $\mathbf{X}_{tgt}$ matches to a prototype in **B**
- Duplicate certain rows & columns in **B** to reconstruct $\mathbf{X}_{tgt}$
- Codebook is indeed a two-sided data representation

Experimental Setup

- Data Sets
  - EachMovie (Auxiliary): 500 users × 500 movies
  - MovieLens (Target): 500 users × 1000 movies
  - Book-Crossing (Target): 500 users × 1000 books

- Compared Methods
  - Pearson Correlation Coefficients (PCC)
  - Scalable Cluster-based Smoothing (CBS)
  - Weighted Low-rank Approximation (WLR)
  - **Codebook Transfer (CBT)**

- Evaluation Protocol
  - First 100/200/300 users for training; last 200 users for testing
  - Given 5/10/15 observable ratings for each test user

Experimental Results (1): Books → Movies

- MAE Comparison on MovieLens
  - average over 10 sampled test sets
  - Lower is better

Training Set	Method	Given5	Given10	Given15
ML100	PCC	0.930	0.883	0.873
	CBS	0.874	0.845	0.839
	WLR	0.915	0.875	0.890
	CBT	**0.840**	**0.802**	**0.786**
ML200	PCC	0.905	0.878	0.878
	CBS	0.871	0.833	0.828
	WLR	0.941	0.903	0.883
	CBT	**0.839**	**0.800**	**0.784**
ML300	PCC	0.897	0.882	0.885
	CBS	0.870	0.834	0.819
	WLR	1.018	0.962	0.938
	CBT	**0.840**	**0.801**	**0.785**

Limitations of Codebook Transfer

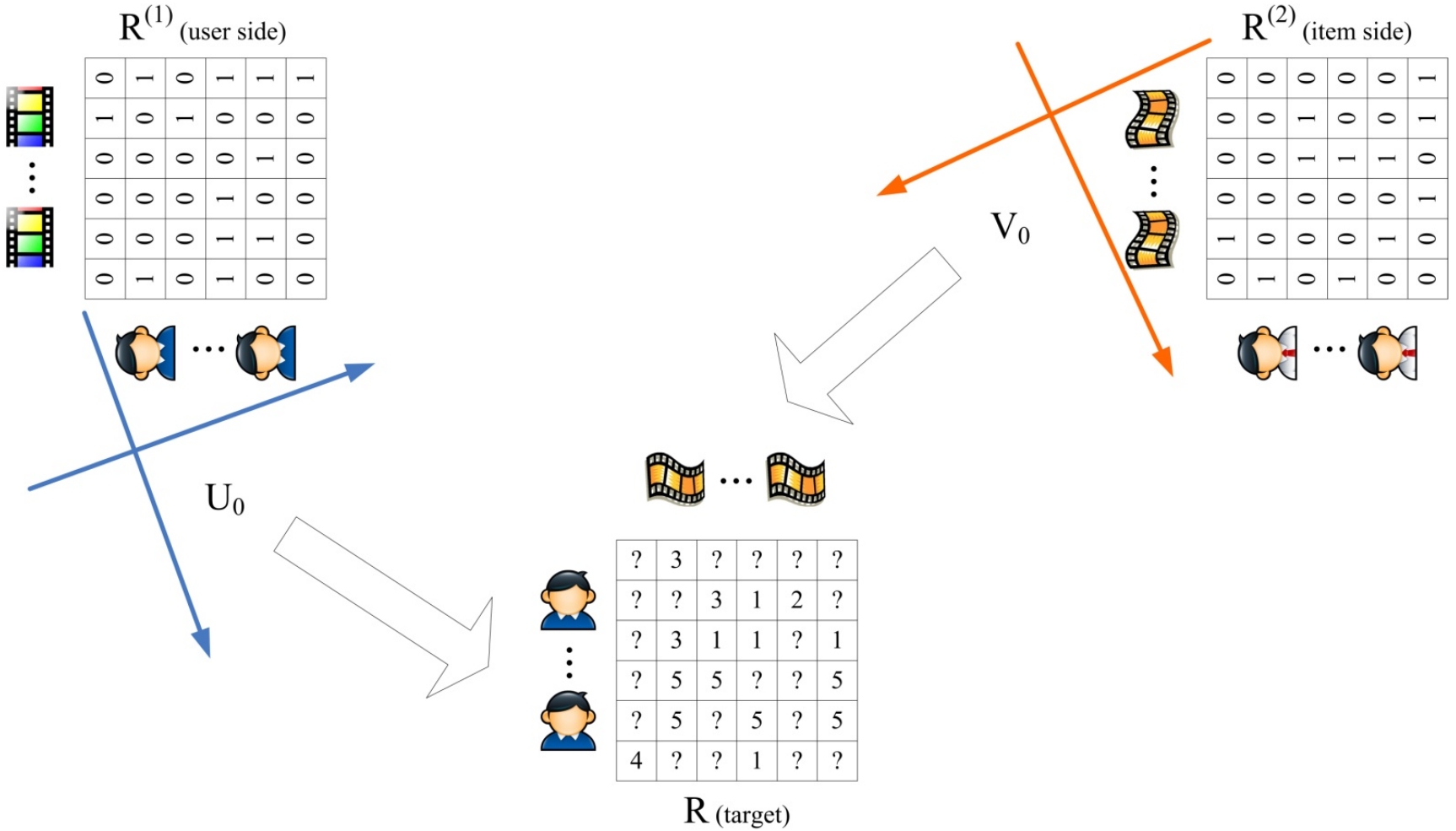
- Same rating range
  - Source and target data must have the same range of ratings [1, 5]
- Homogenous dimensions
  - User and item dimensions must be similar
- In reality
  - Range of ratings can be 0/1 or [1,5]
  - User and item dimensions may be very different

Coordinate System Transfer

- Weike Pan, Evan Xiang, Nathan Liu and Qiang Yang.
- Transfer Learning in Collaborative Filtering for Sparsity Reduction.
- In Proceedings of the 24th AAAI Conference on Artificial Intelligence (**AAAI-10**). Atlanta, Georgia, USA. July 11-15, 2010.

Our Solution: Coordinate System Transfer

- Step 1: Coordinate System Construction ($\mathbf{U}_0, \mathbf{V}_0$)
- Step 2: Coordinate System Adaptation



Step 1: Coordinate System Adaptation

- Adapt the discovered coordinate systems from the auxiliary domain to the target domain,

$$\min_{\mathbf{U},\mathbf{V},\mathbf{B}} ||\mathbf{Y} \odot (\mathbf{R} - \mathbf{U}\mathbf{B}\mathbf{V}^T)|| + \frac{\rho_u}{2}||\mathbf{U} - \mathbf{U}_0||_F^2 + \frac{\rho_v}{2}||\mathbf{V} - \mathbf{V}_0||_F^2$$
$$\text{s.t.} \quad \mathbf{U}^T\mathbf{U} = \mathbf{I}, \mathbf{V}^T\mathbf{V} = \mathbf{I}$$

- The effect from the auxiliary domain
  - Initialization: take $\mathbf{U}_0, \mathbf{V}_0$ as seed model in target domain,
  - Regularization: $||\mathbf{U} - \mathbf{U}_0||_F^2, ||\mathbf{V} - \mathbf{V}_0||_F^2$

Algorithm

Algorithm 1 CST: Coordinate System Transfer.

Input: The target data $\mathbf{R}$, the auxiliary data $\mathbf{R}^{(1)}$, $\mathbf{R}^{(2)}$

Output: $\mathbf{U}$, $\mathbf{V}$, $\mathbf{B}$.

Step 1. Apply sparse SVD on auxiliary data $\mathbf{R}^{(1)}, \mathbf{R}^{(2)}$, and obtain two principle coordinate systems $\mathbf{U}_0 = \mathbf{U}^{(1)}$, $\mathbf{V}_0 = \mathbf{V}^{(2)}$. Initialize the target coordinate systems with $\mathbf{U} = \mathbf{U}_0$, $\mathbf{V} = \mathbf{V}_0$.

repeat

Step 2.1. Fix $\mathbf{U}$, $\mathbf{V}$, and estimate $\mathbf{B}$ from $||\mathbf{Y} \odot (\mathbf{R} - \mathbf{UBV}^T)|| = 0$.

Step 2.2. Fix $\mathbf{B}$, and update $\mathbf{U}$, $\mathbf{V}$ via alternative gradient descent method on Grassmann manifold.

until Convergence

Data Sets and Evaluation Metrics

- Data sets (extracted from MovieLens and Netflix)

Data set		Form	Sparsity
R	target (training)	1-5 (explicit feedback)	<1.0%
	target (test)	1-5 (explicit feedback)	11.3%
$\mathbf{R}^{(1)}$	auxiliary (user side)	0/1 (implicit feedback)	10.0%
$\mathbf{R}^{(2)}$	auxiliary (item side)	0/1 (implicit feedback)	9.5%

- Mean Absolute Error (MAE) and Root Mean Square Error (RMSE),

$$MAE = \sum_{(u,i,r_{ui})\in T_E} |r_{ui} - \hat{r}_{ui}|/|T_E|$$

$$RMSE = \sqrt{\sum_{(u,i,r_{ui})\in T_E} (r_{ui} - \hat{r}_{ui})^2/|T_E|}$$

Where $\hat{r}_{ui}$ and r_{ui} are the true and predicted ratings, respectively, and $|T_E|$ is the number of test ratings.

Baselines and Parameter Settings

- Baselines
  - Average Filling
  - Latent Matrix Factorization (Bell and Koren, ICDM07)
  - Collective Matrix Factorization (Singh and Gordon, KDD08)
  - OptSpace (Keshavan, Montanari and Oh, NIPS10)
- Average results over 10 random trials are reported

Results(1/2)

	Observed (sparsity)	Without Transfer AF	Without Transfer LFM	With Transfer CMF	With Transfer CST
MAE	10 (0.2%)	0.7764 ± 0.0008	0.8934 ± 0.0005	0.7642 ± 0.0024	**0.7481** ± 0.0014
	20 (0.4%)	0.7430 ± 0.0006	0.8243 ± 0.0019	0.7238 ± 0.0012	**0.7056** ± 0.0008
	30 (0.6%)	0.7311 ± 0.0005	0.7626 ± 0.0008	0.7064 ± 0.0008	**0.6907** ± 0.0006
	40 (0.8%)	0.7248 ± 0.0004	0.7359 ± 0.0008	0.6972 ± 0.0007	**0.6835** ± 0.0008
RMSE	10 (0.2%)	0.9853 ± 0.0011	1.0830 ± 0.0000	0.9749 ± 0.0033	**0.9649** ± 0.0019
	20 (0.4%)	0.9430 ± 0.0006	1.0554 ± 0.0016	0.9261 ± 0.0014	**0.9059** ± 0.0013
	30 (0.6%)	0.9280 ± 0.0005	0.9748 ± 0.0012	0.9058 ± 0.0009	**0.8855** ± 0.0010
	40 (0.8%)	0.9202 ± 0.0003	0.9381 ± 0.0010	0.8955 ± 0.0007	**0.8757** ± 0.0011
Time Complexity		$O(p)$	$O(kpd^2 + k \max(n, m)d^3)$	same as LFM	$O(kpd^3 + kd^6)$

- Observations:
  - **CST** performs significantly better (t-test) than all baselines at all sparsity levels,
  - **Transfer learning methods** (CST, CMF) beat two non-transfer learning methods (AF, LFM).

Limitation of CST and CBT

- Different source domains are related to the target domain differently
  - Book to Movies
  - Food to Movies
- Rating bias
  - Users tend to rate items that they like
    - Thus there are more rating = 5 than rating = 2

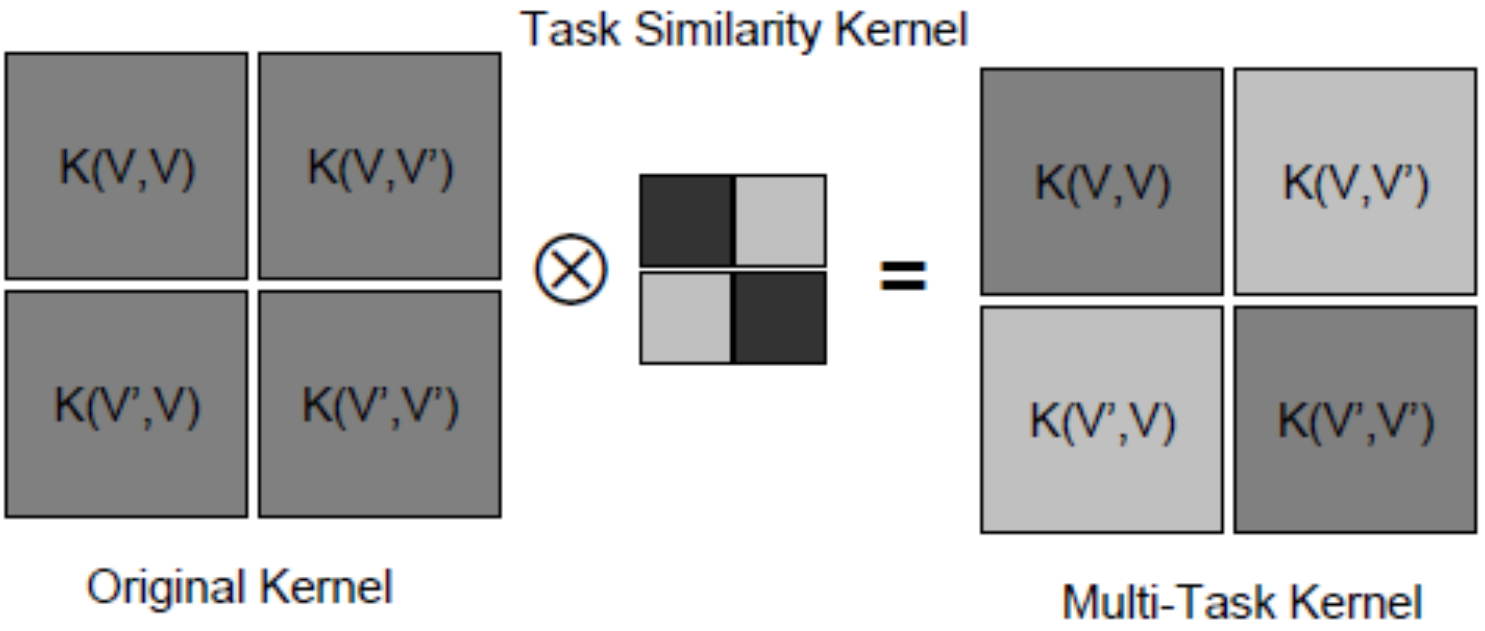
Our Solution: Collective Link Prediction (CLP)

- Jointly learn multiple domains together
  - Learning the similarity of different domains
  - consistency between domains indicates similarity.
  - Introduce a link function to correct the bias

- Bin Cao, Nathan Liu and Qiang Yang.
- Transfer Learning for Collective Link Prediction in Multiple Heterogeneous Domains.
- In Proceedings of 27th International Conference on Machine Learning (**ICML 2010**), Haifa, Israel. June 2010.

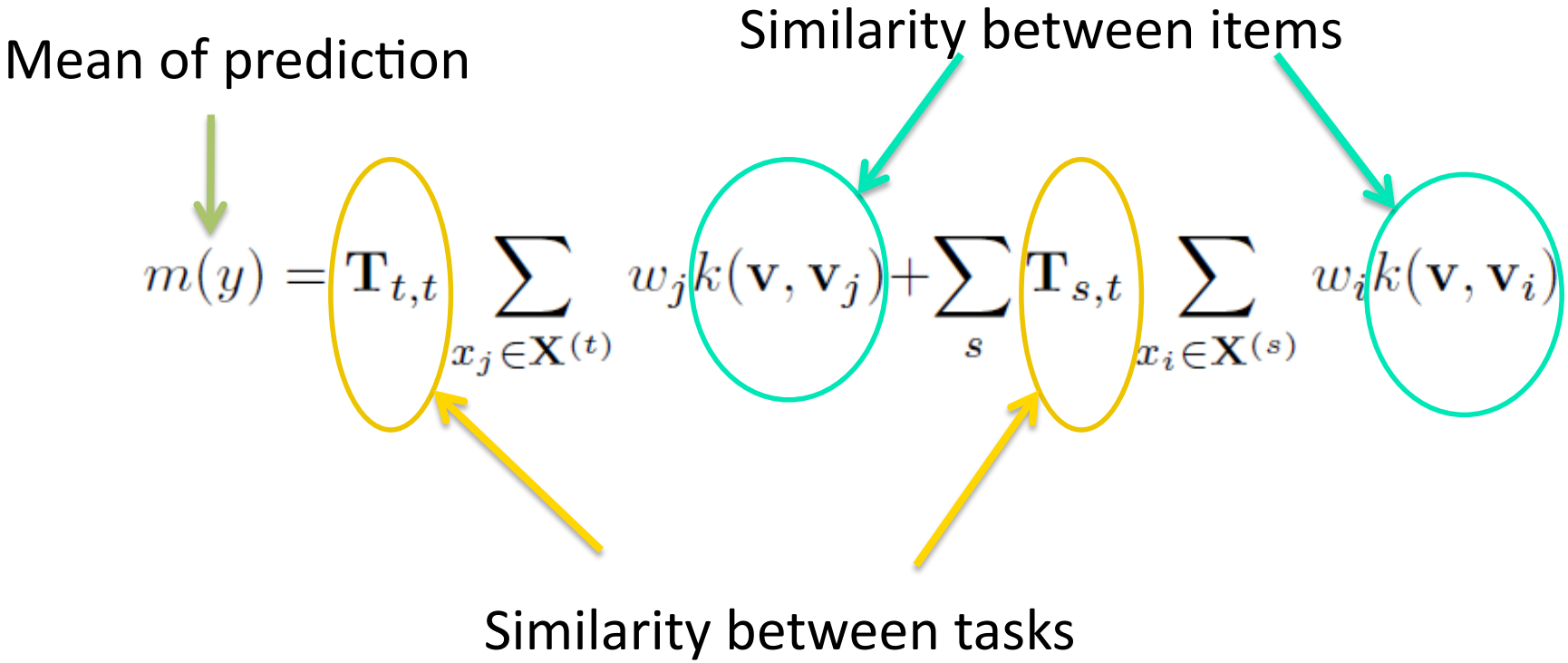
Inter-task Similarity

- Based on Gaussian process models
- Key part is the kernel modeling user relation as well as task relation



Making Prediction

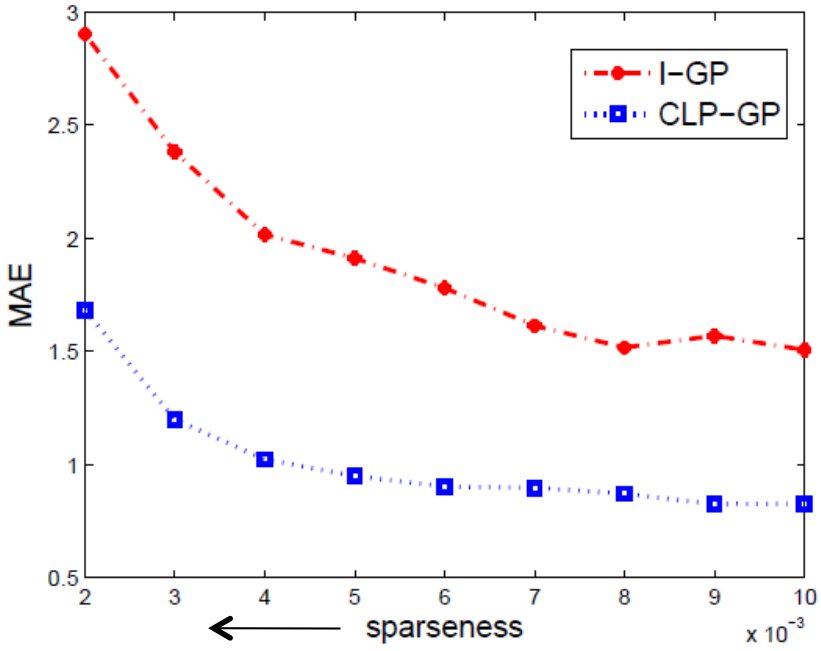
- Similar to memory-based approach

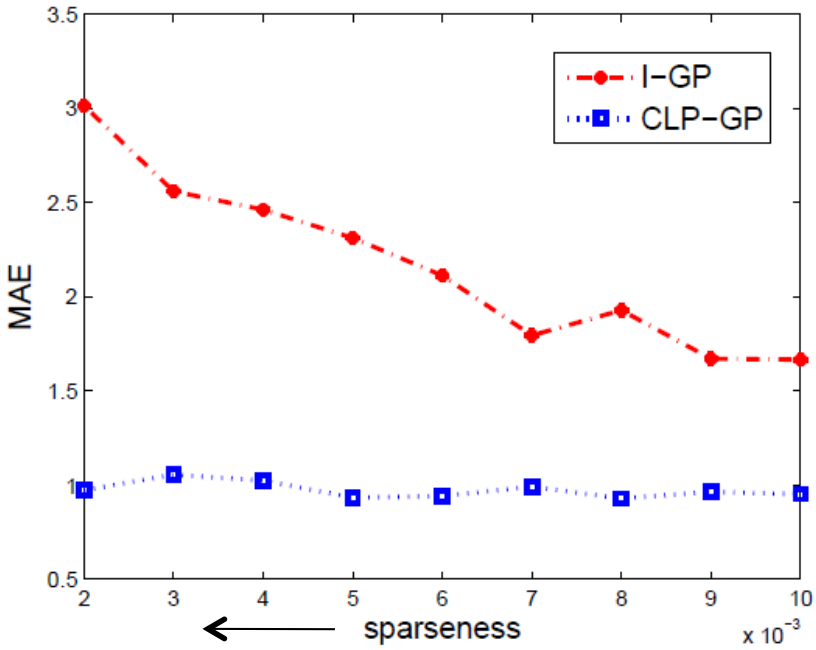


Experimental Results

	Action	Comedy	Drama	Romance	Thriller
Action	1	0.8479	0.8814	0.8953	0.9253
Comedy	0.8479	1	0.8750	0.8936	0.8422
Drama	0.8814	0.8814	1	0.9392	0.8911
Romance	0.8953	0.8936	0.9392	1	0.8862
Thriller	0.9253	0.8422	0.8911	0.8862	1

Table 2. The similarity matrix cross five link prediction tasks on MovieLens.





Conclusions and Future Work

- Transfer Learning (舉一反三)
- Link prediction is an important task in graph/network data mining
- Key Challenge: sparsity
- Transfer learning from other domains helps improve the performance

Acknowledgement

- HKUST:
  - Sinno J. Pan, Huayan Wang, Bin Cao, Evan Wei Xiang, Derek Hao Hu, Nathan Nan Liu, Vincent Wenchen Zheng
- Shanghai Jiaotong University:
  - Wenyuan Dai, Guirong Xue, Yuqiang Chen, Prof. Yong Yu, Xiao Ling, Ou Jin.
- Visiting Students
  - Bin Li (Fudan U.), Xiaoxiao Shi (Zhong Shan U.),