CoNet: Collaborative Cross Networks for Cross-Domain Recommendation

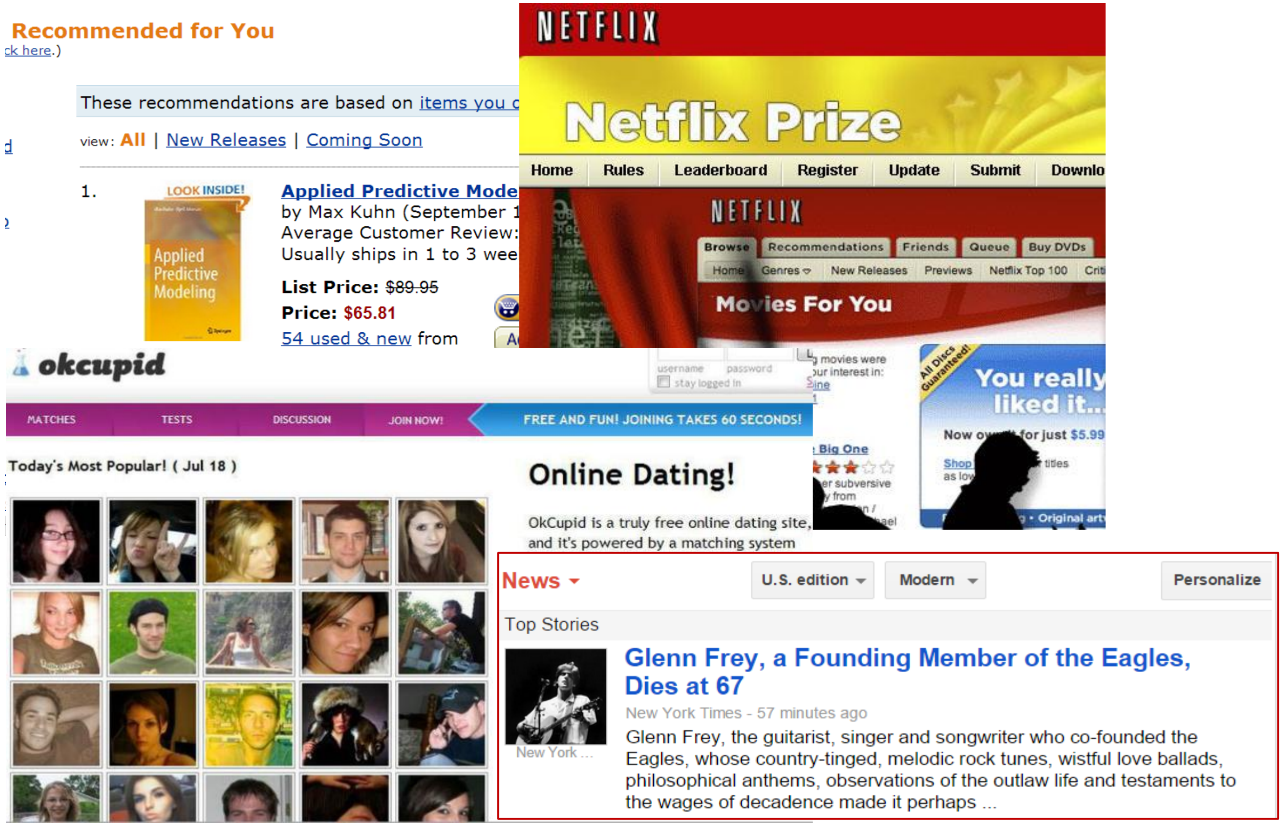
Guangneng Hu*, Yu Zhang, and Qiang Yang

CIKM 2018
Oct 22-26 (Mo-Fr),
Turin, Italy

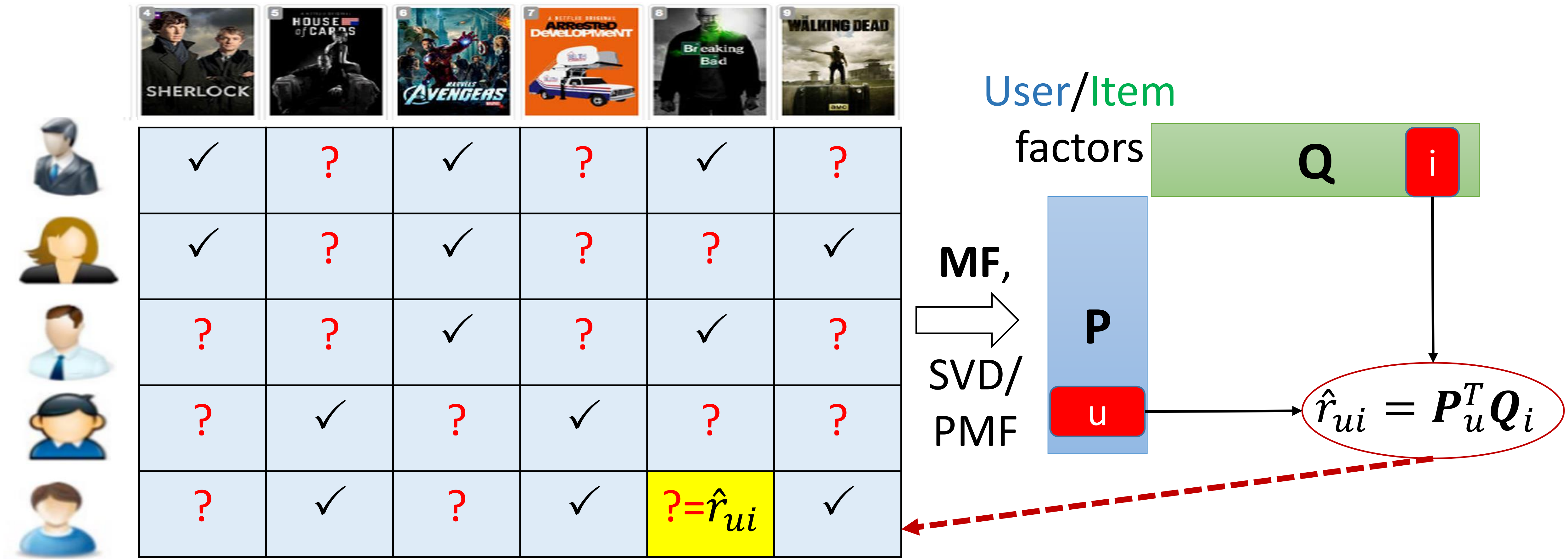
香港科技大學
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

Recommendations Are Ubiquitous: Products, Medias, Entertainment…

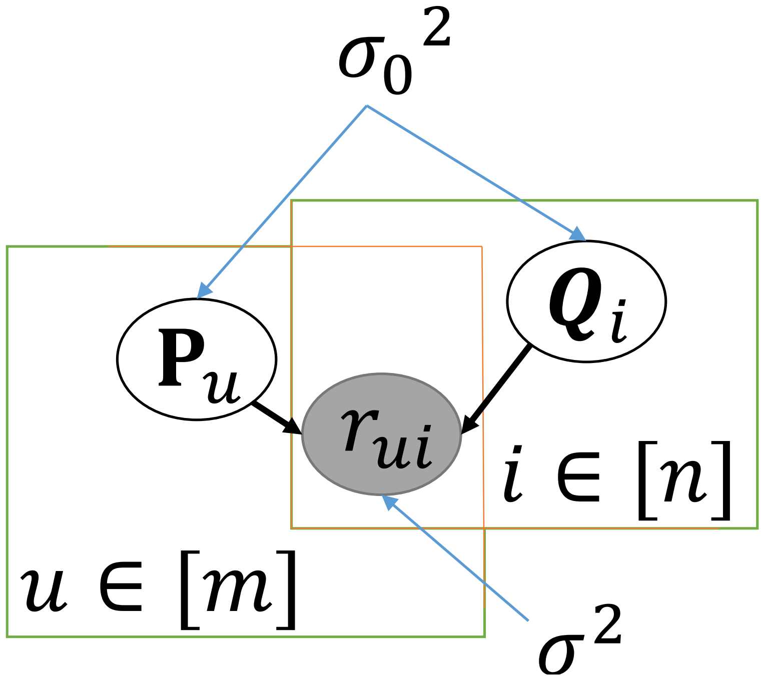
- Amazon
  - 300 million customers
  - 564 million products
- Netflix
  - 480,189 users
  - 17,770 movies
- Spotify
  - 40 million songs
- OkCupid
  - 10 million members



Typical Methods: Matrix Factorization
(Koren KDD’08, KDD 2018 TEST OF TIME award)



Probabilistic Interpretations: PMF



- The objective of matrix factorization

$$\min_{\boldsymbol{P},\boldsymbol{Q}} \sum\nolimits_{r_{ui} \neq 0} (r_{ui} - \hat{r}_{ui})^2 + \lambda(||\boldsymbol{P}||^2_{Frob} + ||\boldsymbol{Q}||^2_{Frob})$$

- Probabilistic interpretations (PMF)
  - Gaussian observations & priors
- Log posterior distribution

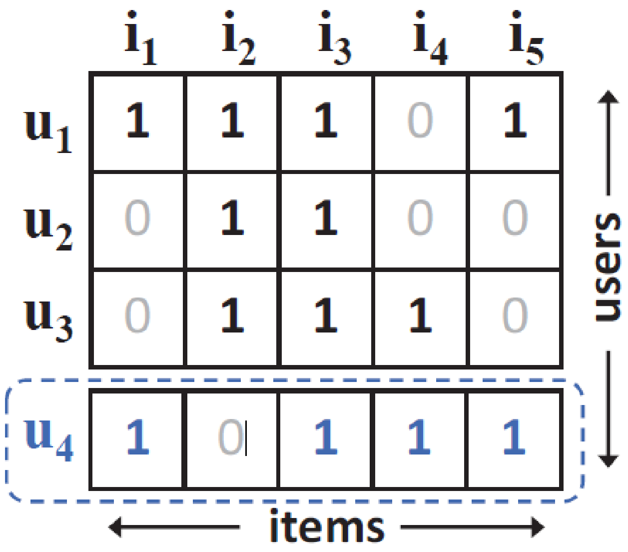
$$\ln p(\Theta|\boldsymbol{R}, \Phi) = -\frac{1}{2\sigma^2} \sum_{u,i} \delta(r_{ui})(r_{ui} - \boldsymbol{P}_u^T \boldsymbol{Q}_i)^2 - \frac{1}{2\sigma_0^2} \left(||\boldsymbol{P}||^2_{Frob} + ||\boldsymbol{Q}||^2_{Frob} \right)$$

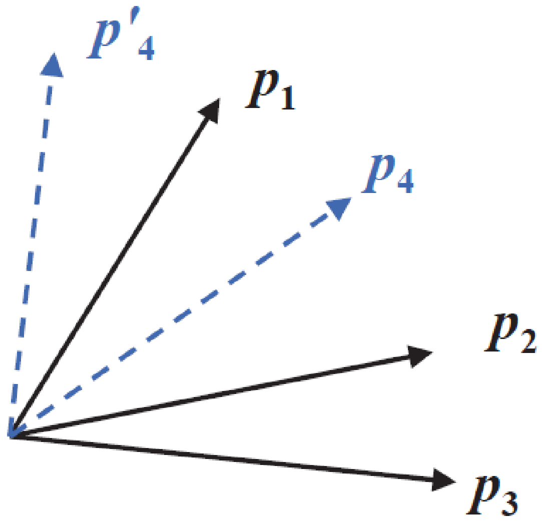
- Maximum a posteriori (MAP) estimation ←→ Minimizing sum-of-squared-errors with quadratic regularization (Loss + Regu)

Mnih & Salakhutdinov. Probabilistic matrix factorization. NIPS'07

Limited Expressiveness of MF: Example I

- Similarity of user u4:
  - Given: Sim(u4,u1) > Sim(u4,u3) > Sim(u4,u2)
  - Q: Where to put the latent factor vector p4?
- MF can not capture highly nonlinear
  - Deep learning, nonlinearity

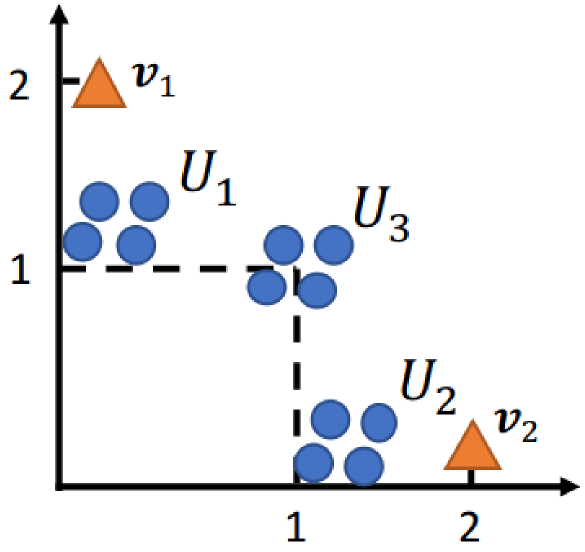


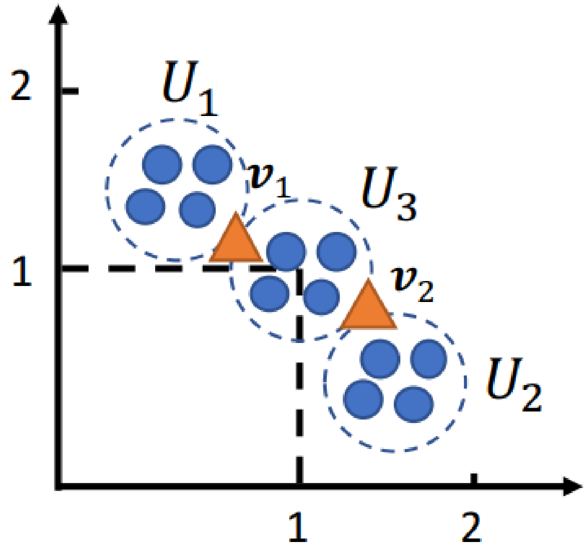


Xiangnan He et al. Neural collaborative filtering. WWW'17

Limited Expressiveness of MF: Example II

- Transitivity of user U3:
  - Given: U3 close to item v1 and v2
  - Q: Where v1 and v2 should be?
- MF can not capture transitivity
  - Metric learning, triangle inequality





Cheng-Kang Hsieh et al. Collaborative metric learning. WWW'17

Modelling Nonlinearity: Generalized Matrix Factorization

- Matrix factorization as a single layer **linear** neural network
  - Input: one-hot encodings of the user and item indices ***(u, i)***
  - Embedding: embedding matrices **(P, Q)**
  - Output: **Hadamard product** between embeddings with an **identity activation** and a fixed **all-one vector h**
- Generalized Matrix Factorization
  - Learning weights **h** instead of fixing it
  - Using non-linear activation (e.g., sigmoid) instead of identity

Hadamard product

$$\hat{r}_{u,i} = \sigma\left(\boldsymbol{h}^T(\boldsymbol{P}_u \odot \boldsymbol{Q}_i)\right)$$

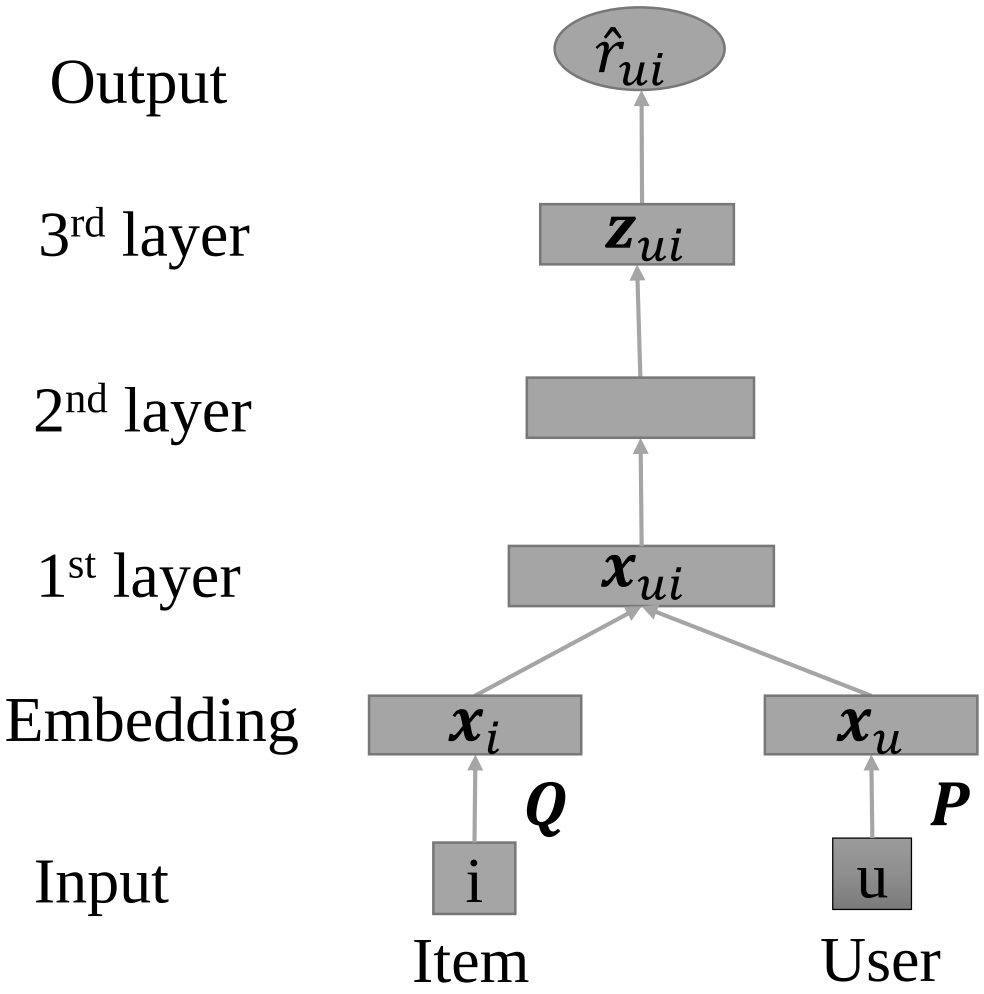
identity activation **all-one vector**

Go Deeper: Neural Collaborative Filtering

- Stack multilayer feedforward NNs to learn highly non-linear representations

$$f(\boldsymbol{x}_{ui}|\boldsymbol{P}, \boldsymbol{Q}, \theta_f) = \phi_o(\phi_L(...(\phi_1(\boldsymbol{x}_{ui}))...))$$

- Capture the complex user-item interaction relationships via the expressiveness of multilayer NNs

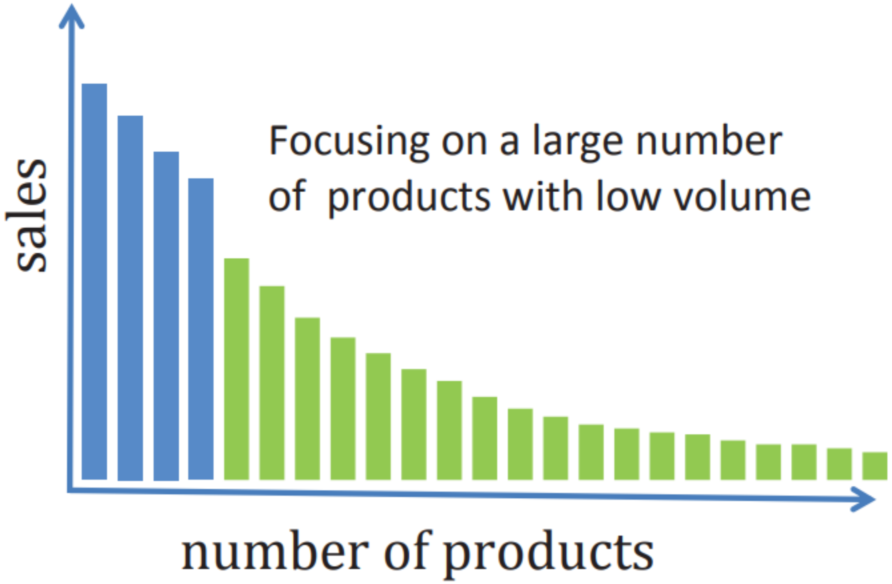


Xiangnan He et al. Neural collaborative filtering. WWW’17

Collaborative Filtering Faces Challenges: Data Sparsity and Long Tail

- Data sparsity
  - Netflix
    - **1.225%**
  - Amazon
    - **0.017%**
- Long tail
  - Pareto principle (80/20 rule):
    - A small proportion (e.g., 20%) of products generate a large proportion (e.g., 80%) of sales

2	?	2	?	5	?
5	?	4	?	?	1
?	?	5	?	2	?
?	1	?	5	?	?
?	5	?	1	?	4



A Solution: Cross-Domain Recommendation

- Two domains
  - A target domain (e.g., Books domain) **R**={(*u*,*i*)},
  - A related source domain (e.g., Movies domain) {(*u*,*j*)}
- Probability of a user prefers an item by two factors
  - His/her individual preferences (in the target domain), and
  - His/her behavior in a related source domain

	The Name of the Wind	American Gods	The Lord of the Rings (2001)	The Matrix (1999)	Star Wars (1977)
Alice	5		?		
Bob	4		5		
Carol		4			3
Dave			5	5	

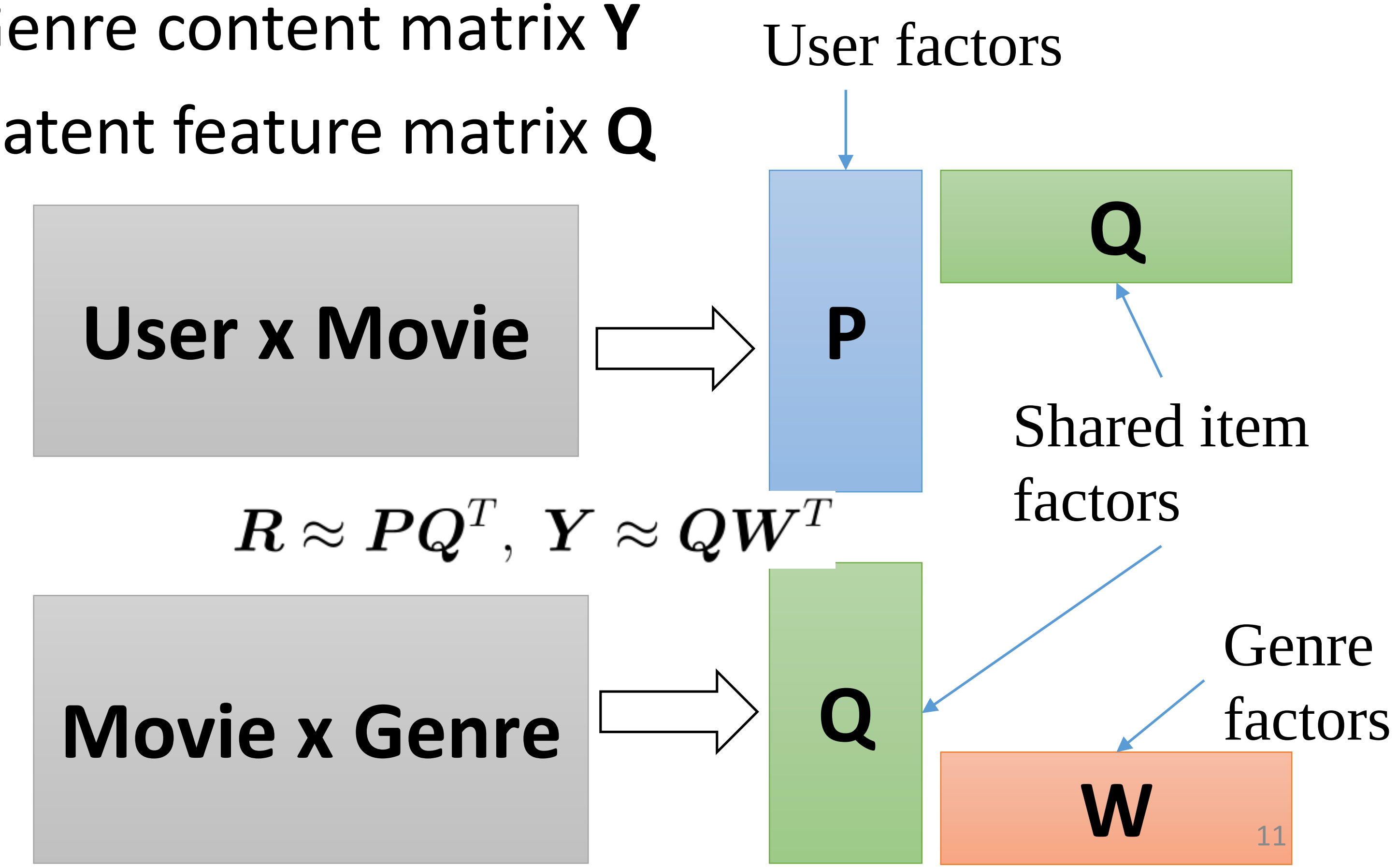
$\mathcal{U}_A$ $\mathcal{U}_B$ $\mathcal{I}_A$ $\mathcal{I}_B$

$$\hat{r}_{ui} \triangleq p(r_{ui} = 1 | u, [j]^u)$$

Typical Methods: Collective Matrix Factorization (Singh & Gordon, KDD'08)

- User-Item interaction matrix **R**
- Relational domain: Item-Genre content matrix **Y**
- Sharing the **item-specific** latent feature matrix **Q**

movie	budget	gross	genre	year
Goodfellas	25M	47M	crime	1990
My Cousin Vinny	11M	64M	comedy	1992
...	...	...	...	...
Clue	15M	15M	comedy	1985

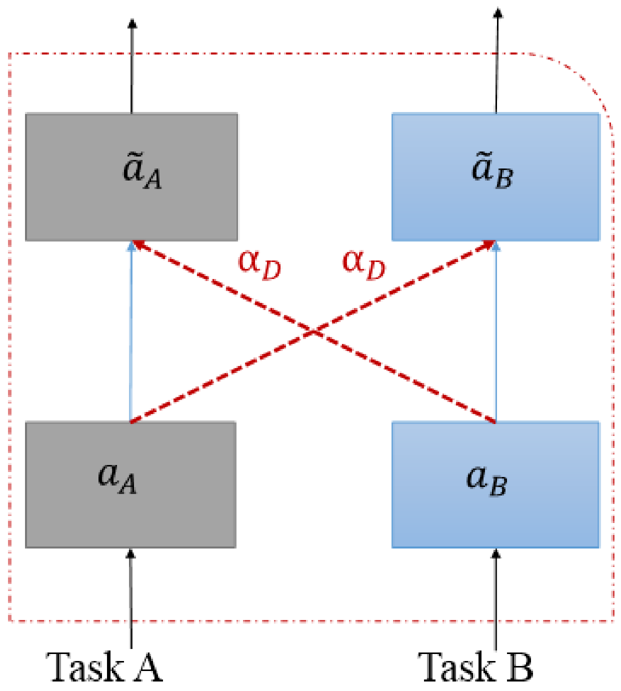


Deep Methods: Cross-Stitch Networks (CSN)

- Linear combination of activation maps from two tasks

$$\tilde{a}_A^{ij} = \alpha_S a_A^{ij} + \alpha_D a_B^{ij}, \quad \tilde{a}_B^{ij} = \alpha_S a_B^{ij} + \alpha_D a_A^{ij},$$

- Strong assumptions (SA)
  - SA 1: Representations from other network are ***equally important*** with weights being all the same scalar
  - SA 2: Representations from other network are ***all useful*** since it transfers activations from every location in a dense way



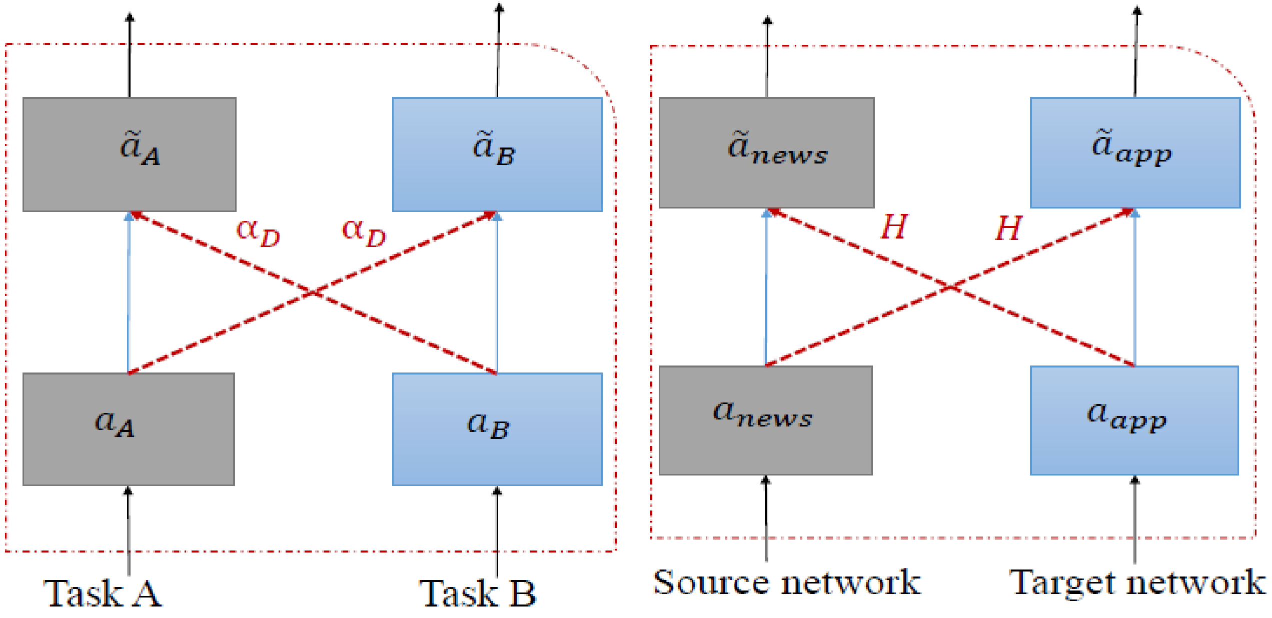
Ishan Misra et al. Cross-stitch networks for multi-task learning. CVPR'16

The Proposed Collaborative Cross Networks

- We propose a novel deep transfer learning method, Collaborative Cross Networks, to
  - Alleviate the data sparsity issue faced by the deep collaborative filtering
    - By transferring knowledge from a related source domain
  - Relax the strong assumptions faced by the existing cross-domain recommendation
    - By transferring knowledge via a matrix and enforcing sparsity-induced regularization

Idea 1: Using a matrix rather than a scalar (used in cross-stitch networks) to transfer

- We can relax the SA 1 assumption (equally important)

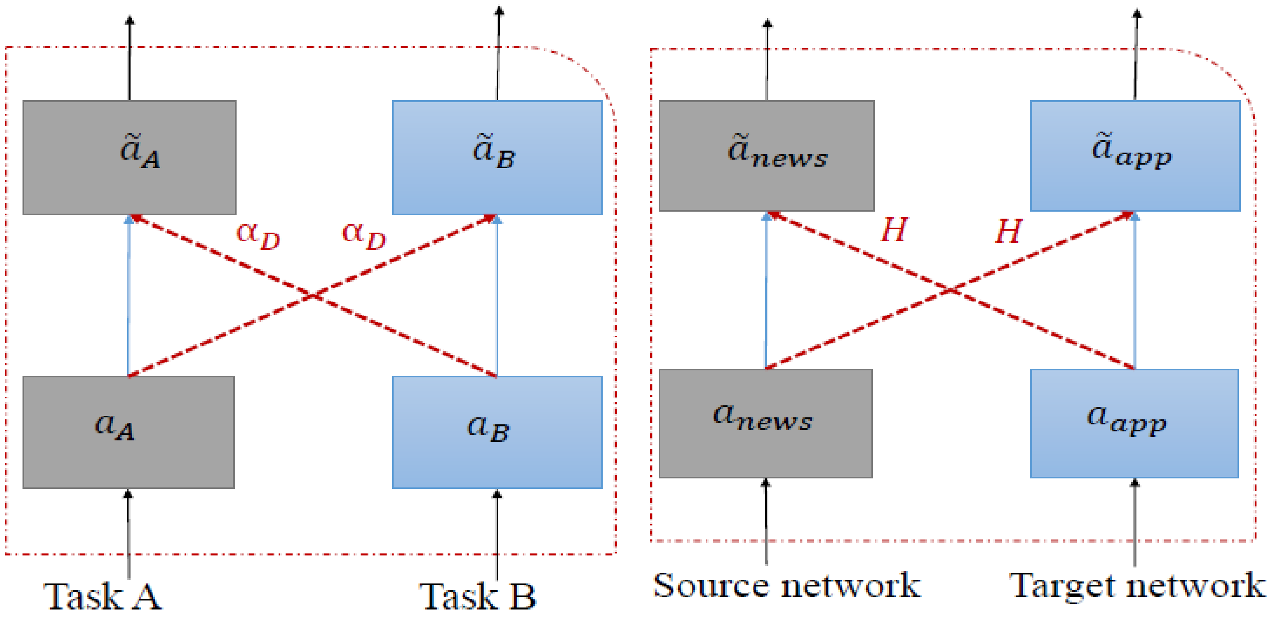


(a) Cross-stitch unit (b) Cross connection unit

$$\boldsymbol{a}_{app}^{l+1} = \sigma(\boldsymbol{W}_{app}^{l}\boldsymbol{a}_{app}^{l} + \boldsymbol{H}^{l}\boldsymbol{a}_{news}^{l}),$$
$$\boldsymbol{a}_{news}^{l+1} = \sigma(\boldsymbol{W}_{news}^{l}\boldsymbol{a}_{news}^{l} + \boldsymbol{H}^{l}\boldsymbol{a}_{app}^{l})$$

Idea 2: Selecting representations via sparsity-induced regularization

- We can relax the SA 2 assumption (all useful)

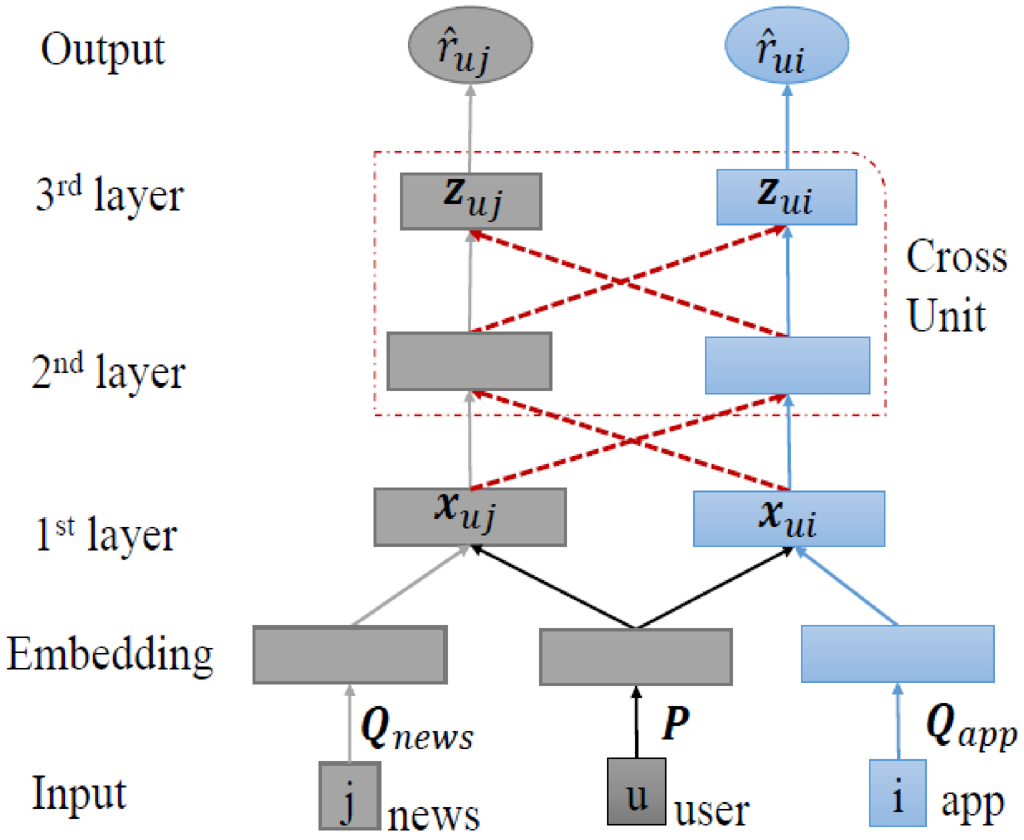


(a) Cross-stitch unit (b) Cross connection unit

$$\Omega(\boldsymbol{H}^l) = \lambda \sum_{i=1}^{r} \sum_{j=1}^{p} |h_{ij}|$$

The Architecture of the CoNet Model

- A version of three hidden layers and two cross units



Model Learning Objective

- The likelihood function (randomly sample negative examples)

$$L(\Theta|\mathcal{S}) = \prod_{(u,i)\in \boldsymbol{R}_T^+} \hat{r}_{ui} \prod_{(u,i)\in \boldsymbol{R}_T^-} (1 - \hat{r}_{ui}),$$

- The negative logarithm likelihood ←→ Binary cross-entropy loss

$$\mathcal{L} = - \sum_{(\boldsymbol{u},i)\in \mathcal{S}} r_{ui} \log \hat{r}_{ui} + (1 - r_{ui}) \log(1 - \hat{r}_{ui}),$$

- Stochastic gradient descent (and variants)

$$\Theta^{new} \leftarrow \Theta^{old} - \eta \frac{\partial L(\Theta)}{\partial \Theta}$$

Model Learning Objective (cont')

- Basic model (CoNet)

$$\mathcal{L}(\Theta) = \mathcal{L}_{app}(\Theta_{app}) + \mathcal{L}_{news}(\Theta_{news})$$

- Adaptive model (SCoNet)
  - Added the sparsity-induced penalty term into the basic model

- Typical deep learning library like TensorFlow (https://www.tensorflow.org) provides automatic differentiation which can be computed by chain rule in back-propagation.

Complexity Analysis

- Model analysis

$$\text{The model parameters } \Theta \text{ include } \{\boldsymbol{P}, (\boldsymbol{H}^l)_{l=1}^L\} \cup \{\boldsymbol{Q}_{app}, (\boldsymbol{W}_{app}^l, \boldsymbol{b}_{app}^l)_{l=1}^L, \boldsymbol{h}_{app}\} \cup \{\boldsymbol{Q}_{news}, (\boldsymbol{W}_{news}^l, \boldsymbol{b}_{news}^l)_{l=1}^L, \ \boldsymbol{h}_{news}\},$$

  - Linear with the input size and is close to the size of typical latent factors models and neural CF approaches
- Learning analysis
  - Update the target network using the target domain data and update the source network using the source domain data
  - The learning procedure is similar to the cross-stitch networks. And the cost of learning each base network is approximately equal to that of running a typical neural CF approach

Dataset and Evaluation Metrics

Dataset	#Users	Target Domain			Source Domain		
		#Items	#Interactions	Density	#Items	#Interactions	Density
Mobile	23,111	14,348	1,164,394	0.351%	29,921	617,146	0.089%
Amazon	80,763	93,799	1,323,101	0.017%	35,896	963,373	0.033%

- Mobile: Apps and News
- Amazon: Books and Movies

- A higher value (HR, NDCG, MRR) with lower cutoff *topK* indicates better performance

$$HR = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \delta(p_u \leq topK),$$

$$NDCG = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{\log 2}{\log(p_u + 1)},$$

$$MRR = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{1}{p_u}.$$

Baselines

- BPRMF: Bayesian personalized ranking
- MLP: Multilayer perceptron
- MLP++: Combine two MLPs by sharing the user embedding matrix
- CDCF: Cross-domain CF with factorization machines
- CMF: Collective MF
- CSN: The cross-stitch network

Baselines	Shallow method	Deep method
Single-domain	BPRMF [36]	MLP [13]
Cross-domain	CDCF [24], CMF [37]	MLP++, CSN [27]

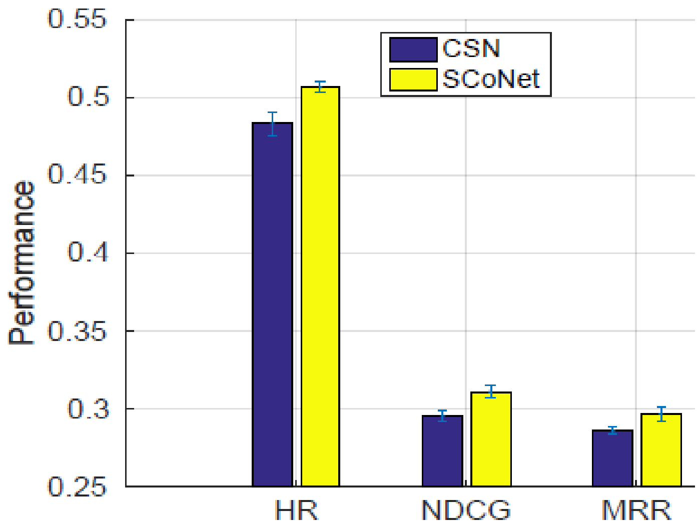
Comparing Different Approaches

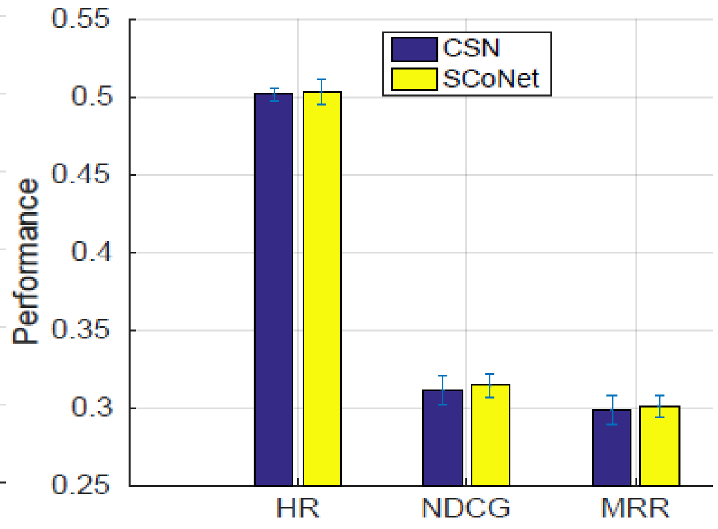
- CSN has some difficulty in benefitting from knowledge transfer on the Amazon since it is inferior to the non-transfer base network MLP
- The proposed model outperforms baselines on real-world datasets under three ranking metrics

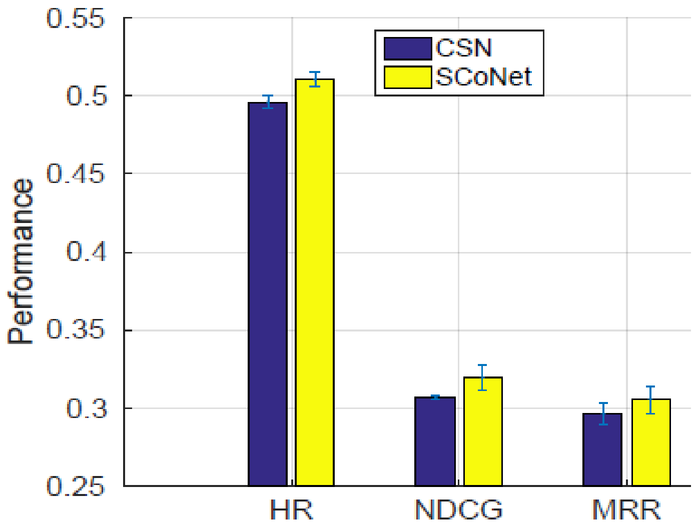
Dataset	Metric	BPRMF	CMF	CDCF	MLP	MLP++	CSN	CoNet	SCoNet	improve
Mobile	HR	.6175	.7879	.7812	.8405	.8445	.8458*	.8480	**.8583**	1.47%
	NDCG	.4891	.5740	.5875	.6615	.6683	.6733*	.6754	**.6887**	2.29%
	MRR	.4489	.5067	.5265	.6210	.6268	.6366*	.6373	**.6475**	1.71%
Amazon	HR	.4723	.3712	.3685	.5014	.5050*	.4962	.5167	**.5338**	5.70%
	NDCG	.3016	.2378	.2307	.3143	.3175*	.3068	.3261	**.3424**	7.84%
	MRR	.2971	.1966	.1884	.3113*	.3053	.2964	.3163	**.3351**	7.65%

Impact of Selecting Representations

- Configurations are {16, 32, 64} * 4, on Mobile data
- Naïve transfer learning approach may confront the negative transfer
- We demonstrate the necessity of adaptively selecting representations to transfer







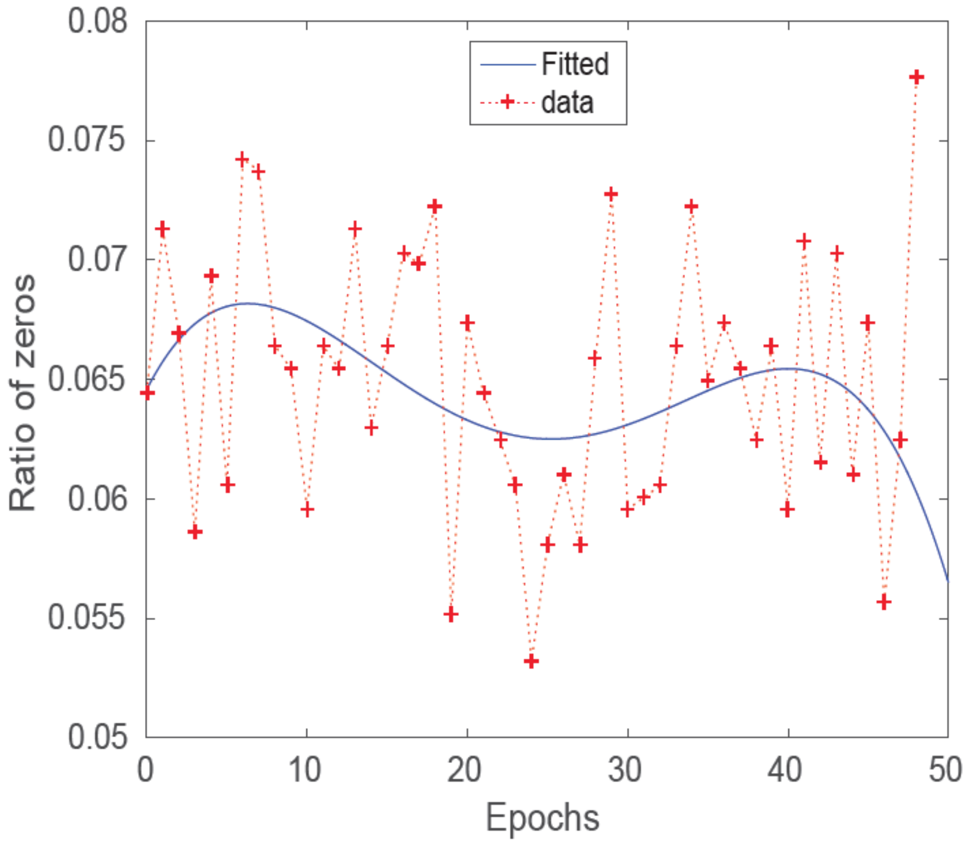
Benefit of Transferring Knowledge

- The more training examples we can reduce, the more benefit we can get from transferring knowledge
- Our model can reduce tens of thousands training examples by comparing with non-transfer methods without performance degradation

Dataset	Method	Reduction		HR	NDCG	MRR
		percent	amount			
Mobile	MLP	0%	0	.8405	.6615	.6210
	SCoNet	0%	0	.8547	.6802	.6431
		2.05%	23,031	.8439	.6640	.6238
		4.06%	45,468	.8347*	.6515*	.6115*
Amazon	MLP	0%	0	.5014	.3143	.3113
	SCoNet	0%	0	.5338	.3424	.3351
		1.11%	12,850	.5110	.3209	.3080*
		2.18%	25,318	.4946*	.3082*	.2968*

Analysis: Ratio of Zeros in Transfer Matrix H

- The percent of zero entries in transfer matrix is 6.5%
- A 4-order polynomial to robustly fit the data
- It may be better to transfer many instead of all representations



Conclusions and Future Works

- In general,
  - Neural/Deep approaches are better than shallow models,
  - Transfer learning approaches are better than non-transfer ones,
  - Shallow models are mainly based on MF techniques,
  - Deep models can be based on various NNs (MLP, CNN, RNN),
- Future works,
  - Data privacy
    - Source domain can not share the raw data, but model parameters
  - Transferable graph convolutional networks

Thanks!

Q & A

Acknowledgment: SIGIR Student Travel Grant